Please note that Cypress is an Infineon Technologies Company.
The document following this cover page is marked as “Cypress” document as this is the company that originally developed the product. Please note that Infineon will continue to offer the product to new and existing customers as part of the Infineon product portfolio.

Continuity of document content
The fact that Infineon offers the following product as part of the Infineon product portfolio does not lead to any changes to this document. Future revisions will occur when appropriate, and any changes will be set out on the document history page.

Continuity of ordering part numbers
Infineon continues to support existing part numbers. Please continue to use the ordering part numbers listed in the datasheet for ordering.



32-Bit Microcontroller TRAVEO™ T1G Family

S6J32XX Series Hardware Manual Platform Part

Document Number: 002-04854 Rev. *H

Cypress Semiconductor
An Infineon Technologies Company
198 Champion Court
San Jose, CA 95134-1709
www.infineon.com

© Cypress Semiconductor Corporation, 2014-2022. This document is the property of Cypress Semiconductor Corporation and its subsidiaries (“Cypress”). This document, including any software or firmware included or referenced in this document (“Software”), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you a personal, non-exclusive, nontransferable license (without the right to sublicense) (1) under its copyright rights in the Software (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units, and (2) under those claims of Cypress’s patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely for use with Cypress hardware products. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE OR ACCOMPANYING HARDWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. No computing device can be absolutely secure. Therefore, despite security measures implemented in Cypress hardware or software products, Cypress shall have no liability arising out of any security breach, such as unauthorized access to or use of a Cypress product. CYPRESS DOES NOT REPRESENT, WARRANT, OR GUARANTEE THAT CYPRESS PRODUCTS, OR SYSTEMS CREATED USING CYPRESS PRODUCTS, WILL BE FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (collectively, “Security Breach”). Cypress disclaims any liability relating to any Security Breach, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from any Security Breach. In addition, the products described in these materials may contain design defects or errors known as errata which may cause the product to deviate from published specifications. To the extent permitted by applicable law, Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. “High-Risk Device” means any device or system whose failure could cause personal injury, death, or property damage. Examples of High-Risk Devices are weapons, nuclear installations, surgical implants, and other medical devices. “Critical Component” means any component of a High-Risk Device whose failure to perform can be reasonably expected to cause, directly or indirectly, the failure of the High-Risk Device, or to affect its safety or effectiveness. Cypress is not liable, in whole or in part, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from any use of a Cypress product as a Critical Component in a High-Risk Device. You shall indemnify and hold Cypress, its directors, officers, employees, agents, affiliates, distributors, and assigns harmless from and against all claims, costs, damages, and expenses, arising out of any claim, including claims for product liability, personal injury or death, or property damage arising from any use of a Cypress product as a Critical Component in a High-Risk Device. Cypress products are not intended or authorized for use as a Critical Component in any High-Risk Device except to the limited extent that (i) Cypress’s published data sheet for the product explicitly states Cypress has qualified the product for use in a specific High-Risk Device, or (ii) Cypress has given you advance written authorization to use the product as a Critical Component in the specific High-Risk Device and you have signed a separate indemnification agreement.

Cypress, the Cypress logo, Spansion, the Spansion logo, and combinations thereof, WICED, PSoC, CapSense, EZ-USB, F-RAM, and Traveo are trademarks or registered trademarks of Cypress in the United States and other countries. For a more complete list of Cypress trademarks, visit cypress.com. Other names and brands may be claimed as property of their respective owners.

Preface

Thank you for your continued use of Cypress semiconductor products.
Read this manual and "Data Sheet" thoroughly before using products in this family.

Purpose of this manual and intended readers

This manual explains the functions and operations of this family and describes how it is used. The manual is intended for engineers engaged in the actual development of products using this family.

** This manual explains the configuration and operation of the peripheral functions, but does not cover the specifics of each device in the family.*
Users should refer to the respective data sheets of devices for device-specific details.

Table of Contents

CHAPTER 1: Platform Overview 12
1. Overview 13
2. Product Configuration 18
3. Configuration 19
4. Abbreviations 20
5. Memory Map 23
6. I/O Map 25
7. Legend 30
8. Caution 31

CHAPTER 2: CPU 33
1. Overview 34
2. Notes 35

CHAPTER 3: Operation Mode 40
1. Overview 41
2. Configuration 42
3. Explanation of Operation 43
4. Register 44

CHAPTER 4: Reset 47
1. Overview 48
2. Configuration 49
3. Operational Description 52
4. Registers 90

CHAPTER 5: Clock System 119
1. Overview 120
2. Configuration and Block Diagram 122
3. Operations 132
4. Setting Procedure Example 143
5. Registers 147
6. Usage Precautions 181

CHAPTER 6: Low Power Consumption 182
1. Overview 183
2. Configuration 184
3. Explanation of Operation 185
4. Operation Procedure 210
5. Registers 218
6. Other 591

CHAPTER 7: Low-voltage Detection 594
1. Overview 595

2. Configuration and Block Diagrams 597
3. Explanation of Operation 599
4. Setting Procedure Examples 601
5. Operation Examples 603
6. Precautions for Using This Device 606
CHAPTER 8: Clock Supervisor 608
1. Overview 609
2. Configuration 610
3. Explanation of Operation 614
4. Setting Procedure Example 616
5. Registers 618
6. Precautions for Using This Device 646
CHAPTER 9: Source Clock Timer 648
1. Overview 649
2. Configuration 650
3. Explanation of Operation 652
4. Setting Procedure Example 653
5. Registers 654
6. Precautions for Using this Device 697
CHAPTER 10: Real Time Clock 698
1. Overview 699
2. Configuration and Block Diagram 700
3. Operation of the Real Time Clock 703
4. Registers 709
5. Usage Precaution 743
CHAPTER 11: CR Calibration 746
1. Overview 747
2. Configuration 748
3. Explanation of Operation 749
4. Registers 751
CHAPTER 12: Backup RAM Interface 758
1. Overview 759
2. Configuration 761
3. Explanation of Operation 764
4. Setting Procedure 774
5. Registers 778
6. Precautions for Using Device 808
CHAPTER 13: External Interrupt 810
1. Overview 811
2. Configuration 812
3. Explanation of Operation 813
4. Setting Procedure Example 815
5. Registers 816
CHAPTER 14: Hardware Watchdog Timer 834
1. Overview 835
2. Configuration and Block Diagram 836
3. Explanation of Operation 837
4. Setting Procedure Example 854

5. Operation Examples 856
6. Register List 860
7. Precautions for Using This Device 885
CHAPTER 15: Software Watchdog Timer 888
1. Overview 889
2. Configuration and Block Diagram 890
3. Explanation of Operation 891
4. Setting Procedure Example 909
5. Operation Examples 910
6. Register List 913
7. Precautions for Using This Device 938
CHAPTER 16: TCRAM Interface 940
1. Overview 941
2. Configuration 942
3. Explanation of Operation 944
4. Setting Procedure Examples 949
5. Registers 953
CHAPTER 17: TCFLASH 984
1. Overview 985
2. Configuration 987
3. Explanation of Operation 993
4. Setting Procedure Examples 1006
5. Registers 1018
6. Others 1044
7. Support for Dual Port Flash Macro 1046
CHAPTER 18: WorkFLASH 1048
1. Overview 1049
2. Configuration 1050
3. Explanation of Operation 1054
4. Setting Procedure 1063
5. Registers 1070
6. Other 1103
7. Secure Hardware Extension (SHE) Support 1104
8. Dual Port FLASH Macro Support 1106
CHAPTER 19: BootROM Hardware Interface 1108
1. Overview 1109
2. Configuration 1110
3. Explanation of Operation 1111
4. Setting Procedure Example 1112
5. Registers 1113
6. Others 1126
CHAPTER 20: BootROM Software Interface 1128
1. Overview 1129
2. BootROM Markers 1130
3. BootROM Operation 1153
4. Notes 1162
CHAPTER 21: Exclusive Access Memory (EAM) 1164
1. Overview 1165

2. Configuration 1166
3. Explanation of Operation 1167
4. Operation Examples 1174
5. Memory Area 1181
CHAPTER 22: Interrupt Controller 1182
1. Overview 1183
2. Configuration 1184
3. Explanation of Operation 1185
4. Setting Procedure Examples 1198
5. Registers 1199
6. Others 1251
CHAPTER 23: Time Protection 1252
1. Overview 1253
2. Configuration 1254
3. Explanation of Operation 1255
4. Setting Procedure Examples 1257
5. Registers 1259
6. Others 1273
CHAPTER 24: Security 1274
1. Overview 1275
2. Security Scope and Access Restriction by the Security 1276
3. Using and Configuring the Security 1279
4. Registers 1292
CHAPTER 25: Memory Protection Unit for AXI of SHE 1328
1. Overview 1329
2. Configuration and Block Diagram 1330
3. Operation of the MPU AXI 1331
4. Registers 1337
5. Notes on Using MPU AXI 1356
CHAPTER 26: Secure Hardware Extension (SHE) 1358
1. Overview 1359
2. Block Diagram of the SHE Module 1360
3. Secure Hardware Extension Registers 1361
4. Operation of the Secure Hardware Extension 1422
5. Notes on Using SHE 1444
6. Enhanced Secure Hardware Extension 1450
7. References 1455
CHAPTER 27: DMA Controller 1456
1. Overview 1457
2. Configuration 1458
3. Operational Description 1460
4. Registers 1481
5. Additional Information 1512
CHAPTER 28: DMA COMPLEX SUBSYSTEM 1514
1. Overview 1515
2. Configuration and Block Diagram 1516
3. Operation of the DMA COMPLEX SUBSYSTEM 1518
4. Registers 1524

5. Overview in DMA COMPLEX SUBSYSTEM 1537
6. Configuration and Block Diagram in DMA COMPLEX SUBSYSTEM 1538
7. Operation of the 32-Bit Reload Timer in DMA COMPLEX SUBSYSTEM 1539
8. Registers in DMA COMPLEX SUBSYSTEM 1547
CHAPTER 29: MPU16 AHB 1556
1. Outline of the MPU16 AHB 1557
2. Configuration and Diagram 1558
3. MPU16 AHB Registers 1559
4. Operation of the MPU16 AHB 1574
CHAPTER 30: CAN FD Controller(MCAN3.0.1) 1578
1. Overview 1579
2. Configuration 1580
3. Explanation of Operations 1582
4. Setup Procedure Examples 1615
5. Registers 1638
6. Message RAM 1731
CHAPTER 31: CAN FD Controller(MCAN3.2) 1745
1. Overview 1746
2. Configuration 1747
3. Explanation of Operations 1750
4. Setup Procedure Examples 1781
5. Registers 1800
6. Message RAM 1896
CHAPTER 32: External Time Stamp Counter For CAN FD 1911
1. Overview 1912
2. Configuration 1913
3. Operations 1914
4. Example of the Operation 1919
5. Register 1920
CHAPTER 33: CAN FD Message RAM ECC Function 1927
1. Overview 1928
2. Configuration 1929
3. Interrupts 1931
4. Explanation of Operation 1932
5. Setup Procedure Examples 1934
6. Registers 1935
7. Precautions 1947
CHAPTER 34: CAN Prescaler 1949
1. Overview 1950
2. Configuration 1950
3. Explanation of Operation 1951
4. Setting Procedure 1952
5. Registers 1954
6. Restriction 1958
CHAPTER 35: Overview of Multi-function Serial Interface 1959
1. Overview of the UART (Asynchronous Serial Interface) 1960
2. Overview of the CSIO (Clock Synchronous Serial Interface) 1962
3. Overview of the LIN Interface (v2.1) (LIN Communication Control Interface (v2.1)) 1963

4. Overview of the I²C Interface (I²C Communication Control Interface) ... 1967
5. Block Diagram ... 1968
6. Registers of the Multi-Function Serial Interface ... 1971
CHAPTER 36: UART (Asynchronous Serial Interface) ... 1985
1. Overview of the UART (Asynchronous Serial Interface) ... 1986
2. Interrupts from the UART ... 1988
3. UART Operation ... 1999
4. Serial Timer Operation ... 2007
5. Test Mode ... 2016
6. Dedicated Baud Rate Generator ... 2017
7. Block Transfer ... 2023
8. Setup Procedure and Program Flow for Operation Mode 0 (Asynchronous Normal Mode) ... 2028
9. Setup Procedure and Program Flow for Operation Mode 1 (Asynchronous Multi-processor Mode) ... 2031
10. Registers of UART (Asynchronous Serial Interface) ... 2035
CHAPTER 37: CSIO (Clock Synchronous Serial Interface) ... 2089
1. Overview of the CSIO (Clock Synchronous Serial Interface) ... 2090
2. Interrupts of the CSIO (Clock Synchronous Serial Interface) ... 2091
3. Operation of the CSIO (Clock Synchronous Serial Interface) ... 2104
4. Serial Timer Operation ... 2138
5. Operation of Serial Chip Select ... 2149
6. Test Mode ... 2161
7. Dedicated Baud Rate Generator ... 2162
8. Block Transfer ... 2169
9. Registers of the CSIO (Clock Synchronous Serial Interface) ... 2174
CHAPTER 38: I²C Interface (I²C Communication Control Interface) ... 2247
1. Overview of the I²C Interface (I²C Communication Control Interface) ... 2248
2. I²C Interface Interrupts ... 2249
3. I²C Interface Operation ... 2253
4. Serial Timer Operation ... 2305
5. Dedicated Baud Rate Generator ... 2308
6. Block Transfer ... 2310
7. I²C Communication Operation Flowchart Examples ... 2315
8. I²C Interface Registers ... 2323
CHAPTER 39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) ... 2395
1. LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) ... 2396
2. LIN Interface (v2.1) Interrupts ... 2400
3. Serial Timer Operation ... 2427
4. Test Mode ... 2432
5. Dedicated Baud Rate Generator ... 2439
6. Block Transfer ... 2445
7. LIN Interface (V2.1) Operation ... 2450
8. Setup Procedure and Program Flow for Operation Mode 3 (LIN Communication Mode) ... 2487
9. Interface (V2.1) Registers ... 2496
CHAPTER 40: Base Timer ... 2583
1. Overview of the Base Timer ... 2584
2. Block Diagrams of the Base Timer ... 2587
3. Operations of the Base Timer ... 2591

4. 32-Bit Mode Operation 2593
5. Interrupts from the Base Timer 2595
6. Start of DMA Controller (DMAC) 2596
7. Registers of the Base Timer 2597
8. Notes on Using the Base Timer 2602
9. Base Timer Description by Function Mode 2604
CHAPTER 41: Base Timer I/O Selection Function 2683
1. Overview 2684
2. Configuration 2685
3. Explanation of Operation 2686
4. Registers 2690
CHAPTER 42: 32-bit Free-run Timer 2697
1. Overview of the 32-Bit Free-Run Timer 2698
2. Configuration and Block Diagram 2699
3. Explanation of the 32-Bit Free-Run Timer Operation 2700
4. Registers of the 32-Bit Free-Run Timer 2710
5. Precautions for Using This Device 2731
CHAPTER 43: 32-bit Input Capture 2733
1. Overview of the 32-Bit Input Capture 2734
2. Configuration and Block Diagram 2735
3. Explanation of 32-Bit Input Capture Operation 2736
4. Registers of the 32-Bit Input Capture 2739
5. Precautions for Using This Device 2751
CHAPTER 44: 32-bit Output Compare 2753
1. Overview 2754
2. Configuration 2755
3. Explanation of Operation 2756
4. Setting Procedure Example 2773
5. Registers 2774
6. Precautions for Using This Device 2796
CHAPTER 45: 32-bit Reload Timer 2797
1. Overview 2798
2. Configuration and Block Diagram 2799
3. Operation of the 32-Bit Reload Timer 2800
4. Registers 2807
CHAPTER 46: Reload Timer Simultaneous Soft Start 2815
1. Overview 2816
2. Configuration 2817
3. Explanation of Operation 2818
4. Registers 2819
CHAPTER 47: QPRC (Quadrature Position/Revolution Counter) 2823
1. Overview 2824
2. Configuration 2826
3. Operations 2827
4. Registers 2840
CHAPTER 48: External Interrupt Capture 2861
1. Overview 2862
2. Configuration 2863

3. Explanation of Operation 2864
4. Setting Procedure Examples 2867
5. Registers 2870
CHAPTER 49: CRC (Cyclic Redundancy Check) 2877
1. Overview 2878
2. Configuration and Diagram 2879
3. Operations 2880
4. Registers 2887
CHAPTER 50: I/O Port 2893
1. Overview 2894
2. Configuration and Block Diagrams 2895
3. Setting Procedure Examples 2897
4. Register List 2904
5. Precautions for Using This Device 2932
CHAPTER 51: Peripheral Protection Unit 2933
1. Overview 2934
2. Configuration and Block Diagram 2935
3. Operation of the PPU 2936
4. Registers 2944
5. Setup Steps of PPU Register 2962
CHAPTER 52: DDR High Speed SPI Controller 2965
1. Overview 2966
2. Configuration 2968
3. Serial Interface 2970
4. Operations of the DDRHSSPI 2990
5. Registers 3028
CHAPTER 53: System SRAM Module (SRAM_IF) 3107
1. Outline of SRAM_IF 3108
2. Configuration and Block Diagram 3109
3. Operation of SRAM_IF 3110
4. SRAM_IF Register Set 3112
5. How to use SRAM_IF 3124
CHAPTER 54: Bit-Band Unit 3125
1. Overview 3126
2. Explanation of Operation 3127

CHAPTER 55: Major Changes 3131
Revision History 4064
Document Revision History 4064

CHAPTER 1: Platform Overview



This chapter explains brief overview of the B/B platform.

1. **Overview**
2. **Product Configuration**
3. **Configuration**
4. **Abbreviations**
5. **Memory Map**
6. **I/O Map**
7. **Legend**
8. **Caution**

1. Overview

This section gives a brief overview of B/B platform.

B/B is the MCU platform which targets instrument cluster, graphics controller, gateway and body controller application area.

- Computation features

Cortex-R5F single core, 16K I-Cache.

Main Flash I/F which have advanced speculative prefetching algorithm.

- Software assist features

2 independent DMA controller instances offer parallelism and much of channels.

DMA with chain function using dedicated Reload Timer.

RMW Proxy Bridge unit minimizes register access for clearing the peripheral status bits.

WorkFlash which has EEPROM emulation.

Interrupt controller which supports up to 512 vectors by 512 IRQs.

- Bus structure

NIC-301 offers high performance, high parallelism and high flexibility.

Dedicated low latency bus for peripheral access.

- Safety features

Peripheral Protection Unit which controls access rights to each peripherals.

Memory protection units. (Cortex-R5F and DMAC)

High-speed and easy-to-use CRC calculation unit x 4.

Software and hardware watchdog timers

Clock supervisors which monitor clock frequencies of each clock sources.

TPU (Time Protection Unit) which verifies correct interrupt processing.

Error correction schemes in RAMs.

■ Security

Security features which protect Flash data from unauthorized reading and unauthorized debugger connections.

SHE engine (Secure boot, encryption/decryption/hash-generation accelerators, random value generator)

■ Power saving features

Rich sets of PSS (Power Saving State) modes including power gating options.

Peripheral operations are available in the PSS mode.

■ Timers

Rich set of timers. (Base Timer, Reload Timer, Freerun-Timer, Input capture, Output compare, QPRC)

■ Connectivity

CAN FD offers CAN protocol and highly flexible message buffering.

Multi-function serial I/F which has not only UART, CSIO and I2C but also LIN hardware assist feature

DDR-HSSPI to connect high-speed serial Flash memory by low pin count

■ Debug assist features

Following peripheral function can be forcedly stopped by entering debug mode.

(1) DMA
(2) Reload Timer in DMA Complex Subsystem
(3) Real Time Clock
(4) Hardware Watchdog Timer
(5) Software Watchdog Timer
(6) Time Protection Unit
(7) Source Clock Timer

Dedicated Reload Times can capture DMA control signal events like DREQ.
EICU can capture pin status of peripherals.

1.1. Function Overview

This section gives a brief overview of each function in B/B.

Function	Summary
Technology	55nm with embedded flash memory
CPU	ARM Cortex™-R5F (revision r1p3) ARMv7/Thumb® instruction set Up to 240MHz operation I-Cache 16KB (configurable by option, present or not) D-Cache 16KB (configurable by option, present or not) Double precision FPU (configurable by option, precision[single/double], present or not) MPU 16 region (configurable by option, 12 or 16 region) SECDED ECC for TCM ports (for RAM) VIC port for low latency interrupt Trace by ETM-R5 with 16KB trace buffer (buffer size configurable by option)
Debug	CoreSight™ Debug Dedicated bus master port for debugger access inside internal bus matrix.
Operation modes	User mode Serial flash writer mode Parallel flash writer mode
Clock controls	Internal fast CR oscillator 4MHz Internal slow CR oscillator 100KHz Main clock by quartz or ceramic resonator Sub clock by 32kHz RTC resonator Up to 4 internal PLL Up to 4 internal SSCG
Reset	Power-on reset LVD reset Reset by external pins Software reset
Low power consumption	PSS (Power saving state) with power shutdown PSS (Power saving state) without power shutdown
MPU	16 region MPU inside CPU core (configurable by option, 12 or 16 region) 16 region MPU for each DMA controller
Peripheral Protection Unit	Protection to peripherals from unauthorized access Individual protection setting per peripheral Return bus error response for the unauthorized access
Time protection unit	Maximum up to eight identical timers which can be used for execution time protection, locking time protection, inter-arrival time protection, or deadline protection 24-bit up-counters with programmable normal and overflow mode Global linear prescaler (1 to 64) to scale down the system clock frequency to required frequency, as input to individual timer prescaler Individual timer prescaler to support four different software-programmable frequencies(Fin, Fin/2, Fin/4, Fin/16)

Function	Summary
Clock supervisor	Clock frequency supervision for Main clock and Sub clock, PLL0-3 clock, SSCG PLL0-3 clock Reset/NMI generation for out-of bounds clock frequencies Clock stop supervision for CR clock and slow CR clock Reset generation for clock stop detection
Watchdog timers	Two types of Watchdog Timer (Hardware Watchdog Timer, Software Watchdog Timer) Both Watchdog Timer supports: 32-bit counter Window Watchdog functionality. Runs in RUN state or PSS state. Each state has its own configuration. Reset or Non-Maskable Interrupt (NMI) generation in Watchdog error Pre-warning interrupt can be generated in Watchdog error, before a Watchdog reset or NMI is generated. Hardware Watchdog Timer supports selection of two clock sources (CRclock or slow CRclock) Software Watchdog Timer supports selection of four clock sources (Main clock, Sub clock, CR clock or slow CR clock)
Low voltage detection (LVD)	Monitor following power supply : External 5V External 3.3V External 1.2V Internal power supply Internal power supply monitoring RAM retention
Main Flash memory (TCFlash)	Up to 4MB (product specification) SECDED ECC by 8-byte via AXI Read access via TCM port Read and write access via AXI TCM buffer for effective access via TCM port The ECC movement in TCM port is based on ECC setting inside the CPU
Work Flash memory	Up to 112KB (product specification) SECDED ECC by 8-byte Dedicated command sequencer for EEPROM emulation
TCRAM	Up to 128KB with SECDED ECC (product specification) SECDED ECC
System RAM	Up to 384KB (product specification) SECDED ECC
DMA Controller	Up to 2 instances 16 channels per instance Up to 512 DMA client interfaces Block / burst transfer Fixed / dynamic / round-robin priority scheme
Interrupt Controller (IRC)	Up to 512 sources for IRQ with 32 levels of priority Up to 32 sources for NMI with 16 level priority Register interface for software interrupt
External Interrupt	Up to 32 inputs for IRQ Up to 1 input for NMI Sensitive : H level, L level, pos-edge, neg-edge, both-edge
Exclusive Access Memory	48-byte memory for creating semaphore using exclusive load / store instructions.

Function	Summary
Read-Modify-Write proxy bridge	Bus bridge which convert a write access to an atomic read-modify-write operation to set one bit inside a byte.
CRC	4 CRC calculators (CRC32, CCITT)
Reload Timer	32-bit Reload Timer
Base Timer	16-bit PWM Timer 16-bit PPG Timer 16/32-bit Reload Timer 16/32-bit PWC Timer
Freerun Timer	32-bit Freerun Timer
Input Capture	32-bit Input Capture
Output Compare	32-bit Output Compare
QPRC (Quad Precision Counter)	Quad Position & Revolution Counter 16-bit position counter
Multi-function Serial Interface	UART mode CSIO (SPI) mode LIN mode I2C mode
CAN FD	Bosch M_CAN
SHE (Secure Hardware Extension)	Provides cryptographic functions and secure key storage Provides AES-128 encryption and decryption operations Supports the generation of the cipher-based message authentication code Provides a random number generation function It has a unique read-only Identification item (UID) It has a data interface for direct access to microcontroller memory locations
Power supply	5V external power supply (For I/O, analog power supply) 3.3V external power supply (For I/O, analog power supply) 1.2V external power supply (For logic) 1.2V internal LDO (For logic in Always-ON area)

2. Product Configuration

This section gives the product configuration.

Memories

Item	Size
Main Flash	Up to 4MB with SECDED ECC via AXI(product specification)
Work Flash	Up to 112KB with SECDED ECC (product specification)
TCRAM	Up to 128KB with SECDED ECC (product specification)
System RAM	Up to 384KB with SECDED ECC (product specification)
Backup RAM	Up to 8KB + 8KB with SECDED ECC (product specification)
BootROM	Up to 16KB (product specification)

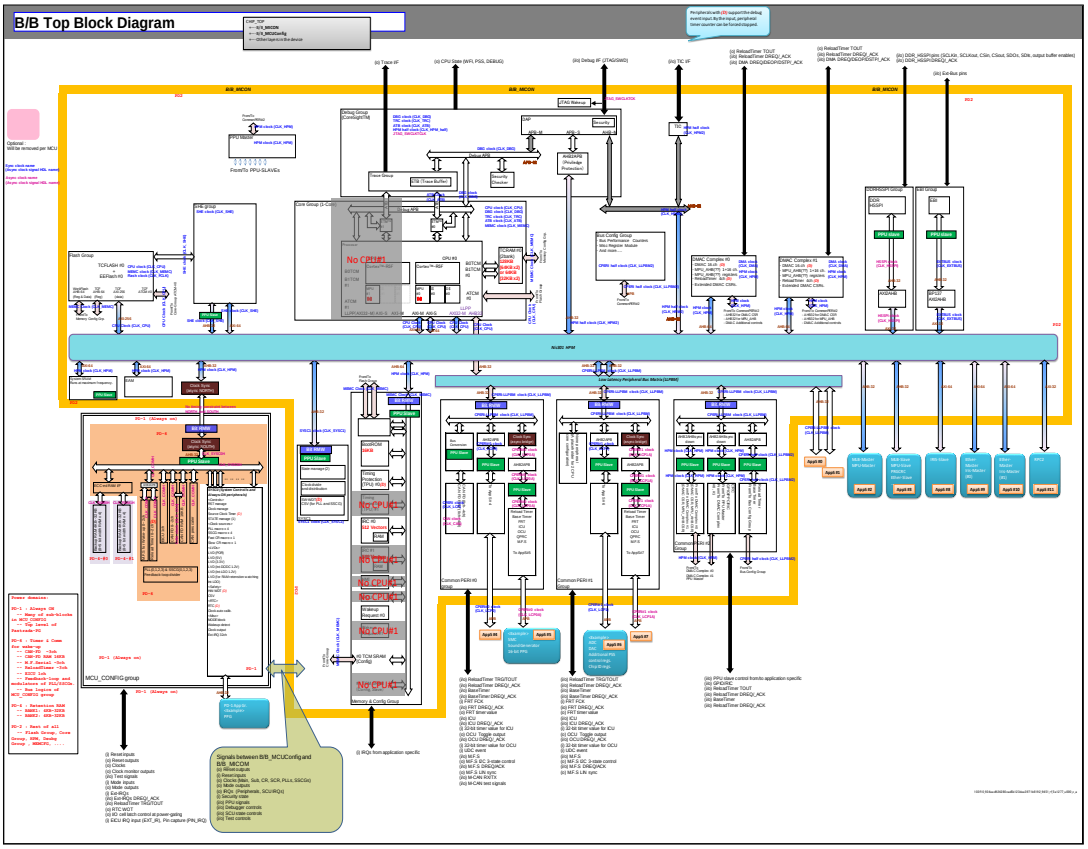
Functions

Item	Specification / Number of Channels
System clock	Up to 240MHz
CR oscillator	Yes (Fast and slow)
Sub oscillator for RTC	Yes
DMA Controller	Up to 2 instances 16 channels per instance Up to 512 DMA client interfaces
Reload Timer	Up to 48 channels + Up to 3 channels in Always-on area
Base Timer	Up to 36 channels
Freerun Timer	Up to 16 channels
Output Compare	Up to 16 instances (32 channels)
Input Capture	Up to 16 instances (32 channels)
QPRC	Up to 16 channels
CRC calculation unit	4 units
Multi-function Serial Interface	Up to 16 channels + Up to 3 channels in Always-on area
CAN-FD	Up to 5 channels + Up to 3 channels in Always-on area
External Interrupt	Up to 32 inputs for IRQ + Up to 1 input for NMI
Exclusive Access Memory (EAM)	Yes
Software Watchdog Timer	Yes
Hardware Watchdog Timer	Yes
Power-on Reset	Yes
Low Voltage Reset	Yes
Debug interface	JTAG, SWJP
GPIO	Up to 16 ports

3. Configuration

This section shows the block diagram of B/B platform.

Figure 3-1 Block Diagram



4. Abbreviations

This section explains abbreviations used in the document.

AHB -- Advanced High-performance Bus

AMBA™ -- Advanced Microcontroller Bus Architecture

APB -- Advanced Peripheral Bus

ATB -- AMBA Trace Bus

ATCM -- TCM-A port

AXI -- Advanced eXtensible Interface

B/B --Base Block

BBU -- Bit Banding Unit

BDR -- Boot Description Record

B0TCM TCM-B0 -- port

B1TCM TCM-B1 -- port

CAN -- Controller Area Network

CD -- Clock Domain

CPU -- Central Processing Unit

CR -- CR Oscillator

CRC -- Cyclic Redundancy Check

CSR -- Configuration and Status Register

CSV -- Clock Supervisor

DAP -- Debug Access Port

DED -- Double Error Detection

DMA -- Direct Memory Access

DMAC -- DMA Controller

EAM -- Exclusive Access Memory

ECC -- Error Checking Code

ETM -- Embedded Trace Macro

EXT-IRC -- External Interrupt Controller

FIQ -- Fast Interrupt Request

FPU -- Floating Point Unit

FRT -- Free Running Timer

GPIO -- General Purpose Input / Output

HPM -- High Performance Matrix

HW-WDT -- Hardware Watchdog Timer

ICU -- Input Capture Unit

IRC -- Interrupt Controller

IRQ -- Interrupt Request

ISR -- Interrupt Service Routine

JTAG -- Joint Test Action Group

LLPP -- Low Latency Peripheral Port

LVD -- Low Voltage Detector

MCU -- Microcontroller Unit

MFS -- Multi-Function Serial Interface

NF -- Noise Filter

NMI -- Non-Maskable Interrupt

OSC -- Oscillator

OCU -- Output Compare Unit

PD -- Power Domain

PLL -- Phase Locked Loop

PONR -- Power-ON Reset

PPC -- Port Pin Configuration

PPU -- Peripheral Protection Unit

PSS -- Power Saving State

RAM -- Random Access Memory

RIC -- Resource Input Configuration

RLT -- Reload Timer

ROM -- Read Only Memory

RUN -- Run State

SCT -- Source Clock Timer

SDR -- Security Description Record

SEC -- Single Error Correction

SRAM -- Static RAM

SSCG -- Spread Spectrum Clock Generator

SW-WDT -- Software Watchdog Timer

SWD -- Serial Wire Debug

SYSC -- System Controller

TCFLASH -- TCM Flash Memory

TCM -- Tightly Coupled Memory

TCRAM -- TCM SRAM

TPU -- Time Protection Unit

VIC -- Vectored Interrupt Controller

WDR -- Watchdog Description Record

WDT -- Watchdog Timer

WorkFLASH -- Work Flash Memory

(product specification) *1 -- Device-specific information

1 Notes:

- *For details, please refer to the Datasheet or Hardware Manual of Device.*
- *Originally, device-specific information should be written in this part.*
  However, in order to avoid the state which Platform Hardware Manual has different contents in each devices, the expression of "(product specification)" are used.

5. Memory Map

This section shows memory map of B/B platform.

Memory Map

	Address		size			module	
	START	END	[Mbyte]	[Kbyte]	[Kbyte]	group	part
	0000_0000	007F_FFFF	256	32768	8192	Internal area for CR5 Complex	TCRAM 8MB@Max
	0080_0000	00FF_FFFF			8192		TCFLASH 8MB@Max
	0100_0000	017F_FFFF			8192		AXI_FLASH 16MB@Max
	0180_0000	019F_FFFF			2048		
	01A0_0000	01BF_FFFF			2048		
	01C0_0000	01FB_FFFF			3840		
	01FC_0000	01FD_FFFF			128		
	01FE_0000	01FF_FFFF			128		
	0200_0000	027F_FFFF		32768	8192	Shared Flash and memory area	System SRAM 8MB@Max
	0280_0000	0280_0FFF			4		Exclusive Access Memory
	0280_1000	03FF_FFFF			24572		Reserved
	0400_0000	05FF_FFFF		32768	32768		AXI_SLAVE_CORE0
	0600_0000	07FF_FFFF		32768	32768		AXI_SLAVE_CORE1
	0800_0000	09FF_FFFF		32768	32768		AXI_SLAVE_CORE2
	0A00_0000	0BFF_FFFF		32768	32768		AXI_SLAVE_CORE3
	0C00_0000	0DFF_FFFF		32768	32768		Reserved
	0E00_0000	0E2F_FFFF		32768	3072		WORK_FLASH 4M@Max
	0E30_0000	0E3F_FFFF			1024		
	0E40_0000	0E7F_FFFF			4096		Reserved
	0E80_0000	0E8F_FFFF			512		BACKUP_RAM 512K@Max
	0E88_0000	0EFF_FFFF			7680		Reserved
	0F00_0000	0FFF_FFFF			16384		Reserved
	1000_0000	1FFF_FFFF	256	262144	262144	Ext-Bus area	EBI_MEMORY(SRAM/FLASH)
	2000_0000	2FFF_FFFF	256	262144	262144	AppS / EBI area	EBI_MEMORY(SDRAM) / AppS area
1G	3000_0000	3FFF_FFFF	256	262144	262144	Reserved	Reserved
	4000_0000	5FFF_FFFF	512	524288	524288	AppS area	AppS area
2G	6000_0000	7FFF_FFFF	512	524288	524288	AppS area	AppS area
	8000_0000	8FFF_FFFF	256	262144	262144	HSSPI MEM area	HSSPI0_MEMORY
	9000_0000	9FFF_FFFF	256	262144	262144	AppS area	AppS area
	A000_0000	AFFF_FFFF	256	262144	262144	Reserved / **AppS-LLPP-AXI32 (when needed)**	Reserved / **AppS-LLPP-AXI32 (when needed)**
	B000_0000	B7FF_FFFF	128	131072	131072	Peri area	EMEM GROUP (reg) System SRAM (reg) MCUOFG GROUP SYSC1 MEMCFG GROUP DEBUG GROUP SHE_CONFIG BUS CONFIG GROUP (via CPERI#2) CPERI#0, #1, #2 **AppS#0--1, #4--7** **Bit RMW alias**
3G	B800_0000	BFFF_FFFF	128	131072	131072	AppS area	AppS area
	C000_0000	EFFF_FFFF	768	786432	786432	Reserved	Reserved
	F000_0000	FFFE_DFFF	256	262144	262072	BootROM/ERRCFG area	Reserved
	FFFE_E000	FFFE_EFFF			4		ERRCFG
	FFFE_F000	FFFE_FFFF			4		
4G	FFFF_0000	FFFF_FFFF			64		BootROM

Notes:

- *Access to reserved areas is responded with bus error.*
- *AXI_SLAVE_CORE1~3 are reserved.*
- *AppS areas are used in each MCU implementations.*
- *Only the CPU core can access 0000_0000 ~ 01FF_FFFF. Bus masters other than the CPU core cannot access the region.*
- *Internal area of CR5 complex (0000_0000 ~ 01FF_FFFF) is mapped to AXI_SLAVE_CORE0. All bus masters can access to internal area of CR5 complex via AXI_SLAVE_CORE0.*

Default Memory Attribute for the CPU Core

	Address range		Instruction memory type		Data memory type		Execute Never
	START	END	Instruction cache enabled	Instruction cache disabled	Data cache enabled	Data cache disabled	
1G	0000_0000	3FFF_FFFF	Normal, Cacheable, Non-shared	Normal, Cacheable, Non-shared	Normal, WBWA Cacheable, Non-shared	Normal, Non-cacheable, Shared	Instruction execution permitted
2G	4000_0000	5FFF_FFFF	Normal, Cacheable, Non-shared	Normal, Cacheable, Non-shared	Normal, WT Cacheable, Non-shared	Normal, Non-cacheable, Shared	Instruction execution permitted
	6000_0000	7FFF_FFFF	Normal, Cacheable, Non-shared	Normal, Cacheable, Non-shared	Normal, Non-Cacheable, shared	Normal, Non-cacheable, Shared	Instruction execution permitted
3G	8000_0000	9FFF_FFFF	-	-	Non-shared Device	Non-shared Device	Execute Never
	A000_0000	B2FF_FFFF	-	-	Shared Device	Shared Device	Execute Never
	B300_0000	BFFF_FFFF					
	C000_0000	EFFF_FFFF	-	-	Strongly Ordered	Strongly Ordered	Execute Never
4G	F000_0000	FFFF_FFFF	Normal Non-cacheable only if HIVECS is TRUE	Normal Non-cacheable only if HIVECS is TRUE	Strongly Ordered	Strongly Ordered	Instruction execution only permitted if HIVECS is TRUE

When MPU attribute of Cortex-R5F is configured as "Normal", store buffer inside Cortex-R5F can operate and write data can be merged. To avoid influence of this data merger, MPU attribute "Device" or "Strongly Ordered" should be used.

MPU attribute "Device" or "Strongly Ordered" must be used for areas below, to avoid this influence.

- Backup RAM area (BACKUP_RAM) [0E80_0000 ~ 0E87_FFFF]
- Peripheral area (Peri area) [B000_0000 ~ B7FF_FFFF]
- Error Config area (ERRCFG) [FFFE_E000 ~ FFFE_FFFF]
- AppS area [B800_0000 ~ BFFF_FFFF]

MPU attribute "Device" or "Strongly Ordered" is required for accesses to areas below, in particular situation.

- FLASH Memory (when writing commands)

6. I/O Map

This section shows I/O map of B/B platform.

I/O Map (1)

	START Address	END Address	Function	Power Domain	Size (DEC)	Size(HEX)	PPU # start	PPU # end
For AppS	B000_0000	B00F_FFFF	Reserved / AppS area for LLPP-AXI32 (1MB)	-	1048576	100000		
EMEM GROP (reg)	B010_0000	B010_03FF	EBI registers	PD2	1024	400	0	0
	B010_0400	B010_0FFF	Reserved	-	3072	C00		
	B010_1000	B010_13FF	DDR_HSSPI	PD2	1024	400	1	1
	B010_1400	B010_7FFF	Reserved	-	27648	6C00		
System SRAM	B010_8000	B010_80FF	System SRAM registers	PD2	256	100	2	2
Reserved	B010_8100	B01F_FFFF	Reserved	-	1015552	F7F00		
Reserved	B020_0000	B02F_FFFF	Reserved	-	1048576	100000		
SYSC1	B030_0000	B03F_FFFF	System Controller #1 **Base address of the Peripheral s:** B030_0000 SYSC1 B030_8000 SWDT **PPU Number of the peripherals:** 4 SYSC1 5 SWDT 6-15: Reserved	PD2	1048576	100000	4	15
MEM Config	B040_0000	B04F_FFFF	MEMORY_CONFIG_GROUP **Base address of the Peripherals :** B040_0000 IRC0 B040_8000 TPU0 B041_0000 TCRAM CSR B041_1000 TCFlash CSR B041_2000 WFlash CSR **Followings are mapped in Error Config Area:** -- IRC0 (IRC0_NMIVASBR) -- IRC0 (IRC_NMIVASBR) -- BootROM I/F **PPU Number of the peripherals:** 16 TCRAM CSR 17 TCFlash CSR 18 WFlash CSR 19 TPU0 20 BootROM I/F (*) 21 IRC0 ***: Readable in privilege, whatever PPU protection setting is.**	PD2	1048576	100000	16	21
DEBUG GROUP	B050_0000	B05F_FFFF	DEBUG_GROUP **Base address offsets of the Peripherals :** +0x0_0000 DAP ROM table +0x0_1000 ETB +0x0_2000 CTI#4 +0x0_3000 TPIU +0x0_4000 TRACE FUNNEL +0x8_0000 CR5_RomTable +0x9_0000 CORE0 +0x9_2000 Reserved +0x9_8000 CTI#0 +0x9_9000 Reserved +0x9_C000 ETM0 +0x9_D000 Reserved +0xC_0000 Security Checker **- No PPU channel is assigned to DEBUG GROUP**	PD2	1048576	100000		
MCU Config	B060_0000	B06F_FFFF	MCU_CONFIG_GROUP [note] BackupRAM is located in 0E80_0000 -- From the CPU core, it is accessed via AXI-M-64 port of CR5 **Base Address : Peripherals in MCU_CONFIG_GROUP** B060_0000 SYSC0 B060_0800 MODEC B060_C000 HWDT B061_0000 Reserved B061_8000 RTC B062_0000 Ext-IRQ B064_0000 App.Gr AlwaysON B066_0000 reserved(App.Gr PD6) B068_0000 BackupRAM CSR B068_8000 EICU B068_8400 CR_CALIBRATION B068_8800 MCG_IRS B068_8C00 CAN_PRESCALER B069_0000 ReloadTimer MCU_Config_Ch.0 B069_0400 ReloadTimer MCU_Config_Ch.1 B069_0800 ReloadTimer MCU_Config_Ch.2 B06A_8000 M.F.S MCU_Config_Ch.0 B06A_8400 M.F.S MCU_Config_Ch.1 B06A_8800 M.F.S MCU_Config_Ch.2 B06C_0000 CAN FD MCU_Config_Ch.0 B06D_0000 CAN FD MCU_Config_Ch.1 B06E_0000 CAN FD MCU_Config_Ch.2 PD1 PD6 **PPU Number : Peripherals in MCU_CONFIG_GROUP** 32 RTC 33 SCT(Slow CR) 34 SCT(Fast CR) 35 SCT(Main clock) 36 SCT(Sub clock) 37 EICU 38 CR_Calibration 39 RLT(MCU_Config_Ch.0) 40 RLT(MCU_Config_Ch.1) 41 RLT(MCU_Config_Ch.2) 42 MCG_IRS 43 CAN Prescaler 44 MFS(MCU_Config_Ch.0) 45 MFS(MCU_Config_Ch.1) 46 MFS(MCU_Config_Ch.2) 47 CAN FD(MCU_Config_Ch.0) 48 CAN FD(MCU_Config_Ch.1) 49 CAN FD(MCU_Config_Ch.2) 50 BackupRAM CSR 51 SYSC0 52 HWDT 53 Ext-IRQ 54 Reserved 55 MODEC 56-62: Reserved	PD1, PD6 (refer informations in left for details)	1048576	100000	32	62
Reserved	B070_0000	B07F_FFFF	Reserved (1MB)		1048576	100000		
Bit RMW alias	B080_0000	B0FF_FFFF	Bit RMW alias for MCU Config (Covers B060_0000 -- B06F_FFFF)	PD6	8388608	800000		
Bit RMW alias	B100_0000	B10F_FFFF	Bit RMW alias for SYSC1 (Covers B030_0000 -- B031_FFFF)	PD2	1048576	100000		
Bit RMW alias	B110_0000	B11F_FFFF	Bit RMW alias for MEMC (Covers B040_0000 -- B041_FFFF)	PD2	1048576	100000		
Reserved	B120_0000	B1FF_FFFF	Reserved (14MB)	-	14680064	E00000		
SHE_CONFIG	B200_0000	B20F_FFFF	SHE configuration registers (1MB) B200_0000 SHE B200_0400 MPU_AXI_SHE B200_0800 SHE	PD2	1048576	100000	63	63
For AppS	B210_0000	B3FF_FFFF	Reserved / AppS area for LLPP-AXI32 (31MB)	-	32505856	1F00000		

R01L04

Note:

- *Access to reserved areas is responded with bus error.*

I/O Map (2)

Reserved	B400_0000	B46F_FFFF	Reserved (7MB)	-	7340032	700000	
Common PERI#2	B470_0000	B470_3FFF	DMAC #0 registers	PD2	16384	4000	64
1MB	B470_4000	B470_7FFF	DMAC #1 registers	PD2	16384	4000	65
	B470_8000	B470_FFFF	Reserved	-	32768	8000	
	B471_0000	B471_0FFF	MPU for DMAC#0	PD2	4096	1000	66
	B471_1000	B471_1FFF	MPU for DMAC#1	PD2	4096	1000	67
	B471_2000	B471_2FFF	Reserved	-	8192	2000	
	B471_4000	B471_4FFF	DMAC Complex #0 registers (Additional registers, RLTs)	PD2	4096	1000	68
	B471_5000	B471_5FFF		PD2	4096	1000	69
	B471_6000	B471_7FFF	Reserved	-	8192	2000	
	B471_8000	B471_83FF	CRC#0	PD2	1024	400	70
	B471_8400	B471_87FF	CRC#1	PD2	1024	400	71
	B471_8800	B471_8BFF	CRC#2	PD2	1024	400	72
	B471_8C00	B471_8FFF	CRC#3	PD2	1024	400	73
	B471_9000	B473_7FFF	Reserved	-	126976	1F000	
	B473_8000	B473_FFFF	GPIO	PD2	32768	8000	74
	B474_0000	B474_7FFF	PPC	PD2	32768	8000	75
	B474_8000	B474_FFFF	RIC	PD2	32768	8000	76
	B475_0000	B475_7FFF	Reserved	-	32768	8000	77
	B475_8000	B475_FFFF	Reserved	-	163840	28000	
	B478_0000	B478_03FF	Base Timer ch 24	PD2	1024	400	112
	B478_0400	B478_07FF	Base Timer ch 25	PD2	1024	400	113
	B478_0800	B478_0BFF	Base Timer ch 26	PD2	1024	400	114
	B478_0C00	B478_0FFF	Base Timer ch 27	PD2	1024	400	115
	B478_1000	B478_13FF	Base Timer ch 28	PD2	1024	400	116
	B478_1400	B478_17FF	Base Timer ch 29	PD2	1024	400	117
	B478_1800	B478_1BFF	Base Timer ch 30	PD2	1024	400	118
	B478_1C00	B478_1FFF	Base Timer ch 31	PD2	1024	400	119
	B478_2000	B478_23FF	Base Timer ch 32	PD2	1024	400	120
	B478_2400	B478_27FF	Base Timer ch 33	PD2	1024	400	121
	B478_2800	B478_2BFF	Base Timer ch 34	PD2	1024	400	122
	B478_2C00	B478_2FFF	Base Timer ch 35	PD2	1024	400	123
	B478_3000	B478_7FFF	Reserved	-	20480	5000	
	B478_8000	B478_83FF	Reload Timer ch 32	PD2	1024	400	160
	B478_8400	B478_87FF	Reload Timer ch 33	PD2	1024	400	161
	B478_8800	B478_8BFF	Reload Timer ch 34	PD2	1024	400	162
	B478_8C00	B478_8FFF	Reload Timer ch 35	PD2	1024	400	163
	B478_9000	B478_93FF	Reload Timer ch 36	PD2	1024	400	164
	B478_9400	B478_97FF	Reload Timer ch 37	PD2	1024	400	165
	B478_9800	B478_9BFF	Reload Timer ch 38	PD2	1024	400	166
	B478_9C00	B478_9FFF	Reload Timer ch 39	PD2	1024	400	167
	B478_A000	B478_A3FF	Reload Timer ch 40	PD2	1024	400	168
	B478_A400	B478_A7FF	Reload Timer ch 41	PD2	1024	400	169
	B478_A800	B478_ABFF	Reload Timer ch 42	PD2	1024	400	170
	B478_AC00	B478_AFFF	Reload Timer ch 43	PD2	1024	400	171
	B478_B000	B478_B3FF	Reload Timer ch 44	PD2	1024	400	172
	B478_B400	B478_B7FF	Reload Timer ch 45	PD2	1024	400	173
	B478_B800	B478_BBFF	Reload Timer ch 46	PD2	1024	400	174
	B478_BC00	B478_BFFF	Reload Timer ch 47	PD2	1024	400	175
	B478_C000	B478_FBFF	Reserved	-	15360	3C00	
	B478_FC00	B478_FFFF	Misc registers	PD2	1024	400	82
	B479_0000	B47F_FFFF	To Bus Config Group -- offset 0x0000 - 0x03FF: Bus performance counter -- offset 0x0400 - : Reserved	PD2	458752	70000	83

Note:

- *Access to reserved areas is responded with bus error.*

I/O Map (3)

Common PERI #0 512KB	B480_0000	B480_03FF	M.F.Serial ch.0	PD2	1024	400	176	
	B480_0400	B480_07FF	M.F.Serial ch.1	PD2	1024	400	177	
	B480_0800	B480_0BFF	M.F.Serial ch.2	PD2	1024	400	178	
	B480_0C00	B480_0FFF	M.F.Serial ch.3	PD2	1024	400	179	
	B480_1000	B480_13FF	M.F.Serial ch.4	PD2	1024	400	180	
	B480_1400	B480_17FF	M.F.Serial ch.5	PD2	1024	400	181	
	B480_1800	B480_1BFF	M.F.Serial ch.6	PD2	1024	400	182	
	B480_1C00	B480_1FFF	M.F.Serial ch.7	PD2	1024	400	183	
	B480_2000	B480_7FFF	Reserved	-	24576	6000		
	B480_8000	B480_83FF	BaseTimer ch.0	PD2	1024	400	88	
	B480_8400	B480_87FF	BaseTimer ch.1	PD2	1024	400	89	
	B480_8800	B480_8BFF	BaseTimer ch.2	PD2	1024	400	90	
	B480_8C00	B480_8FFF	BaseTimer ch.3	PD2	1024	400	91	
	B480_9000	B480_93FF	BaseTimer ch.4	PD2	1024	400	92	
	B480_9400	B480_97FF	BaseTimer ch.5	PD2	1024	400	93	
	B480_9800	B480_9BFF	BaseTimer ch.6	PD2	1024	400	94	
	B480_9C00	B480_9FFF	BaseTimer ch.7	PD2	1024	400	95	
	B480_A000	B480_A3FF	BaseTimer ch.8	PD2	1024	400	96	
	B480_A400	B480_A7FF	BaseTimer ch.9	PD2	1024	400	97	
	B480_A800	B480_ABFF	BaseTimer ch.10	PD2	1024	400	98	
	B480_AC00	B480_AFFF	BaseTimer ch.11	PD2	1024	400	99	
	B480_B000	B480_FFFF	Reserved	-	20480	5000		
	B481_0000	B481_03FF	Reload Timer ch.0	PD2	1024	400	128	
	B481_0400	B481_07FF	Reload Timer ch.1	PD2	1024	400	129	
	B481_0800	B481_0BFF	Reload Timer ch.2	PD2	1024	400	130	
	B481_0C00	B481_0FFF	Reload Timer ch.3	PD2	1024	400	131	
	B481_1000	B481_13FF	Reload Timer ch.4	PD2	1024	400	132	
	B481_1400	B481_17FF	Reload Timer ch.5	PD2	1024	400	133	
	B481_1800	B481_1BFF	Reload Timer ch.6	PD2	1024	400	134	
	B481_1C00	B481_1FFF	Reload Timer ch.7	PD2	1024	400	135	
	B481_2000	B481_23FF	Reload Timer ch.8	PD2	1024	400	136	
	B481_2400	B481_27FF	Reload Timer ch.9	PD2	1024	400	137	
	B481_2800	B481_2BFF	Reload Timer ch.10	PD2	1024	400	138	
	B481_2C00	B481_2FFF	Reload Timer ch.11	PD2	1024	400	139	
	B481_3000	B481_33FF	Reload Timer ch.12	PD2	1024	400	140	
	B481_3400	B481_37FF	Reload Timer ch.13	PD2	1024	400	141	
	B481_3800	B481_3BFF	Reload Timer ch.14	PD2	1024	400	142	
	B481_3C00	B481_3FFF	Reload Timer ch.15	PD2	1024	400	143	
	B481_4000	B481_7FFF	Reserved	-	16384	4000		
	B481_8000	B481_83FF	QPRC ch.0	PD2	1024	400	192	
	B481_8400	B481_87FF	QPRC ch.1	PD2	1024	400	193	
	B481_8800	B481_8BFF	QPRC ch.2	PD2	1024	400	194	
	B481_8C00	B481_8FFF	QPRC ch.3	PD2	1024	400	195	
	B481_9000	B481_93FF	QPRC ch.4	PD2	1024	400	196	
	B481_9400	B481_97FF	QPRC ch.5	PD2	1024	400	197	
	B481_9800	B481_9BFF	QPRC ch.6	PD2	1024	400	198	
	B481_9C00	B481_9FFF	QPRC ch.7	PD2	1024	400	199	
	B481_A000	B481_FFFF	Reserved	-	24576	6000		
	B482_0000	B482_03FF	FRT ch.0	PD2	1024	400	208	
	B482_0400	B482_07FF	FRT ch.1	PD2	1024	400	209	
	B482_0800	B482_0BFF	FRT ch.2	PD2	1024	400	210	
	B482_0C00	B482_0FFF	FRT ch.3	PD2	1024	400	211	
	B482_1000	B482_13FF	FRT ch.4	PD2	1024	400	212	
	B482_1400	B482_17FF	FRT ch.5	PD2	1024	400	213	
	B482_1800	B482_1BFF	FRT ch.6	PD2	1024	400	214	
	B482_1C00	B482_1FFF	FRT ch.7	PD2	1024	400	215	
	B482_2000	B482_7FFF	Reserved	-	24576	6000		
	B482_8000	B482_83FF	ICU pair ch.0	PD2	1024	400	224	
	B482_8400	B482_87FF	ICU pair ch.1	PD2	1024	400	225	
	B482_8800	B482_8BFF	ICU pair ch.2	PD2	1024	400	226	
	B482_8C00	B482_8FFF	ICU pair ch.3	PD2	1024	400	227	
	B482_9000	B482_93FF	ICU pair ch.4	PD2	1024	400	228	
	B482_9400	B482_97FF	ICU pair ch.5	PD2	1024	400	229	
	B482_9800	B482_9BFF	ICU pair ch.6	PD2	1024	400	230	
	B482_9C00	B482_9FFF	ICU pair ch.7	PD2	1024	400	231	
	B482_A000	B482_FFFF	Reserved	-	24576	6000		
	B483_0000	B483_03FF	OCU pair ch.0	PD2	1024	400	240	
	B483_0400	B483_07FF	OCU pair ch.1	PD2	1024	400	241	
	B483_0800	B483_0BFF	OCU pair ch.2	PD2	1024	400	242	
	B483_0C00	B483_0FFF	OCU pair ch.3	PD2	1024	400	243	
	B483_1000	B483_13FF	OCU pair ch.4	PD2	1024	400	244	
	B483_1400	B483_17FF	OCU pair ch.5	PD2	1024	400	245	
	B483_1800	B483_1BFF	OCU pair ch.6	PD2	1024	400	246	
	B483_1C00	B483_1FFF	OCU pair ch.7	PD2	1024	400	247	
	B483_2000	B483_FBFF	Reserved	-	56320	DC00		
	B483_FC00	B483_FFFF	Misc registers	PD2	1024	400	80	
	B484_0000	B484_FFFF	APPS#5 area (PSEL x 64 ; 1KB per one PSEL)	PD2	65536	10000	A5	A5+N5-1
	B485_0000	B485_FFFF	APPS#4 area (PSEL x 64 ; 1KB per one PSEL)	PD2	65536	10000	A4	A4+N4-1
	B486_0000	B487_FFFF	Reserved	-	131072	20000		

Note:

- *Access to reserved areas is responded with bus error.*

I/O Map (4)

Common PERI #1 512KB	B488_0000	B488_03FF	M.F.Serial ch.8	PD2	1024	400	184	
	B488_0400	B488_07FF	M.F.Serial ch.9	PD2	1024	400	185	
	B488_0800	B488_0BFF	M.F.Serial ch.10	PD2	1024	400	186	
	B488_0C00	B488_0FFF	M.F.Serial ch.11	PD2	1024	400	187	
	B488_1000	B488_13FF	M.F.Serial ch.12	PD2	1024	400	188	
	B488_1400	B488_17FF	M.F.Serial ch.13	PD2	1024	400	189	
	B488_1800	B488_1BFF	M.F.Serial ch.14	PD2	1024	400	190	
	B488_1C00	B488_1FFF	M.F.Serial ch.15	PD2	1024	400	191	
	B488_2000	B488_7FFF	Reserved	-	24576	6000		
	B488_8000	B488_83FF	BaseTimer ch.12	PD2	1024	400	100	
	B488_8400	B488_87FF	BaseTimer ch.13	PD2	1024	400	101	
	B488_8800	B488_8BFF	BaseTimer ch.14	PD2	1024	400	102	
	B488_8C00	B488_8FFF	BaseTimer ch.15	PD2	1024	400	103	
	B488_9000	B488_93FF	BaseTimer ch.16	PD2	1024	400	104	
	B488_9400	B488_97FF	BaseTimer ch.17	PD2	1024	400	105	
	B488_9800	B488_9BFF	BaseTimer ch.18	PD2	1024	400	106	
	B488_9C00	B488_9FFF	BaseTimer ch.19	PD2	1024	400	107	
	B488_A000	B488_A3FF	BaseTimer ch.20	PD2	1024	400	108	
	B488_A400	B488_A7FF	BaseTimer ch.21	PD2	1024	400	109	
	B488_A800	B488_ABFF	BaseTimer ch.22	PD2	1024	400	110	
	B488_AC00	B488_AFFF	BaseTimer ch.23	PD2	1024	400	111	
	B488_B000	B488_FFFF	Reserved	-	20480	5000		
	B489_0000	B489_03FF	Reload Timer ch.16	PD2	1024	400	144	
	B489_0400	B489_07FF	Reload Timer ch.17	PD2	1024	400	145	
	B489_0800	B489_0BFF	Reload Timer ch.18	PD2	1024	400	146	
	B489_0C00	B489_0FFF	Reload Timer ch.19	PD2	1024	400	147	
	B489_1000	B489_13FF	Reload Timer ch.20	PD2	1024	400	148	
	B489_1400	B489_17FF	Reload Timer ch.21	PD2	1024	400	149	
	B489_1800	B489_1BFF	Reload Timer ch.22	PD2	1024	400	150	
	B489_1C00	B489_1FFF	Reload Timer ch.23	PD2	1024	400	151	
	B489_2000	B489_23FF	Reload Timer ch.24	PD2	1024	400	152	
	B489_2400	B489_27FF	Reload Timer ch.25	PD2	1024	400	153	
	B489_2800	B489_2BFF	Reload Timer ch.26	PD2	1024	400	154	
	B489_2C00	B489_2FFF	Reload Timer ch.27	PD2	1024	400	155	
	B489_3000	B489_33FF	Reload Timer ch.28	PD2	1024	400	156	
	B489_3400	B489_37FF	Reload Timer ch.29	PD2	1024	400	157	
	B489_3800	B489_3BFF	Reload Timer ch.30	PD2	1024	400	158	
	B489_3C00	B489_3FFF	Reload Timer ch.31	PD2	1024	400	159	
	B489_4000	B489_7FFF	Reserved	-	16384	4000		
	B489_8000	B489_83FF	QPRC ch.8	PD2	1024	400	200	
	B489_8400	B489_87FF	QPRC ch.9	PD2	1024	400	201	
	B489_8800	B489_8BFF	QPRC ch.10	PD2	1024	400	202	
	B489_8C00	B489_8FFF	QPRC ch.11	PD2	1024	400	203	
	B489_9000	B489_93FF	QPRC ch.12	PD2	1024	400	204	
	B489_9400	B489_97FF	QPRC ch.13	PD2	1024	400	205	
	B489_9800	B489_9BFF	QPRC ch.14	PD2	1024	400	206	
	B489_9C00	B489_9FFF	QPRC ch.15	PD2	1024	400	207	
	B489_A000	B489_FFFF	Reserved	-	24576	6000		
	B48A_0000	B48A_03FF	FRT ch.8	PD2	1024	400	216	
	B48A_0400	B48A_07FF	FRT ch.9	PD2	1024	400	217	
	B48A_0800	B48A_0BFF	FRT ch.10	PD2	1024	400	218	
	B48A_0C00	B48A_0FFF	FRT ch.11	PD2	1024	400	219	
	B48A_1000	B48A_13FF	FRT ch.12	PD2	1024	400	220	
	B48A_1400	B48A_17FF	FRT ch.13	PD2	1024	400	221	
	B48A_1800	B48A_1BFF	FRT ch.14	PD2	1024	400	222	
	B48A_1C00	B48A_1FFF	FRT ch.15	PD2	1024	400	223	
	B48A_2000	B48A_7FFF	Reserved	-	24576	6000		
	B48A_8000	B48A_83FF	ICU pair ch.8	PD2	1024	400	232	
	B48A_8400	B48A_87FF	ICU pair ch.9	PD2	1024	400	233	
	B48A_8800	B48A_8BFF	ICU pair ch.10	PD2	1024	400	234	
	B48A_8C00	B48A_8FFF	ICU pair ch.11	PD2	1024	400	235	
	B48A_9000	B48A_93FF	ICU pair ch.12	PD2	1024	400	236	
	B48A_9400	B48A_97FF	ICU pair ch.13	PD2	1024	400	237	
	B48A_9800	B48A_9BFF	ICU pair ch.14	PD2	1024	400	238	
	B48A_9C00	B48A_9FFF	ICU pair ch.15	PD2	1024	400	239	
	B48A_A000	B48A_FFFF	Reserved	-	24576	6000		
	B48B_0000	B48B_03FF	OCU pair ch.8	PD2	1024	400	248	
	B48B_0400	B48B_07FF	OCU pair ch.9	PD2	1024	400	249	
	B48B_0800	B48B_0BFF	OCU pair ch.10	PD2	1024	400	250	
	B48B_0C00	B48B_0FFF	OCU pair ch.11	PD2	1024	400	251	
	B48B_1000	B48B_13FF	OCU pair ch.12	PD2	1024	400	252	
	B48B_1400	B48B_17FF	OCU pair ch.13	PD2	1024	400	253	
	B48B_1800	B48B_1BFF	OCU pair ch.14	PD2	1024	400	254	
	B48B_1C00	B48B_1FFF	OCU pair ch.15	PD2	1024	400	255	
	B48B_2000	B48B_FBFF	Reserved	-	56320	DC00		
	B48B_FC00	B48B_FFFF	Misc registers	PD2	1024	400	81	
	B48C_0000	B48C_FFFF	APPS#7 area (PSEL x 64 ; 1KB per one PSEL)	PD2	65536	10000	A7	A7+N7-1
	B48D_0000	B48D_FFFF	APPS#6 area (PSEL x 64 ; 1KB per one PSEL)	PD2	65536	10000	A6	A6+N6-1
	B48E_0000	B48F_FFFF	Reserved	-	131072	20000		
Common Peri#0	B490_0000	B490_FFFF	CAN FD ch0	PD2	65536	10000	256	
	B491_0000	B491_FFFF	CAN FD ch1	PD2	65536	10000	257	
	B492_0000	B492_FFFF	CAN FD ch2	PD2	65536	10000	258	
	B493_0000	B493_FFFF	CAN FD ch3	PD2	65536	10000	259	
	B494_0000	B494_FFFF	CAN FD ch4	PD2	65536	10000	260	
	B495_0000	B497_FFFF	Reserved (196K)	-	196608	30000	261	263
Reserved	B498_0000	B4BF_FFFF	Reserved (2.5MB)	-	2621440	280000		
Bit RMW alias	B4C0_0000	B4FF_FFFF	Bit RMW alias for CommonPERI#0 (Covers B490_0000 -- B497_FFFF)	PD2	4194304	400000		
					16777216	1000000		
					16777216	1000000		
Bit RMW alias	B700_0000	B77F_FFFF	Bit RMW alias for CommonPERI#2 (Covers B470_0000 -- B47F_FFFF)	PD2	8388608	800000		
Bit RMW alias	B780_0000	B7BF_FFFF	Bit RMW alias for CommonPERI#0 (Covers B480_0000 -- B487_FFFF)	PD2	4194304	400000		
Bit RMW alias	B7C0_0000	B7FF_FFFF	Bit RMW alias for CommonPERI#1 (Covers B488_0000 -- B48F_FFFF)	PD2	4194304	400000		

N4 : Number of PSEL area for Apps#4
N5 : Number of PSEL area for Apps#5
N6 : Number of PSEL area for AppS#6
N7: Number of PSEL area for AppS#7

A4: 264
A5: 264 + N4
A6: 264 + N4 + N5
A7 : 264 + N4 + N5 + N6

N0123 : Number of PPU slaves defined for AppS#0/1/2/3 (outside of PF)

---=== PPU ch assign for AppS#0/1/2/3 (outside PF) ===---
PPU start #: 264 + N4 + N5 + N6 + N7
PPU end #: 264 + N4 + N5 + N6 + N7 + N0123 -1

I/O Map (5)

					Size (DEC)	Size(HEX)
Error Config	FFFE_E000	FFFE_E3FF	IRC0 (IRC0_NMIVASBR)	PD2	1024	400
	FFFE_E400	FFFE_F7FF	Reserved	-	5120	1400
	FFFE_F800	FFFE_FBFF	IRC (IRC_NMIVASBR) Mirror *	PD2	1024	400
	FFFE_FC00	FFFE_FFFF	BootROM I/F	PD2	1024	400
			Total		8192	2000

*: Only CPU0 can access to IRC0 in this area. This area is reserved for masters other than CPU0.

Note:

- *Access to reserved areas is responded with bus error.*

Notes:

- *For the base address, please refer to the Hardware Manual of Device.*
- *For the register names, please refer to the Hardware Manual of Platform.*

7. Legend

This section describes the legend used throughout this manual.

7.1. Read/Write Attribute (R/W Attribute)

R/W Attribute	Description
R	Readable
R0	Read value is always 0
R1	Read value is always 1
RX	Read access returns unknown value
W	Writable
W0	Write value must be 0
W1	Write value must be 1
WX	Write invalid (write operation has no influence)
/ (slash mark)	Writable and readable (read value is equal to write value)
, (comma)	Different behavior in read and write (read value is different with write value)

Example

R/W : Readable and writable [read value is equal to write value]

R,W0 : Readable, Write value must be 0 [read value is different with write value]

Notes:

- *In case factors that influences read value are following, R/W attribute is described with / (slash mark).*
  - *Bit set by bit set register operation*
  - *Bit clear by bit clear register operation*

7.2. Protection Attribute

Protection Attribute	Description
RP	Readable in privilege mode
RS	Readable, with protected sequence
RPS	Readable in privilege mode, with protected sequence
WP	Writeable in privilege mode
WS	Writeable, with protected sequence
WPS	Writeable in privilege mode, with protected sequence
- (hyphen)	No protection in both read and write
/ (slash mark)	Both read and write has protection left side of slash mark shows read protection right side of slash mark shows write protection

8. Caution

8.1. How to Use Reset Factor Register

When the reset occurs, it is possible for a user to know which reset factor generated in reading a reset factor register. Then, the user can do initialization or return processing by his own program according to the factor. However, it is assumed that another reset occurs again in this program. Therefore, when the reset factor register clears a factor, pay attention to the following item.

・ Clear a reset factor in the end of the reset factor processing.

Even if a different reset occurs on the way of the factor processing, the older reset factor remains just as it is inside the register.

・ Clear in turn from the low reset factor of the order of priority.

Even if the different reset occurs while processing, the high reset factor of the order of priority which you should process can be left.

・ Clear all reset registers by lump-sum,

It is possible to process early when clearing by the writing in of Word.

These are the reset factor registers which adopt above explanation.

SYSC_RSTCAUSEUR, SYSC_EXCSVRSTCAUSEUR,

SYSC_PDRSTCAUSEUR,SYSC_RSTCAUSEBT, SYSC_EXCSVRSTCAUSEBT,

SYSC_PDRSTCAUSEBT, HWDG_RSTCAUSE, SWDG_RSTCAUSE

8.2. Caution in Setting and clearing Bit

The AHB bus doesn't have the signal which shows being the period of RMW. Therefore, the feature that is "controlling the operation of hardware in writing 0 or 1 ", like interrupt request flags which were used at our F2MC/FR family, are changed into "the set bit register " "the clear bit register " in the ARM type CPU including TRAVEO™ T1G. But, there are some peripheral circuits which cannot support yet. In that case, when you access to the other bit which is contained the register that also have the above bit, please set to the value which doesn't have an influence for the bit.

[ex1] SomeRegister has "0 write clear bit" in bit 1 and you want to clear bit 2 in SomeRegister.

```
SomeRegister=(SomeRegister && 0b1111_1011) || 0b0000_0010; // no influence writing 1 to bit1
```

[ex2] SomeRegister has "0 write clear bit" in bit 1 and you want to set bit 2 in SomeRegister

```
SomeRegister=SomeRegister || 0b0000_0110; // no influence writing 1 to bit1
```

We will change the feature that is "controlling the operation of hardware in writing 0 or 1 " to "the set bit register " "the clear bit register " step by step. Please use this feature from now on.

These are the reset factor registers which adopt above explanation.

RTC_WTCR, RTC_WTSR, QECR, QICRL, QICRH

And there are the registers which also have "the set bit register " "the clear bit register ".

CU_CUCR1, BURIF_EECSR, BURIF_TTCR, BURIF_TICR, FCR0, TCCS, TECCS, IR(CAN FD)

CHAPTER 2: CPU

This chapter provides an overview of the CPU and related notes.

1. **Overview**
2. **Notes**

CPU-TXXPT03P01R01L05-E1-XX

1. Overview

This section explains the features of the CPU.

Features

- The processor is a Cortex-R5™ Single Core (1 CPU) processor.
- The CPU is equipped with 16 KB of instruction cache with ECC support.
- The CPU is equipped with 16 KB of data cache with ECC support.
- The CPU is equipped with a memory protection unit (MPU) with 16 regions.
- The CPU is equipped with a double precision floating-point unit (FPU).
- TCFLASH is connected to the ATCM of the CPU.
- TCRAM is connected to the BTCM of the CPU.
- The ATCM supports 64-bit ECC.
- The BTCM supports 32-bit ECC.
- B0TCM and B1TCM are used in interleaving.
- The CPU is equipped with the 64-bit AXI Master and Slave interfaces.
- The CPU is equipped with the 32-bit AXI Master interface for the low latency peripheral port (LLPP) and with the 32-bit AHB-Lite Master interface.
- The CPU is equipped with a VIC port, which enables direct retrieval of interrupt vector addresses.
- For details, see the Cortex™-R5 Revision:r1p2 Technical Reference Manual (ARM DDI 0460D), an ARM manual. For example, there are such notes.
  - If a write to a peripheral on one interface causes a side effect on a peripheral on a different interface, there is no implicit ordering to ensure the side effect is observed by a subsequent access to the second peripheral, even if both are in Device-type memory.
  - In this situation, you must perform a read from the first peripheral to ensure that the write has completed, followed by a DMB to ensure ordering before performing the second access.
  - On the Cortex-R5 processor, a DMB alone is sufficient to force this ordering, but this is not architectural and cannot be relied on in the general case.

2. Notes

This section provides notes on the CPU.

Low Latency Peripheral Port

The low latency peripheral port is used as the port for accessing peripherals.

The Peripheral Interface Region Register, Normal AXI Peripheral Interface Region, Virtual AXI Peripheral Interface Region, AHB Peripheral Interface Region: En bit must not be changed.

WFE Instruction

The WFE instruction is not supported and must not be used.

VIC Port

Interrupts (IRQs) are processed through the VIC port.

To enable the VIC port, the System Control Register:VE, which is an internal core register, is set to 1 by the BootROM software.

If the VIC port is disabled, interrupts cannot be processed properly, so the VIC port must not be disabled.

Flag Clear

Note that the hardware operation of a write access to a register which has status flags may be later than program operations of software. The delay results from the fact that the write accesses and their signals are operated through multiple buses.

For example, when a software program intends to return from interrupt software routine (ISR) after clearing the interrupt flag, the return instruction may practically be executed before the completion of the write access as for hardware operation. That is, the CPU execution may immediately jump to the ISR again after returning from the ISR because the flag is not cleared.

To avoid such a phenomenon, the execution of Data Memory Barrier (DMB) instruction between the write access and the return instruction is recommended for the program. The return instruction is never executed before the completion of DMB execution.

It should be considered that the method takes a number of execution cycles and that it may cause some influence to application performance. For example, one time of the DMB execution is enough after a number of continuous flag clear operations.

Build Option

The build option of Cortex-R5 is shown below.

Table 2-1 Cortex-R5 Build Option

Feature	Options	Sub-options	Configuration
Number of CPUs	Single-CPU (no redundancy)	-	Single-CPU (no redundancy)
	Redundant CPU	-	
	Twin-CPU (no redundancy)	-	
	Split/Lock	Safety-mode (redundancy)	
		Performance-mode (twin CPU)	
Instruction cache	No I-Cache	-	-
	I-Cache included	No error checking	64-bit ECC error checking
		Parity error checking	
		64-bit ECC error checking	
		4KB (4x1KB ways)	16KB(4x4KB ways)
		8KB (4x2KB ways)	
		16KB (4x4KB ways)	
		32KB (4x8KB ways)	
		64KB (4x16KB ways)	
Data cache	No D-Cache	-	-
	D-Cache included	No error checking	32-bit ECC error checking
		Parity error checking	
		32-bit ECC error checking	
		4KB (4x1KB ways) 8KB (4x2KB ways) 16KB (4x4KB ways) 32KB (4x8KB ways) 64KB (4x16KB ways)	16KB(4x4KB ways)
ATCM	No ATCM ports	-	-
	One ATCM port	No error checking	64-bit ECC error checking
		32-bit ECC error checking	
		64-bit ECC error checking	
		4KB, 8KB, 16KB, 32KB, 64KB, 128KB, 256KB, 512KB, 1MB, 2MB, 4MB, or 8MB	8MB

Feature	Options	Sub-options	Configuration
BTCM	No BTCM ports	-	-
	One BTCM port (B0TCM)	No error checking	-
		32-bit ECC error checking	
		64-bit ECC error checking	
		4KB, 8KB, 16KB, 32KB, 64KB, 128KB, 256KB, 512KB, 1MB, 2MB, 4MB, or 8MB	-
	Two BTCM ports (B0TCM and B1TCM)	No error checking	32-bit ECC error checking
		32-bit ECC error checking	
		64-bit ECC error checking	
		2x2KB, 2x4KB, 2x8KB, 2x16KB, 2x32KB, 2x64KB, 2x128KB, 2x256KB, 2x512KB, 2x1MB, 2x2MB, or 2x4MB	Product Specification
		Interleaved on 64-bit granularity in memory	Interleaved on 64-bit granularity in memory
		Adjacent in memory	
Instruction endianness	Little-endian	-	Little-endian
	Pin-configured	Little-endian	
		Big-endian	
Floating point (VFP)	No FPU	-	Product Specification
	FPU included	Full implementation	
		Single-precision only	
MPU	No MPU	-	16 MPU regions
	MPU included	12 MPU regions	
		16 MPU regions	
TCM bus parity	No TCM address and control bus parity	-	No TCM address and control bus parity
	TCM address and control bus parity generated	-	
AXI bus ECC/parity on AXI-master, AXI-slave (if included) and ACP (if included)	No AXI bus ECC/parity	-	No AXI bus ECC/parity
	AXI bus ECC/parity generated/ checked	-	
Bus ECC/parity on AXI peripheral port and AHB peripheral port (if included)	No peripheral port bus ECC/parity	-	No peripheral port bus ECC/parity
	Peripheral port bus ECC/parity generated/checked	-	
Breakpoints	2-8 breakpoint register pairs	-	8
Watchpoints	1-8 watchpoint registers	-	8

Feature	Options	Sub-options	Configuration
ATCM at reset	Disabled	-	Base address configured
	Enabled	Base address 0x0	
		Base address configured	
BTCM at reset	Disabled	-	Base address 0x0
	Enabled	Base address configured	
		Base address 0x0	
Peripheral ID RevAnd field	Any 4-bit value	-	0000
AXI slave interface	No AXI-slave	-	AXI-slave included
	AXI-slave included	-	
TCM Hard Error Cache	No TCM Hard Error Cache	-	TCM Hard Error Cache included
	TCM Hard Error Cache included	-	
Non-Maskable FIQ Interrupt	Disabled (FIQ can be masked by software)	-	Enabled
	Enabled	-	
Parity type	Odd parity	-	Odd parity
	Even parity	-	
AXI coherency port (ACP)	No ACP	-	No ACP
	ACP included	-	
AHB peripheral port	AXI peripheral port only	-	-
	AXI and AHB peripheral ports	AHB peripheral port region size: 4KB, 8KB, 16KB, 32KB, 64KB, 128KB, 256KB, 512KB, 1MB, 2MB, 4MB, or 8MB. 16MB, 32MB, 64MB, 128MB, 256MB, 512MB, 1GB, 2GB, 4GB	64MB
		AHB peripheral port base address: any size-aligned address	0xB400_0
AXI peripheral interface region size	4KB, 8KB, 16KB, 32KB, 64KB, 128KB, 256KB, 512KB, 1MB, 2MB, 4MB, or 8MB. 16MB, 32MB, 64MB, 128MB, 256MB, 512MB, 1GB, 2GB, 4GB	-	64MB
AXI peripheral interface base address	Any size-aligned address	-	0xB000_0
Virtual AXI peripheral interface region size	4KB, 8KB, 16KB, 32KB, 64KB, 128KB, 256KB, 512KB, 1MB, 2MB, 4MB, or 8MB. 16MB, 32MB, 64MB, 128MB, 256MB, 512MB, 1GB, 2GB, 4GB	-	-

Feature	Options	Sub-options	Configuration
Virtual AXI peripheral interface base address	Any size-aligned address	-	-

CHAPTER 3: Operation Mode

This chapter explains the operation modes.

1. **Overview**
2. **Configuration**
3. **Explanation of Operation**
4. **Register**

1. Overview

This section provides an overview of the operation mode.

The mode controller determines the device operation mode. This device has the following operation modes.

User Mode

User Mode executes the user program from the Flash.

Board Mode

Board Mode has below sub-mode.

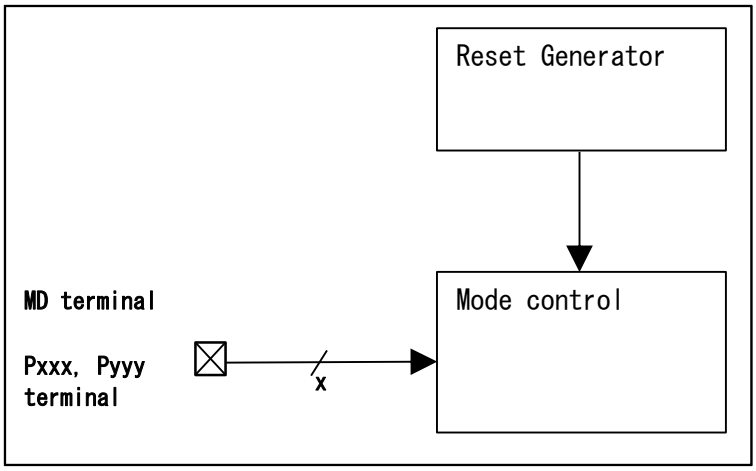
Serial Writer Mode

This mode executes program using a serial writer.

2. Configuration

The following figure is a block diagram of the Mode Controller.

Figure 2-1 Mode Controller



3. Explanation of Operation

This section explains the operation of the operation mode.

Determination of Operation Mode

Operation Mode is determined at the release of the reset. The MD pin input level has to stay unchanged at all time.

4. Register

This section lists the operation mode registers.

Table 4-1 List of Operation Mode Registers

Abbreviated Register Name	Register Name	Reference
MODEC_MODER	Mode Register	4.1

Table 4-2 shows the register address map.

Table 4-2 Operation Mode Register Map

Offset	Register Name/Initial Value
0x0000_0000	MODEC_MODER *0000000_00000000_000*0000_00000000

Notes:

- **: Initial value "0" or "1" according to the setting*
- *Initial value depends on the MD pin status during the start-up period.*

Table 4-3 List of Register Mirror Area

Address of Register Mirror Area	Mirrored Peripheral	Mirror Area Behavior	PPU
0x0000_0900 to 0x0000_09FF 0x0000_0A00 to 0x0000_0AFF 0x0000_0B00 to 0x0000_0BFF 0x0000_0C00 to 0x0000_0CFF 0x0000_0D00 to 0x0000_0DFF 0x0000_0E00 to 0x0000_0EFF 0x0000_0F00 to 0x0000_0FFF	MODEC	Accessing this region results to the access to MODEC register area with following offset. Offset: (addr & 0000_007F)	This area is covered by PPU #55 as well as the mirrored peripheral.

4.1. Mode Register (MODEC_MODER)

Mode register (MODEC_MODER) shows states of the operation mode and the MD pin determined during the device start-up.

Bit	31	30	29	28	27	26	25	24
Field	USER MODE	Reserved						
R/W ATTRIBUTE	R,WX	R0,WX						
Protection Attribute	-							
Initial Value	*	0000000						

Bit	23	22	21	20	19	18	17	16
Field	Reserved							
R/W ATTRIBUTE	R0,WX							
Protection Attribute	-							
Initial Value	00000000							

Bit	15	14	13	12	11	10	9	8
Field	Reserved			MD	Reserved			
R/W ATTRIBUTE	R0,WX			R,WX	R0,WX			
Protection Attribute	-							
Initial Value	000			*	0000			

Bit	7	6	5	4	3	2	1	0
Field	Reserved							
R/W ATTRIBUTE	R0,WX							
Protection Attribute	-							
Initial Value	00000000							

* Initial value depends on the MD pin status during the start-up period.

[bit31] USERMODE: User Mode Bit

This bit indicates whether the operation mode is in Board Mode or User Mode.

Bit	Description
0	Board Mode
1	User Mode

[bit30:13] Reserved: Reserved Bits

[bit12] MD: Mode Bit

Bit	Description
0	MD pin input value is "0"
1	MD pin input value is "1"

[bit11:0] Reserved: Reserved Bits

CHAPTER 4: Reset

This chapter explains the reset.

1. **Overview**
2. **Configuration**
3. **Operational Description**
4. **Registers**

RESET-TXXPT03P01R01L11-E1-XX

1. Overview

This product has the following reset factors. Depending on each factor, a reset is issued for initialization within the device. The reset factors are shown below.

Table 1-1 Reset Factor

Reset Category	Reset Factor
Hard reset	Power-on reset (PONR) RAM retention low-voltage detection reset (RVD) INITX (INITX) Illegal mode detection reset (IMR) Internal power supply low-voltage detection reset (LVDL1R) External power supply low-voltage detection reset (LVDH1R) Extended internal power supply low-voltage detection reset (LVDL2R) Extended external power supply low-voltage detection reset (LVDH2R) Clock stop wait timeout reset (CKTOR) RSTX pin input reset (RSTX) Hardware watchdog reset (HWDR) Software watchdog reset (SWDR) Main clock supervisor reset (CSVMOR) Sub clock supervisor reset (CSVSOR) PLL clock supervisor reset (CSVPRn) SSCG clock supervisor reset (CSVSRn) Fast CR clock supervisor reset (CSVFCRR) Slow CR clock supervisor reset (CSVSCRR) Profile error reset (PRFERR) Software trigger hard reset (SHRST) nSRST pin input reset (SRSTX)
Soft reset	Software reset (SRST)
Debugger reset	TRSTX pin input reset (TRSTX) Software debugger reset (SDBGRST)
Standby transition reset	PowerDomain reset 2 (PD2R) PowerDomain reset 3 (PD3Rn) PowerDomain reset 4 (PD4Rn) PowerDomain reset 5 (PD5Rn) PowerDomain reset 6 (PD6Rn)

Notes:

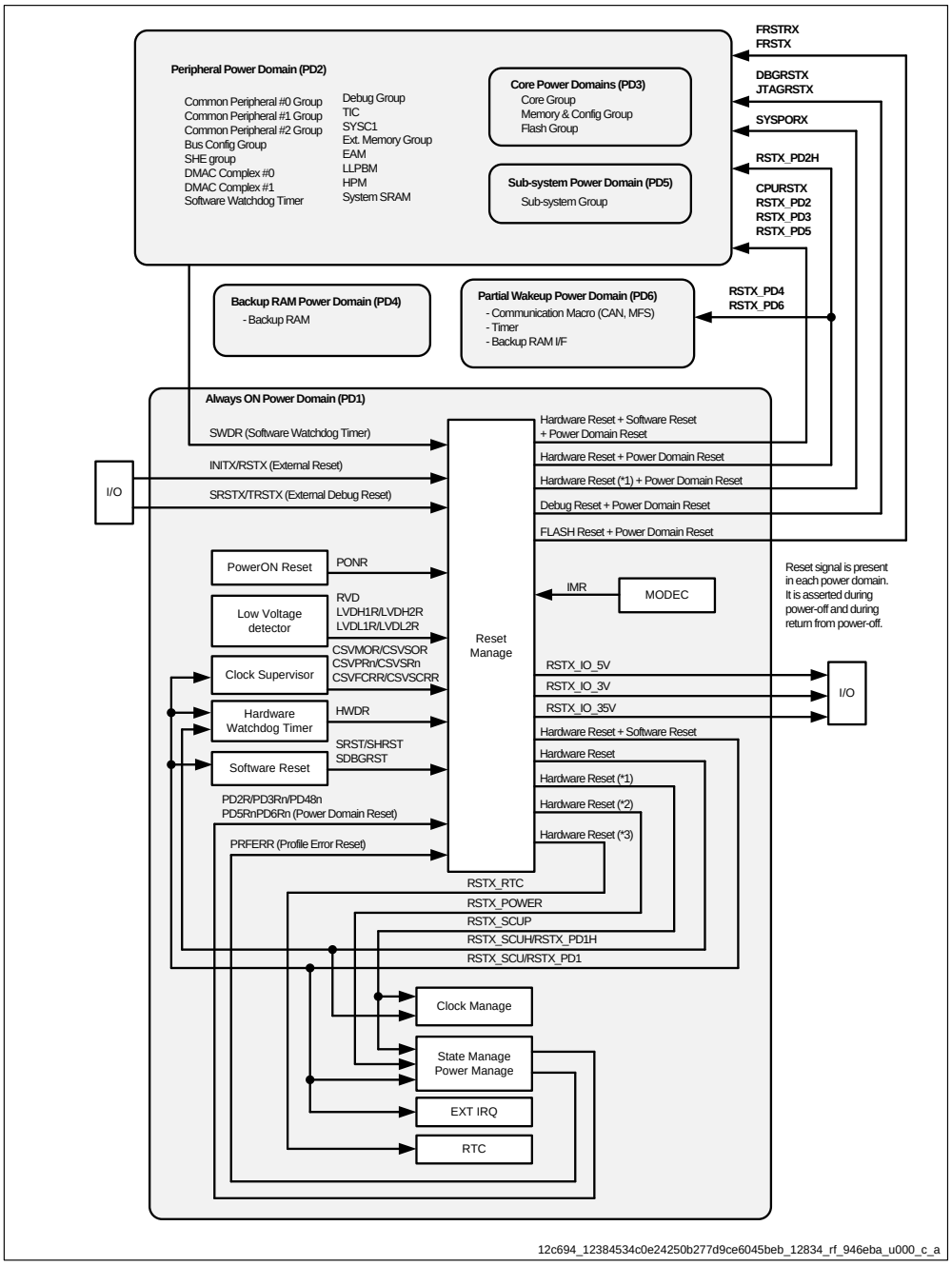
- *nSRST pin input reset cannot be used. Therefore, please ignore the description about nSRST in RESET chapter.*
- *Supported reset factors depend on product. For details, see product specification.*

2. Configuration

This section shows block diagrams of reset system.

Following block diagram show connections between the control logics and Power Domains.

Figure 2-1 Reset System Block Diagram



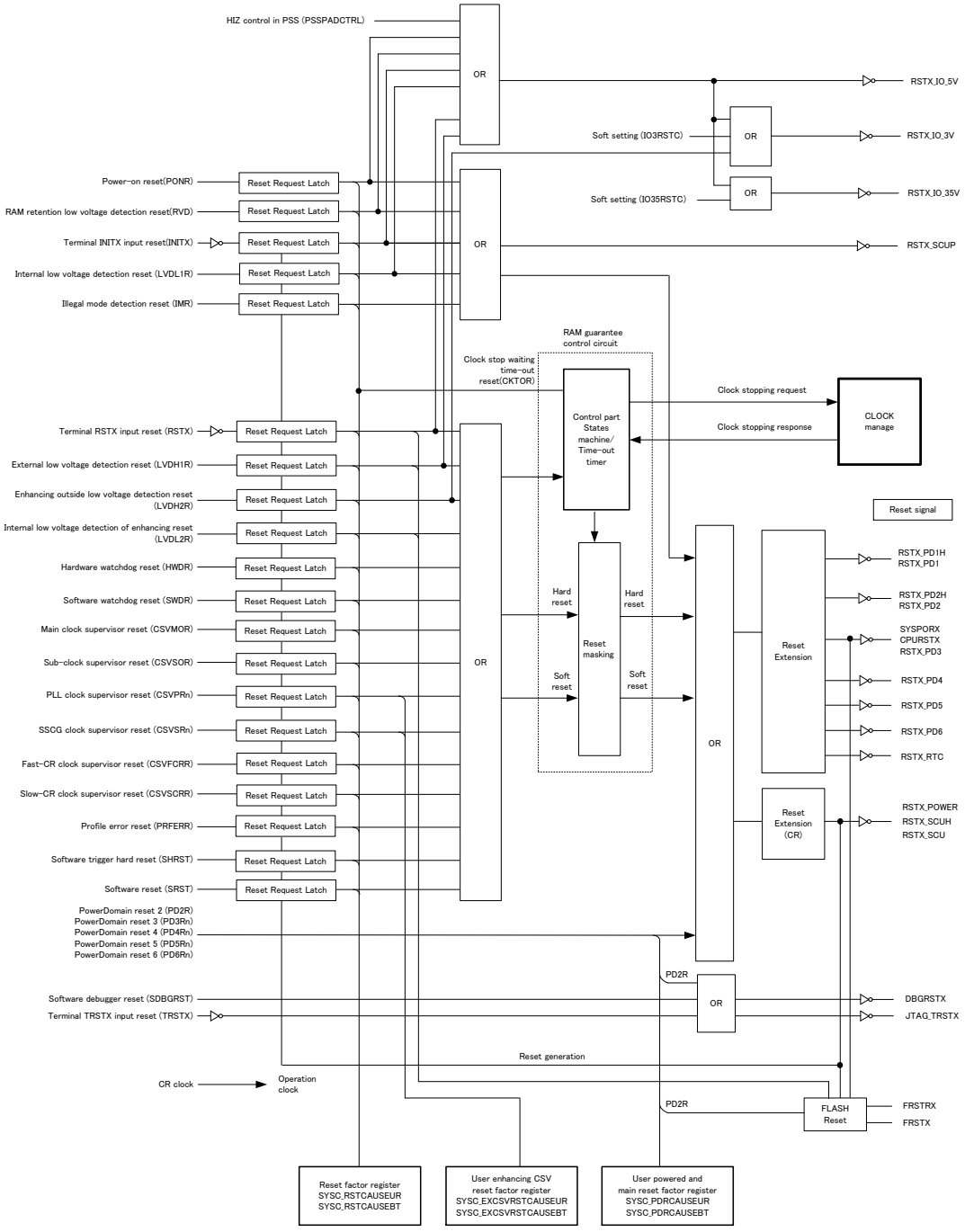
*1: It consists of PONR/INITX/RVD/LVDL1R.

*2: It consists of PONR/INITX/RVD/LVDL1R/HWDR.

*3: It consists of PONR/INITX/RVD/LVDL1R/HWDR/CSVMOR/CSVSOR/CSVSCRR

Following block diagram shows the control logic of the reset.

Figure 2-2 RST Manage Block Diagram



Notes:

- *As platform, RSTX_IO_3V is selectable the use or the nonuse. The use or the nonuse of RSTX_IO_3V is device-specific for each product. Please refer to Datasheet for any device-specific information.*

- *As platform, RSTX_IO_35V is selectable the use or the nonuse. The use or the nonuse of RSTX_IO_35V is device-specific for each product. Please refer to Datasheet for any device-specific information.*

3. Operational Description

This section explains operations of the reset.

3.1 Reset Factor

3.2 Internal Reset of Device

3.3 Reset Sequence

3.4 Operations after Reset Release

3.1. Reset Factor

This section explains details of each reset factors.

3.1.1 Power-on Reset (PONR)
3.1.2 RAM Retention Low-Voltage Detection Reset (RVD)
3.1.3 INITX (INITX)
3.1.4 Illegal Mode Detection Reset(IMR)
3.1.5 Internal Power Supply Low-Voltage Detection Reset (LVDL1R)
3.1.6 External Power Supply Low-Voltage Detection Reset (LVDH1R)
3.1.7 Extended Internal Power Supply Low-Voltage Reset (LVDL2R)
3.1.8 Extended External Power Supply Low-Voltage Reset (LVDH2R)
3.1.9 Clock Stop Wait Timeout Reset (CKTOR)
3.1.10 RSTX Pin Input Reset (RSTX)
3.1.11 Hardware Watchdog Reset (HWDR)
3.1.12 Software Watchdog Reset (SWDR)
3.1.13 Main Clock Supervisor Reset (CSVMOR)
3.1.14 Sub Clock Supervisor Reset (CSVSOR)
3.1.15 PLL Clock Supervisor Reset (CSVPRn)
3.1.16 SSCG Clock Supervisor Reset (CSVSRn)
3.1.17 Fast CR Clock Supervisor Reset (CSVFCRR)
3.1.18 Slow CR Clock Supervisor Reset (CSVSCRR)
3.1.19 Profile Error Reset (PRFERR)
3.1.20 Software Trigger Hard Reset (SHRST)
3.1.21 Software Reset (SRST)
3.1.22 TRSTX Pin Input Reset (TRSTX)
3.1.23 Software Debugger Reset (SDBGRST)
3.1.24 PowerDomain Reset 2 (PD2R)
3.1.25 PowerDomain Reset 3 (PD3Rn)
3.1.26 PowerDomain Reset 4 (PD4Rn)
3.1.27 PowerDomain Reset 5 (PD5Rn)
3.1.28 PowerDomain Reset 6 (PD6Rn)

3.1.1. Power-on Reset

It explains power-on reset.

Table 3-1 Power-on Reset

Occurrence factor	A power-on reset occurs when the internal PONR generator detects power-on of the 5 V system.
Release factor	The internal PONR generator releases the reset after the power-on reset release time elapses.
Initialization target	All registers
Corresponding flag	bit0 of User Reset Factor Register (SYSC_RSTCAUSEUR) bit0 of BootROM Reset Factor Register (SYSC_RSTCAUSEBT)
Remarks	Please refer to the "Data Sheet" for occurrence/release of Power-on Reset

Note:

- *Please refer to User Reset Factor Register (SYSC_RSTCAUSEUR) and BootROM Reset Factor Register (SYSC_RSTCAUSEBT) for the corresponding flag.*

3.1.2. RAM Retention Low-Voltage Detection Reset

It explains RAM Retention Low-voltage Detection reset.

Table 3-2 RAM Retention Low-Voltage Detection Reset

Occurrence factor	This occurs when the low-voltage detector (LVD) that is monitoring the 1.2 V power supply status enters the detection state.
Release factor	This is due to the release state of the low-voltage detector (LVD) that is monitoring the 1.2 V power supply status.
Initialization target	All registers except the following Reset factor registers
Corresponding flag	bit1 of User Reset Factor Register (SYSC_RSTCAUSEUR) bit1 of BootROM Reset Factor Register (SYSC_RSTCAUSEBT)
Remarks	Please refer to the "Data Sheet" for occurrence/release of RAM retention low-voltage detection reset

Note:

- *Please refer to User Reset Factor Register (SYSC_RSTCAUSEUR) and BootROM Reset Factor Register (SYSC_RSTCAUSEBT) for the corresponding flag.*

3.1.3. INITX

It explains INITX.

Table 3-3 INITX

Occurrence factor	Or,it is issued by inputting "L" level to simultaneous assert of RSTX pin and MD pin.
Release factor	Or,it is released by inputting other than "L" level to simultaneous assert of RSTX pin and MD pin.
Initialization target	All registers except the following: Reset factor registers
Corresponding flag	bit2 of User Reset Factor Register (SYSC_RSTCAUSEUR) bit2 of BootROM Reset Factor Register (SYSC_RSTCAUSEBT)
Remarks	For the noise filter removal width, see the Data Sheet.

Note:

- *Please refer to User Reset Factor Register (SYSC_RSTCAUSEUR) and BootROM Reset Factor Register (SYSC_RSTCAUSEBT) for the corresponding flag.*

3.1.4. Illegal Mode Detection Reset

It explains illegal mode detection reset.

Table 3-4 Illegal Mode Detection Reset

Occurrence factor	Illegal state is detected by comparing MD pin value and the device internal state. (MODEC_MODER.USERMODE="0" && MODEC_MODER.MD="1")
Release factor	Reset is released when the state is cleared.
Initialization target	All registers except the following: Reset factor registers
Corresponding flag	bit8 of User Reset Factor Register (SYSC_RSTCAUSEUR) bit8 of BootROM Reset Factor Register (SYSC_RSTCAUSEBT)
Remarks	-

Note:

- *Please refer to User Reset Factor Register (SYSC_RSTCAUSEUR) and BootROM Reset Factor Register (SYSC_RSTCAUSEBT) for the corresponding flag.*

3.1.5. Internal Power Supply Low-Voltage Detection Reset

It explains internal power supply low-voltage detection reset.

Table 3-5 Internal Power Supply Low-Voltage Detection Reset

Occurrence factor	This occurs when the low-voltage detector (LVD) which is monitoring the 1.2 V power supply status enters the detection state.
Release factor	This is due to the release state of the low-voltage detector (LVD) which is monitoring the 1.2 V power supply status.
Initialization target	All registers except the following: – Reset factor registers
Corresponding flag	bit30 of User Reset Factor Register (SYSC_RSTCAUSEUR) bit30 of BootROM Reset Factor Register (SYSC_RSTCAUSEBT)
Remarks	Please refer to the "Data Sheet" for occurrence/release of internal power supply low-voltage detection reset

Note:

– *Please refer to User Reset Factor Register (SYSC_RSTCAUSEUR) and BootROM Reset Factor Register (SYSC_RSTCAUSEBT) for the corresponding flag.*

3.1.6. External Power Supply Low-Voltage Detection Reset

It explains external power supply low-voltage detection reset.

Table 3-6 External Power Supply Low-Voltage Detection Reset

Occurrence factor	This occurs when the low-voltage detector (LVD) which is monitoring the 5.0 V power supply status enters the detection state.
Release factor	This is due to the release state of the low-voltage detector (LVD) which is monitoring the 5.0 V power supply status.
Initialization target	All registers except the following: – Reset factor registers – Clock supervisor output enable register – Debugging circuit – Real Time Clock
Corresponding flag	bit6 of User Reset Factor Register (SYSC_RSTCAUSEUR) bit6 of BootROM Reset Factor Register (SYSC_RSTCAUSEBT)
Remarks	Please refer to the "Data Sheet" for occurrence/release of external power supply low-voltage detection reset

Notes:

– *Please refer to User Reset Factor Register (SYSC_RSTCAUSEUR) and BootROM Reset Factor Register (SYSC_RSTCAUSEBT) for the corresponding flag.*

– *Initialization target registers are described outline only. Please refer to the register description of the appropriate chapter about the details.*

3.1.7. Extended Internal Power Supply Low-Voltage Reset

It explains extended internal power supply low-voltage reset.

Table 3-7 Extended Internal Power Supply Low-Voltage Reset

Occurrence factor	This occurs when the low-voltage detector (LVD) which is monitoring the 1.2 V power supply status enters the detection state.
Release factor	This is due to the release state of the low-voltage detector (LVD) which is monitoring the 1.2 V power supply status.
Initialization target	All registers except the following: – Reset factor registers – Clock supervisor output enable register – Debugging circuit – Real Time Clock
Corresponding flag	bit31 of User Reset Factor Register (SYSC_RSTCAUSEUR) bit31 of BootROM Reset Factor Register (SYSC_RSTCAUSEBT)
Remarks	Please refer to the "Data Sheet" for occurrence/release of extended internal power supply low-voltage detection reset

Notes:

- *Please refer to User Reset Factor Register (SYSC_RSTCAUSEUR) and BootROM Reset Factor Register (SYSC_RSTCAUSEBT) for the corresponding flag.*
- *Initialization target registers are described outline only. Please refer to the register description of the appropriate chapter about the details.*

3.1.8. Extended External Power Supply Low-Voltage Reset

It explains extended external power supply low-voltage reset.

Table 3-8 Extended External Power Supply Low-Voltage Reset

Occurrence factor	This occurs when the low-voltage detector (LVD) which is monitoring the external power supply status enters the detection state.
Release factor	This is due to the release state of the low-voltage detector (LVD) which is monitoring the external power supply status.
Initialization target	All registers except the following: – Reset factor registers – Clock supervisor output enable register – Debugging circuit – Real Time Clock
Corresponding flag	bit7 of User Reset Factor Register (SYSC_RSTCAUSEUR) bit7 of BootROM Reset Factor Register (SYSC_RSTCAUSEBT)
Remarks	Please refer to the "Data Sheet" for occurrence/release of extended external power supply low-voltage detection reset

Notes:

- *Please refer to User Reset Factor Register (SYSC_RSTCAUSEUR) and BootROM Reset Factor Register (SYSC_RSTCAUSEBT) for the corresponding flag.*
- *Initialization target registers are described outline only. Please refer to the register description of the appropriate chapter about the details.*

3.1.9. Clock Stop Wait Timeout Reset

It explains clock stop wait timeout reset.

Table 3-9 Clock Stop Wait Timeout Reset

Occurrence factor	This occurs when the clock stop wait timer expiration.
Release factor	This is automatically released after a reset is issued
Initialization target	Initialization target changes with each the reset factor.
Corresponding flag	bit3 of User Reset Factor Register (SYSC_RSTCAUSEUR) bit3 of BootROM Reset Factor Register (SYSC_RSTCAUSEBT)
Remarks	Timer expiration time: CR(4MHz) × 1024 cycle(typ = 256µs)

Notes:

- *Please refer to Operations after Reset Release for the functions of clock stop wait timeout reset.*
- *Please refer to User Reset Factor Register (SYSC_RSTCAUSEUR) and BootROM Reset Factor Register (SYSC_RSTCAUSEBT) for the corresponding flag.*

3.1.10. RSTX Pin Input Reset

It explains RSTX pin input reset.

Table 3-10 RSTX Pin Input Reset

Occurrence factor	This occurs because the "L" level is input to the RSTX pin
Release factor	This is released by input of the "H" level to the RSTX pin
Initialization target	All registers except the following: – Reset factor registers – Clock supervisor output enable register – Debugging circuit – Real Time Clock
Corresponding flag	bit4 of User Reset Factor Register (SYSC_RSTCAUSEUR) bit4 of BootROM Reset Factor Register (SYSC_RSTCAUSEBT)
Remarks	For the noise filter removal width, see the Data Sheet

Notes:

- *Please refer to User Reset Factor Register (SYSC_RSTCAUSEUR) and BootROM Reset Factor Register (SYSC_RSTCAUSEBT) for the corresponding flag.*
- *Initialization target registers are described outline only. Please refer to the register description of the appropriate chapter about the details.*

3.1.11. Hardware Watchdog Reset

It explains hardware watchdog reset.

Table 3-11 Hardware Watchdog Reset

Occurrence factor	A reset occurs when the CPU does not input a trigger within the window defined for the hardware watchdog.
Release factor	This is automatically released after the reset is issued.
Initialization target	All registers except the following: – Reset factor registers – Clock supervisor output enable register – Debugging circuit – Real Time Clock
Corresponding flag	bit11 of User Reset Factor Register (SYSC_RSTCAUSEUR) bit11 of BootROM Reset Factor Register (SYSC_RSTCAUSEBT)

Notes:

- *Please refer to User Reset Factor Register (SYSC_RSTCAUSEUR) and BootROM Reset Factor Register (SYSC_RSTCAUSEBT) for the corresponding flag.*
- *Initialization target registers are described outline only. Please refer to the register description of the appropriate chapter about the details.*

3.1.12. Software Watchdog Reset

It explains software watchdog reset.

Table 3-12 Software Watchdog Reset

Occurrence factor	A reset occurs when the CPU does not input a trigger within the specified window period for the software watchdog.
Release factor	After reset is issued, it automatically releases it.
Initialization target	All registers except the following: – Reset factor registers – Clock supervisor output enable register – Debugging circuit – Real Time Clock
Corresponding flag	bit12 of User Reset Factor Register (SYSC_RSTCAUSEUR) bit12 of BootROM Reset Factor Register (SYSC_RSTCAUSEBT)

Notes:

- *Please refer to User Reset Factor Register (SYSC_RSTCAUSEUR) and BootROM Reset Factor Register (SYSC_RSTCAUSEBT) for the corresponding flag.*
- *Initialization target registers are described outline only. Please refer to the register description of the appropriate chapter about the details.*

3.1.13. Main Clock Supervisor Reset

It explains main clock supervisor reset.

Table 3-13 Main Clock Supervisor Reset

Occurrence factor	This occurs when the CSV for monitoring the main clock detects a reset abnormality (reset occurrence condition).
Release factor	This is automatically released after a reset is issued.
Initialization target	All registers except the following: – Reset factor registers – Clock supervisor output enable register – Debugging circuit – Real Time Clock
Corresponding flag	bit24 of User Reset Factor Register (SYSC_RSTCAUSEUR) bit24 of BootROM Reset Factor Register (SYSC_RSTCAUSEBT)

Notes:

- *Please refer to "Chapter: Clock Supervisor" about monitoring main clock.*
- *Please refer to User Reset Factor Register (SYSC_RSTCAUSEUR) and BootROM Reset Factor Register (SYSC_RSTCAUSEBT) for the corresponding flag.*
- *Real Time Clock listed in the above table will be initialized if the operation clock of Real Time Clock is set to main clock.*
- *Initialization target registers are described outline only. Please refer to the register description of the appropriate chapter about the details.*

3.1.14. Sub Clock Supervisor Reset

It explains sub clock supervisor reset.

Table 3-14 Sub Clock Supervisor Reset

Occurrence factor	This occurs when the CSV for monitoring the sub clock detects a reset abnormality (reset occurrence condition).
Release factor	This is automatically released after a reset is issued.
Initialization target	All registers except the following: – Reset factor registers – Clock supervisor output enable register – Debugging circuit – Real Time Clock
Corresponding flag	bit25 of User Reset Factor Register (SYSC_RSTCAUSEUR) bit25 of BootROM Reset Factor Register (SYSC_RSTCAUSEBT)

Notes:

- *Please refer to "Chapter: Clock Supervisor" about monitoring sub clock.*
- *Please refer to User Reset Factor Register (SYSC_RSTCAUSEUR) and BootROM Reset Factor Register (SYSC_RSTCAUSEBT) for the corresponding flag.*
- *Real Time Clock listed in the above table will be initialized if the operation clock of Real Time Clock is set to sub clock.*
- *Initialization target registers are described outline only. Please refer to the register description of the appropriate chapter about the details.*

3.1.15. PLL Clock Supervisor Reset

It explains PLL clock supervisor reset.

Table 3-15 PLL Clock Supervisor Reset

Occurrence factor	This occurs when the CSV for monitoring the PLL clock detects a reset abnormality (reset occurrence condition).
Release factor	This is automatically released after a reset is issued.
Initialization target	All registers except the following: – Reset factor registers – Clock supervisor output enable register – Debugging circuit – Real Time Clock
Corresponding flag	For PLL0: bit26 of User Reset Factor Register (SYSC_RSTCAUSEUR) bit26 of BootROM Reset Factor Register (SYSC_RSTCAUSEBT) For PLL1 to 3: bit3 to 1 of User Extended CSV Reset Factor Register (SYSC_EXCSVRSTCAUSEUR) bit3 to 1 of BootROM Extended CSV Reset Factor Register (SYSC_EXCSVRSTCAUSEBT)

Notes:

- *Please refer to "Chapter: Clock Supervisor" about monitoring PLL clock.*
- *Please refer to User Reset Factor Register (SYSC_RSTCAUSEUR) and BootROM Reset Factor Register (SYSC_RSTCAUSEBT) for the corresponding flag for PLL0.*
- *Please refer to User Extended CSV Reset Factor Register (SYSC_EXCSVRSTCAUSEUR) and BootROM Extended CSV Reset Factor Register (SYSC_EXCSVRSTCAUSEBT) for the corresponding flag for PLL1 to 3.*
- *Initialization target registers are described outline only. Please refer to the register description of the appropriate chapter about the details.*

3.1.16. SSCG Clock Supervisor Reset

It explains SSCG Clock Supervisor reset.

Table 3-16 SSCG Clock Supervisor Reset

Occurrence factor	This occurs when the CSV for monitoring the SSCG clock detects a reset abnormality (reset occurrence condition).
Release factor	This is automatically released after a reset is issued.
Initialization target	All registers except the following: – Reset factor registers – Clock supervisor output enable register – Debugging circuit – Real Time Clock
Corresponding flag	For SSCG0: bit27 of User Reset Factor Register (SYSC_RSTCAUSEUR) bit27 of BootROM Reset Factor Register (SYSC_RSTCAUSEBT) For SSCG1 to 3: bit7 to 5 of User Extended CSV Reset Factor Register (SYSC_EXCSVRSTCAUSEUR) bit7 to 5 of BootROM Extended CSV Reset Factor Register (SYSC_EXCSVRSTCAUSEBT)

Notes:

- *Please refer to "Chapter: Clock Supervisor" about monitoring PLL clock.*
- *Please refer to User Reset Factor Register (SYSC_RSTCAUSEUR) and BootROM Reset Factor Register (SYSC_RSTCAUSEBT) for the corresponding flag for SSCG0.*
- *Please refer to User Extended CSV Reset Factor Register (SYSC_EXCSVRSTCAUSEUR) and BootROM Extended CSV Reset Factor Register (SYSC_EXCSVRSTCAUSEBT) for the corresponding flag for SSCG1 to 3.*
- *Initialization target registers are described outline only. Please refer to the register description of the appropriate chapter about the details.*

3.1.17. Fast CR Clock Supervisor Reset

It explains Fast CR Clock Supervisor reset.

Table 3-17 Fast CR Clock Supervisor Reset

Occurrence factor	This occurs when the CSV for monitoring the Fast CR clock detects a reset abnormality (reset occurrence condition).
Release factor	This is automatically released after a reset is issued.
Initialization target	All registers except the following: – Reset factor registers – Clock supervisor output enable register – Debugging circuit – Real Time Clock
Corresponding flag	bit28 of User Reset Factor Register (SYSC_RSTCAUSEUR) bit28 of BootROM Reset Factor Register (SYSC_RSTCAUSEBT)

Notes:

- *Please refer to "Chapter: Clock Supervisor" about monitoring Fast CR clock.*
- *Please refer to User Reset Factor Register (SYSC_RSTCAUSEUR) and BootROM Reset Factor Register (SYSC_RSTCAUSEBT) for the corresponding flag.*
- *Initialization target registers are described outline only. Please refer to the register description of the appropriate chapter about the details.*

3.1.18. Slow CR Clock Supervisor Reset

It explains Slow CR Clock Supervisor reset.

Table 3-18 Slow CR Clock Supervisor Reset

Occurrence factor	This occurs when the CSV for monitoring the Slow CR clock detects a reset abnormality (reset occurrence condition).
Release factor	This is automatically released after a reset is issued.
Initialization target	All registers except the following: – Reset factor registers – Clock supervisor output enable register – Debugging circuit – Real Time Clock
Corresponding flag	bit29 of User Reset Factor Register (SYSC_RSTCAUSEUR) bit29 of BootROM Reset Factor Register (SYSC_RSTCAUSEBT)

Notes:

- *Please refer to "Chapter: Clock Supervisor" about monitoring Slow CR clock.*
- *Please refer to User Reset Factor Register (SYSC_RSTCAUSEUR) and BootROM Reset Factor Register (SYSC_RSTCAUSEBT) for the corresponding flag.*
- *Initialization target registers are described outline only. Please refer to the register description of the appropriate chapter about the details.*
- *Real Time Clock listed in the above table will be initialized if the operation clock of Real Time Clock is set to Slow CR clock.*

3.1.19. Profile Error Reset

It explains profile error reset.

Table 3-19 Profile Error Reset

Occurrence factor	The reset occurs when an error is detected in the RUN profile in a transition from PSS to RUN
Release factor	This is automatically released after a reset is issued.
Initialization target	All registers except the following: – Reset factor registers – Clock supervisor output enable register – Debugging circuit – Real Time Clock
Corresponding flag	bit10 of User Reset Factor Register (SYSC_RSTCAUSEUR) bit10 of BootROM Reset Factor Register (SYSC_RSTCAUSEBT)

Notes:

- *Please refer to User Reset Factor Register (SYSC_RSTCAUSEUR) and BootROM Reset Factor Register (SYSC_RSTCAUSEBT) for the corresponding flag.*
- *Please refer to "Chapter: Low-power Consumption" for profile error.*
- *Initialization target registers are described outline only. Please refer to the register description of the appropriate chapter about the details.*

3.1.20. Software Trigger Hard Reset

It explains software trigger hard reset.

Table 3-20 Software Trigger Hard Reset

Occurrence factor	SYSC_RSTCNTR.SWHRST The reset occurs because 0xA5 is written to the control register.
Release factor	This is automatically released after a reset is issued.
Initialization target	All registers except the following: – Reset factor registers – Clock supervisor output enable register – Debugging circuit – Real Time Clock
Corresponding flag	bit20 of User Reset Factor Register (SYSC_RSTCAUSEUR) bit20 of BootROM Reset Factor Register (SYSC_RSTCAUSEBT)

Notes:

- *Please refer to User Reset Factor Register (SYSC_RSTCAUSEUR) and BootROM Reset Factor Register (SYSC_RSTCAUSEBT) for the corresponding flag.*
- *Initialization target registers are described outline only. Please refer to the register description of the appropriate chapter about the details.*

3.1.21. Software Reset

It explains software reset.

Table 3-21 Software Reset

Occurrence factor	SYSC_RSTCNTR.SWRST The reset occurs because 0x5A is written to the register.
Release factor	This is automatically released after a reset is issued.
Initialization target	All registers except the following: – Reset factor registers – Clock supervisor output enable register – Debugging circuit – Software watchdog setting register – Hardware watchdog setting register – System controller clock setting register – Register for the clock supervisor in the system controller – Clock system setting register – Clock supervisor setting register – Source clock timer setting register – Real Time Clock – Hardware watchdog trigger sequence monitoring logic – Software watchdog trigger sequence monitoring logic
Corresponding flag	bit16 of User Reset Factor Register (SYSC_RSTCAUSEUR) bit16 of BootROM Reset Factor Register (SYSC_RSTCAUSEBT)

Notes:

- *Please refer to User Reset Factor Register (SYSC_RSTCAUSEUR) and BootROM Reset Factor Register (SYSC_RSTCAUSEBT) for the corresponding flag.*
- *In case of generating the software reset, please write to SYSC_RSTCNTR after clearing the hardware watchdog counter in advance. After writing to SYSC_RSTCNTR, please do not clear the watchdog counter until the reset is released.*
- *Initialization target registers are described outline only. Please refer to the register description of the appropriate chapter about the details.*

3.1.22. TRSTX Pin Input Reset

It explains TRSTX Pin Input

Table 3-22 TRSTX Pin Input

Occurrence factor	This occurs because the "L" level is input to the TRSTX pin.
Release factor	This is released by input of the "H" level to the TRSTX pin.
Initialization target	Debugging circuit
Corresponding flag	None

Note:

- *Initialization target registers are described outline only. Please refer to the register description of the appropriate chapter about the details.*

3.1.23. Software Debugger Reset

It explains software debugger reset.

Table 3-23 Software Debugger Reset

Occurrence factor	SYSC_RSTCNTR.DBGR The reset occurs because 0xDA is written to the control register.
Release factor	After reset is issued, it automatically releases it.
Initialization target	Debugging circuit
Corresponding flag	None

Note:

- *Initialization target registers are described outline only. Please refer to the register description of the appropriate chapter about the details.*

3.1.24. PowerDomain Reset 2

It explains PowerDomain reset 2.

Table 3-24 PowerDomain Reset 2

Occurrence factor	This occurs because of a power shutdown.
Release factor	This is released by recovery from the power shutdown.
Initialization target	PowerDomain 2
Corresponding flag	bit0 of User PowerDomain Reset Factor Register (SYSC_PDRSTCAUSEUR) bit0 of BootROM PowerDomain Reset Factor Register (SYSC_PDRSTCAUSEBT)

Notes:

- *Please refer to "Chapter: Low-power Consumption" for the definition of PowerDomain.*
- *Please refer to User PowerDomain Reset Factor Register (SYSC_PDRSTCAUSEUR) and BootROM PowerDomain Reset Factor Register (SYSC_PDRSTCAUSEBT) for the corresponding flag.*
- *Initialization target registers are described outline only. Please refer to the register description of the appropriate chapter about the details.*

3.1.25. PowerDomain Reset 3

It explains PowerDomain reset 3.

Table 3-25 PowerDomain Reset 3

Occurrence factor	This occurs because of a power shutdown.
Release factor	This is released by recovery from the power shutdown.
Initialization target	PowerDomain 3
Corresponding flag	bit4 of User PowerDomain Reset Factor Register (SYSC_PDRSTCAUSEUR) bit4 of BootROM PowerDomain Reset Factor Register (SYSC_PDRSTCAUSEBT)

Notes:

- *Please refer to "Chapter: Low-power Consumption" for the definition of PowerDomain.*
- *Please refer to User PowerDomain Reset Factor Register (SYSC_PDRSTCAUSEUR) and BootROM PowerDomain Reset Factor Register (SYSC_PDRSTCAUSEBT) for the corresponding flag.*
- *Initialization target registers are described outline only. Please refer to the register description of the appropriate chapter about the details.*

3.1.26. PowerDomain Reset 4

It explains PowerDomain reset 4.

Table 3-26 PowerDomain Reset 4

Occurrence factor	This occurs because of a power shutdown.
Release factor	This is released by recovery from the power shutdown.
Initialization target	PowerDomain 4
Corresponding flag	bit9-8 of User PowerDomain Reset Factor Register (SYSC_PDRSTCAUSEUR) bit9-8 of BootROM PowerDomain Reset Factor Register (SYSC_PDRSTCAUSEBT)

Notes:

- *Please refer to "Chapter: Low-power Consumption" for the definition of PowerDomain.*
- *Please refer to User PowerDomain Reset Factor Register (SYSC_PDRSTCAUSEUR) and BootROM PowerDomain Reset Factor Register (SYSC_PDRSTCAUSEBT) for the corresponding flag.*
- *Initialization target registers are described outline only. Please refer to the register description of the appropriate chapter about the details.*

3.1.27. PowerDomain Reset 5

It explains PowerDomain reset 5.

Table 3-27 PowerDomain Reset 5

Occurrence factor	This occurs because of a power shutdown.
Release factor	This is released by recovery from the power shutdown.
Initialization target	PowerDomain 5
Corresponding flag	bit15-12 of User PowerDomain Reset Factor Register (SYSC_PDRSTCAUSEUR) bit15-12 of BootROM PowerDomain Reset Factor Register (SYSC_PDRSTCAUSEBT)

Notes:

- *Please refer to "Chapter: Low-power Consumption" for the definition of PowerDomain.*
- *Please refer to User PowerDomain Reset Factor Register (SYSC_PDRSTCAUSEUR) and BootROM PowerDomain Reset Factor Register (SYSC_PDRSTCAUSEBT) for the corresponding flag.*
- *Initialization target registers are described outline only. Please refer to the register description of the appropriate chapter about the details.*

3.1.28. PowerDomain Reset 6

It explains PowerDomain reset 6.

Table 3-28 PowerDomain Reset 6

Occurrence factor	This occurs because of a power shutdown.
Release factor	This is released by recovery from the power shutdown.
Initialization target	PowerDomain 6
Corresponding flag	bit17-16 of User PowerDomain Reset Factor Register (SYSC_PDRSTCAUSEUR) bit17-16 of BootROM PowerDomain Reset Factor Register (SYSC_PDRSTCAUSEBT)

Notes:

- *Please refer to "Chapter: Low-power Consumption" for the definition of PowerDomain.*
- *Please refer to User PowerDomain Reset Factor Register (SYSC_PDRSTCAUSEUR) and BootROM PowerDomain Reset Factor Register (SYSC_PDRSTCAUSEBT) for the corresponding flag.*
- *Initialization target registers are described outline only. Please refer to the register description of the appropriate chapter about the details.*

3.2. Internal Reset of Device

This section explains the internal reset signals of this device.

The reset connected to this device are reset input to CPU, input to peripheral circuits and input to external 1.2V.

- CPU reset
- I/O reset
- External 1.2V power supply reset circuit

3.2.1. CPU Reset

This section explains the CPU reset. There are 4 types of CPU reset signals. The following shows the corresponding reset factor and initialize target.

Table 3-29 CPU Reset

Signal Name	Initialization Target	Reset Category	Reset Factor
SYSPORX	Core Group Debug Group	Hard reset	Power-on reset RAM retention low-voltage detection reset Internal power supply low-voltage detection reset Extended internal power supply low-voltage detection reset INITX Illegal mode detection reset
		Standby transition reset	PowerDomain reset 2
CPURSTX	Core Group Debug Group	Hard reset	All reset factors
		Soft reset	All reset factors
		Standby transition reset	PowerDomain reset 2
DBGRSTX	Debug Group	Hard reset	Power-on reset RAM retention low-voltage detection reset Internal power supply low-voltage detection reset Extended internal power supply low-voltage detection reset INITX Illegal mode detection reset
		Debugger reset	Software debugger reset
		Standby transition reset	PowerDomain reset 2
JTAGRSTX	Debug Group	Hard reset	Power-on reset RAM retention low-voltage detection reset Internal power supply low-voltage detection reset Extended internal power supply low-voltage detection reset INITX Illegal mode detection reset
		Debugger reset	TRSTX pin input reset
		Standby transition reset	PowerDomain reset 2

3.2.2. I/O Reset

This section explains the I/O reset. The types of reset signals for I/O are shown below. The following table shows the corresponding reset factor and initialize target.

Table 3-30 I/O Reset

Signal Name	Initialization Target	Reset Category	Reset Factor
RSTX_IO_5V	I/O cell driven by 5V power supply	Hard reset	Power-on reset RAM retention low-voltage detection reset Internal power supply low-voltage detection reset External power supply low-voltage detection reset RSTX pin input reset INITX I/O Hi-z control at PSS *3
RSTX_IO_3V	I/O cell driven by 3V power supply	Hard reset	Power-on reset RAM retention low-voltage detection reset Internal power supply low-voltage detection reset External power supply low-voltage detection reset RSTX pin input reset INITX Extended external power supply low-voltage detection reset Soft setting *1 I/O Hi-z control at PSS *3
RSTX_IO_35V	I/O cell driven by 3V/5V power supply	Hard reset	Power-on reset RAM retention low-voltage detection reset Internal power supply low-voltage detection reset External power supply low-voltage detection reset RSTX pin input reset INITX Soft setting *2 I/O Hi-z control at PSS *3

The I/O cell connected to the external pin of the device is initialized when the I/O reset is issued. During the initialization, the I/O cell is either outputting "L" or in the high impedance state. Refer to the "Pin State Table" for pin state during the initialization. The I/O cell is asynchronously initialized when a reset factor is received. It stays in this state until an internal reset is issued and the reset factor is released.

Notes:

- **1: It is controlled by IO3RSTC in system special setting register (SYSC0_SPECFGR). RSTX_IO_3V becomes active by IO3RSTC, when IO3RSTC is initialized by hard reset or soft reset.*
- **2: It is controlled by IO35RSTC in system special setting register (SYSC0_SPECFGR). RSTX_IO_35V becomes active by IO35RSTC, when IO35RSTC is initialized by hard reset or soft reset.*
- **3: It is activated by the transition to the PSS in case of PSSPADCTRL="1" in system special setting register (SYSC0_SPECFGR).*

3.2.3. Reset of External Power Supply Control

This section explains the reset of the external power supply control.

Table 3-31 Reset of External Power Supply Control

Signal Name	Initialization Target	Reset Category	Reset Factor
EX12VRST	1.2V external power supply control	Hard reset	Power-on reset RAM retention low-voltage detection reset Internal power supply low-voltage detection reset External power supply low-voltage detection reset *1 Extended internal power supply low-voltage detection reset INITX Illegal mode detection reset Hardware watchdog reset
		Standby transition reset	PowerDomain reset 2
EX5VRST	5/3.3V external power supply control	Hard reset	Power-on reset RAM retention low-voltage detection reset Internal power supply low-voltage detection reset External power supply low-voltage detection reset *1 Extended internal power supply low-voltage detection reset INITX Illegal mode detection reset Hardware watchdog reset
		Standby transition reset	PowerDomain reset 2

Note:

- **1: It is activated by EXVRSTCNT=0 in system special setting register (SYSC0_SPECFGR).*

There are some cases 1.2V external power supply control to be high.

Table 3-32 Behavior of External Power Supply Control

Condition	Reset trigger0	Reset trigger1	Reset trigger2	1.2V external power supply control
In power on sequence, Reset trigger1 and Reset trigger2 release during Fast-CR stabilization time	1	x	x	Case0
In power on sequence, Reset trigger1, Reset trigger2 release after Fast-CR stabilization time	0	1	x	Case1
	0	0	1	Case2
	0	1	1	Case3
In PSS mode with Fast-CR oscillating, Reset trigger0, Reset trigger1, Reset trigger2 occur	1	x	x	Case0
	0	1	1	Case4
In PSS mode with Fast-CR stopping, Reset trigger0, Reset trigger1, Reset trigger2 occur	1	x	x	Case0
	0	1	x	Case5
	0	0	1	Case6

- Reset trigger0:

Power-on reset, RAM retention low-voltage detection reset, Internal power supply low-voltage detection reset,
Extended internal power supply low-voltage detection reset, INITX, Illegal mode detection reset, Hardware watchdog reset

- Reset trigger1:

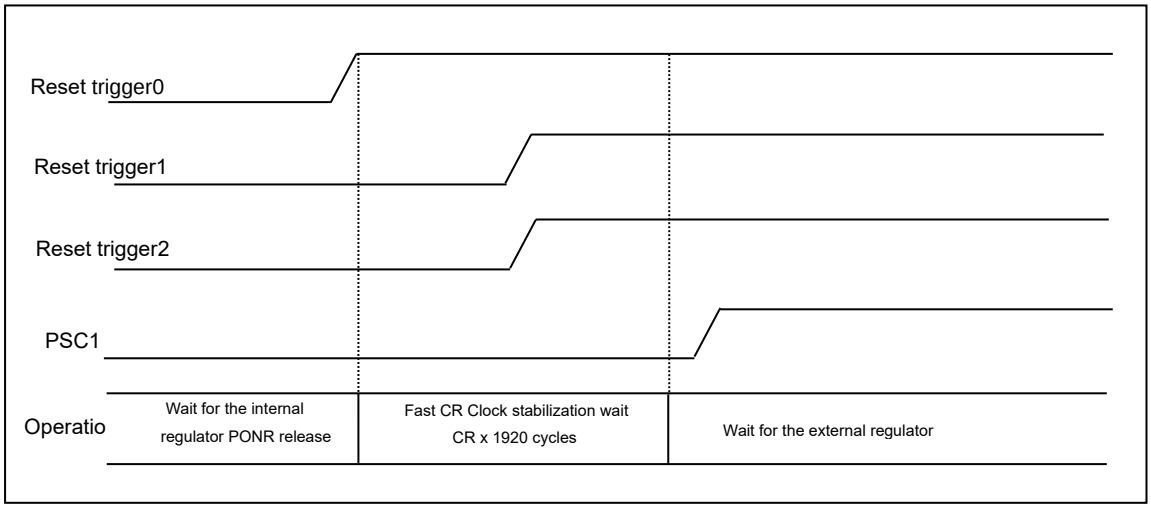
  External power supply low-voltage detection reset

- Reset trigger2:

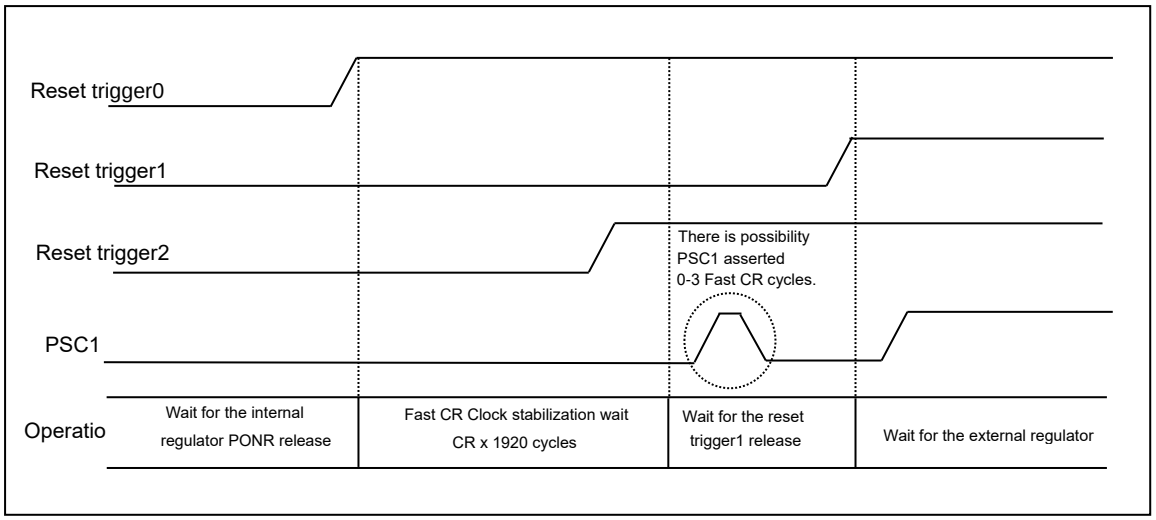
  RSTX

1: reset active
0: reset inactive
x: reset inactive or reset inactive before completion of Fast-CR stabilization wait

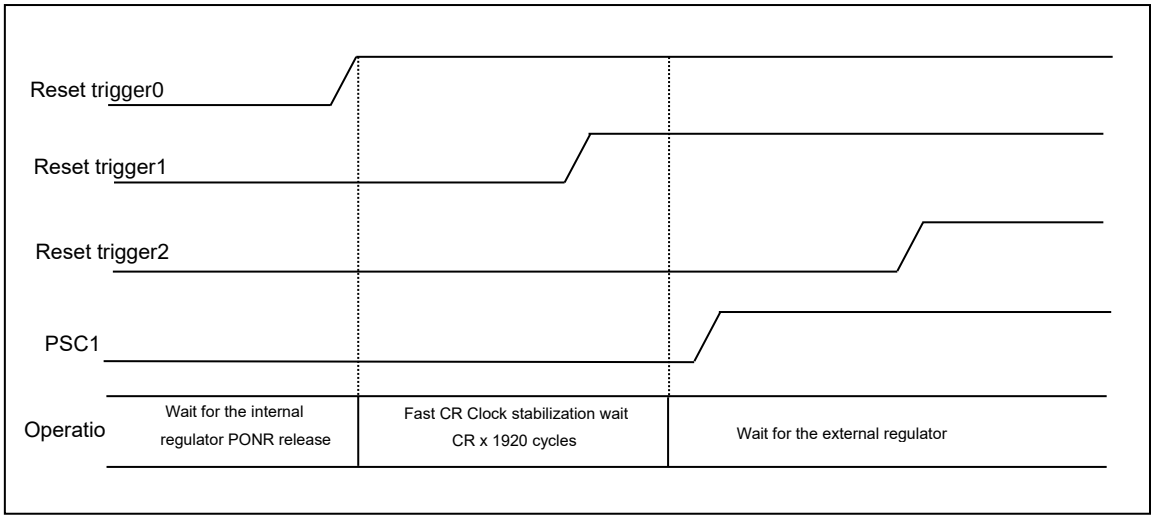
Case0) In power on sequence/PSS, Reset trigger1 and 2 release during Fast-CR stabilization time



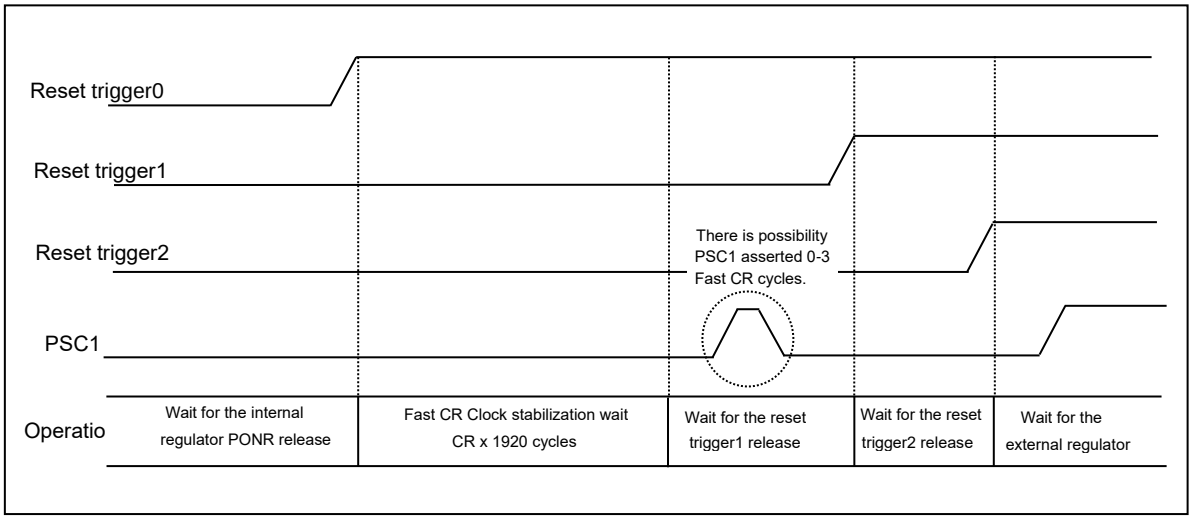
Case1) In power on sequence, Reset trigger1 release after Fast-CR stabilization time



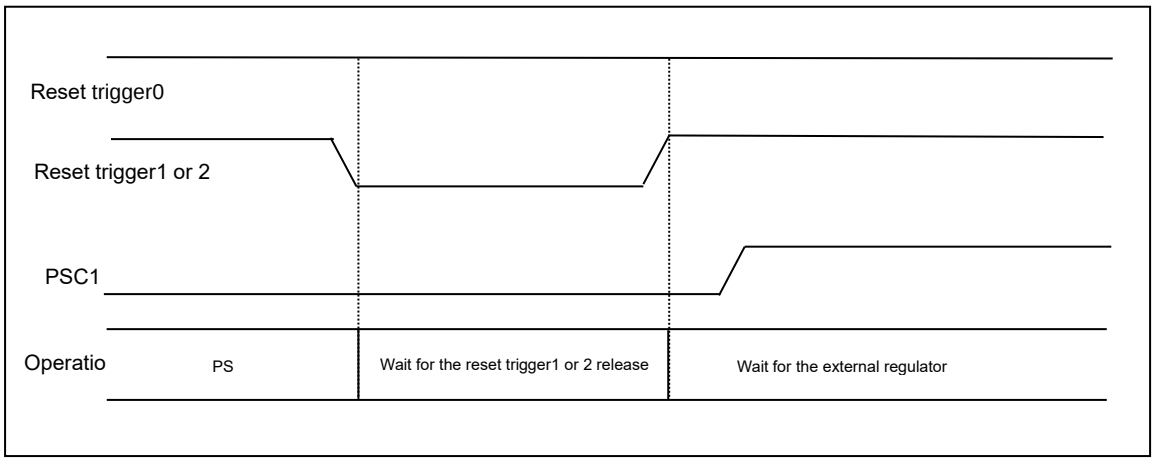
Case2) In power on sequence, Reset trigger2 release after Fast-CR stabilization time



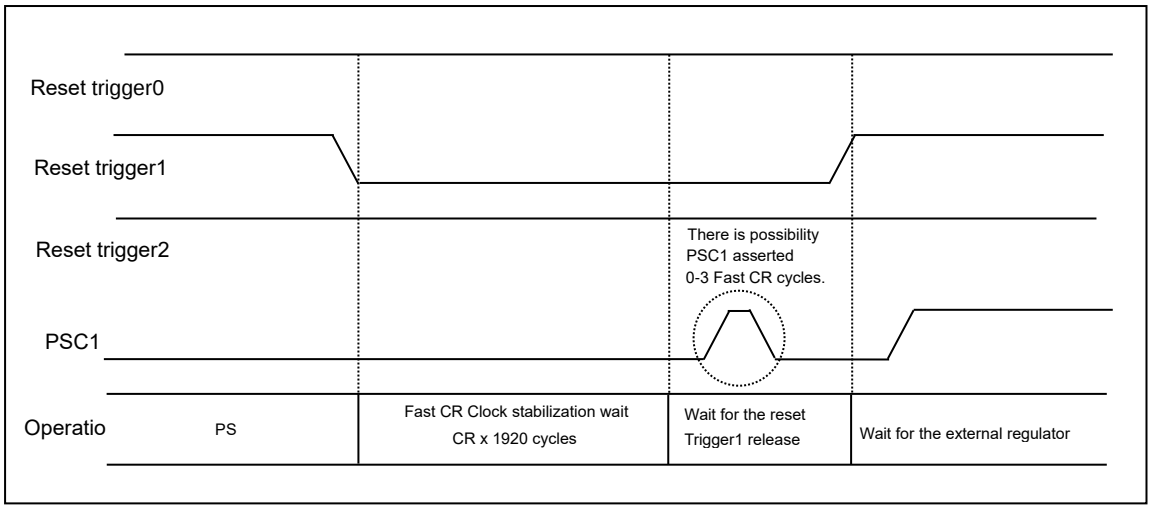
Case3) In power on sequence, Reset trigger1 and Reset trigger2 release after Fast-CR stabilization time



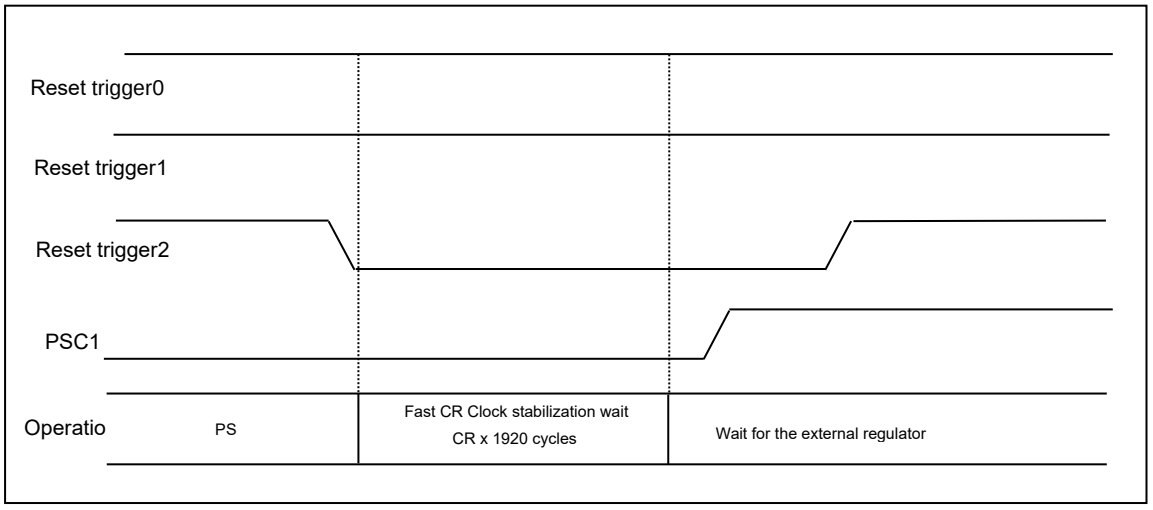
Case4) In PSS mode with Fast-CR oscillating, Reset trigger1 or Reset trigger2 occur



Case5) In PSS mode with Fast-CR stopped, Reset trigger1 occur and release after Fast-CR stabilization time



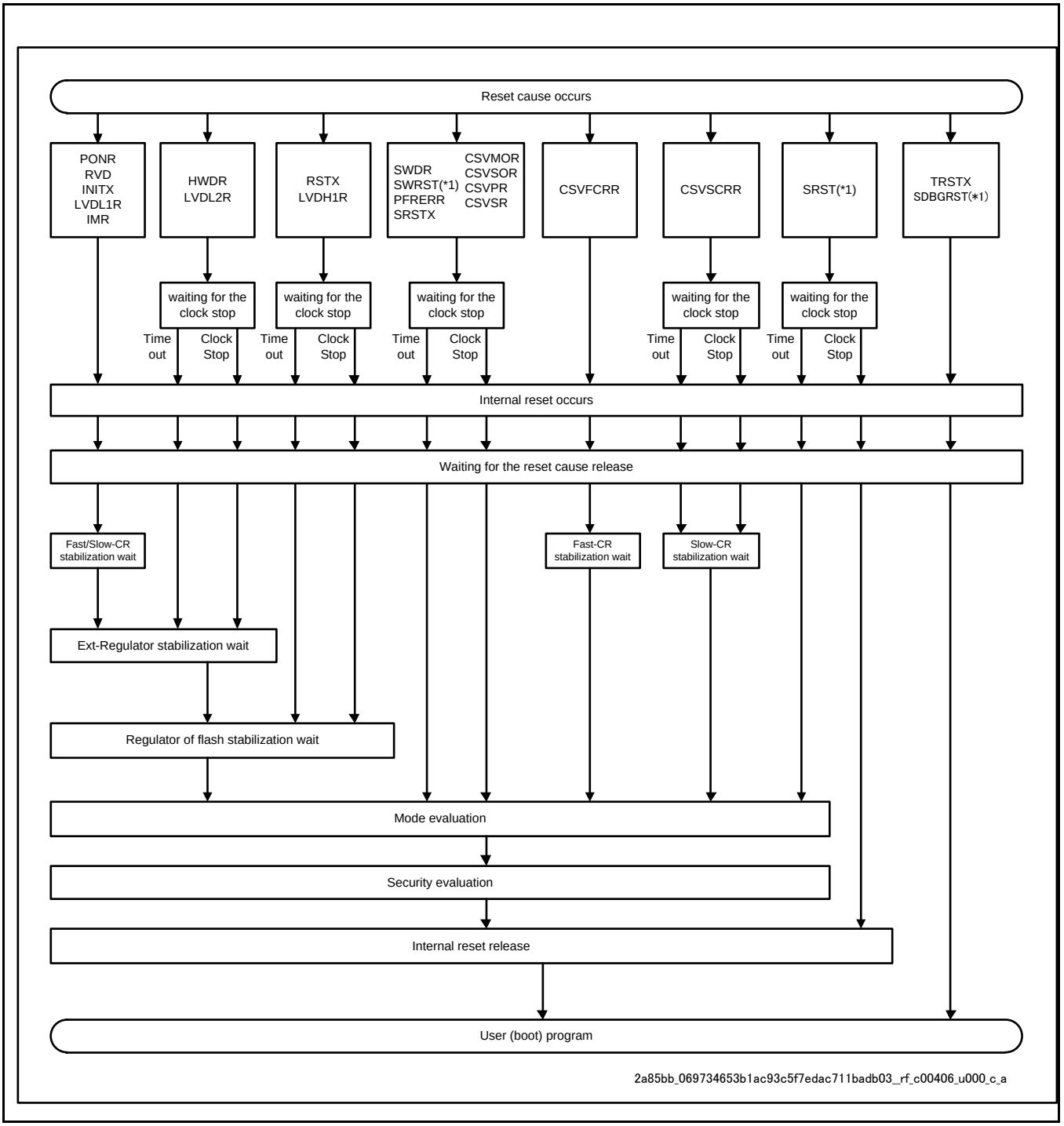
Case6) In PSS mode with Fast-CR stopped, Reset trigger2 occur and release after Fast-CR stabilization time



3.3. Reset Sequence

This section explains the reset sequence of this device. Program and hardware operation will start from initial state by releasing the reset. The sequence from reset to operation start is called reset sequence.

Figure 3-1 Reset Sequence



Note:

- *At Power Saving State (It is *1 in figure), the lying reset factor (hard software trigger reset, software reset, and software debugger reset) doesn't generate CPU.*

Reset Sequence

1. Reset cause occurred
   Reset cause is captured and will be held until a reset is issued inside the device.
2. Waiting for the clock stop
   Before a reset, clock source is stopped for the resets which need RAM(TCRAM, Backup RAM, System RAM) to be guaranteed. Timeover will occur if the clock does not stop in the specified time.
3. Internal reset occurs
   Reset will be issued to internal when reset preparation is done.
4. Waiting for the reset factor release
   Wait until the reset cause of the issued reset to be released. Necessary process will be done internally for reset release after the reset cause is released.
5. Fast/Slow-CR stabilization wait
   Wait for Fast/Slow-CR clock stabilization.
6. Ext-Regulator stabilization wait
   Wait for external power supply (1.2V) to stabilize.
7. Regulator of flash stabilization wait
   Wait for Flash macro's power supply to stabilize.
8. Mode evaluation
   Operation mode is set and announced to each hardware.
9. Security evaluation
   Security is judged and announced to each hardware.
10. Internal reset release
    Release of internal reset is done.
11. Reset vector fetch (Running the Boot ROM program)
    CPU starts to fetch the reset vectors. CPU executes the program Boot ROM.
12. Running the User program
    After Boot ROM program execution, CPU will jump to the user program. CPU starts user program operation.

3.4. Operations after Reset Release

The operation after the reset factor and reset are released is shown below.

Table 3-33 Operations after Reset Release

Reset Category	Reset Factor	Operation after Reset is Released				
		Waiting for Stabilization	Mode Evaluation	Security Evaluation	Operation Clock	RAM Guarantee
Hardware Reset	Power-on reset	Fast CR Slow CR FLASH 1.2V external power supply control Regulator	Yes	Yes	Fast CR	No
	RAM retention low-voltage detection reset Internal power supply low-voltage detection reset INITX Illegal mode detection reset	Fast CR Slow CR FLASH 1.2V external power supply control	Yes	Yes	Fast CR	No
	Fast CR clock supervisor reset	Fast CR	Yes	Yes	Fast CR	No
	Slow CR clock supervisor reset	Slow CR	Yes	Yes	Fast CR	No
	Clock stop wait timeout reset Main clock supervisor reset Sub-clock supervisor reset PLL clock supervisor reset SSCG clock supervisor reset	N/A	Yes	Yes	Fast CR	No
	Hardware watchdog reset Extended internal power supply low-voltage detection reset	FLASH 1.2V external power supply control	Yes	Yes	Fast CR	Yes*1
	RSTX pin input reset External power supply low-voltage detection reset	FLASH	Yes	Yes	Fast CR	Yes*3
	Software watchdog reset External power supply low-voltage detection reset Extended external Power supply low-voltage detection reset Profile error reset Software trigger hard reset	N/A	Yes	Yes	Fast CR	Yes*3
	nSRST pin input reset	N/A	No	No	No change	Yes*3
Software Reset	Software reset	N/A	No	No	No change	Yes*3
Debugger reset	TRSTX pin input reset Software debugger reset	N/A	No	No	No change	Yes*2

Several hard reset factors guarantee the RAM data after the reset but the others do not. The reset factors which do not guarantee the RAM data issue the reset regardless RAM access, so the RAM data are not guaranteed after the reset. The reset factors which guarantee the RAM data issue the reset after the clocks have stopped.

However, in a case that a reset factor which can guarantee the RAM data has not completed within a certain time period, another reset is issued (Clock stop wait timeout reset). In this case, the RAM data after recovery from the reset is not guaranteed.

Notes:

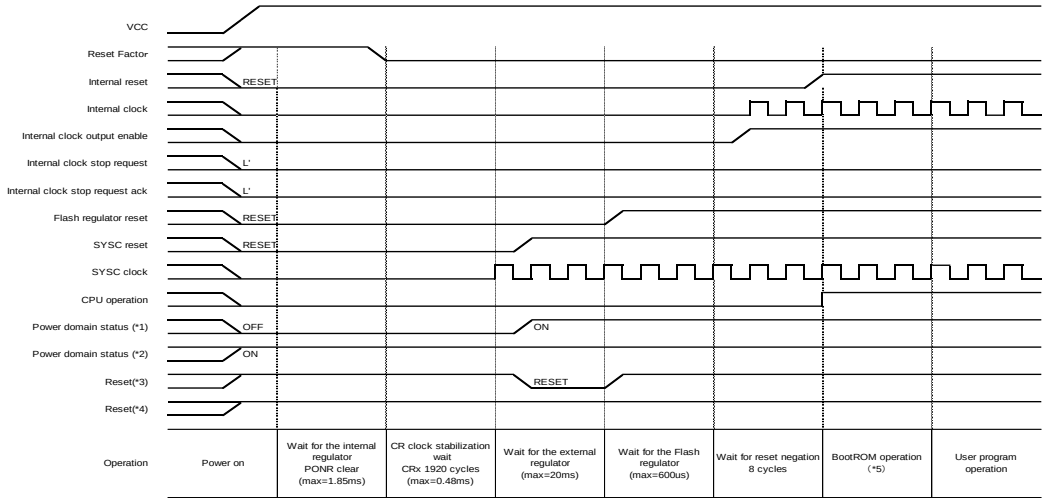
- **1: Contents of the Backup RAM are guaranteed. The other RAM contents are not guaranteed.*
- **2: The reset does not impact to RAM contents. But ETB (Embedded Trace Buffer) RAM contents are not guaranteed in case of Software debugger reset.*
- **3: Contents of TCRAM, System SRAM and Backup RAM are guaranteed. The other RAM contents are not guaranteed.*
- *nSRST pin input reset cannot be used. Therefore, please ignore the description about nSRST in RESET chapter.*

Note:

- *Supported reset factors depend on product. For details, see product specification.*

3.4.1. Hard Reset Operation

Figure 3-2 Operation at Power-on Reset



*1: External power supply PowerDomain area status (PD2)

*2: Internal power supply PowerDomain area status (PD4,PD6)

*3: External power supply PowerDomain reset

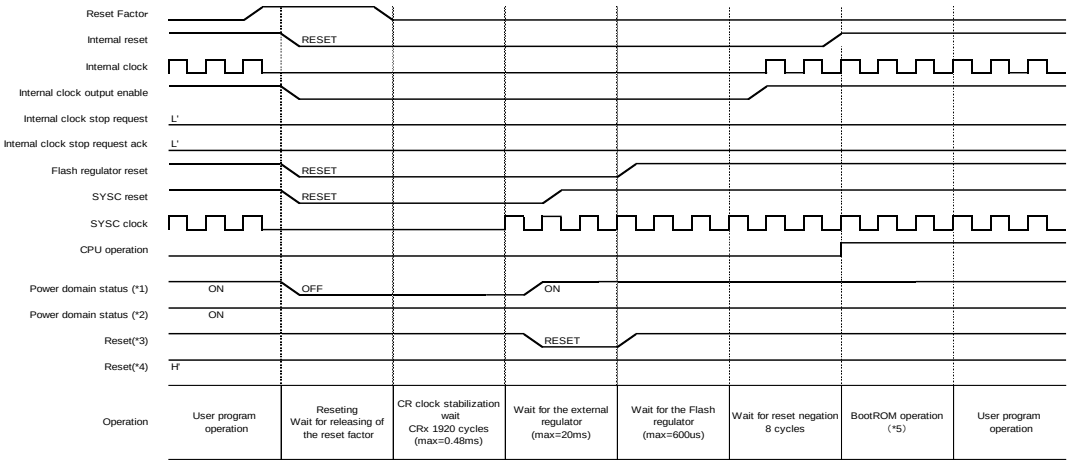
*4: Internal power supply PowerDomain reset

*5: Processing time: (Example) 1.31ms at SHE enable and BDR_SBMM=0

Notes:

- *As platform, the frequency of Fast CR clock is selectable in the range between 4MHz and 8MHz. However, the actual specification for Fast CR clock is device-specific for each product. Please refer to Datasheet for any device-specific information.*
- *As platform, SHE is selectable the enable or the disable. However, the enable or the disable of SHE is device-specific for each product. Please refer to Datasheet for any device-specific information.*

Figure 3-3 Operation at Internal Power Supply Low-voltage Detection Reset or INITX



*1: External power supply PowerDomain area status (PD2)

*2: Internal power supply PowerDomain area status (PD4,PD6)

*3: External power supply PowerDomain reset

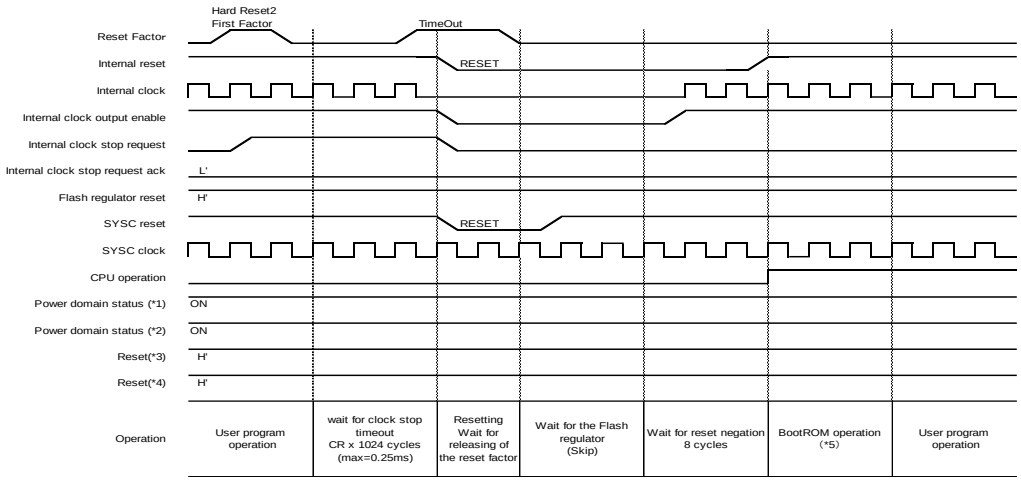
*4: Internal power supply PowerDomain reset

*5: Processing time: (Example) 1.31ms at SHE enable and BDR_SBMM=0

Notes:

- *As platform, the frequency of Fast CR clock is selectable in the range between 4MHz and 8MHz. However, the actual specification for Fast CR clock is device-specific for each product. Please refer to Datasheet for any device-specific information.*
- *As platform, SHE is selectable the enable or the disable. However, the enable or the disable of SHE is device-specific for each product. Please refer to Datasheet for any device-specific information.*

Figure 3-4 Operation at Clock Stop Wait Timeout Reset



*1: External power supply PowerDomain area status (PD2)

*2: Internal power supply PowerDomain area status (PD4,PD6)

*3: External power supply PowerDomain reset

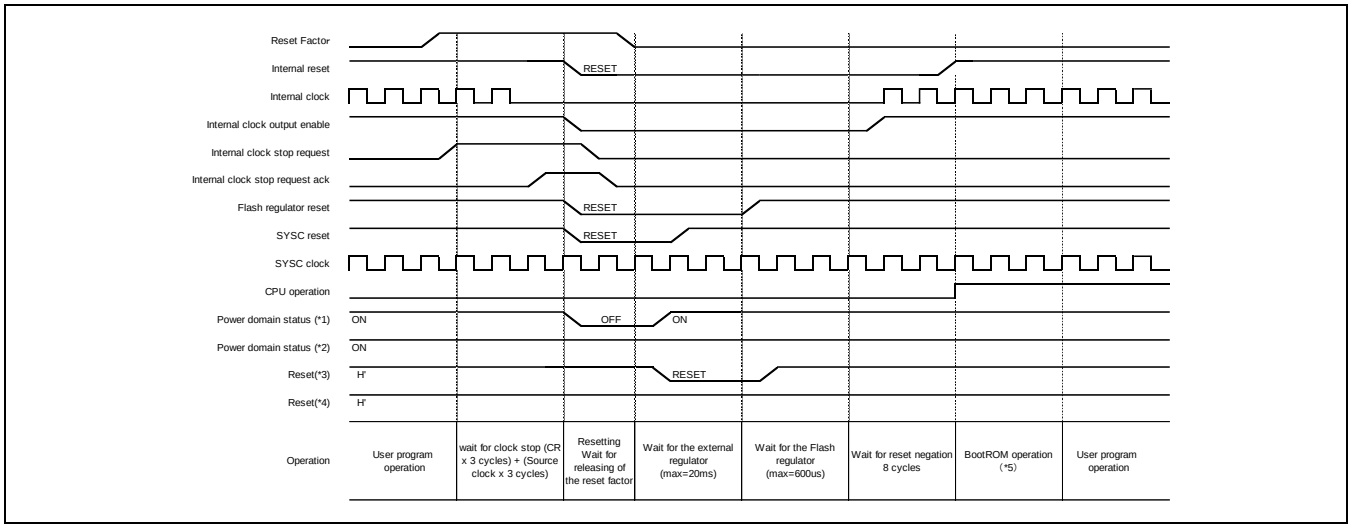
*4: Internal power supply PowerDomain reset

*5: Processing time: (Example) 1.31ms at SHE enable and BDR_SBMM=0

Notes:

- *As platform, the frequency of Fast CR clock is selectable in the range between 4MHz and 8MHz. However, the actual specification for Fast CR clock is device-specific for each product. Please refer to Datasheet for any device-specific information.*
- *As platform, SHE is selectable the enable or the disable. However, the enable or the disable of SHE is device-specific for each product. Please refer to Datasheet for any device-specific information.*

Figure 3-5 Operation at Hardware Watchdog Reset



*1: External power supply PowerDomain area status (PD2)

*2: Internal power supply PowerDomain area status (PD4,PD6)

*3: External power supply PowerDomain reset

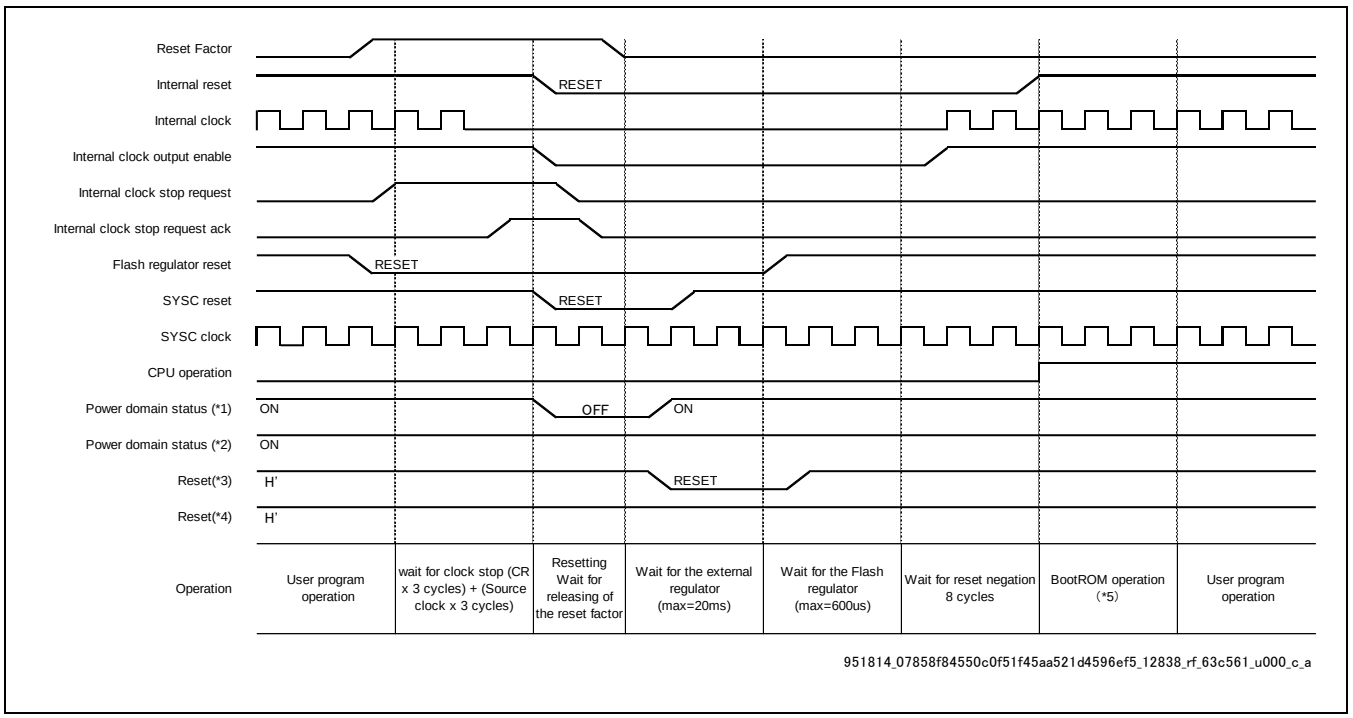
*4: Internal power supply PowerDomain reset

*5: Processing time: (Example) 1.31ms at SHE enable and BDR_SBMM=0

Note:

- *As platform, SHE is selectable the enable or the disable. However, the enable or the disable of SHE is device-specific for each product. Please refer to Datasheet for any device-specific information.*

Figure 3-6 Extended Internal Power Supply Low-Voltage Reset



*1: External power supply PowerDomain area status (PD2)

*2: Internal power supply PowerDomain area status (PD4,PD6)

*3: External power supply PowerDomain reset

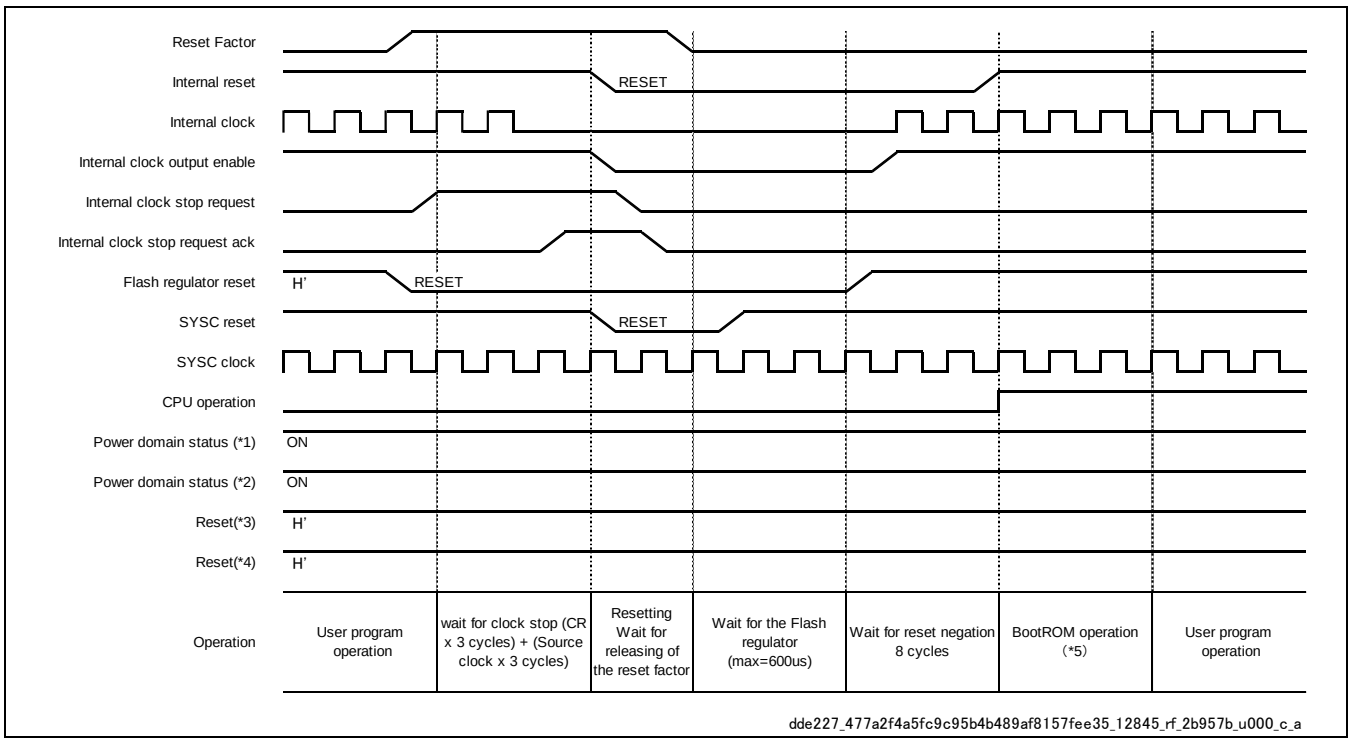
*4: Internal power supply PowerDomain reset

*5: Processing time: (Example) 1.31ms at SHE enable and BDR_SBMM=0

Note:

- *As platform, SHE is selectable the enable or the disable. However, the enable or the disable of SHE is device-specific for each product. Please refer to Datasheet for any device-specific information.*

Figure 3-7 Operation at RSTX Pin Input Reset or External power supply low-voltage detection reset



*1: External power supply PowerDomain area status (PD2)

*2: Internal power supply PowerDomain area status (PD4,PD6)

*3: External power supply PowerDomain reset

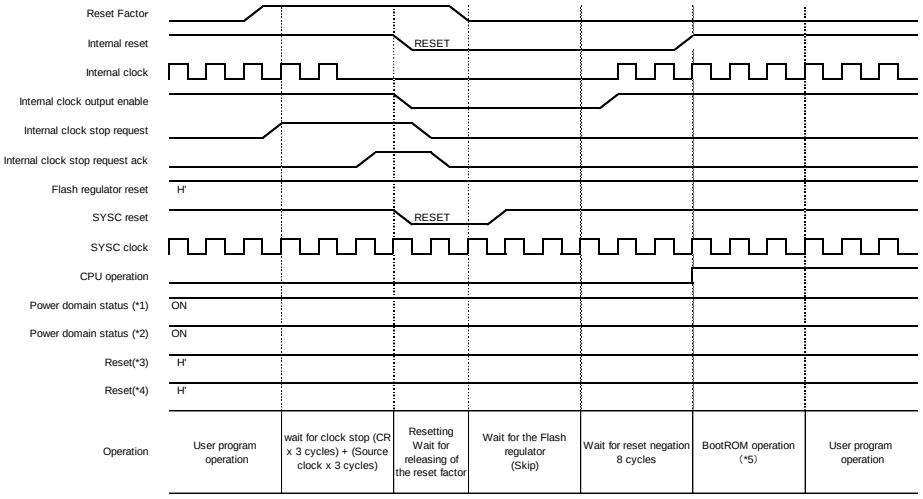
*4: Internal power supply PowerDomain reset

*5: Processing time: (Example) 1.31ms at SHE enable and BDR_SBMM=0

Notes:

- *The clocks in figure do not indicate actual numbers of clock cycle of the operations.*
- *As platform, SHE is selectable the enable or the disable. However, the enable or the disable of SHE is device-specific for each product. Please refer to Datasheet for any device-specific inform*

Figure 3-8 Operation at Other Hard Reset Factors



*1: External power supply PowerDomain area status (PD2)

*2: Internal power supply PowerDomain area status (PD4,PD6)

*3: External power supply PowerDomain reset

*4: Internal power supply PowerDomain reset

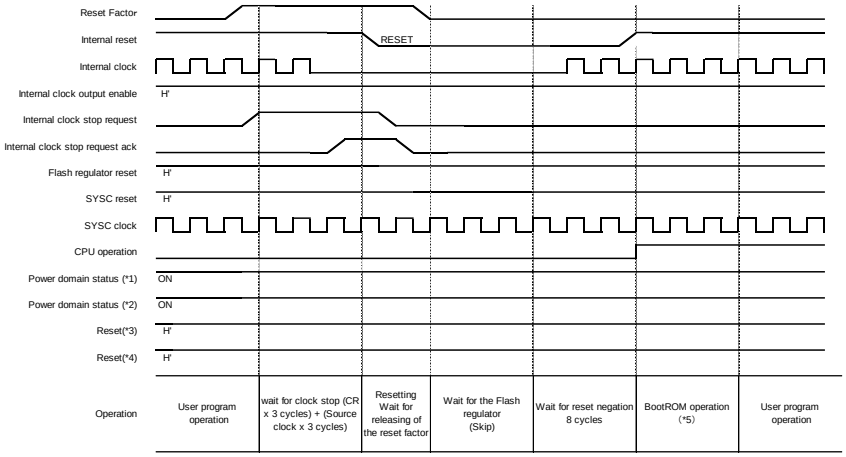
*5: Processing time: (Example) 1.31ms at SHE enable and BDR_SBMM=0

Notes:

- *The clocks in figure do not indicate actual numbers of clock cycle of the operations.*
- *As platform, SHE is selectable the enable or the disable. However, the enable or the disable of SHE is device-specific for each product. Please refer to Datasheet for any device-specific information.*

3.4.2. Soft Reset Operation

Figure 3-9 Operation at Software Reset



*1: External power supply PowerDomain area status (PD2)

*2: Internal power supply PowerDomain area status (PD4,PD6)

*3: External power supply PowerDomain reset

*4: Internal power supply PowerDomain reset

*5: Processing time: (Example) 1.31ms at SHE enable and BDR_SBMM=0

Notes:

- *The clocks in figure do not indicate actual numbers of clock cycle of the operations.*
- *As platform, SHE is selectable the enable or the disable. However, the enable or the disable of SHE is device-specific for each product. Please refer to Datasheet for any device-specific information.*

4. Registers

It explains the list of the register of reset.

Table 4-1 Reset Register List

Register Abbreviation	Register Name	Refer to:
SYSC_RSTCNTR	Reset control register	4.1
SYSC_RSTCAUSEUR	User reset factor register	4.2
SYSC_EXCSVRSTCAUSEUR	User extended CSV reset factor register	4.3
SYSC_PDRSTCAUSEUR	User PowerDomain reset factor register	4.4
SYSC_RSTCAUSEBT	BootROM reset factor register	4.5
SYSC_EXCSVRSTCAUSEBT	BootROM extended CSV reset factor register	4.6
SYSC_PDRSTCAUSEBT	BootROM PowerDomain reset factor register	4.7
SYSC_PDRSTSTATUS	PowerDomain reset status register	4.8

Table 4-2 Reset Register Memory Layout

Offset	Register Name			
	+3	+2	+1	+0
0x0000_0000	SYSC_RSTCNTR 00000000_00000000_00000000_00000000			
0x0000_0004	Reserved			
0x0000_0008	Reserved			
0x0000_000C	Reserved			
0x0000_0010	SYSC_RSTCAUSEUR 00000000_00000000_00000000_00000001			
0x0000_0014	SYSC_EXCSVRSTCAUSEUR 00000000_00000000_00000000_00000000			
0x0000_0018	SYSC_PDRSTCAUSEUR 00000000_00000000_00000000_0000000*			
0x0000_001C	Reserved			
0x0000_0020	SYSC_RSTCAUSEBT 00000000_00000000_00000000_00000001			
0x0000_0024	SYSC_EXCSVRSTCAUSEBT 00000000_00000000_00000000_00000000			
0x0000_0028	SYSC_PDRSTCAUSEBT 00000000_00000000_00000000_0000000 *			
0x0000_002C	Reserved			
0x0000_0030	Reserved			
0x0000_0034	SYSC_PDRSTSTATUS 00000000_00000000_00000000_00000000			

Notes:

- *Reserved bits are always read "0"; Write "0" when writing.*
- *The registers are protected by protection key setting register (SYSC0_PROTKEYR). For details on the protection, see the following chapter 6: "Low-power Consumption."*
- *Access to reserved address results bus error response.*
- **: When PD2 voltage is internal LDO, initial value is "0". When PD2 voltage is external LDO initial value is "1".*

4.1. Reset Control Register (SYSC_RSTCNTR)

This register controls resets of the CPU. It generates a soft reset, software trigger hard reset, and debugging reset. Protection key code input is required for writing.

Register Structure

Bit	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit 24
Field	DBGR							
R/W attribute	R0,W							
Protection attribute	WPS							
Initial value	00000000							

Bit	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
Field	SWHRST							
R/W attribute	R0,W							
Protection attribute	WPS							
Initial value	00000000							

Bit	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
Field	Reserved							
R/W attribute	R0,W0							
Protection attribute	WPS							
Initial value	00000000							

Bit	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Field	SWRST							
R/W attribute	R0,W							
Protection attribute	WPS							
Initial value	00000000							

Register Function

[bit31:24] DBGR: Software debugger Reset Register Bit

DBGR	Explanation
Write	The bits are used to generate the software debugger reset. Write 0xDA in this register to generate it. Refer to "Reset factor" for details of it.
Read	This register is automatically cleared after writing. The read value is always "0".

[bit23:16] SWHRST: Software Trigger hard Reset Register Bit

SWHRST	Explanation
Write	The bits are used to generate the software trigger hard reset. Write 0xA5 in this register to generate it. Refer to "Reset factor" for details of it.
Read	This register is automatically cleared after writing. The read value is always "0".

[bit15:8] Reserved

[bit7:0] SWRST: Software Reset Register Bit

SWRST	Explanation
Write	The bits are used to generate the software reset. Write 0x5A in this register to generate it. Refer to "Reset factor" for details of it.
Read	This register is automatically cleared after writing. The read value is always "0".

Notes:

- *Protection key code input is required for writing.*
- *It results bus error response for the violation to above restriction.*

4.2. User Reset Factor Register (SYSC_RSTCAUSEUR)

This register displays the last reset factor. Clear this reset factor before the next reset. Without doing it, it will be difficult to identify the next reset factors because the register bits are overwritten by new reset factors. The register bits except RVD are initialized by the power-on reset (PONR). RVD bit is initialized by the RSTX Pin Input Reset(RSTX) or the INITX(INITX). Protection key code input is required for writing.

Register Structure

Bit	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit 24
Field	LVDL2R	LVDL1R	CSV SCRR	CSV FCRR	CSVSR0	CSVPR0	CSVSOR	CSVMOR
R/W attribute	R,W	R,W	R,W	R,W	R,W	R,W	R, W	R, W
Protection attribute	WPS							
Initial value	0	0	0	0	0	0	0	0

Bit	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
Field	Reserved			SHRST	Reserved			SRST
R/W attribute	R0,W0			R,W	R0,W0			R,W
Protection attribute	WPS							
Initial value	0	0	0	0	0	0	0	0

Bit	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
Field	Reserved			SWDR	HWDR	PRFERR	SRSTX	IMR
R/W attribute	R0,W0			R,W	R,W	R,W	R,W	R,W
Protection attribute	WPS							
Initial value	0	0	0	0	0	0	0	0

Bit	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Field	LVDH2R	LVDH1R	Reserved	RSTX	CKTOR	INITX	RVD	PONR
R/W attribute	R,W	R,W	R0,W0	R,W	R,W	R,W	R,W	R,W
Protection attribute	WPS							
Initial value	0	0	0	0	0	0	0	1

Register Function

[bit31] LVDL2R: Extended Internal Power Supply Low-Voltage Detection Reset Detection Bit

This bit is used to confirm detection of the extended internal power supply low-voltage detection reset. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

[bit30] LVDL1R: Internal Power Supply Low-Voltage Detection Reset Detection Bit

This bit is used to confirm detection of the internal power supply low-voltage detection reset. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

[bit29] CSVSCRR: Slow CR Clock Supervisor Reset Detection Bit

This bit is used to confirm detection of the Slow CR clock supervisor reset. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

[bit28] CSVFCRR: Fast CR Clock Supervisor Reset Detection Bit

This bit is used to confirm detection of the Fast CR clock supervisor reset. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

[bit27] CSVSR0: SSCG0 Clock Supervisor Reset Detection Bit

This bit is used to confirm detection of the SSCG0 clock supervisor reset. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

[bit26] CSVPR0: PLL0 Clock Supervisor Reset Detection Bit

This bit is used to confirm detection of the PLL0 clock supervisor reset. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

[bit25] CSVSOR: Sub Clock Supervisor Reset Detection Bit

This bit is used to confirm detection of the Sub clock supervisor reset. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

[bit24] CSVMOR: Main Clock Supervisor Reset Detection Bit

This bit is used to confirm detection of the main clock supervisor reset. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

[bit23:21] Reserved

[bit20] SHRST: Software Trigger hard Reset Detection Bit

This bit is used to confirm detection of the software trigger hard reset. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

[bit19:17] Reserved

[bit16] SRST: Software Reset Detection Bit

This bit is used to confirm detection of the software reset. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

[bit15]:13 Reserved

[bit12] SWDR: Software Watchdog Reset Detection Bit

This bit is used to confirm detection of the software watchdog reset. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

[bit11] HWDR: Hardware Watchdog Reset Detection Bit

This bit is used to confirm detection of the hardware watchdog reset. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

[bit10] PRFERR: Profile Error Reset Detection Bit

This bit is used to confirm detection of the profile error reset. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

[bit9] SRSTX: nSRST Pin Input Reset Detection Bit

This bit is used to confirm detection of the nSRST pin input reset. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

Note:

- *nSRST pin input reset cannot be used. Therefore, please ignore the description about nSRST in RESET chapter.*

[bit8] IMR: Illegal Mode Detection Reset Detection Bit

This bit is used to confirm detection of the illegal mode detection reset. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

[bit7] LVDH2R: Extended External Power Supply Low-Voltage Detection Reset Detection Bit

This bit is used to confirm detection of the extended external power supply low-voltage detection reset. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

[bit6] LVDH1R: External Power Supply Low-Voltage Detection Reset Detection Bit

This bit is used to confirm detection of the external power supply low-voltage detection reset. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

[bit5] Reserved

[bit4] RSTX: RSTX Pin Input Reset Detection Bit

This bit is used to confirm detection of the RSTX pin input reset. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

[bit3] CKTOR: Clock Stop Wait Timeout Reset Detection Bit

This bit is used to confirm detection of the clock stop wait timeout reset. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

[bit2] INITX: INITX Detection Bit

This bit is used to confirm detection of the INITX. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

[bit1] RVD: RAM Retention Low-Voltage Detection Reset Detection Bit

This bit is used to confirm detection of the RAM retention low-voltage detection reset. Before the next reset is generated, clear this by writing "0".This bit is cleared by external reset (RSTX/INITX). When reset is generated by RVD, the PONR(bit0) detection bit is also set.

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

[bit0] PONR: Power-on Reset Detection Bit

This bit is used to confirm detection of the power-on reset. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

Notes:

- *Protection key code input is required for writing.*

- *It results bus error response for the violation to above restriction.*
- *Initial value of the bit is undefined when the electrical characteristics specification (both DC and AC) of the power supply is not kept.*

Notes:

- *Supported reset factors depend on product. For details, see product specification.*
- *Non support reset factor can always be read zero.*

4.3. User Extended CSV Reset Factor Register (SYSC_EXCSVRSTCAUSEUR)

This register displays the last reset factor. Clear this reset factor before the next reset. Without doing it, it will be difficult to identify the next reset factors because the register bits are overwritten by new reset factors. The register bits are initialized by the power-on reset (PONR). Protection key code input is required for writing.

Register Structure

Bit	bit 31	bit 8
Field	Reserved	
R/W attribute	R0, W0	
Protection attribute	WPS	
Initial value	00000000_00000000_00000000	

Bit	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Field	CSVSR3	CSVSR2	CSVSR1	Reserved	CSVPR3	CSVPR2	CSVPR1	Reserved
R/W attribute	R,W	R,W	R,W	R0, W0	R,W	R,W	R,W	R0, W0
Protection attribute	WPS							
Initial value	0	0	0	0	0	0	0	0

Register Function

[bit31:8] Reserved

[bit7] CSVSR3: SSCG3 Clock Supervisor Reset Detection Bit

This bit is used to confirm detection of the SSCG3 clock supervisor reset. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

[bit6] CSVSR2: SSCG2 Clock Supervisor Reset Detection Bit

This bit is used to confirm detection of the SSCG2 clock supervisor reset. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

[bit5] CSVSR1: SSCG1 Clock Supervisor Reset Detection Bit

This bit is used to confirm detection of the SSCG1 clock supervisor reset. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

[bit4] Reserved

[bit3] CSVPR3: PLL3 Clock Supervisor Reset Detection Bit

This bit is used to confirm detection of the PLL3 clock supervisor reset. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

[bit2] CSVPR2: PLL2 Clock Supervisor Reset Detection Bit

This bit is used to confirm detection of the PLL2 clock supervisor reset. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

[bit1] CSVPR1: PLL1 Clock Supervisor Reset Detection Bit

This bit is used to confirm detection of the PLL1 clock supervisor reset. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

[bit0] Reserved

Notes:

- *Protection key code input is required for writing.*
- *It results bus error response for the violation to above restriction.*

Notes:

- *Supported reset factors depend on product. For details, see product specification.*
- *Non support reset factor can always be read zero.*

4.4. User PowerDomain Reset Factor Register (SYSC_PDRSTCAUSEUR)

This register displays the status of the reset factors by the power shutdown. Reset is issued to the shutdown area at recovery from power shutdown. The register bits are initialized by the power-on reset (PONR). Protection key code input is required for writing.

Register Structure

Bit	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit 24
Field	Reserved							
R/W attribute	R0,W0							
Protection attribute	WPS							
Initial value	0	0	0	0	0	0	0	0

Bit	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
Field	Reserved						PD6R1	PD6R0
R/W attribute	R0,W0						R,W	R,W
Protection attribute	WPS							
Initial value	0	0	0	0	0	0	0	0

Bit	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
Field	PD5R3	PD5R2	PD5R1	PD5R0	Reserved		PD4R1	PD4R0
R/W attribute	R,W	R,W	R,W	R,W	R0,W0		R,W	R,W
Protection attribute	WPS							
Initial value	0	0	0	0	0	0	0	0

Bit	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Field	Reserved			PD3R0	Reserved			PD2R0
R/W attribute	R0,W0			R,W	R0,W0			R,W
Protection attribute	WPS							
Initial value	0	0	0	0	0	0	0	*

Note:

- ****: When PD2 voltage is internal LDO, initial value is "0". When PD2 voltage is external LDO initial value is "1".***

Register Function

[bit31:18] Reserved

[bit17] PD6R1: PowerDomain6_1 Reset Detection Bit

This bit is used to confirm detection of the PowerDomain reset 6_1. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

[bit16] PD6R0: PowerDomain6_0 Reset Detection Bit

This bit is used to confirm detection of the PowerDomain reset 6_0. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

[bit15] PD5R3: PowerDomain5_3 Reset Detection Bit

This bit is used to confirm detection of the PowerDomain reset 5_3. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

[bit14] PD5R2: PowerDomain5_2 Reset Detection Bit

This bit is used to confirm detection of the PowerDomain reset 5_2. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

[bit13] PD5R1: PowerDomain5_1 Reset Detection Bit

This bit is used to confirm detection of the PowerDomain reset 5_1. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

[bit12] PD5R0: PowerDomain5_0 Reset Detection Bit

This bit is used to confirm detection of the PowerDomain reset 5_0. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

[bit11:10] Reserved

[bit9] PD4R1: PowerDomain4_1 Reset Detection Bit

This bit is used to confirm detection of the PowerDomain reset 4_1. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

[bit8] PD4R0: PowerDomain4_0 Reset Detection Bit

This bit is used to confirm detection of the PowerDomain reset 4_0. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

[bit7:5] Reserved

[bit4] PD3R0: PowerDomain3_0 Reset Detection Bit

This bit is used to confirm detection of the PowerDomain reset 3_0. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

[bit3:1] Reserved

[bit0] PD2R0: PowerDomain2 Reset Detection Bit

This bit is used to confirm detection of the PowerDomain reset 2. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

Notes:

- *Protection key code input is required for writing.*
- *It results bus error response for the violation to above restriction.*

Notes:

- *Supported reset factors depend on product. For details, see product specification.*
- *Non support reset factor can always be read zero.*

4.5. BootROM Reset Factor Register (SYSC_RSTCAUSEBT)

This register displays the last reset factor. Clear this reset factor before the next reset. Without doing it, it will be difficult to identify the next reset factors because the register bits are overwritten by new reset factors. The register bits except RVD are initialized by the power-on reset (PONR). RVD bit is initialized by the RSTX Pin Input Reset(RSTX) or the INITX(INITX). Protection key code input is required for writing.

Register Structure

Bit	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit 24
Field	LVDL2R	LVDL1R	CSV SCRR	CSV FCRR	CSVSR0	CSVPR0	CSVSOR	CSVMOR
R/W attribute	R,W	R,W	R,W	R,W	R,W	R,W	R, W	R, W
Protection attribute	WPS							
Initial value	0	0	0	0	0	0	0	0

Bit	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
Field	Reserved			SHRST	Reserved			SRST
R/W attribute	R0,W0			R,W	R0,W0			R,W
Protection attribute	WPS							
Initial value	0	0	0	0	0	0	0	0

Bit	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
Field	Reserved			SWDR	HWDR	PRFERR	SRSTX	IMR
R/W attribute	R0,W0			R,W	R,W	R,W	R,W	R,W
Protection attribute	WPS							
Initial value	0	0	0	0	0	0	0	0

Bit	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Field	LVDH2R	LVDH1R	Reserved	RSTX	CKTOR	INITX	RVD	PONR
R/W attribute	R,W	R,W	R0,W0	R,W	R,W	R,W	R,W	R,W
Protection attribute	WPS							
Initial value	0	0	0	0	0	0	0	1

Register Function

[bit31] LVDL2R: Extended Internal Power Supply Low-Voltage Detection Reset Detection Bit

This bit is used to confirm detection of the extended internal power supply low-voltage detection reset. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

[bit30] LVDL1R: Internal Power Supply Low-Voltage Detection Reset Detection Bit

This bit is used to confirm detection of the internal power supply low-voltage detection reset. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

[bit29] CSVSCRR: Slow CR Clock Supervisor Reset Detection Bit

This bit is used to confirm detection of the Slow CR clock supervisor reset. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

[bit28] CSVFCRR: Fast CR Clock Supervisor Reset Detection Bit

This bit is used to confirm detection of the Fast CR clock supervisor reset. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

[bit27] CSVSR0: SSCG0 Clock Supervisor Reset Detection Bit

This bit is used to confirm detection of the SSCG0 clock supervisor reset. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

[bit26] CSVPR0: PLL0 Clock Supervisor Reset Detection Bit

This bit is used to confirm detection of the PLL0 clock supervisor reset. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

[bit25] CSVSOR: Sub Clock Supervisor Reset Detection Bit

This bit is used to confirm detection of the sub clock supervisor reset. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

[bit24] CSVMOR: Main Clock Supervisor Reset Detection Bit

This bit is used to confirm detection of the main clock supervisor reset. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

[bit23:21] Reserved

[bit20] SHRST: Software Trigger hard Reset Detection Bit

This bit is used to confirm detection of the software trigger hard reset. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

[bit19:17] Reserved

[bit16] SRST: Software Reset Detection Bit

This bit is used to confirm detection of the software reset. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

[bit15:13] Reserved

[bit12] SWDR: Software Watchdog Reset Detection Bit

This bit is used to confirm detection of the software watchdog reset. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

[bit11] HWDR: Hardware Watchdog Reset Detection Bit

This bit is used to confirm detection of the Hardware watchdog reset. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

[bit10] PRFERR: Profile Error Reset Detection Bit

This bit is used to confirm detection of the profile error reset. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

[bit9] SRSTX: nSRST Pin Input Reset Detection Bit

This bit is used to confirm detection of the nSRST pin input reset. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

Note:

- *nSRST pin input reset cannot be used. Therefore, please ignore the description about nSRST in RESET chapter.*

[bit8] IMR: Illegal Mode Detection Reset Detection Bit

This bit is used to confirm detection of the illegal mode detection reset. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

[bit7] LVDH2R: Extended External Power Supply Low-Voltage Detection Reset Detection Bit

This bit is used to confirm detection of the extended external power supply low-voltage detection reset. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

[bit6] LVDH1R: External Power Supply Low-Voltage Detection Reset Detection Bit

This bit is used to confirm detection of the external power supply low-voltage detection reset. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

[bit5] Reserved

[bit4] RSTX: RSTX Pin Input Reset Detection Bit

This bit is used to confirm detection of the RSTX pin input reset. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

[bit3] CKTOR: Clock Stop Wait Timeout Reset Detection Bit

This bit is used to confirm detection of the clock stop wait timeout reset. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

[bit2] INITX: INITX Detection Bit

This bit is used to confirm detection of the INITX. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

[bit1] RVD: RAM Retention Low-Voltage Detection Reset Detection Bit

This bit is used to confirm detection of the RAM retention low-voltage detection reset. Before the next reset is generated, clear this by writing "0". This bit is cleared by external reset (RSTX/INITX). When reset is generated by RVD, the PONR(bit0) detection bit is also set.

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

[bit0] PONR: Power-on Reset Detection Bit

This bit is used to confirm detection of the power-on reset. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

Notes:

- *Protection key code input is required for writing.*
- *It results bus error response for the violation to above restriction.*
- *Initial value of the bit is undefined when the electrical characteristics specification (both DC and AC) of the power supply is not kept.*
- *BootROM is used by this register. BootROM initializes this register in the startup process. Can not be read "1" set the status of each reset factor from the user program.*

Notes:

- *Supported reset factors depend on product. For details, see product specification.*
- *Non support reset factor can always be read zero.*

4.6. BootROM Extended CSV Reset Factor Register (SYSC_EXCSVRSTCAUSEBT)

This register displays the last reset factor. Clear this reset factor before the next reset. Without doing it, it will be difficult to identify the next reset factors because the register bits are overwritten by new reset factors. The register bits are initialized by the power-on reset (PONR). Protection key code input is required for writing.

Register Structure

Bit	bit 31	bit 8
Field	Reserved	
R/W attribute	R0, W0	
Protection attribute	WPS	
Initial value	00000000_00000000_00000000	

Bit	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Field	CSVSR3	CSVSR2	CSVSR1	Reserved	CSVPR3	CSVPR2	CSVPR1	Reserved
R/W attribute	R,W	R,W	R,W	R0, W0	R,W	R,W	R,W	R0, W0
Protection attribute	WPS							
Initial value	0	0	0	0	0	0	0	0

Register Function

[bit31:8] Reserved

[bit7] CSVSR3: SSCG3 Clock Supervisor Reset Detection Bit

This bit is used to confirm detection of the SSCG3 clock supervisor reset. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

[bit6] CSVSR2: SSCG2 Clock Supervisor Reset Detection Bit

This bit is used to confirm detection of the SSCG2 clock supervisor reset. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

[bit5] CSVSR1: SSCG1 Clock Supervisor Reset Detection Bit

This bit is used to confirm detection of the SSCG1 clock supervisor reset. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

[bit4] Reserved

[bit3] CSVPR3: PLL3 Clock Supervisor Reset Detection Bit

This bit is used to confirm detection of the PLL3 clock supervisor reset. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

[bit2] CSVPR2: PLL2 Clock Supervisor Reset Detection Bit

This bit is used to confirm detection of the PLL2 clock supervisor reset. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

[bit1] CSVPR1: PLL1 Clock Supervisor Reset Detection Bit

This bit is used to confirm detection of the PLL1 clock supervisor reset. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

[bit0] Reserved

Notes:

- *Protection key code input is required for writing.*
- *It results bus error response for the violation to above restriction.*
- *BootROM is used by this register. BootROM initializes this register in the startup process. Can not be read "1" set the status of each reset factor from the user program.*

Notes:

- *Supported reset factors depend on product. For details, see product specification.*
- *Non support reset factor can always be read zero.*

4.7. BootROM PowerDomain Reset Factor Register (SYSC_PDRSTCAUSEBT)

This register displays the status of the reset factors by the power shutdown. Reset is issued to the shutdown area at recovery from power shutdown. This register is initialized by power-on reset (PONR). Protection key code input is required for writing.

Register Structure

Bit	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit 24
Field	Reserved							
R/W attribute	R0,W0							
Protection attribute	WPS							
Initial value	0	0	0	0	0	0	0	0

Bit	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
Field	Reserved						PD6R1	PD6R0
R/W attribute	R0,W0						R,W	R,W
Protection attribute	WPS							
Initial value	0	0	0	0	0	0	0	0

Bit	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
Field	PD5R3	PD5R2	PD5R1	PD5R0	Reserved		PD4R1	PD4R0
R/W attribute	R,W	R,W	R,W	R,W	R0,W0		R,W	R,W
Protection attribute	WPS							
Initial value	0	0	0	0	0	0	0	0

Bit	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Field	Reserved			PD3R0	Reserved			PD2R0
R/W attribute	R0,W0			R,W	R0,W0			R,W
Protection attribute	WPS							
Initial value	0	0	0	0	0	0	0	*

Note:

- ****:** When PD2 voltage is internal LDO, initial value is "0". When PD2 voltage is external LDO initial value is "1".*

Register Function

[bit31:18] Reserved

[bit17] PD6R1: PowerDomain6_1 Reset Detection Bit

This bit is used to confirm detection of the PowerDomain reset 6_1 reset. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

[bit16] PD6R0: PowerDomain6_0 Reset Detection Bit

This bit is used to confirm detection of the PowerDomain reset 6_0. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

[bit15] PD5R3: PowerDomain5_3 Reset Detection Bit

This bit is used to confirm detection of the PowerDomain reset 5_3. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

[bit14] PD5R2: PowerDomain5_2 Reset Detection Bit

This bit is used to confirm detection of the PowerDomain reset 5_2. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

[bit13] PD5R1: PowerDomain5_1 Reset Detection Bit

This bit is used to confirm detection of the PowerDomain reset 5_1. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

[bit12] PD5R0: PowerDomain5_0 Reset Detection Bit

This bit is used to confirm detection of the PowerDomain reset 5_0. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

[bit11:10] Reserved

[bit9] PD4R1: PowerDomain4_1 Reset Detection Bit

This bit is used to confirm detection of the PowerDomain reset 4_1. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

[bit8] PD4R0: PowerDomain4_0 Reset Detection Bit

This bit is used to confirm detection of the Q PowerDomain reset 4_0. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

[bit7:5] Reserved

[bit4] PD3R0: PowerDomain3_0 Reset Detection Bit

This bit is used to confirm detection of the PowerDomain reset 3_0. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

[bit3:1] Reserved

[bit0] PD2R0: PowerDomain2 Reset Detection Bit

This bit is used to confirm detection of the PowerDomain reset 2. Before the next reset is generated, clear this by writing "0".

Bit	Explanation
0	The reset has not been detected.
1	The reset has been detected.

Notes:

- *Protection key code input is required for writing.*
- *It results bus error response for the violation to above restriction.*
- *BootROM is used by this register. BootROM initializes this register in the startup process. Can not be read "1" set the status of each reset factor from the user program.*

Notes:

- *Supported reset factors depend on product. For details, see product specification.*
- *Non support reset factor can always be read zero.*

4.8. PowerDomain Reset Status Register (SYSC_PDRSTSTATUS)

This register displays the status of the reset issuing status to the PowerDomains by the power shutdown.

Register Structure

Bit	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit 24
Field	Reserved							
R/W attribute	R0,WX							
Protection attribute	WPS							
Initial value	0	0	0	0	0	0	0	0

Bit	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
Field	Reserved						PD6RS1	PD6RS0
R/W attribute	R0,WX						R,WX	R,WX
Protection attribute	WPS							
Initial value	0	0	0	0	0	0	0	0

Bit	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
Field	PD5RS3	PD5RS2	PD5RS1	PD5RS0	Reserved		PD4RS1	PD4RS0
R/W attribute	R,WX	R,WX	R,WX	R,WX	R0,WX		R,WX	R,WX
Protection attribute	WPS							
Initial value	0	0	0	0	0	0	0	0

Bit	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Field	Reserved			PD3RS0	Reserved			PD2RS0
R/W attribute	R0,WX			R,WX	R0,WX			R,WX
Protection attribute	WPS							
Initial value	0	0	0	0	0	0	0	0

Register Function

[bit31:18] Reserved

[bit17] PD6RS1: PowerDomain6_1 Reset Status Bit

This bit displays the reset status of PowerDomain6_1.

Bit	Explanation
0	The reset is released.
1	The reset is issued.

[bit16] PD6RS0: PowerDomain6_0 Reset Detection Bit

This bit displays the reset status of PowerDomain6_0.

Bit	Explanation
0	The reset is released.
1	The reset is issued.

[bit15] PD5RS3: PowerDomain5_3 Reset Detection Bit

This bit displays the reset status of PowerDomain5_3.

Bit	Explanation
0	The reset is released.
1	The reset is issued.

[bit14] PD5RS2: PowerDomain5_2 Reset Detection Bit

This bit displays the reset status of PowerDomain5_2.

Bit	Explanation
0	The reset is released.
1	The reset is issued.

[bit13] PD5RS1: PowerDomain5_1 Reset Detection Bit

This bit displays the reset status of PowerDomain5_1.

Bit	Explanation
0	The reset is released.
1	The reset is issued.

[bit12] PD5RS0: PowerDomain5_0 Reset Detection Bit

This bit displays the reset status of PowerDomain5_0.

Bit	Explanation
0	The reset is released.
1	The reset is issued.

[bit11:10] Reserved

[bit9] PD4RS1: PowerDomain4_1 Reset Detection Bit

This bit displays the reset status of PowerDomain4_1.

Bit	Explanation
0	The reset is released.
1	The reset is issued.

[bit8] PD4RS0: PowerDomain4_0 Reset Detection Bit

This bit displays the reset status of PowerDomain4_0.

Bit	Explanation
0	The reset is released.
1	The reset is issued.

[bit7:5] Reserved

[bit4] PD3RS0: PowerDomain3_0 Reset Detection Bit

This bit displays the reset status of PowerDomain3_0.

Bit	Explanation
0	The reset is released.
1	The reset is issued.

[bit3:1] Reserved

[bit0] PD2RS0: PowerDomain2 Reset Detection Bit

This bit displays the reset status of PowerDomain2.

Bit	Explanation
0	The reset is released.
1	The reset is issued.

Notes:

- *Supported reset factors depend on product. For details, see product specification.*
- *Non support reset factor can always be read zero.*

CHAPTER 5: Clock System

This chapter explains the clock system.

1. Overview
2. Configuration and Block Diagram
3. Operations
4. Setting Procedure Example
5. Registers
6. Usage Precautions

CLKSYS-TXXPT03P01R01L10-E1-XX

1. Overview

This section explains the overview of the clock system.

The clock system provides various clocks for MCU operation.

Source clocks are generated from MCU external/internal oscillation circuit. Source clocks are used to generate internal clock for MCU operation

The clock system generates the following source clocks.

- Fast-CR clock
- Slow-CR clock
- Main clock/Main clock * 1/2
- Sub clock
- PLL0 clock
- PLL1 clock
- PLL2 clock
- PLL3 clock
- SSCG PLL0 clock
- SSCG PLL1 clock
- SSCG PLL2 clock
- SSCG PLL3 clock

MCU external/internal oscillation circuit generates Fast-CR clock and Slow-CR clock, Main clock, Sub clock.

And 8 PLL circuits generate PLL clocks and SSCG PLL clocks.

Internal clocks generate from same source clock are provided to clock domain. At least clock domain has 1 internal clock.

The clock domains are the following in the clock system.

- Clock domain0
- Clock domain1
- Clock domain2
- Clock domain3
- Clock domain4
- Clock domain5
- TRC clock domain
- HSSPI clock domain
- MCUC clock domain
- CLKO clock domain

The only 1 source clock is selected for each clock domain. Each clock domain uses the selected source clock and divides to generate internal clock.

Features

The features of clock system are the following.

4 kinds of clock source (Fast-CR clock, slow-CR clock , Main clock ,Sub clock)

- All four clock sources have Source Clock Timer, which mask the clock till it is stable.

- All PLLs have stabilization counters, which masks the clock till it is stable.
- Clock gating for low power operation
- Fast-CR clock is used for complete device during initial power up, user can switch to other clocks later
- Clock Supervisor circuits are present for both external clock sources (Main osc & Sub osc) which make sure that input clocks are within configured frequency ranges
- Clock Supervisor circuits are also present for all PLLs to detect if PLL goes out of lock due to high jitter on the input clock

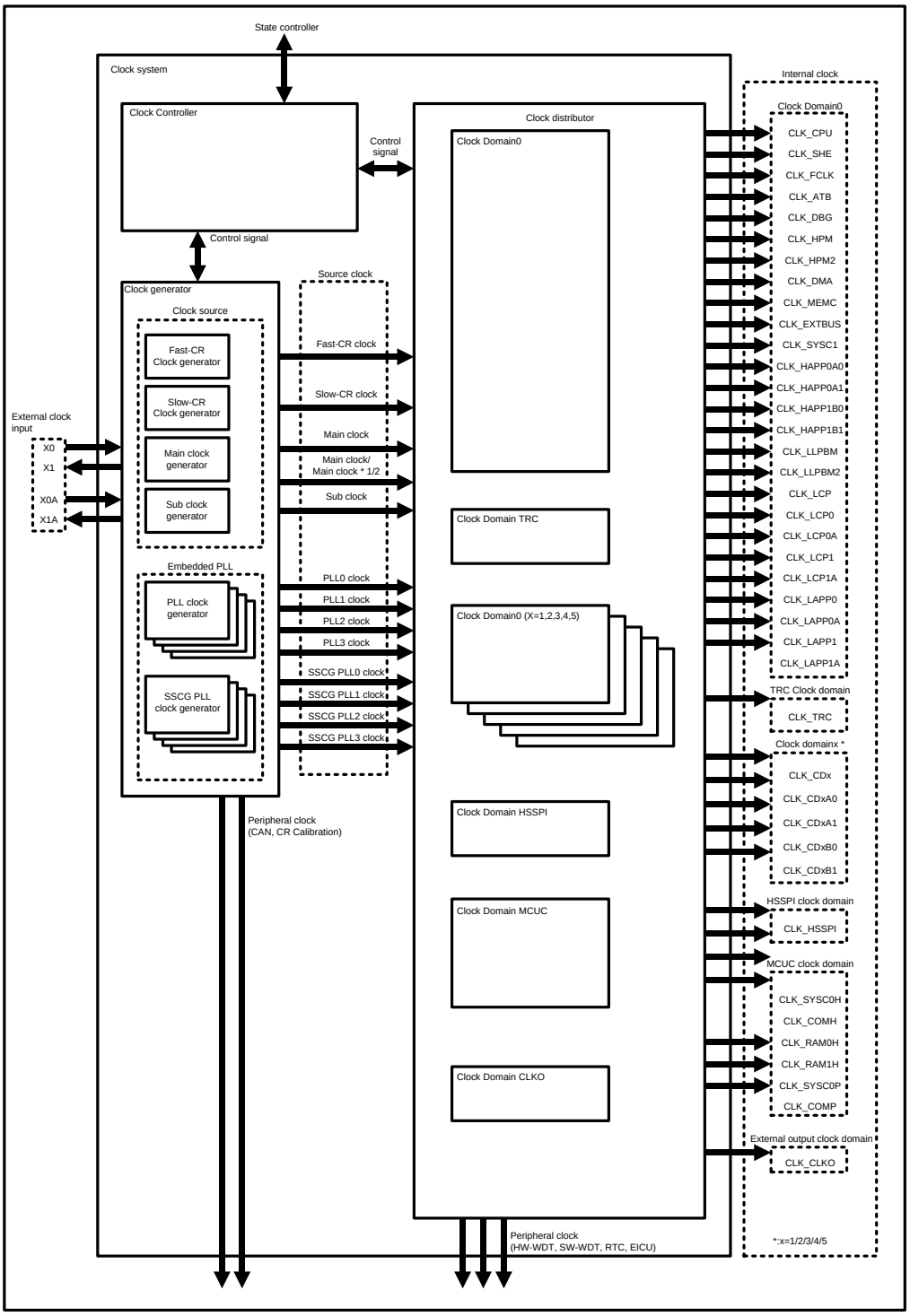
2. Configuration and Block Diagram

This section explains the block diagram of the clock system.

2.1. Clock System Configuration and Block Diagram

This section shows the block diagram of the clock system.

Figure 2-1 Clock System Block Diagram



Clock Controller
Clock controller generates control signal from configuration register value for clock system.

Clock Generator
Clock generator generates source clock from external/internal oscillation circuit.

Source Clocks
Source clocks are the following clocks.

- Fast-CR clock
- Slow-CR clock
- Main clock/Main clock * 1/2
- Sub clock
- PLLx clock (x=(product specification))
- SSCG PLLx clock (x=(product specification))

Domain Clock
Clock domain used for internal clocks which are divided into same source clock. Domain clock is selected each clock domain

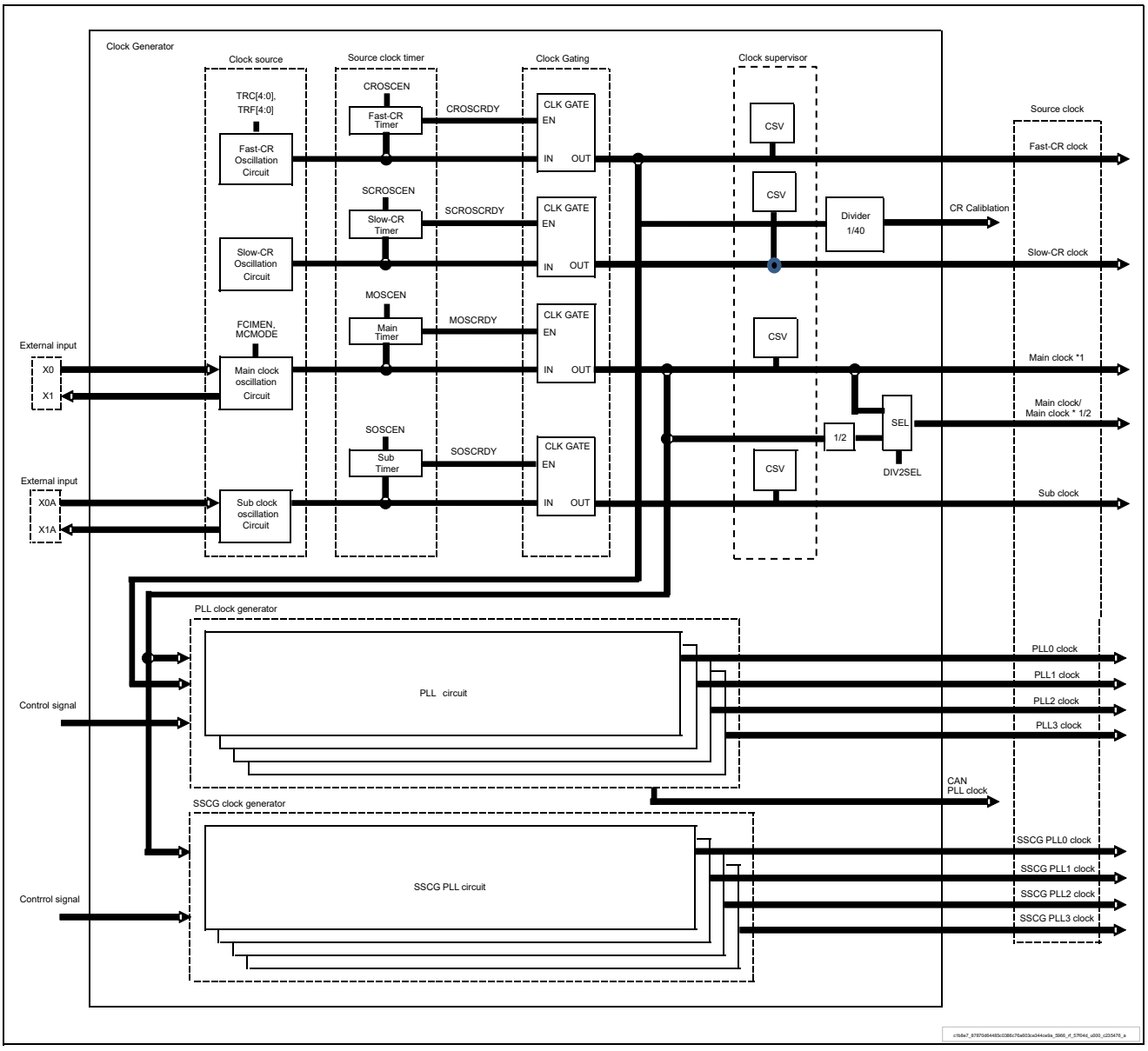
Clock Distributor
Internal clocks generate from dividing into domain clock.

Internal Clock
Internal clocks used for each resource operation in MCU.

2.2. Clock Generator Configuration and Block Diagram

This section shows the block diagram of the clock generator.

Figure 2-2 Clock Generator Block Diagram

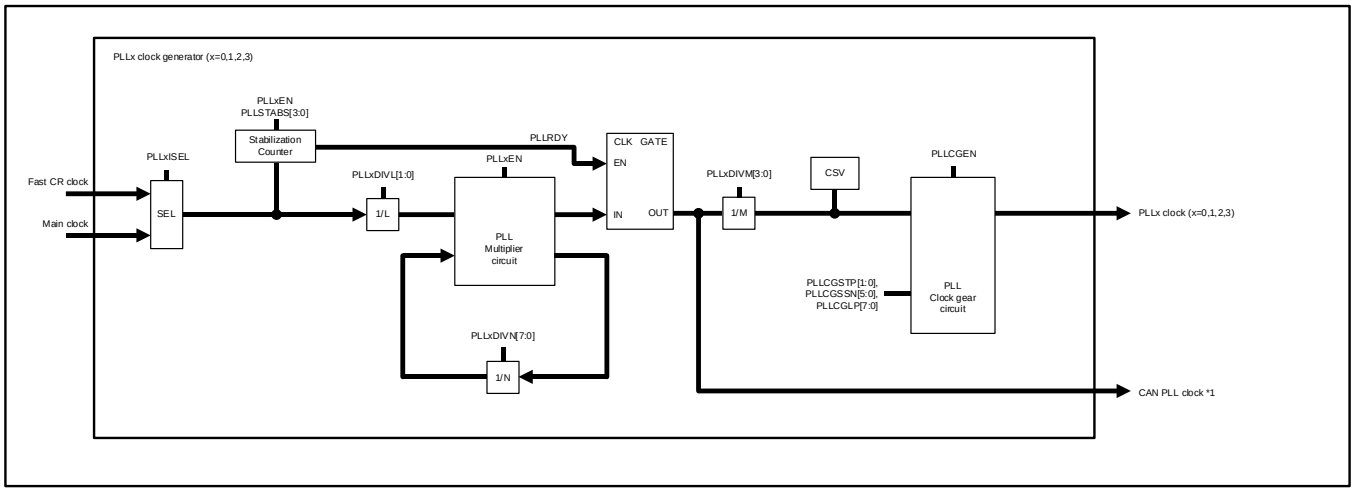


*1 : Real Time Clock and Software Watchdog use this as their Main clock source.

2.2.1. PLL Clock Generator Configuration and Block Diagram

This section shows the block diagram of the PLL clock generator.

Figure 2-3 PLLx (x=0,1,2,3)Clock Generator Block Diagram

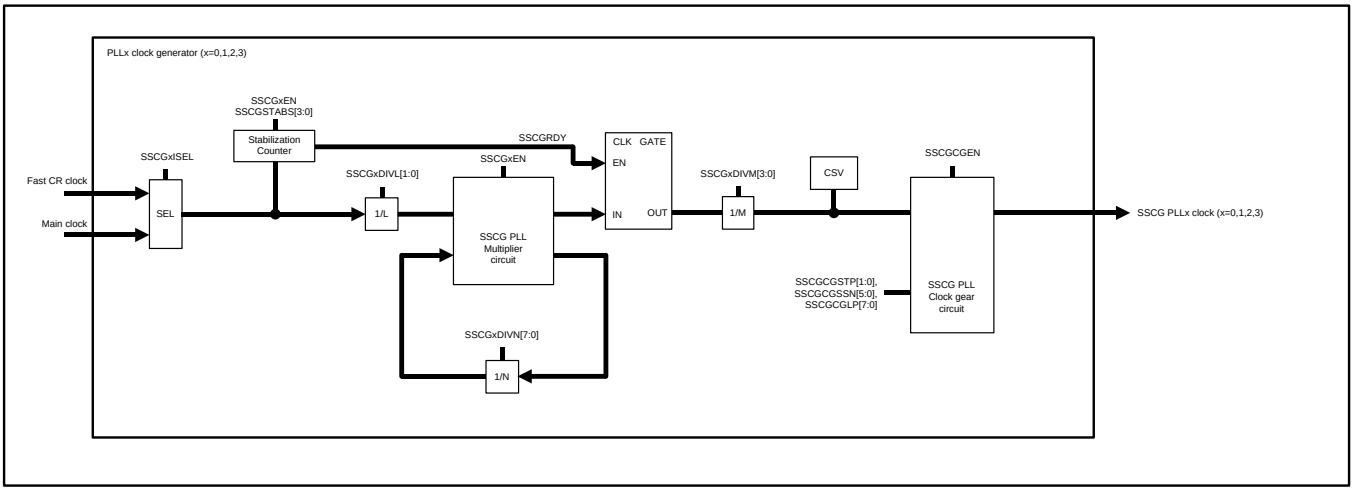


*1 : Only "(product specification)" supports the CAN PLL clock.

2.2.2. SSCG PLL Clock Generator Configuration and Block Diagram

This section shows the block diagram of the SSCG PLL clock generator.

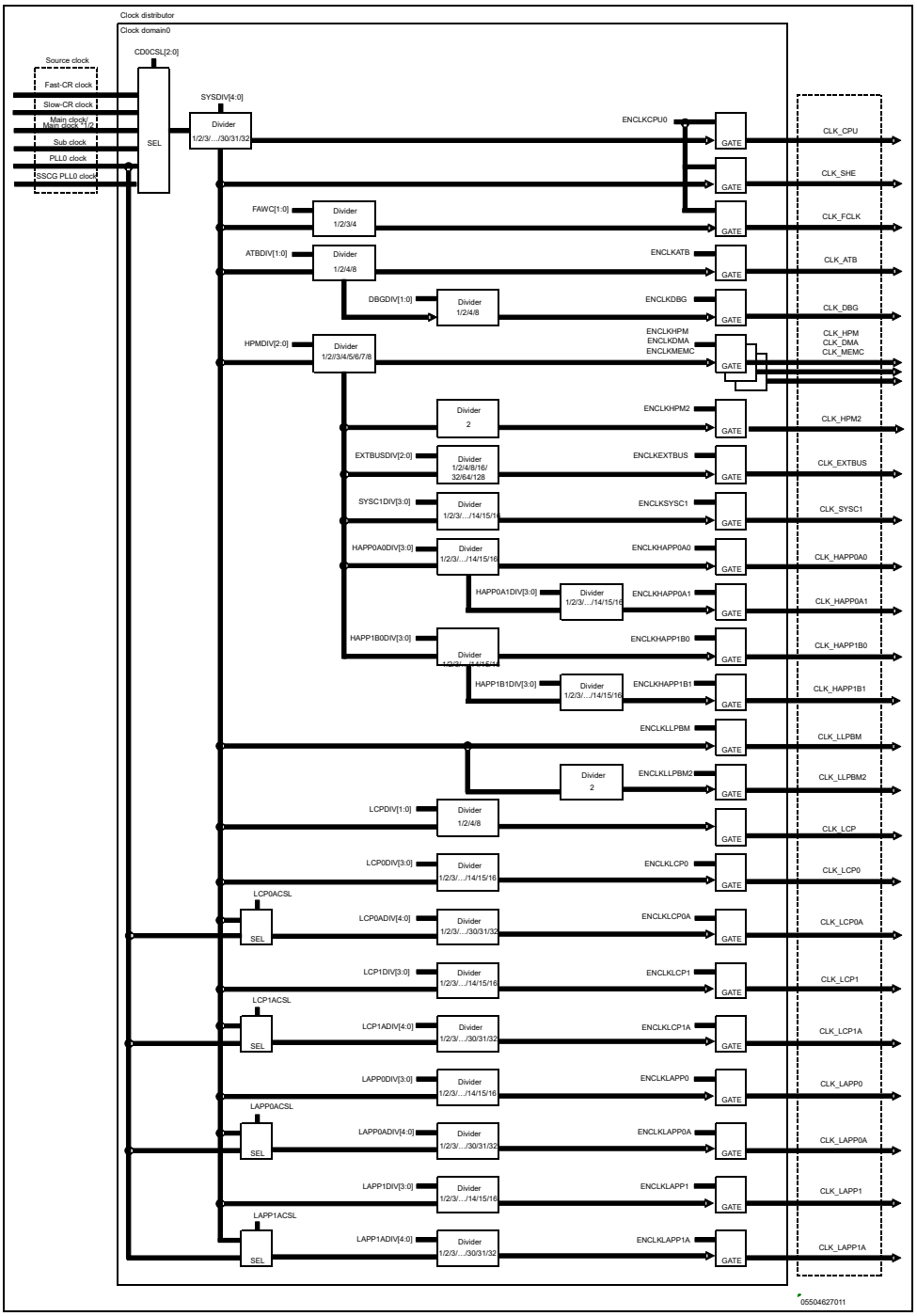
Figure 2-4 SSCG PLLx (x=0,1,2,3)Clock Generator Block Diagram



2.3. Clock Distributor Configuration and Block Diagram

This section shows the block diagram of the clock distributor.

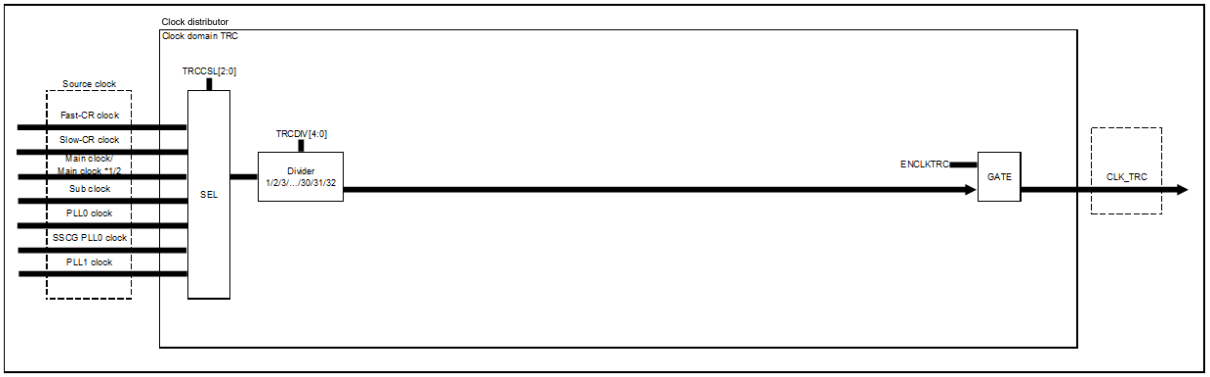
Figure 2-5 Clock domain0 Block Diagram



Note:

- *Do not use Sub clock for the domain clock.*

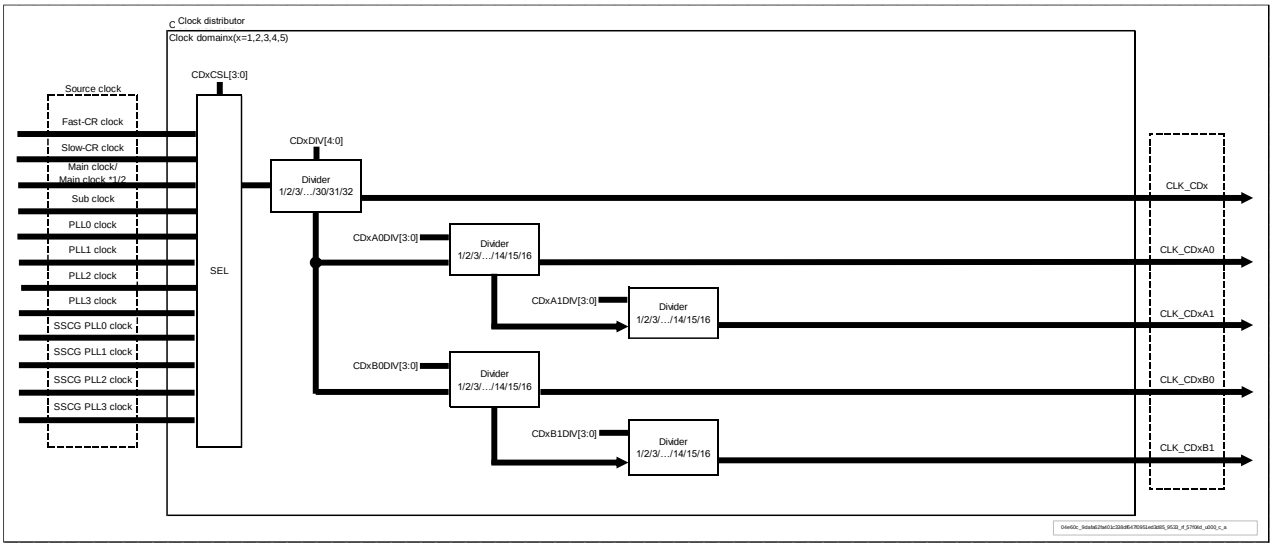
Figure 2-6 Clock Domain TRC Block Diagram



Note:

- *Do not use Sub clock for the domain clock.*

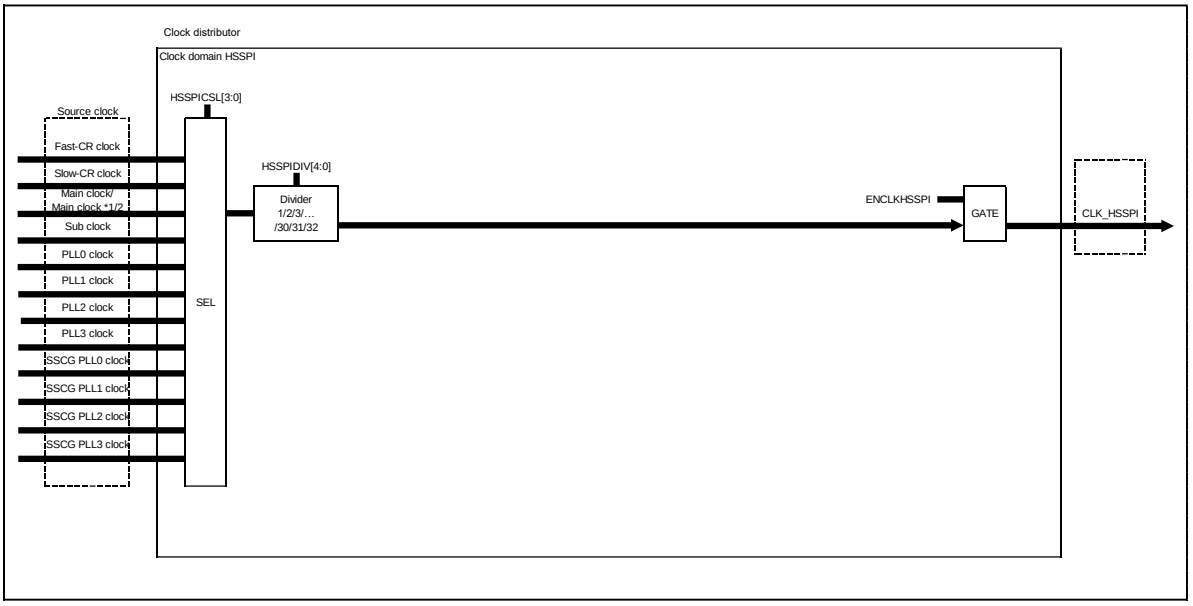
Figure 2-7 Clock domainx(x=1,2,3,4,5) Block Diagram



Note:

- *Do not use Sub clock for the domain clock.*

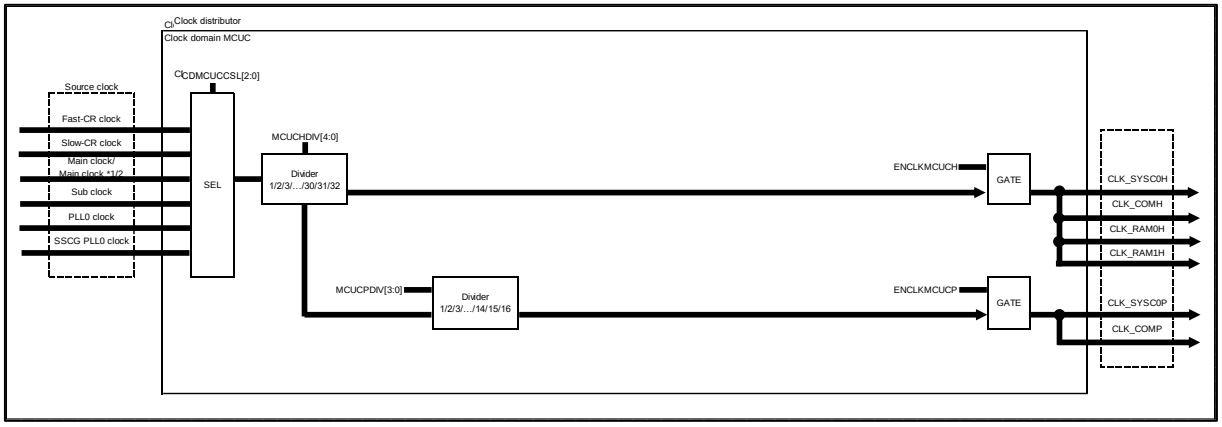
Figure 2-8 Clock Domain HSSPI Block Diagram



Note:

- *Do not use Sub clock for the domain clock.*

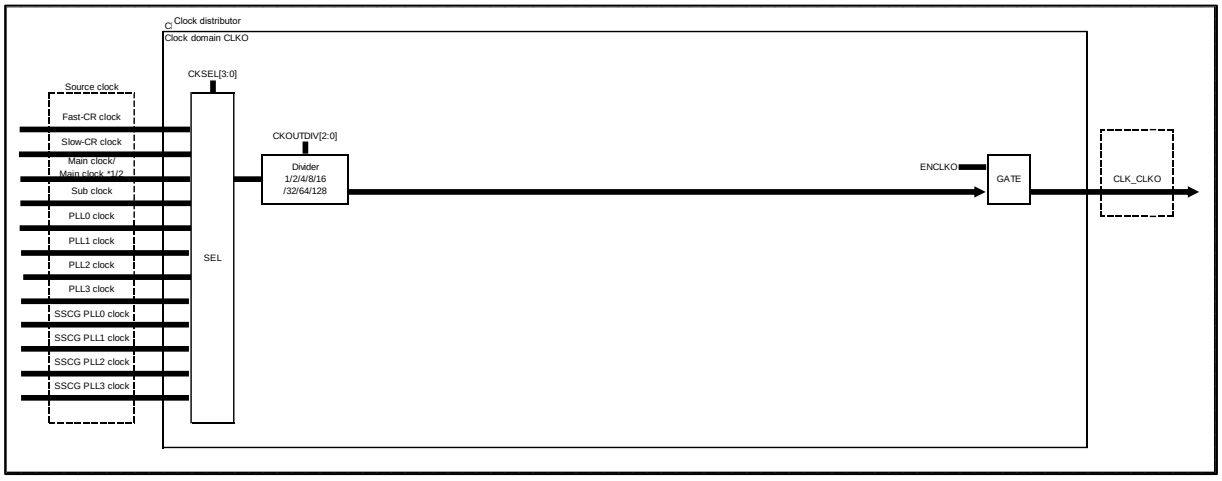
Figure 2-9 Clock Domain MCUC Block Diagram



Note:

- *Do not use Sub clock for the domain clock.*

Figure 2-10 Clock Domain CLKO Block Diagram



Note:

- *Do not use Sub clock for the domain clock.*

3. Operations

This section shows the configuration of the clock system.

3.1. Source Clock Generation

This section explains source clock generation.

Source Clock Enable/Disable

Source clocks not used in the device can be enabled and disabled in the profile with the corresponding enable bits in the profile.

The Main clock must be enabled if the PLL clock should be enabled, otherwise profile error interrupt will be asserted.

Table 3-1 Source Clock Enable/Disable

Source Clock	Initial State	RUN	PSS
Fast-CR clock	Enable	Enable	Programmable
Slow-CR clock	Enable	Enable	Programmable
Main clock/Main clock * 1/2	Enable	Programmable	Programmable
Sub clock	(product specification)	Programmable	Programmable
PLL0 clock	Disable	Programmable	Programmable
PLL1 clock	Disable	Programmable	Programmable
PLL2 clock	Disable	Programmable	Programmable
PLL3 clock	Disable	Programmable	Programmable
SSCG PLL0 clock	Disable	Programmable	Programmable
SSCG PLL1 clock	Disable	Programmable	Programmable
SSCG PLL2 clock	Disable	Programmable	Programmable
SSCG PLL3 clock	Disable	Programmable	Programmable

3.2. PLL/SSCG PLL Clock

This section explains PLL/SSCG PLL clock.

Clock system has several PLL/SSCG PLL multiplier circuits which is used to generate the PLL/SSCG PLL clock out of the Fast-CR clock / Main clock.

PLL Clock Configuration

Following steps are required to enable PLL:

1. Set the PLL stabilization time (SYSC_PLLSSCGSTCNTR.PLLSTABS)
2. Set the RUN/PSS profile register for PLLx configuration.(SYSC0_RUNPLLxCNTR/SYSC0_PSSPLLxCNTR)
3. Set the RUN/PSS profile register for PLLxEN (SYSC0_RUNCKSRER/SYSC0_PSSCKSRER)
4. Set profile Trigger Register or execute WFI instruction to update profile registers.
5. Device state control logic first enables the PLL with configured settings, waits for the PLL stabilization time and then switches the clock source domain to PLL source.

SSCG PLL Clock Configuration

Following steps are required to enable SSCG PLL:

1. Set the SSCG PLL stabilization time (SYSC_PLLSSCGSTCNTR.PLLSTABS)
2. Set the RUN/PSS profile register for SSCG PLLx configuration.(SYSC0_RUNSSCGxCNTR0 and SYSC0_RUNSSCGxCNTR1/SYSC0_PSSSSCGxCNTR0 and SYSC0_PSSSSCGxCNTR1)
3. Set the RUN/PSS profile register for SSCG PLLxEN (SYSC0_RUNCKSRER/SYSC0_PSSCKSRER)
4. Set profile Trigger Register or execute WFI instruction to update profile registers.
5. Device state control logic first enables the PLL with configured settings, waits for the SSCG PLL stabilization time and then switches the clock source domain to SSCG PLL source.

Configuration of PLL/SSCG PLL Clock Stabilization Time

For details refer to '5.1.4 PLL/SSCG stabilization time control register(SYSC_PLLSSCGSTCNTR)'.

Multiplying Configuration of PLL Clock

Table 3-2 Example of PLL Clock Configuration

Input Clock	Input Clock Divider Configuration	PLLin	PLL Multiplier Configuration	PLLout	Output Divider Configuration	PLL Clock
4MHz	1	4MHz	120	480MHz	2	240MHz
8MHz	2	4MHz	120	480MHz	2	240MHz
8MHz	1	8MHz	60	480MHz	2	240MHz
16MHz	4	4MHz	120	480MHz	2	240MHz
16MHz	2	8MHz	60	480MHz	2	240MHz
16MHz	1	16MHz	30	480MHz	2	240MHz
20MHZ	4	5MHz	96	480MHz	2	240MHz
20MHz	2	10MHz	48	480MHz	2	240MHz
20MHz	1	20MHz	24	480MHz	2	240MHz

Note:

- *Make the input clock frequency division setting so that the PLL input clock (PLLin) is at 3.6 MHz to 32 MHz. Make the multiplication setting so that the PLL output clock (PLLout) is at 400 MHz to (product specification) MHz.*

Multiplying Configuration of SSCG PLL Clock

Table 3-3 Example of SSCG PLL Clock Configuration

Input Clock	Input Clock Divider Configuration	SSCG PLLin	SSCG PLL Multiplier Configuration	SSCG PLLout	Output Divider Configuration	SSCG PLL Clock
4MHz	1	4MHz	120	480MHz	2	240MHz
8MHz	2	4MHz	120	480MHz	2	240MHz
8MHz	1	8MHz	60	480MHz	2	240MHz
16MHz	4	4MHz	120	480MHz	2	240MHz
16MHz	2	8MHz	60	480MHz	2	240MHz
16MHz	1	16MHz	30	480MHz	2	240MHz
20MHZ	4	5MHz	96	480MHz	2	240MHz
20MHz	2	10MHz	48	480MHz	2	240MHz
20MHz	1	20MHz	24	480MHz	2	240MHz

Note:

- *Make the input clock frequency division setting so that the SSCG PLL input clock (SSCG PLLin) is at 3.6 MHz to 32 MHz. Make the multiplication setting so that the SSCG PLL output clock (SSCG PLLout) is at 400 MHz to (product specification) MHz.*

3.3. Clock Distributor

This section explains clock distributor.

Clock distributor has source clock selector and clock divider for each clock domain.

Source Clock Selector Configuration

After each hard reset, Fast-CR clock is selected in all clock domain.

Source clock can be selected for each clock domain by RUN/PSS profile registers.

Table 3-4 Source Clock Selector Configuration

Clock Domain	Domain Clock	Resources	Initial State	RUN/PSS
CD0	CD0_CLK	CPU, TCFLASH/WFLASH, ATB, DBG, TRC, High Performance Matrix, MCU Config Group1, Memory & Config Group, Common Peripheral Group, Application Specific Group, Ext Bus Group	Fast-CR clock	Programmable
CD1	CD1_CLK	(product specification)	Fast-CR clock	Programmable
CD2	CD2_CLK	(product specification)	Fast-CR clock	Programmable
CD3	CD3_CLK	(product specification)	Fast-CR clock	Programmable
CD4	CD4_CLK	(product specification)	Fast-CR clock	Programmable
CD5	CD5_CLK	(product specification)	Fast-CR clock	Programmable
CD_TRC	TRC_CLK	Debug Group	Fast-CR clock	Programmable
CD_HSSPI	HSSPI_CLK	DDRHSSPI Group	Fast-CR clock	Programmable
CD_MCUC	MCUC_CLK	MCU Config Group0	Fast-CR clock	Programmable
CD_CLKO	CLKO_CLK	Clock output function	Fast-CR clock	Programmable

Clock Divider Configuration

There are 1-4 divider circuits on each clock path.

Divided clock of source clock is distributed as internal operating clock.

Table 3-5 Clock Divider Configuration and Maximum Frequency

Clock Domain	Internal Clock	Resources	Parent Clock	Initial State	RUN/PSS	Maximum Frequency
CD0	CLK_CPU	CPU	Source clock selected by CD0CSL[2:0]	1	Programmable	240MHz
	CLK_SHE	SHE	CLK_CPU	1	No div of CLK_CPU	Same as CLK_CPU
	CLK_FCLK	TCFLASH WFLASH	CLK_CPU	4	Programmable	100MHz
	CLK_ATB	ATB	CLK_CPU	1	Programmable	120MHz
	CLK_DBG	DBG	CLK_ATB	1	Programmable	120MHz
	CLK_HPM	High Performance Matrix, Common Peripheral2	CLK_CPU	1	Programmable	240MHZ
	CLK_HPM2		CLK_HPM	2	2	120MHz
	CLK_DMA	DMA	CLK_HPM	1	Programmable	240MHz
	CLK_MEMC	Memory & Config Group	CLK_HPM	1	Programmable	240MHz
	CLK_EXTBUS	Ext Bus Group	CLK_HPM	1	Programmable	120MHz
	CLK_SYSC1	MCU config Group1	CLK_HPM	1	Programmable	120MHz
	CLK_HAPP0A0	Application Specific0 (HPM)	CLK_HPM	1	Programmable	120MHz
	CLK_HAPP0A1		CLK_HAPP0A0	1	Programmable	60MHz
	CLK_HAPP0B0	Application Specific1 (HPM)	CLK_HPM	1	Programmable	120MHz
	CLK_HAPP0B1		CLK_HAPP0B0	1	Programmable	60MHz
	CLK_LLPBM	Common Peripheral (LLPBM)	CLK_CPU	1	Programmable	240MHz
	CLK_LLPBM2	Common Peripheral0/1/2	CLK_LLPBM2	2	2	120MHz
	CLK_LCP	Common Peripheral0/1	CLK_CPU	1	Programmable	120MHz
	CLK_LCP0	Common Peripheral0	CLK_CPU	1	Programmable	120MHz
	CLK_LCP0A		Source clock selected by LCP0ACSEL	1	Programmable	120MHz
	CLK_LCP1	Common Peripheral1	CLK_CPU	1	Programmable	120MHz
	CLK_LCP1A		Source clock selected by LCP1ACSEL	1	Programmable	120MHz

Clock Domain	Internal Clock	Resources	Parent Clock	Initial State	RUN/PSS	Maximum Frequency
	CLK_LAPP0	Application Specific0	CLK_CPU	1	Programmable	120MHz
	CLK_LAPP0A	(LLPPBM)	Source clock selected by LAPP0ACSEL	1	Programmable	120MHz
	CLK_LAPP1	Application Specific1	CLK_CPU	1	Programmable	120MHz
	CLK_LAPP1A	(LLPPBM)	Source clock selected by LAPPAACSEL	1	Programmable	120MHz
CDx (x=1/2/3/4/5)	CLK_CDx	(product specification)	Source clock selected by CDxCSL	1	Programmable	240MHz
	CLK_CDxA0		CLK_CDx	1	Programmable	120MHz
	CLK_CDxA1		CLK_CDxA0	1	Programmable	60MHz
	CLK_CDxB0		CLK_CDx	1	Programmable	120MHz
	CLK_CDxB1		CLK_CDxB0	1	Programmable	60MHz
CD_TRC	CLK_TRC	Debug Group	Source clock selected by TRCCSL	1	Programmable	120MHz
CD_HSSPI	CLK_HSSPI	HSSPI	Source clock selected by HSSPICSL	1	Programmable	200MHz
CD_MCUC	CLK_SYSC0H	MCU Config Group (AHB)	Source clock selected by CDMCUCCSL	1	Programmable	120MHz
	CLK_COMH	Communication Peripheral (AHB)	Source clock selected by CDMCUCCSL	1	Programmable	120MHz
	CLK_RAM0H	Back-up RAM0	Source clock selected by CDMCUCCSL	1	Programmable	120MHz
	CLK_RAM1H	Back-up RAM1	Source clock selected by CDMCUCCSL	1	Programmable	120MHz
	CLK_SYSC0P	MCU Config Group0 (APB)	CLK_SYSC0H	1	Programmable	60MHz
	CLK_COMP	Communication Peripheral (APB)	CLK_SYSC0H	1	Programmable	60MHz
CD_CLKO	CLK_CLKO	Clock output function	Source clock selected by CKSEL	1	Programmable	240MHz

Note:

- *Make the frequency division setting so that no clocks are supplied that exceed the maximum operating frequency of any internal operating clock.*

3.4. Clock Gear

This section explains clock gear.

- Due to drastic frequency difference, switching from/to the main clock to/from PLL/SSCG clock causes huge fluctuation of power supply current. By utilizing the clock gear circuit equipped in the clock switch section, you can gradually shift the operating frequency from/to high-frequency to/from low-frequency, reducing sudden surge of power supply current.

Clock Gear Control

Gear clock is output from clock gear circuit. The frequency of gear clock is gradually changed by the step-by-step outputting of the input clock.

Gear clock is controlled the following configuration.

- Start step configuration
  - Setting to select the range ("START STEP" to STEP 63) during Gear operation that defined by Start Step configuration (xxxCGSSN).
  - It is selectable from STEP 0 to 63.
  - At Gear Up, the Gear operation starts from the selected STEP 'n' and stops at STEP 63. (Refer to Step Width Configuration for the increment width of 'n' after 1 step operation.)
  - At Gear Down, the Gear operation starts from STEP 63 and stops at STEP 'n' the selected. (Refer to Step Width Configuration for the decrement width of 'n' after 1 step operation.)
- Step loop configuration
  - STEP:
    - Loop that is repeated a defined number of times when the Loop configuration(xxxCGLP).
    - It is configurable from 1 to 256 loops.
    - 1 step is xxxCGLP <loop> = xxxCGLP x 64 <cycle>.
  - LOOP:
    - The minimum operation unit that configurates gear operation.
    - 1 Loop fixed at 64 <cycle>.
- Step width configuration
  - The width defined by Step Width configration (xxxCGSTP) which "n" of STEP "n" is increment / decrement (when Gear Up / Gear Down) after 1 STEP execution.
  - Step width can be set to 1 to 4.
  - At Gear Up, the width to increment 'n' after 1 step of STEP 'n'. (In the case of Width = 2, STEP 0, STEP 2, STEP 4 ...).
  - At Gear Down, the width to decrement 'n' after 1 step of STEP 'n'. (In the case of Width 4, STEP 63, STEP 59, STEP 55 ...)

Note::

STEP 0 to 63: Clock output patterns of Gear operation.
As the number of step increases, the clock output increases.
STEP 0 : 1 clock/loop
STEP 1 : 2 clock/loop
:
:
:
STEP 63 : 64 clock/loop (Clock output for all cycles in loop)

 # Refer to Figure 3-1 for details pattern.

The length of STEP 0 to 63 (the number of Loop repetitions) is determined by the setting of STEP (xxxCGLP x 64 <cycle>).

Figure 3-1 Gear Clock Output

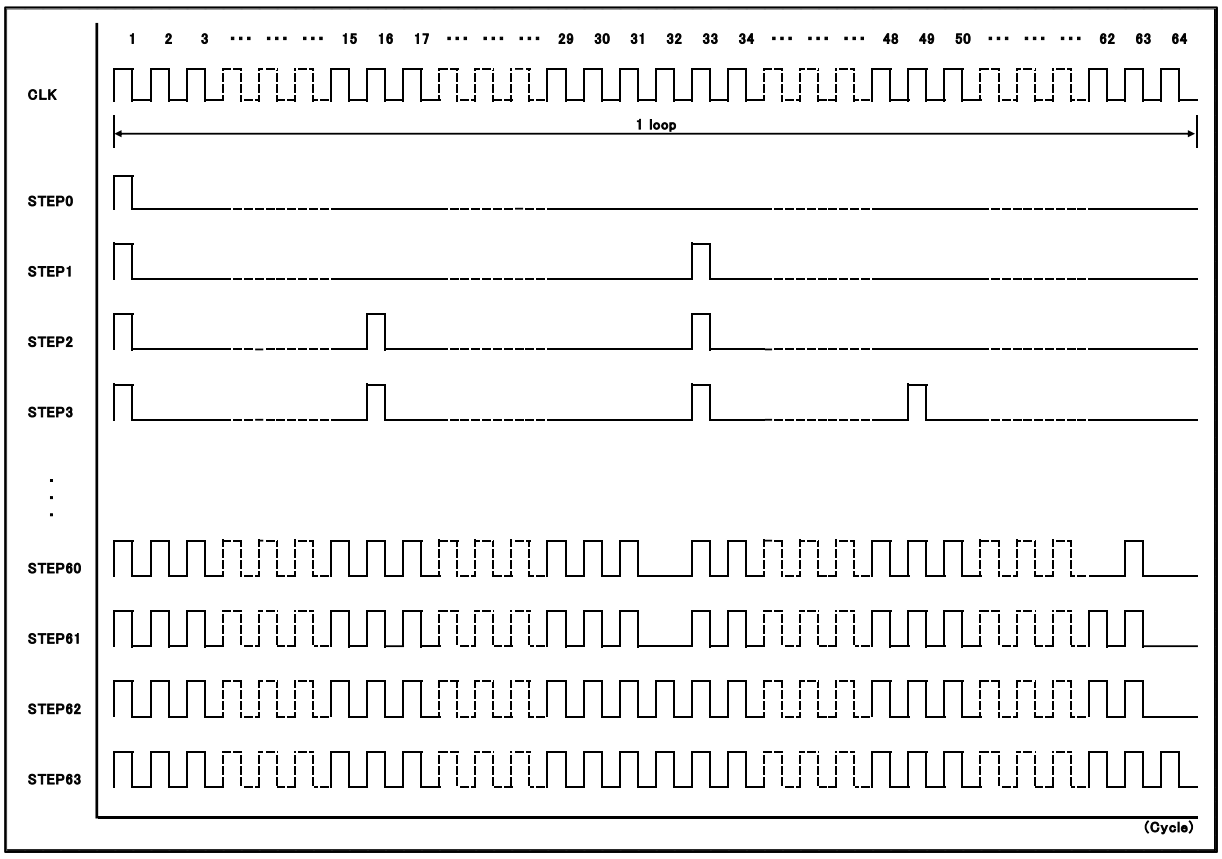
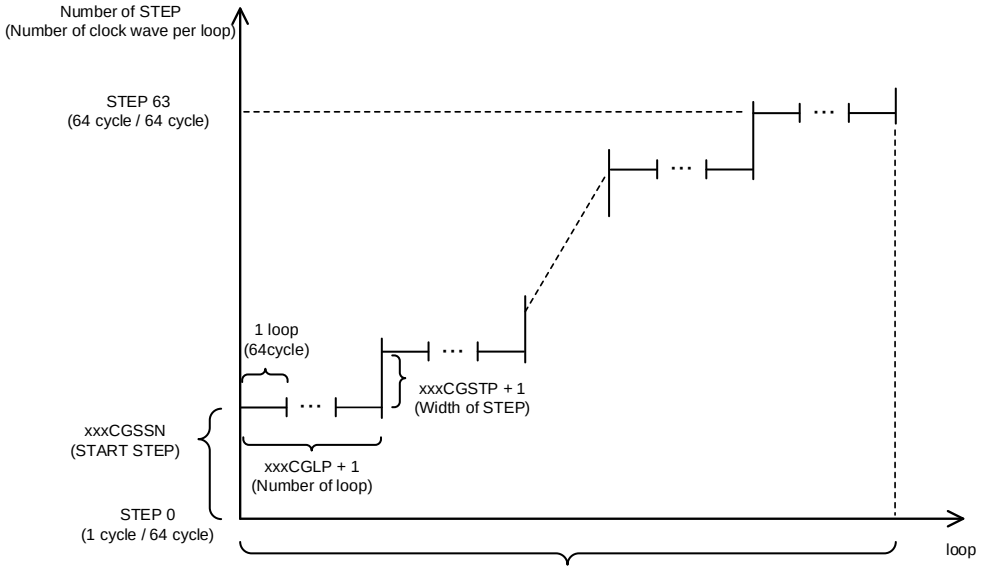


Figure.3-2(Gear Up) and Figure.3-3(Gear Down) show the relationship between the register configuration and the Clock Gear STEP.

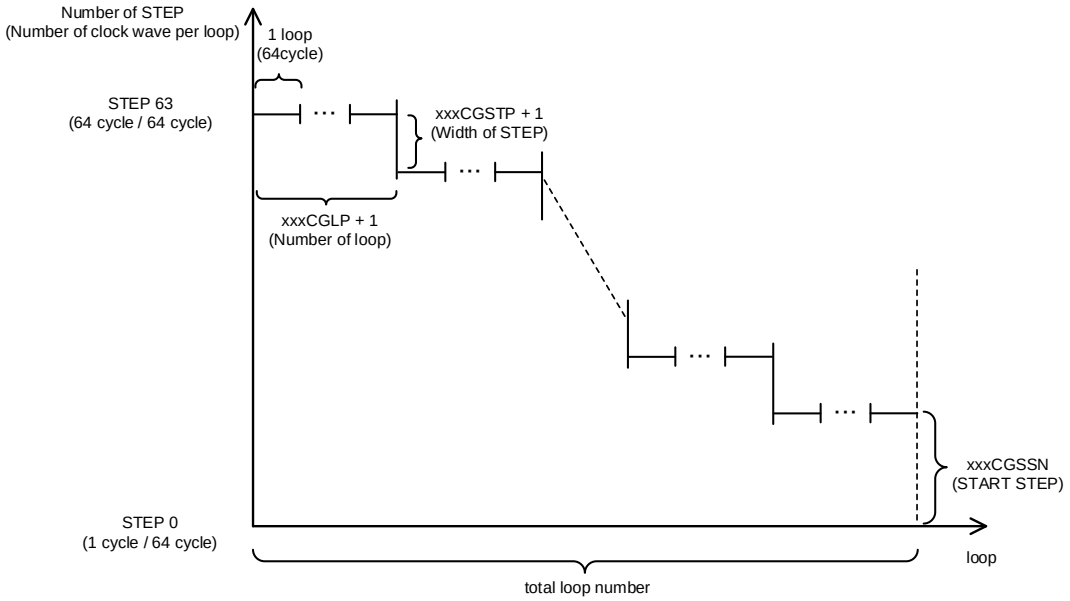
Figure 3-2 Clock Gear STEP by register configuration (Gear Up)



Transition time of clock gear = total loop number * 64cycle * (PLL or SSCG period)
= (xxxCGLP + 1) * (64 − xxxCGSSN) / (xxxCGSTP + 1) * 64cycle * (PLL or SSCG period)

Note: The number is rounded up after the decimal point in "(64 − xxxCGSSN) / (xxxCGSTP + 1)".

Figure 3-3 Clock Gear STEP by register configuration (Gear Down)



Transition time of clock gear = Same as gear up (Figure.3-2)

3.4.1. Gear Up Procedure

This section explains gear up procedure.

1. After completion of oscillation stabilization standby timer, the initial step clock, selected to clock gear start step, is output.
2. When GEAR_START is set to "1", and its rising is detected, GEAR_STATUS shifts from "00" → "01". (gear-up start.)
3. Following the clock gear step selection and repeat time selection, gear-up is initiated. Give smaller steps and more repetition time for smoother change.
4. When clock reaches the maximum step, GEAR_STATUS shifts from "01" to "10". (Gear-up completed, gear stopped)
5. After stopping of the gear, GEAR_START is cleared to "0" by hardware.

Note:

- *Poll the clock gear status flag during clock gear UP/DOWN operation and wait for the clock Lower/Upper stop status.*

3.4.2. Gear Down Procedure

This section explains gear down procedure.

1. When GEAR_START is set to "1", and its rising is detected, GEAR_STATUS shifts from "10" → "11". (gear-down start.)
2. Following the clock gear step selection and repeat time selection, gear-down is initiated. Give smaller steps and more repetition time for smoother change.
3. When clock reaches the minimum step, GEAR_STATUS shifts from "11" to "00". (Gear-down completed, gear stopped)
4. After stopping of the gear, GEAR_START is cleared to "0" by hardware.

Note:

- *Poll the clock gear status flag during clock gear UP/DOWN operation and wait for the clock Lower/Upper stop status.*

3.5. Clock Stabilization Time

This section explains clock stabilization time.

Source clock needs to wait for stabilization when source clock is not ready. Source clock does not provide for other resources during waiting clock stabilization time. After clock stabilization time, source clock can be used for each clock domain.

For more details refer to 'chapter:source clock timer'.

3.6. Interrupt

This section explains interrupt factor of clock system.

Interrupt Factor

The clock system has the following interrupt factor.

- Clock supervisor interrupt
  For more details refer to 'chapter:clock supervisor'.
- Profile Error interrupt
  For more details refer to 'chapter: Low-Power Consumption'.

4. Setting Procedure Example

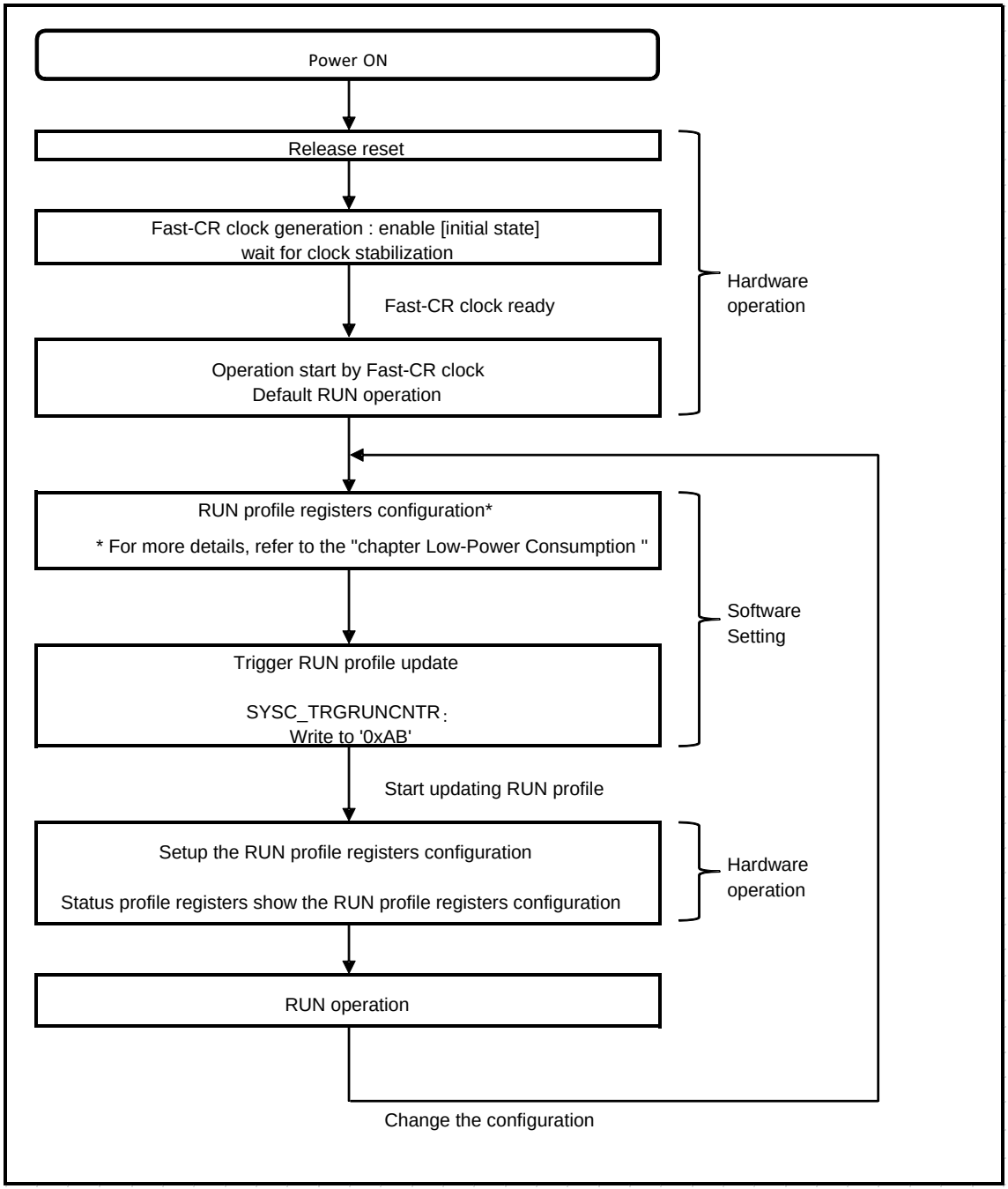
This section explains a setting procedure example of the clock system.

4.1. RUN Configuration

This section explains RUN configuration of clock system.

Setting Procedure Example for RUN Configuration

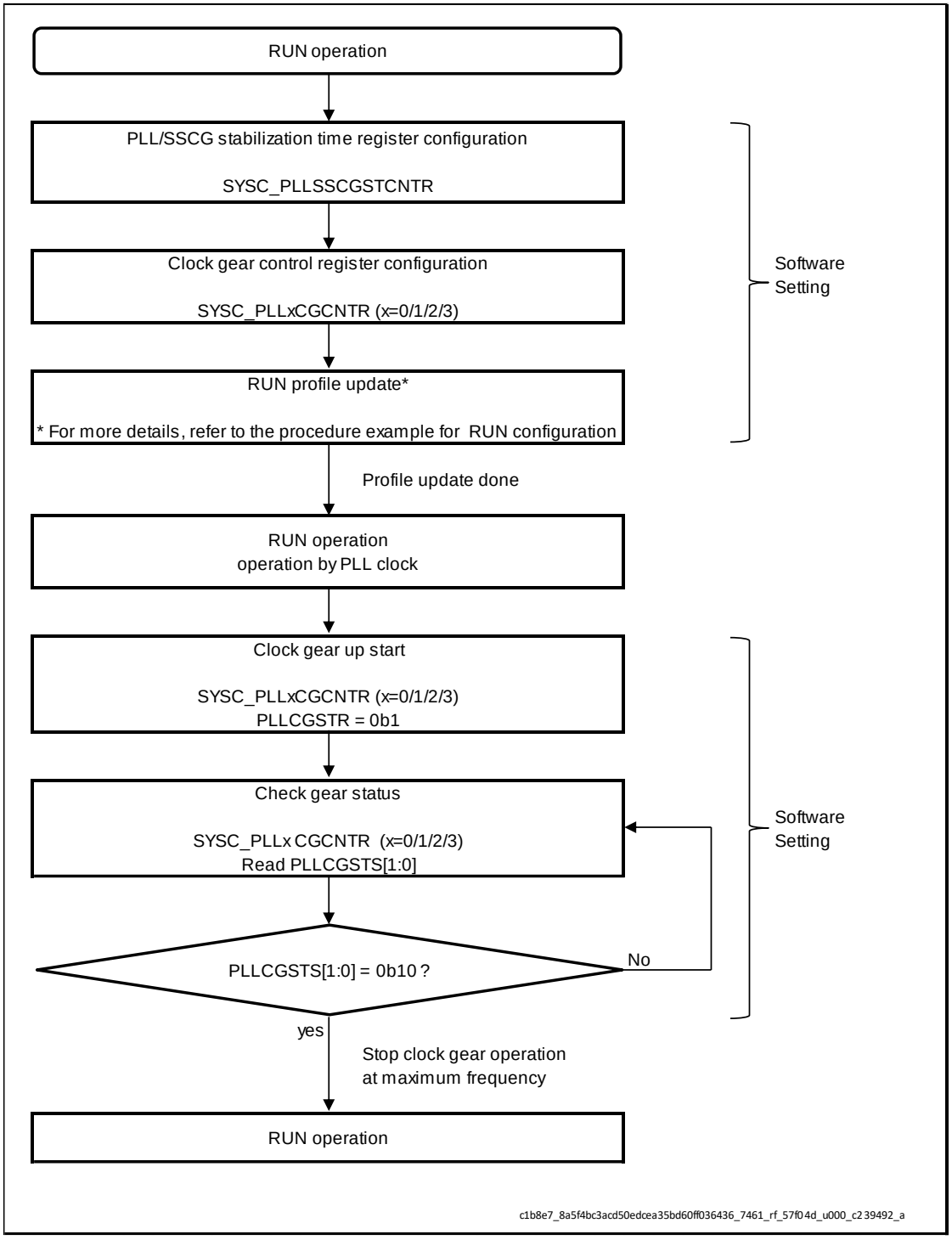
Figure 4-1 Setting Procedure Example for RUN Configuration



Note:

- *To use the main clock and the PLL/SSCG PLL clock, the clock supervisor must also be set. For details, see the following chapter: "CLOCK SUPERVISOR."*

Figure 4-2 Setting Procedure Example for Using PLL Clock



Note:

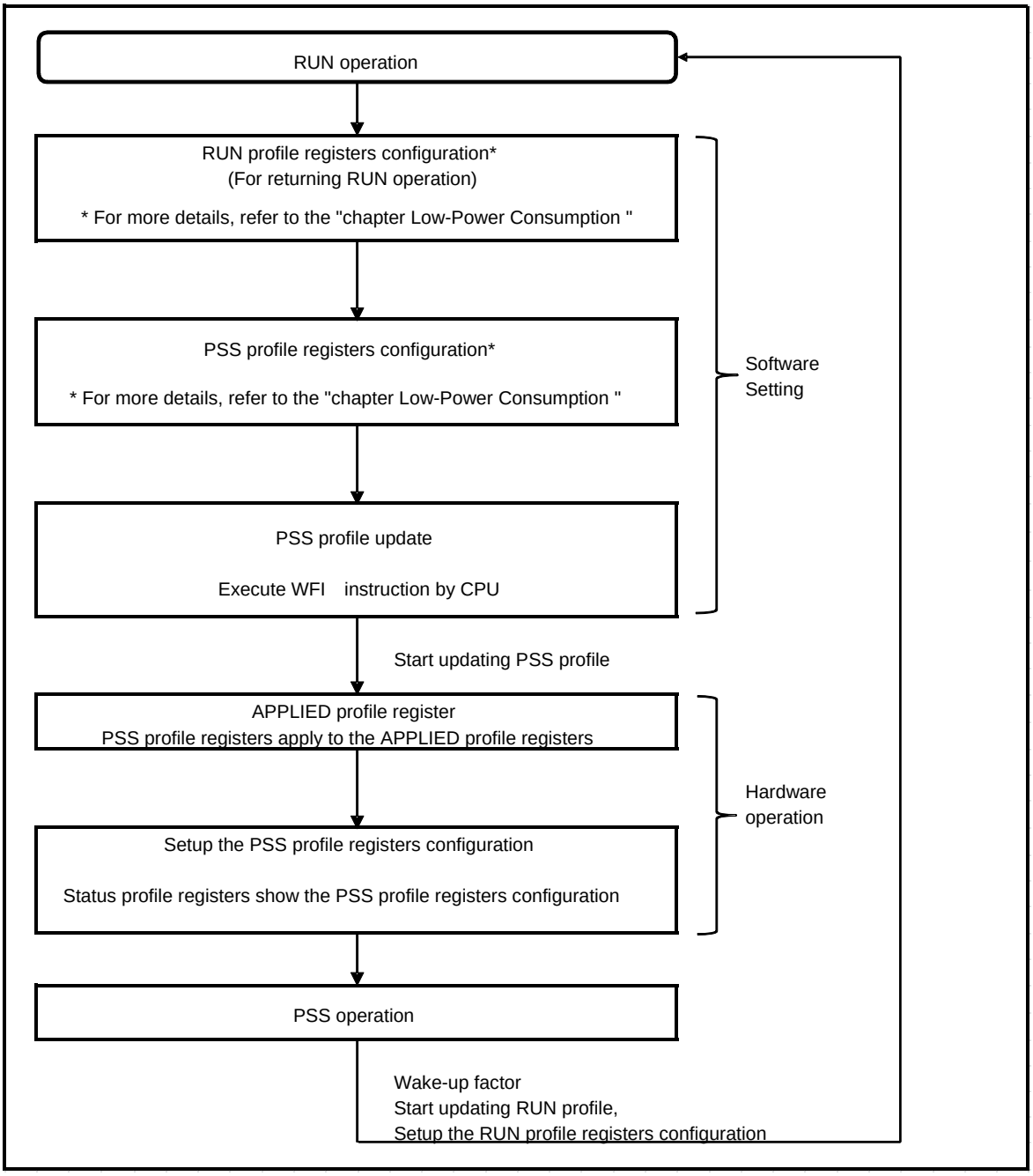
- *To use the pll clock or sscg pll clock, clock gear must be used. Make sure to set the start gear step does not exceed the maximum frequency of every internal clock in the clock domain.*

4.2. PSS Configuration

This section explains PSS configuration of clock system.

Setting Procedure Example for PSS Configuration

Figure 4-3 Setting Procedure Example for PSS Configuration



Note:

- *To use the main clock and the (product specification) clock, the clock supervisor must also be set. For details, see the following chapter: "CLOCK SUPERVISOR."*

5. Registers

This section explains the registers of clock system.

There are 6 register groups in clock system.

- RUN profile registers
- PSS profile registers
- APPLIED profile registers
- Status profile registers
- Common Configuration registers
- Clock Output function register

For more details about Run profile registers and PSS profile registers, APPLIED profile registers, Status profile registers refer to the "chapter: Low-Power Consumption".

Registers of Clock System (Common Configuration Registers)

Table 5-1 Registers of Clock System (Common Configuration Registers)

Abbreviated expressions	Register name	Reference
SYSC_CRCNTR	CR clock control register	5.1.1
SYSC_MOSCCNTR	Main oscillator control register	5.1.2
SYSC_SOSCCNTR	Sub oscillator control register	5.1.3
SYSC_PLLSSCGSTCNTR	PLL/SSCG stabilization time control register	5.1.4
SYSC_PLL0CGCNTR	PLL0 clock gear control register	5.1.5
SYSC_PLL1CGCNTR	PLL1 clock gear control register	5.1.6
SYSC_PLL2CGCNTR	PLL2 clock gear control register	5.1.7
SYSC_PLL3CGCNTR	PLL3 clock gear control register	5.1.8
SYSC_SSCG0CGCNTR	SSCG PLL0 clock gear control register	5.1.9
SYSC_SSCG1CGCNTR	SSCG PLL1 clock gear control register	5.1.10
SYSC_SSCG2CGCNTR	SSCG PLL2 clock gear control register	5.1.11
SYSC_SSCG3CGCNTR	SSCG PLL3 clock gear control register	5.1.12

Registers of Clock System (Clock Output Function Register)

Table 5-2 Registers of Clock System (Common Configuration Registers)

Abbreviated expressions	Register name	Reference
SYSC_CKOTCNTR	Clock output function control register	5.2.1

Table 5-3 Register Map for Clock System

offset	Register / initial value
0x0000_0000	SYSC_CRCNTR 00000000_000*****_000*****_000*****
0x0000_0004	SYSC_MOSCCNTR *000**00_00000000_00000001_00000000
0x0000_0008	SYSC_SOSCCNTR 00000000_0000000*_00000000_00000000
0x0000_000C	SYSC_PLLSSCGSTCNTR 00000000_00000000_00000000_11111111
0x0000_0010	SYSC_PLL0CGCNTR 00000000_********_********_0000000*
0x0000_0014	SYSC_PLL1CGCNTR 00000000_********_********_0000000*
0x0000_0018	SYSC_PLL2CGCNTR 00000000_********_********_0000000*
0x0000_001C	SYSC_PLL3CGCNTR 00000000_********_********_0000000*
0x0000_0020	SYSC_SSCG0CGCNTR 00000000_********_********_0000000*
0x0000_0024	SYSC_SSCG1CGCNTR 00000000_********_********_0000000*
0x0000_0028	SYSC_SSCG2CGCNTR 00000000_********_********_0000000*
0x0000_002C	SYSC_SSCG3CGCNTR 00000000_********_********_0000000*
0x0000_0030	SYSC_CKOTCNTR 00000000_00000000_00000000_00000000
0x0000_0034 - 0x0000_007C	-

Notes:

- **: product specification*
- *The registers are protected by protection key setting register (SYSC0_PROTKEYR). For details on the protection, see the following chapter: "Low-power Consumption."*
- *Access to reserved address results bus error response.*

5.1. Common Configuration Registers

5.1.1. CR Clock Control Register(SYSC_CRCNTR)

CR clock control register(SYSC_CRCNTR) controls CR clock.

bit	bit 31	bit 16
Field	Reserved	
Data Attribute	R0,WX	
Prot_Attr	WPS	
Initial Value	00000000 00000000	

bit	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
Field	Reserved			TRC[4:0]				
Data Attribute	R0,WX			R/W				
Prot_Attr	WPS							
Initial Value	000			(product specification)				

bit	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Field	Reserved			TRF[4:0]				
Data Attribute	R0,WX			R/W				
Prot_Attr	WPS							
Initial Value	000			(product specification)				

[bit31:16] Reserved

[bit12:8] TRC[4:0] : Coarse Trimming Bit

These bits are used to trim CR clock frequency

For more details on trim value refer to "chapter:CR calibration".

[bit4:0] TRF[4:0] : Fine Trimming Bit

These bits are used to trim CR clock frequency

For more details on trim value refer to "chapter:CR calibration".

5.1.2. Main Oscillator Control Register(SYSC_MOSCCNTR)

Main oscillator control register(SYSC_MOSCCNTR) is used to control main clock.

bit	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24
Field	MCMODE	Reserved			MCGAIN[1:0]		Reserved	Reserved
Data Attribute	R/W	R0,WX			R/W		R0,WX	R/W0
Prot_Attr	WPS							
Initial Value	(product specification)	000			(product specification)		0	0

bit	bit 23	bit 22	bit 21	bit 20	bit 19	bit 18	bit 17	bit 16
Field	Reserved							
Data Attribute	R0,WX							
Prot_Attr	WPS							
Initial Value	00000000							

bit	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
Field	Reserved							DIV2SEL
Data Attribute	R0,WX							R/W
Prot_Attr	WPS							
Initial Value	0000000							1

bit	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Field	Reserved							FCIMEN
Data Attribute	R0,WX							R0,W0
Prot_Attr	WPS							
Initial Value	0000000							0

[bit31] MCMODE : Main Clock Amplifier Oscillation Mode Bit

This bit is used to set Main clock amplifier oscillation mode.

Bit	Description
0	Main clock amplifier oscillation mode
1	Main clock amplifier oscillation stop mode

Note:

The initial value (value "0") of this bit specifies High drive mode.
To reduce noise, set the mode to low drive mode (value "1") after transition to the user program.

[bit30:28] Reserved

[bit27:26] MCGAIN[1:0] : Main Clock GAIN Bit

The setting of these bits can control gain characteristic of I/O cell.

When Main clock goes through I/O cell, its amplitude is decayed with respect to the gain characteristic.

bit	Description
00	4(3.6)MHz
01	8MHz
10	16MHz
11	25MHz

Note:

These frequency values are just reference value.
Set optimum value according to oscillator matching evaluation.
Generally, MCMODE should be "1" (Stop mode) to optimize by MCGAIN setting.

[bit25:24] Reserved

[bit8] DIV2SEL : Select Bit

This bit is used to divide main clock into two.

bit	Description
0	Main clock
1	Main clock divided by two

[bit7:1] Reserved

[bit0] FCIMEN : Fast Main Clock Input Enable Control Bit

Always set to 0.

5.1.3. Sub Oscillator Control Register(SYSC_SOSCCNTR)

Sub oscillator control register(SYSC_SOSCCNTR) is used to control sub clock.

bit	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24
Field	Reserved							Reserved
Data Attribute	R0,WX							R/W0
Prot_Attr	WPS							
Initial Value	0000000							0

bit	bit 23							bit 16
Field	Reserved							SUBPORT
Data Attribute	R0,WX							R/W
Prot_Attr	WPS							
Initial Value	0000000							0

bit	bit 15							bit 0
Field	Reserved							
Data Attribute	R0,WX							
Prot_Attr	WPS							
Initial Value	00000000 00000000							

[bit31:17] Reserved

[bit16] SUBPORT

This bit is used to set Port function for sub clock input

bit	Description
0	Sub clock input disable (port)
1	Sub clock input enable

[bit15:0] Reserved

5.1.4. PLL/SSCG Stabilization Time Control Register (SYSC_PLLSSCGSTCNTR)

PLL/SSCG stabilization time control register(SYSC_PLLSSCGSTCNTR) is used to control clock stabilization time.

bit	bit 31							bit 8
Field	Reserved							
Data Attribute	R0,WX							
Prot_Attr	WPS							
Initial Value	00000000 00000000 00000000							
bit	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Field	SSCGSTABS[3:0]				PLLSTABS[3:0]			
Data Attribute	R1,WX	R/W			R1,WX	R/W		
Prot_Attr	WPS							
Initial Value	1111				1111			

[bit31:8] Reserved

[bit7:4] SSCGSTABS[3:0] :

These bits select the stabilization time for SSCG PLL0/1/2/3 clock.

bit7:4	Description
1000	Stabilization time : (clock period of selected clock from Main clock or Fast CR clock) [s] * 2^9 [cycle]
1001	Stabilization time : (clock period of selected clock from Main clock or Fast CR clock) [s] * 2^10 [cycle]
1010	Stabilization time : (clock period of selected clock from Main clock or Fast CR clock) [s] * 2^11 [cycle]
1011	Stabilization time : (clock period of selected clock from Main clock or Fast CR clock) [s] * 2^12 [cycle]
1100	Stabilization time : (clock period of selected clock from Main clock or Fast CR clock) [s] * 2^13 [cycle]
1101	Stabilization time : (clock period of selected clock from Main clock or Fast CR clock) [s] * 2^14 [cycle]
1110	Stabilization time : (clock period of selected clock from Main clock or Fast CR clock) [s] * 2^15 [cycle]
1111	Stabilization time : (clock period of selected clock from Main clock or Fast CR clock) [s] * 2^16 [cycle]

Reading SSCGSTABS[3] bit always returns '1'

Notes:

- *These bits have to be set before SSCG PLL clock enable setting.*
- *Changing these bits are prohibited after SSCG PLL clock enable setting.*

[bit3:0] PLLSTABS[3:0] :

These bits select the stabilization time for PLL0/1/2/3 clock.

bit3:0	Description
1000	Stabilization time : (clock period of selected clock from Main clock or Fast CR clock) [s] * 2^9 [cycle]
1001	Stabilization time : (clock period of selected clock from Main clock or Fast CR clock) [s] * 2^10 [cycle]
1010	Stabilization time : (clock period of selected clock from Main clock or Fast CR clock) [s] * 2^11 [cycle]
1011	Stabilization time : (clock period of selected clock from Main clock or Fast CR clock) [s] * 2^12 [cycle]
1100	Stabilization time : (clock period of selected clock from Main clock or Fast CR clock) [s] * 2^13 [cycle]
1101	Stabilization time : (clock period of selected clock from Main clock or Fast CR clock) [s] * 2^14 [cycle]
1110	Stabilization time : (clock period of selected clock from Main clock or Fast CR clock) [s] * 2^15 [cycle]
1111	Stabilization time : (clock period of selected clock from Main clock or Fast CR clock) [s] * 2^16 [cycle]

Reading PLLSTABS[3] bit always returns '1'

Notes:

- *These bits have to be set before PLL clock enable setting.*
- *Changing these bits are prohibited after PLL clock enable setting.*

5.1.5. PLL0 Clock Gear Control Register (SYSC_PLL0CGCNTR)

PLL0 clock gear control register(SYSC_PLL0CGCNTR) controls PLL0 clock gear.

bit	bit 31							bit 24
Field	Reserved							
Data Attribute	R0,WX							
Prot_Attr	WPS							
Initial Value	00000000							

bit	bit 23							bit 16
Field	PLLCGLP[7:0]							
Data Attribute	R/W							
Prot_Attr	WPS							
Initial Value	(product specification)							

bit	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
Field	PLLCGSTP[1:0]		PLLCGSSN[5:0]					
Data Attribute	R/W		R/W					
Prot_Attr	WPS							
Initial Value	(product specification)		(product specification)					

bit	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Field	PLLCGSTS[1:0]		Reserved				PLLCG STR	PLLCG EN
Data Attribute	R,WX		R0,WX				R,W	R/W
Prot_Attr	WPS							
Initial Value	00		0000				0	(product specification)

[bit31:24] Reserved

[bit23:16] PLLCGLP[7:0] : PLL Clock Gear Loop Configuration Bit

These bits select the number of loop per step.

bit23:16	Description
00000000	1 loop
00000001	2 loops
00000010	3 loops
…	
11111101	254 loops
11111110	255 loops
11111111	256 loops

Notes:

- *These bits have to be set before PLL clock enable setting.*

- *Changing these bits are prohibited after PLL clock enable setting.*

[bit15:14] PLLCGSTP[1:0] : PLL Clock Gear Step Configuration Bit

These bits select the number of step width at gear up/down operation.

bit15:14	Description
00	1 step
01	2 steps
10	3 steps
11	4 steps

Notes:

- *These bits have to be set before PLL clock enable setting.*
- *Changing these bits are prohibited after PLL clock enable setting.*

[bit13:8] PLLCGSSN[5:0] : PLL Clock Gear Start Step Configuration Bit

These bits set the start step.

bit13:8	Description
000000	0
000001	1
000010	2
…	
111101	61
111110	62
111111	63

Notes:

- *These bits have to be set before PLL clock enable setting.*
- *Changing these bits are prohibited after PLL clock enable setting.*

[bit7:6] PLLCGSTS[1:0] : PLL Clock Gear Status Bit

These bits show the PLL clock gear status.

Bit7:6	Description
00	Clock gear operation reaches minimum frequency
01	Gear up operation
10	Clock gear operation reaches maximum frequency
11	Gear down operation

Notes:

- *In the PLL clock stop or PLL clock stabilization wait, return a "00" status.*
- *if clock gear not used, the status is "X" (undefined).*

[bit5:2] Reserved

[bit1] PLLCGSTR : PLL Clock Gear Start Bit

This bit is used to start clock gear operation.

bit	Description
0	No operation
1	Start clock gear operation

Note:

- *This bit is cleared after completion of clock gear operation.*

[bit0] PLLCGEN : PLL Clock Gear Enable Bit

This bit controls enable/disable clock gear operation.

bit	Description
0	PLL clock gear disable
1	PLL clock gear enable

Notes:

- *These bits have to be set before PLL clock enable setting.*
- *Changing these bits are prohibited after PLL clock enable setting.*

5.1.6. PLL1 Clock Gear Control Register (SYSC_PLL1CGCNTR)

PLL1 clock gear control register(SYSC_PLL1CGCNTR) controls PLL1 clock gear.

bit	bit 31							bit 24
Field	Reserved							
Data Attribute	R0,WX							
Prot_Attr	WPS							
Initial Value	00000000							

bit	bit 23							bit 16
Field	PLLCGLP[7:0]							
Data Attribute	R/W							
Prot_Attr	WPS							
Initial Value	(product specification)							

bit	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
Field	PLLCGSTP[1:0]		PLLCGSSN[5:0]					
Data Attribute	R/W		R/W					
Prot_Attr	WPS							
Initial Value	(product specification)		(product specification)					

bit	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Field	PLLCGSTS[1:0]		Reserved				PLLCG STR	PLLCG EN
Data Attribute	R,WX		R0,WX				R,W	R/W
Prot_Attr	WPS							
Initial Value	00		0000				0	(product specification)

[bit31:24] Reserved

[bit23:16] PLLCGLP[7:0] : PLL Clock Gear Loop Configuration Bit

These bits select the number of loop per step.

bit23:16	Description
00000000	1 loop
00000001	2 loops
00000010	3 loops
…	
11111101	254 loops
11111110	255 loops
11111111	256 loops

Notes:

- *These bits have to be set before PLL clock enable setting.*
- *Changing these bits are prohibited after PLL clock enable setting.*

[bit15:14] PLLCGSTP[1:0] : PLL Clock Gear Step Configuration Bit

These bits select the number of step width at gear up/down operation.

bit15:14	Description
00	1 step
01	2 steps
10	3 steps
11	4 steps

Notes:

- *These bits have to be set before PLL clock enable setting.*
- *Changing these bits are prohibited after PLL clock enable setting.*

[bit13:8] PLLCGSSN[5:0] : PLL Clock Gear Start Step Configuration Bit

These bits set the start step.

bit13:8	Description
000000	0
000001	1
000010	2
…	
111101	61
111110	62
111111	63

Notes:

- *These bits have to be set before PLL clock enable setting.*
- *Changing these bits are prohibited after PLL clock enable setting.*

[bit7:6] PLLCGSTS[1:0] : PLL Clock Gear Status Bit

These bits show the PLL clock gear status.

Bit7:6	Description
00	Clock gear operation reaches minimum frequency
01	Gear up operation
10	Clock gear operation reaches maximum frequency
11	Gear down operation

Notes:

- *In the PLL clock stop or PLL clock stabilization wait, return a "00" status.*
- *if clock gear not used, the status is "X" (undefined).*

[bit5:2] Reserved

[bit1] PLLCGSTR : PLL Clock Gear Start Bit

This bit is used to start clock gear operation.

bit	Description
0	No operation
1	Start clock gear operation

Note:

- *This bit is cleared after completion of clock gear operation.*

[bit0] PLLCGEN : PLL Clock Gear Enable Bit

This bit controls enable/disable clock gear operation.

bit	Description
0	PLL clock gear disable
1	PLL clock gear enable

Notes:

- *These bits have to be set before PLL clock enable setting.*
- *Changing these bits are prohibited after PLL clock enable setting.*

5.1.7. PLL2 Clock Gear Control Register (SYSC_PLL2CGCNTR)

PLL2 clock gear control register(SYSC_PLL2CGCNTR) controls PLL2 clock gear.

bit	bit 31							bit 24
Field	Reserved							
Data Attribute	R0,WX							
Prot_Attr	WPS							
Initial Value	00000000							

bit	bit 23							bit 16
Field	PLLCGLP[7:0]							
Data Attribute	R/W							
Prot_Attr	WPS							
Initial Value	(product specification)							

bit	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
Field	PLLCGSTP[1:0]		PLLCGSSN[5:0]					
Data Attribute	R/W		R/W					
Prot_Attr	WPS							
Initial Value	(product specification)		(product specification)					

bit	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Field	PLLCGSTS[1:0]		Reserved				PLLCG STR	PLLCG EN
Data Attribute	R,WX		R0,WX				R,W	R/W
Prot_Attr	WPS							
Initial Value	00		0000				0	(product specification)

[bit31:24] Reserved

[bit23:16] PLLCGLP[7:0] : PLL Clock Gear Loop Configuration Bit

These bits select the number of loop per step.

bit23:16	Description
00000000	1 loop
00000001	2 loops
00000010	3 loops
…	
11111101	254 loops
11111110	255 loops
11111111	256 loops

Notes:

- *These bits have to be set before PLL clock enable setting.*
- *Changing these bits are prohibited after PLL clock enable setting.*

[bit15:14] PLLCGSTP[1:0] : PLL Clock Gear Step Configuration Bit

These bits select the number of step width at gear up/down operation.

bit15:14	Description
00	1 step
01	2 steps
10	3 steps
11	4 steps

Notes:

- *These bits have to be set before PLL clock enable setting.*
- *Changing these bits are prohibited after PLL clock enable setting.*

[bit13:8] PLLCGSSN[5:0] : PLL Clock Gear Start Step Configuration Bit

These bits set the start step.

bit13:8	Description
000000	0
000001	1
000010	2
…	
111101	61
111110	62
111111	63

Notes:

- *These bits have to be set before PLL clock enable setting.*
- *Changing these bits are prohibited after PLL clock enable setting.*

[bit7:6] PLLCGSTS[1:0] : PLL Clock Gear Status Bit

These bits show the PLL clock gear status.

Bit7:6	Description
00	Clock gear operation reaches minimum frequency
01	Gear up operation
10	Clock gear operation reaches maximum frequency
11	Gear down operation

Notes:

- *In the PLL clock stop or PLL clock stabilization wait, return a "00" status.*
- *if clock gear not used, the status is "X" (undefined).*

[bit5:2] Reserved

[bit1] PLLCGSTR : PLL Clock Gear Start Bit

This bit is used to start clock gear operation.

bit	Description
0	No operation
1	Start clock gear operation

Note:

- *This bit is cleared after completion of clock gear operation.*

[bit0] PLLCGEN : PLL Clock Gear Enable Bit

This bit controls enable/disable clock gear operation.

bit	Description
0	PLL clock gear disable
1	PLL clock gear enable

Notes:

- *These bits have to be set before PLL clock enable setting.*
- *Changing these bits are prohibited after PLL clock enable setting.*

5.1.8. PLL3 Clock Gear Control Register (SYSC_PLL3CGCNTR)

PLL3 clock gear control register(SYSC_PLL3CGCNTR) controls PLL3 clock gear.

bit	bit 31	bit 24
Field	Reserved	
Data Attribute	R0,WX	
Prot_Attr	WPS	
Initial Value	00000000	

bit	bit 23	bit 16
Field	PLLCGLP[7:0]	
Data Attribute	R/W	
Prot_Attr	WPS	
Initial Value	(product specification)	

bit	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
Field	PLLCGSTP[1:0]		PLLCGSSN[5:0]					
Data Attribute	R/W		R/W					
Prot_Attr	WPS							
Initial Value	(product specification)		(product specification)					

bit	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Field	PLLCGSTS[1:0]		Reserved				PLLCG STR	PLLCG EN
Data Attribute	R,WX		R0,WX				R,W	R/W
Prot_Attr	WPS							
Initial Value	00		0000				0	(product specification)

[bit31:24] Reserved

[bit23:16] PLLCGLP[7:0] : PLL Clock Gear Loop Configuration Bit

These bits select the number of loop per step.

bit23:16	Description
00000000	1 loop
00000001	2 loops
00000010	3 loops
…	
11111101	254 loops
11111110	255 loops
11111111	256 loops

Notes:

- *These bits have to be set before PLL clock enable setting.*
- *Changing these bits are prohibited after PLL clock enable setting.*

[bit15:14] PLLCGSTP[1:0] : PLL Clock Gear Step Configuration Bit

These bits select the number of step width at gear up/down operation.

bit15:14	Description
00	1 step
01	2 steps
10	3 steps
11	4 steps

Notes:

- *These bits have to be set before PLL clock enable setting.*
- *Changing these bits are prohibited after PLL clock enable setting.*

[bit13:8] PLLCGSSN[5:0] : PLL Clock Gear Start Step Configuration Bit

These bits set the start step.

bit13:8	Description
000000	0
000001	1
000010	2
…	
111101	61
111110	62
111111	63

Notes:

- *These bits have to be set before PLL clock enable setting.*
- *Changing these bits are prohibited after PLL clock enable setting.*

[bit7:6] PLLCGSTS[1:0] : PLL Clock Gear Status Bit

These bits show the PLL clock gear status.

Bit7:6	Description
00	Clock gear operation reaches minimum frequency
01	Gear up operation
10	Clock gear operation reaches maximum frequency
11	Gear down operation

Notes:

- *In the PLL clock stop or PLL clock stabilization wait, return a "00" status.*
- *if clock gear not used, the status is "X" (undefined).*

[bit5:2] Reserved

[bit1] PLLCGSTR : PLL Clock Gear Start Bit

This bit is used to start clock gear operation.

bit	Description
0	No operation
1	Start clock gear operation

Note:

- *This bit is cleared after completion of clock gear operation.*

[bit0] PLLCGEN : PLL Clock Gear Enable Bit

This bit controls enable/disable clock gear operation.

bit	Description
0	PLL clock gear disable
1	PLL clock gear enable

Notes:

- *These bits have to be set before PLL clock enable setting.*
- *Changing these bits are prohibited after PLL clock enable setting.*

5.1.9. SSCG PLL0 Clock Gear Control Register (SYSC_SSCG0CGCNTR)

SSCG PLL0 clock gear control register(SYSC_SSCG0CGCNTR) controls SSCG PLL0 clock gear.

bit	bit 31	bit 24
Field	Reserved	
Data Attribute	R0,WX	
Prot_Attr	WPS	
Initial Value	00000000	

bit	bit 23	bit 16
Field	SSCGCGLP[7:0]	
Data Attribute	R/W	
Prot_Attr	WPS	
Initial Value	(product specification)	

bit	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
Field	SSCGCGSTP[1:0]		SSCGCGSSN[5:0]					
Data Attribute	R/W		R/W					
Prot_Attr	WPS							
Initial Value	(product specification)		(product specification)					

bit	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Field	SSCGCGSTS[1:0]		Reserved				SSCGCGSTR	SSCGCGEN
Data Attribute	R,WX		R0,WX				R,W	R/W
Prot_Attr	WPS							
Initial Value	00		0000				0	(product specification)

[bit31:24] Reserved

[bit23:16] SSCGCGLP[7:0] : SSCG PLL Clock Gear Loop Configuration Bit

These bits select the number of loop per step.

bit23:16	Description
00000000	1 loop
00000001	2 loops
00000010	3 loops
…	
11111101	254 loops
11111110	255 loops
11111111	256 loops

Notes:

- *These bits have to be set before SSCG PLL clock enable setting.*

- *Changing these bits are prohibited after SSCG PLL clock enable setting.*

[bit15:14] SSCGCGSTP[1:0] : SSCG PLL Clock Gear Step Configuration Bit

These bits select the number of step width at gear up/down operation.

bit15:14	Description
00	1 step
01	2 steps
10	3 steps
11	4 steps

Notes:

- *These bits have to be set before SSCG PLL clock enable setting.*
- *Changing these bits are prohibited after SSCG PLL clock enable setting.*

[bit13:8] SSCGCGSSN[5:0] : SSCG PLL Clock Gear Start Step Configuration Bit

These bits set the start step.

bit13:8	Description
000000	0
000001	1
000010	2
…	
111101	61
111110	62
111111	63

Notes:

- *These bits have to be set before SSCG PLL clock enable setting.*
- *Changing these bits are prohibited after SSCG PLL clock enable setting.*

[bit7:6] SSCGCGSTS[1:0] : SSCG PLL Clock Gear Status Bit

These bits show the SSCG PLL clock gear status.

Bit7:6	Description
00	Clock gear operation reaches minimum frequency
01	Gear up operation
10	Clock gear operation reaches maximum frequency
11	Gear down operation

Notes:

- *In the SSCG PLL clock stop or SSCG PLL clock stabilization wait, return a "00" status.*
- *if clock gear not used, the status is "X" (undefined).*

[bit5:2] Reserved

[bit1] SSCGCGSTR : SSCG PLL Clock Gear Start Bit

This bit is used to start clock gear operation.

bit	Description
0	No operation
1	Start clock gear operation

Note:

- *This bit is cleared after completion of clock gear operation.*

[bit0] SSCGCGEN : SSCG PLL Clock Gear Enable Bit

This bit controls enable/disable clock gear operation.

bit	Description
0	SSCG PLL clock gear disable
1	SSCG PLL clock gear enable

Notes:

- *These bits have to be set before SSCG PLL clock enable setting.*
- *Changing these bits are prohibited after SSCG PLL clock enable setting.*

5.1.10. SSCG PLL1 Clock Gear Control Register (SYSC_SSCG1CGCNTR)

SSCG PLL1 clock gear control register(SYSC_SSCG1CGCNTR) controls SSCG PLL1 clock gear.

bit	bit 31							bit 24
Field	Reserved							
Data Attribute	R0,WX							
Prot_Attr	WPS							
Initial Value	00000000							

bit	bit 23							bit 16
Field	SSCGCGLP[7:0]							
Data Attribute	R/W							
Prot_Attr	WPS							
Initial Value	(product specification)							

bit	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
Field	SSCGCGSTP[1:0]		SSCGCGSSN[5:0]					
Data Attribute	R/W		R/W					
Prot_Attr	WPS							
Initial Value	(product specification)		(product specification)					

bit	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Field	SSCGCGSTS[1:0]		Reserved				SSCGCGSTR	SSCGCGEN
Data Attribute	R,WX		R0,WX				R,W	R/W
Prot_Attr	WPS							
Initial Value	00		0000				0	(product specification)

[bit31:24] Reserved

[bit23:16] SSCGCGLP[7:0] : SSCG PLL Clock Gear Loop Configuration Bit

These bits select the number of loop per step.

bit23:16	Description
00000000	1 loop
00000001	2 loops
00000010	3 loops
…	
11111101	254 loops
11111110	255 loops
11111111	256 loops

Notes:

- *These bits have to be set before SSCG PLL clock enable setting.*

- *Changing these bits are prohibited after SSCG PLL clock enable setting.*

[bit15:14] SSCGCGSTP[1:0] : SSCG PLL Clock Gear Step Configuration Bit

These bits select the number of step width at gear up/down operation.

bit15:14	Description
00	1 step
01	2 steps
10	3 steps
11	4 steps

Notes:

- *These bits have to be set before SSCG PLL clock enable setting.*
- *Changing these bits are prohibited after SSCG PLL clock enable setting.*

[bit13:8] SSCGCGSSN[5:0] : SSCG PLL Clock Gear Start Step Configuration Bit

These bits set the start step.

bit13:8	Description
000000	0
000001	1
000010	2
…	
111101	61
111110	62
111111	63

Notes:

- *These bits have to be set before SSCG PLL clock enable setting.*
- *Changing these bits are prohibited after SSCG PLL clock enable setting.*

[bit7:6] SSCGCGSTS[1:0] : SSCG PLL Clock Gear Status Bit

These bits show the SSCG PLL clock gear status.

Bit7:6	Description
00	Clock gear operation reaches minimum frequency
01	Gear up operation
10	Clock gear operation reaches maximum frequency
11	Gear down operation

Notes:

- *In the SSCG PLL clock stop or SSCG PLL clock stabilization wait, return a "00" status.*
- *if clock gear not used, the status is "X" (undefined).*

[bit5:2] Reserved

[bit1] SSCGCGSTR : SSCG PLL Clock Gear Start Bit

This bit is used to start clock gear operation.

bit	Description
0	No operation
1	Start clock gear operation

Note:

- *This bit is cleared after completion of clock gear operation.*

[bit0] SSCGCGEN : SSCG PLL Clock Gear Enable Bit

This bit controls enable/disable clock gear operation.

bit	Description
0	SSCG PLL clock gear disable
1	SSCG PLL clock gear enable

Notes:

- *These bits have to be set before SSCG PLL clock enable setting.*
- *Changing these bits are prohibited after SSCG PLL clock enable setting.*

5.1.11. SSCG PLL2 Clock Gear Control Register (SYSC_SSCG2CGCNTR)

SSCG PLL2 clock gear control register(SYSC_SSCG2CGCNTR) controls SSCG PLL2 clock gear.

bit	bit 31							bit 24
Field	Reserved							
Data Attribute	R0,WX							
Prot_Attr	WPS							
Initial Value	00000000							

bit	bit 23							bit 16
Field	SSCGCGLP[7:0]							
Data Attribute	R/W							
Prot_Attr	WPS							
Initial Value	(product specification)							

bit	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
Field	SSCGCGSTP[1:0]		SSCGCGSSN[5:0]					
Data Attribute	R/W		R/W					
Prot_Attr	WPS							
Initial Value	(product specification)		(product specification)					

bit	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Field	SSCGCGSTS[1:0]		Reserved				SSCGCG STR	SSCGCG EN
Data Attribute	R,WX		R0,WX				R,W	R/W
Prot_Attr	WPS							
Initial Value	00		0000				0	(product specification)

[bit31:24] Reserved

[bit23:16] SSCGCGLP[7:0] : SSCG PLL Clock Gear Loop Configuration Bit

These bits select the number of loop per step.

bit23:16	Description
00000000	1 loop
00000001	2 loops
00000010	3 loops
…	
11111101	254 loops
11111110	255 loops
11111111	256 loops

Notes:

- *These bits have to be set before SSCG PLL clock enable setting.*

- *Changing these bits are prohibited after SSCG PLL clock enable setting.*

[bit15:14] SSCGCGSTP[1:0] : SSCG PLL Clock Gear Step Configuration Bit

These bits select the number of step width at gear up/down operation.

bit15:14	Description
00	1 step
01	2 steps
10	3 steps
11	4 steps

Notes:

- *These bits have to be set before SSCG PLL clock enable setting.*
- *Changing these bits are prohibited after SSCG PLL clock enable setting.*

[bit13:8] SSCGCGSSN[5:0] : SSCG PLL Clock Gear Start Step Configuration Bit

These bits set the start step.

bit13:8	Description
000000	0
000001	1
000010	2
…	
111101	61
111110	62
111111	63

Notes:

- *These bits have to be set before SSCG PLL clock enable setting.*
- *Changing these bits are prohibited after SSCG PLL clock enable setting.*

[bit7:6] SSCGCGSTS[1:0] : SSCG PLL Clock Gear Status Bit

These bits show the SSCG PLL clock gear status.

Bit7:6	Description
00	Clock gear operation reaches minimum frequency
01	Gear up operation
10	Clock gear operation reaches maximum frequency
11	Gear down operation

Notes:

- *In the SSCG PLL clock stop or SSCG PLL clock stabilization wait, return a "00" status.*
- *if clock gear not used, the status is "X" (undefined).*

[bit5:2] Reserved

[bit1] SSCGCGSTR : SSCG PLL Clock Gear Start Bit

This bit is used to start clock gear operation.

bit	Description
0	No operation
1	Start clock gear operation

Note:

- *This bit is cleared after completion of clock gear operation.*

[bit0] SSCGCGEN : SSCG PLL Clock Gear Enable Bit

This bit controls enable/disable clock gear operation.

bit	Description
0	SSCG PLL clock gear disable
1	SSCG PLL clock gear enable

Notes:

- *These bits have to be set before SSCG PLL clock enable setting.*
- *Changing these bits are prohibited after SSCG PLL clock enable setting.*

5.1.12. SSCG PLL3 Clock Gear Control Register (SYSC_SSCG3CGCNTR)

SSCG PLL3 clock gear control register(SYSC_SSCG3CGCNTR) controls SSCG PLL3 clock gear.

bit	bit 31							bit 24
Field	Reserved							
Data Attribute	R0,WX							
Prot_Attr	WPS							
Initial Value	00000000							

bit	bit 23							bit 16
Field	SSCGCGLP[7:0]							
Data Attribute	R/W							
Prot_Attr	WPS							
Initial Value	(product specification)							

bit	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
Field	SSCGCGSTP[1:0]		SSCGCGSSN[5:0]					
Data Attribute	R/W		R/W					
Prot_Attr	WPS							
Initial Value	(product specification)		(product specification)					

bit	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Field	SSCGCGSTS[1:0]		Reserved				SSCGCGSTR	SSCGCGEN
Data Attribute	R,WX		R0,WX				R,W	R/W
Prot_Attr	WPS							
Initial Value	00		0000				0	(product specification)

[bit31:24] Reserved

[bit23:16] SSCGCGLP[7:0] : SSCG PLL Clock Gear Loop Configuration Bit

These bits select the number of loop per step.

bit23:16	Description
00000000	1 loop
00000001	2 loops
00000010	3 loops
…	
11111101	254 loops
11111110	255 loops
11111111	256 loops

Notes:

- *These bits have to be set before SSCG PLL clock enable setting.*

- *Changing these bits are prohibited after SSCG PLL clock enable setting.*

[bit15:14] SSCGCGSTP[1:0] : SSCG PLL Clock Gear Step Configuration Bit

These bits select the number of step width at gear up/down operation.

bit15:14	Description
00	1 step
01	2 steps
10	3 steps
11	4 steps

Notes:

- *These bits have to be set before SSCG PLL clock enable setting.*
- *Changing these bits are prohibited after SSCG PLL clock enable setting.*

[bit13:8] SSCGCGSSN[5:0] : SSCG PLL Clock Gear Start Step Configuration Bit

These bits set the start step.

bit13:8	Description
000000	0
000001	1
000010	2
…	
111101	61
111110	62
111111	63

Notes:

- *These bits have to be set before SSCG PLL clock enable setting.*
- *Changing these bits are prohibited after SSCG PLL clock enable setting.*

[bit7:6] SSCGCGSTS[1:0] : SSCG PLL Clock Gear Status Bit

These bits show the SSCG PLL clock gear status.

Bit7:6	Description
00	Clock gear operation reaches minimum frequency
01	Gear up operation
10	Clock gear operation reaches maximum frequency
11	Gear down operation

Notes:

- *In the SSCG PLL clock stop or SSCG PLL clock stabilization wait, return a "00" status.*
- *if clock gear not used, the status is "X" (undefined).*

[bit5:2] Reserved

[bit1] SSCGCGSTR : SSCG PLL Clock Gear Start Bit

This bit is used to start clock gear operation.

bit	Description
0	No operation
1	Start clock gear operation

Note:

- *This bit is cleared after completion of clock gear operation.*

[bit0] SSCGCGEN : SSCG PLL Clock Gear Enable Bit

This bit controls enable/disable clock gear operation.

bit	Description
0	SSCG PLL clock gear disable
1	SSCG PLL clock gear enable

Notes:

- *These bits have to be set before SSCG PLL clock enable setting.*
- *Changing these bits are prohibited after SSCG PLL clock enable setting.*

5.2. Clock Output Function Register

Clock Output Function register is used to control clock output function.

5.2.1. Clock Output Function Control Register (SYSC_CKOTCNTR)

Clock output function control register(SYSC_CKOTCNTR) is used to control clock output function.

bit	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24
Field	Reserved							ENCLKO
Data Attribute	R0,WX							R/W
Prot_Attr	WPS							
Initial Value	0000000							0

bit	bit 23	bit 22	bit 21	bit 20	bit 19	bit 18	bit 17	bit 16
Field	Reserved							
Data Attribute	R0,WX							
Prot_Attr	WPS							
Initial Value	00000000							

bit	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
Field	Reserved					CKOUTDIV[2:0]		
Data Attribute	R0,WX					R/W		
Prot_Attr	WPS							
Initial Value	00000					000		

bit	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Field	Reserved				CKSEL[3:0]			
Data Attribute	R0,WX				R/W			
Prot_Attr	WPS							
Initial Value	0000				0000			

[bit31:25] Reserved

[bit24] ENCLKO :

This bit controls enable/disable clock output function

bit	Description
0	Clock output function disable
1	Clock output function enable

[bit23:11] Reserved

[bit10:8] CKOUTDIV[2:0] : Clock Division Bits

These bits configure the clock output divider.

bit10:8	Description
000	Output clock is not divided
001	Output clock is divide by 2
010	Output clock is divide by 4
011	Output clock is divide by 8
100	Output clock is divide by 16
101	Output clock is divide by 32
110	Output clock is divide by 64
111	Output clock is divide by 128

[bit7:4] Reserved

[bit3:0] CKSEL[3:0] : Clock Select Bits

These bits select the source clock for clock output function.

bit3:0	Description
0000	Fast-CR clock is selected
0001	Slow-CR clock is selected
0010	Main clock is selected
0011	Sub clock is selected
0100	PLL0 clock is selected
0101	PLL1 clock is selected
0110	PLL2 clock is selected
0111	PLL3 clock is selected
1000	SSCG PLL0 clock is selected
1001	SSCG PLL1 clock is selected
1010	SSCG PLL2 clock is selected
1011	SSCG PLL3 clock is selected
1100	Prohibit (Fast-CR clock is selected)
1101	Prohibit (Fast-CR clock is selected)
1110	Prohibit (Fast-CR clock is selected)
1111	Clock is tied to low

6. Usage Precautions

The section explains the precautions when using the clock system.

The implementation of PLL1-3 and SSCG0-3 is dependent of product variation.

Please refer to the TRM of the each product.

If you select a clock that does not exist, Fast CR clock is selected.

The following registers for non-existent clocks no longer exist and become the reserve area.

SYSC_PLLxCGCNTR

SYSC_SSCGxCGCNTR

- The change of the PLL/SSCG PLL configuration
  The PLL/SSCG PLL configurations have to be set before PLL/SSCG PLL clock enable setting. Changing the PLL/SSCG PLL configurations are prohibited after PLL/SSCG PLL clock enable setting.

  Set the same configurations to RUN and PSS profiles if PLL/SSCG PLL clocks are enabled in PSS.

- The change of the clock gear configuration
  The clock gear configurations have to be set before PLL/SSCG PLL clock enable setting. Changing the clock gear configurations are prohibited after PLL/SSCG PLL clock enable setting

- The operation of the clock gear
  Poll the clock gear status flag during clock gear UP/DOWN operation and wait for the clock gearr stop status.

- The internal operating clock
  Do not access the peripheral resource whose internal operating clock is disabled.

- Peripherals with independent clock control the following peripherals have independent clock control from clock system.
  - Hardware watchdog timer
  - Software watchdog timer
  - RTC
  - CAN prescaler
  - EICU

- The selection of Domain clock
  Do not use Sub clock for the domain clock.

  For more details, refer to the each chapter of peripheral.

CHAPTER 6: Low Power Consumption

This chapter explains the functions and operations for low power consumption.

1. **Overview**
2. **Configuration**
3. **Explanation of Operation**
4. **Operation Procedure**
5. **Registers**
6. **Other**

LOWPOWER-TXXPT03P01R01L08-E1-XX

1. Overview

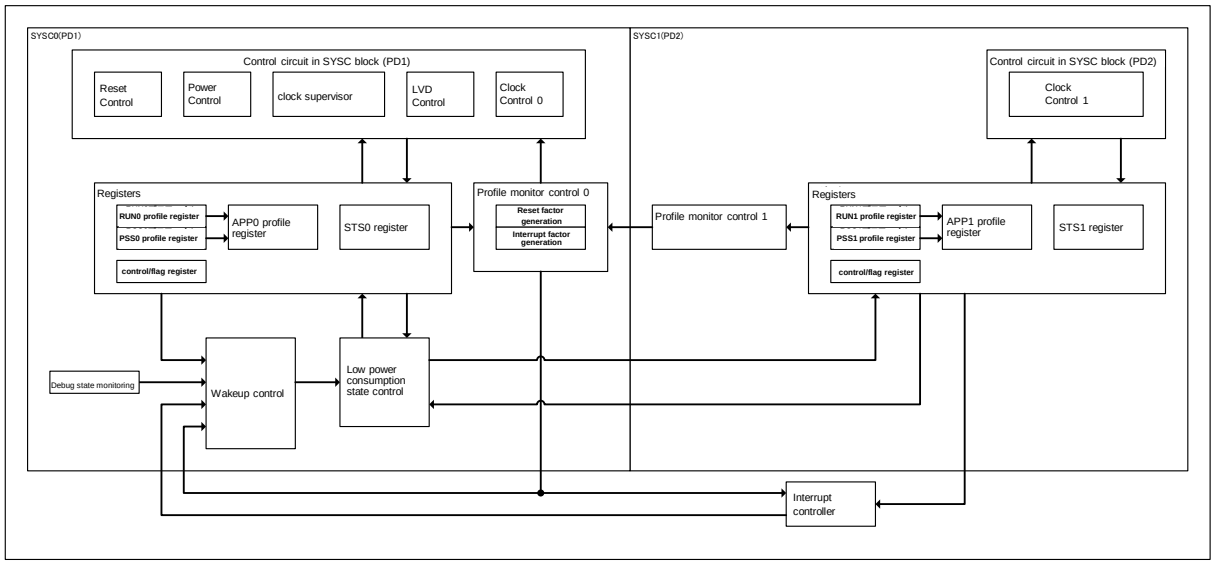
This section provides an overview of low power consumption.

This product has a wide variety of power consumption settings and can control power consumption according to the situation.

2. Configuration

This section explains the configuration for low power consumption.

Figure 2-1 Block Diagram of Low Power Consumption Control



- SYSC (SYStem Controller 0/1)
  This is the name for a collection of blocks including the clock, reset, and LVD control blocks.
- Registers
  The RUN/PSS profile setting register is updated to the APP profile register, and these results can be confirmed with the STS register. For the detailed functions of the registers, see section Registers.
- Wakeup control
  This block controls factors for the CPU in returning from the program stop state.
- Low power consumption state control
  This block manages and controls the low power consumption states.
- Profile monitoring control 0/1
  This block monitors the contents of the RUN/PSS profile.
- Reset/Interrupt factor generation
  The block generates a reset or interrupt factor when there is a profile error.
- Debug state monitoring
  The block determines the occurrence of the debug state.

3. Explanation of Operation

This section explains the operations for low power consumption.

3.1. Low Power Consumption State

This section explains low power consumption states.

Broadly, there are 2 chip states.

- RUN (Normal Operation)
- PSS (Power Saving State)

RUN

This is the state in which the CPU is operating programs or has stopped programs.

The program stop state is a state in which RUN overrides PSS enable and the WFI instruction is executed.

The CPU operates in this state (programs are operating) after an initialization reset.

In this state, a RUN profile update and a transition to PSS are possible.

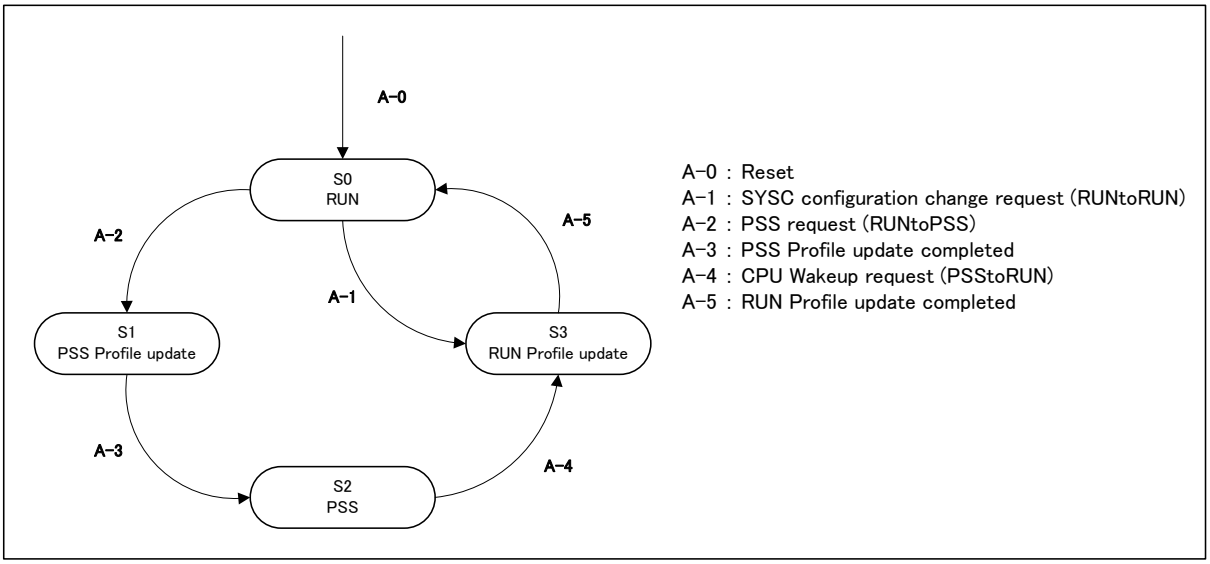
PSS

This is the state in which the CPU has stopped programs and can set low power consumption.

Access becomes impossible since the Flash macro is in low power state.

A variety of low power consumption states are supported through freely changed profile settings.

Figure 3-1 State Transition Diagram



- S0 : RUN
- S1 : Updating PSS Profile (Updating system by the setting in PSS Profile Registers)
- S2 : PSS
- S3 : Updating RUN Profile (Updating system by the setting in RUN Profile Registers)

You can set the following items freely with RUN/PSS.

- Clock related
  - Source clock oscillation enable/stop
  - Clock domain control (source selection, division, and oscillation enable and stop of each domain clock)
- CSV setting
  - ON/OFF setting
- LVD setting
  - ON/OFF setting
  - Detection voltage
  - Interrupt/Reset selection
- Regulator setting
  - Output voltage setting
  - Mode selection
- Each power domain
  - Power supply ON/OFF

3.2. Power Supply Domain

This section shows the configuration of a power supply domain.

Hierarchy

This product has the following power supply domains.

- Always ON power supply domain (PD1)
- Peripheral power supply domain (PD2)
- Core power supply domain (PD3)
- Backup RAM power supply domain (PD4_0/1)
- Extended Peripheral power supply domain (PD5_0-3)
- Sub Peripheral power supply domain (PD6_0/1)

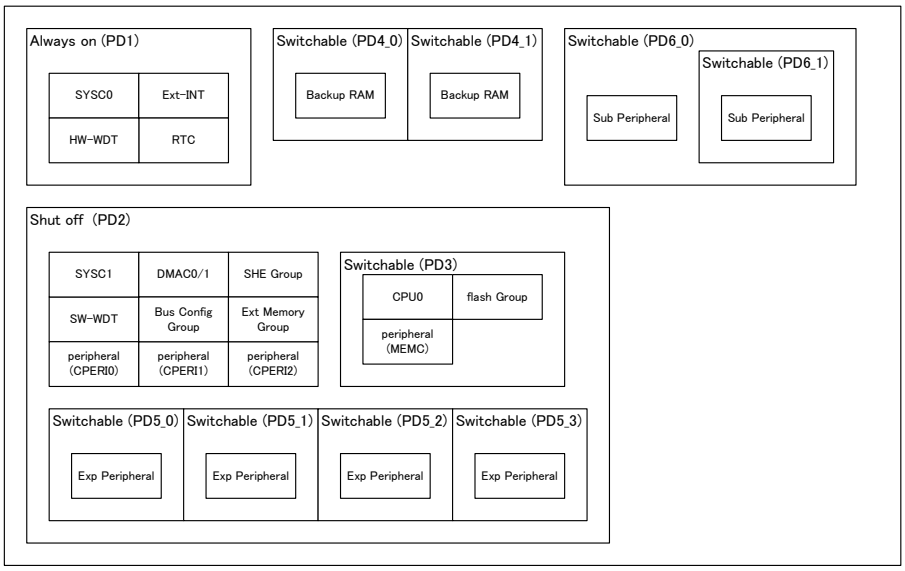
Power is always supplied to the Always ON power supply domain (PD1). Other domains can select their supply of power. Some power supply domains have a parent-child relationship. PD2 and PD3 can have only settings that are the same. To turn OFF the power supply of the Core power supply domain or Peripheral power supply domain (PD2/PD3), you need to turn OFF the power supply of the Extended Peripheral domain (PD5_0-3). The same relationship also applies to a Sub Peripheral power supply domain (PD6_0) and another Sub Peripheral power supply domain (PD6_1).

Table 3-1 List of Power Supply Domain States

State	PD1	PD2	PD3	PD4_0/1	PD5_0-3	PD6_0	PD6_1
RUN	ON	ON	ON	ON/OFF	ON	ON	ON
	ON	ON	ON	ON/OFF	OFF	ON	ON
PSS	ON	ON	ON	ON/OFF	ON	ON	ON
	ON	ON	ON	ON/OFF	OFF	ON	ON
	ON	OFF	OFF	ON/OFF	OFF	ON	ON
	ON	OFF	OFF	ON/OFF	OFF	ON	OFF
	ON	OFF	OFF	ON/OFF	OFF	OFF	OFF

Whether to use the PD3, PD5_0-3, or PD6_1 power supply domain depends on the user.

Figure 3-2 Block Diagram of Power Supply Domain



Supply of Power

The supply method is different for each power supply domain.

1. Internal regulator supply area
   PD1, PD4_0, PD4_1, PD6_0, PD6_1
2. External regulator supply area
   PD2, PD5_0, PD5_1, PD5_2, PD5_3

Startup and shutdown sequences

To start up the power supply, turning ON all Power Supply Domains power supply or external power simultaneously.

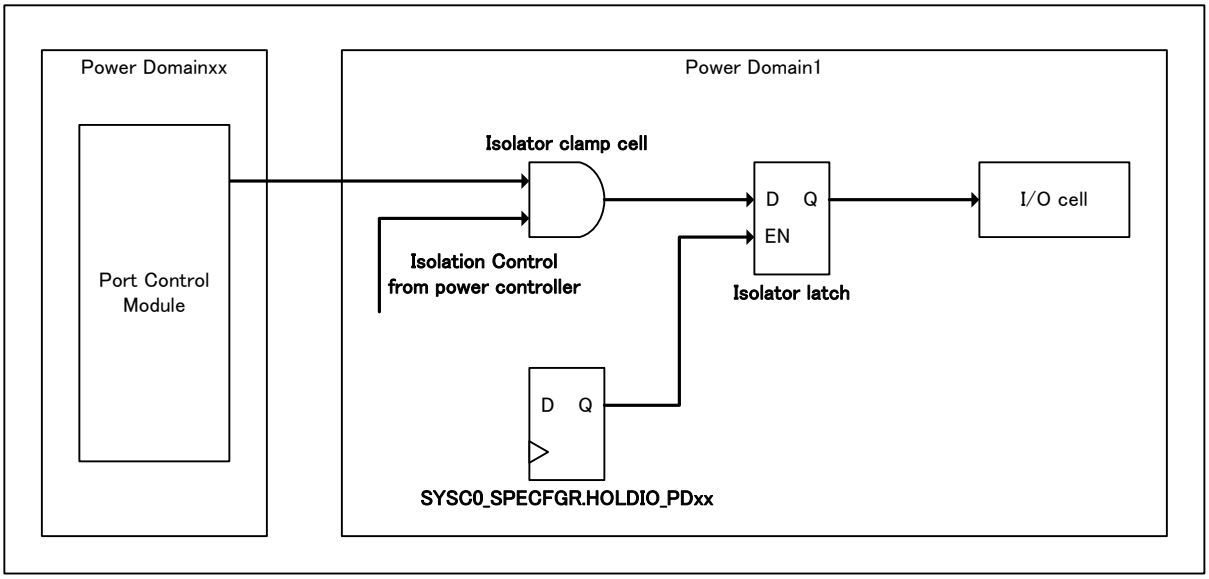
To shut down the power supply, turning OFF all Power Supply Domains power supply simultaneously.

I/O Control when Power Supply is OFF

Turning OFF the power supply of power domain also turns OFF the power supply of the circuit controlling I/O in the power domain. A latch to retain an I/O control signal is available for retaining the I/O control state at that time.

The following figure shows the circuit configuration of the I/O control block.

Figure 3-3 I/O Control Circuit Configuration



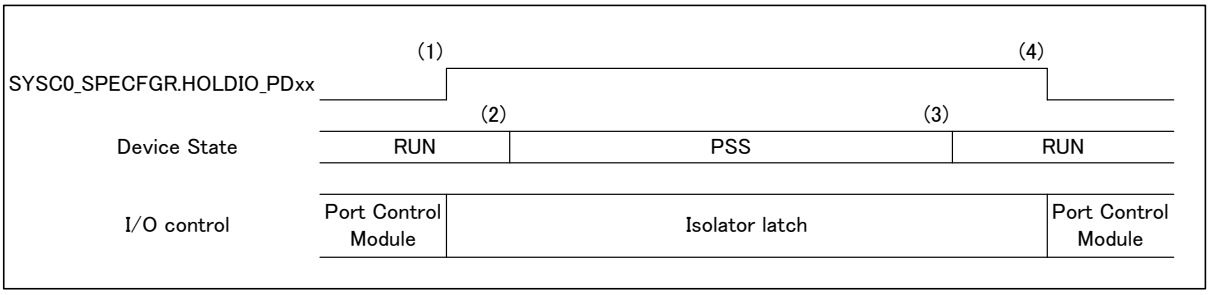
Note:

- *PDxx = PD2/PD5_0/PD5_1/PD5_2/PD5_3/PD6_0/PD6_1*

Output of the isolator clamp cell becomes fixed by control from the power controller when the PDxx power supply is turned OFF. Therefore (to prevent a signal from PDxx from having midpoint potential and destroying the PD1 cell), control for the I/O cell becomes fixed. To prevent this, the isolator latch is inserted between the I/O cell and isolator clamp cell. Before the PDxx power supply is turned OFF, register control retains a control signal for the I/O cell. As a result, the I/O cell state is retained even when the PDxx power supply is turned OFF.

The following figure shows an example of I/O control when the PDxx power supply is turned OFF.

Figure 3-4 I/O Control Time Chart



- (1) Write "1" to SYSC0_SPECFGR.HOLDIO_PDxx before a transition to PSS (turning OFF the PDxx power supply).
- (2) Upon receipt of the PSS transition trigger, there is a transition to PSS.
- (3) Factors such as interrupts cause the CPU to return from PSS and transition to RUN.
- (4) After updating the I/O control register with the user program, write "0" to SYSC0_SPECFGR.HOLDIO_PDxx.

3.3. Profile Setting Items

This section explains profile setting items.

Programs can freely set both RUN and PSS, except for some settings.

Table 3-2 shows parameters that can be set with the RUN/PSS profile in each mode.

Table 3-2 List of RUN/PSS Profile Setting Parameters

Category	Setting Item	RUN Setting (Initial Value)	RUN Setting	PSS Setting	Control Block
Power	Power domain 6_1 power supply control	ON	ON	Enable control	SYSC0
	Power domain 6_0 power supply control	ON	ON	Enable control	SYSC0
	Power domain 5_3 power supply control	ON	Enable control	Enable control	SYSC0
	Power domain 5_2 power supply control	ON	Enable control	Enable control	SYSC0
	Power domain 5_1 power supply control	ON	Enable control	Enable control	SYSC0
	Power domain 5_0 power supply control	ON	Enable control	Enable control	SYSC0
	Power domain 4_1 power supply control	ON	Enable control	Enable control	SYSC0
	Power domain 4_0 power supply control	ON	Enable control	Enable control	SYSC0
	Power domain 3 power supply control	ON	Enable control	Enable control	SYSC0
	Power domain 2 power supply control	ON	ON	Enable control	SYSC0
Clock	High-speed CR oscillation enable	Enable	Enable	Enable control	SYSC0
	Low-speed CR oscillation enable	Enable	Enable	Enable control	SYSC0
	Main clock oscillation enable	Enable	Enable control	Enable control	SYSC0
	Sub clock oscillation enable	Enable/Disable	Enable control	Enable control	SYSC0
	PLL0 oscillation enable	Disable	Enable control	Enable control	SYSC0
	PLL1 oscillation enable	Disable	Enable control	Enable control	SYSC0
	PLL2 oscillation enable	Disable	Enable control	Enable control	SYSC0
	PLL3 oscillation enable	Disable	Enable control	Enable control	SYSC0
	SSCG PLL0 oscillation enable	Disable	Enable control	Enable control	SYSC0
	SSCG PLL1 oscillation enable	Disable	Enable control	Enable control	SYSC0
	SSCG PLL2 oscillation enable	Disable	Enable control	Enable control	SYSC0

Category	Setting Item	RUN Setting (Initial Value)	RUN Setting	PSS Setting	Control Block
Clock	SSCG PLL3 oscillation enable	Disable	Enable control	Enable control	SYSC0
	Clock domain MCUC clock selection	High-speed CR	Enable control	Enable control	SYSC0
	Clock domain 0 clock selection	High-speed CR	Enable control	Enable control	SYSC1
	Clock domain 1 clock selection	High-speed CR	Enable control	Enable control	SYSC1
	Clock domain 2 clock selection	High-speed CR	Enable control	Enable control	SYSC1
	Clock domain 3 clock selection	High-speed CR	Enable control	Enable control	SYSC1
	Clock domain 4 clock selection	High-speed CR	Enable control	Enable control	SYSC1
	Clock domain 5 clock selection	High-speed CR	Enable control	Enable control	SYSC1
	Clock Domain TRC clock selection	High-speed CR	Enable control	Enable control	SYSC1
	HSSPI clock selection	High-speed CR	Enable control	Enable control	SYSC1
	TRC clock selection	High-speed CR	Enable	Enable control	SYSC1
	CPU clock oscillation enable	Enable	Enable	Disable	SYSC1
	FCLK clock oscillation enable	Enable	Enable	Enable control	SYSC1
	ATB clock oscillation enable	Enable	Enable	Enable control	SYSC1
	DBG clock oscillation enable	Enable	Enable	Enable control	SYSC1
	TRC clock oscillation enable	Enable	Enable	Enable control	SYSC1
	HPM clock oscillation enable	Enable	Enable	Enable control	SYSC1
	HPM2 clock oscillation enable	Enable	Enable	Enable control	SYSC1
	DMA clock oscillation enable	Enable	Enable control	Enable control	SYSC1
	MEMC clock oscillation enable	Enable	Enable	Enable control	SYSC1
	EXTBUS clock oscillation enable	Enable	Enable	Enable control	SYSC1
	SYSC1 clock oscillation enable	Enable	Enable	Enable control	SYSC1
	HAPP0A0 clock oscillation enabled	Enable	Enable	Enable control	SYSC1
	HAPP0A1 clock oscillation enable	Enable	Enable	Enable control	SYSC1
	HAPP1B0 clock oscillation enable	Enable	Enable	Enable control	SYSC1
	HAPP1B1 clock oscillation enable	Enable	Enable	Enable control	SYSC1

Category	Setting Item	RUN Setting (Initial Value)	RUN Setting	PSS Setting	Control Block
Clock	LLPBM clock oscillation enable	Enable	Enable	Enable control	SYSC1
	LLPBM2 clock oscillation enable	Enable	Enable	Enable control	SYSC1
	LCP clock oscillation enable	Enable	Enable	Enable control	SYSC1
	LCP0 clock oscillation enable	Enable	Enable	Enable control	SYSC1
	LCP1 clock oscillation enable	Enable	Enable	Enable control	SYSC1
	LAPP0 clock oscillation enabled	Enable	Enable	Enable control	SYSC1
	LAPP1 clock oscillation enable	Enable	Enable	Enable control	SYSC1
	LCP0A clock oscillation enable	Enable	Enable	Enable control	SYSC1
	LCP1A clock oscillation enable	Enable	Enable	Enable control	SYSC1
	LAPP0A clock oscillation enable	Enable	Enable	Enable control	SYSC1
	LAPP1A clock oscillation enable	Enable	Enable	Enable control	SYSC1
	CD1 clock oscillation enable	Enable	Enable control	Enable control	SYSC1
	CD1A0 clock oscillation enable	Enable	Enable control	Enable control	SYSC1
	CD1A1 clock oscillation enable	Enable	Enable control	Enable control	SYSC1
	CD1B0 clock oscillation enable	Enable	Enable control	Enable control	SYSC1
	CD1B1 clock oscillation enable	Enable	Enable control	Enable control	SYSC1
	CD2 clock oscillation enable	Enable	Enable control	Enable control	SYSC1
	CD2A0 clock oscillation enable	Enable	Enable control	Enable control	SYSC1
	CD2A1 clock oscillation enable	Enable	Enable control	Enable control	SYSC1
	CD2B0 clock oscillation enable	Enable	Enable control	Enable control	SYSC1
	CD2B1 clock oscillation enable	Enable	Enable control	Enable control	SYSC1
	CD3 clock oscillation enable	Enable	Enable control	Enable control	SYSC1
	CD3A0 clock oscillation enable	Enable	Enable control	Enable control	SYSC1
	CD3A1 clock oscillation enable	Enable	Enable control	Enable control	SYSC1
	CD3B0 clock oscillation enable	Enable	Enable control	Enable control	SYSC1

Category	Setting Item	RUN Setting (Initial Value)	RUN Setting	PSS Setting	Control Block
Clock	CD3B1 clock oscillation enable	Enable	Enable control	Enable control	SYSC1
	CD4 clock oscillation enable	Enable	Enable control	Enable control	SYSC1
	CD4A0 clock oscillation enable	Enable	Enable control	Enable control	SYSC1
	CD4A1 clock oscillation enable	Enable	Enable control	Enable control	SYSC1
	CD4B0 clock oscillation enable	Enable	Enable control	Enable control	SYSC1
	CD4B1 clock oscillation enable	Enable	Enable control	Enable control	SYSC1
	CD5 clock oscillation enable	Enable	Enable control	Enable control	SYSC1
	CD5A0 clock oscillation enable	Enable	Enable control	Enable control	SYSC1
	CD5A1 clock oscillation enable	Enable	Enable control	Enable control	SYSC1
	CD5B0 clock oscillation enable	Enable	Enable control	Enable control	SYSC1
	CD5B1 clock oscillation enable	Enable	Enable control	Enable control	SYSC1
	MCUCP clock oscillation enable	Enable	Enable	Enable control	SYSC1
	MCUCH clock oscillation enable	Enable	Enable	Enable control	SYSC0
	SYS clock divider	1	Enable control	Enable control	SYSC1
	ATB clock divider	1	Enable control	Enable control	SYSC1
	DBG clock divider	1	Enable control	Enable control	SYSC1
	TRC clock divider	1	Enable control	Enable control	SYSC1
	HPM clock divider	1	Enable control	Enable control	SYSC1
	SYSC1 clock divider	1	Enable control	Enable control	SYSC1
	HAPP0A0 clock divider	1	Enable control	Enable control	SYSC1
	HAPP0A1 clock divider	1	Enable control	Enable control	SYSC1
	HAPP1B0 clock divider	1	Enable control	Enable control	SYSC1
	HAPP1B1 clock divider	1	Enable control	Enable control	SYSC1
	LCP clock divider	1	Enable control	Enable control	SYSC1
	LCP0 clock divider	1	Enable control	Enable control	SYSC1
	LCP1 clock divider	1	Enable control	Enable control	SYSC1
	LAPP0 clock divider	1	Enable control	Enable control	SYSC1
	LAPP1 clock divider	1	Enable control	Enable control	SYSC1
	LCP0A clock divider	1	Enable control	Enable control	SYSC1
	LCP1A clock divider	1	Enable control	Enable control	SYSC1
	LAPP0A clock divider	1	Enable control	Enable control	SYSC1
	LAPP1A clock divider	1	Enable control	Enable control	SYSC1

Category	Setting Item	RUN Setting (Initial Value)	RUN Setting	PSS Setting	Control Block
Clock	CD1DIV clock divider	1	Enable control	Enable control	SYSC1
	CD1A0 clock divider	1	Enable control	Enable control	SYSC1
	CD1A1 clock divider	1	Enable control	Enable control	SYSC1
	CD1B0 clock divider	1	Enable control	Enable control	SYSC1
	CD1B1 clock divider	1	Enable control	Enable control	SYSC1
	CD2DIV clock divider	1	Enable control	Enable control	SYSC1
	CD2A0 clock divider	1	Enable control	Enable control	SYSC1
	CD2A1 clock divider	1	Enable control	Enable control	SYSC1
	CD2B0 clock divider	1	Enable control	Enable control	SYSC1
	CD2B1 clock divider	1	Enable control	Enable control	SYSC1
	CD3DIV clock divider	1	Enable control	Enable control	SYSC1
	CD3A0 clock divider	1	Enable control	Enable control	SYSC1
	CD3A1 clock divider	1	Enable control	Enable control	SYSC1
	CD3B0 clock divider	1	Enable control	Enable control	SYSC1
	CD3B1 clock divider	1	Enable control	Enable control	SYSC1
	CD4DIV clock divider	1	Enable control	Enable control	SYSC1
	CD4A0 clock divider	1	Enable control	Enable control	SYSC1
	CD4A1 clock divider	1	Enable control	Enable control	SYSC1
	CD4B0 clock divider	1	Enable control	Enable control	SYSC1
	CD4B1 clock divider	1	Enable control	Enable control	SYSC1
	CD5DIV clock divider	1	Enable control	Enable control	SYSC1
	CD5A0 clock divider	1	Enable control	Enable control	SYSC1
	CD5A1 clock divider	1	Enable control	Enable control	SYSC1
	CD5B0 clock divider	1	Enable control	Enable control	SYSC1
	CD5B1 clock divider	1	Enable control	Enable control	SYSC1
	MCUCH clock divider	1	Enable control	Enable control	SYSC0
	MCUCP clock divider	1	Enable control	Enable control	SYSC0
PLL0/1/2/3	L division	0x0	Enable control	Enable control	SYSC0
	M division	0x1	Enable control	Enable control	SYSC0
	N multiplier	0x0D	Enable control	Enable control	SYSC0
SSCG0/1/2/3	L division	0x0	Enable control	Enable control	SYSC0
	M division	0x1	Enable control	Enable control	SYSC0
	N multiplier	0x0D	Enable control	Enable control	SYSC0
	Modulation factor control	0x29	Enable control	Enable control	SYSC0
	Modulation mode	0	Enable control	Enable control	SYSC0
	Modulation frequency selection	0	Enable control	Enable control	SYSC0
	Modulation enable	0	Enable control	Enable control	SYSC0
LVDL1	Operation selection	(product specification)	(product specification)	(product specification)	SYSC0
	Detection voltage	(product specification)	(product specification)	(product specification)	SYSC0
	Operation enable	(product specification)	(product specification)	(product specification)	SYSC0
LVDL2	Operation selection	(product specification)	(product specification)	(product specification)	SYSC0
	Detection voltage	(product specification)	(product specification)	(product specification)	SYSC0
	Operation enable	(product specification)	(product specification)	(product specification)	SYSC0
LVDH1	Operation selection	(product specification)	(product specification)	(product specification)	SYSC0
	Detection voltage	(product specification)	(product specification)	(product specification)	SYSC0
	Operation enable	(product specification)	(product specification)	(product specification)	SYSC0

Category	Setting Item	RUN Setting (Initial Value)	RUN Setting	PSS Setting	Control Block
LVDH2	Operation selection	(product specification)	(product specification)	(product specification)	SYSC0
LVDH2	Detection voltage	(product specification)	(product specification)	(product specification)	SYSC0
	Operation enable	(product specification)	(product specification)	(product specification)	SYSC0
CSV	SSCG0 operation enable	Stop operation	Enable control	Enable control	SYSC0
	SSCG1 operation enable	Stop operation	Enable control	Enable control	SYSC0
	SSCG2 operation enable	Stop operation	Enable control	Enable control	SYSC0
	SSCG3 operation enable	Stop operation	Enable control	Enable control	SYSC0
	PLL0 operation enable	Stop operation	Enable control	Enable control	SYSC0
	PLL1 operation enable	Stop operation	Enable control	Enable control	SYSC0
	PLL2 operation enable	Stop operation	Enable control	Enable control	SYSC0
	PLL3 operation enable	Stop operation	Enable control	Enable control	SYSC0
	Sub clock operation enable	Stop operation	Enable control	Enable control	SYSC0
	Main clock operation enable	Stop operation	Enable control	Enable control	SYSC0
	High-speed CR operation enable	Stop operation	Enable control	Enable control	SYSC0
	Low-speed CR operation enable	Stop operation	Enable control	Enable control	SYSC0
Regulator	Mode	Main	Main	Enable control	SYSC0
	Output voltage	(product specification)	Enable control	Fixed	SYSC0

3.4. Profile

This section explains profiles.

For detailed descriptions of the profiles, see "List of RUN/PSS Profile Setting Parameters" For a profile update, the control circuit changes individual settings after software has started the update. A combination with a problem in the profile settings is called a profile error. A profile in the profile error state cannot be updated. For details on profile errors, see "Profile errors." Also, you can check the profile error state with the register.

This notation will be used in the following sections.

X: Don't care

0/1: Active value

C: Object to be compare

3.4.1. Profile errors

The following table shows descriptions of profile errors.

Table 3-3 List of Profile Errors

Classification	Error Description	Target State RUN/PSS	Error Flag
Clock	The PLL input clock stops when PLL0/SSCG0 clock oscillation enable is set.	RUN/PSS	SYSC0_SYSRUNPEFR:PEF0 SYSC0_SYSPSSPEFR:PEF0
	The selected source clock of clock domain 0/MCUC clock domain is a clock for which oscillation is disabled.	RUN/PSS	SYSC0_SYSRUNPEFR:PEF1 SYSC0_SYSPSSPEFR:PEF1
	The source clock for clock domain 0/MCUC clock domain was fixed at "L" in RUN.	RUN	SYSC0_SYSRUNPEFR:PEF2
	The PLL0/SSCG0 setting was changed in the PLL0/SSCG0 oscillation state.	RUN/PSS	SYSC0_SYSRUNPEFR:PEF3 SYSC0_SYSPSSPEFR:PEF3
CSV	CSV operation was enabled while monitoring/reference clock oscillation was disabled.	RUN/PSS	SYSC0_SYSRUNPEFR:PEF4 SYSC0_SYSPSSPEFR:PEF4
	The oscillation of the monitoring clock did not stop simultaneously when CSV operation in progress was disabled.	RUN/PSS	SYSC0_SYSRUNPEFR:PEF5 SYSC0_SYSPSSPEFR:PEF5
SW-WDT	The clock used by the software watchdog has stopped.	RUN/PSS	SYSC0_SYSRUNPEFR:PEF6 SYSC0_SYSPSSPEFR:PEF6
HW-WDT	The clock used by the hardware watchdog has stopped.	PSS	SYSC0_SYSPSSPEFR:PEF7
Clock Regulator	The regulator mode changed to standby during PLL oscillation enable.	PSS	SYSC0_SYSPSSPEFR:PEF8
Clock Power	The set source clock of clock domain 0 was any clock other than the high-speed CR clock, with the RUN profile that is set at the return time from the PSS that powered OFF PD2 (when a Security reevaluation is necessary).	PSS	SYSC0_SYSPSSPEFR:PEF9
Power	Power domain restrictions (hierarchical relationship/combination) were ignored.	PSS	SYSC0_SYSPSSPEFR:PEF10

Notes:

- *If sub/PLL/SSCG is not implemented, those items are inapplicable for profile error.*

3.4.1.1 The PLL0/SSCG0 input clock stops when PLL0/SSCG0 clock oscillation enable is set.

RUN

SYSC0_RUNSSCG0CNTR0	SYSC0_RUNPLL0CNTR	SYSC0_RUNCKSRER			
SSCG0ISEL	PLL0ISEL	SSCG0EN	PLL0EN	MOSCEN	CROSCEN
X	0	X	1	0	X
0	X	1	X	0	X

PSS

SYSC0_PSSSSCG0CNTR0	SYSC0_PSSPLL0CNTR	SYSC0_PSSCKSRER			
SSCG0ISEL	PLL0ISEL	SSCG0EN	PLL0EN	MOSCEN	CROSCEN
X	1	X	1	X	0
1	X	1	X	X	0
X	0	X	1	0	X
0	X	1	X	0	X

3.4.1.2 The selected source clock of clock domain 0/MCUC clock domain is a clock for which oscillation is disabled.

RUN

SYSC0_RUNCKSRER						SYSC0_ RUNCKSELR			SYSC1_ RUNCKSELR0		
SSCG0 EN	PLL0 EN	SOSC EN	MOSC EN	SCROSC EN	CROSC EN	CDMCUCCSL[2:0]			CD0CSL[2:0]		
0	X	X	X	X	X	1	0	1	X	X	X
						X	X	X	1	0	1
X	0	X	X	X	X	1	0	0	X	X	X
						X	X	X	1	0	0
X	X	0	X	X	X	0	1	1	X	X	X
						X	X	X	0	1	1
X	X	X	0	X	X	0	1	0	X	X	X
						X	X	X	0	1	0

PSS

SYSC0_PSSCKSRER						SYSC0_ PSSCKSELR			SYSC1_ PSSCKSELR0		
SSCG0 EN	PLL0 EN	SOSC EN	MOSC EN	SCROSC EN	CROSC EN	CDMCUCCSL[2:0]			CD0CSL[2:0]		
0	X	X	X	X	X	1	0	1	X	X	X
						X	X	X	1	0	1
X	0	X	X	X	X	1	0	0	X	X	X
						X	X	X	1	0	0
X	X	0	X	X	X	0	1	1	X	X	X
						X	X	X	0	1	1
X	X	X	0	X	X	0	1	0	X	X	X
						X	X	X	0	1	0
X	X	X	X	0	X	0	0	1	X	X	X
						X	X	X	0	0	1
X	X	X	X	X	0	0	0	0	X	X	X
						X	X	X	0	0	0

3.4.1.3 The source clocks of clock domain 0 and clock domain MCUC were set to fix them at "L" during CPU operation.

RUN

SYSC0_ RUNCKSELR			SYSC1_RUNCKSELR0		
CDMCUCCSL[2:0]			CD0CSL[2:0]		
0	1	1	X	X	X
1	0	1	X	X	X
1	1	1	X	X	X
X	X	X	0	1	1
X	X	X	1	0	1
X	X	X	1	1	1

- In the product which supports SSCG PLL0 clock, it doesn't include " 101 " of the set value in the profile error.
- In the product which supports Sub clock, it doesn't include " 011 " of the set value in the profile error.

PSS

There is no error condition.

3.4.1.4 The PLL0/SSCG0 setting was changed during PLL0/SSCG0 oscillation.

RUN (PLL0)

SYSC0_RUNCKSRER	SYSC0_PSSCKSRER	SYSC0_STSCKSRER	SYSC0_RUNPLL0CNTR	SYSC0_PSSPLL0CNTR	SYSC0_STSPLL0CNTR
PLL0EN	PLL0EN	PLL0EN	PLL0ISEL PLL0DIVN[7:0] PLL0DIVM[3:0] PLL0DIVL[1:0]	PLL0ISEL PLL0DIVN[7:0] PLL0DIVM[3:0] PLL0DIVL[1:0]	PLL0ISEL PLL0DIVN[7:0] PLL0DIVM[3:0] PLL0DIVL[1:0]
0	X	1	C	X	C
1	X	1	C	X	C

The registers in the categories with the letter C are compared.

RUN (SSCG0)

SYSC0_RUNCKSRER	SYSC0_PSSCKSRER	SYSC0_STSCKSRER	SYSC0_RUNSSCG0CNTR1	SYSC0_PSSSSCG0CNTR0/1	SYSC0_STSSSCG0CNTR0/1
SSCG0EN	SSCG0EN	SSCG0EN	SSCG0ISEL SSCG0DIVN[7:0] SSCG0DIVM[3:0] SSCG0DIVL[1:0] SSCG0RATE[9:0] SSCG0MODE SSCG0FREQ[1:0] SSCG0SSEN	SSCG0ISEL SSCG0DIVN[7:0] SSCG0DIVM[3:0] SSCG0DIVL[1:0] SSCG0RATE[9:0] SSCG0MODE SSCG0FREQ[1:0] SSCG0SSEN	SSCG0ISEL SSCG0DIVN[7:0] SSCG0DIVM[3:0] SSCG0DIVL[1:0] SSCG0RATE[9:0] SSCG0MODE SSCG0FREQ[1:0] SSCG0SSEN
0	X	1	C	X	C
1	X	1	C	X	C

The registers in the categories with the letter C are compared.

PSS (PLL0)

SYSC0_RUN CKSRER	SYSC0_PSS CKSRER	SYSC0_STS CKSRER	SYSC0_RUN PLL0CNTR	SYSC0_PSS PLL0CNTR	SYSC0_STS PLL0CNTR
PLL0EN	PLL0EN	PLL0EN	PLL0ISEL PLL0DIVN[7:0] PLL0DIVM[3:0] PLL0DIVL[1:0]	PLL0ISEL PLL0DIVN[7:0] PLL0DIVM[3:0] PLL0DIVL[1:0]	PLL0ISEL PLL0DIVN[7:0] PLL0DIVM[3:0] PLL0DIVL[1:0]
0	1	0	C	C	X
1	1	0	C	C	X
0	0	1	X	C	C
1	0	1	X	C	C
0	1	1	C	C	C
1	1	1	C	C	C

The registers in the categories with the letter C are compared.

PSS (SSCG0)

SYSC0_RUN CKSRER	SYSC0_PSS CKSRER	SYSC0_STS CKSRER	SYSC0_RUN SSCG0CNTR0/1	SYSC0_PSS SSCG0CNTR0/1	SYSC0_STS SSCG0CNTR0/1
SSCG0EN	SSCG0EN	SSCG0EN	SSCG0ISEL SSCG0DIVN[7:0] SSCG0DIVM[3:0] SSCG0DIVL[1:0] SSCG0RATE[9:0] SSCG0MODE SSCG0FREQ[1:0] SSCG0SSEN	SSCG0ISEL SSCG0DIVN[7:0] SSCG0DIVM[3:0] SSCG0DIVL[1:0] SSCG0RATE[9:0] SSCG0MODE SSCG0FREQ[1:0] SSCG0SSEN	SSCG0ISEL SSCG0DIVN[7:0] SSCG0DIVM[3:0] SSCG0DIVL[1:0] SSCG0RATE[9:0] SSCG0MODE SSCG0FREQ[1:0] SSCG0SSEN
0	1	0	C	C	X
1	1	0	C	C	X
0	0	1	X	C	C
1	0	1	X	C	C
0	1	1	C	C	C
1	1	1	C	C	C

The registers in the categories with the letter C are compared.

3.4.1.5 CSV operation was enabled while monitoring/reference clock oscillation was disabled.

RUN

SYSC0_RUNCKSRER						SYSC0_RUNCSVCFGR					
SSCG0 EN	PLL0 EN	SOSC EN	MOSC EN	SCR OSC EN	CR OSC EN	SSCG0 CSVE	PLL0 CSVE	SO CSVE	MO CSVE	SCR CSVE	FCR CSVE
1	X	X	0	X	X	1	X	X	X	X	X
0	X	X	1	X	X	1	X	X	X	X	X
X	1	X	0	X	X	X	1	X	X	X	X
X	0	X	1	X	X	X	1	X	X	X	X
X	X	0	X	X	X	X	X	1	X	X	X
X	X	X	0	X	X	X	X	X	1	X	X
X	X	X	0	X	X	X	X	X	X	1	X
X	X	X	0	X	X	X	X	X	X	X	1

PSS

SYSC0_PSSCKSRER						SYSC0_PSSCSVCFGR					
SSCG0 EN	PLL0 EN	SOSC EN	MOSC EN	SCR OSC EN	CR OSC EN	SSCG0 CSVE	PLL0 CSVE	SO CSVE	MO CSVE	SCR CSVE	FCR CSVE
1	X	X	0	X	X	1	X	X	X	X	X
0	X	X	1	X	X	1	X	X	X	X	X
X	1	X	0	X	X	X	1	X	X	X	X
X	0	X	1	X	X	X	1	X	X	X	X
X	X	0	X	X	X	X	X	1	X	X	X
X	X	X	X	0	X	X	X	1	X	X	X
X	X	X	0	X	X	X	X	X	1	X	X
X	X	X	X	0	X	X	X	X	1	X	X
X	X	X	X	0	X	X	X	X	X	1	X
X	X	X	0	X	X	X	X	X	X	1	X
X	X	X	X	X	0	X	X	X	X	X	1
X	X	X	0	X	X	X	X	X	X	X	1

3.4.1.6 The oscillation of the monitoring clock did not stop simultaneously when CSV operation in progress was disabled.

RUN (SSCG0)

SYSC0_RUNCSVCFGR	SYSC0_PSSCSVCFGR	SYSC0_STSCSVCFGR	SYSC0_RUNCKSRER
SSCG0CSVE	SSCG0CSVE	SSCG0CSVE	SSCG0EN
0	X	1	1

RUN (PLL0)

SYSC0_RUNCSVCFGR	SYSC0_PSSCSVCFGR	SYSC0_STSCSVCFGR	SYSC0_RUNCKSRER
PLL0CSVE	PLL0CSVE	PLL0CSVE	PLL0EN
0	X	1	1

RUN (Sub)

SYSC0_RUNCSVCFGR	SYSC0_PSSCSVCFGR	SYSC0_STSCSVCFGR	SYSC0_RUNCKSRER
SOCSVE	SOCSVE	SOCSVE	SOSCEN
0	X	1	1

RUN (Main)

SYSC0_RUNCSVCFGR	SYSC0_PSSCSVCFGR	SYSC0_STSCSVCFGR	SYSC0_RUNCKSRER
MOCSVE	MOCSVE	MOCSVE	MOSCEN
0	X	1	1

RUN(SCR)

No error condition

Note:

- *Low-speed CR oscillation cannot stop during RUN Operation. So, this Profile error does not apply in case of stopping CSV at the time of RUN Profile update.*

RUN(CR)

No error condition

Note:

- *High-speed CR oscillation cannot stop during RUN Operation. So, this Profile error does not apply in case of stopping CSV at the time of RUN Profile update.*

PSS (SSCG0)

SYSC0_RUN CSVCFGR	SYSC0_PSS CSVCFGR	SYSC0_STS CSVCFGR	SYSC0_RUN CKSRER	SYSC0_PSS CKSRER
SSCG0CSVE	SSCG0CSVE	SSCG0CSVE	SSCG0EN	SSCG0EN
0	1	0	1	X
0	0	1	X	1
1	0	1	X	1
0	1	1	1	X

PSS (PLL0)

SYSC0_RUN CSVCFGR	SYSC0_PSS CSVCFGR	SYSC0_STS CSVCFGR	SYSC0_RUN CKSRER	SYSC0_PSS CKSRER
PLL0CSVE	PLL0CSVE	PLL0CSVE	PLL0EN	PLL0EN
0	1	0	1	X
0	0	1	X	1
1	0	1	X	1
0	1	1	1	X

PSS (Sub)

SYSC0_RUN CSVCFGR	SYSC0_PSS CSVCFGR	SYSC0_STS CSVCFGR	SYSC0_RUN CKSRER	SYSC0_PSS CKSRER
SOCSVE	SOCSVE	SOCSVE	SOSCEN	SOSCEN
0	1	0	1	X
0	0	1	X	1
1	0	1	X	1
0	1	1	1	X

PSS (Main)

SYSC0_RUN CSVCFGR	SYSC0_PSS CSVCFGR	SYSC0_STS CSVCFGR	SYSC0_RUN CKSRER	SYSC0_PSS CKSRER
MOCSVE	MOCSVE	MOCSVE	MOSCEN	MOSCEN
0	1	0	1	X
0	0	1	X	1
1	0	1	X	1
0	1	1	1	X

PSS(SCR)

SYSC0_RUN CSVCFGR	SYSC0_PSS CSVCFGR	SYSC0_STS CSVCFGR	SYSC0_RUN CKSRER	SYSC0_PSS CKSRER
SCRCSVE	SCRCSVE	SCRCSVE	SCROSCEN	SCROSCEN
0	0	1	X	1
1	0	1	X	1

Note:

- *Low-speed CR oscillation cannot stop during RUN Operation. So, this Profile error does not apply in case of stopping CSV at the time of return to RUN from PSS.*

RUN(FCR)

SYSC0_RUN CSVCFGR	SYSC0_PSS CSVCFGR	SYSC0_STS CSVCFGR	SYSC0_RUN CKSRER	SYSC0_PSS CKSRER
CRCSVE	CRCSVE	CRCSVE	CROSCEN	CROSCEN
0	0	1	X	1
1	0	1	X	1

Note:

- *Low-speed CR oscillation cannot stop during RUN Operation. So, this Profile error does not apply in case of stopping CSV at the time of return to RUN from PSS.*

3.4.1.7 The clock used by the software watchdog has stopped.

RUN

SYSC0_RUNCKSRER				SWDG_CFG				
SOSCEN	MOSC EN	SCROSC EN	CROSC EN	WDENRUN	WDENPSS	ALLOW STOPCLK	CLKSEL[1:0]	
X	0	X	X	1	X	X	1	1
0	X	X	X	1	X	X	1	0

PSS

SYSC0_PSSCKSRER				SWDG_CFG				
SOSCEN	MOSC EN	SCROSC EN	CROSC EN	WDENRUN	WDENPSS	ALLOW STOPCLK	CLKSEL[1:0]	
X	0	X	X	X	1	0	1	1
0	X	X	X	X	1	0	1	0
X	X	0	X	X	1	0	0	1
X	X	X	0	X	1	0	0	0

3.4.1.8 The clock used by the hardware watchdog has stopped.

RUN

There is no error condition.

PSS

SYSC0_PSSCKSRER		HWDG_CFG				
SCROSCEN	CROSCEN	WDENRUN	WDENPSS	ALLOW STOPCLK	CLKSEL[1:0]	
0	X	X	1	0	X	1
X	0	X	1	0	X	0

3.4.1.9 The regulator mode changed to standby during PLL oscillation enable.

RUN

There is no error condition.

PSS

SYSC0_PSSCKSRER		SYSC0_PSSREGCFGR
PLL0/1/2/3EN	SSCG0/1/2/3EN	RMSEL
1	X	1
X	1	1

3.4.1.10 The set source clock for clock domain 0 was any clock other than the high-speed CR clock, with the RUN profile at the return time from the PSS that powered OFF PD2.

RUN

There is no error condition.

PSS

SYSC0_PSSPDCFGR		SYSC1_RUNCKSELR0		
PD3EN	PD2EN	CD0CSL[2:0]		
0	0	1	1	1
		1	1	0
		1	0	1
		1	0	0
		0	1	1
		0	1	0
		0	0	1

3.4.1.11 Power domain restrictions (hierarchical relationship/combination) were ignored.

RUN

There is no error condition.

PSS

SYSC0_PSSPDCFGR							
PD6_1EN	PD6_0EN	PD5_3EN	PD5_2EN	PD5_1EN	PD5_0EN	PD3EN	PD2EN
1	0	X	X	X	X	X	X
X	X	1	X	X	X	X	0
X	X	X	1	X	X	X	0
X	X	X	X	1	X	X	0
X	X	X	X	X	1	X	0
X	X	X	X	X	X	1	0
X	0	X	X	X	X	X	1

3.5. Interrupts

This section explains interrupts.

The following table shows interrupts that lead to low power consumption.

Table 3-4 List of Interrupts

Generation Condition	Factor Register	Enabling Condition	Clearing Condition
When a RUN profile update is completed	SYSC0_SYSSTSR: RUNDF0	Writing "1" to the SYSC0_SYSINTER:RUNDIE0 bit	Writing "1" to the SYSC0_SYSICLR:RUNDFCLR 0 bit
When the RUN profile register contents have an error and the RUN profile update trigger is enabled	SYSC0_SYSERRIR1: RUNERRIF0	Always enabled because of the NMI level	Writing "1" to the SYSC0_SYSERRICLR1:RUNE RRICLR0 bit
When the RUN profile register contents have an error during a transition to PSS (execution of the WFI instruction)	SYSC0_SYSERRIR1: RUNWKERRIF0	Always enabled because of the NMI level	Writing "1" to the SYSC0_SYSERRICLR1:RUNW KERRICLR0 bit
When the PSS profile register contents have an error during a transition to PSS (execution of the WFI instruction)	SYSC0_SYSERRIR1: PSSERRIF0	Always enabled because of the NMI level	Writing "1" to the SYSC0_SYSERRICLR1:PSSE RRICLR0 bit
When the value written to the PSS update enable register is invalid	SYSC0_SYSERRIR1: PSSENERRIF0	Always enabled because of the NMI level	Writing "1" to the SYSC0_SYSERRICLR1:PSSE NERRICLR0 bit
When a transition cancellation occurs during a PSS transition	SYSC0_SYSERRIR1: PSSTRGCIF0	Always enabled because of the NMI level	Writing "1" to the SYSC0_SYSERRICLR1:PSSTR GCICLR0 bit
When the value written to the RUN Profile Update Trigger Register is invalid	SYSC0_SYSERRIR1: RUNTRGERRIF	Always enabled because of the NMI level	Writing "1" to the SYSC0_SYSERRICLR1:RUNT RGERRICLR bit
When the RUN profile is updated again during a RUN profile update	SYSC0_SYSERRIR1: TRGERRIF	Always enabled because of the NMI level	Writing "1" to the SYSC0_SYSERRICLR1:TRGE RRICLR bit

3.6. Bus Error Response

This section explains bus error responses.

A bus error response is made when a register is accessed in the following circumstances.

1. An invalid value was written to the protection key setting register.
2. The protection target register was write-accessed without protection being released. (Each SCT register is a target only during the oscillation stabilization wait time.)
3. The protection key setting register was write-accessed while the protection was being released.
4. The protection target register was write-accessed from anything other than the master that released the protection key.
5. The RUN profile register group (SYSC0) was write-accessed during a RUN profile update. In this case, the written data becomes invalid.
6. The RUN/PSS profile register group (SYSC1) was write-accessed after PSSEN became valid. In this case, the written data becomes invalid.
7. The RUN profile register group (SYSC1) was write-accessed after RUN profile update enable register (SYSC1_RUNENR) became valid. In this case, the written data becomes invalid.
8. The RUN/PSS profile register group (SYSC1) was write-accessed after PSS profile update enable register became valid. In this case, the written data became invalid.

4. Operation Procedure

The section explains the operation procedures for low power consumption.

4.1. RUN Profile Update

The section explains an example of an operation procedure for RUN profile update.

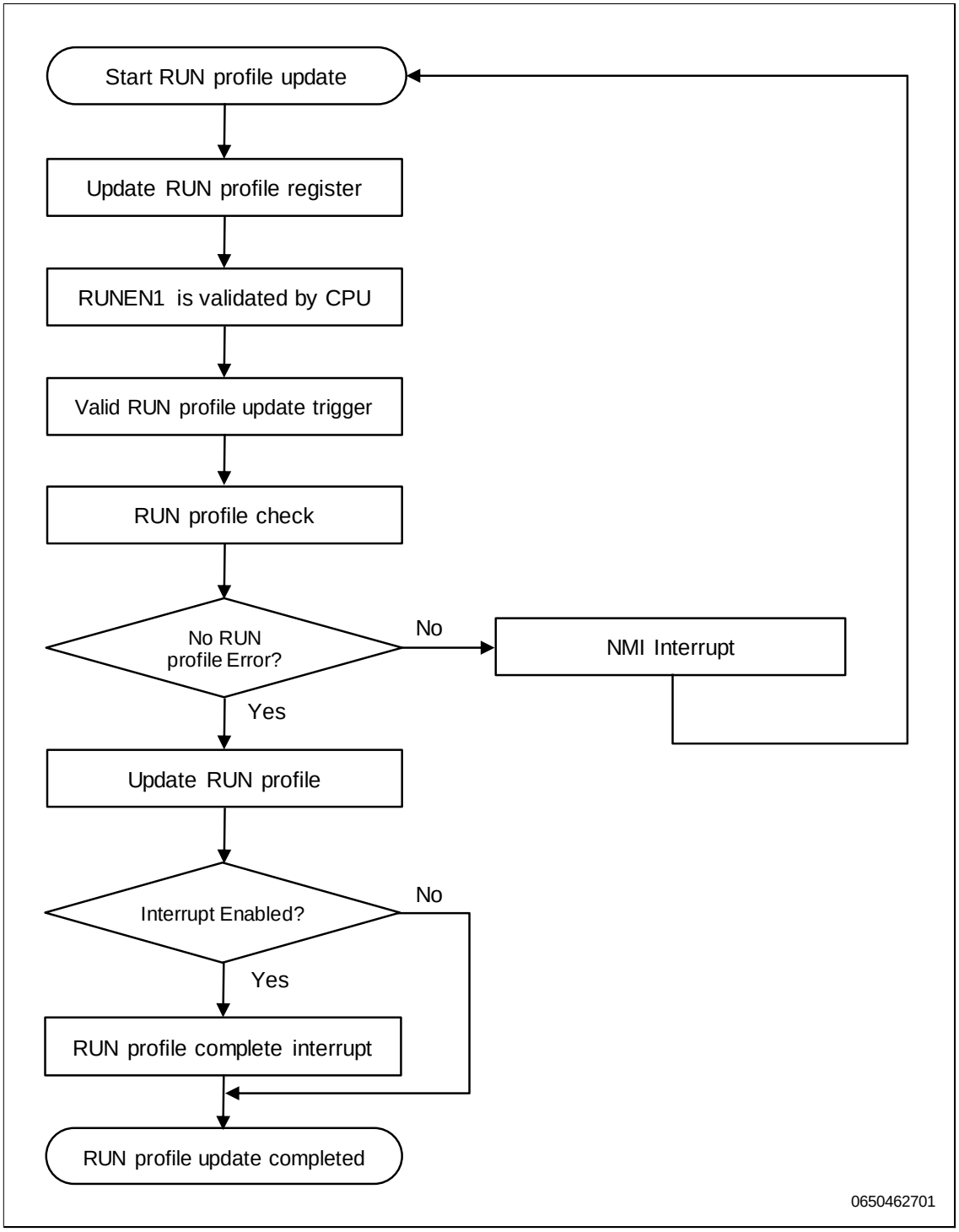
The following example explains an operation procedure for RUN profile update.

1. The user prepares a new RUN profile (register settings).
2. Write 0xAB to RUN profile update enable register (SYSC1_RUNENR).
3. Start a profile update by writing 0xAB to the RUN profile update trigger register (SYSC0_TRGRUNCNTR:APPLY_RUN).
4. The control circuit confirms the new RUN profile contents. If the contents of settings have any problem, a profile error occurs, and an error interrupt (NMI) is generated. At the same time, "1" is set in system error interrupt factor register 1 (SYSC0_SYSERRIR1:RUNERRIF0). The new profile contents are discarded, and the circuit operates with the contents of the profile currently in use.
5. When the new RUN profile contents have no problem, the control circuit reflects the profile contents as follows.
    1. The circuit sets "1" in the system status register (SYSC0_SYSSTSR:RUNSTS0).
    2. The circuit copies the RUN profile contents to the APP profile.
    3. The copying reflects the following: clock oscillation enable/stop (also including oscillation stabilization wait), CSV setting changes, LVD setting changes, clock operation settings (source clock change, division, ON/OFF of each clock source), ON/OFF of each power domain, and clock stop settings (stopping a source clock).
    4. When the RUN profile update is completed, the circuit clears the system status register (SYSC0_SYSSTSR:RUNSTS0) to "0" and sets "1" in the system status register (SYSC0_SYSSTSR:RUNDF0). Also, if the system status interrupt enable register (SYSC0_SYSINTER:RUNDIE0) is enabled, an interrupt is generated.

Notes:

- *Any attempt to update a RUN profile again while a new RUN profile is being updated causes the generation of a system error interrupt (SYSC0_SYSERRIR1:RUNTRGERRIF). Furthermore, the RUN profile of the re-update attempt becomes invalid.*
- *Before updating the RUN profile, confirm in the flags of the profile status register (SYSC0_SYSPROSTSR:RUNPSTS) that there is no profile error. Any update of a RUN profile in the profile error state causes the generation of an NMI interrupt. Furthermore, the RUN profile settings are discarded.*

Figure 4-1 RUN Profile Update Operation Flowchart



4.2. Transition from RUN to PSS

This section explains an example of an operation procedure for a transition from RUN to PSS.

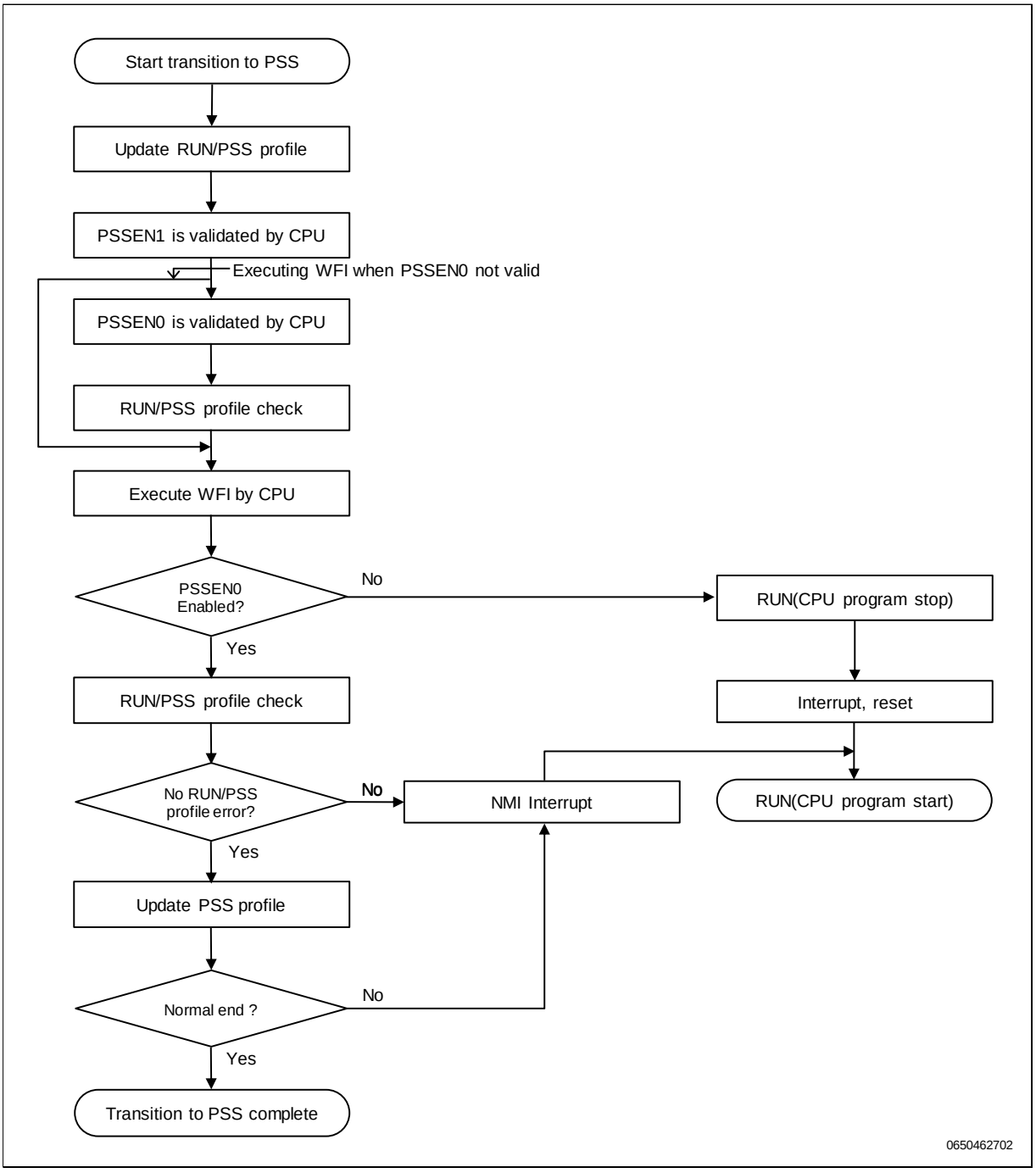
The following example explains an operation procedure for the transition to PSS.

1. The user prepares a new PSS profile and RUN profile (register settings). For the RUN profile, make settings for returning from PSS.
2. Write 0xBA to the PSS profile update enable register (SYSC1_PSSENR:PSSEN1).
3. Write 0xBA to the PSS profile update enable register (SYSC0_PSSENR:PSSEN0).
4. Have the CPU execute the WFI instruction. A profile update begins, with the instruction as a trigger.
5. Confirm the contents of the PSS/RUN profile to update. If the contents of the RUN/PSS profile settings have a problem, the system error interrupt flag register (SYSC0_SYSERRIR1:RUNWKERRIF0/PSSERRIF0) becomes valid. If either of the profiles has an error, an NMI interrupt is generated. The interrupt cannot be made invalid. Check the flag register contents for a description of the profile error, and correct the part that has the problem.
6. When the PSS profile contents have no problem, the control circuit reflects the profile contents as follows.
    1. The circuit sets the PSS profile setting state of the CPU (SYSC0_SYSSTSR:PSSSTS0) to "1".
    2. The circuit copies the PSS profile contents to the APP profile.
    3. The circuit checks whether a WAKEUP request has been generated. If a WAKEUP request has been generated, the transition to PSS is canceled. Then, an NMI interrupt is generated, and the system error interrupt factor register (SYSC0_SYSERRIR1:PSSTRGCIF0) is set to "1".
    4. The circuit handles the following: clock oscillation enable/stop (also including oscillation stabilization wait), CSV setting changes, LVD setting changes, clock operation settings (source clock change, division, ON/OFF of each clock source), ON/OFF of each power domain, and clock stop settings (stopping a source clock).
    5. The circuit sets the PSS profile setting state (SYSC0_SYSSTSR:PSSSTS0) to "0" and the PSS profile completion flag (SYSC0_SYSSTSR:PSSDF0) "1".
7. The transition to PSS is completed. If the regulator settings have been changed at the same time, the settings have changed.

■ Notes:

- *If the WIF instruction is executed without the PSS profile update enable register (SYSC0_PSSENR:PSSEN0) enabled, there is no transition to PSS. The CPU remains in RUN (programs stopped). To return from WFI, return with an interrupt, reset, etc.*
- *To turn OFF PD2, enable the HOLD data latch setting bit. If you do not set it, I/O state can not be retained.*
- *When returning to Run mode, Main_OSC must be valid. Therefore, be sure to enable Main_OSC setting of the RUN mode profile before entering the PSS mode.*

Figure 4-2 RUN to PSS Operation Flowchart



■ Prepare Wakeup from PSS Mode by Interrupt

There are multiple ways to setup interrupts for wakeup from PSS mode. The major steps that apply in most cases are described in the following list.

Table 4-1 Steps to prepare Wakeup from PSS by Interrupt

Step	Purpose	Action
S1	Make sure that the CPU accepts interrupts.	Set the I-flag in PSR of the CPU core to "0". This enables any interrupt that might occur.
S2	If it is intended to finish pending interrupts before PSS mode, wait for the indication from IRCn.	Poll the IRCn_IRQST:nIRQ flag. While reading "0", not all ISRs are finished. Once the logic value "1" is read, the IRC will not accept any subsequent interrupts. (*1)
S3	Configure the desired wake-up sources.	Enable the interrupts that are intended for wake-up, disable others. (*2)
S4	Remove the lock from IRC.	Write to any of the registers that are listed in the description of IRCn_IRQST:nIRQ. - IRCn_IRQVAr - IRCn_IRQPL0-127 - IRCn_IRQS0-15 - IRCn_IRQR0-15 - IRCn_IRQCES0-15 - IRCn_IRQCEC0-15 - IRCn_IRQCE0-15 - IRCn_IRQHC - IRCn_IRQPLM - IRCn_CSR
S5	Prepare the transition from RUN to PSS mode	Follow the steps for transition to PSS mode as described above.
S6	Transition to PSS mode	Execute the WFI instruction.

Notes:

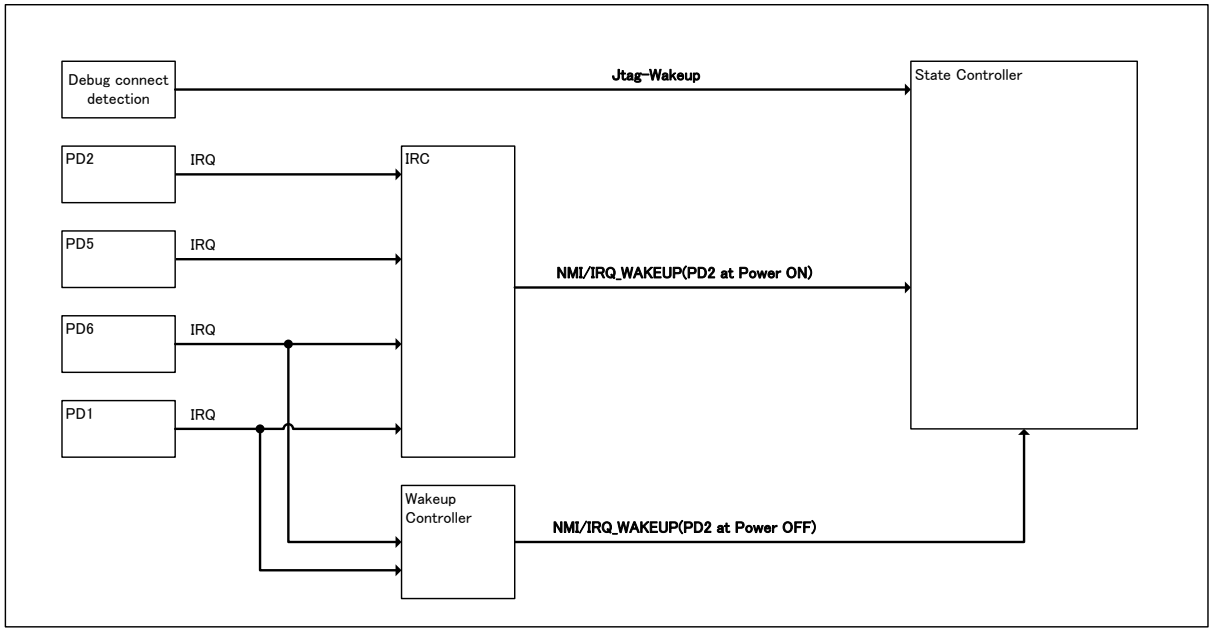
- *(*1): If the read access in step S2 returns a pending interrupt request, the processor will execute the interrupt service routine. After return from interrupt, the IRQ interrupt status bit nIRQ will show "end of return from interrupt instruction".*
- *(*2): If the interrupt should be used for wakeup without a specific interrupt service routine, this can be achieved by setting the I-Flag to "1" before S3. It must be considered that any write access to one of the registers listed in S4 will remove the lock from the IRC.*

4.3. Transition from PSS to RUN

This section explains an example of an operation procedure for a transition from PSS to RUN.

The following figure is a schematic diagram of the Wakeup circuit.

Figure 4-3 Schematic Diagram of Wakeup Circuit



The return factors have some restrictions due to disconnection of the power supply. The following table shows factors in returning from PSS.

Table 4-1 Factors in Returning from PSS (Interrupts)

Power Supply State						Interrupt Return Factor	Return Path
PD1	PD2/3	PD4_x	PD5_x	PD6_0	PD6_1		
ON	ON	ON/OFF	ON	ON	ON	Peripheral interrupt (PD1) Peripheral interrupt (PD2) Peripheral interrupt (PD5) Peripheral interrupt (PD6)	IRC
ON	ON	ON/OFF	OFF	ON	ON	Peripheral interrupt (PD1) Peripheral interrupt (PD2) Peripheral interrupt (PD6)	IRC
ON	OFF	ON/OFF	OFF	ON	ON	Peripheral interrupt (PD1) Peripheral interrupt (PD6)	Wakeup Controller
ON	OFF	ON/OFF	OFF	ON	OFF	Peripheral interrupt (PD1) Peripheral interrupt (PD6)	Wakeup Controller
ON	OFF	ON/OFF	OFF	OFF	OFF	Peripheral interrupt (PD1)	Wakeup Controller

- Peripheral interrupt (PD1): External interrupt, RTC, NMI, low-voltage detection *1, HW-WDT *1
- Peripheral interrupt (PD2): Interrupt of peripheral mounted resource, SW-WDT, etc.

- Peripheral interrupt (PD5): Interrupt of extended peripheral, etc.
- Peripheral interrupt (PD6): Interrupt of Sub peripheral, etc.

*1 Where used as an interrupt when an abnormality is detected

Register settings (JTAG return register) can be return factors when a tool is connected.

The following example shows a procedure that returns CPU from PSS to RUN.

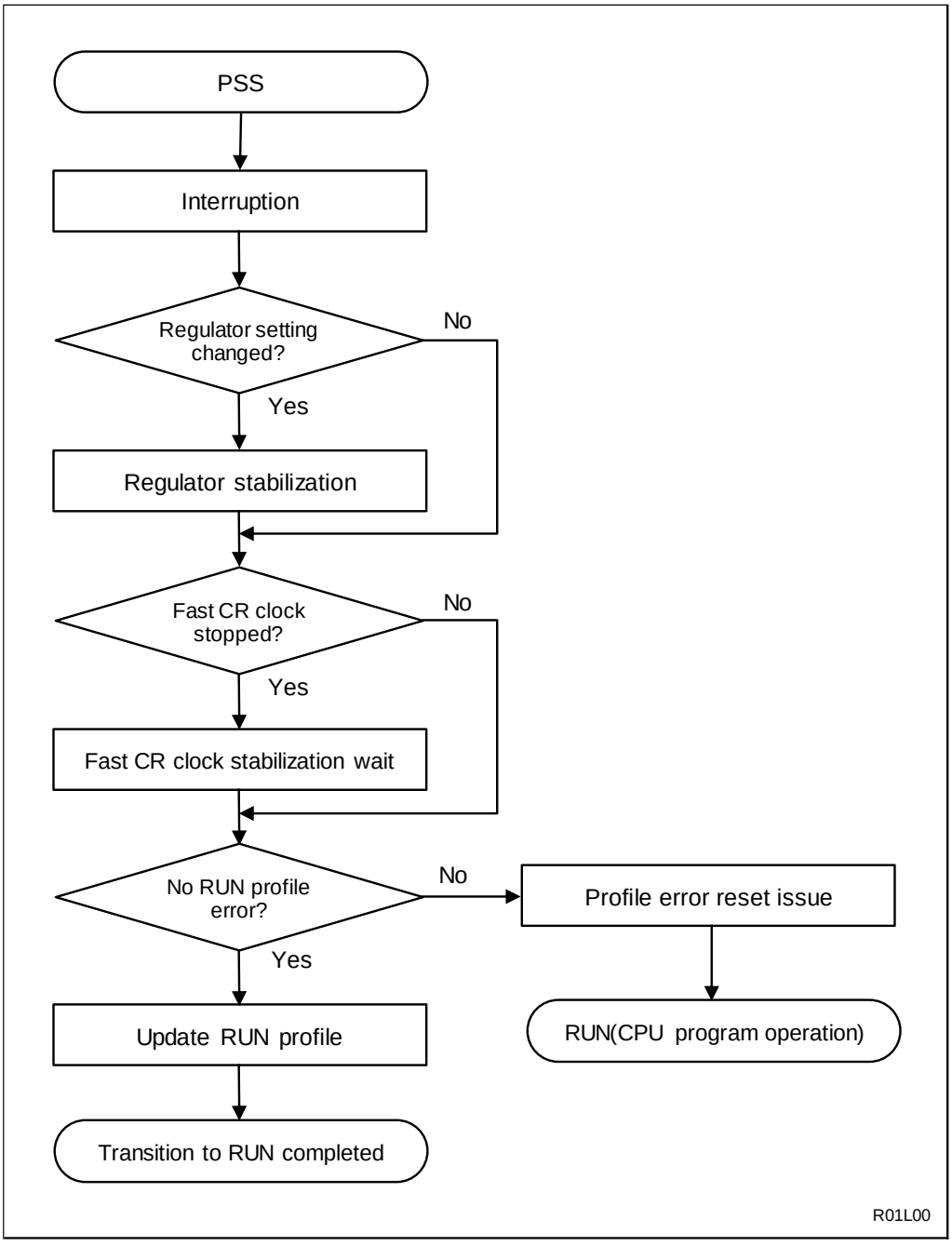
1. Generate a return factor. After the control circuit recognizes the return factor, the CPU starts to return.
2. If the high-speed CR clock has stopped, the CPU waits for oscillation of the high-speed CR clock to stabilize.
3. If the contents of the RUN profile settings have a problem (bit error, etc.), a profile error reset is generated.
4. When the RUN profile contents have no problem, the control circuit reflects the profile contents as follows.
    1. The circuit sets the RUN profile setting state (SYSC0_SYSSTSR:RUNSTS(product specification)) to "1".
    2. The circuit copies the RUN profile contents to the APP profile.
    3. The circuit handles the following: clock oscillation enable/stop (also including oscillation stabilization wait), CSV setting changes, LVD setting changes, clock operation settings (source clock change, division, ON/OFF of each clock source), ON/OFF of each power domain, and clock stop settings (stopping a source clock). The circuit sets the RUN profile setting state (SYSC0_SYSSTSR:RUNSTS(product specification)) to "0".
5. CPU waits for Flash macro to stabilize from low power state to normal state. Stabilize wait time will occur for regulator if voltage/mode change occurs.
6. There is a transition to RUN, and the CPU starts program operations.

Notes:

- *If a profile error occurs, the RUN profile settings have the initial values.*
- *Disable PSSEN when updating profile register. Written data is invalid while write-access occurs to profile register when PSSEN is enabled. Bus error will occur.*
- *If the power domain 6 is turned OFF in PSS mode, it is prohibited to enable PLL/SSCG oscillation. After the power domain 6 is turn ON in RUN mode, please enable PLL/SSCG oscillation in RUN profile updating.*

The following figure is a flowchart of PSS to RUN transition operations.

Figure 4-4 PSS to RUN Operation Flowchart



5. Registers

This section explains the low power consumption registers.

The system controller registers consist of the following register groups.

- SYSC0
  - Protection register group
  - RUN profile register group
  - PSS profile register group
  - APP profile register group
  - STS profile register group
  - System register group
  - Clock supervisor register group
  - Reset control register group
  - Source clock timer register group for low-speed CR
  - Source clock timer register group for high-speed CR
  - Source clock timer register group for main oscillation
  - Source clock timer register group for sub oscillation
  - Clock control register group
  - Special setting register group
  - Debug register group
- MCU CONFIG
  - Read interruption at one time
- SYSC1
  - Protection register group
  - RUN profile register group
  - PSS profile register group
  - APP profile register group
  - STS profile register group

- For details on the clock supervisor register group in the above items, see the following chapter: "CLOCK SUPERVISOR."
- For details on the reset control register group in the above items, see the following chapter: "RESET."
- For details on the respective source clock timer register groups in the above items, see the following chapter: "SOURCE CLOCK TIMER."
- For details on the clock control register group in the above items, see the following chapter: "CLOCK SYSTEM."

About "x" Used for Register names

- PLL number 0/1/2/3 will substitute for PLL"x".
- SSCG number 0/1/2/3 will substitute for SSCG"x".
-

Notes:

- *Protection is applied to the SYSC register. To write to the register, the protection key must first be unlocked. Invalid written data is discarded. The following shows an example of the procedure for accessing the SYSC register.*

  1. The accessing master unlocks the protection key.
  2. The master that unlocked the protection key writes to the SYSC register.

3. The hardware locks the protection key.
   Repeat steps 1 to 3 until completing the required settings.

– *The following shows protection effective conditions.*

  1. The protection target register was write-accessed from the master that released the protection key.
  2. The protection target register was write-accessed from anything other than the master that unlocked the protection key.

– *Protection range of SYSC0_PROTKEYR is 0xB060_0080-0xB060_06FC.*
– *Protection range of SYSC1_PROTKEYR is 0xB030_0080-0xB030_027C.*
– *Protection is not applied to the debug register group.*
– *PLL number 0/1/2/3 will substitute for PLL"x".*
– *SSCG number 0/1/2/3 will substitute for SSCG"x".*

SYSC0 Register groups

Table 5-1 Protection Register Group

Abbreviated Register Name	Register Name	Reference
SYSC0_PROTKEYR	Protection key setting register	5.1.1

Table 5-2 RUN Profile Register Group

Abbreviated Register Name	Register Name	Reference
SYSC0_RUNPDCFGR	RUN power domain setting register	5.2.1
SYSC0_RUNCKSRER	RUN clock source enable register	5.2.2
SYSC0_RUNCKSELR	RUN clock selection register	5.2.3
SYSC0_RUNCKER	RUN clock enable register	5.2.4
SYSC0_RUNCKDIVR	RUN clock divider register	5.2.5
SYSC0_RUNPLLxCNTR	RUN PLLx control register	5.2.6
SYSC0_RUNSSCGxCNTR0	RUN SSCGx control register 0	5.2.7
SYSC0_RUNSSCGxCNTR1	RUN SSCGx control register 1	5.2.8
SYSC0_RUNLVDCFGR	RUN low-voltage detection setting register	5.2.9
SYSC0_RUNCSVCFGR	RUN clock supervisor setting register	5.2.10
SYSC0_RUNREGCFGR	RUN regulator setting register	5.2.11
SYSC0_TRGRUNCNTR	RUN profile update trigger register	5.2.12

x=0, 1, 2, 3

Table 5-3 PSS Profile Register Group

Abbreviated Register Name	Register Name	Reference
SYSC0_PSSPDCFGR	PSS power domain setting register	5.3.1
SYSC0_PSSCKSRER	PSS clock source enable register	5.3.2
SYSC0_PSSCKSELR	PSS clock selection register	5.3.3
SYSC0_PSSCKER	PSS clock enable register	5.3.4
SYSC0_PSSCKDIVR	PSS clock divider register	5.3.5
SYSC0_PSSPLLxCNTR	PSS PLLx control register	5.3.6
SYSC0_PSSSSCGxCNTR0	PSS SSCGx control register 0	5.3.7
SYSC0_PSSSSCGxCNTR1	PSS SSCGx control register 1	5.3.8
SYSC0_PSSLVDCFGR	PSS low-voltage detection setting register	5.3.9
SYSC0_PSSCSVCFGR	PSS clock supervisor setting register	5.3.10
SYSC0_PSSREGCFGR	PSS regulator setting register	5.3.11
SYSC0_PSSENR	PSS profile update enable register	5.3.12

x=0, 1, 2, 3

Table 5-4 APP Profile Register Group

Abbreviated Register Name	Register Name	Reference
SYSC0_APPPDCFGR	APP power domain setting register	5.4.1
SYSC0_APPCKSRER	APP clock source enable register	5.4.2
SYSC0_APPCKSELR	APP clock selection register	5.4.3
SYSC0_APPCKER	APP clock enable register	5.4.4
SYSC0_APPCKDIVR	APP clock divider register	5.4.5
SYSC0_APPPLLxCNTR	APP PLLx control register	5.4.6
SYSC0_APPSSCGxCNTR0	APP SSCGx control register 0	5.4.7
SYSC0_APPSSCGxCNTR1	APP SSCGx control register 1	5.4.8
SYSC0_APPLVDCFGR	APP low-voltage detection setting register	5.4.9
SYSC0_APPCSVCFGR	APP clock supervisor setting register	5.4.10
SYSC0_APPREGCFGR	APP regulator setting register	5.4.11

x=0, 1, 2, 3

Table 5-5 STS Profile Register Group

Abbreviated Register Name	Register Name	Reference
SYSC0_STSPDCFGR	STS power domain setting register	5.5.1
SYSC0_STSCKSRER	STS clock source enable register	5.5.2
SYSC0_STSCKSELR	STS clock selection register	5.5.3
SYSC0_STSCKER	STS clock enable register	5.5.4
SYSC0_STSCKDIVR	STS clock divider register	5.5.5
SYSC0_STSPLLxCNTR	STS PLLx control register	5.5.6
SYSC0_STSSSCGxCNTR0	STS SSCGx control register 0	5.5.7
SYSC0_STSSSCGxCNTR1	STS SSCGx control register 1	5.5.8
SYSC0_STSLVDCFGR	STS low-voltage detection setting register	5.5.9
SYSC0_STSCSVCFGR	STS clock supervisor setting register	5.5.10
SYSC0_STSREGCFGR	STS regulator setting register	5.5.11

x=0, 1, 2, 3

Table 5-6 System Register Group

Abbreviated Register Name	Register Name	Reference
SYSC0_SYSIDR	System ID register	5.6.1
SYSC0_SYSPFIDR	Platform ID register	5.6.2
SYSC0_SYSSTSR	System status register	5.6.3
SYSC0_SYSINTER	System status interrupt enable register	5.6.4
SYSC0_SYSICLR	System status flag interrupt clear register	5.6.5
SYSC0_SYSERRIR0	System error interrupt factor register 0	5.6.6
SYSC0_SYSERRIR1	System error interrupt factor register 1	5.6.7
SYSC0_SYSERRICLR0	System error interrupt factor clear register 0	5.6.8
SYSC0_SYSERRICLR1	System error interrupt factor clear register 1	5.6.9
SYSC0_SYSPROSTSR	Profile status register	5.6.10
SYSC0_SYSRUNPEFR	RUN profile error flag register	5.6.11
SYSC0_SYSPSSPEFR	PSS profile error flag register	5.6.12

Table 5-7 Special Setting Register Group

Abbreviated Register Name	Register Name	Reference
SYSC0_SPECFGR	System special setting register	5.7.1

Table 5-8 Debug Register Group

Abbreviated Register Name	Register Name	Reference
SYSC0_JTAGDETECT	JTAG detection register	5.8.1
SYSC0_JTAGCNFG	JTAG setting register	5.8.2
SYSC0_JTAGWAKEUP	JTAG restore register	5.8.3

MCU Config Register groups

Table 5-9 Interruption Read Register Group

Abbreviated Register Name	Register Name	Reference
MCG_IRSR0	MCU_Config_Group Interrupt Request Status Register0	5.9.1
MCG_IRSR1	MCU_Config_Group Interrupt Request Status Register1	5.9.2
MCG_IRSR2	MCU_Config_Group Interrupt Request Status Register2	5.9.3
MCG_IRSR3	MCU_Config_Group Interrupt Request Status Register3	5.9.4
MCG_IRSR4	MCU_Config_Group Interrupt Request Status Register4	5.9.5

SYSC1 Register groups

Table 5-10 Protection Register Group

Abbreviated Register Name	Register Name	Reference
SYSC1_PROTKEYR	Protection key setting register	5.10.1

Table 5-11 RUN Profile Register Group

Abbreviated Register Name	Register Name	Reference
SYSC1_RUNCKSELR0	RUN clock selection register 0	5.11.1
SYSC1_RUNCKSELR1	RUN clock selection register 1	5.11.2
SYSC1_RUNCKSELR2	RUN clock selection register 2	5.11.3
SYSC1_RUNCKER0	RUN clock enable register 0	5.11.4
SYSC1_RUNCKER1	RUN clock enable register 1	5.11.5
SYSC1_RUNCKER2	RUN clock enable register 2	5.11.6
SYSC1_RUNCKDIVR0	RUN clock divider register 0	5.11.7
SYSC1_RUNCKDIVR1	RUN clock divider register 1	5.11.8
SYSC1_RUNCKDIVR2	RUN clock divider register 2	5.11.9
SYSC1_RUNCKDIVR3	RUN clock divider register 3	5.11.10
SYSC1_RUNCKDIVR4	RUN clock divider register 4	5.11.11
SYSC1_RUNCKDIVR5	RUN clock divider register 5	5.11.12
SYSC1_RUNCKDIVR6	RUN clock divider register 6	5.11.13
SYSC1_RUNCKDIVR7	RUN clock divider register 7	5.11.14
SYSC1_RUNCKDIVR8	RUN clock divider register 8	5.11.15
SYSC1_RUNCKDIVR9	RUN clock divider register 9	5.11.16
SYSC1_RUNENR	RUN profile update enable register	5.11.17

Table 5-12 PSS Profile Register Group

Abbreviated Register Name	Register Name	Reference
SYSC1_PSSCKSELR0	PSS clock selection register 0	5.12.1
SYSC1_PSSCKSELR1	PSS clock selection register 1	5.12.2
SYSC1_PSSCKSELR2	PSS clock selection register 2	5.12.3
SYSC1_PSSCKER0	PSS clock enable register 0	5.12.4
SYSC1_PSSCKER1	PSS clock enable register 1	5.12.5
SYSC1_PSSCKER2	PSS clock enable register 2	5.12.6
SYSC1_PSSCKDIVR0	PSS clock divider register 0	5.12.7
SYSC1_PSSCKDIVR1	PSS clock divider register 1	5.12.8
SYSC1_PSSCKDIVR2	PSS clock divider register 2	5.12.9
SYSC1_PSSCKDIVR3	PSS clock divider register 3	5.12.10
SYSC1_PSSCKDIVR4	PSS clock divider register 4	5.12.11
SYSC1_PSSCKDIVR5	PSS clock divider register 5	5.12.12
SYSC1_PSSCKDIVR6	PSS clock divider register 6	5.12.13
SYSC1_PSSCKDIVR7	PSS clock divider register 7	5.12.14
SYSC1_PSSCKDIVR8	PSS clock divider register 8	5.12.15
SYSC1_PSSCKDIVR9	PSS clock divider register 9	5.12.16
SYSC1_PSSENR	PSS profile update enable register	5.12.17

Table 5-13 APP Profile Register Group

Abbreviated Register Name	Register Name	Reference
SYSC1_APPCKSELR0	APP clock selection register 0	5.13.1
SYSC1_APPCKSELR1	APP clock selection register 1	5.13.2
SYSC1_APPCKSELR2	APP clock selection register 2	5.13.3
SYSC1_APPCKER0	APP clock source enable register 0	5.13.4
SYSC1_APPCKER1	APP clock source enable register 1	5.13.5
SYSC1_APPCKER2	APP clock source enable register 2	5.13.6
SYSC1_APPCKDIVR0	APP clock divider register 0	5.13.7
SYSC1_APPCKDIVR1	APP clock divider register 1	5.13.8
SYSC1_APPCKDIVR2	APP clock divider register 2	5.13.9
SYSC1_APPCKDIVR3	APP clock divider register 3	5.13.10
SYSC1_APPCKDIVR4	APP clock divider register 4	5.13.11
SYSC1_APPCKDIVR5	APP clock divider register 5	5.13.12
SYSC1_APPCKDIVR6	APP clock divider register 6	5.13.13
SYSC1_APPCKDIVR7	APP clock divider register 7	5.13.14
SYSC1_APPCKDIVR8	APP clock divider register 8	5.13.15
SYSC1_APPCKDIVR9	APP clock divider register 9	5.13.16

Table 5-14 STS Profile Register Group

Abbreviated Register Name	Register Name	Reference
SYSC1_STSCKSELR0	STS clock selection register 0	5.14.1
SYSC1_STSCKSELR1	STS clock selection register 1	5.14.2
SYSC1_STSCKSELR2	STS clock selection register 2	5.14.3
SYSC1_STSCKER0	STS clock source enable register 0	5.14.4
SYSC1_STSCKER1	STS clock source enable register 1	5.14.5
SYSC1_STSCKER2	STS clock source enable register 2	5.14.6
SYSC1_STSCKDIVR0	STS clock divider register 0	5.14.7
SYSC1_STSCKDIVR1	STS clock divider register 1	5.14.8
SYSC1_STSCKDIVR2	STS clock divider register 2	5.14.9
SYSC1_STSCKDIVR3	STS clock divider register 3	5.14.10
SYSC1_STSCKDIVR4	STS clock divider register 4	5.14.11
SYSC1_STSCKDIVR5	STS clock divider register 5	5.14.12
SYSC1_STSCKDIVR6	STS clock divider register 6	5.14.13
SYSC1_STSCKDIVR7	STS clock divider register 7	5.14.14
SYSC1_STSCKDIVR8	STS clock divider register 8	5.14.15
SYSC1_STSCKDIVR9	STS clock divider register 9	5.14.16

Register Map (SYSC0)

Table 5-15 shows a register address map.

Table 5-15 Low Power Consumption Register Address Map (SYSC0)

Offset	+3	+2	+1	+0	Area
0x0000_0000	SYSC0_PROTKEYR 00000000_00000000_00000000_00000000				Protection register area
0x0000_0004 - 0x0000_007C	Reserved				
0x0000_0080	SYSC0_RUNPDCFGR 00000011_11110011_00010001_00000000				RUN profile register area
0x0000_0084	SYSC0_RUNCKSRER 00000000_00000000_00000000_0000*111				
0x0000_0088	SYSC0_RUNCKSELR 00000000_00000000_00000000_00000000				
0x0000_008C	SYSC0_RUNCKER 00000000_00000000_00000000_00000011				
0x0000_0090	SYSC0_RUNCKDIVR 00000000_00000000_00000000_00000000				
0x0000_0094	SYSC0_RUNPLL0CNTR 00000000_00001101_00000001_00000000				
0x0000_0098	SYSC0_RUNPLL1CNTR 00000000_00001101_00000001_00000000				
0x0000_009C	SYSC0_RUNPLL2CNTR 00000000_00001101_00000001_00000000				
0x0000_00A0	SYSC0_RUNPLL3CNTR 00000000_00001101_00000001_00000000				
0x0000_00A4	SYSC0_RUNSSCG0CNTR0 00000000_00001101_00000001_00000000				
0x0000_00A8	SYSC0_RUNSSCG0CNTR1 00000000_00000000_00000000_00101001				
0x0000_00AC	SYSC0_RUNSSCG1CNTR0 00000000_00001101_00000001_00000000				
0x0000_00B0	SYSC0_RUNSSCG1CNTR1 00000000_00000000_00000000_00101001				
0x0000_00B4	SYSC0_RUNSSCG2CNTR0 00000000_00001101_00000001_00000000				
0x0000_00B8	SYSC0_RUNSSCG2CNTR1 00000000_00000000_00000000_00101001				
0x0000_00BC	SYSC0_RUNSSCG3CNTR0 00000000_00001101_00000001_00000000				
0x0000_00C0	SYSC0_RUNSSCG3CNTR1 00000000_00000000_00000000_00101001				

Offset	+3	+2	+1	+0	Area
0x0000_00C4	SYSC0_RUNLVDCFGR 0*000***_0*000***_0*0*****_0*0*****				RUN profile register area
0x0000_00C8	SYSC0_RUNCSVCFGR 00000000_00000000_00000000_00000000				
0x0000_00CC	SYSC0_RUNREGCFGR 00000000_00000000_00000000_0000000*				
0x0000_00D0 - 0x0000_00F8	Reserved				
0x0000_00FC	SYSC0_TRGRUNCNTR 00000000_00000000_00000000_00000000				
0x0000_0100	SYSC0_PSSPDCFGR 00000011_11110011_00010001_00000000				PSS profile register area
0x0000_0104	SYSC0_PSSCKSRER 00000000_00000000_00000000_0000*111				
0x0000_0108	SYSC0_PSSCKSELR 00000000_00000000_00000000_00000000				
0x0000_010C	SYSC0_PSSCKER 00000000_00000000_00000000_00000011				
0x0000_0110	SYSC0_PSSCKDIVR 00000000_00000000_00000000_00000000				
0x0000_0114	SYSC0_PSSPLL0CNTR 00000000_00001101_00000001_00000000				
0x0000_0118	SYSC0_PSSPLL1CNTR 00000000_00001101_00000001_00000000				
0x0000_011C	SYSC0_PSSPLL2CNTR 00000000_00001101_00000001_00000000				
0x0000_0120	SYSC0_PSSPLL3CNTR 00000000_00001101_00000001_00000000				
0x0000_0124	SYSC0_PSSSSCG0CNTR0 00000000_00001101_00000001_00000000				
0x0000_0128	SYSC0_PSSSSCG0CNTR1 00000000_00000000_00000000_00101001				
0x0000_012C	SYSC0_PSSSSCG1CNTR0 00000000_00001101_00000001_00000000				
0x0000_0130	SYSC0_PSSSSCG1CNTR1 00000000_00000000_00000000_00101001				
0x0000_0134	SYSC0_PSSSSCG2CNTR0 00000000_00001101_00000001_00000000				
0x0000_0138	SYSC0_PSSSSCG2CNTR1 00000000_00000000_00000000_00101001				
0x0000_013C	SYSC0_PSSSSCG3CNTR0 00000000_00001101_00000001_00000000				
0x0000_0140	SYSC0_PSSSSCG3CNTR1 00000000_00000000_00000000_00101001				

Offset	+3	+2	+1	+0	Area
0x0000_0144	SYSC0_PSSLVDCFGR 0*000***_0*000***_0*0*****_0*0*****				PSS profile register area
0x0000_0148	SYSC0_PSSCSVCFGR 00000000_00000000_00000000_00000000				
0x0000_014C	SYSC0_PSSREGCFGR 00000000_00000000_00000000_0000000*				
0x0000_0150 - 0x0000_0178	Reserved				
0x0000_017C	SYSC0_PSSENR 00000000_00000000_00000000_00000000				
0x0000_0180	SYSC0_APPPDCFGR 00000011_11110011_00010001_00000000				APP profile register area
0x0000_0184	SYSC0_APPCKSRER 00000000_00000000_00000000_0000*111				
0x0000_0188	SYSC0_APPCKSELR 00000000_00000000_00000000_00000000				
0x0000_018C	SYSC0_APPCKER 00000000_00000000_00000000_00000011				
0x0000_0190	SYSC0_APPCKDIVR 00000000_00000000_00000000_00000000				
0x0000_0194	SYSC0_APPPLL0CNTR 00000000_00001101_00000001_00000000				
0x0000_0198	SYSC0_APPPLL1CNTR 00000000_00001101_00000001_00000000				
0x0000_019C	SYSC0_APPPLL2CNTR 00000000_00001101_00000001_00000000				
0x0000_01A0	SYSC0_APPPLL3CNTR 00000000_00001101_00000001_00000000				
0x0000_01A4	SYSC0_APPSSCG0CNTR0 00000000_00001101_00000001_00000000				
0x0000_01A8	SYSC0_APPSSCG0CNTR1 00000000_00000000_00000000_00101001				
0x0000_01AC	SYSC0_APPSSCG1CNTR0 00000000_00001101_00000001_00000000				
0x0000_01B0	SYSC0_APPSSCG1CNTR1 00000000_00000000_00000000_00101001				
0x0000_01B4	SYSC0_APPSSCG2CNTR0 00000000_00001101_00000001_00000000				
0x0000_01B8	SYSC0_APPSSCG2CNTR1 00000000_00000000_00000000_00101001				
0x0000_01BC	SYSC0_APPSSCG3CNTR0 00000000_00001101_00000001_00000000				
0x0000_01C0	SYSC0_APPSSCG3CNTR1 00000000_00000000_00000000_00101001				

Offset	+3	+2	+1	+0	Area
0x0000_01C4	SYSC0_APPLVDCFGR 0*000***_0*000***_0*0*****_0*0*****				APP profile register area
0x0000_01C8	SYSC0_APPCSVCFGR 00000000_00000000_00000000_00000000				
0x0000_01CC	SYSC0_APPREGCFGR 00000000_00000000_00000000_0000000*				
0x0000_01D0 - 0x0000_01FC	Reserved				
0x0000_0200	SYSC0_STSPDCFGR 00000011_11110011_00010001_00000000				STS profile register area
0x0000_0204	SYSC0_STSCKSRER 00000000_00000000_00000000_0000*111				
0x0000_0208	SYSC0_STSCKSELR 00000000_00000000_00000000_00000000				
0x0000_020C	SYSC0_STSCKER 00000000_00000000_00000000_00000011				
0x0000_0210	SYSC0_STSCKDIVR 00000000_00000000_00000000_00000000				
0x0000_0214	SYSC0_STSPLL0CNTR 00000000_00001101_00000001_00000000				
0x0000_0218	SYSC0_STSPLL1CNTR 00000000_00001101_00000001_00000000				
0x0000_021C	SYSC0_STSPLL2CNTR 00000000_00001101_00000001_00000000				
0x0000_0220	SYSC0_STSPLL3CNTR 00000000_00001101_00000001_00000000				
0x0000_0224	SYSC0_STSSSCG0CNTR0 00000000_00001101_00000001_00000000				
0x0000_0228	SYSC0_STSSSCG0CNTR1 00000000_00000000_00000000_00101001				
0x0000_022C	SYSC0_STSSSCG1CNTR0 00000000_00001101_00000001_00000000				
0x0000_0230	SYSC0_STSSSCG1CNTR1 00000000_00000000_00000000_00101001				
0x0000_0234	SYSC0_STSSSCG2CNTR0 00000000_00001101_00000001_00000000				
0x0000_0238	SYSC0_STSSSCG2CNTR1 00000000_00000000_00000000_00101001				
0x0000_023C	SYSC0_STSSSCG3CNTR0 00000000_00001101_00000001_00000000				
0x0000_0240	SYSC0_STSSSCG3CNTR1 00000000_00000000_00000000_00101001				
0x0000_0244	SYSC0_STSLVDCFGR **000***_**000***_**0*****_**0*****				

Offset	+3	+2	+1	+0	Area
0x0000_0248	SYSC0_STSCSVCFGR 00000000_00000000_00000000_00000000				STS profile register area
0x0000_024C	SYSC0_STSREGCFGR 00000000_00000000_00000000_0000000*				
0x0000_0250 - 0x0000_027C	Reserved				
0x0000_0280	SYSC0_SYSIDR				System register area
0x0000_0284	SYSC0_SYSPFIDR				
0x0000_0288	SYSC0_SYSSTSR 00000000_00000000_00000000_00000001				
0x0000_028C	SYSC0_SYSINTER 00000000_00000000_00000000_00000000				
0x0000_0290	SYSC0_SYSICLR 00000000_00000000_00000000_00000000				
0x0000_0294	SYSC0_SYSERRIR0 00000000_00000000_00000000_00000000				
0x0000_0298	SYSC0_SYSERRIR1 00000000_00000000_00000000_00000000				
0x0000_029C	SYSC0_SYSERRICLR0 00000000_00000000_00000000_00000000				
0x0000_02A0	SYSC0_SYSERRICLR1 00000000_00000000_00000000_00000000				
0x0000_02A4	SYSC0_SYSPROSTSR 00000000_00000000_00000000_00000000				
0x0000_02A8	SYSC0_SYSRUNPEFR 00000000_00000000_00000000_00000000				
0x0000_02AC	SYSC0_SYSPSSPEFR 00000000_00000000_00000000_00000000				
0x0000_02B0 - 0x0000_02FC	Reserved				
0x0000_0680	SYSC0_SPECFGR 00000000_01100000_00000000_11101110				Special register area
0x0000_0684	Reserved				
0x0000_0688	Reserved				
0x0000_068C	Reserved				
0x0000_0690	Reserved				
0x0000_0694	Reserved				
0x0000_0698 - 0x0000_06FC	Reserved				
0x0000_0700	SYSC0_JTAGDETECT 00000000_00000000_00000000_00000000				Debug register area
0x0000_0704	SYSC0_JTAGCNFG 00000000_00000000_00000000_00000001				
0x0000_0708	SYSC0_JTAGWAKEUP 00000000_00000000_00000000_00000000				
0x0000_070C - 0x0000_07FC	Reserved				

Register Map (MCU CONFIG)

Table 5-16 shows a register address map.

Table 5-16 Low Power Consumption Register Address Map (MCU CONFIG)

Offset	+3	+2	+1	+0	Area
0x0000_0000	MCG_IRSR0 00000000_00000000_00000000_00000000				MCU CONFIG register area
0x0000_0004	MCG_IRSR1 00000000_00000000_00000000_00000000				
0x0000_0008	MCG_IRSR2 00000000_00000000_00000000_00000000				
0x0000_000C	MCG_IRSR3 00000000_00000000_00000000_00000000				
0x0000_0010	MCG_IRSR4 00000000_00000000_00000000_00000000				

Register Map (SYSC1)

Table 5-17 shows a register address map.

Table 5-17 Low Power Consumption Register Address Map (SYSC1)

Offset	+3	+2	+1	+0	Area
0x0000_0000	SYSC1_PROTKEYR 00000000_00000000_00000000_00000000				Protection register area
0x0000_0004 - 0x0000_007C	Reserved				
0x0000_0080	SYSC1_RUNCKSELR0 00000000_00000000_00000000_00000000				RUN profile register area
0x0000_0084	SYSC1_RUNCKSELR1 00000000_00000000_00000000_00000000				
0x0000_0088	SYSC1_RUNCKSELR2 00000000_00000000_00000000_00000000				
0x0000_008C	SYSC1_RUNCKER0 11111111_10111111_01001111_11110001				
0x0000_0090	SYSC1_RUNCKER1 00011111_00011111_00011111_00000001				
0x0000_0094	SYSC1_RUNCKER2 00000000_00000000_00011111_00011111				
0x0000_0098	SYSC1_RUNCKDIVR0 00000000_00000000_00000000_00000000				
0x0000_009C	SYSC1_RUNCKDIVR1 00000000_00000000_00000000_00000000				
0x0000_00A0	SYSC1_RUNCKDIVR2 00000000_00000000_00000000_00000000				RUN profile register area
0x0000_00A4	SYSC1_RUNCKDIVR3 00000000_00000000_00000000_00000000				
0x0000_00A8	SYSC1_RUNCKDIVR4 00000000_00000000_00000000_00000000				
0x0000_00AC	SYSC1_RUNCKDIVR5 00000000_00000000_00000000_00000000				
0x0000_00B0	SYSC1_RUNCKDIVR6 00000000_00000000_00000000_00000000				
0x0000_00B4	SYSC1_RUNCKDIVR7 00000000_00000000_00000000_00000000				
0x0000_00B8	SYSC1_RUNCKDIVR8 00000000_00000000_00000000_00000000				

Offset	+3	+2	+1	+0	Area
0x0000_00BC	SYSC1_RUNCKDIVR9 00000000_00000000_00000000_00000000				RUN profile register area
0x0000_00C0 - 0x0000_00F8	Reserved				
0x0000_00FC	SYSC1_RUNENR 00000000_00000000_00000000_00000000				
0x0000_0100	SYSC1_PSSCKSELR0 00000000_00000000_00000000_00000000				PSS profile register area
0x0000_0104	SYSC1_PSSCKSELR1 00000000_00000000_00000000_00000000				
0x0000_0108	SYSC1_PSSCKSELR2 00000000_00000000_00000000_00000000				
0x0000_010C	SYSC1_PSSCKER0 11111111_10111111_01001111_11110000				
0x0000_0110	SYSC1_PSSCKER1 00011111_00011111_00011111_00000001				
0x0000_0114	SYSC1_PSSCKER2 00000000_00000000_00011111_00011111				
0x0000_0118	SYSC1_PSSCKDIVR0 00000000_00000000_00000000_00000000				
0x0000_011C	SYSC1_PSSCKDIVR1 00000000_00000000_00000000_00000000				
0x0000_0120	SYSC1_PSSCKDIVR2 00000000_00000000_00000000_00000000				
0x0000_0124	SYSC1_PSSCKDIVR3 00000000_00000000_00000000_00000000				
0x0000_0128	SYSC1_PSSCKDIVR4 00000000_00000000_00000000_00000000				
0x0000_012C	SYSC1_PSSCKDIVR5 00000000_00000000_00000000_00000000				
0x0000_0130	SYSC1_PSSCKDIVR6 00000000_00000000_00000000_00000000				
0x0000_0134	SYSC1_PSSCKDIVR7 00000000_00000000_00000000_00000000				
0x0000_0138	SYSC1_PSSCKDIVR8 00000000_00000000_00000000_00000000				

Offset	+3	+2	+1	+0	Area
0x0000_013C	SYSC1_PSSCKDIVR9 00000000_00000000_00000000_00000000				PSS profile register area
0x0000_0140 - 0x0000_0178	Reserved				
0x0000_017C	SYSC1_PSSENR 00000000_00000000_00000000_00000000				
0x0000_0180	SYSC1_APPCKSELR0 00000000_00000000_00000000_00000000				APP profile register area
0x0000_0184	SYSC1_APPCKSELR1 00000000_00000000_00000000_00000000				
0x0000_0188	SYSC1_APPCKSELR2 00000000_00000000_00000000_00000000				
0x0000_018C	SYSC1_APPCKER0 11111111_10111111_01001111_11110001				
0x0000_0190	SYSC1_APPCKER1 00011111_00011111_00011111_00000001				
0x0000_0194	SYSC1_APPCKER2 00000000_00000000_00011111_00011111				
0x0000_0198	SYSC1_APPCKDIVR0 00000000_00000000_00000000_00000000				
0x0000_019C	SYSC1_APPCKDIVR1 00000000_00000000_00000000_00000000				
0x0000_01A0	SYSC1_APPCKDIVR2 00000000_00000000_00000000_00000000				
0x0000_01A4	SYSC1_APPCKDIVR3 00000000_00000000_00000000_00000000				
0x0000_01A8	SYSC1_APPCKDIVR4 00000000_00000000_00000000_00000000				
0x0000_01AC	SYSC1_APPCKDIVR5 00000000_00000000_00000000_00000000				
0x0000_01B0	SYSC1_APPCKDIVR6 00000000_00000000_00000000_00000000				
0x0000_01B4	SYSC1_APPCKDIVR7 00000000_00000000_00000000_00000000				
0x0000_01B8	SYSC1_APPCKDIVR8 00000000_00000000_00000000_00000000				

Offset	+3	+2	+1	+0	Area
0x0000_01BC	SYSC1_APPCKDIVR9 00000000_00000000_00000000_00000000				APP profile register area
0x0000_01C0 - 0x0000_01FC	Reserved				
0x0000_0200	SYSC1_STSCKSELR0 00000000_00000000_00000000_00000000				STS profile register area
0x0000_0204	SYSC1_STSCKSELR1 00000000_00000000_00000000_00000000				
0x0000_0208	SYSC1_STSCKSELR2 00000000_00000000_00000000_00000000				
0x0000_020C	SYSC1_STSCKER0 11111111_10111111_01001111_11110001				
0x0000_0210	SYSC1_STSCKER1 00011111_00011111_00011111_00000001				
0x0000_0214	SYSC1_STSCKER2 00000000_00000000_00011111_00011111				
0x0000_0218	SYSC1_STSCKDIVR0 00000000_00000000_00000000_00000000				
0x0000_021C	SYSC1_STSCKDIVR1 00000000_00000000_00000000_00000000				
0x0000_0220	SYSC1_STSCKDIVR2 00000000_00000000_00000000_00000000				
0x0000_0224	SYSC1_STSCKDIVR3 00000000_00000000_00000000_00000000				
0x0000_0228	SYSC1_STSCKDIVR4 00000000_00000000_00000000_00000000				
0x0000_022C	SYSC1_STSCKDIVR5 00000000_00000000_00000000_00000000				
0x0000_0230	SYSC1_STSCKDIVR6 00000000_00000000_00000000_00000000				
0x0000_0234	SYSC1_STSCKDIVR7 00000000_00000000_00000000_00000000				
0x0000_0238	SYSC1_STSCKDIVR8 00000000_00000000_00000000_00000000				
0x0000_023C	SYSC1_STSCKDIVR9 00000000_00000000_00000000_00000000				

Offset	+3	+2	+1	+0	Area
0x0000_0240 - 0x0000_027C	Reserved				STS profile register area

5.1. Protection Register Group (SYSC0)

The register in this group is a control register for accessing SYSC0 registers.

5.1.1. Protection Key Setting Register (SYSC0_PROTKEYR)

The SYSC0_PROTKEYR register has settings for unlocking the protection of SYSC0 registers.

bit	31	0
Field	PROTKEY	
R/W Attribute	R,W	
Protection Attribute	WP	
Initial Value	00000000_00000000_00000000_00000000	

[bit31:0] PROTKEY: Protection Unlock Setting Bits

These bits compose the register for unlocking protection keys.

Write

bit31:0	Description
0x5CAC_CE55	Unlock
Other than above	Invalid

Read

bit31:0	Description
0x0000_0000	Locked state
0xFFFF_FFFF	Unlocked state

Note:

- *This register permits word access only. Any other access is invalid.*

5.2. RUN Profile Register Group (SYSC0)

The registers in this group are RUN profile control setting registers.

5.2.1. RUN Power Domain Setting Register (SYSC0_RUNPDCFGR)

The SYSC0_RUNPDCFGR register sets power ON or OFF for each power domain.

bit	31	30	29	28	27	26	25	24
Field	Reserved						PD6_1 EN	PD6_0 EN
R/W Attribute	R0,WX						R1,WX	R1,WX
Protection Attribute	WPS							
Initial Value	000000						1	1

bit	23	22	21	20	19	18	17	16
Field	PD5_3 EN	PD5_2 EN	PD5_1 EN	PD5_0 EN	Reserved		PD4_1 EN	PD4_0 EN
R/W Attribute	R/W	R/W	R/W	R/W	R0,WX		R/W	R/W
Protection Attribute	WPS							
Initial Value	1	1	1	1	00		1	1

bit	15	14	13	12	11	10	9	8
Field	Reserved			PD3 EN	Reserved			PD2 EN
R/W Attribute	R0,WX			R1,WX	R0,WX			R1,WX
Protection Attribute	WPS							
Initial Value	000			1	000			1

bit	7	6	5	4	3	2	1	0
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	WPS							
Initial Value	00000000							

[bit31:26] Reserved: Reserved Bits

[bit25] PD6_1EN: Power Domain 6_1 Power Supply Control Bit

This bit controls power supply to power domain 6_1.

Bit	Description
0	-
1	Supply power.

[bit24] PD6_0 EN: Power Domain 6_0 Power Supply Control Bit

This bit controls power supply to power domain 6_0.

Bit	Description
0	-
1	Supply power.

[bit23] PD5_3 EN: Power Domain 5_3 Power Supply Control Bit

This bit controls power supply to power domain 5_3.

Bit	Description
0	Shut off power supply.
1	Supply power.

[bit22] PD5_2 EN: Power Domain 5_2 Power Supply Control Bit

This bit controls power supply to power domain 5_2.

Bit	Description
0	Shut off power supply.
1	Supply power.

[bit21] PD5_1 EN: Power Domain 5_1 Power Supply Control Bit

This bit controls power supply to power domain 5_1.

Bit	Description
0	Shut off power supply.
1	Supply power.

[bit20] PD5_0 EN: Power Domain 5_0 Power Supply Control Bit

This bit controls power supply to power domain 5_0.

Bit	Description
0	Shut off power supply.
1	Supply power.

[bit19:18] Reserved: Reserved Bits

[bit17] PD4_1 EN: Power Domain 4_1 Power Supply Control Bit

This bit controls power supply to power domain 4_1.

Bit	Description
0	Shut off power supply.
1	Supply power.

[bit16] PD4_0 EN: Power Domain 4_0 Power Supply Control Bit

This bit controls power supply to power domain 4_0.

Bit	Description
0	Shut off power supply.
1	Supply power.

[bit15:13] Reserved: Reserved Bits

[bit12] PD3 EN: Power Domain 3 Power Supply Control Bit

This bit controls power supply to power domain 3.

Bit	Description
0	-
1	Supply power.

[bit11:9] Reserved: Reserved Bits

[bit8] PD2 EN: Power Domain 2 Power Supply Control Bit

This bit controls power supply to power domain 2.

Bit	Description
0	-
1	Supply power.

[bit7:0] Reserved: Reserved Bits

5.2.2. RUN Clock Source Enable Register (SYSC0_RUNCKSRER)

The SYSC0_RUNCKSRER register sets whether to enable or disable oscillation of the source clock.

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	WPS							
Initial Value	00000000							

bit	23	22	21	20	19	18	17	16
Field	Reserved				SSCG3 EN	SSCG2 EN	SSCG1 EN	SSCG0 EN
R/W Attribute	R0,WX				R/W	R/W	R/W	R/W
Protection Attribute	WPS							
Initial Value	0000				0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	Reserved				PLL3EN	PLL2EN	PLL1EN	PLL0EN
R/W Attribute	R0,WX				R/W	R/W	R/W	R/W
Protection Attribute	WPS							
Initial Value	0000				0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	Reserved				SOSCEN	MOSC EN	SCROSC EN	CROSC EN
R/W Attribute	R0,WX				R/W	R/W	R1,WX	R1,WX
Protection Attribute	WPS							
Initial Value	0000				0	1	1	1

[bit31:20] Reserved: Reserved Bits

[bit19] SSCG3EN: SSCG PLL3 Clock Oscillation Enable Bit

This bit controls oscillation of the SSCG PLL3 clock.

Bit	Description
0	Disable oscillation of the SSCG PLL3 clock.
1	Enable oscillation of the SSCG PLL3 clock.

[bit18] SSCG2EN: SSCG PLL2 Clock Oscillation Enable Bit

This bit controls oscillation of the SSCG PLL2 clock.

Bit	Description
0	Disable oscillation of the SSCG PLL2 clock.
1	Enable oscillation of the SSCG PLL2 clock.

[bit17] SSCG1EN: SSCG PLL1 Clock Oscillation Enable Bit

This bit controls oscillation of the SSCG PLL1 clock.

Bit	Description
0	Disable oscillation of the SSCG PLL1 clock.
1	Enable oscillation of the SSCG PLL1 clock.

[bit16] SSCG0EN: SSCG PLL0 Clock Oscillation Enable Bit

This bit controls oscillation of the SSCG PLL0 clock.

Bit	Description
0	Disable oscillation of the SSCG PLL0 clock.
1	Enable oscillation of the SSCG PLL0 clock.

[bit15:12] Reserved: Reserved Bits

[bit11] PLL3EN: PLL3 Clock Oscillation Enable Bit

This bit controls oscillation of the PLL3 clock.

Bit	Description
0	Disable oscillation of the PLL3 clock.
1	Enable oscillation of the PLL3 clock.

[bit10] PLL2EN: PLL2 Clock Oscillation Enable Bit

This bit controls oscillation of the PLL2 clock.

Bit	Description
0	Disable oscillation of the PLL2 clock.
1	Enable oscillation of the PLL2 clock.

[bit9] PLL1EN: PLL1 Clock Oscillation Enable Bit

This bit controls oscillation of the PLL1 clock.

Bit	Description
0	Disable oscillation of the PLL1 clock.
1	Enable oscillation of the PLL1 clock.

[bit8] PLL0EN: PLL0 Clock Oscillation Enable Bit

This bit controls oscillation of the PLL0 clock.

Bit	Description
0	Disable oscillation of the PLL0 clock.
1	Enable oscillation of the PLL0 clock.

[bit7:4] Reserved: Reserved Bits

[bit3] SOSCEN: Sub Clock Oscillation Enable Bit

This bit controls oscillation of the sub clock.

Bit	Description
0	Disable oscillation of the sub clock.
1	Enable oscillation of the sub clock.

[bit2] MOSCEN: Main Clock Oscillation Enable Bit

This bit controls oscillation of the main clock.

Bit	Description
0	Disable oscillation of the main clock.
1	Enable oscillation of the main clock.

[bit1] SCROSCEN: Low-Speed CR Clock Oscillation Enable Bit

This bit controls oscillation of the low-speed CR clock.

Bit	Description
0	-
1	Enable oscillation of the low-speed CR clock.

[bit0] CROSCEN: High-Speed CR Clock Oscillation Enable Bit

This bit controls oscillation of the high-speed CR clock.

Bit	Description
0	-
1	Enable oscillation of the high-speed CR clock.

5.2.3. RUN Clock Selection Register (SYSC0_RUNCKSELR)

The SYSC0_RUNCKSELR register sets the clock source for the clock domain.

bit	31	8
Field	Reserved	
R/W Attribute	R0,WX	
Protection Attribute	WPS	
Initial Value	00000000_00000000_00000000	

bit	7	6	5	4	3	2	1	0
Field	Reserved					CDMCUCCSL		
R/W Attribute	R0,WX					R/W		
Protection Attribute	WPS							
Initial Value	00000					000		

[bit31:3] Reserved: Reserved Bits

[bit2:0] CDMCUCCSL: Clock Domain MCUC Clock Selection Bits

These bits select the source clock for the clock domain MCUC.

bit2:0	Description
000	High-speed CR clock
001	Low-speed CR clock
010	Main clock
011	Sub clock (Do not use this setting)
100	PLL0 clock
101	SSCG PLL0 clock
110	Setting Prohibited
111	Clock fixed at "L"

5.2.4. RUN Clock Enable Register (SYSC0_RUNCKER)

The SYSC0_RUNCKER register sets whether to stop or enable an internal clock.

bit	31	8
Field	Reserved	
R/W Attribute	R0,WX	
Protection Attribute	WPS	
Initial Value	00000000_00000000_00000000	

bit	7	6	5	4	3	2	1	0
Field	Reserved						ENCLKMCUCP	ENCLKMCUCH
R/W Attribute	R0,WX						R1,WX	R1,WX
Protection Attribute	WPS							
Initial Value	000000						1	1

[bit31:2] Reserved: Reserved Bits

[bit1] ENCLKMCUCP: MCUconfig APB Clock Oscillation Enable Bit

This bit sets whether to enable or disable oscillation of the APB clock.

Bit	Description
0	-
1	Enable clock oscillation.

[bit0] ENCLKMCUCH: MCUconfig AHB Clock Oscillation Enable Bit

This bit sets whether to enable or disable oscillation of the MCUconfig AHB clock.

Bit	Description
0	-
1	Enable clock oscillation.

5.2.5. RUN Clock Divider Register (SYSC0_RUNCKDIVR)

The SYSC0_RUNCKDIVR register sets the division ratio of the clock.

bit	31															16
Field	Reserved															
R/W Attribute	R0,WX															
Protection Attribute	WPS															
Initial Value	00000000_00000000															

bit	15	14	13	12	11	10	9	8
Field	Reserved				MCUCPDIV			
R/W Attribute	R0,WX				R/W			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	7	6	5	4	3	2	1	0
Field	Reserved			MCUCHDIV				
R/W Attribute	R0,WX			R/W				
Protection Attribute	WPS							
Initial Value	000			00000				

[bit31:12] Reserved: Reserved Bits

[bit11:8] MCUCPDIV: MCUconfig APB Clock Divider Setting Bits

These bits set the division ratio of the PCOM clock from the HCOM clock.

bit11:8	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

[bit7:5] Reserved: Reserved Bits

[bit4:0] MCUCHDIV: MCUconfig AHB Clock Divider Setting Bits

These bits set the division ratio from the source clock of the clock domain MCUC.

bit4:0	Description
0_0000	No division
0_0001	Divided by 2
0_0010	Divided by 3
...	...
11101	Divided by 30
11110	Divided by 31
11111	Divided by 32

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

5.2.6. RUN PLLx Control Register (SYSC0_RUNPLLxCNTR)

The SYSC0_RUNPLLxCNTR register sets the division, multiplication rate, etc. for PLLx.

bit	31	30	29	28	27	26	25	24
Field	PLLx ISEL	Reserved						
R/W Attribute	R/W	R0,WX						
Protection Attribute	WPS							
Initial Value	0	0000000						

bit	23	22	21	20	19	18	17	16
Field	PLLxDIVN							
R/W Attribute	R/W							
Protection Attribute	WPS							
Initial Value	00001101							

bit	15	14	13	12	11	10	9	8
Field	Reserved				PLLxDIVM			
R/W Attribute	R0,WX				R/W			
Protection Attribute	WPS							
Initial Value	0000				0001			

bit	7	6	5	4	3	2	1	0
Field	Reserved						PLLxDIVL	
R/W Attribute	R0,WX						R/W	
Protection Attribute	WPS							
Initial Value	000000						00	

[bit31] PLLxISEL: PLLx Input Clock Selection Bit

This bit sets the clock to supply to PLLx.

Bit	Description
0	Main clock
1	High-speed CR clock

[bit30:24] Reserved: Reserved Bits

[bit23:16] PLLxDIVN: PLLx Clock N-multiplier Setting Bits

These bits set the multiplication rate of the PLLx clock.

bit23:16	Description
0000_0000	Setting prohibited
...	...
0000_1100	Setting prohibited
0000_1101	Multiply by 13
0000_1110	Multiply by 14
...	...
1100_0110	Multiply by 198
1100_0111	Multiply by 199
1100_1000	Multiply by 200
1100_1001	Setting prohibited
...	...
1100_1111	Setting prohibited

[bit15:12] Reserved: Reserved Bits

[bit11:8] PLLxDIVM: PLLx Clock M-Divider Setting Bits

These bits set the division ratio for PLLx clock output.

bit11:8	Description
000x	Divided by 2
0010	Divided by 4
0011	Divided by 6
0100	Divided by 8
0101	Divided by 10
0110	Divided by 12
0111	Divided by 14
1000	Divided by 16
1001	Divided by 18
1010	Divided by 20
1011	Divided by 22
1010	Divided by 24
1101	Divided by 26
1110	Divided by 28
1111	Divided by 30

[bit7:2] Reserved: Reserved Bits

[bit1:0] PLLxDIVL: PLLx Input Clock Divider Setting Bits

These bits set the division ratio of the PLLx input clock.

bit1:0	Description
00	No division
01	Divided by 2
10	Divided by 4
11	Divided by 6

5.2.7. RUN SSCGx Control Register 0 (SYSC0_RUNSSCGxCNTR0)

The SYSC0_RUNSSCGxCNTR0 register sets the division, multiplication rate, etc. for SSCG PLLx.

bit	31	30	29	28	27	26	25	24
Field	SSCGx ISEL	Reserved						
R/W Attribute	R/W	R0,WX						
Protection Attribute	WPS							
Initial Value	0	0000000						

bit	23	22	21	20	19	18	17	16
Field	SSCGxDIVN							
R/W Attribute	R/W							
Protection Attribute	WPS							
Initial Value	00001101							

bit	15	14	13	12	11	10	9	8
Field	Reserved				SSCGxDIVM			
R/W Attribute	R0,WX				R/W			
Protection Attribute	WPS							
Initial Value	0000				0001			

bit	7	6	5	4	3	2	1	0
Field	Reserved						SSCGxDIVL	
R/W Attribute	R0,WX						R/W	
Protection Attribute	WPS							
Initial Value	000000						00	

[bit31] SSCGxISEL: SSCG PLLx Input Clock Selection Bit

This bit sets the clock to supply to SSCG PLLx.

Bit	Description
0	Main clock
1	High-speed CR clock

[bit30:24] Reserved: Reserved Bits

[bit23:16] SSCGxDIVN: SSCG PLLx Clock N-multiplier Setting Bits

These bits set the N multiplication rate of the SSCG PLLx clock.

bit23:16	Description
0000_0000	Setting prohibited
...	...
0000_1100	Setting prohibited
0000_1101	Multiply by 13
0000_1110	Multiply by 14
...	...
1100_0110	Multiply by 198
1100_0111	Multiply by 199
1100_1000	Multiply by 200
1100_1001	Setting prohibited
...	...
1100_1111	Setting prohibited

[bit15:12] Reserved: Reserved Bits

[bit11:8] SSCGxDIVM: SSCG PLLx Clock M-Division Setting Bits

These bits set the division ratio for SSCG PLLx clock output.

bit11:8	Description
000x	Divided by 2
0010	Divided by 4
0011	Divided by 6
0100	Divided by 8
0101	Divided by 10
0110	Divided by 12
0111	Divided by 14
1000	Divided by 16
1001	Divided by 18
1010	Divided by 20
1011	Divided by 22
1010	Divided by 24
1101	Divided by 26
1110	Divided by 28
1111	Divided by 30

[bit7:2] Reserved: Reserved Bits

[bit1:0] SSCGxDIVL: SSCG PLLx Input Clock Divider Setting Bits

These bits set the division ratio of the SSCG PLLx input clock.

bit1:0	Description
00	No division
01	Divided by 2
10	Divided by 4
11	Divided by 6

5.2.8. RUN SSCGx Control Register 1 (SYSC0_RUNSSCGxCNTR1)

The SYSC0_RUNSSCGxCNTR1 register sets modulation enable, the modulation mode, etc. for SSCG PLLx.

bit	31	30	29	28	27	26	25	24
Field	Reserved							SSCGx SSEN
R/W Attribute	R0,WX							R/W
Protection Attribute	WPS							
Initial Value	0000000							0

bit	23	22	21	20	19	18	17	16
Field	Reserved					SSCGxFREQ		SSCGx MODE
R/W Attribute	R0,WX					R/W		R/W
Protection Attribute	WPS							
Initial Value	00000					00		0

bit	15	14	13	12	11	10	9	8
Field	Reserved						SSCGxRATE	
R/W Attribute	R0,WX						R/W	
Protection Attribute	WPS							
Initial Value	000000						00	

bit	7	6	5	4	3	2	1	0
Field	SSCGxRATE							
R/W Attribute	R/W							
Protection Attribute	WPS							
Initial Value	00101001							

[bit31:25] Reserved: Reserved Bits

[bit24] SSCGxSSEN: SSCG PLLx modulation Enable Setting Bit

This bit controls SSCG PLLx modulation.

Bit	Description
0	Disable modulation.
1	Enable modulation.

[bit23:19] Reserved: Reserved Bits

[bit18:17] SSCGxFREQ: SSCG PLLx Clock modulation Frequency Selection Bits

These bits set the modulation frequency of the SSCG PLLx clock.

bit18:17	Description
00	Fmod = (1/1024) x Fin
01	Fmod = (1/2048) x Fin
10 or 11	Fmod = (1/4096) x Fin

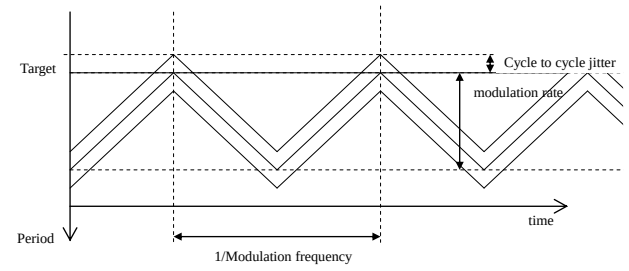
* Fin: SSCG reference clock frequency, Fmod: Modulation frequency

[bit16] SSCGxMODE: SSCG PLLx modulation Mode Setting Bit

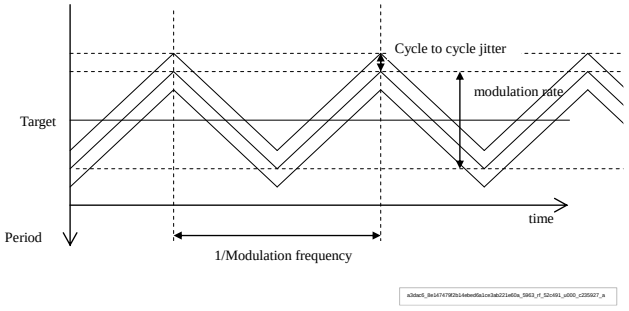
This bit sets the modulation mode of the SSCG PLLx clock.

Bit	Description
0	Set Down spread mode.
1	Set Center spread mode.

- Down spread



- Center spread



[bit15:10] Reserved: Reserved Bits

[bit9:0] SSCGxRATE: SSCG PLLx Clock modulation Ratio Control Bits

These bits set the modulation ratio of the SSCG PLLx clock.

bit9:0	Description
00_0010_1001	Modulation ratio of 0.5%
00_0101_0010	Modulation ratio of 1.0%
00_1010_0100	Modulation ratio of 2.0%
00_1111_0110	Modulation ratio of 3.0%
01_0100_1000	Modulation ratio of 4.0%
01_1001_1010	Modulation ratio of 5.0%
Others	Setting prohibited

5.2.9. RUN Low-Voltage Detection Setting Register (SYSC0_RUNLVDCFGR)

The SYSC0_RUNLVDCFGR register sets each internal low-voltage detection.

bit	31	30	29	28	27	26	25	24
Field	Reserved	LVDL1S	Reserved			LVDL1V		LVDL1E
R/W Attribute	R0,WX	(product specification)	R0,WX			(product specification)		(product specification)
Protection Attribute	WPS							
Initial Value	0	(product specification)	000			(product specification)		(product specification)

bit	23	22	21	20	19	18	17	16
Field	Reserved	LVDL2S	Reserved			LVDL2V		LVDL2E
R/W Attribute	R0,WX	(product specification)	R0,WX			(product specification)		(product specification)
Protection Attribute	WPS							
Initial Value	0	(product specification)	000			(product specification)		(product specification)

bit	15	14	13	12	11	10	9	8
Field	Reserved	LVDH1S	Reserved	LVDH1V				LVDH1E
R/W Attribute	R0,WX	(product specification)	R0,WX	(product specification)				(product specification)
Protection Attribute	WPS							
Initial Value	0	(product specification)	0	(product specification)				(product specification)

bit	7	6	5	4	3	2	1	0
Field	Reserved	LVDH2S	Reserved	LVDH2V				LVDH2E
R/W Attribute	R0,WX	(product specification)	R0,WX	(product specification)				(product specification)
Protection Attribute	WPS							
Initial Value	0	(product specification)	0	(product specification)				(product specification)

[bit31] Reserved: Reserved Bit

[bit30] LVDL1S: Internal Low-Voltage Detection Operation Selection Bit

This bit selects an operation for the internal low-voltage detection.

Bit	Description
0	Reset.
1	Interrupt.

[bit29:27] Reserved: Reserved Bits

[bit26:25] LVDL1V: Internal Low-Voltage Detection Voltage Setting Bits

These bits set the detection voltage for internal low-voltage detection.

bit18:17	Description
00	(product specification)
01	(product specification)
10	(product specification)
11	(product specification)

Note:

- *After the detection voltage is changed, low-voltage detection is unavailable until the low-voltage detection stability time elapses.*

[bit24] LVDL1E: Internal Low-Voltage Detection Operation Enable Bit

This bit enables the internal low-voltage detection operation.

Bit	Description
0	Stop operation.
1	Enable operation.

Note:

- *After this setting is changed from "0" (stop operation) to "1" (enable operation), low-voltage detection is unavailable until the low-voltage detection stability time elapses.*

[bit23] Reserved: Reserved Bit

[bit22] LVDL2S: Extended Internal Low-Voltage Detection Operation Selection Bit

This bit selects an operation for the extended internal low-voltage detection.

Bit	Description
0	Reset.
1	Interrupt.

[bit21:19] Reserved: Reserved Bits

[bit18:17] LVDL2V: Extended Internal Low-Voltage Detection Voltage Setting Bits

These bits set the detection voltage for extended internal low-voltage detection.

Bit	Description
00	(product specification)
01	(product specification)
10	(product specification)
11	(product specification)

Note:

- *After the detection voltage is changed, low-voltage detection is unavailable until the low-voltage detection stability time elapses.*

[bit16] LVDL2E: Extended Internal Low-Voltage Detection Operation Enable Bit

This bit enables the extended internal low-voltage detection operation.

Bit	Description
0	Stop operation.
1	Enable operation.

Note:

- *After this setting is changed from "0" (stop operation) to "1" (enable operation), low-voltage detection is unavailable until the low-voltage detection stability time elapses.*

[bit15] Reserved: Reserved Bit

[bit14] LVDH1S: External Low-Voltage Detection Operation Selection Bit

This bit selects an operation for the external low-voltage detection.

Bit	Description
0	Reset.
1	Interrupt.

[bit13] Reserved: Reserved Bit

[bit12:9] LVDH1V: External Low-Voltage Detection Voltage Setting Bits

These bits set the detection voltage for external low-voltage detection.

Bit	Description
0000	(product specification)
0001	(product specification)
0010	(product specification)
0011	(product specification)
0100	(product specification)
0101	(product specification)
0110	(product specification)
0111	(product specification)
1xxx	(product specification)

Note:

- *After the detection voltage is changed, low-voltage detection is unavailable until the low-voltage detection stability time elapses.*

[bit8] LVDH1E: External Low-Voltage Detection Operation Enable Bit

This bit sets the external low-voltage detection operation.

Bit	Description
0	Stop operation.
1	Enable operation.

Note:

- *After this setting is changed from "0" (stop operation) to "1" (enable operation), low-voltage detection is unavailable until the low-voltage detection stability time elapses.*

[bit7] Reserved: Reserved Bit

[bit6] LVDH2S: Extended External Low-Voltage Detection Operation Selection Bit

This bit selects an operation for the extended external low-voltage detection.

Bit	Description
0	Reset.
1	Interrupt.

[bit5] Reserved: Reserved Bit

[bit4:1] LVDH2V: Extended External Low-Voltage Detection Voltage Setting Bits

These bits set the detection voltage for extended external low-voltage detection.

Bit	Description
0000	(product specification)
0001	(product specification)
0010	(product specification)
0011	(product specification)
0100	(product specification)
0101	(product specification)
0110	(product specification)
0111	(product specification)
1xxx	(product specification)

Note:

- *After the detection voltage is changed, low-voltage detection is unavailable until the low-voltage detection stability time elapses.*

[bit0] LVDH2E: Extended External Low-Voltage Detection Operation Enable Bit

This bit sets the extended external low-voltage detection operation.

Bit	Description
0	Stop operation.
1	Enable operation.

Note:

- *After this setting is changed from "0" (stop operation) to "1" (enable operation), low-voltage detection is unavailable until the low-voltage detection stability time elapses.*

5.2.10. RUN Clock Supervisor Setting Register (SYSC0_RUNCSVCFGR)

The SYSC0_RUNCSVCFGR register sets whether to enable or disable operation of each clock supervisor.

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	WPS							
Initial Value	00000000							

bit	23	22	21	20	19	18	17	16
Field	Reserved				SSCG3 CSVE	SSCG2 CSVE	SSCG1 CSVE	SSCG0 CSVE
R/W Attribute	R0,WX				R/W	R/W	R/W	R/W
Protection Attribute	WPS							
Initial Value	0000				0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	Reserved				PLL3 CSVE	PLL2 CSVE	PLL1 CSVE	PLL0 CSVE
R/W Attribute	R0,WX				R/W	R/W	R/W	R/W
Protection Attribute	WPS							
Initial Value	0000				0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	Reserved				SCR CSVE	FCR CSVE	SO CSVE	MO CSVE
R/W Attribute	R0,WX				R/W	R/W	R/W	R/W
Protection Attribute	WPS							
Initial Value	0000				0	0	0	0

[bit31:20] Reserved: Reserved Bits

[bit19] SSCG3CSVE: SSCG PLL3 Clock Supervisor Enable Bit

This bit sets whether to enable or disable the SSCG PLL3 clock supervisor.

Bit	Description
0	Disable the clock supervisor.
1	Enable the clock supervisor.

[bit18] SSCG2CSVE: SSCG PLL2 Clock Supervisor Enable Bit

This bit sets whether to enable or disable the SSCG PLL2 clock supervisor.

Bit	Description
0	Disable the clock supervisor.
1	Enable the clock supervisor.

[bit17] SSCG1CSVE: SSCG PLL1 Clock Supervisor Enable Bit

This bit sets whether to enable or disable the SSCG PLL1 clock supervisor.

Bit	Description
0	Disable the clock supervisor.
1	Enable the clock supervisor.

[bit16] SSCG0CSVE: SSCG PLL0 Clock Supervisor Enable Bit

This bit sets whether to enable or disable the SSCG PLL0 clock supervisor.

Bit	Description
0	Disable the clock supervisor.
1	Enable the clock supervisor.

[bit15:12] Reserved: Reserved Bits

[bit11] PLL3CSVE: PLL3 Clock Supervisor Enable Bit

This bit sets whether to enable or disable the PLL3 clock supervisor.

Bit	Description
0	Disable the clock supervisor.
1	Enable the clock supervisor.

[bit10] PLL2CSVE: PLL2 Clock Supervisor Enable Bit

This bit sets whether to enable or disable the PLL2 clock supervisor.

Bit	Description
0	Disable the clock supervisor.
1	Enable the clock supervisor.

[bit9] PLL1CSVE: PLL1 Clock Supervisor Enable Bit

This bit sets whether to enable or disable the PLL1 clock supervisor.

Bit	Description
0	Disable the clock supervisor.
1	Enable the clock supervisor.

[bit8] PLL0CSVE: PLL0 Clock Supervisor Enable Bit

This bit sets whether to enable or disable the PLL0 clock supervisor.

Bit	Description
0	Disable the clock supervisor.
1	Enable the clock supervisor.

[bit7:4] Reserved：Reserved Bits

[bit3] SCRCSVE：Low-Speed CR Clock Supervisor Enable Bit

This bit sets whether to enable or disable the Low-speed CR clock supervisor.

Bit	Description
0	Disable the clock supervisor.
1	Enable the clock supervisor.

[bit2] FCRCSVE：High-Speed CR Clock Supervisor Enable Bit

This bit sets whether to enable or disable the High-speed CR clock supervisor.

Bit	Description
0	Disable the clock supervisor.
1	Enable the clock supervisor.

[bit1] SOCSVE: Sub Clock Supervisor Enable Bit

This bit sets whether to enable or disable the sub clock supervisor.

Bit	Description
0	Disable the clock supervisor.
1	Enable the clock supervisor.

[bit0] MOCSVE: Main Clock Supervisor Enable Bit

This bit sets whether to enable or disable the main clock supervisor.

Bit	Description
0	Disable the clock supervisor.
1	Enable the clock supervisor.

5.2.11. RUN Regulator Setting Register (SYSC0_RUNREGCFGR)

The SYSC0_RUNREGCFGR register sets the regulator mode and output voltage.

bit	31	8
Field	Reserved	
R/W Attribute	R0,WX	
Protection Attribute	WPS	
Initial Value	00000000_00000000_00000000	

bit	7	6	5	4	3	2	1	0
Field	RMSEL	Reserved						RVSEL
R/W Attribute	R0,WX	R0,WX						R/W
Protection Attribute	WPS							
Initial Value	0	000000						(product specification)

[bit31:8] Reserved: Reserved Bits

[bit7] RMSEL: Regulator Mode Setting Bit

This bit indicates the regulator mode setting.

Bit	Description
0	Main mode
1	-

[bit6:1] Reserved: Reserved Bits

[bit0] RVSEL: Regulator Output Voltage Setting Bits

These bits set the regulator output voltage.

Bit	Description
0	1.20V
1	1.25V

5.2.12. RUN Profile Update Trigger Register (SYSC0_TRGRUNCNTR)

The SYSC0_TRGRUNCNTR register is a RUN profile update trigger.

bit	31	8
Field	Reserved	
R/W Attribute	R0,WX	
Protection Attribute	WPS	
Initial Value	00000000_00000000_00000000	

bit	7	6	5	4	3	2	1	0
Field	APPLY_RUN							
R/W Attribute	R0,W							
Protection Attribute	WPS							
Initial Value	00000000							

[bit31:8] Reserved: Reserved Bits

[bit7:0] APPLY_RUN: RUN profile Update Trigger Setting Bits

These bits compose the RUN profile update start register.

bit7:0	Description
During write operation	When "0xAB" is written: Start updating the RUN profile. When other than "0xAB" is written: Invalid
During read operation	"0x00" is always read.

Notes:

- *If the value written in this register is invalid, an error flag (SYSC0_SYSERRIR1:RUNTRGERRIF is "1") is enabled.*
- *If the RUN profile is updated, it cannot be re-updated until that the update process completes. If "0xAB" is written in this register during a RUN profile update, an error flag (SYSC0_SYSERRIR1:TRGERRIF is "1") is enabled.*

5.3. PSS Profile Register Group (SYSC0)

The registers in this group are PSS profile control setting registers.

5.3.1. PSS Power Domain Setting Register (SYSC0_PSSPDCFGR)

The SYSC0_PSSPDCFGR register sets power ON or OFF for each power domain.

bit	31	30	29	28	27	26	25	24
Field	Reserved						PD6_1 EN	PD6_0 EN
R/W Attribute	R0,WX						R/W	R/W
Protection Attribute	WPS							
Initial Value	000000						1	1

bit	23	22	21	20	19	18	17	16
Field	PD5_3 EN	PD5_2 EN	PD5_1 EN	PD5_0 EN	Reserved		PD4_1 EN	PD4_0 EN
R/W Attribute	R/W	R/W	R/W	R/W	R0,WX		R/W	R/W
Protection Attribute	WPS							
Initial Value	1	1	1	1	00		1	1

bit	15	14	13	12	11	10	9	8
Field	Reserved			PD3 EN	Reserved			PD2 EN
R/W Attribute	R0,WX			R/W	R0,WX			R/W
Protection Attribute	WPS							
Initial Value	000			1	000			1

bit	7	6	5	4	3	2	1	0
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	WPS							
Initial Value	00000000							

[bit31:26] Reserved: Reserved Bits

[bit25] PD6_1EN: Power Domain 6_1 Power Supply Control Bit

This bit controls power supply to power domain 6_1.

Bit	Description
0	Shut off power supply.
1	Supply power.

[bit24] PD6_0 EN: Power Domain 6_0 Power Supply Control Bit

This bit controls power supply to power domain 6_0.

Bit	Description
0	Shut off power supply.
1	Supply power.

[bit23] PD5_3 EN: Power Domain 5_3 Power Supply Control Bit

This bit controls power supply to power domain 5_3.

Bit	Description
0	Shut off power supply.
1	Supply power.

[bit22] PD5_2 EN: Power Domain 5_2 Power Supply Control Bit

This bit controls power supply to power domain 5_2.

Bit	Description
0	Shut off power supply.
1	Supply power.

[bit21] PD5_1 EN: Power Domain 5_1 Power Supply Control Bit

This bit controls power supply to power domain 5_1.

Bit	Description
0	Shut off power supply.
1	Supply power.

[bit20] PD5_0 EN: Power Domain 5_0 Power Supply Control Bit

This bit controls power supply to power domain 5_0.

Bit	Description
0	Shut off power supply.
1	Supply power.

[bit19:18] Reserved: Reserved Bits

[bit17] PD4_1 EN: Power Domain 4_1 Power Supply Control Bit

This bit controls power supply to power domain 4_1.

Bit	Description
0	Shut off power supply.
1	Supply power.

[bit16] PD4_0 EN: Power Domain 4_0 Power Supply Control Bit

This bit controls power supply to power domain 4_0.

Bit	Description
0	Shut off power supply.
1	Supply power.

[bit15:13] Reserved: Reserved Bits

[bit12] PD3 EN: Power Domain 3 Power Supply Control Bit

This bit controls power supply to power domain 3.

Bit	Description
0	Shut off power supply.
1	Supply power.

[bit11:9] Reserved: Reserved Bits

[bit8] PD2 EN: Power Domain 2 Power Supply Control Bit

This bit controls power supply to power domain 2.

Bit	Description
0	Shut off power supply.
1	Supply power.

[bit7:0] Reserved: Reserved Bits

5.3.2. PSS Clock Source Enable Register (SYSC0_PSSCKSRER)

The SYSC0_PSSCKSRER sets whether to enable or disable oscillation of the source clock.

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	WPS							
Initial Value	00000000							

bit	23	22	21	20	19	18	17	16
Field	Reserved				SSCG3 EN	SSCG2 EN	SSCG1 EN	SSCG0 EN
R/W Attribute	R0,WX				R/W	R/W	R/W	R/W
Protection Attribute	WPS							
Initial Value	0000				0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	Reserved				PLL3EN	PLL2EN	PLL1EN	PLL0EN
R/W Attribute	R0,WX				R/W	R/W	R/W	R/W
Protection Attribute	WPS							
Initial Value	0000				0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	Reserved				SOSCEN	MOSC EN	SCROSC EN	CROSC EN
R/W Attribute	R0,WX				R/W	R/W	R/W	R/W
Protection Attribute	WPS							
Initial Value	0000				0	1	1	1

[bit31:20] Reserved: Reserved Bits

[bit19] SSCG3EN: SSCG PLL3 Clock Oscillation Enable Bit

This bit controls oscillation of the SSCG PLL3 clock.

Bit	Description
0	Disable oscillation of the SSCG PLL3 clock.
1	Enable oscillation of the SSCG PLL3 clock.

[bit18] SSCG2EN: SSCG PLL2 Clock Oscillation Enable Bit

This bit controls oscillation of the SSCG PLL2 clock.

Bit	Description
0	Disable oscillation of the SSCG PLL2 clock.
1	Enable oscillation of the SSCG PLL2 clock.

[bit17] SSCG1EN: SSCG PLL1 Clock Oscillation Enable Bit

This bit controls oscillation of the SSCG PLL1 clock.

Bit	Description
0	Disable oscillation of the SSCG PLL1 clock.
1	Enable oscillation of the SSCG PLL1 clock.

[bit16] SSCG0EN: SSCG PLL0 Clock Oscillation Enable Bit

This bit controls oscillation of the SSCG PLL0 clock.

Bit	Description
0	Disable oscillation of the SSCG PLL0 clock.
1	Enable oscillation of the SSCG PLL0 clock.

[bit15:12] Reserved: Reserved Bits

[bit11] PLL3EN:PLL3 Clock Oscillation Enable Bit

This bit controls oscillation of the PLL3 clock.

Bit	Description
0	Disable oscillation of the PLL3 clock.
1	Enable oscillation of the PLL3 clock.

[bit10] PLL2EN: PLL2 Clock Oscillation Enable Bit

This bit controls oscillation of the PLL2 clock.

Bit	Description
0	Disable oscillation of the PLL2 clock.
1	Enable oscillation of the PLL2 clock.

[bit9] PLL1EN: PLL1 Clock Oscillation Enable Bit

This bit controls oscillation of the PLL1 clock.

Bit	Description
0	Disable oscillation of the PLL1 clock.
1	Enable oscillation of the PLL1 clock.

[bit8] PLL0EN: PLL0 Clock Oscillation Enable Bit

This bit controls oscillation of the PLL0 clock.

Bit	Description
0	Disable oscillation of the PLL0 clock.
1	Enable oscillation of the PLL0 clock.

[bit7:4] Reserved: Reserved Bits

[bit3] SOSCEN: Sub Clock Oscillation Enable Bit

This bit controls oscillation of the sub clock.

Bit	Description
0	Disable oscillation of the sub clock.
1	Enable oscillation of the sub clock.

[bit2] MOSCEN: Main Clock Oscillation Enable Bit

This bit controls oscillation of the main clock.

Bit	Description
0	Disable oscillation of the main clock.
1	Enable oscillation of the main clock.

[bit1] SCROSCEN: Low-Speed CR Clock Oscillation Enable Bit

This bit controls oscillation of the low-speed CR clock.

Bit	Description
0	Disable oscillation of the low-speed CR clock.
1	Enable oscillation of the low-speed CR clock.

[bit0] CROSCEN: High-Speed CR Clock Oscillation Enable Bit

This bit controls oscillation of the high-speed CR clock.

Bit	Description
0	Disable oscillation of the high-speed CR clock.
1	Enable oscillation of the high-speed CR clock.

5.3.3. PSS Clock Selection Register (SYSC0_PSSCKSELR)

The SYSC0_PSSCKSELR register sets the clock source for the clock domain.

bit	31	8
Field	Reserved	
R/W Attribute	R0,WX	
Protection Attribute	WPS	
Initial Value	00000000_00000000_00000000	

bit	7	6	5	4	3	2	1	0
Field	Reserved					CDMCUCCSL		
R/W Attribute	R0,WX					R/W		
Protection Attribute	WPS							
Initial Value	00000					000		

[bit31:3] Reserved: Reserved Bits

[bit2:0] CDMCUCCSL: Clock Domain MCUC Clock Selection Bits

These bits select the source clock for the clock domain MCUC.

bit2:0	Description
000	High-speed CR clock
001	Low-speed CR clock
010	Main clock
011	Sub clock (Do not use this setting)
100	PLL0 clock
101	SSCG PLL0 clock
110	Setting Prohibited
111	Clock fixed at "L"

5.3.4. PSS Clock Enable Register (SYSC0_PSSCKER)

The SYSC0_PSSCKER register sets whether to stop or enable an internal clock.

bit	31	8
Field	Reserved	
R/W Attribute	R0,WX	
Protection Attribute	WPS	
Initial Value	00000000_00000000_00000000	

bit	7	6	5	4	3	2	1	0
Field	Reserved						ENCLK MCUCP	ENCLK MCUCH
R/W Attribute	R0,WX						R/W	R/W
Protection Attribute	WPS							
Initial Value	000000						1	1

[bit31:2] Reserved: Reserved Bits

[bit1] ENCLKMCUCP: MCUconfig APB Clock Oscillation Enable Bit

This bit sets whether to enable or disable oscillation of the MCUconfig APB clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit0] ENCLKMCUCH: MCUconfig AHB Clock Oscillation Enable Bit

This bit sets whether to enable or disable oscillation of the MCUconfig AHB clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

5.3.5. PSS Clock Divider Register (SYSC0_PSSCKDIVR)

The SYSC0_PSSCKDIVR register sets the clock division ratio.

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	Reserved															
R/W Attribute	R0,WX															
Protection Attribute	WPS															
Initial Value	00000000_00000000															

bit	15	14	13	12	11	10	9	8
Field	Reserved				MCUCPDIV			
R/W Attribute	R0,WX				R/W			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	7	6	5	4	3	2	1	0
Field	Reserved			MCUCHDIV				
R/W Attribute	R0,WX			R/W				
Protection Attribute	WPS							
Initial Value	000			00000				

[bit31:12] Reserved: Reserved Bits

[bit11:8] MCUCPDIV: MCUconfig APB Clock Divider Setting Bits

These bits set the division ratio of the PCOM clock from the HCOM clock.

bit11:8	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

[bit7:5] Reserved: Reserved Bits

[bit4:0] MCUCHDIV: MCUconfig AHB Clock Divider Setting Bits

These bits set the division ratio from the source clock of the clock domain MCUC.

bit4:0	Description
0_0000	No division
0_0001	Divided by 2
0_0010	Divided by 3
...	...
11101	Divided by 30
11110	Divided by 31
11111	Divided by 32

5.3.6. PSS PLLx Control Register (SYSC0_PSSPLLxCNTR)

The SYSC0_PSSPLLxCNTR register sets the division, multiplication rate, etc. for PLLx.

bit	31	30	29	28	27	26	25	24
Field	PLLx ISEL	Reserved						
R/W Attribute	R/W	R0,WX						
Protection Attribute	WPS							
Initial Value	0	0000000						

bit	23	22	21	20	19	18	17	16
Field	PLLxDIVN							
R/W Attribute	R/W							
Protection Attribute	WPS							
Initial Value	00001101							

bit	15	14	13	12	11	10	9	8
Field	Reserved				PLLxDIVM			
R/W Attribute	R0,WX				R/W			
Protection Attribute	WPS							
Initial Value	0000				0001			

bit	7	6	5	4	3	2	1	0
Field	Reserved						PLLxDIVL	
R/W Attribute	R0,WX						R/W	
Protection Attribute	WPS							
Initial Value	000000						00	

[bit31] PLLxISEL: PLLx Input Clock Selection Bit

This bit sets the clock to supply to PLLx.

Bit	Description
0	Main clock
1	High-speed CR clock

[bit30:24] Reserved: Reserved Bits

[bit23:16] PLLxDIVN: PLLx Clock N-multiplier Setting Bits

These bits set the multiplication rate of the PLLx clock.

bit23:16	Description
0000_0000	Setting prohibited
...	...
0000_1100	Setting prohibited

bit23:16	Description
0000_1101	Multiply by 13
0000_1110	Multiply by 14
...	...
1100_0110	Multiply by 198
1100_0111	Multiply by 199
1100_1000	Multiply by 200
1100_1001	Setting prohibited
...	...
1100_1111	Setting prohibited

[bit15:12] Reserved: Reserved Bits

[bit11:8] PLLxDIVM: PLLx Clock M-Divider Setting Bits

These bits set the division ratio for PLLx clock output.

bit11:8	Description
000x	Divided by 2
0010	Divided by 4
0011	Divided by 6
0100	Divided by 8
0101	Divided by 10
0110	Divided by 12
0111	Divided by 14
1000	Divided by 16
1001	Divided by 18
1010	Divided by 20
1011	Divided by 22
1010	Divided by 24
1101	Divided by 26
1110	Divided by 28
1111	Divided by 30

[bit7:2] Reserved: Reserved Bits

[bit1:0] PLLxDIVL: PLLx Input Clock Divider Setting Bits

These bits set the division ratio of the PLLx input clock.

bit1:0	Description
00	No division
01	Divided by 2
10	Divided by 4
11	Divided by 6

5.3.7. PSS SSCGx Control Register 0 (SYSC0_PSSSSCGxCNTR0)

The SYSC0_PSSSSCGxCNTR0 register sets the division, multiplication rate, etc. for SSCG PLLx.

bit	31	30	29	28	27	26	25	24
Field	SSCGx ISEL	Reserved						
R/W Attribute	R/W	R0,WX						
Protection Attribute	WPS							
Initial Value	0	0000000						

bit	23	22	21	20	19	18	17	16
Field	SSCGxDIVN							
R/W Attribute	R/W							
Protection Attribute	WPS							
Initial Value	00001101							

bit	15	14	13	12	11	10	9	8
Field	Reserved				SSCGxDIVM			
R/W Attribute	R0,WX				R/W			
Protection Attribute	WPS							
Initial Value	0000				0001			

bit	7	6	5	4	3	2	1	0
Field	Reserved						SSCGxDIVL	
R/W Attribute	R0,WX						R/W	
Protection Attribute	WPS							
Initial Value	000000						00	

[bit31] SSCGxISEL: SSCG PLLx Input Clock Selection Bit

This bit sets the clock to supply to SSCG PLLx.

Bit	Description
0	Main clock
1	High-speed CR clock

[bit30:24] Reserved: Reserved Bits

[bit23:16] SSCGxDIVN: SSCG PLLx Clock N-multiplier Setting Bits

These bits set the N multiplication rate of the SSCG PLLx clock.

bit23:16	Description
0000_0000	Setting prohibited
...	...
0000_1100	Setting prohibited

bit23:16	Description
0000_1101	Multiply by 13
0000_1110	Multiply by 14
...	...
1100_0110	Multiply by 198
1100_0111	Multiply by 199
1100_1000	Multiply by 200
1100_1001	Setting prohibited
...	...
1100_1111	Setting prohibited

[bit15:12] Reserved: Reserved Bits

[bit11:8] SSCGxDIVM: SSCG PLLx Clock M-Division Setting Bits

These bits set the division ratio for SSCG PLLx clock output.

bit11:8	Description
000x	Divided by 2
0010	Divided by 4
0011	Divided by 6
0100	Divided by 8
0101	Divided by 10
0110	Divided by 12
0111	Divided by 14
1000	Divided by 16
1001	Divided by 18
1010	Divided by 20
1011	Divided by 22
1010	Divided by 24
1101	Divided by 26
1110	Divided by 28
1111	Divided by 30

[bit7:2] Reserved: Reserved Bits

[bit1:0] SSCGxDIVL: SSCG PLLx Input Clock Divider Setting Bits

These bits set the division ratio of the SSCG PLLx input clock.

bit1:0	Description
00	No division
01	Divided by 2
10	Divided by 4
11	Divided by 6

5.3.8. PSS SSCGx Control Register 1 (SYSC0_PSSSSCGxCNTR1)

The SYSC0_PSSSSCGxCNTR1 register sets modulation enable, the modulation mode, etc. for SSCG PLLx.

bit	31	30	29	28	27	26	25	24
Field	Reserved							SSCGx SSEN
R/W Attribute	R0,WX							R/W
Protection Attribute	WPS							
Initial Value	0000000							0

bit	23	22	21	20	19	18	17	16
Field	Reserved					SSCGxFREQ		SSCGx MODE
R/W Attribute	R0,WX					R/W		R/W
Protection Attribute	WPS							
Initial Value	00000					00		0

bit	15	14	13	12	11	10	9	8
Field	Reserved						SSCGxRATE	
R/W Attribute	R0,WX						R/W	
Protection Attribute	WPS							
Initial Value	000000						00	

bit	7	6	5	4	3	2	1	0
Field	SSCGxRATE							
R/W Attribute	R/W							
Protection Attribute	WPS							
Initial Value	00101001							

[bit31:25] Reserved: Reserved Bits

[bit24] SSCGxSSEN: SSCG PLLx modulation Enable Setting Bit

This bit controls SSCG PLLx modulation.

Bit	Description
0	Disable modulation.
1	Enable modulation.

[bit23:19] Reserved: Reserved Bits

[bit18:17] SSCGxFREQ: SSCG PLLx Clock modulation Frequency Selection Bits

These bits set the modulation frequency of the SSCG PLL clock.

bit18:17	Description
00	Fmod = (1/1024) x Fin
01	Fmod = (1/2048) x Fin
10 or 11	Fmod = (1/4096) x Fin

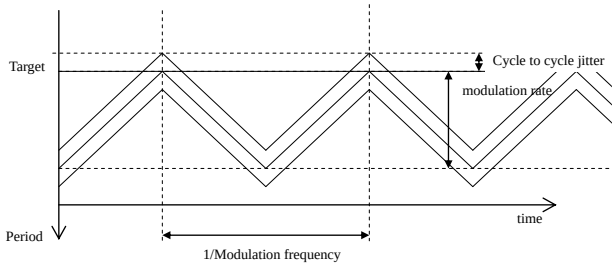
* Fin: SSCG reference clock frequency, Fmod: Modulation frequency

[bit16] SSCGxMODE: SSCG PLLx modulation Mode Setting Bit

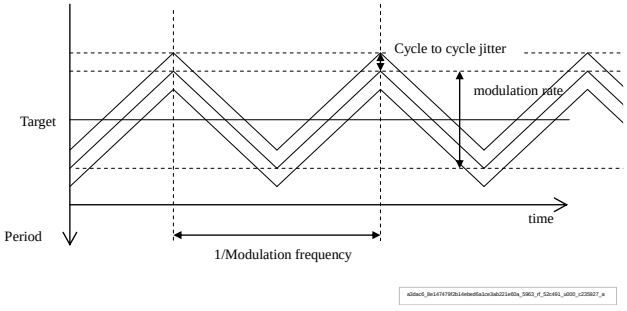
This bit sets the modulation mode of the SSCG PLL clock.

Bit	Description
0	Set Down spread mode.
1	Set Center spread mode.

- Down spread



- Center spread



[bit15:10] Reserved: Reserved Bits

[bit9:0] SSCGxRATE: SSCG PLLx Clock modulation Ratio Control Bits

These bits set the modulation ratio of the SSCG PLL clock.

bit9:0	Description
00_0010_1001	Modulation ratio of 0.5%
00_0101_0010	Modulation ratio of 1.0%
00_1010_0100	Modulation ratio of 2.0%
00_1111_0110	Modulation ratio of 3.0%
01_0100_1000	Modulation ratio of 4.0%
01_1001_1010	Modulation ratio of 5.0%
Others	Setting prohibited

5.3.9. PSS Low-Voltage Detection Setting Register (SYSC0_PSSLVDCFGR)

The SYSC0_PSSLVDCFGR register sets each internal low-voltage detection.

bit	31	30	29	28	27	26	25	24
Field	Reserved	LVDL1S	Reserved			LVDL1V		LVDL1E
R/W Attribute	R0,WX	(product specification)	R0,WX			(product specification)		(product specification)
Protection Attribute	WPS							
Initial Value	0	(product specification)	000			(product specification)		(product specification)

bit	23	22	21	20	19	18	17	16
Field	Reserved	LVDL2S	Reserved			LVDL2V		LVDL2E
R/W Attribute	R0,WX	(product specification)	R0,WX			(product specification)		(product specification)
Protection Attribute	WPS							
Initial Value	0	(product specification)	000			(product specification)		(product specification)

bit	15	14	13	12	11	10	9	8
Field	Reserved	LVDH1S	Reserved	LVDH1V				LVDH1E
R/W Attribute	R0,WX	(product specification)	R0,WX	(product specification)				(product specification)
Protection Attribute	WPS							
Initial Value	0	(product specification)	0	(product specification)				(product specification)

bit	7	6	5	4	3	2	1	0
Field	Reserved	LVDH2S	Reserved	LVDH2V				LVDH2E
R/W Attribute	R0,WX	(product specification)	R0,WX	(product specification)				(product specification)
Protection Attribute	WPS							
Initial Value	0	(product specification)	0	(product specification)				(product specification)

[bit31] Reserved: Reserved Bit

[bit30] LVDL1S: Internal Low-Voltage Detection Operation Selection Bit

This bit selects an operation when the internal low-voltage detection occurs.

Bit	Description
0	Reset.
1	Interrupt.

[bit29:27] Reserved: Reserved Bits

[bit26:25] LVDL1V: Internal Low-Voltage Detection Voltage Setting Bits

These bits set the detection voltage for internal low-voltage detection.

Bit	Description
00	(product specification)
01	(product specification)
10	(product specification)
11	(product specification)

Note:

- *After the detection voltage is changed, low-voltage detection is unavailable until the low-voltage detection stability time elapses.*

[bit24] LVDL1E: Internal Low-Voltage Detection Operation Enable Bit

This bit sets the internal low-voltage detection operation.

Bit	Description
0	Stop operation.
1	Enable operation.

Note:

- *After this setting is changed from "0" (stop operation) to "1" (enable operation), low-voltage detection is unavailable until the low-voltage detection stability time elapses.*

[bit23] Reserved: Reserved Bit

[bit22] LVDL2S: Extended Internal Low-Voltage Detection Operation Selection Bit

This bit selects an operation when the extended internal low-voltage detection occurs.

Bit	Description
0	Reset.
1	Interrupt.

[bit21:19] Reserved: Reserved Bits

[bit18:17] LVDL2V: Extended Internal Low-Voltage Detection Voltage Setting Bits

These bits set the detection voltage for extended internal low-voltage detection.

Bit	Description
00	(product specification)
01	(product specification)
10	(product specification)
11	(product specification)

Note:

- *After the detection voltage is changed, low-voltage detection is unavailable until the low-voltage detection stability time elapses.*

[bit16] LVDL2E: Extended Internal Low-Voltage Detection Operation Enable Bit

This bit sets the extended internal low-voltage detection operation.

Bit	Description
0	Stop operation.
1	Enable operation.

Note:

- *After this setting is changed from "0" (stop operation) to "1" (enable operation), low-voltage detection is unavailable until the low-voltage detection stability time elapses.*

[bit15] Reserved: Reserved Bit

[bit14] LVDH1S: External Low-Voltage Detection Operation Selection Bit

This bit selects an operation when the external low-voltage detection occurs.

Bit	Description
0	Reset.
1	Interrupt.

[bit13] Reserved: Reserved Bit

[bit12:9] LVDH1V: External Low-Voltage Detection Voltage Setting Bits

These bits set the detection voltage for external low-voltage detection.

Bit	Description
0000	(product specification)
0001	(product specification)
0010	(product specification)
0011	(product specification)
0100	(product specification)
0101	(product specification)
0110	(product specification)
0111	(product specification)
1xxx	(product specification)

Note:

- *After the detection voltage is changed, low-voltage detection is unavailable until the low-voltage detection stability time elapses.*

[bit8] LVDH1E: External Low-Voltage Detection Operation Enable Bit

This bit sets the external low-voltage detection operation.

Bit	Description
0	Stop operation.
1	Enable operation.

Note:

- *After this setting is changed from "0" (stop operation) to "1" (enable operation), low-voltage detection is unavailable until the low-voltage detection stability time elapses.*

[bit7] Reserved: Reserved Bit

[bit6] LVDH2S: Extended External Low-Voltage Detection Operation Selection Bit

This bit selects an operation when the extended external low-voltage detection occurs.

Bit	Description
0	Reset.
1	Interrupt.

[bit5] Reserved: Reserved Bit

[bit4:1] LVDH2V: Extended External Low-Voltage Detection Voltage Setting Bits

These bits set the detection voltage for extended external low-voltage detection.

Bit	Description
0000	(product specification)
0001	(product specification)
0010	(product specification)
0011	(product specification)
0100	(product specification)
0101	(product specification)
0110	(product specification)
0111	(product specification)
1xxx	(product specification)

Note:

- *After the detection voltage is changed, low-voltage detection is unavailable until the low-voltage detection stability time elapses.*

[bit0] LVDH2E: Extended External Low-Voltage Detection Operation Enable Bit

This bit sets the extended external low-voltage detection operation.

Bit	Description
0	Stop operation.
1	Enable operation.

Note:

- *After this setting is changed from "0" (stop operation) to "1" (enable operation), low-voltage detection is unavailable until the low-voltage detection stability time elapses.*

5.3.10. PSS Clock Supervisor Setting Register (SYSC0_PSSCSVCFGR)

The SYSC0_PSSCSVCFGR register sets whether to enable or disable operation of each clock supervisor.

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	WPS							
Initial Value	00000000							

bit	23	22	21	20	19	18	17	16
Field	Reserved				SSCG3 CSVE	SSCG2 CSVE	SSCG1 CSVE	SSCG0 CSVE
R/W Attribute	R0,WX				R/W	R/W	R/W	R/W
Protection Attribute	WPS							
Initial Value	0000				0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	Reserved				PLL3 CSVE	PLL2 CSVE	PLL1 CSVE	PLL0 CSVE
R/W Attribute	R0,WX				R/W	R/W	R/W	R/W
Protection Attribute	WPS							
Initial Value	0000				0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	Reserved				SCR CSVE	CR CSVE	SO CSVE	MO CSVE
R/W Attribute	R0,WX				R/W	R/W	R/W	R/W
Protection Attribute	WPS							
Initial Value	0000				0	0	0	0

[bit31:20] Reserved: Reserved Bits

[bit19] SSCG3CSVE: SSCG PLL3 Clock Supervisor Enable Bit

This bit sets whether to enable or disable the SSCG PLL3 clock supervisor.

Bit	Description
0	Disable the clock supervisor.
1	Enable the clock supervisor.

[bit18] SSCG2CSVE: SSCG PLL2 Clock Supervisor Enable Bit

This bit sets whether to enable or disable the SSCG PLL2 clock supervisor.

Bit	Description
0	Disable the clock supervisor.
1	Enable the clock supervisor.

[bit17] SSCG1CSVE: SSCG PLL1 Clock Supervisor Enable Bit

This bit sets whether to enable or disable the SSCG PLL1 clock supervisor.

Bit	Description
0	Disable the clock supervisor.
1	Enable the clock supervisor.

[bit16] SSCG0CSVE: SSCG PLL0 Clock Supervisor Enable Bit

This bit sets whether to enable or disable the SSCG PLL0 clock supervisor.

Bit	Description
0	Disable the clock supervisor.
1	Enable the clock supervisor.

[bit15:12] Reserved: Reserved Bits

[bit11] PLL3CSVE: PLL3 Clock Supervisor Enable Bit

This bit sets whether to enable or disable the PLL3 clock supervisor.

Bit	Description
0	Disable the clock supervisor.
1	Enable the clock supervisor.

[bit10] PLL2CSVE: PLL2 Clock Supervisor Enable Bit

This bit sets whether to enable or disable the PLL2 clock supervisor.

Bit	Description
0	Disable the clock supervisor.
1	Enable the clock supervisor.

[bit9] PLL1CSVE: PLL1 Clock Supervisor Enable Bit

This bit sets whether to enable or disable the PLL1 clock supervisor.

Bit	Description
0	Disable the clock supervisor.
1	Enable the clock supervisor.

[bit8] PLL0CSVE: PLL0 Clock Supervisor Enable Bit

This bit sets whether to enable or disable the PLL0 clock supervisor.

Bit	Description
0	Disable the clock supervisor.
1	Enable the clock supervisor.

[bit7:4] Reserved: Reserved Bits

[bit3] SCRCSVE：Low-Speed CR Clock Supervisor Enable Bit

This bit sets whether to enable or disable the sub clock supervisor.

Bit	Description
0	Disable the clock supervisor.
1	Enable the clock supervisor.

[bit2] CRCSVE：High-Speed CR Clock Supervisor Enable Bit

This bit sets whether to enable or disable the main clock supervisor.

Bit	Description
0	Disable the clock supervisor.
1	Enable the clock supervisor.

[bit1] SOCSVE: Sub Clock Supervisor Enable Bit

This bit sets whether to enable or disable the sub clock supervisor.

Bit	Description
0	Disable the clock supervisor.
1	Enable the clock supervisor.

[bit0] MOCSVE: Main Clock Supervisor Enable Bit

This bit sets whether to enable or disable the main clock supervisor.

Bit	Description
0	Disable the clock supervisor.
1	Enable the clock supervisor.

5.3.11. PSS Regulator Setting Register (SYSC0_PSSREGCFGR)

The SYSC0_PSSREGCFGR register sets the regulator mode and output voltage.

bit	31							8
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	WPS							
Initial Value	00000000_00000000_00000000							

bit	7	6	5	4	3	2	1	0
Field	RMSEL	Reserved						RVSEL
R/W Attribute	R/W	R0,WX						R,WX
Protection Attribute	WPS							
Initial Value	0	000000						(product specification)

[bit31:8] Reserved: Reserved Bits

[bit7] RMSEL: Regulator Mode Setting Bit

This bit set the regulator mode setting.

Bit	Description
0	Main mode
1	Standby mode

Main mode is normal operation in regulator.

Standby mode is low power operation in regulator.

In `Standby mode`, current consumption in PSS mode can be reduced than Main mode by setting this bit.

Note: Specific value of current consumption depend on each product.

Please refer to Datasheet for any device-specific information.

[bit6:1] Reserved: Reserved Bits

[bit0] RVSEL: Regulator Output Voltage Setting Bits

These bits indicate the regulator output voltage.

Bit	Description
0	1.20V
1	1.25V

Note:

- *The setting of this bit will be as same as SYSC0_RUNREGCFGR .RVSEL.*

5.3.12. PSS Profile Update Enable Register (SYSC0_PSSENR)

This register enables PSS profile updates.

bit	31	8
Field	Reserved	
R/W Attribute	R0, WX	
Protection Attribute	WPS	
Initial Value	00000000_00000000_00000000	

bit	7	6	5	4	3	2	1	0
Field	PSSEN0							
R/W Attribute	R,W							
Protection Attribute	WPS							
Initial Value	00000000							

[bit31:8] Reserved: Reserved Bits

[bit7:0] PSSEN0: PSS profile Update Enable Setting Bits

These bits compose the PSS profile update start register.

bit7:0	Description
During write operation	When "0xBA" is written: Write "0xBA" to the register and PSSEN will be valid. When "0x00" is written: Write "0x00" to the register and PSSEN will be invalid. When other than "0xBA"/"0x00" is written: Invalid
During read operation	When "0xBA" is written: "0xBA" is read. Otherwise: "0x00" is read.

Notes:

- *If the value written in this register invalid, an error flag (SYSC0_SYSERRIR1:PSSENERRIF0 is "1") is enabled.*
- *This register permits byte access only. Any other access is invalid. If the above is not complied with, a bus error occurs.*

5.4. APP Profile Register Group (SYSC0)

The registers in this group are APP profile control setting registers.

5.4.1. APP Power Domain Setting Register (SYSC0_APPPDCFGR)

The SYSC0_APPPDCFGR register indicates the ON/OFF of the power domain which is going to be updated.

bit	31	30	29	28	27	26	25	24
Field	Reserved						PD6_1 EN	PD6_0 EN
R/W Attribute	R0,WX						R,WX	R,WX
Protection Attribute	WPS							
Initial Value	000000						1	1

bit	23	22	21	20	19	18	17	16
Field	PD5_3 EN	PD5_2 EN	PD5_1 EN	PD5_0 EN	Reserved		PD4_1 EN	PD4_0 EN
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R0,WX		R,WX	R,WX
Protection Attribute	WPS							
Initial Value	1	1	1	1	00		1	1

bit	15	14	13	12	11	10	9	8
Field	Reserved			PD3 EN	Reserved			PD2 EN
R/W Attribute	R0,WX			R,WX	R0,WX			R,WX
Protection Attribute	WPS							
Initial Value	000			1	000			1

bit	7	6	5	4	3	2	1	0
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	WPS							
Initial Value	00000000							

[bit31:26] Reserved: Reserved Bits

[bit25] PD6_1EN: Power Control Bit for Power Domain 6_1

This bit sets the ON/OFF of power domain 6_1 when it is updated.

Bit	Description
0	OFF
1	ON

[bit24] PD6_0 EN: Power Control Bit for Power Domain 6_0

This bit sets the ON/OFF of power domain 6_0 when it is updated.

Bit	Description
0	OFF
1	ON

[bit23] PD5_3 EN: Power Control Bit for Power Domain 5_3

This bit sets the ON/OFF of power domain 5_3 when it is updated.

Bit	Description
0	OFF
1	ON

[bit22] PD5_2 EN: Power Control Bit for Power Domain 5_2

This bit sets the ON/OFF of power domain 5_2 when it is updated.

Bit	Description
0	OFF
1	ON

[bit21] PD5_1 EN: Power Control Bit for Power Domain 5_1

This bit sets the ON/OFF of power domain 5_1 when it is updated.

Bit	Description
0	OFF
1	ON

[bit20] PD5_0 EN: Power Control Bit for Power Domain 5_0

This bit sets the ON/OFF of power domain 5_0 when it is updated.

Bit	Description
0	OFF
1	ON

[bit19:18] Reserved: Reserved Bits

[bit17] PD4_1 EN: Power Control Bit for Power Domain 4_1

This bit sets the ON/OFF of power domain 4_1 when it is updated.

Bit	Description
0	OFF
1	ON

[bit16] PD4_0 EN: Power Control Bit for Power Domain 4_0

This bit sets the ON/OFF of power domain 4_0 when it is updated.

Bit	Description
0	OFF
1	ON

[bit15:13] Reserved: Reserved Bits

[bit12] PD3 EN: Power Control Bit for Power Domain 3

This bit sets the ON/OFF of power domain 3 when it is updated.

Bit	Description
0	OFF
1	ON

[bit11:9] Reserved: Reserved Bits

[bit8] PD2 EN: Power Control Bit for Power Domain 2

This bit sets the ON/OFF of power domain 2 when it is updated.

Bit	Description
0	OFF
1	ON

[bit7:0] Reserved: Reserved Bits

5.4.2. APP Clock Source Enable Register (SYSC0_APPCKSRER)

The SYSC0_APPCKSRER register indicates the value for setting whether to enable/disable oscillation of the internal operating clock to be updated.

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	WPS							
Initial Value	00000000							

bit	23	22	21	20	19	18	17	16
Field	Reserved				SSCG3EN	SSCG2EN	SSCG1EN	SSCG0EN
R/W Attribute	R0,WX				R,WX	R,WX	R,WX	R,WX
Protection Attribute	WPS							
Initial Value	0000				0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	Reserved				PLL3EN	PLL2EN	PLL1EN	PLL0EN
R/W Attribute	R0,WX				R,WX	R,WX	R,WX	R,WX
Protection Attribute	WPS							
Initial Value	0000				0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	Reserved				SOSCEN	MOSCEN	SCROSCEN	CROSCEN
R/W Attribute	R0,WX				R,WX	R,WX	R,WX	R,WX
Protection Attribute	WPS							
Initial Value	0000				0	1	1	1

[bit31:20] Reserved: Reserved Bits

[bit19] SSCG3EN: SSCG PLL3 Clock Oscillation Enable Bit

This bit indicates the set value of the SSCG PLL3 clock to be updated.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit18] SSCG2EN: SSCG PLL2 Clock Oscillation Enable Bit

This bit indicates the set value of the SSCG PLL2 clock to be updated.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit17] SSCG1EN: SSCG PLL1 Clock Oscillation Enable Bit

This bit indicates the set value of the SSCG PLL1 clock to be updated.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit16] SSCG0EN: SSCG PLL0 Clock Oscillation Enable Bit

This bit indicates the set value of the SSCG PLL0 clock to be updated.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit15:12] Reserved: Reserved Bits

[bit11] PLL3EN: PLL3 Clock Oscillation Enable Bit

This bit indicates the set value of the PLL3 clock to be updated.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit10] PLL2EN: PLL2 Clock Oscillation Enable Bit

This bit indicates the set value of the PLL2 clock to be updated.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit9] PLL1EN: PLL1 Clock Oscillation Enable Bit

This bit indicates the set value of the PLL1 clock to be updated.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit8] PLL0EN: PLL0 Clock Oscillation Enable Bit

This bit indicates the set value of the PLL0 clock to be updated.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit7:4] Reserved: Reserved Bits

[bit3] SOSCEN: Sub Clock Oscillation Enable Bit

This bit indicates the set value of the sub clock to be updated.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit2] MOSCEN: Main Clock Oscillation Enable Bit

This bit indicates the set value of the main clock to be updated.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit1] SCROSCEN: Low-Speed CR Clock Oscillation Enable Bit

This bit indicates the set value of the low-speed CR clock to be updated.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit0] CROSCEN: High-Speed CR Clock Oscillation Enable Bit

This bit indicates the set value of the high-speed CR clock to be updated.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

5.4.3. APP Clock Selection Register (SYSC0_APPCKSELR)

The SYSC0_APPCKSELR register indicates the set values of the clock source for the clock domain to be updated.

bit	31	8
Field	Reserved	
R/W Attribute	R0,WX	
Protection Attribute	WPS	
Initial Value	00000000_00000000_00000000	

bit	7	6	5	4	3	2	1	0
Field	Reserved					CDMCUCCSL		
R/W Attribute	R0,WX					R,WX		
Protection Attribute	WPS							
Initial Value	00000					000		

[bit31:3] Reserved: Reserved Bits

[bit2:0] CDMCUCCSL: Clock Domain MCUC Clock Selection Bits.

These bits indicate the set value of clock domain MCUC clock to be updated.

bit2:0	Description
000	High-speed CR clock
001	Low-speed CR clock
010	Main clock
011	Sub clock (Do not use this setting)
100	PLL0 clock
101	SSCG PLL0 clock
110	Setting Prohibited
111	Clock fixed at "L"

5.4.4. APP Clock Enable Register (SYSC0_APPCKER)

The SYSC0_APPCKER register indicates the value for setting whether to enable/disable oscillation of the internal operating clock to be updated.

bit	31	8
Field	Reserved	
R/W Attribute	R0,WX	
Protection Attribute	WPS	
Initial Value	00000000_00000000_00000000	

bit	7	6	5	4	3	2	1	0
Field	Reserved						ENCLKMCUCP	ENCLKMCUCH
R/W Attribute	R0,WX						R,WX	R,WX
Protection Attribute	WPS							
Initial Value	000000						1	1

[bit31:2] Reserved: Reserved Bits

[bit1] ENCLKMCUCP: MCU config APB Clock Oscillation Enable Bit

This bit indicates the set value of the MCU config APB clock to be updated.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit0] ENCLKMCUCH: MCU config AHB Clock Oscillation Enable Bit

This bit indicates the set value of the MCU config AHB clock to be updated.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

5.4.5. APP Clock Divider Register (SYSC0_APPCKDIVR)

The SYSC0_APPCKDIVR register indicates the set values of the division ratio of each clock to be updated.

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	Reserved															
R/W Attribute	R0,WX															
Protection Attribute	WPS															
Initial Value	00000000_00000000															

bit	15	14	13	12	11	10	9	8
Field	Reserved				MCUCPDIV			
R/W Attribute	R0,WX				R,WX			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	7	6	5	4	3	2	1	0
Field	Reserved			MCUCHDIV				
R/W Attribute	R0,WX			R,WX				
Protection Attribute	WPS							
Initial Value	000			00000				

[bit31:12] Reserved: Reserved Bits

[bit11:8] MCUCPDIV: MCU config APB Clock Divider Setting Bits

These bits indicate the division set value of the PCOM clock from HCOM clock to be updated.

bit11:8	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
…	…
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

[bit7:5] Reserved: Reserved Bits

[bit4:0] MCUCHDIV: MCU config AHB Clock Divider Setting Bits

These bits indicate the division set value of the clock domain MCUC.

bit4:0	Description
0_0000	No division
0_0001	Divided by 2
0_0010	Divided by 3
…	...
1_1101	Divided by 30
1_1110	Divided by 31
1_1111	Divided by 32

5.4.6. APP PLLx Control Register (SYSC0_APPPLLxCNTR)

The SYSC0_APPPLLxCNTR register indicates the division and the multiplier of the PLLx to be updated.

bit	31	30	29	28	27	26	25	24
Field	PLLx ISEL	Reserved						
R/W Attribute	R,WX	R0,WX						
Protection Attribute	WPS							
Initial Value	0	0000000						

bit	23	22	21	20	19	18	17	16
Field	PLLxDIVN							
R/W Attribute	R,WX							
Protection Attribute	WPS							
Initial Value	00001101							

bit	15	14	13	12	11	10	9	8
Field	Reserved				PLLxDIVM			
R/W Attribute	R0,WX				R,WX			
Protection Attribute	WPS							
Initial Value	0000				0001			

bit	7	6	5	4	3	2	1	0
Field	Reserved						PLLxDIVL	
R/W Attribute	R0,WX						R,WX	
Protection Attribute	WPS							
Initial Value	000000						00	

[bit31] PLLxISEL: PLLx Input Clock Selection Bit

This bit indicates the setting of input source clock to PLLx to be updated.

Bit	Description
0	Main clock
1	High-speed CR clock

[bit30:24] Reserved: Reserved Bits

[bit23:16] PLLxDIVN: NPLLx Clock N-multiplier Ratio Setting Bits

These bits set the multiplication rate of the NPLLx clock.

bit23:16	Description
0000_0000	prohibited
…	…
0000_1100	prohibited
0000_1101	Multiply by 13
0000_1110	Multiply by 14
…	…
1100_0110	Multiply by 198
1100_0111	Multiply by 199
1100_1000	Multiply by 200
1100_1001	prohibited
…	…
1100_1111	prohibited

[bit15:12] Reserved: Reserved Bits

[bit11:8] PLLxDIVM: PLLx Clock M-Divider Setting Bits

These bits set the division ratio for PLLx clock output.

bit11:8	Description
000x	Divided by 2
0010	Divided by 4
0011	Divided by 6
0100	Divided by 8
0101	Divided by 10
0110	Divided by 12
0111	Divided by 14
1000	Divided by 16
1001	Divided by 18
1010	Divided by 20
1011	Divided by 22
1010	Divided by 24
1101	Divided by 26
1110	Divided by 28
1111	Divided by 30

[bit7:2] Reserved: Reserved Bits

[bit1:0] PLLxDIVL: PLLx Input Clock Divider Setting Bits

These bits set the division ratio of the PLLx input clock.

bit1:0	Description
00	No division
01	Divided by 2
10	Divided by 4
11	Divided by 6

5.4.7. APP SSCGx Control Register 0 (SYSC0_APPSSCGxCNTR0)

The SYSC0_APPSSCGxCNTR0 register indicates the set values of the division, multiplication rate, etc. for SSCG PLLx to be updated.

bit	31	30	29	28	27	26	25	24
Field	SSCGx ISEL	Reserved						
R/W Attribute	R,WX	R0,WX						
Protection Attribute	WPS							
Initial Value	0	0000000						

bit	23	22	21	20	19	18	17	16
Field	SSCGxDIVN							
R/W Attribute	R,WX							
Protection Attribute	WPS							
Initial Value	00001101							

bit	15	14	13	12	11	10	9	8
Field	Reserved				SSCGxDIVM			
R/W Attribute	R0,WX				R,WX			
Protection Attribute	WPS							
Initial Value	0000				0001			

bit	7	6	5	4	3	2	1	0
Field	Reserved						SSCGxDIVL	
R/W Attribute	R0,WX						R,WX	
Protection Attribute	WPS							
Initial Value	000000						00	

[bit31] SSCGxISEL: SSCG PLLx Input Clock Selection Bit

This bit sets the clock to supply to SSCG PLLx.

Bit	Description
0	Main clock
1	High-speed CR clock

[bit30:24] Reserved: Reserved Bits

[bit23:16] SSCGxDIVN: SSCG PLLx Clock N-multiplier Setting Bits

These bits set the N multiplication rate of the SSCG PLLx clock.

bit23:16	Description
0000_0000	Setting prohibited
...	...
0000_1100	Setting prohibited
0000_1101	Multiply by 13
0000_1110	Multiply by 14
...	...
1100_0110	Multiply by 198
1100_0111	Multiply by 199
1100_1000	Multiply by 200
1100_1001	Setting prohibited
...	...
1100_1111	Setting prohibited

[bit15:12] Reserved: Reserved Bits

[bit11:8] SSCGxDIVM: SSCG PLLx Clock M-Division Setting Bits

These bits set the division ratio for SSCG PLLx clock output.

bit11:8	Description
000x	Divided by 2
0010	Divided by 4
0011	Divided by 6
0100	Divided by 8
0101	Divided by 10
0110	Divided by 12
0111	Divided by 14
1000	Divided by 16
1001	Divided by 18
1010	Divided by 20
1011	Divided by 22
1010	Divided by 24
1101	Divided by 26
1110	Divided by 28
1111	Divided by 30

[bit7:2] Reserved: Reserved Bits

[bit1:0] SSCGxDIVL: SSCG PLLx Input Clock Divider Setting Bits

These bits set the division ratio of the SSCG PLLx input clock.

bit1:0	Description
00	No division
01	Divided by 2
10	Divided by 4
11	Divided by 6

5.4.8. APP SSCGx Control Register 1 (SYSC0_APPSSCGxCNTR1)

The SYSC0_APPSSCGxCNTR1 register indicates the set values for modulation enable, the modulation mode, etc. for SSCG PLLx to be updated.

bit	31	30	29	28	27	26	25	24
Field	Reserved							SSCGx SSEN
R/W Attribute	R0,WX							R,WX
Protection Attribute	WPS							
Initial Value	0000000							0

bit	23	22	21	20	19	18	17	16
Field	Reserved					SSCGxFREQ		SSCGx MODE
R/W Attribute	R0,WX					R,WX		R,WX
Protection Attribute	WPS							
Initial Value	00000					00		0

bit	15	14	13	12	11	10	9	8
Field	Reserved						SSCGxRATE	
R/W Attribute	R0,WX						R,WX	
Protection Attribute	WPS							
Initial Value	000000						00	

bit	7	6	5	4	3	2	1	0
Field	SSCGxRATE							
R/W Attribute	R,WX							
Protection Attribute	WPS							
Initial Value	00101001							

[bit31:25] Reserved: Reserved Bits

[bit24] SSCGxSSEN: SSCG PLLx modulation Enable Setting Bit

This bit controls SSCG PLLx modulation.

Bit	Description
0	Disable modulation.
1	Enable modulation.

[bit23:19] Reserved: Reserved Bits

[bit18:17] SSCGxFREQ: SSCG PLLx Clock modulation Frequency Selection Bits

These bits set the modulation frequency of the SSCG PLLx clock.

bit18:17	Description
00	Fmod = (1/1024) x Fin
01	Fmod = (1/2048) x Fin
10 or 11	Fmod = (1/4096) x Fin

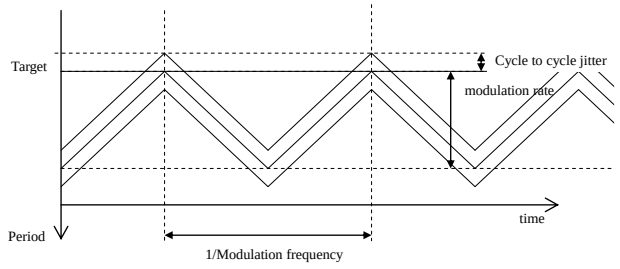
* Fin: SSCG reference clock frequency, Fmod: Modulation frequency

[bit16] SSCGxMODE: SSCG PLLx modulation Mode Setting Bit

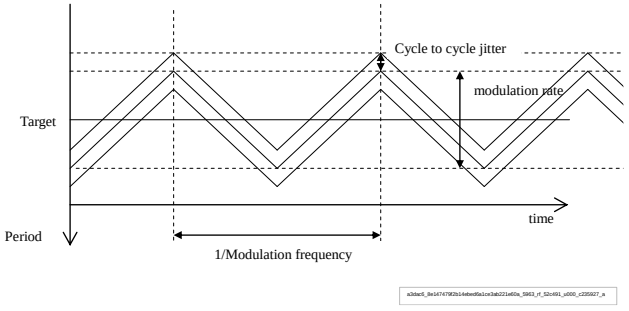
This bit sets the modulation mode of the SSCG PLLx clock.

Bit	Description
0	Set Down spread mode.
1	Set Center spread mode.

- Down spread



- Center spread



[bit15:10] Reserved: Reserved Bits

[bit9:0] SSCGxRATE: SSCG PLLx Clock modulation Ratio Control Bits

These bits set the modulation ratio of the SSCG PLLx clock.

bit9:0	Description
00_0010_1001	Modulation ratio of 0.5%
00_0101_0010	Modulation ratio of 1.0%
00_1010_0100	Modulation ratio of 2.0%
00_1111_0110	Modulation ratio of 3.0%
01_0100_1000	Modulation ratio of 4.0%
01_1001_1010	Modulation ratio of 5.0%
Others	Setting prohibited

5.4.9. APP Low-Voltage Detection Setting Register (SYSC0_APPLVDCFGR)

The SYSC0_APPLVDCFGR register indicates the set values for each internal low-voltage detection to be updated.

bit	31	30	29	28	27	26	25	24
Field	Reserved	LVDL1S	Reserved			LVDL1V		LVDL1E
R/W Attribute	R0,WX	R,WX	R0,WX			R,WX		R,WX
Protection Attribute	WPS							
Initial Value	0	(product specification)	000			(product specification)		(product specification)

bit	23	22	21	20	19	18	17	16
Field	Reserved	LVDL2S	Reserved			LVDL2V		LVDL2E
R/W Attribute	R0,WX	R,WX	R0,WX			R,WX		R,WX
Protection Attribute	WPS							
Initial Value	0	(product specification)	000			(product specification)		(product specification)

bit	15	14	13	12	11	10	9	8
Field	Reserved	LVDH1S	Reserved	LVDH1V				LVDH1E
R/W Attribute	R0,WX	R,WX	R0,WX	R,WX				R,WX
Protection Attribute	WPS							
Initial Value	0	(product specification)	0	(product specification)				(product specification)

bit	7	6	5	4	3	2	1	0
Field	Reserved	LVDH2S	Reserved	LVDH2V				LVDH2E
R/W Attribute	R0,WX	R,WX	R0,WX	R,WX				R,WX
Protection Attribute	WPS							
Initial Value	0	(product specification)	0	(product specification)				(product specification)

[bit31] Reserved: Reserved Bit

[bit30] LVDL1S: Internal Low-Voltage Detection Operation Selection Bit

This bit selects an operation for the internal low-voltage detection.

Bit	Description
0	Reset.
1	Interrupt.

[bit29:27] Reserved: Reserved Bits

[bit26:25] LVDL1V: Internal Low-Voltage Detection Voltage Setting Bits

These bits set the detection voltage for internal low-voltage detection.

Bit26:25	Description
00	(product specification)
01	(product specification)
10	(product specification)
11	(product specification)

Note:

- *After the detection voltage is changed, low-voltage detection is unavailable until the low-voltage detection stability time elapses.*

[bit24] LVDL1E: Internal Low-Voltage Detection Operation Enable Bit

This bit enables the internal low-voltage detection operation.

Bit	Description
0	Stop operation.
1	Enable operation.

Note:

- *After this setting is changed from "0" (stop operation) to "1" (enable operation), low-voltage detection is unavailable until the low-voltage detection stability time elapses.*

[bit23] Reserved: Reserved Bit

[bit22] LVDL2S: Extended Internal Low-Voltage Detection Operation Selection Bit

This bit selects an operation for the extended internal low-voltage detection.

Bit	Description
0	Reset.
1	Interrupt.

[bit21:19] Reserved: Reserved Bits

[bit18:17] LVDL2V: Extended Internal Low-Voltage Detection Voltage Setting Bits

These bits set the detection voltage for extended internal low-voltage detection.

Bit18:17	Description
00	(product specification)
01	(product specification)
10	(product specification)
11	(product specification)

Note:

- *After the detection voltage is changed, low-voltage detection is unavailable until the low-voltage detection stability time elapses.*

[bit16] LVDL2E: Extended Internal Low-Voltage Detection Operation Enable Bit

This bit enables the extended internal low-voltage detection operation.

Bit	Description
0	Stop operation.
1	Enable operation.

Note:

- *After this setting is changed from "0" (stop operation) to "1" (enable operation), low-voltage detection is unavailable until the low-voltage detection stability time elapses.*

[bit15] Reserved: Reserved Bit

[bit14] LVDH1S: External Low-Voltage Detection Operation Selection Bit

This bit selects an operation for the external low-voltage detection.

Bit	Description
0	Reset.
1	Interrupt.

[bit13] Reserved: Reserved Bit

[bit12:9] LVDH1V: External Low-Voltage Detection Voltage Setting Bits

These bits set the detection voltage for external low-voltage detection.

Bit12:9	Description
0000	(product specification)
0001	(product specification)
0010	(product specification)
0011	(product specification)
0100	(product specification)
0101	(product specification)
0110	(product specification)
0111	(product specification)
1xxx	(product specification)

Note:

- *After the detection voltage is changed, low-voltage detection is unavailable until the low-voltage detection stability time elapses.*

[bit8] LVDH1E: External Low-Voltage Detection Operation Enable Bit

This bit sets the external low-voltage detection operation.

Bit	Description
0	Stop operation.
1	Enable operation.

Note:

- *After this setting is changed from "0" (stop operation) to "1" (enable operation), low-voltage detection is unavailable until the low-voltage detection stability time elapses.*

[bit7] Reserved: Reserved Bit

[bit6] LVDH2S: Extended External Low-Voltage Detection Operation Selection Bit

This bit selects an operation for the extended external low-voltage detection.

Bit	Description
0	Reset.
1	Interrupt.

[bit5] Reserved: Reserved Bit

[bit4:1] LVDH2V: Extended External Low-Voltage Detection Voltage Setting Bits

These bits set the detection voltage for extended external low-voltage detection.

Bit4:1	Description
0000	(product specification)
0001	(product specification)
0010	(product specification)
0011	(product specification)
0100	(product specification)
0101	(product specification)
0110	(product specification)
0111	(product specification)
1xxx	(product specification)

Note:

- *After the detection voltage is changed, low-voltage detection is unavailable until the low-voltage detection stability time elapses.*

[bit0] LVDH2E: Extended External Low-Voltage Detection Operation Enable Bit

This bit sets the extended external low-voltage detection operation.

Bit	Description
0	Stop operation.
1	Enable operation.

Note:

- *After this setting is changed from "0" (stop operation) to "1" (enable operation), low-voltage detection is unavailable until the low-voltage detection stability time elapses.*

5.4.10. APP Clock Supervisor Setting Register (SYSC0_APPCSVCFGR)

The SYSC0_APPCSVCFGR register indicates the value for setting whether to enable or disable operation of each clock supervisor to be updated.

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	WPS							
Initial Value	00000000							

bit	23	22	21	20	19	18	17	16
Field	Reserved				SSCG3 CSVE	SSCG2 CSVE	SSCG1 CSVE	SSCG0 CSVE
R/W Attribute	R0,WX				R,WX	R,WX	R,WX	R,WX
Protection Attribute	WPS							
Initial Value	0000				0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	Reserved				PLL3 CSVE	PLL2 CSVE	PLL1 CSVE	PLL0 CSVE
R/W Attribute	R0,WX				R,WX	R,WX	R,WX	R,WX
Protection Attribute	WPS							
Initial Value	0000				0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	Reserved				SCR CSVE	CR CSVE	SO CSVE	MO CSVE
R/W Attribute	R0,WX				R,WX	R,WX	R,WX	R,WX
Protection Attribute	WPS							
Initial Value	0000				0	0	0	0

[bit31:20] Reserved: Reserved Bits

[bit19] SSCG3CSVE: SSCG PLL3 Clock Supervisor Enable Bit

This bit sets whether to enable or disable the SSCG PLL3 clock supervisor.

Bit	Description
0	Disable the clock supervisor.
1	Enable the clock supervisor.

[bit18] SSCG2CSVE: SSCG PLL2 Clock Supervisor Enable Bit

This bit sets whether to enable or disable the SSCG PLL2 clock supervisor.

Bit	Description
0	Disable the clock supervisor.
1	Enable the clock supervisor.

[bit17] SSCG1CSVE: SSCG PLL1 Clock Supervisor Enable Bit

This bit sets whether to enable or disable the SSCG PLL1 clock supervisor.

Bit	Description
0	Disable the clock supervisor.
1	Enable the clock supervisor.

[bit16] SSCG0CSVE: SSCG PLL0 Clock Supervisor Enable Bit

This bit sets whether to enable or disable the SSCG PLL0 clock supervisor.

Bit	Description
0	Disable the clock supervisor.
1	Enable the clock supervisor.

[bit15:12] Reserved: Reserved Bits

[bit11] PLL3CSVE: PLL3 Clock Supervisor Enable Bit

This bit sets whether to enable or disable the PLL3 clock supervisor.

Bit	Description
0	Disable the clock supervisor.
1	Enable the clock supervisor.

[bit10] PLL2CSVE: PLL2 Clock Supervisor Enable Bit

This bit sets whether to enable or disable the PLL2 clock supervisor.

Bit	Description
0	Disable the clock supervisor.
1	Enable the clock supervisor.

[bit9] PLL1CSVE: PLL1 Clock Supervisor Enable Bit

This bit sets whether to enable or disable the PLL1 clock supervisor.

Bit	Description
0	Disable the clock supervisor.
1	Enable the clock supervisor.

[bit8] PLL0CSVE: PLL0 Clock Supervisor Enable Bit

This bit sets whether to enable or disable the PLL0 clock supervisor.

Bit	Description
0	Disable the clock supervisor.
1	Enable the clock supervisor.

[bit7:4] Reserved: Reserved Bits

[bit3] SCRCSVE：Low-Speed CR Clock Supervisor Enable Bit

This bit displays whether to enable or disable the low-speed CR clock supervisor.

Bit	Description
0	Disable the clock supervisor.
1	Enable the clock supervisor.

[bit2] CRCSVE: High-Speed CR Clock Supervisor Enable Bit

This bit displays whether to enable or disable the high-speed CR clock supervisor.

Bit	Description
0	Disable the clock supervisor.
1	Enable the clock supervisor.

[bit1] SOCSVE: Sub Clock Supervisor Enable Bit

This bit sets whether to enable or disable the sub clock supervisor.

Bit	Description
0	Disable the clock supervisor.
1	Enable the clock supervisor.

[bit0] MOCSVE: Main Clock Supervisor Enable Bit

This bit sets whether to enable or disable the main clock supervisor.

Bit	Description
0	Disable the clock supervisor.
1	Enable the clock supervisor.

5.4.11. APP Regulator Setting Register (SYSC0_APPREGCFGR)

The SYSC0_APPREGCFGR register indicates the set values for the regulator mode and output voltage to be updated.

bit	31	8
Field	Reserved	
R/W Attribute	R0,WX	
Protection Attribute	WPS	
Initial Value	00000000_00000000_00000000	

bit	7	6	5	4	3	2	1	0
Field	RMSEL	Reserved						RVSEL
R/W Attribute	R,WX	R0,WX						R,WX
Protection Attribute	WPS							
Initial Value	0	000000						(product specification)

[bit31:8] Reserved: Reserved Bits

[bit7] RMSEL: Regulator Mode Setting Bit

This bit indicates the regulator mode setting.

Bit	Description
0	Main mode
1	Standby mode

[bit6:1] Reserved: Reserved Bits

[bit0] RVSEL: Regulator Output Voltage Setting Bits

These bits set the regulator output voltage.

Bit	Description
0	1.20V
1	1.25V

5.5. STS Profile Register Group (SYSC0)

The registers in this group are STS profile control setting registers.

5.5.1. STS Power Domain Setting Register (SYSC0_STSPDCFGR)

The SYSC0_STSPDCFGR register indicates the set value of power ON or OFF for each power domain.

bit	31	30	29	28	27	26	25	24
Field	Reserved						PD6_1 EN	PD6_0 EN
R/W Attribute	R0,WX						R,WX	R,WX
Protection Attribute	WPS							
Initial Value	000000						1	1

bit	23	22	21	20	19	18	17	16
Field	PD5_3 EN	PD5_2 EN	PD5_1 EN	PD5_0 EN	Reserved		PD4_1 EN	PD4_0 EN
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R0,WX		R,WX	R,WX
Protection Attribute	WPS							
Initial Value	1	1	1	1	00		1	1

bit	15	14	13	12	11	10	9	8
Field	Reserved			PD3 EN	Reserved			PD2 EN
R/W Attribute	R0,WX			R,WX	R0,WX			R,WX
Protection Attribute	WPS							
Initial Value	000			1	000			1

bit	7	6	5	4	3	2	1	0
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	WPS							
Initial Value	00000000							

[bit31:26] Reserved: Reserved Bits

[bit25] PD6_1EN: Power Domain 6_1 Power Supply Control Bit

This bit indicates the set value for the power supply control of power domain 6_1.

Bit	Description
0	Shut off power supply.
1	Supply power.

[bit24] PD6_0 EN: Power Domain 6_0 Power Supply Control Bit

This bit indicates the set value for the power supply control of power domain 6_0.

Bit	Description
0	Shut off power supply.
1	Supply power.

[bit23] PD5_3 EN: Power Domain 5_3 Power Supply Control Bit

This bit indicates the set value for the power supply control of power domain 5_3.

Bit	Description
0	Shut off power supply.
1	Supply power.

[bit22] PD5_2 EN: Power Domain 5_2 Power Supply Control Bit

This bit indicates the set value for the power supply control of power domain 5_2.

Bit	Description
0	Shut off power supply.
1	Supply power.

[bit21] PD5_1 EN: Power Domain 5_1 Power Supply Control Bit

This bit indicates the set value for the power supply control of power domain 5_1.

Bit	Description
0	Shut off power supply.
1	Supply power.

[bit20] PD5_0 EN: Power Domain 5_0 Power Supply Control Bit

This bit indicates the set value for the power supply control of power domain 5_0.

Bit	Description
0	Shut off power supply.
1	Supply power.

[bit19:18] Reserved: Reserved Bits

[bit17] PD4_1 EN: Power Domain 4_1 Power Supply Control Bit

This bit indicates the set value for the power supply control of power domain 4_1.

Bit	Description
0	Shut off power supply.
1	Supply power.

[bit16] PD4_0 EN: Power Domain 4_0 Power Supply Control Bit

This bit indicates the set value for the power supply control of power domain 4_0.

Bit	Description
0	Shut off power supply.
1	Supply power.

[bit15:13] Reserved: Reserved Bits

[bit12] PD3 EN: Power Domain 3 Power Supply Control Bit

This bit indicates the set value for the power supply control of power domain 3.

Bit	Description
0	Shut off power supply.
1	Supply power.

[bit11:9] Reserved: Reserved Bits

[bit8] PD2 EN: Power Domain 2 Power Supply Control Bit

This bit indicates the set value for the power supply control of power domain 2.

Bit	Description
0	Shut off power supply.
1	Supply power.

[bit7:0] Reserved: Reserved Bits

5.5.2. STS Clock Source Enable Register (SYSC0_STSCKSRER)

The SYSC0_STSCKSRER register indicates the value for setting whether to enable/disable oscillation of the source clock.

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	WPS							
Initial Value	00000000							

bit	23	22	21	20	19	18	17	16
Field	SSCG3 RDY	SSCG2 RDY	SSCG1 RDY	SSCG0 RDY	SSCG3 EN	SSCG2 EN	SSCG1 EN	SSCG0 EN
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	WPS							
Initial Value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	PLL3 RDY	PLL2 RDY	PLL1 RDY	PLL0 RDY	PLL3EN	PLL2EN	PLL1EN	PLL0EN
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	WPS							
Initial Value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	SOSC RDY	MOSC RDY	SCROSC RDY	CROSC RDY	SOSCEN	MOSC EN	SCROSC EN	CROSC EN
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	WPS							
Initial Value	0	0	0	0	0	1	1	1

[bit31:24] Reserved: Reserved Bits

[bit23] SSCG3RDY: SSCG PLL3 Clock Oscillation Stabilization Bit

This bit indicates the status of the SSCG PLL3 clock oscillation.

Bit	Description
0	Stabilization wait or oscillation stop state
1	Stable state

[bit22] SSCG2RDY: SSCG PLL2 Clock Oscillation Stabilization Bit

This bit indicates the status of the SSCG PLL2 clock oscillation.

Bit	Description
0	Stabilization wait or oscillation stop state
1	Stable state

[bit21] SSCG1RDY: SSCG PLL1 Clock Oscillation Stabilization Bit

This bit indicates the status of the SSCG PLL1 clock oscillation.

Bit	Description
0	Stabilization wait or oscillation stop state
1	Stable state

[bit20] SSCG0RDY: SSCG PLL0 Clock Oscillation Stabilization Bit

This bit indicates the status of the SSCG PLL0 clock oscillation.

Bit	Description
0	Stabilization wait or oscillation stop state
1	Stable state

[bit19] SSCG3EN: SSCG PLL3 Clock Oscillation Enable Bit

This bit indicates the set value for SSCG PLL3 clock oscillation control.

Bit	Description
0	Disable oscillation of the SSCG PLL3 clock.
1	Enable oscillation of the SSCG PLL3 clock.

[bit18] SSCG2EN: SSCG PLL2 Clock Oscillation Enable Bit

This bit indicates the set value for SSCG PLL2 clock oscillation control.

Bit	Description
0	Disable oscillation of the SSCG PLL2 clock.
1	Enable oscillation of the SSCG PLL2 clock.

[bit17] SSCG1EN: SSCG PLL1 Clock Oscillation Enable Bit

This bit indicates the set value for SSCG PLL1 clock oscillation control.

Bit	Description
0	Disable oscillation of the SSCG PLL1 clock.
1	Enable oscillation of the SSCG PLL1 clock.

[bit16] SSCG0EN: SSCG PLL0 Clock Oscillation Enable Bit

This bit indicates the set value for SSCG PLL0 clock oscillation control.

Bit	Description
0	Disable oscillation of the SSCG PLL0 clock.
1	Enable oscillation of the SSCG PLL0 clock.

[bit15] PLL3RDY: PLL3 Clock Oscillation Stabilization Bit

This bit indicates the status of the PLL3 clock oscillation.

Bit	Description
0	Stabilization wait or oscillation stop state
1	Stable state

[bit14] PLL2RDY: PLL2 Clock Oscillation Stabilization Bit

This bit indicates the status of the PLL2 clock oscillation.

Bit	Description
0	Stabilization wait or oscillation stop state
1	Stable state

[bit13] PLL1RDY: PLL1 Clock Oscillation Stabilization Bit

This bit indicates the status of the PLL1 clock oscillation.

Bit	Description
0	Stabilization wait or oscillation stop state
1	Stable state

[bit12] PLL0RDY: PLL0 Clock Oscillation Stabilization Bit

This bit indicates the status of the PLL0 clock oscillation.

Bit	Description
0	Stabilization wait or oscillation stop state
1	Stable state

[bit11] PLL3EN: PLL3 Clock Oscillation Enable Bit

This bit indicates the set value for PLL3 clock oscillation control.

Bit	Description
0	Disable oscillation of the PLL3 clock.
1	Enable oscillation of the PLL3 clock.

[bit10] PLL2EN: PLL2 Clock Oscillation Enable Bit

This bit indicates the set value for PLL2 clock oscillation control.

Bit	Description
0	Disable oscillation of the PLL2 clock.
1	Enable oscillation of the PLL2 clock.

[bit9] PLL1EN: PLL1 Clock Oscillation Enable Bit

This bit indicates the set value for PLL1 clock oscillation control.

Bit	Description
0	Disable oscillation of the PLL1 clock.
1	Enable oscillation of the PLL1 clock.

[bit8] PLL0EN: PLL0 Clock Oscillation Enable Bit

This bit indicates the set value for PLL0 clock oscillation control.

Bit	Description
0	Disable oscillation of the PLL0 clock.
1	Enable oscillation of the PLL0 clock.

[bit7] SOSCRDY: Sub Clock Oscillation Stabilization Bit

This bit indicates the status of the sub clock oscillation.

Bit	Description
0	Stabilization wait or oscillation stop state
1	Stable state

Note:

- *This bit is read as "1" if stabilization is completed and sub clock is stable.*

[bit6] MOSCRDY: Main Clock Oscillation Stabilization Bit

This bit indicates the status of the main clock oscillation.

Bit	Description
0	Stabilization wait or oscillation stop state
1	Stable state

Note:

- *This bit is read as "1" if stabilization is completed and main clock is stable.*

[bit5] SCROSCRDY: Low-Speed CR Clock Oscillation Stabilization Bit

This bit indicates the status of the low-speed CR clock oscillation.

Bit	Description
0	Stabilization wait or oscillation stop state
1	Stable state

Note:

- *This bit is read as "1" if stabilization is completed and Low-speed CR clock is stable.*

[bit4] CROSCRDY: High-Speed CR Clock Oscillation Stabilization Bit

This bit indicates the status of the high-speed CR clock oscillation.

Bit	Description
0	Stabilization wait or oscillation stop state
1	Stable state

Note:

- *This bit is read as "1" if stabilization is completed and High-speed CR clock is stable.*

[bit3] SOSCEN: Sub Clock Oscillation Enable Bit

This bit indicates the set value for sub clock oscillation control.

Bit	Description
0	Disable oscillation of the sub clock.
1	Enable oscillation of the sub clock.

[bit2] MOSCEN: Main Clock Oscillation Enable Bit

This bit indicates the set value for main clock oscillation control.

Bit	Description
0	Disable oscillation of the main clock.
1	Enable oscillation of the main clock.

[bit1] SCROSCEN: Low-Speed CR Clock Oscillation Enable Bit

This bit indicates the set value for low-speed CR clock oscillation control.

Bit	Description
0	Disable oscillation of the low-speed CR clock.
1	Enable oscillation of the low-speed CR clock.

[bit0] CROSCEN: High-Speed CR Clock Oscillation Enable Bit

This bit indicates the set value for high-speed CR clock oscillation control.

Bit	Description
0	Disable oscillation of the high-speed CR clock.
1	Enable oscillation of the high-speed CR clock.

5.5.3. STS Clock Selection Register (SYSC0_STSCKSELR)

The SYSC0_STSCKSELR register indicates the set values of the clock source for the clock domain.

bit	31	8
Field	Reserved	
R/W Attribute	R0,WX	
Protection Attribute	WPS	
Initial Value	00000000_00000000_00000000	

bit	7	6	5	4	3	2	1	0
Field	Reserved	CDMCUCCM			Reserved	CDMCUCCSL		
R/W Attribute	R0,WX	R,WX			R0,WX	R,WX		
Protection Attribute	WPS							
Initial Value	0	000			0	000		

[bit31:7] Reserved: Reserved Bits

[bit6:4] CDMCUCCM: Clock Domain MCUC Clock Selection Status Bits

These bits indicate the selection status of the source clock for the clock domain MCUC.

bit6:4	Description
000	High-speed CR clock
001	Low-speed CR clock
010	Main clock
011	Sub clock (Do not use this setting)
100	PLL0 clock
101	SSCG PLL0 clock
110	Setting Prohibited
111	Clock fixed at "L"

[bit3] Reserved: Reserved Bit

[bit2:0] CDMCUCCSL: Clock Domain MCUC Clock Selection Bits

These bits indicate the value for selection of the source clock for the clock domain MCUC.

bit2:0	Description
000	High-speed CR clock
001	Low-speed CR clock
010	Main clock
011	Sub clock (Do not use this setting)
100	PLL0 clock
101	SSCG PLL0 clock
110	Setting Prohibited
11x	Clock fixed at "L"

5.5.4. STS Clock Enable Register (SYSC0_STSCKER)

The SYSC0_STSCKER register indicates the set value for stopping or enabling the internal clock.

bit	31	8
Field	Reserved	
R/W Attribute	R0,WX	
Protection Attribute	WPS	
Initial Value	00000000_00000000_00000000	

bit	7	6	5	4	3	2	1	0
Field	Reserved						ENCLK MCUCP	ENCLK MCUCH
R/W Attribute	R0,WX						R,WX	R,WX
Protection Attribute	WPS							
Initial Value	000000						1	1

[bit31:2] Reserved: Reserved Bits

[bit1] ENCLKMCUCP: MCUconfig APB Clock Oscillation Enable Bit

This bit indicates the value for setting whether to enable/disable MCUconfig APB clock oscillation.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit0] ENCLKMCUCH: MCUconfig AHB Clock Oscillation Enable Bit

This bit indicates the value for setting whether to enable/disable MCUconfig AHB clock oscillation.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

5.5.5. STS Clock Division Register (SYSC0_STSCKDIVR)

The SYSC0_STSCKER register indicates the set value of a clock division ratio.

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	Reserved															
R/W Attribute	R0,WX															
Protection Attribute	WPS															
Initial Value	00000000_00000000															

bit	15	14	13	12	11	10	9	8
Field	Reserved				MCUCPDIV			
R/W Attribute	R0,WX				R,WX			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	7	6	5	4	3	2	1	0
Field	Reserved			MCUCHDIV				
R/W Attribute	R0,WX			R,WX				
Protection Attribute	WPS							
Initial Value	000			00000				

[bit31:12] Reserved: Reserved Bits

[bit11:8] MCUCPDIV: MCUconfig APB Clock Division Setting Bits

These bits indicate the set value of the division ratio of the PCOM clock from the HCOM clock.

bit11:8	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

[bit7:5] Reserved: Reserved Bits

[bit4:0] MCUCHDIV: MCUconfig AHB Clock Division Setting Bits

These bits indicate the set value of the division ratio from the source clock of the clock domain MCUC.

bit4:0	Description
0_0000	No division
0_0001	Divided by 2
0_0010	Divided by 3
...	...
11101	Divided by 30
11110	Divided by 31
11111	Divided by 32

5.5.6. STS PLLx Control Register (SYSC0_STSPLLxCNTR)

The SYSC0_STSPLLxCNTR register indicates the set values of the division ratio, multiplication rate, etc. for PLLx.

bit	31	30	29	28	27	26	25	24
Field	PLLx ISEL	Reserved						
R/W Attribute	R,WX	R0,WX						
Protection Attribute	WPS							
Initial Value	0	0000000						

bit	23	22	21	20	19	18	17	16
Field	PLLxDIVN							
R/W Attribute	R,WX							
Protection Attribute	WPS							
Initial Value	00001101							

bit	15	14	13	12	11	10	9	8
Field	Reserved				PLLxDIVM			
R/W Attribute	R0,WX				R,WX			
Protection Attribute	WPS							
Initial Value	0000				0001			

bit	7	6	5	4	3	2	1	0
Field	Reserved						PLLxDIVL	
R/W Attribute	R0,WX						R,WX	
Protection Attribute	WPS							
Initial Value	000000						00	

[bit31] PLLxISEL: PLLx Input Clock Selection Bit

This bit indicates the value for setting the clock to supply to PLLx.

Bit	Description
0	Main clock
1	High-speed CR clock

[bit30:24] Reserved: Reserved Bits

[bit23:16] PLLxDIVN: PLLx Clock N-Multiplication Rate Setting Bits

These bits indicate the set value of the N multiplication rate of the PLLx clock.

bit23:16	Description
0000_0000	Setting prohibited
...	...
0000_1100	Setting prohibited
0000_1101	Multiply by 13
0000_1110	Multiply by 14
...	...
1100_0110	Multiply by 198
1100_0111	Multiply by 199
1100_1000	Multiply by 200
1100_1001	Setting prohibited
...	...
1100_1111	Setting prohibited

[bit15:12] Reserved: Reserved Bits

[bit11:8] PLLxDIVM: PLLx Output Clock M-Division Ratio Setting Bits

These bits indicate the set value of the M division ratio of the PLLx output clock.

bit11:8	Description
000x	Divided by 2
0010	Divided by 4
0011	Divided by 6
0100	Divided by 8
0101	Divided by 10
0110	Divided by 12
0111	Divided by 14
1000	Divided by 16
1001	Divided by 18
1010	Divided by 20
1011	Divided by 22
1010	Divided by 24
1101	Divided by 26
1110	Divided by 28
1111	Divided by 30

[bit7:2] Reserved: Reserved Bits

[bit1:0] PLLxDIVL: PLLx Clock L-Division Ratio Setting Bits

These bits indicate the set value of the L division ratio of the PLLx input clock.

bit1:0	Description
00	No division
01	Divided by 2
10	Divided by 4
11	Divided by 6

5.5.7. STS SSCGx Control Register 0 (SYSC0_STSSSCGxCNTR0)

The SYSC0_STSSSCGxCNTR0 register indicates the set values of the division ratio, multiplication rate, etc. for SSCG PLLx.

bit	31	30	29	28	27	26	25	24
Field	SSCGx ISEL	Reserved						
R/W Attribute	R,WX	R0,WX						
Protection Attribute	WPS							
Initial Value	0	0000000						

bit	23	22	21	20	19	18	17	16
Field	SSCGxDIVN							
R/W Attribute	R,WX							
Protection Attribute	WPS							
Initial Value	00001101							

bit	15	14	13	12	11	10	9	8
Field	Reserved				SSCGxDIVM			
R/W Attribute	R0,WX				R,WX			
Protection Attribute	WPS							
Initial Value	0000				0001			

bit	7	6	5	4	3	2	1	0
Field	Reserved						SSCGxDIVL	
R/W Attribute	R0,WX						R,WX	
Protection Attribute	WPS							
Initial Value	000000						00	

[bit31] SSCGxISEL: SSCG PLLx Input Clock Selection Bit

This bit indicates the value for setting the clock to supply to SSCG PLLx.

Bit	Description
0	Main clock
1	High-speed CR clock

[bit30:24] Reserved: Reserved Bits

[bit23:16] SSCGxDIVN: SSCG PLLx Clock N-multiplier Setting Bits

These bits indicate the set value of the N multiplication rate of the SSCG PLLx clock.

bit23:16	Description
0000_0000	Setting prohibited
...	...
0000_1100	Setting prohibited
0000_1101	Multiply by 13
0000_1110	Multiply by 14
...	...
1100_0110	Multiply by 198
1100_0111	Multiply by 199
1100_1000	Multiply by 200
1100_1001	Setting prohibited
...	...
1100_1111	Setting prohibited

[bit15:12] Reserved: Reserved Bits

[bit11:8] SSCGxDIVM: SSCG PLLx Clock M-Division Setting Bits

These bits indicate the set value of the M division ratio of the SSCG PLLx output clock.

bit11:8	Description
000x	Divided by 2
0010	Divided by 4
0011	Divided by 6
0100	Divided by 8
0101	Divided by 10
0110	Divided by 12
0111	Divided by 14
1000	Divided by 16
1001	Divided by 18
1010	Divided by 20
1011	Divided by 22
1010	Divided by 24
1101	Divided by 26
1110	Divided by 28
1111	Divided by 30

[bit7:2] Reserved: Reserved Bits

[bit1:0] SSCGxDIVL: SSCG PLLx Input Clock Division Setting Bits

These bits indicate the set value of the L division ratio of the SSCG PLLx input clock.

bit1:0	Description
00	No division
01	Divided by 2
10	Divided by 4
11	Divided by 6

5.5.8. STS SSCGx Control Register 1 (SYSC0_STSSSCGxCNTR1)

The SYSC0_STSSSCGxCNTR1 register indicates the set values for modulation enable, the modulation mode, etc. for SSCG PLLx.

bit	31	30	29	28	27	26	25	24
Field	Reserved							SSCGx SSEN
R/W Attribute	R0,WX							R,WX
Protection Attribute	WPS							
Initial Value	0000000							0

bit	23	22	21	20	19	18	17	16
Field	Reserved					SSCGxFREQ		SSCGx MODE
R/W Attribute	R0,WX					R,WX		R,WX
Protection Attribute	WPS							
Initial Value	00000					00		0

bit	15	14	13	12	11	10	9	8
Field	Reserved						SSCGxRATE	
R/W Attribute	R0,WX						R,WX	
Protection Attribute	WPS							
Initial Value	000000						00	

bit	7	6	5	4	3	2	1	0
Field	SSCGxRATE							
R/W Attribute	R,WX							
Protection Attribute	WPS							
Initial Value	00101001							

[bit31:25] Reserved: Reserved Bits

[bit24] SSCGxSSEN: SSCG PLLx modulation Enable Setting Bit

This bit indicates the set value for SSCG PLLx clock modulation control.

Bit	Description
0	Disable modulation.
1	Enable modulation.

[bit23:19] Reserved: Reserved Bits

[bit18:17] SSCGxFREQ: SSCG PLLx Clock modulation Frequency Selection Bits

These bits indicate the set value of the modulation frequency of the SSCG PLLx clock.

bit18:17	Description
00	Fmod = (1/1024) x Fin
01	Fmod = (1/2048) x Fin
10 or 11	Fmod = (1/4096) x Fin

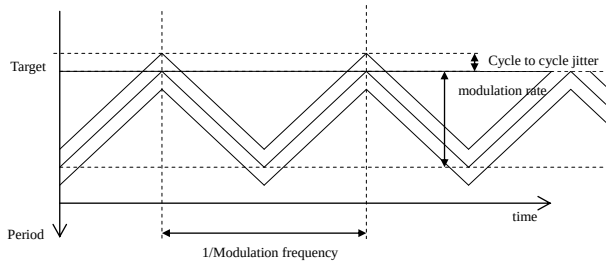
* Fin: SSCG reference clock frequency, Fmod: Modulation frequency

[bit16] SSCGxMODE: SSCG PLLx modulation Mode Setting Bit

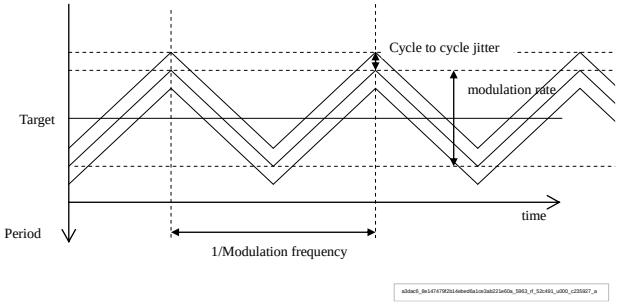
This bit indicates the set value for the modulation mode of the SSCG PLLx clock.

Bit	Description
0	Set Down spread mode.
1	Set Center spread mode.

- Down spread



- Center spread



[bit15:10] Reserved: Reserved Bits

[bit9:0] SSCGxRATE: SSCG PLLx Clock modulation Ratio Control Bits

These bits indicate the set value of the modulation ratio of the SSCG PLLx clock.

bit9:0	Description
00_0010_1001	Modulation ratio of 0.5%
00_0101_0010	Modulation ratio of 1.0%
00_1010_0100	Modulation ratio of 2.0%
00_1111_0110	Modulation ratio of 3.0%
01_0100_1000	Modulation ratio of 4.0%
01_1001_1010	Modulation ratio of 5.0%
Others	Setting prohibited

5.5.9. STS Low-Voltage Detection Setting Register (SYSC0_STSLVDCFGR)

The SYSC0_STSLVDCFGR register indicates the set values for each internal low-voltage detection.

bit	31	30	29	28	27	26	25	24
Field	LVDL1R	LVDL1S	Reserved			LVDL1V		LVDL1E
R/W Attribute	R,WX	R,WX	R0,WX			R,WX		R,WX
Protection Attribute	WPS							
Initial Value	(product specification)	(product specification)	000			(product specification)		(product specification)

bit	23	22	21	20	19	18	17	16
Field	LVDL2R	LVDL2S	Reserved			LVDL2V		LVDL2E
R/W Attribute	R,WX	R,WX	R0,WX			R,WX		R,WX
Protection Attribute	WPS							
Initial Value	(product specification)	(product specification)	000			(product specification)		(product specification)

bit	15	14	13	12	11	10	9	8
Field	LVDH1R	LVDH1S	Reserved	LVDH1V				LVDH1E
R/W Attribute	R,WX	R,WX	R0,WX	R,WX				R,WX
Protection Attribute	WPS							
Initial Value	(product specification)	(product specification)	0	(product specification)				(product specification)

bit	7	6	5	4	3	2	1	0
Field	LVDH2R	LVDH2S	Reserved	LVDH2V				LVDH2E
R/W Attribute	R,WX	R,WX	R0,WX	R,WX				R,WX
Protection Attribute	WPS							
Initial Value	(product specification)	(product specification)	0	(product specification)				(product specification)

[bit31] LVDL1R: Internal Low-Voltage Detection Operation Status Bit

This bit shows the operation status of internal low-voltage detection.

Bit	Description
0	Stabilization wait or monitoring stop status
1	Monitoring status

[bit30] LVDL1S: Internal Low-Voltage Detection Operation Selection Bit

This bit selects an operation when the internal low-voltage detection occurs.

Bit	Description
0	Reset.
1	Interrupt.

[bit29:27] Reserved: Reserved Bits

[bit26:25] LVDL1V: Internal Low-Voltage Detection Voltage Setting Bits

These bits set the detection voltage for internal low-voltage detection.

Bit	Description
00	(product specification)
01	(product specification)
10	(product specification)
11	(product specification)

Note:

- *After the detection voltage is changed, low-voltage detection is unavailable until the low-voltage detection stability time elapses.*

[bit24] LVDL1E: Internal Low-Voltage Detection Operation Enable Bit

This bit sets the internal low-voltage detection operation.

Bit	Description
0	Stop operation.
1	Enable operation.

Note:

- *After this setting is changed from "0" (stop operation) to "1" (enable operation), low-voltage detection is unavailable until the low-voltage detection stability time elapses.*

[bit23] LVDL2R: Extended Internal Low-Voltage Detection Operation Status Bit

This bit shows the operation status of extended internal low-voltage detection.

Bit	Description
0	Stabilization wait or monitoring stop status
1	Monitoring status

[bit22] LVDL2S: Extended Internal Low-Voltage Detection Operation Selection Bit

This bit selects an operation when the extended internal low-voltage detection occurs.

Bit	Description
0	Reset.
1	Interrupt.

[bit21:19] Reserved: Reserved Bits

[bit18:17] LVDL2V: Extended Internal Low-Voltage Detection Voltage Setting Bits

These bits set the detection voltage for extended internal low-voltage detection.

Bit	Description
00	(product specification)
01	(product specification)
10	(product specification)
11	(product specification)

Note:

- *After the detection voltage is changed, low-voltage detection is unavailable until the low-voltage detection stability time elapses.*

[bit16] LVDL2E: Extended Internal Low-Voltage Detection Operation Enable Bit

This bit sets the extended internal low-voltage detection operation.

Bit	Description
0	Stop operation.
1	Enable operation.

Note:

- *After this setting is changed from "0" (stop operation) to "1" (enable operation), low-voltage detection is unavailable until the low-voltage detection stability time elapses.*

[bit15] LVDH1R: External Low-Voltage Detection Operation Status Bit

This bit shows the operation status of external low-voltage detection.

Bit	Description
0	Stabilization wait or monitoring stop status
1	Monitoring status

[bit14] LVDH1S: External Low-Voltage Detection Operation Selection Bit

This bit selects an operation when the external low-voltage detection occurs.

Bit	Description
0	Reset.
1	Interrupt.

[bit13] Reserved: Reserved Bit

[bit12:9] LVDH1V: External Low-Voltage Detection Voltage Setting Bits

These bits set the detection voltage for external low-voltage detection.

Bit	Description
0000	(product specification)
0001	(product specification)
0010	(product specification)
0011	(product specification)
0100	(product specification)
0101	(product specification)
0110	(product specification)
0111	(product specification)
1xxx	(product specification)

Note:

- *After the detection voltage is changed, low-voltage detection is unavailable until the low-voltage detection stability time elapses.*

[bit8] LVDH1E: External Low-Voltage Detection Operation Enable Bit

This bit sets the external low-voltage detection operation.

Bit	Description
0	Stop operation.
1	Enable operation.

Note:

- *After this setting is changed from "0" (stop operation) to "1" (enable operation), low-voltage detection is unavailable until the low-voltage detection stability time elapses.*

[bit7] LVDH2R: Extended External Low-Voltage Detection Operation Status Bit

This bit shows the operation status of extended external low-voltage detection.

Bit	Description
0	Stabilization wait or monitoring stop status
1	Monitoring status

[bit6] LVDH2S: Extended External Low-Voltage Detection Operation Selection Bit

This bit selects an operation when the extended external low-voltage detection occurs.

Bit	Description
0	Reset.
1	Interrupt.

[bit5] Reserved: Reserved Bit

[bit4:1] LVDH2V: Extended External Low-Voltage Detection Voltage Setting Bits

These bits set the detection voltage for extended external low-voltage detection.

Bit	Description
0000	(product specification)
0001	(product specification)
0010	(product specification)
0011	(product specification)
0100	(product specification)
0101	(product specification)
0110	(product specification)
0111	(product specification)
1xxx	(product specification)

Note:

- *After changing the detection voltage, a low-voltage cannot be detected until the low-voltage detection stability time elapses.*

[bit0] LVDH2E: Extended External Low-Voltage Detection Operation Enable Bit

This bit sets the extended external low-voltage detection operation.

Bit	Description
0	Stop operation.
1	Enable operation.

Note:

- *After this setting is changed from "0" (stop operation) to "1" (enable operation), low-voltage detection is unavailable until the low-voltage detection stability time elapses.*

5.5.10. STS Clock Supervisor Setting Register (SYSC0_STSCSVCFGR)

The SYSC0_STSCSVCFGR register indicates the value for setting whether to enable/disable operation of each clock supervisor.

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	WPS							
Initial Value	00000000							

bit	23	22	21	20	19	18	17	16
Field	Reserved				SSCG3 CSVE	SSCG2 CSVE	SSCG1 CSVE	SSCG0 CSVE
R/W Attribute	R0,WX				R,WX	R,WX	R,WX	R,WX
Protection Attribute	WPS							
Initial Value	0000				0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	Reserved				PLL3 CSVE	PLL2 CSVE	PLL1 CSVE	PLL0 CSVE
R/W Attribute	R0,WX				R,WX	R,WX	R,WX	R,WX
Protection Attribute	WPS							
Initial Value	0000				0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	Reserved				SCR CSVE	FCR CSVE	SO CSVE	MO CSVE
R/W Attribute	R0,WX				R,WX	R,WX	R,WX	R,WX
Protection Attribute	WPS							
Initial Value	0000				0	0	0	0

[bit31:20] Reserved: Reserved Bits

[bit19] SSCG3CSVE: SSCG PLL3 Clock Supervisor Enable Bit

This bit indicates the value for setting whether to enable/disable the SSCG PLL3 clock supervisor.

Bit	Description
0	Disable the clock supervisor.
1	Enable the clock supervisor.

[bit18] SSCG2CSVE: SSCG PLL2 Clock Supervisor Enable Bit

This bit indicates the value for setting whether to enable/disable the SSCG PLL2 clock supervisor.

Bit	Description
0	Disable the clock supervisor.
1	Enable the clock supervisor.

[bit17] SSCG1CSVE: SSCG PLL1 Clock Supervisor Enable Bit

This bit indicates the value for setting whether to enable/disable the SSCG PLL1 clock supervisor.

Bit	Description
0	Disable the clock supervisor.
1	Enable the clock supervisor.

[bit16] SSCG0CSVE: SSCG PLL0 Clock Supervisor Enable Bit

This bit indicates the value for setting whether to enable/disable the SSCG PLL0 clock supervisor.

Bit	Description
0	Disable the clock supervisor.
1	Enable the clock supervisor.

[bit15:12] Reserved: Reserved Bits

[bit11] PLL3CSVE: PLL3 Clock Supervisor Enable Bit

This bit indicates the value for setting whether to enable/disable the PLL3 clock supervisor.

Bit	Description
0	Disable the clock supervisor.
1	Enable the clock supervisor.

[bit10] PLL2CSVE: PLL2 Clock Supervisor Enable Bit

This bit indicates the value for setting whether to enable/disable the PLL2 clock supervisor.

Bit	Description
0	Disable the clock supervisor.
1	Enable the clock supervisor.

[bit9] PLL1CSVE: PLL1 Clock Supervisor Enable Bit

This bit indicates the value for setting whether to enable/disable the PLL1 clock supervisor.

Bit	Description
0	Disable the clock supervisor.
1	Enable the clock supervisor.

[bit8] PLL0CSVE: PLL0 Clock Supervisor Enable Bit

This bit indicates the value for setting whether to enable/disable the PLL0 clock supervisor.

Bit	Description
0	Disable the clock supervisor.
1	Enable the clock supervisor.

[bit7:4] Reserved: Reserved Bits

[bit3] SCRCSVE：Low-Speed CR Clock Supervisor Enable Bit

This bit indicates the value for setting whether to enable/disable the low-speed CR clock supervisor.

Bit	Description
0	Disable the clock supervisor.
1	Enable the clock supervisor.

[bit2] FCRCSVE：High-Speed CR Clock Supervisor Enable Bit

This bit indicates the value for setting whether to enable/disable the high-speed CR clock supervisor.

Bit	Description
0	Disable the clock supervisor.
1	Enable the clock supervisor.

[bit1] SOCSVE: Sub Clock Supervisor Enable Bit

This bit indicates the value for setting whether to enable/disable the sub clock supervisor.

Bit	Description
0	Disable the clock supervisor.
1	Enable the clock supervisor.

[bit0] MOCSVE: Main Clock Supervisor Enable Bit

This bit indicates the value for setting whether to enable/disable the main clock supervisor.

Bit	Description
0	Disable the clock supervisor.
1	Enable the clock supervisor.

5.5.11. STS Regulator Setting Register (SYSC0_STSREGCFGR)

The SYSC0_STSREGCFGR register indicates the set values for the regulator mode and output voltage.

bit	31							8
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	WPS							
Initial Value	00000000_00000000_00000000							

bit	7	6	5	4	3	2	1	0
Field	RMSEL	Reserved						RVSEL
R/W Attribute	R,WX	R0,WX						R,WX
Protection Attribute	WPS							
Initial Value	0	000000						(product specification)

[bit31:8] Reserved: Reserved Bits

[bit7] RMSEL: Regulator Mode Setting Bit

This bit indicates the set value for the regulator mode.

Bit	Description
0	Main mode
1	Standby mode

[bit6:1] Reserved: Reserved Bits

[bit0] RVSEL: Regulator Output Voltage Setting Bits

These bits indicate the set value of the regulator output voltage.

Bit	Description
0	1.20V
1	1.25V

5.6. System Register Group (SYSC0)

The registers in this group are system control-related registers.

5.6.1. System ID Register (SYSC0_SYSIDR)

The SYSC0_SYSIDR register displays model information.

bit	31	0
Field	CHIPID	
R/W Attribute	R,WX	
Protection Attribute	WPS	
Initial Value	Product Specification	

[bit31:0] CHIPID: Chip ID Display Bits

Refer to the product hardware manual.

5.6.2. Platform ID Register (SYSC0_SYSPFIDR)

The SYSC0_SYSPFIDR register indicates platform information.

bit	31	0
Field	PFID	
R/W attribute	R,WX	
Protection Attribute	WPS	
Initial Value	Product Specification	

[bit31:0] PFID: Platform ID Display Bits

This bit is a specific fixed value. The value has no meaning.

Write access has no effect.

5.6.3. System Status Register (SYSC0_SYSSTSR)

The SYSC0_SYSSTSR register indicates each transition control status, the CPU status, the system operation status, and the transition control completion flag.

bit	31	8
Field	Reserved	
R/W Attribute	R0,WX	
Protection Attribute	WPS	
Initial Value	00000000_00000000_00000000	

bit	7	6	5	4	3	2	1	0
Field	PSSSTS0	RUNSTS0	PSSDF0	RUNDF0	Reserved		CPUSTS0	DVSTS0
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R0,WX		R,WX	R,WX
Protection Attribute	WPS							
Initial Value	0	0	0	0	00		0	1

[bit31:8] Reserved: Reserved Bits

[bit7] PSSSTS0: PSS profile Update Status Bit

This bit indicates the PSS profile update status.

Bit	Description
0	The PSS profile update has completed, or the profile was not updated.
1	The PSS profile is being updated.

Note:

- *This register does not become "1" immediately after the PSS profile update. The register becomes "1" after operation of the internal circuit.*

[bit6] RUNSTS0: RUN profile Update Status Bit

This bit indicates the RUN profile update status.

Bit	Description
0	The RUN profile update has completed, or the profile was not updated.
1	The RUN profile is being updated.

Note:

- *This register does not become "1" immediately after the RUN profile update. The register becomes "1" after operation of the internal circuit.*

[bit5] PSSDF0: PSS profile Update Completion Flag Bit

This bit is the PSS profile update completion flag.

Bit	Description
0	The PSS profile is being updated, or there is no update.
1	The PSS profile has been completed.

Notes:

- *To clear this register, write "1" in SYSC0_SYSICLR:PSSDFCLR0.*
- *The flags of this register and the PSSSTS0 register may appear to be "1" at circuit operation. This state shows that the PSS profile update has completed.*

[bit4] RUNDF0: RUN profile Update Completion (Main Status Control) Flag Bit

This bit is the RUN profile update completion (main status control) flag.

Bit	Description
0	The RUN profile is being updated, or there is no update.
1	The RUN profile has been completed.

Notes:

- *To clear this register, write "1" in SYSC0_SYSICLR:RUNDFCLR0.*
- *To use this register as an interrupt, write "1" in SYSC0_SYSINTER:RUNDIE0.*
- *The flags of this register and the RUNSTS0 register may appear to be "1" at circuit operation. This state shows that the RUN profile update has completed.*

[bit3:2] Reserved: Reserved Bits

[bit1] CPUSTS0: CPU0 Device Status Bit

This bit indicates the device status of CPU0.

Bit	Description
0	Operation status
1	WFI status

[bit0] DVSTS0: Device Status Bit

This bit indicates the device status.

Bit	Description
0	PSS
1	RUN

5.6.4. System Status Interrupt Enable Register (SYSC0_SYSINTER)

The SYSC0_SYSINTER register sets whether RUN profile execution completion interrupts are enabled.

bit	31	8
Field	Reserved	
R/W Attribute	R0,WX	
Protection Attribute	WPS	
Initial Value	00000000_00000000_00000000	

bit	7	6	5	4	3	2	1	0
Field	Reserved			RUNDIE0	Reserved			
R/W Attribute	R0,WX			R/W	R0,WX			
Protection Attribute	WPS							
Initial Value	000			0	0000			

[bit31:5] Reserved: Reserved Bits

[bit4] RUNDIE0: RUN profile Update Completion Interrupt Enable Bit

This bit enables RUN profile update completion interrupts.

Bit	Description
0	Disable
1	Enable

[bit3:0] Reserved: Reserved Bits

5.6.5. System Status Flag and Interrupt Clear Register (SYSC0_SYSICLR)

The SYSC0_SYSICLR register clears the RUN/PSS profile completion flag.

bit	31	8
Field	Reserved	
R/W Attribute	R0,WX	
Protection Attribute	WPS	
Initial Value	00000000_00000000_00000000	

bit	7	6	5	4	3	2	1	0
Field	Reserved		PSSDFCL R0	RUNDFCL R0	Reserved			
R/W Attribute	R0,WX		R0,W	R0,W	R0,WX			
Protection Attribute	WPS							
Initial Value	00		0	0	0000			

[bit31:6] Reserved: Reserved Bits

[bit5] PSSDFCLR0: PSS profile Update Completion Flag Clear Bit

This bit clears the PSS profile update completion flag.

Bit	Description
During write operation	When "0" is written: Invalid When "1" is written: Clear the interrupt factor.
During read operation	The read value is always "0".

[bit4] RUNDFCLR0: RUN profile Update Completion Flag Clear Bit

This bit clears the RUN profile update completion flag.

Bit	Description
During write operation	When "0" is written: Invalid When "1" is written: Clear the interrupt factor.
During read operation	The read value is always "0".

[bit3:0] Reserved: Reserved Bits

5.6.6. System Error Interrupt Factor Register 0 (SYSC0_SYSERRIR0)

The SYSC0_SYSERRIR0 register retains the factors of low-voltage detection (interrupt) and clock supervisor interrupt requests.

bit	31	30	29	28	27	26	25	24
Field	Reserved		LVDH2IF	LVDH1IF	Reserved		LVDL2IF	LVDL1IF
R/W Attribute	R0,WX		R,WX	R,WX	R0,WX		R,WX	R,WX
Protection Attribute	WPS							
Initial Value	00		0	0	00		0	0

bit	23	22	21	20	19	18	17	16
Field	Reserved				SSCG3IF	SSCG2IF	SSCG1IF	SSCG0IF
R/W Attribute	R0,WX				R,WX	R,WX	R,WX	R,WX
Protection Attribute	WPS							
Initial Value	0000				0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	Reserved				PLL3IF	PLL2IF	PLL1IF	PLL0IF
R/W Attribute	R0,WX				R,WX	R,WX	R,WX	R,WX
Protection Attribute	WPS							
Initial Value	0000				0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	Reserved						SOSCIF	MOSCIF
R/W Attribute	R0,WX						R,WX	R,WX
Protection Attribute	WPS							
Initial Value	000000						0	0

[bit31:30] Reserved: Reserved Bits

[bit29] LVDH2IF: Extended External Low-Voltage Detection Interrupt Request Bit

This bit retains the fact that an interrupt request was generated at the detection of extended external low voltage.

Bit	Description
0	No interrupt request is detected.
1	An interrupt request is detected.

[bit28] LVDH1IF: External Low-Voltage Detection Interrupt Request Bit

This bit retains the fact that an interrupt request was generated at the detection of external low voltage.

Bit	Description
0	No interrupt request is detected.
1	An interrupt request is detected.

[bit27:26] Reserved: Reserved Bits

[bit25] LVDL2IF: Extended Internal Low-Voltage Detection Interrupt Request Bit

This bit retains the fact that an interrupt request was generated at the detection of extended internal low voltage.

Bit	Description
0	No interrupt request is detected.
1	An interrupt request is detected.

[bit24] LVDL1IF: Internal Low-Voltage Detection Interrupt Request Bit

This bit retains the fact that an interrupt request was generated at the detection of internal low voltage.

Bit	Description
0	No interrupt request is detected.
1	An interrupt request is detected.

[bit23:20] Reserved: Reserved Bits

[bit19] SSCG3IF: SSCG PLL3 Abnormality Detection Error Interrupt Request Bit

This bit retains the fact that an interrupt request was generated for an SSCG PLL3 abnormality detection error.

Bit	Description
0	No interrupt request is detected.
1	An interrupt request is detected.

[bit18] SSCG2IF: SSCG PLL2 Abnormality Detection Error Interrupt Request Bit

This bit retains the fact that an interrupt request was generated for an SSCG PLL2 abnormality detection error.

Bit	Description
0	No interrupt request is detected.
1	An interrupt request is detected.

[bit17] SSCG1IF: SSCG PLL1 Abnormality Detection Error Interrupt Request Bit

This bit retains the fact that an interrupt request was generated for an SSCG PLL1 abnormality detection error.

Bit	Description
0	No interrupt request is detected.
1	An interrupt request is detected.

[bit16] SSCG0IF: SSCG PLL0 Abnormality Detection Error Interrupt Request Bit

This bit retains the fact that an interrupt request was generated for an SSCG PLL0 abnormality detection error.

Bit	Description
0	No interrupt request is detected.
1	An interrupt request is detected.

[bit15:12] Reserved: Reserved Bits

[bit11] PLL3IF: PLL3 Abnormality Detection Error Interrupt Request Bit

This bit retains the fact that an interrupt request was generated for a PLL3 abnormality detection error.

Bit	Description
0	No interrupt request is detected.
1	An interrupt request is detected.

[bit10] PLL2IF: PLL2 Abnormality Detection Error Interrupt Request Bit

This bit retains the fact that an interrupt request was generated for a PLL2 abnormality detection error.

Bit	Description
0	No interrupt request is detected.
1	An interrupt request is detected.

[bit9] PLL1IF: PLL1 Abnormality Detection Error Interrupt Request Bit

This bit retains the fact that an interrupt request was generated for a PLL1 abnormality detection error.

Bit	Description
0	No interrupt request is detected.
1	An interrupt request is detected.

[bit8] PLL0IF: PLL0 Abnormality Detection Error Interrupt Request Bit

This bit retains the fact that an interrupt request was generated for a PLL0 abnormality detection error.

Bit	Description
0	No interrupt request is detected.
1	An interrupt request is detected.

[bit7:2] Reserved: Reserved Bits

[bit1] SOSCIF: Sub Oscillation Abnormality Detection Error Interrupt Request Bit

This bit retains the fact that an interrupt request was generated for a sub oscillation abnormality detection error.

Bit	Description
0	No interrupt request is detected.
1	An interrupt request is detected.

[bit0] MOSCIF: Main Oscillation Abnormality Detection Error Interrupt Request Bit

This bit retains the fact that an interrupt request was generated for a main oscillation abnormality detection error.

Bit	Description
0	No interrupt request is detected.
1	An interrupt request is detected.

5.6.7. System Error Interrupt Factor Register 1 (SYSC0_SYSERRIR1)

The SYSC0_SYSERRIR1 register retains the factors of profile error interrupt requests.

bit	31							8
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	WPS							
Initial Value	00000000_00000000_00000000							

bit	7	6	5	4	3	2	1	0
Field	Reserved	PSSERRIF0	RUNWKERRIF0	RUNERRIF0	PSSENERRIF0	PSSTRGCIF0	RUNTRGERRIF	TRGERRIF
R/W Attribute	R0,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	WPS							
Initial Value	0	0	0	0	0	0	0	0

[bit31:7] Reserved: Reserved Bits

[bit6] PSSERRIF0: PSS profile Error Interrupt Request Bit

This bit retains the fact that an interrupt request was generated for a PSS profile error.

Bit	Description
0	No interrupt request is detected.
1	An interrupt request is detected.

[bit5] RUNWKERRIF0: RUN profile (PSS Recovery Time) Error Interrupt Request Bit

This bit retains the fact that an interrupt request was generated for a RUN profile error.

Bit	Description
0	No interrupt request is detected.
1	An interrupt request is detected.

[bit4] RUNERRIF0: RUN profile Error Interrupt Request Bit

This bit retains the fact that an interrupt request was generated for an error during a RUN profile update.

Bit	Description
0	No interrupt request is detected.
1	An interrupt request is detected.

[bit3] PSSENERRIF0: PSS profile Update Enable Write Error Interrupt Request Bit

This bit retains the fact that an interrupt request was generated for a write error in the PSS profile update enable register (SYSC0_PSSENR:PSSEN0).

Bit	Description
0	No interrupt request is detected.
1	An interrupt request is detected.

[bit2] PSSTRGCIF0: PSS Trigger cancel Interrupt Request Bit

This bit retains the fact that a cancel request was generated during a PSS profile update.

Bit	Description
0	No interrupt request is detected.
1	An interrupt request is detected.

[bit1] RUNTRGERRIF: RUN profile Update Enable Write Error Interrupt Request Bit

This bit retains the fact that an interrupt request was generated for a write error in the RUN profile update enable register.

Bit	Description
0	No interrupt request is detected.
1	An interrupt request is detected.

[bit0] TRGERRIF: Trigger Error Interrupt Request Bit

This bit retains that fact that an interrupt request was generated for a trigger error that caused an attempt to update the RUN profile again during a RUN profile update.

Bit	Description
0	No interrupt request is detected.
1	An interrupt request is detected.

5.6.8. System Error Interrupt Factor Clear Register 0 (SYSC0_SYSERRICLR0)

The SYSC0_SYSERRICLR0 register clears low-voltage detection (interrupt) and clock supervisor (interrupt) requests.

bit	31	30	29	28	27	26	25	24
Field	Reserved		LVDH2 ICLR	LVDH1 ICLR	Reserved		LVDL2 ICLR	LVDL1 ICLR
R/W Attribute	R0,WX		R0,W	R0,W	R0,WX		R0,W	R0,W
Protection Attribute	WPS							
Initial Value	00		0	0	00		0	0

bit	23	22	21	20	19	18	17	16
Field	Reserved				SSCG3 ICLR	SSCG2 ICLR	SSCG1 ICLR	SSCG0 ICLR
R/W Attribute	R0,WX				R0,W	R0,W	R0,W	R0,W
Protection Attribute	WPS							
Initial Value	0000				0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	Reserved				PLL3 ICLR	PLL2 ICLR	PLL1 ICLR	PLL0 ICLR
R/W Attribute	R0,WX				R0,W	R0,W	R0,W	R0,W
Protection Attribute	WPS							
Initial Value	0000				0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	Reserved						SOSC ICLR	MOSC ICLR
R/W Attribute	R0,WX						R0,W	R0,W
Protection Attribute	WPS							
Initial Value	000000						0	0

[bit31:30] Reserved: Reserved Bits

[bit29] LVDH2ICLR: Extended External Low-Voltage Detection Interrupt Factor Clear Bit

This bit clears the interrupt factor of the detection of extended external low voltage.

Bit	Description
During write operation	When "0" is written: Invalid When "1" is written: Clear the interrupt factor.
During read operation	The read value is always "0".

[bit28] LVDH1ICLR: External Low-Voltage Detection Interrupt Factor Clear Bit

This bit clears the interrupt factor of the detection of external low voltage.

Bit	Description
During write operation	When "0" is written: Invalid When "1" is written: Clear the interrupt factor.
During read operation	The read value is always "0".

[bit27:26] Reserved: Reserved Bits

[bit25] LVDL2ICLR: Extended Internal Low-Voltage Detection Interrupt Factor Clear Bit

This bit clears the interrupt factor of the detection of extended internal low voltage.

Bit	Description
During write operation	When "0" is written: Invalid When "1" is written: Clear the interrupt factor.
During read operation	The read value is always "0".

[bit24] LVDL1ICLR: Internal Low-Voltage Detection Interrupt Factor Clear Bit

This bit clears the interrupt factor of the detection of internal low voltage.

Bit	Description
During write operation	When "0" is written: Invalid When "1" is written: Clear the interrupt factor.
During read operation	The read value is always "0".

[bit23:20] Reserved: Reserved Bits

[bit19] SSCG3ICLR: SSCG PLL3 Abnormality Detection Error Interrupt Factor Clear Bit

This bit clears the interrupt factor of an SSCG PLL3 abnormality detection error.

Bit	Description
During write operation	When "0" is written: Invalid When "1" is written: Clear the interrupt factor.
During read operation	The read value is always "0".

[bit18] SSCG2ICLR: SSCG PLL2 Abnormality Detection Error Interrupt Factor Clear Bit

This bit clears the interrupt factor of an SSCG PLL2 abnormality detection error.

Bit	Description
During write operation	When "0" is written: Invalid When "1" is written: Clear the interrupt factor.
During read operation	The read value is always "0".

[bit17] SSCG1ICLR: SSCG PLL1 Abnormality Detection Error Interrupt Factor Clear Bit

This bit clears the interrupt factor of an SSCG PLL1 abnormality detection error.

Bit	Description
During write operation	When "0" is written: Invalid When "1" is written: Clear the interrupt factor.
During read operation	The read value is always "0".

[bit16] SSCG0ICLR: SSCG PLL0 Abnormality Detection Error Interrupt Factor Clear Bit

This bit clears the interrupt factor of an SSCG PLL0 abnormality detection error.

Bit	Description
During write operation	When "0" is written: Invalid When "1" is written: Clear the interrupt factor.
During read operation	The read value is always "0".

[bit15:12] Reserved: Reserved Bits

[bit11] PLL3ICLR: PLL3 Abnormality Detection Error Interrupt Factor Clear Bit

This bit clears the interrupt factor of a PLL3 abnormality detection error.

Bit	Description
During write operation	When "0" is written: Invalid When "1" is written: Clear the interrupt factor.
During read operation	The read value is always "0".

[bit10] PLL2ICLR: PLL2 Abnormality Detection Error Interrupt Factor Clear Bit

This bit clears the interrupt factor of a PLL2 abnormality detection error.

Bit	Description
During write operation	When "0" is written: Invalid When "1" is written: Clear the interrupt factor.
During read operation	The read value is always "0".

[bit9] PLL1ICLR: PLL1 Abnormality Detection Error Interrupt Factor Clear Bit

This bit clears the interrupt factor of a PLL1 abnormality detection error.

Bit	Description
During write operation	When "0" is written: Invalid When "1" is written: Clear the interrupt factor.
During read operation	The read value is always "0".

[bit8] PLL0ICLR: PLL0 Abnormality Detection Error Interrupt Factor Clear Bit

This bit clears the interrupt factor of a PLL0 abnormality detection error.

Bit	Description
During write operation	When "0" is written: Invalid When "1" is written: Clear the interrupt factor.
During read operation	The read value is always "0".

[bit7:2] Reserved: Reserved Bits

[bit1] SOSCICLR: Sub Oscillation Abnormality Detection Error Interrupt Factor Clear Bit

This bit clears the interrupt factor of a sub oscillation abnormality detection error.

Bit	Description
During write operation	When "0" is written: Invalid When "1" is written: Clear the interrupt factor.
During read operation	The read value is always "0".

[bit0] MOSCICLR: Main Oscillation Abnormality Detection Error Interrupt Factor Clear Bit

This bit clears the interrupt factor of a main oscillation abnormality detection error.

Bit	Description
During write operation	When "0" is written: Invalid When "1" is written: Clear the interrupt factor.
During read operation	The read value is always "0".

5.6.9. System Error Interrupt Factor Clear Register 1 (SYSC0_SYSERRICLR1)

The SYSC0_SYSERRICLR1 register clears profile error interrupt requests.

bit	31	8
Field	Reserved	
R/W Attribute	R0,WX	
Protection Attribute	WPS	
Initial Value	00000000_00000000_00000000	

bit	7	6	5	4	3	2	1	0
Field	Reserved	PSSERRICLR0	RUNWKERRICLR0	RUNERRICLR0	PSSENERICLR0	PSSTRGCICLR0	RUNTRGERRICLR	TRGERRICLR
R/W Attribute	R0,WX	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
Protection Attribute	WPS							
Initial Value	0	0	0	0	0	0	0	0

[bit31:7] Reserved: Reserved Bits

[bit6] PSSERRICLR0: PSS profile Error Interrupt Request Bit

This bit retains the fact that an interrupt request was generated for a PSS profile error.

Bit	Description
During write operation	When "0" is written: Invalid When "1" is written: Clear the interrupt factor.
During read operation	The read value is always "0".

[bit5] RUNWKERRICLR0: RUN profile (PSS Recovery Time) Error Interrupt Request Bit

This bit retains the fact that an interrupt request was generated for a RUN profile error.

Bit	Description
During write operation	When "0" is written: Invalid When "1" is written: Clear the interrupt factor.
During read operation	The read value is always "0".

[bit4] RUNERRICLR0: RUN profile Error Interrupt Request Bit

This bit retains the fact that an interrupt request was generated for an error during a RUN profile update.

Bit	Description
During write operation	When "0" is written: Invalid When "1" is written: Clear the interrupt factor.
During read operation	The read value is always "0".

[bit3] PSSENERRICLR0: PSS profile Update Enable Write Error Interrupt Request Bit

This bit retains the fact that an interrupt request was generated for a write error in the PSS profile update enable register (SYSC0_PSSENR:PSSEN0).

Bit	Description
During write operation	When "0" is written: Invalid When "1" is written: Clear the interrupt factor.
During read operation	The read value is always "0".

[bit2] PSSTRGCICLR0: PSS Trigger cancel Interrupt Request Bit

This bit retains the fact that a cancel request was generated in a PSS profile update.

Bit	Description
During write operation	When "0" is written: Invalid When "1" is written: Clear the interrupt factor.
During read operation	The read value is always "0".

[bit1] RUNTRGERRICLR: RUN profile Update Enable Write Error Interrupt Request Bit

This bit retains the fact that an interrupt request was generated for a write error in the RUN profile update enable register.

Bit	Description
During write operation	When "0" is written: Invalid When "1" is written: Clear the interrupt factor.
During Read Operation	The read value is always "0".

[bit0] TRGERRICLR: Trigger Error Interrupt Request Bit

This bit retains that fact that an interrupt request was generated for a trigger error that caused an attempt to update the RUN profile again during a RUN profile update.

Bit	Description
During write operation	When "0" is written: Invalid When "1" is written: Clear the interrupt factor.
During read operation	The read value is always "0".

5.6.10. Profile Status Register (SYSC0_SYSPROSTSR)

The SYSC0_SYSPROSTSR register indicates the error status of each profile.

bit	31	8
Field	Reserved	
R/W Attribute	R0,WX	
Protection Attribute	WPS	
Initial Value	00000000_00000000_00000000	

bit	7	6	5	4	3	2	1	0
Field	Reserved					PSSPSTS	RUNWKPSTS	RUNPSTS
R/W Attribute	R0,WX					R,WX	R,WX	R,WX
Protection Attribute	WPS							
Initial Value	00000					0	0	0

[bit31:3] Reserved: Reserved Bits

[bit2] PSSPSTS: PSS profile Setting Status Bit

This bit indicates the PSS profile setting status during a transition from RUN to PSS.

Bit	Description
0	There is no PSS profile error.
1	There is a PSS profile error.

Note:

- *We recommend confirming that there is no problem in the PSS profile by checking this register value before the transition to PSS.*

[bit1] RUNWKPSTS: RUN profile (PSS Recovery Time) Setting Status Bit

This bit indicates the RUN profile setting status during a transition from PSS to RUN.

Bit	Description
0	There is no RUN profile error.
1	There is a RUN profile error.

Note:

- *We recommend confirming that there is no problem in the RUN profile by checking this register value before the transition to PSS.*

[bit0] RUNPSTS: RUN profile Setting Status Bit

This bit indicates the RUN profile setting status.

Bit	Description
0	There is no RUN profile error.
1	There is a RUN profile error.

Note:

- *We recommend confirming that there is no problem in the RUN profile by checking this register value when updating the RUN profile.*

5.6.11. RUN Profile Error Flag Register (SYSC0_SYSRUNPEFR)

The SYSC0_SYSRUNPEFR register indicates each RUN profile error flag.

bit	31	8
Field	Reserved	
R/W Attribute	R0,WX	
Protection Attribute	WPS	
Initial Value	00000000_00000000_00000000	

bit	7	6	5	4	3	2	1	0
Field	Reserved	PEF6	PEF5	PEF4	PEF3	PEF2	PEF1	PEF0
R/W Attribute	R0,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	WPS							
Initial Value	0	0	0	0	0	0	0	0

[bit31:7] Reserved: Reserved Bits

[bit6:0] PEF<n>: Profile Error Flag Bits

These bits indicate the error status of each RUN profile. For detailed contents of profile errors, see "Profile error."

bit6:0	Status
0	There is no error.
1	There is an error.

<n> is a number from 6 to 0.

5.6.12. PSS Profile Error Flag Register (SYSC0_SYSPSSPEFR)

The SYSC0_SYSPSSPEFR register indicates each PSS profile error flag.

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	Reserved															
R/W Attribute	R0,WX															
Protection Attribute	WPS															
Initial Value	00000000_00000000															

bit	15	14	13	12	11	10	9	8
Field	Reserved					PEF10	PEF9	PEF8
R/W Attribute	R0,WX					R,WX	R,WX	R,WX
Protection Attribute	WPS							
Initial Value	00000					0	0	0

bit	7	6	5	4	3	2	1	0
Field	PEF7	PEF6	PEF5	PEF4	PEF3	Reserved	PEF1	PEF0
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R0,WX	R,WX	R,WX
Protection Attribute	WPS							
Initial Value	0	0	0	0	0	0	0	0

[bit31:11] [bit2] Reserved: Reserved Bits

[bit10:3,bit1:0] PEF<n>: Profile Error Flag Bits

These bits indicate each PSS profile error status. For detailed contents of profile errors, see "Profile error."

bit10:3, bit1:0	Status
0	There is no error.
1	There is an error.

<n> is a number from the following: 10 to 3 and 1 to 0.

5.7. Special Setting Register Group (SYSC0)

The registers in this group are special setting registers.

5.7.1. System Special Setting Register (SYSC0_SPECFGR)

The SYSC0_SPECFGR register sets special settings.

bit	31	30	29	28	27	26	25	24
Field	HOLDIO_PD6_1	HOLDIO_PD6_0	HOLDIO_PD5_3	HOLDIO_PD5_2	HOLDIO_PD5_1	HOLDIO_PD5_0	Reserved	HOLDIO_PD2
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R0,WX	R/W
Protection Attribute	WPS							
Initial Value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	PSSPADCTRL	IO3RSTC	IO35RSTC	Reserved				
R/W Attribute	R/W	R/W	R/W	R0,WX				
Protection Attribute	WPS							
Initial Value	0	1	1	00000				

bit	15	14	13	12	11	10	9	8
Field	Reserved						EXVRSTCNT	BRAMSC
R/W Attribute	R0,WX						R/W	R/W
Protection Attribute	WPS							
Initial Value	000000						0	0

bit	7	6	5	4	3	2	1	0
Field	EX5VRSTCNT				EX12VRSTCNT			
R/W Attribute	R/W				R/W			
Protection Attribute	WPS							
Initial Value	1110				1110			

[bit31] HOLDIO_PD6_1 : Setting Bit for Power Domain 6_1 HOLD Data Latch

This bit sets whether to retain I/O control in power domain 6_1 when shutting off the power supply.

Bit	Description
0	Do not retain control.
1	Retain control.

[bit30] HOLDIO_PD6_0 : Setting Bit for Power Domain 6_0 HOLD Data Latch

This bit sets whether to retain I/O control in power domain 6_0 when shutting off the power supply.

Bit	Description
0	Do not retain control.
1	Retain control.

[bit29] HOLDIO_PD5_3 : Setting Bit for Power Domain 5_3 HOLD Data Latch

This bit sets whether to retain I/O control in power domain 5_3 when shutting off the power supply.

Bit	Description
0	Do not retain control.
1	Retain control.

[bit28] HOLDIO_PD5_2 : Setting Bit for Power Domain 5_2 HOLD Data Latch

This bit sets whether to retain I/O control in power domain 5_2 when shutting off the power supply.

Bit	Description
0	Do not retain control.
1	Retain control.

[bit27] HOLDIO_PD5_1 : Setting Bit for Power Domain 5_1 HOLD Data Latch

This bit sets whether to retain I/O control in power domain 5_1 when shutting off the power supply.

Bit	Description
0	Do not retain control.
1	Retain control.

[bit25] Reserved: Reserved Bits

[bit26] HOLDIO_PD5_0 : Setting Bit for Power Domain 5_0 HOLD Data Latch

This bit sets whether to retain I/O control in power domain 5_0 when shutting off the power supply.

Bit	Description
0	Do not retain control.
1	Retain control.

[bit24] HOLDIO_PD2: Power Domain 2 HOLD Data Latch Setting Bit

This bit sets whether to retain I/O control in power domain 2 when shutting off the power supply.

Bit	Description
0	Do not retain control.
1	Retain control.

Note:

- *Make the decision on the model side about whether to use this setting.*

[bit23] PSSPADCTRL: PSS-Time Port configuring Bit

This bit sets whether to perform Hi-z control of I/O at the PSS time.

Bit	Description
0	Do not perform Hi-z control.
1	Perform Hi-z control.

[bit22] IO3RSTC：I/O3V Reset configuring Bit

This bit sets 3V I/O reset control.

Bit	Description
0	I/O reset released
1	I/O reset asserted

[bit21] IO35RSTC：I/O5/3V Reset configuring Bit

This bit sets 5/3V I/O reset control.

Bit	Description
0	I/O reset released
1	I/O reset asserted

[bit20:10] Reserved

[bit9] EXVRSTCNT：Reset Level configuring Bit for External Power Supply Control

This bit sets requested reset level to the external power supply control.

Bit	Description
0	External power supply control is reset by LVDH1
1	External power supply control is not reset by LVDH1

Note:

- *This bit is reset by RSTX_POWER. Refer Section 3.2.3 Reset of circuit in surrounding in CHAPTER RESET.*

[bit8] BRAMSC: Backup RAM Standby Setting Bit

This bit controls SLEEP setting for Backup RAM while PSS.

Bit	Description
0	Disable SLEEP control for Backup RAM while PSS
1	Enable SLEEP control for Backup RAM while PSS

[bit7:4] EX5VRSTCNT: 5/3.3V External Power Supply Stabilization Time Setting Bits

These bits set the stabilization time for 5/3.3V external power supply.

bit7:4	High-Speed CR Clock	
	4MHz (ms)	8MHz (ms)
0000	1.0	0.5
0001	2.0	1.0
0010	3.0	1.5
0011	4.0	2.0
0100	5.0	2.5
0101	6.0	3.0
0110	7.0	3.5
0111	8.0	4.0
1000	9.0	4.5
1001	10.0	5.0
1010	12.0	6.0
1011	14.0	7.0
1100	16.0	8.0
1101	18.0	9.0
1110	20.0	10.0
1111	30.0	15.0

Note:

- *These bits are reset by RSTX_POWER. Refer Section 3.2.3 Reset of circuit in surrounding in CHAPTER RESET.*

[bit3:0] EX12VRSTCNT: 1.2V External Power Supply Stabilization Time Setting Bits

These bits set the stabilization time for 1.2V external power supply.

bit3:0	High-Speed CR Clock	
	4MHz (ms)	8MHz (ms)
0000	1.0	0.5
0001	2.0	1.0
0010	3.0	1.5
0011	4.0	2.0
0100	5.0	2.5
0101	6.0	3.0
0110	7.0	3.5
0111	8.0	4.0
1000	9.0	4.5
1001	10.0	5.0
1010	12.0	6.0
1011	14.0	7.0
1100	16.0	8.0
1101	18.0	9.0
1110	20.0	10.0
1111	30.0	15.0

Note:

- *These bits are reset by RSTX_POWER. Refer Section 3.2.3 Reset of circuit in surrounding in CHAPTER RESET.*

5.8. Debug Register Group (SYSC0)

The registers in this group are debug-related registers.

5.8.1. JTAG Detection Register (SYSC0_JTAGDETECT)

The SYSC0_JTAGDETECT register indicates the status of the connection with the debugger.

bit	31	8
Field	Reserved	
R/W Attribute	R0,WX	
Protection Attribute	-	
Initial Value	00000000_00000000_00000000	

bit	7	6	5	4	3	2	1	0
Field	Reserved							DBGCON
R/W Attribute	R0,WX							R,WX
Protection Attribute	-							
Initial Value	0000000							0

[bit31:1] Reserved: Reserved Bits

[bit0] DBGCON: Debugger Connection Status Bit

This bit indicates the status of the connection with the debugger.

Bit	Description
0	Not connected
1	Connected

5.8.2. JTAG Setting Register (SYSC0_JTAGCNFG)

The SYSC0_JTAGCNFG register indicates the completion of debugger settings (for BootROM).

bit	31	8
Field	Reserved	
R/W Attribute	R0,WX	
Protection Attribute	-	
Initial Value	00000000_00000000_00000000	

bit	7	6	5	4	3	2	1	0
Field	Reserved							Reserved
R/W Attribute	R0,WX							R/W
Protection Attribute	-							
Initial Value	0000000							1

[bit31:1] Reserved: Reserved Bits

[bit0] Reserved: Reserved Bit

Do not write any data to this bit.

5.8.3. JTAG Recovery Register (SYSC0_JTAGWAKEUP)

The SYSC0_JTAGWAKEUP register sets whether to enable recovery of a debugger connection.

bit	31	8
Field	Reserved	
R/W Attribute	R0,WX	
Protection Attribute	-	
Initial Value	00000000_00000000_00000000	

bit	7	6	5	4	3	2	1	0
Field	Reserved							DBGWKEN
R/W Attribute	R0,WX							R/W
Protection Attribute	-							
Initial Value	0000000							0

[bit31:1] Reserved: Reserved Bits

[bit0] DBGWKEN: Debugger wakeup Enable Bit

This bit sets recovery of a debugger connection.

Bit	Description
0	Mask recovery.
1	Enable recovery.

Notes:

- *To enable JTAG-Wakeup when connecting the debugger, write "1" to this bit.*
- *If recovery is masked by this bit, JTAG-Wakeup recovery is disabled.*

5.9. MCU_Config_Group Interrupt Batch Read Register Group (MCG_IRS)

The multiple interrupt requests that are assigned to MCU_CONFIG_GROUP can be collectively read from the registers in this group. When returning from the standby, It is possible to confirm from these registers which interrupt requests occur.

5.9.1. MCU_Config_Group Interrupt Request Status Register0 (MCG_IRSR0)

The MCG_IRSR0 register displays interrupt signals in the MCU_CONFIG_GROUP at once.

bit	31	30	29	28	27	26	25	24
Field	Reserved							IRQ_PW
R/W Attribute	R0,WX							R,WX
Protection Attribute	-							
Initial Value	0000000							0

bit	23	22	21	20	19	18	17	16
Field	Reserved				IRQ_SCT_SUB	IRQ_SCT_MAIN	IRQ_SCT_SCR	IRQ_SCT_CR
R/W Attribute	R0,WX				R,WX	R,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	0000				0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	Reserved				Reserved	IRQ_RTC	IRQ_SCU	IRQ_HWDG
R/W Attribute	R0,WX				R0,WX	R,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	0000				0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	Reserved			Reserved	NMI_EXTINT	NMI_HWDG	NMI_SCU	NMI_LVD
R/W Attribute	R0,WX			R0,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	000			0	0	0	0	0

[bit31:25] Reserved: Reserved Bits

[bit24] IRQ_PW: Partial wakeup Interrupt Detection Status

Bit	Description
0	The interrupt does not occur.
1	The interrupt occurs.

[bit23:20] Reserved: Reserved Bits

[bit19] IRQ_SCT_SUB: Sub Source Clock Timer Interrupt Detection Status

Bit	Description
0	The interrupt does not occur.
1	The interrupt occurs.

[bit18] IRQ_SCT_MAIN: Main Source Clock Timer Interrupt Detection Status

Bit	Description
0	The interrupt does not occur.
1	The interrupt occurs.

[bit17] IRQ_SCT_SCR: Low-Speed Source Clock Timer Interrupt Detection Status

Bit	Description
0	The interrupt does not occur.
1	The interrupt occurs.

[bit16] IRQ_SCT_CR: High-Speed Source Clock Timer Interrupt Detection Status

Bit	Description
0	The interrupt does not occur.
1	The interrupt occurs.

[bit15:11] Reserved: Reserved Bits

[bit10] IRQ_RTC: RTC 0.5 Seconds Interrupt Detection Status

Bit	Description
0	The interrupt does not occur.
1	The interrupt occurs.

[bit9] IRQ_SCU: RUN profile Update Complete Interrupt Status

Bit	Description
0	The interrupt does not occur.
1	The interrupt occurs.

[bit8] IRQ_HWDG: Hardware Watchdog Timer advance warning Interrupt Status

Bit	Description
0	The interrupt does not occur.
1	The interrupt occurs.

[bit7:4] Reserved: Reserved Bits

[bit3] NMI_EXTINT: NMI Detection Interrupt Status (NMI Level)

Bit	Description
0	The interrupt does not occur.
1	The interrupt occurs.

[bit2] NMI_HWDG: Hardware Watchdog Timer Detection Interrupt Status (NMI Level)

Bit	Description
0	The interrupt does not occur.
1	The interrupt occurs.

[bit1] NMI_SCU: Profile Error Detection Interrupt Status (NMI Level)

Bit	Description
0	The interrupt does not occur.
1	The interrupt occurs.

[bit0] NMI_LVD: Low Power Detection Interrupt Status (NMI Level)

Bit	Description
0	The interrupt does not occur.
1	The interrupt occurs.

5.9.2. MCU_Config_Group Interrupt Request Status Register1 (MCG_IRSR1)

The MCG_IRSR1 register displays interrupt signals in the MCU_CONFIG_GROUP at once.

bit	31	0
Field	IRQ_EXTINT[31:0]	
R/W Attribute	R,WX	
Protection Attribute	-	
Initial Value	00000000_00000000_00000000_00000000	

[bit31:0] IRQ_EXTINT: External Interrupt Detection Status

Bit	Description
0	The interrupt does not occur.
1	The interrupt occurs.

5.9.3. MCU_Config_Group Interrupt Request Status Register2 (MCG_IRSR2)

The MCG_IRSR2 register displays interrupt signals in the MCU_CONFIG_GROUP at once.

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	-							
Initial Value	00000000							

bit	23	22	21	20	19	18	17	16
Field	Reserved					IRQ_RLT2	IRQ_RLT1	IRQ_RLT0
R/W Attribute	R0,WX					R,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	00000					0	0	0

bit	15	14	13	12	11	10	9	8
Field	Reserved						IRQ_CRCAL	IRQ_EICU
R/W Attribute	R0,WX						R,WX	R,WX
Protection Attribute	-							
Initial Value	000000						0	0

bit	7	6	5	4	3	2	1	0
Field	Reserved			IRQ_RAMIC	IRQ_RAMTE	IRQ_RAMTC	IRQ_RAMSE	NMI_RAMDE
R/W Attribute	R0,WX			R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	000			0	0	0	0	0

[bit31:19] Reserved: Reserved Bits

[bit18] IRQ_RLT2: Reload Timer ch2 Interrupt Status

Bit	Description
0	The interrupt does not occur.
1	The interrupt occurs.

[bit17] IRQ_RLT1: Reload Timer ch1 Interrupt Status

Bit	Description
0	The interrupt does not occur.
1	The interrupt occurs.

[bit16] IRQ_RLT0: Reload Timer ch0 Interrupt Status

Bit	Description
0	The interrupt does not occur.
1	The interrupt occurs.

[bit15:10] Reserved: Reserved Bits

[bit9] IRQ_CRCAL: CR Calibration Complete Interrupt Status

Bit	Description
0	The interrupt does not occur.
1	The interrupt occurs.

[bit8] IRQ_EICU: EICU Complete Interrupt Status

Bit	Description
0	The interrupt does not occur.
1	The interrupt occurs.

[bit7:5] Reserved: Reserved Bits

[bit4] IRQ_RAMIC: BackUP-RAM Initialization End Interrupt Status

Bit	Description
0	The interrupt does not occur.
1	The interrupt occurs.

[bit3] IRQ_RAMTE: BackUP-RAM Diagnosis Error Interrupt Status

Bit	Description
0	The interrupt does not occur.
1	The interrupt occurs.

[bit2] IRQ_RAMTC: BackUP-RAM Diagnosis End Interrupt Status

Bit	Description
0	The interrupt does not occur.
1	The interrupt occurs.

[bit1] IRQ_RAMSE: BackUP-RAM Single-Bit Error Interrupt Status

Bit	Description
0	The interrupt does not occur.
1	The interrupt occurs.

[bit0] NMI_RAMDE: BackUP-RAM double-Bit Error Interrupt Status (NMI Level)

Bit	Description
0	The interrupt does not occur.
1	The interrupt occurs.

5.9.4. MCU_Config_Group Interrupt Request Status Register3 (MCG_IRSR3)

The MCG_IRSR3 register displays interrupt signals in the MCU_CONFIG_GROUP at once.

bit	31	30	29	28	27	26	25	24
Field	Reserved	IRQ_MCAN2_INT1	IRQ_MCAN1_INT1	IRQ_MCAN0_INT1	Reserved	IRQ_MCAN2_INT0	IRQ_MCAN1_INT0	IRQ_MCAN0_INT0
R/W Attribute	R0,WX	R,WX	R,WX	R,WX	R0,WX	R,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	Reserved	IRQ_MCAN2_SE	IRQ_MCAN1_SE	IRQ_MCAN0_SE	Reserved	NMI_MCAN2_DE	NMI_MCAN1_DE	NMI_MCAN0_DE
R/W Attribute	R0,WX	R,WX	R,WX	R,WX	R0,WX	R,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	Reserved					IRQ_MFS2_SIRQ	IRQ_MFS1_SIRQ	IRQ_MFS0_SIRQ
R/W Attribute	R0,WX					R,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	00000					0	0	0

bit	7	6	5	4	3	2	1	0
Field	Reserved	IRQ_MFS2_RIRQ	IRQ_MFS1_RIRQ	IRQ_MFS0_RIRQ	Reserved	IRQ_MFS2_TIRQ	IRQ_MFS1_TIRQ	IRQ_MFS0_TIRQ
R/W Attribute	R0,WX	R,WX	R,WX	R,WX	R0,WX	R,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

[bit31] Reserved: Reserved Bit

[bit30] IRQ_MCAN2_INT1: CAN-FD ch2 Interrupt 1 Status

Bit	Description
0	The interrupt does not occur.
1	The interrupt occurs.

[bit29] IRQ_MCAN1_INT1: CAN-FD ch1 Interrupt 1 Status

Bit	Description
0	The interrupt does not occur.
1	The interrupt occurs.

[bit28] IRQ_MCAN0_INT1: CAN-FD ch0 Interrupt 1 Status

Bit	Description
0	The interrupt does not occur.
1	The interrupt occurs.

[bit27] Reserved: Reserved Bit

[bit26] IRQ_MCAN2_INT0: CAN-FD ch2 Interrupt 0 Status

Bit	Description
0	The interrupt does not occur.
1	The interrupt occurs.

[bit25] IRQ_MCAN1_INT0: CAN-FD ch1 Interrupt 0 Status

Bit	Description
0	The interrupt does not occur.
1	The interrupt occurs.

[bit24] IRQ_MCAN0_INT0: CAN-FD ch0 Interrupt 0 Status

Bit	Description
0	The interrupt does not occur.
1	The interrupt occurs.

[bit23] Reserved: Reserved Bit

[bit22] IRQ_MCAN2_SE: CAN-FD ch2 RAM ECC Single-Bit Error Detection Interrupt Status

Bit	Description
0	The interrupt does not occur.
1	The interrupt occurs.

[bit21] IRQ_MCAN1_SE: CAN-FD ch1 RAM ECC Single-Bit Error Detection Interrupt Status

Bit	Description
0	The interrupt does not occur.
1	The interrupt occurs.

[bit20] IRQ_MCAN0_SE: CAN-FD ch0 RAM ECC Single-Bit Error Detection Interrupt Status

Bit	Description
0	The interrupt does not occur.
1	The interrupt occurs.

[bit19] Reserved: Reserved Bit

[bit18] NMI_MCAN2_DE: CAN-FD ch2 RAM ECC double-Bit Error Detection Interrupt Status (NMI Level)

Bit	Description
0	The interrupt does not occur.
1	The interrupt occurs.

[bit17] NMI_MCAN1_DE: CAN-FD ch1 RAM ECC double-Bit Error Detection Interrupt Status (NMI Level)

Bit	Description
0	The interrupt does not occur.
1	The interrupt occurs.

[bit16] NMI_MCAN0_DE: CAN-FD ch0 RAM ECC double-Bit Error Detection Interrupt Status (NMI Level)

Bit	Description
0	The interrupt does not occur.
1	The interrupt occurs.

[bit15:11] Reserved: Reserved Bits

[bit10] IRQ_MFS2_SIRQ: MFS ch2 Sync Field Detection Interrupt Status

Bit	Description
0	The interrupt does not occur.
1	The interrupt occurred.

[bit9] IRQ_MFS1_SIRQ: MFS ch1 Sync Field Detection Interrupt Status

Bit	Description
0	The interrupt does not occur.
1	The interrupt occurs.

[bit8] IRQ_MFS0_SIRQ: MFS ch0 Sync Field Detection Interrupt Status

Bit	Description
0	The interrupt does not occur.
1	The interrupt occurs.

[bit7] Reserved: Reserved Bit

[bit6] IRQ_MFS2_RIRQ: MFS ch2 Reception Interrupt Status

Bit	Description
0	The interrupt does not occur.
1	The interrupt occurs.

[bit5] IRQ_MFS1_RIRQ: MFS ch1 Reception Interrupt Status

Bit	Description
0	The interrupt does not occur.
1	The interrupt occurs.

[bit4] IRQ_MFS0_RIRQ: MFS ch0 Reception Interrupt Status

Bit	Description
0	The interrupt does not occur.
1	The interrupt occurs.

[bit3] Reserved: Reserved Bit

[bit2] IRQ_MFS2_TIRQ: MFS ch2 Transmission Interrupt Status

Bit	Description
0	The interrupt does not occur.
1	The interrupt occurs.

[bit1] IRQ_MFS1_TIRQ: MFS ch1 Transmission Interrupt Status

Bit	Description
0	The interrupt does not occur.
1	The interrupt occurs.

[bit0] IRQ_MFS0_TIRQ: MFS ch0 Transmission Interrupt Status

Bit	Description
0	The interrupt does not occur.
1	The interrupt occurs.

5.9.5. MCU_Config_Group Interrupt Request Status Register4 (MCG_IRSR4)

The MCG_IRSR4 register displays an interrupt signals in MCU_CONFIG_GROUP at once.

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	-							
Initial Value	00000000							

bit	23	22	21	20	19	18	17	16
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	-							
Initial Value	00000000							

bit	15	14	13	12	11	10	9	8
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	-							
Initial Value	00000000							

bit	7	6	5	4	3	2	1	0
Field	Reserved	IRQ_MFS2_REIRQ	IRQ_MFS1_REIRQ	IRQ_MFS0_REIRQ	Reserved	IRQ_MFS2_TEIRQ	IRQ_MFS1_TEIRQ	IRQ_MFS0_TEIRQ
R/W Attribute	R0,WX	R,WX	R,WX	R,WX	R0,WX	R,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

[bit31:7] Reserved: Reserved Bits

[bit6] IRQ_MFS2_REIRQ: MFS ch2 Reception Error Interrupt Status

Bit	Description
0	The interrupt does not occur.
1	The interrupt occurs.

[bit5] IRQ_MFS1_REIRQ: MFS ch1 Reception Error Interrupt Status

Bit	Description
0	The interrupt does not occur.
1	The interrupt occurs.

[bit4] IRQ_MFS0_REIRQ: MFS ch0 Reception Error Interrupt Status

Bit	Description
0	The interrupt does not occur.
1	The interrupt occurs.

[bit3] Reserved: Reserved Bit

[bit2] IRQ_MFS2_TEIRQ: MFS ch2 Transmission Error Interrupt Status

Bit	Description
0	The interrupt does not occur.
1	The interrupt occurs.

[bit1] IRQ_MFS1_TEIRQ: MFS ch1 Transmission Error Interrupt Status

Bit	Description
0	The interrupt does not occur.
1	The interrupt occurs.

[bit0] IRQ_MFS0_TEIRQ: MFS ch0 Transmission Error Interrupt Status

Bit	Description
0	The interrupt does not occur.
1	The interrupt occurs.

5.10. Protection Register Group (SYSC1)

The register in this group is a control register for accessing SYSC1 registers.

5.10.1. Protection Key Setting Register (SYSC1_PROTKEYR)

The SYSC1_PROTKEYR register has settings for unlocking the protection of SYSC1 registers.

bit	31	0
Field	PROTKEY	
R/W Attribute	R,W	
Protection Attribute	WP	
Initial Value	00000000_00000000_00000000_00000000	

[bit31:0]PROTKEY: Protection Unlock Setting Bits

These bits compose the register for unlocking protection keys.

Write

bit31:0	Description
0x5CAC_CE55	Unlock
Other than above	Invalid

Read

bit31:0	Description
0x0000_0000	Locked state
0xFFFF_FFFF	Unlocked state

Note:

- *Only word access is valid to this register. Any other access is invalid.*

5.11. RUN Profile Register Group (SYSC1)

The registers in this group are RUN profile control setting registers.

5.11.1. RUN Clock Selection Register 0 (SYSC1_RUNCKSELR0)

The SYSC1_RUNCKSELR0 register sets the clock source for the clock domain.

bit	31	30	29	28	27	26	25	24
Field	Reserved				HSSPICSL			
R/W Attribute	R0,WX				R/W			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	23	22	21	20	19	18	17	16
Field	Reserved			LAPP1ACSL	Reserved			LAPP0ACSL
R/W Attribute	R0,WX			R/W	R0,WX			R/W
Protection Attribute	WPS							
Initial Value	000			0	000			0

bit	15	14	13	12	11	10	9	8
Field	Reserved			LCP1ACSL	Reserved			LCP0ACSL
R/W Attribute	R0,WX			R/W	R0,WX			R/W
Protection Attribute	WPS							
Initial Value	000			0	000			0

bit	7	6	5	4	3	2	1	0
Field	Reserved					CD0CSL		
R/W Attribute	R0,WX					R/W		
Protection Attribute	WPS							
Initial Value	00000					000		

[bit31:28] Reserved: Reserved Bits

[bit27:24] HSSPICSL: HSSPI Clock Selection Bits

These bits select the source clock for HSSPI clock.

bit27:24	Description
0000	High-speed CR clock
0001	Low-speed CR clock
0010	Main clock
0011	Sub clock (Do not use this setting)
0100	PLL0 clock
0101	PLL1 clock
0110	PLL2 clock
0111	PLL3 clock
1000	SSCG PLL0 clock
1001	SSCG PLL1 clock
1010	SSCG PLL2 clock
1011	SSCG PLL3 clock
1100	Setting prohibited
1101	Setting prohibited
1110	Setting prohibited
1111	Clock fixed at "L"

[bit23:21] Reserved: Reserved Bits

[bit20] LAPP1ACSL：LAPP1A Clock Selection Bit

This bit selects the source clock for LAPP1A clock.

bit20	Description
0	CD0 clock
1	PLL0 clock

Note:

- *When it is attempted to change the clock selection by this bit, both PLL0 clock and CD0 clock must be supplied.*

[bit19:17] Reserved: Reserved Bits

[bit16] LAPP0ACSL：LAPP0A Clock Selection Bit

This bit selects the source clock for LAPP0A clock.

bit16	Description
0	CD0 clock
1	PLL0 clock

Note:

- *When it is attempted to change the clock selection by this bit, both PLL0 clock and CD0 clock must be supplied.*

[bit15:13] Reserved: Reserved Bits

[bit12] LCP1ACSL：LCP1A Clock Selection Bit

This bit selects the source clock for LCP1A clock.

bit12	Description
0	CD0 clock
1	PLL0 clock

Note:

- *When it is attempted to change the clock selection by this bit, both PLL0 clock and CD0 clock must be supplied.*

[bit11:9] Reserved: Reserved Bits

[bit8] LCP0ACSL：LCP0A Clock Selection Bit

This bit selects the source clock for LCP0A clock.

bit8	Description
0	CD0 clock
1	PLL0 clock

Note:

- *When it is attempted to change the clock selection by this bit, both PLL0 clock and CD0 clock must be supplied.*

[bit7:3] Reserved: Reserved Bits

[bit2:0] CD0CSL: Clock Domain 0 Clock Selection Bits

These bits select the source clock for clock domain 0.

bit2:0	Description
000	High-speed CR clock
001	Low-speed CR clock
010	Main clock
011	Sub clock (Do not use this setting)
100	PLL0 clock
101	SSCG0 clock
110	Setting prohibited
111	Clock fixed at "L"

5.11.2. RUN Clock Selection Register 1 (SYSC1_RUNCKSELR1)

The SYSC1_RUNCKSELR1 register sets the clock source for the clock domain.

bit	31	30	29	28	27	26	25	24
Field	Reserved				CD4CSL			
R/W Attribute	R0,WX				R/W			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	23	22	21	20	19	18	17	16
Field	Reserved				CD3CSL			
R/W Attribute	R0,WX				R/W			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	15	14	13	12	11	10	9	8
Field	Reserved				CD2CSL			
R/W Attribute	R0,WX				R/W			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	7	6	5	4	3	2	1	0
Field	Reserved				CD1CSL			
R/W Attribute	R0,WX				R/W			
Protection Attribute	WPS							
Initial Value	0000				0000			

[bit31:28] Reserved: Reserved Bits

[bit27:24] CD4CSL: Clock Domain 4 Clock Selection Bits

These bits select the source clock for clock domain 4.

bit27:24	Description
0000	High-speed CR clock
0001	Low-speed CR clock
0010	Main clock
0011	Sub clock (Do not use this setting)
0100	PLL0 clock
0101	PLL1 clock
0110	PLL2 clock
0111	PLL3 clock
1000	SSCG PLL0 clock
1001	SSCG PLL1 clock
1010	SSCG PLL2 clock
1011	SSCG PLL3 clock
1100	Setting prohibited
1101	Setting prohibited
1110	Setting prohibited
1111	Clock fixed at "L"

[bit23:20] Reserved: Reserved Bits

[bit19:16] CD3CSL: Clock Domain 3 Clock Selection Bits

These bits select the source clock for clock domain 3.

bit19:16	Description
0000	High-speed CR clock
0001	Low-speed CR clock
0010	Main clock
0011	Sub clock (Do not use this setting)
0100	PLL0 clock
0101	PLL1 clock
0110	PLL2 clock
0111	PLL3 clock
1000	SSCG PLL0 clock
1001	SSCG PLL1 clock
1010	SSCG PLL2 clock
1011	SSCG PLL3 clock
1100	Setting prohibited
1101	Setting prohibited
1110	Setting prohibited
1111	Clock fixed at "L"

[bit15:12] Reserved: Reserved Bits

[bit11:8] CD2CSL: Clock Domain 2 Clock Selection Bits

These bits select the source clock for clock domain 2.

bit11:8	Description
0000	High-speed CR clock
0001	Low-speed CR clock
0010	Main clock
0011	Sub clock (Do not use this setting)
0100	PLL0 clock
0101	PLL1 clock
0110	PLL2 clock
0111	PLL3 clock
1000	SSCG PLL0 clock
1001	SSCG PLL1 clock
1010	SSCG PLL2 clock
1011	SSCG PLL3 clock
1100	Setting prohibited
1101	Setting prohibited
1110	Setting prohibited
1111	Clock fixed at "L"

[bit7:4] Reserved: Reserved Bits

[bit3:0] CD1CSL: Clock Domain 1 Clock Selection Bits

These bits select the source clock for clock domain 1.

bit3:0	Description
0000	High-speed CR clock
0001	Low-speed CR clock
0010	Main clock
0011	Sub clock (Do not use this setting)
0100	PLL0 clock
0101	PLL1 clock
0110	PLL2 clock
0111	PLL3 clock
1000	SSCG PLL0 clock
1001	SSCG PLL1 clock
1010	SSCG PLL2 clock
1011	SSCG PLL3 clock
1100	Setting prohibited
1101	Setting prohibited
1110	Setting prohibited
1111	Clock fixed at "L"

5.11.3. RUN Clock Selection Register 2 (SYSC1_RUNCKSELR2)

The SYSC1_RUNCKSELR2 register sets the clock source for the clock domain.

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	Reserved															
R/W Attribute	R0,WX															
Protection Attribute	WPS															
Initial Value	00000000_00000000															

bit	15	14	13	12	11	10	9	8
Field	Reserved					TRCCSL		
R/W Attribute	R0,WX					R/W		
Protection Attribute	WPS							
Initial Value	00000					000		

bit	7	6	5	4	3	2	1	0
Field	Reserved				CD5CSL			
R/W Attribute	R0,WX				R/W			
Protection Attribute	WPS							
Initial Value	0000				0000			

[bit31:11] Reserved: Reserved Bits

[bit10:8] TRCCSL: TRC Clock Selection Bits

These bits select the source clock for TRC clock.

bit10:8	Description
000	High-speed CR clock
001	Low-speed CR clock
010	Main clock
011	Sub clock (Do not use this setting)
100	PLL0 clock
101	SSCG0 clock
110	PLL1 clock
111	Clock fixed at "L"

[bit7:4] Reserved: Reserved Bits

[bit3:0] CD5CSL: Clock Domain 5 Clock Selection Bits

These bits select the source clock for clock domain 5.

bit3:0	Description
0000	High-speed CR clock
0001	Low-speed CR clock
0010	Main clock
0011	Sub clock (Do not use this setting)
0100	PLL0 clock
0101	PLL1 clock
0110	PLL2 clock
0111	PLL3 clock
1000	SSCG PLL0 clock
1001	SSCG PLL1 clock
1010	SSCG PLL2 clock
1011	SSCG PLL3 clock
1100	Setting prohibited
1101	Setting prohibited
1110	Setting prohibited
1111	Clock fixed at "L"

5.11.4. RUN Clock Source Enable Register 0 (SYSC1_RUNCKER0)

The SYSC1_RUNCKER0 register sets whether to enable/disable oscillation of the internal operating clock.

bit	31	30	29	28	27	26	25	24
Field	ENCLK LAPP1A	ENCLK LAPP0A	ENCLK LCP1A	ENCLK LCP0A	ENCLK LAPP1	ENCLK LAPP0	ENCLK LCP1	ENCLK LCP0
R/W Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX
Protection Attribute	WPS							
Initial Value	1	1	1	1	1	1	1	1

bit	23	22	21	20	19	18	17	16
Field	ENCLK LCP	Reserved	ENCLK LLPBM2	ENCLK LLPBM	ENCLK HAPP1B1	ENCLK HAPP1B0	ENCLK HAPP0A1	ENCLK HAPP0A0
R/W Attribute	R1,WX	R0,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX
Protection Attribute	WPS							
Initial Value	1	0	1	1	1	1	1	1

bit	15	14	13	12	11	10	9	8
Field	Reserved	ENCLK SYSC1	Reserved		ENCLK EXTBUS	ENCLK MEMC	ENCLK DMA	ENCLK HPM
R/W Attribute	R0,WX	R1,WX	R0,WX		R1,WX	R1,WX	R/W	R1,WX
Protection Attribute	WPS							
Initial Value	0	1	00		1	1	1	1

bit	7	6	5	4	3	2	1	0
Field	ENCLK HPM2	ENCLK TRC	ENCLK DBG	ENCLK ATB	Reserved			ENCLK CPU0
R/W Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R0,WX			R1,WX
Protection Attribute	WPS							
Initial Value	1	1	1	1	000			1

[bit31] ENCLKLAPP1A: LAPP1A Clock Oscillation Enable Bit

This bit sets the LAPP1A clock.

Bit	Description
0	-
1	Enable clock oscillation.

[bit30] ENCLKLAPP0A: LAPP0A Clock Oscillation Enable Bit

This bit sets the LAPP0A clock.

Bit	Description
0	-
1	Enable clock oscillation.

[bit29] ENCLKLCP1A: LCP1A Clock Oscillation Enable Bit

This bit sets the LCP1A clock.

Bit	Description
0	-
1	Enable clock oscillation.

[bit28] ENCLKLCP0A: LCP0A Clock Oscillation Enable Bit

This bit sets the LCP0A clock.

Bit	Description
0	-
1	Enable clock oscillation.

[bit27] ENCLKLAPP1: LAPP1 Clock Oscillation Enable Bit

This bit sets the LAPP1 clock.

Bit	Description
0	-
1	Enable clock oscillation.

[bit26] ENCLKLAPP0: LAPP0 Clock Oscillation Enable Bit

This bit sets the LAPP0 clock.

Bit	Description
0	-
1	Enable clock oscillation.

[bit25] ENCLKLCP1: LCP1 Clock Oscillation Enable Bit

This bit sets the LCP1 clock.

Bit	Description
0	-
1	Enable clock oscillation.

[bit24] ENCLKLCP0: LCP0 Clock Oscillation Enable Bit

This bit sets the LCP0 clock.

Bit	Description
0	-
1	Enable clock oscillation.

[bit23] ENCLKLCP: LCP Clock Oscillation Enable Bit

This bit sets the LCP clock.

Bit	Description
0	-
1	Enable clock oscillation.

[bit22] Reserved: Reserved Bit

[bit21] ENCLKLLPBM2: LLPBM2 Clock Oscillation Enable Bit

This bit sets the LLPBM2 clock.

Bit	Description
0	-
1	Enable clock oscillation.

[bit20] ENCLKLLPBM: LLPBM Clock Oscillation Enable Bit

This bit sets the LLPBM clock.

Bit	Description
0	-
1	Enable clock oscillation.

[bit19] ENCLKHAPP1B1: HAPP1B1 Clock Oscillation Enable Bit

This bit sets the HAPP1B1 clock.

Bit	Description
0	-
1	Enable clock oscillation.

[bit18] ENCLKHAPP1B0: HAPP1B0 Clock Oscillation Enable Bit

This bit sets the HAPP1B0 clock.

Bit	Description
0	-
1	Enable clock oscillation.

[bit17] ENCLKHAPP0A1: HAPP0A1 Clock Oscillation Enable Bit

This bit sets the HAPP0A1 clock.

Bit	Description
0	-
1	Enable clock oscillation.

[bit16] ENCLKHAPP0A0: HAPP0A0 Clock Oscillation Enable Bit

This bit sets the HAPP0A0 clock.

Bit	Description
0	-
1	Enable clock oscillation.

[bit15] Reserved: Reserved Bit

[bit14] ENCLKSYSC1: ENCLKSYSC1 Clock Oscillation Enable Bit

This bit sets the ENCLKSYSC1 clock.

Bit	Description
0	-
1	Enable clock oscillation.

[bit13:12] Reserved: Reserved Bits

[bit11] ENCLKEXTBUS: ENCLKEXTBUS Clock Oscillation Enable Bit

This bit sets the ENCLKEXTBUS clock.

Bit	Description
0	-
1	Enable clock oscillation.

[bit10] ENCLKMEMC: ENCLKMEMC Clock Oscillation Enable Bit

This bit sets the ENCLKMEMC clock.

Bit	Description
0	-
1	Enable clock oscillation.

[bit9] ENCLKDMA: ENCLKDMA Clock Oscillation Enable Bit

This bit sets the ENCLKDMA clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit8] ENCLKHPM: ENCLKHPM Clock Oscillation Enable Bit

This bit sets the ENCLKHPM clock.

Bit	Description
0	-
1	Enable clock oscillation.

[bit7] ENCLKHPM2: ENCLKHPM2 Clock Oscillation Enable Bit

This bit sets the ENCLKHPM2 clock.

Bit	Description
0	-
1	Enable clock oscillation.

[bit6] ENCLKTRC: ENCLKTRC Clock Oscillation Enable Bit

This bit sets the ENCLKTRC clock.

Bit	Description
0	-
1	Enable clock oscillation.

[bit5] ENCLKDBG: ENCLKDBG Clock Oscillation Enable Bit

This bit sets the ENCLKDBG clock.

Bit	Description
0	-
1	Enable clock oscillation.

[bit4] ENCLKATB: ENCLKATB Clock Oscillation Enable Bit

This bit sets the ENCLKATB clock.

Bit	Description
0	-
1	Enable clock oscillation.

[bit3:1] Reserved: Reserved Bits

[bit0] ENCLKCPU0: ENCLKCPU0 Clock Oscillation Enable Bit

This bit sets the ENCLKCPU0 clock.

Bit	Description
0	-
1	Enable clock oscillation.

5.11.5. RUN Clock Source Enable Register 1 (SYSC1_RUNCKER1)

The SYSC1_RUNCKER1 register sets whether to enable/disable oscillation of the internal operating clock.

bit	31	30	29	28	27	26	25	24
Field	Reserved			ENCLK CD3B1	ENCLK CD3B0	ENCLK CD3A1	ENCLK CD3A0	ENCLK CD3
R/W Attribute	R0,WX			R/W1	R/W1	R/W1	R/W1	R/W1
Protection Attribute	WPS							
Initial Value	000			1	1	1	1	1

bit	23	22	21	20	19	18	17	16
Field	Reserved			ENCLK CD2B1	ENCLK CD2B0	ENCLK CD2A1	ENCLK CD2A0	ENCLK CD2
R/W Attribute	R0,WX			R/W1	R/W1	R/W1	R/W1	R/W1
Protection Attribute	WPS							
Initial Value	000			1	1	1	1	1

bit	15	14	13	12	11	10	9	8
Field	Reserved			ENCLK CD1B1	ENCLK CD1B0	ENCLK CD1A1	ENCLK CD1A0	ENCLK CD1
R/W Attribute	R0,WX			R/W1	R/W1	R/W1	R/W1	R/W1
Protection Attribute	WPS							
Initial Value	000			1	1	1	1	1

Field	Reserved	ENCLK HSSPI
R/W Attribute	R0,WX	R/W
Protection Attribute	WPS	
Initial Value	0000000	1

[bit31:29] Reserved: Reserved Bits

[bit28] ENCLKCD3B1: CD3B1 Clock Oscillation Enable Bit

This bit sets the CD3B1 clock.

Bit	Description
0	Setting '0' is prohibited.
1	Enable clock oscillation.

[bit27] ENCLKCD3B0: CD3B0 Clock Oscillation Enable Bit

This bit sets the CD3B0 clock.

Bit	Description
0	Setting '0' is prohibited.
1	Enable clock oscillation.

[bit26] ENCLKCD3A1: CD3A1 Clock Oscillation Enable Bit

This bit sets the CD3A1 clock.

Bit	Description
0	Setting '0' is prohibited.
1	Enable clock oscillation.

[bit25] ENCLKCD3A0: CD3A0 Clock Oscillation Enable Bit

This bit sets the CD3A0 clock.

Bit	Description
0	Setting '0' is prohibited.
1	Enable clock oscillation.

[bit24] ENCLKCD3: CD3 Clock Oscillation Enable Bit

This bit sets the CD3 clock.

Bit	Description
0	Setting '0' is prohibited.
1	Enable clock oscillation.

[bit23:21] Reserved: Reserved Bits

[bit20] ENCLKCD2B1: CD2B1 Clock Oscillation Enable Bit

This bit sets the CD2B1 clock.

Bit	Description
0	Setting '0' is prohibited.
1	Enable clock oscillation.

[bit19] ENCLKCD2B0: CD2B0 Clock Oscillation Enable Bit

This bit sets the CD2B0 clock.

Bit	Description
0	Setting '0' is prohibited.
1	Enable clock oscillation.

[bit18] ENCLKCD2A1: CD2A1 Clock Oscillation Enable Bit

This bit sets the CD2A1 clock.

Bit	Description
0	Setting '0' is prohibited.
1	Enable clock oscillation.

[bit17] ENCLKCD2A0: CD2A0 Clock Oscillation Enable Bit

This bit sets the CD2A0 clock.

Bit	Description
0	Setting '0' is prohibited.
1	Enable clock oscillation.

[bit16] ENCLKCD2: CD2 Clock Oscillation Enable Bit

This bit sets the CD2 clock.

Bit	Description
0	Setting '0' is prohibited.
1	Enable clock oscillation.

[bit15:13] Reserved: Reserved Bits

[bit12] ENCLKCD1B1: CD1B1 Clock Oscillation Enable Bit

This bit sets the CD1B1 clock.

Bit	Description
0	Setting '0' is prohibited.
1	Enable clock oscillation.

[bit11] ENCLKCD1B0: CD1B0 Clock Oscillation Enable Bit

This bit sets the CD1B0 clock.

Bit	Description
0	Setting '0' is prohibited.
1	Enable clock oscillation.

[bit10] ENCLKCD1A1: CD1A1 Clock Oscillation Enable Bit

This bit sets the CD1A1 clock.

Bit	Description
0	Setting '0' is prohibited.
1	Enable clock oscillation.

[bit9] ENCLKCD1A0: CD1A0 Clock Oscillation Enable Bit

This bit sets the CD1A0 clock.

Bit	Description
0	Setting '0' is prohibited.
1	Enable clock oscillation.

[bit8] ENCLKCD1: CD1 Clock Oscillation Enable Bit

This bit sets the CD1 clock.

Bit	Description
0	Setting '0' is prohibited.
1	Enable clock oscillation.

[bit7:1] Reserved: Reserved Bits

[bit0] ENCLKHSSPI: HSSPI Clock Oscillation Enable Bit

This bit sets the HSSPI clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

5.11.6. RUN Clock Source Enable Register 2 (SYSC1_RUNCKER2)

The SYSC1_RUNCKER2register sets whether to enable/disable oscillation of the internal operating clock.

bit	31							16
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	WPS							
Initial Value	00000000_00000000							

bit	15	14	13	12	11	10	9	8
Field	Reserved			ENCLKCD5B1	ENCLKCD5B0	ENCLKCD5A1	ENCLKCD5A0	ENCLKCD5
R/W Attribute	R0,WX			R/W1	R/W1	R/W1	R/W1	R/W1
Protection Attribute	WPS							
Initial Value	000			1	1	1	1	1

bit	7	6	5	4	3	2	1	0
Field	Reserved			ENCLKCD4B1	ENCLKCD4B0	ENCLKCD4A1	ENCLKCD4A0	ENCLKCD4
R/W Attribute	R0,WX			R/W1	R/W1	R/W1	R/W1	R/W1
Protection Attribute	WPS							
Initial Value	000			1	1	1	1	1

[bit31:13] Reserved: Reserved Bits

[bit12] ENCLKCD5B1: CD5B1 Clock Oscillation Enable Bit

This bit sets the CD5B1 clock.

Bit	Description
0	Setting '0' is prohibited.
1	Enable clock oscillation.

[bit11] ENCLKCD5B0: CD5B0 Clock Oscillation Enable Bit

This bit sets the CD5B0 clock.

Bit	Description
0	Setting '0' is prohibited.
1	Enable clock oscillation.

[bit10] ENCLKCD5A1: CD5A1 Clock Oscillation Enable Bit

This bit sets the CD5A1 clock.

Bit	Description
0	Setting '0' is prohibited.
1	Enable clock oscillation.

[bit9] ENCLKCD5A0: CD5A0 Clock Oscillation Enable Bit

This bit sets the CD5A0 clock.

Bit	Description
0	Setting '0' is prohibited.
1	Enable clock oscillation.

[bit8] ENCLKCD5: CD5 Clock Oscillation Enable Bit

This bit sets the CD5 clock.

Bit	Description
0	Setting '0' is prohibited.
1	Enable clock oscillation.

[bit7:5] Reserved: Reserved Bits

[bit4] ENCLKCD4B1: CD4B1 Clock Oscillation Enable Bit

This bit sets the CD4B1 clock.

Bit	Description
0	Setting '0' is prohibited.
1	Enable clock oscillation.

[bit3] ENCLKCD4B0: CD4B0 Clock Oscillation Enable Bit

This bit sets the CD4B0 clock.

Bit	Description
0	Setting '0' is prohibited.
1	Enable clock oscillation.

[bit2] ENCLKCD4A1: CD4A1 Clock Oscillation Enable Bit

This bit sets the CD4A1 clock.

Bit	Description
0	Setting '0' is prohibited.
1	Enable clock oscillation.

[bit1] ENCLKCD4A0: CD4A0 Clock Oscillation Enable Bit

This bit sets the CD4A0 clock.

Bit	Description
0	Setting '0' is prohibited.
1	Enable clock oscillation.

[bit0] ENCLKCD4: CD4 Clock Oscillation Enable Bit

This bit sets the CD4 clock.

Bit	Description
0	Setting '0' is prohibited.
1	Enable clock oscillation.

5.11.7. RUN Clock Divider Register 0 (SYSC1_RUNCKDIVR0)

The SYSC1_RUNCKDIVR0 register sets the division ratio of each internal operating clock.

bit	31	30	29	28	27	26	25	24
Field	Reserved					HPMDIV		
R/W Attribute	R0,WX					R/W		
Protection Attribute	WPS							
Initial Value	00000					000		

bit	23	22	21	20	19	18	17	16
Field	Reserved			TRCDIV				
R/W Attribute	R0,WX			R/W				
Protection Attribute	WPS							
Initial Value	000			00000				

bit	15	14	13	12	11	10	9	8
Field	Reserved		DBGDIV		Reserved		ATBDIV	
R/W Attribute	R0,WX		R/W		R0,WX		R/W	
Protection Attribute	WPS							
Initial Value	00		00		00		00	

bit	7	6	5	4	3	2	1	0
Field	Reserved			SYSDIV				
R/W Attribute	R0,WX			R/W				
Protection Attribute	WPS							
Initial Value	000			00000				

[bit31:27] Reserved: Reserved Bits

[bit26:24] HPMDIV: HPM Clock Divider Setting Bits

These bits set the division ratio of the HPM clock, DMA clock, and MEMC clock from the SYS clock.

bit26:24	Description
000	No division
001	Divided by 2
010	Divided by 3
011	Divided by 4
100	Divided by 5
101	Divided by 6
110	Divided by 7
111	Divided by 8

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

[bit23:21] Reserved: Reserved Bits

[bit20:16] TRCDIV: TRC Clock Divider Setting Bits

These bits set the division ratio of the TRC clock from the source clock (clock domain TRC).

bit20:16	Description
00000	No division
00001	Divided by 2
00010	Divided by 3
00011	Divided by 4
. . .	. . .
11101	Divided by 30
11110	Divided by 31
11111	Divided by 32

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

[bit15:14] Reserved: Reserved Bits

[bit13:12] DBGDIV: DBG Clock Divider Setting Bits

These bits set the division ratio of the DBG clock from the ATB clock.

bit13:12	Description
00	No division
01	Divided by 2
10	Divided by 4
11	Divided by 8

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

[bit11:10] Reserved: Reserved Bits

[bit9:8] ATBDIV [1:0]: ATB Clock Divider Setting Bits

These bits set the division ratio of the ATB clock from the SYS clock.

bit9:8	Description
00	No division
01	Divided by 2
10	Divided by 4
11	Divided by 8

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

[bit7:5] Reserved: Reserved Bits

[bit4:0] SYSDIV: SYS Clock Divider Setting Bits

These bits set the division ratio of the CPU0 clock from the source clock (clock domain 0).

bit4:0	Description
00000	No division
00001	Divided by 2
00010	Divided by 3
...	...
11101	Divided by 30
11110	Divided by 31
11111	Divided by 32

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

5.11.8. RUN Clock Divider Register 1 (SYSC1_RUNCKDIVR1)

The SYSC1_RUNCKDIVR1 register sets the division ratio of each internal operating clock.

bit	31	30	29	28	27	26	25	24
Field	HAPP1B1DIV				HAPP1B0DIV			
R/W Attribute	R/W				R/W			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	23	22	21	20	19	18	17	16
Field	HAPP0A1DIV				HAPP0A0DIV			
R/W Attribute	R/W				R/W			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	15	14	13	12	11	10	9	8
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	WPS							
Initial Value	00000000							

bit	7	6	5	4	3	2	1	0
Field	SYSC1DIV				Reserved	EXTBUSDIV		
R/W Attribute	R/W				R0,WX	R/W		
Protection Attribute	WPS							
Initial Value	0000				0	000		

[bit31:28] HAPP1B1DIV: HAPP1B1 Clock Divider Setting Bits

These bits set the division ratio of the HAPP1B1 clock from the HAPP1B0 clock.

bit31:28	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
0011	Divided by 4
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

[bit27:24] HAPP1B0DIV: HAPP1B0 Clock Divider Setting Bits

These bits set the division ratio of the HAPP1B0 clock from the HPM clock.

bit27:24	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
0011	Divided by 4
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

[bit23:20] HAPP0A1DIV: HAPP0A1 Clock Divider Setting Bits

These bits set the division ratio of the HAPP0A1 clock from the HAPP0A0 clock.

bit23:20	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
0011	Divided by 4
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

[bit19:16] HAPP0A0DIV: HAPP0A0 Clock Divider Setting Bits

These bits set the division ratio of the HAPP0A0 clock from the HPM clock.

bit19:16	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
0011	Divided by 4
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

[bit7:4] SYSC1DIV: SYSC1 Clock Divider Setting Bits

These bits set the division ratio of the SYSC1 clock from the HPM clock.

bit7:4	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
0011	Divided by 4
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

[bit3] Reserved: Reserved Bit

[bit2:0] EXTBUSDIV: EXTBUS Clock Divider Setting Bits

These bits set the division ratio of the EXTBUS clock from the HPM clock.

bit2:0	Description
000	No division
001	Divided by 2
010	Divided by 4
011	Divided by 8
100	Divided by 16
101	Divided by 32
110	Divided by 64
111	Divided by 128

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

5.11.9. RUN Clock Divider Register 2 (SYSC1_RUNCKDIVR2)

The SYSC1_RUNCKDIVR2 register sets the division ratio of each internal operating clock.

bit	31	30	29	28	27	26	25	24
Field	Reserved				LAPP1DIV			
R/W Attribute	R0,WX				R/W			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	23	22	21	20	19	18	17	16
Field	Reserved				LAPP0DIV			
R/W Attribute	R0,WX				R/W			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	15	14	13	12	11	10	9	8
Field	Reserved				LCP1DIV			
R/W Attribute	R0,WX				R/W			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	7	6	5	4	3	2	1	0
Field	LCP0DIV				Reserved		LCPDIV	
R/W Attribute	R/W				R0,WX		R/W	
Protection Attribute	WPS							
Initial Value	0000				00		00	

[bit31:28] Reserved: Reserved Bits

[bit27:24] LAPP1DIV: LAPP1 Clock Divider Setting Bits

These bits set the division ratio of the LAPP1 clock from the CPU clock.

bit27:24	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
0011	Divided by 4
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

[bit23:20] Reserved: Reserved Bits

[bit19:16] LAPP0DIV: LAPP0 Clock Divider Setting Bits

These bits set the division ratio of the LAPP0 clock from the CPU clock.

bit19:16	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
0011	Divided by 4
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

[bit15:12] Reserved: Reserved Bits

[bit11:8] LCP1DIV: LCP1 Clock Divider Setting Bits

These bits set the division ratio of the LCP1 clock from the CPU clock.

bit11:8	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
0011	Divided by 4
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

[bit7:4] LCP0DIV: LCP0 Clock Divider Setting Bits

These bits set the division ratio of the LCP0 clock from the CPU clock.

bit7:4	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
0011	Divided by 4
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

[bit3:2] Reserved: Reserved Bits

[bit1:0] LCPDIV: LCP Clock Divider Setting Bits

These bits set the division ratio of the LCP clock from the CPU clock.

bit1:0	Description
00	No division
01	Divided by 2
10	Divided by 4
11	Divided by 8

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

5.11.10. RUN Clock Divider Register 3 (SYSC1_RUNCKDIVR3)

The SYSC1_RUNCKDIVR3 register sets the division ratio of each internal operating clock.

bit	31	30	29	28	27	26	25	24
Field	Reserved			LAPP1ADIV				
R/W Attribute	R0,WX			R/W				
Protection Attribute	WPS							
Initial Value	000			00000				

bit	23	22	21	20	19	18	17	16
Field	Reserved			LAPP0ADIV				
R/W Attribute	R0,WX			R/W				
Protection Attribute	WPS							
Initial Value	000			00000				

bit	15	14	13	12	11	10	9	8
Field	Reserved			LCP1ADIV				
R/W Attribute	R0,WX			R/W				
Protection Attribute	WPS							
Initial Value	000			00000				

bit	7	6	5	4	3	2	1	0
Field	Reserved			LCP0ADIV				
R/W Attribute	R0,WX			R/W				
Protection Attribute	WPS							
Initial Value	000			00000				

[bit31:29] Reserved: Reserved Bits

[bit28:24] LAPP1ADIV: LAPP1A Clock Divider Setting Bits

These bits set the division ratio of the LAPP1A clock from the CPU/PLL0 clock.

bit28:24	Description
00000	No division
00001	Divided by 2
00010	Divided by 3
00011	Divided by 4
...	...
11101	Divided by 30
11110	Divided by 31
11111	Divided by 32

[bit23:21] Reserved: Reserved Bits

[bit20:16] LAPP0ADIV: LAPP0A Clock Divider Setting Bits

These bits set the division ratio of the LAPP0A clock from the CPU/PLL0 clock.

bit20:16	Description
00000	No division
00001	Divided by 2
00010	Divided by 3
00011	Divided by 4
...	...
11101	Divided by 30
11110	Divided by 31
11111	Divided by 32

[bit15:13] Reserved: Reserved Bits

[bit12:8] LCP1ADIV: LCP1A Clock Divider Setting Bits

These bits set the division ratio of the LCP1A clock from the CPU/PLL0 clock.

bit12:8	Description
00000	No division
00001	Divided by 2
00010	Divided by 3
00011	Divided by 4
...	...
11101	Divided by 30
11110	Divided by 31
1111	Divided by 32

[bit7:5] Reserved: Reserved Bits

[bit4:0] LCP0ADIV: LCP0A Clock Divider Setting Bits

These bits set the division ratio of the LCP0A clock from the CPU/PLL0 clock.

bit4:0	Description
00000	No division
00001	Divided by 2
00010	Divided by 3
00011	Divided by 4
...	...
11101	Divided by 30
11110	Divided by 31
11111	Divided by 32

5.11.11. RUN Clock Divider Register 4 (SYSC1_RUNCKDIVR4)

The SYSC1_RUNCKDIVR4 register sets the division ratio of each internal operating clock.

bit	31	8
Field	Reserved	
R/W Attribute	R0,WX	
Protection Attribute	WPS	
Initial Value	00000000_00000000_00000000	

bit	7	6	5	4	3	2	1	0
Field	Reserved			HSSPIDIV				
R/W Attribute	R0,WX			R/W				
Protection Attribute	WPS							
Initial Value	000			00000				

[bit31:5] Reserved: Reserved Bits

[bit4:0] HSSPIDIV: HSSPI Clock Divider Setting Bits

These bits set the division ratio of the HSSPI clock from the source clock (HSSPI clock domain).

bit4:0	Description
00000	No division
00001	Divided by 2
00010	Divided by 3
...	...
11101	Divided by 30
11110	Divided by 31
11111	Divided by 32

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

5.11.12. RUN Clock Divider Register 5 (SYSC1_RUNCKDIVR5)

The SYSC1_RUNCKDIVR5 register sets the division ratio of each internal operating clock.

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	WPS							
Initial Value	00000000							

bit	23	22	21	20	19	18	17	16
Field	CD1B1DIV				CD1B0DIV			
R/W Attribute	R/W				R/W			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	15	14	13	12	11	10	9	8
Field	CD1A1DIV				CD1A0DIV			
R/W Attribute	R/W				R/W			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	7	6	5	4	3	2	1	0
Field	Reserved			CD1DIV				
R/W Attribute	R0,WX			R/W				
Protection Attribute	WPS							
Initial Value	000			00000				

[bit31:24] Reserved: Reserved Bits

[bit23:20] CD1B1DIV: CD1B1 Clock Divider Setting Bits

These bits set the division ratio of the CD1B1 clock from the CD1B0 clock.

bit23:20	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

[bit19:16] CD1B0DIV: CD1B0 Clock Divider Setting Bits

These bits set the division ratio of the CD1B0 clock from the CD1 clock.

bit19:16	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

[bit15:12] CD1A1DIV: CD1A1 Clock Divider Setting Bits

These bits set the division ratio of the CD1A1 clock from the CD1A0 clock.

bit15:12	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

[bit11:8] CD1A0DIV: CD1A0 Clock Divider Setting Bits

These bits set the division ratio of the CD1A0 clock from the CD1 clock.

bit11:8	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

[bit7:5] Reserved: Reserved Bits

[bit4:0] CD1DIV: CD1 Clock Divider Setting Bits

These bits set the division ratio of the CD1 clock from the source clock (clock domain 1).

bit4:0	Description
00000	No division
00001	Divided by 2
00010	Divided by 3
...	...
11101	Divided by 30
11110	Divided by 31
11111	Divided by 32

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

5.11.13. RUN Clock Divider Register 6 (SYSC1_RUNCKDIVR6)

The SYSC1_RUNCKDIVR6 register sets the division ratio of each internal operating clock.

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	WPS							
Initial Value	00000000							

bit	23	22	21	20	19	18	17	16
Field	CD2B1DIV				CD2B0DIV			
R/W Attribute	R/W				R/W			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	15	14	13	12	11	10	9	8
Field	CD2A1DIV				CD2A0DIV			
R/W Attribute	R/W				R/W			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	7	6	5	4	3	2	1	0
Field	Reserved			CD2DIV				
R/W Attribute	R0,WX			R/W				
Protection Attribute	WPS							
Initial Value	000			00000				

[bit31:24] Reserved: Reserved Bits

[bit23:20] CD2B1DIV: CD2B1 Clock Divider Setting Bits

These bits set the division ratio of the CD2B1 clock from the CD2B0 clock.

bit23:20	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

[bit19:16] CD2B0DIV: CD2B0 Clock Divider Setting Bits

These bits set the division ratio of the CD2B0 clock from the CD2 clock.

bit19:16	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

[bit15:12] CD2A1DIV: CD2A1 Clock Divider Setting Bits

These bits set the division ratio of the CD2A1 clock from the CD2A0 clock.

bit15:12	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

[bit11:8] CD2A0DIV: CD2A0 Clock Divider Setting Bits

These bits set the division ratio of the CD2A0 clock from the CD2 clock.

bit11:8	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

[bit7:5] Reserved: Reserved Bits

[bit4:0] CD2DIV: CD2 Clock Divider Setting Bits

These bits set the division ratio of the CD2 clock from the source clock (clock domain 2).

bit4:0	Description
00000	No division
00001	Divided by 2
00010	Divided by 3
...	...
11101	Divided by 30
11110	Divided by 31
11111	Divided by 32

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

5.11.14. RUN Clock Divider Register 7 (SYSC1_RUNCKDIVR7)

The SYSC1_RUNCKDIVR7 register sets the division ratio of each internal operating clock.

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	WPS							
Initial Value	00000000							

bit	23	22	21	20	19	18	17	16
Field	CD3B1DIV				CD3B0DIV			
R/W Attribute	R/W				R/W			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	15	14	13	12	11	10	9	8
Field	CD3A1DIV				CD3A0DIV			
R/W Attribute	R/W				R/W			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	7	6	5	4	3	2	1	0
Field	Reserved			CD3DIV				
R/W Attribute	R0,WX			R/W				
Protection Attribute	WPS							
Initial Value	000			00000				

[bit31:24] Reserved: Reserved Bits

[bit23:20] CD3B1DIV: CD3B1 Clock Divider Setting Bits

These bits set the division ratio of the CD3B1 clock from the CD3B0 clock.

bit23:20	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

[bit19:16] CD3B0DIV: CD3B0 Clock Divider Setting Bits

These bits set the division ratio of the CD3B0 clock from the CD3 clock.

bit19:16	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

[bit15:12] CD3A1DIV: CD3A1 Clock Divider Setting Bits

These bits set the division ratio of the CD3A1 clock from the CD3A0 clock.

bit15:12	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

[bit11:8] CD3A0DIV: CD3A0 Clock Divider Setting Bits

These bits set the division ratio of the CD3A0 clock from the CD3 clock.

bit11:8	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

[bit7:5] Reserved: Reserved Bits

[bit4:0] CD3DIV: CD3 Clock Divider Setting Bits

These bits set the division ratio of the CD3 clock from the source clock (clock domain 3).

bit4:0	Description
00000	No division
00001	Divided by 2
00010	Divided by 3
...	...
11101	Divided by 30
11110	Divided by 31
11111	Divided by 32

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

5.11.15. RUN Clock Divider Register 8 (SYSC1_RUNCKDIVR8)

The SYSC1_RUNCKDIVR8 register sets the division ratio of each internal operating clock.

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	WPS							
Initial Value	00000000							

bit	23	22	21	20	19	18	17	16
Field	CD4B1DIV				CD4B0DIV			
R/W Attribute	R/W				R/W			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	15	14	13	12	11	10	9	8
Field	CD4A1DIV				CD4A0DIV			
R/W Attribute	R/W				R/W			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	7	6	5	4	3	2	1	0
Field	Reserved			CD4DIV				
R/W Attribute	R0,WX			R/W				
Protection Attribute	WPS							
Initial Value	000			00000				

[bit31:24] Reserved: Reserved Bits

[bit23:20] CD4B1DIV: CD4B1 Clock Divider Setting Bits

These bits set the division ratio of the CD4B1 clock from the CD4B0 clock.

bit23:20	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

[bit19:16] CD4B0DIV: CD4B0 Clock Divider Setting Bits

These bits set the division ratio of the CD4B0 clock from the CD4 clock.

bit19:16	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

[bit15:12] CD4A1DIV: CD4A1 Clock Divider Setting Bits

These bits set the division ratio of the CD4A1 clock from the CD4A0 clock.

bit15:12	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

[bit11:8] CD4A0DIV: CD4A0 Clock Divider Setting Bits

These bits set the division ratio of the CD4A0 clock from the CD4 clock.

bit11:8	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

[bit7:5] Reserved: Reserved Bits

[bit4:0] CD4DIV: CD4 Clock Divider Setting Bits

These bits set the division ratio of the CD4 clock from the source clock (clock domain 4).

bit4:0	Description
00000	No division
00001	Divided by 2
00010	Divided by 3
...	...
11101	Divided by 30
11110	Divided by 31
11111	Divided by 32

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

5.11.16. RUN Clock Divider Register 9 (SYSC1_RUNCKDIVR9)

The SYSC1_RUNCKDIVR9 register sets the division ratio of each internal operating clock.

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	WPS							
Initial Value	00000000							

bit	23	22	21	20	19	18	17	16
Field	CD5B1DIV				CD5B0DIV			
R/W Attribute	R/W				R/W			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	15	14	13	12	11	10	9	8
Field	CD5A1DIV				CD5A0DIV			
R/W Attribute	R/W				R/W			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	7	6	5	4	3	2	1	0
Field	Reserved			CD5DIV				
R/W Attribute	R0,WX			R/W				
Protection Attribute	WPS							
Initial Value	000			00000				

[bit31:24] Reserved: Reserved Bits

[bit23:20] CD5B1DIV: CD5B1 Clock Divider Setting Bits

These bits set the division ratio of the CD5B1 clock from the CD5B0 clock.

bit23:20	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

[bit19:16] CD5B0DIV: CD5B0 Clock Divider Setting Bits

These bits set the division ratio of the CD5B0 clock from the CD5 clock.

bit19:16	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

[bit15:12] CD5A1DIV: CD5A1 Clock Divider Setting Bits

These bits set the division ratio of the CD5A1 clock from the CD5A0 clock.

bit15:12	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

[bit11:8] CD5A0DIV: CD5A0 Clock Divider Setting Bits

These bits set the division ratio of the CD5A0 clock from the CD5 clock.

bit11:8	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

[bit7:5] Reserved: Reserved Bits

[bit4:0] CD5DIV: CD5 Clock Divider Setting Bits

These bits set the division ratio of the CD5 clock from the source clock (clock domain 5).

bit4:0	Description
00000	No division
00001	Divided by 2
00010	Divided by 3
...	...
11101	Divided by 30
11110	Divided by 31
11111	Divided by 32

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

5.11.17. RUN Profile Update Enable Register (SYSC1_RUNENR)

The SYSC1_RUNENR register sets the profile update enable.

bit	31							8
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	WPS							
Initial Value	00000000_00000000_00000000							

bit	7	6	5	4	3	2	1	0
Field	RUNEN1							
R/W Attribute	R,W							
Protection Attribute	WPS							
Initial Value	00000000							

[bit31:8] Reserved：Reserved Bits

[bit7:0] RUNEN1：RUN Profile Update Enable Bit

This bit is the RUN profile update enable register.

bit7:0	Description
During write operation	When "0xAB" is written: RUN profile update enable When "0x00" is written: RUN profile update prohibited When other than "0xAB" is written: Disable
During read operation	When "0xAB" is read: RUN profile update enable is valid When "0x00" is read: Invalid

Notes:

- *This register is cleared to "0x00" by hardware after RUN profile (configuration of SYSC1 register) is updated.*
- *This register permits byte access only. Any other access is invalid. If the above is not complied with, a bus error occurs.*

5.12. PSS Profile Register Group (SYSC1)

The registers in this group are PSS profile control setting registers.

5.12.1. PSS Clock Selection Register 0 (SYSC1_PSSCKSELR0)

The SYSC1_PSSCKSELR0 register sets the clock source for the clock domain.

bit	31	30	29	28	27	26	25	24
Field	Reserved				HSSPICSL			
R/W Attribute	R0,WX				R/W			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	23	22	21	20	19	18	17	16
Field	Reserved			LAPP1ACSL	Reserved			LAPP0ACSL
R/W Attribute	R0,WX			R/W	R0,WX			R/W
Protection Attribute	WPS							
Initial Value	000			0	000			0

bit	15	14	13	12	11	10	9	8
Field	Reserved			LCP1ACSL	Reserved			LCP0ACSL
R/W Attribute	R0,WX			R/W	R0,WX			R/W
Protection Attribute	WPS							
Initial Value	000			0	000			0

bit	7	6	5	4	3	2	1	0
Field	Reserved					CD0CSL		
R/W Attribute	R0,WX					R/W		
Protection Attribute	WPS							
Initial Value	00000					000		

[bit31:28] Reserved: Reserved Bits

[bit27:24] HSSPICSL: HSSPI Clock Selection Bits

These bits select the source clock for HSSPI.

bit27:24	Description
0000	High-speed CR clock
0001	Low-speed CR clock
0010	Main clock
0011	Sub clock (Do not use this setting)
0100	PLL0 clock

bit27:24	Description
0101	PLL1 clock
0110	PLL2 clock
0111	PLL3 clock
1000	SSCG PLL0 clock
1001	SSCG PLL1 clock
1010	SSCG PLL2 clock
1011	SSCG PLL3 clock
1100	Setting prohibited
1101	Setting prohibited
1110	Setting prohibited
1111	Clock fixed at "L"

[bit23:21] Reserved: Reserved Bits

[bit20] LAPP1ACSL: LAPP1A Clock Selection Bit

This bit selects the source clock for LAPP1A.

bit20	Description
0	CD0 clock
1	PLL0 clock

Note:

- *When it is attempted to change the clock selection by this bit, both PLL0 clock and CD0 clock must be supplied.*

[bit19:17] Reserved: Reserved Bits

[bit16] LAPP0ACSL: LAPP0A Clock Selection Bit

This bit selects the source clock for LAPP0A.

bit16	Description
0	CD0 clock
1	PLL0 clock

Note:

- *When it is attempted to change the clock selection by this bit, both PLL0 clock and CD0 clock must be supplied.*

[bit15:13] Reserved: Reserved Bits

[bit12] LCP1ACSL: LCP1A Clock Selection Bit

This bit selects the source clock for LCP1A.

bit12	Description
0	CD0 clock
1	PLL0 clock

Note:

- *When it is attempted to change the clock selection by this bit, both PLL0 clock and CD0 clock must be supplied.*

[bit11:9] Reserved: Reserved Bits

[bit8] LCP0ACSL: LCP0A Clock Selection Bit

This bit selects the source clock for LCP0A.

bit8	Description
0	CD0 clock
1	PLL0 clock

Note:

- *When it is attempted to change the clock selection by this bit, both PLL0 clock and CD0 clock must be supplied.*

[bit7:3] Reserved: Reserved Bits

[bit2:0] CD0CSL: Clock Domain 0 Clock Selection Bits

These bits select the source clock for clock domain 0.

bit2:0	Description
000	High-speed CR clock
001	Low-speed CR clock
010	Main clock
011	Sub clock (Do not use this setting)
100	PLL0 clock
101	SSCG0 clock
110	Setting prohibited
111	Clock fixed at "L"

Notes:

- *To use the source clock selection "Clock fixed at "L"", following bit must be also set to '0'.*
  SYSC1_PSSCKER0:ENCLKLAPP1A
  SYSC1_PSSCKER0:ENCLKLAPP0A
  SYSC1_PSSCKER0:ENCLKLCP1A
  SYSC1_PSSCKER0:ENCLKLCP0A
  Please refer to section 5.12.4 for the definition of the bits.

5.12.2. PSS Clock Selection Register 1 (SYSC1_PSSCKSELR1)

The SYSC1_PSSCKSELR1 register sets the clock source for the clock domain.

bit	31	30	29	28	27	26	25	24
Field	Reserved				CD4CSL			
R/W Attribute	R0,WX				R/W			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	23	22	21	20	19	18	17	16
Field	Reserved				CD3CSL			
R/W Attribute	R0,WX				R/W			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	15	14	13	12	11	10	9	8
Field	Reserved				CD2CSL			
R/W Attribute	R0,WX				R/W			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	7	6	5	4	3	2	1	0
Field	Reserved				CD1CSL			
R/W Attribute	R0,WX				R/W			
Protection Attribute	WPS							
Initial Value	0000				0000			

[bit31:28] Reserved: Reserved Bits

[bit27:24] CD4CSL: Clock Domain 4 Clock Selection Bits

These bits select the source clock for clock domain 4.

bit27:24	Description
0000	High-speed CR clock
0001	Low-speed CR clock
0010	Main clock
0011	Sub clock (Do not use this setting)
0100	PLL0 clock
0101	PLL1 clock
0110	PLL2 clock
0111	PLL3 clock
1000	SSCG PLL0 clock
1001	SSCG PLL1 clock

bit27:24	Description
1010	SSCG PLL2 clock
1011	SSCG PLL3 clock
1100	Setting prohibited
1101	Setting prohibited
1110	Setting prohibited
1111	Clock fixed at "L"

[bit23:20] Reserved: Reserved Bits

[bit19:16] CD3CSL: Clock Domain 3 Clock Selection Bits

These bits select the source clock for clock domain 3.

bit19:16	Description
0000	High-speed CR clock
0001	Low-speed CR clock
0010	Main clock
0011	Sub clock (Do not use this setting)
0100	PLL0 clock
0101	PLL1 clock
0110	PLL2 clock
0111	PLL3 clock
1000	SSCG PLL0 clock
1001	SSCG PLL1 clock
1010	SSCG PLL2 clock
1011	SSCG PLL3 clock
1100	Setting prohibited
1101	Setting prohibited
1110	Setting prohibited
1111	Clock fixed at "L"

[bit15:12] Reserved: Reserved Bits

[bit11:8] CD2CSL: Clock Domain 2 Clock Selection Bits

These bits select the source clock for clock domain 2.

bit11:8	Description
0000	High-speed CR clock
0001	Low-speed CR clock
0010	Main clock
0011	Sub clock (Do not use this setting)
0100	PLL0 clock
0101	PLL1 clock
0110	PLL2 clock

bit11:8	Description
0111	PLL3 clock
1000	SSCG PLL0 clock
1001	SSCG PLL1 clock
1010	SSCG PLL2 clock
1011	SSCG PLL3 clock
1100	Setting prohibited
1101	Setting prohibited
1110	Setting prohibited
1111	Clock fixed at "L"

[bit7:4] Reserved: Reserved Bits

[bit3:0] CD1CSL: Clock Domain 1 Clock Selection Bits

These bits select the source clock for clock domain 1.

bit3:0	Description
0000	High-speed CR clock
0001	Low-speed CR clock
0010	Main clock
0011	Sub clock (Do not use this setting)
0100	PLL0 clock
0101	PLL1 clock
0110	PLL2 clock
0111	PLL3 clock
1000	SSCG PLL0 clock
1001	SSCG PLL1 clock
1010	SSCG PLL2 clock
1011	SSCG PLL3 clock
1100	Setting prohibited
1101	Setting prohibited
1110	Setting prohibited
1111	Clock fixed at "L"

5.12.3. PSS Clock Selection Register 2 (SYSC1_PSSCKSELR2)

The SYSC1_PSSCKSELR2 register sets the clock source for the clock domain.

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	Reserved															
R/W Attribute	R0,WX															
Protection Attribute	WPS															
Initial Value	00000000_00000000															

bit	15	14	13	12	11	10	9	8
Field	Reserved					TRCCSL		
R/W Attribute	R0,WX					R/W		
Protection Attribute	WPS							
Initial Value	00000					000		

bit	7	6	5	4	3	2	1	0
Field	Reserved				CD5CSL			
R/W Attribute	R0,WX				R/W			
Protection Attribute	WPS							
Initial Value	0000				0000			

[bit31:11] Reserved: Reserved Bits

[bit10:8] TRCCSL: TRC Clock Selection Bits

These bits select the source clock for TRC clock.

bit10:8	Description
000	High-speed CR clock
001	Low-speed CR clock
010	Main clock
011	Sub clock (Do not use this setting)
100	PLL0 clock
101	SSCG0 clock
110	PLL1 clock
111	Clock fixed at "L"

[bit7:4] Reserved: Reserved Bits

[bit3:0] CD5CSL: Clock Domain 5 Clock Selection Bits

These bits select the source clock for clock domain 5.

bit3:0	Description
0000	High-speed CR clock
0001	Low-speed CR clock
0010	Main clock
0011	Sub clock (Do not use this setting)
0100	PLL0 clock
0101	PLL1 clock
0110	PLL2 clock
0111	PLL3 clock
1000	SSCG PLL0 clock
1001	SSCG PLL1 clock
1010	SSCG PLL2 clock
1011	SSCG PLL3 clock
1100	Setting prohibited
1101	Setting prohibited
1110	Setting prohibited
1111	Clock fixed at "L"

5.12.4. PSS Clock Source Enable Register 0 (SYSC1_PSSCKER0)

The SYSC1_PSSCKER0 register sets whether to enable/disable oscillation of the internal operating clock.

bit	31	30	29	28	27	26	25	24
Field	ENCLK LAPP1A	ENCLK LAPP0A	ENCLK LCP1A	ENCLK LCP0A	ENCLK LAPP1	ENCLK LAPP0	ENCLK LCP1	ENCLK LCP0
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	WPS							
Initial Value	1	1	1	1	1	1	1	1

bit	23	22	21	20	19	18	17	16
Field	ENCLK LCP	Reserved	ENCLK LLPBM2	ENCLK LLPBM	ENCLK HAPP1B1	ENCLK HAPP1B0	ENCLK HAPP0A1	ENCLK HAPP0A0
R/W Attribute	R/W	R0,WX	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	WPS							
Initial Value	1	0	1	1	1	1	1	1

bit	15	14	13	12	11	10	9	8
Field	Reserved	ENCLK SYSC1	Reserved		ENCLK EXTBUS	ENCLK MEMC	ENCLK DMA	ENCLK HPM
R/W Attribute	R0,WX	R/W	R0,WX		R/W	R/W	R/W	R/W
Protection Attribute	WPS							
Initial Value	0	1	00		1	1	1	1

bit	7	6	5	4	3	2	1	0
Field	ENCLK HPM2	ENCLK TRC	ENCLK DBG	ENCLK ATB	Reserved			ENCLK CPU0
R/W Attribute	R/W	R/W	R/W	R/W	R0,WX			R0,WX
Protection Attribute	WPS							
Initial Value	1	1	1	1	000			0

[bit31] ENCLKLAPP1A: LAPP1A Clock Oscillation Enable Bit

This bit sets the LAPP1A clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit30] ENCLKLAPP0A: LAPP0A Clock Oscillation Enable Bit

This bit sets the LAPP0A clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit29] ENCLKLCP1A: LCP1A Clock Oscillation Enable Bit

This bit sets the LCP1A clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit28] ENCLKLCP0A: LCP0A Clock Oscillation Enable Bit

This bit sets the LCP0A clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit27] ENCLKLAPP1: LAPP1 Clock Oscillation Enable Bit

This bit sets the LAPP1 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit26] ENCLKLAPP0: LAPP0 Clock Oscillation Enable Bit

This bit sets the LAPP0 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit25] ENCLKLCP1: LCP1 Clock Oscillation Enable Bit

This bit sets the LCP1 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit24] ENCLKLCP0: LCP0 Clock Oscillation Enable Bit

This bit sets the LCP0 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit23] ENCLKLCP: LCP Clock Oscillation Enable Bit

This bit sets the LCP clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit22] Reserved: Reserved Bit

[bit21] ENCLKLLPBM2: LLPBM2 Clock Oscillation Enable Bit

This bit sets the LLPBM2 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit20] ENCLKLLPBM: LLPBM Clock Oscillation Enable Bit

This bit sets the LLPBM clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit19] ENCLKHAPP1B1: HAPP1B1 Clock Oscillation Enable Bit

This bit sets the HAPP1B1 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit18] ENCLKHAPP1B0: HAPP1B0 Clock Oscillation Enable Bit

This bit sets the HAPP1B0 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit17] ENCLKHAPP0A1: HAPP0A1 Clock Oscillation Enable Bit

This bit sets the HAPP0A1 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit16] ENCLKHAPP0A0: HAPP0A0 Clock Oscillation Enable Bit

This bit sets the HAPP0A0 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit15] Reserved: Reserved Bit

[bit14] ENCLKSYSC1: ENCLKSYSC1 Clock Oscillation Enable Bit

This bit sets the ENCLKSYSC1 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit13:12] Reserved: Reserved Bits

[bit11] ENCLKEXTBUS: ENCLKEXTBUS Clock Oscillation Enable Bit

This bit sets the ENCLKEXTBUS clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit10] ENCLKMEMC: ENCLKMEMC Clock Oscillation Enable Bit

This bit sets the ENCLKMEMC clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit9] ENCLKDMA: ENCLKDMA Clock Oscillation Enable Bit

This bit sets the ENCLKDMA clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit8] ENCLKHPM: ENCLKHPM Clock Oscillation Enable Bit

This bit sets the ENCLKHPM clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit7] ENCLKHPM2: ENCLKHPM2 Clock Oscillation Enable Bit

This bit sets the ENCLKHPM2 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit6] ENCLKTRC: ENCLKTRC Clock Oscillation Enable Bit

This bit sets the ENCLKTRC clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit5] ENCLKDBG: ENCLKDBG Clock Oscillation Enable Bit

This bit sets the ENCLKDBG clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit4] ENCLKATB: ENCLKATB Clock Oscillation Enable Bit

This bit sets the ENCLKATB clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit3:1] Reserved: Reserved Bits

[bit0] ENCLKCPU0: ENCLKCPU0 Clock Oscillation Enable Bit

This bit sets the ENCLKCPU0 clock.

Bit	Description
0	Disable clock oscillation.
1	-

5.12.5. PSS Clock Source Enable Register 1 (SYSC1_PSSCKER1)

The SYSC1_PSSCKER1register sets whether to enable/disable oscillation of the internal operating clock.

bit	31	30	29	28	27	26	25	24
Field	Reserved			ENCLK CD3B1	ENCLK CD3B0	ENCLK CD3A1	ENCLK CD3A0	ENCLK CD3
R/W Attribute	R0,WX			R/W1	R/W1	R/W1	R/W1	R/W1
Protection Attribute	WPS							
Initial Value	000			1	1	1	1	1

bit	23	22	21	20	19	18	17	16
Field	Reserved			ENCLK CD2B1	ENCLK CD2B0	ENCLK CD2A1	ENCLK CD2A0	ENCLK CD2
R/W Attribute	R0,WX			R/W1	R/W1	R/W1	R/W1	R/W1
Protection Attribute	WPS							
Initial Value	000			1	1	1	1	1

bit	15	14	13	12	11	10	9	8
Field	Reserved			ENCLK CD1B1	ENCLK CD1B0	ENCLK CD1A1	ENCLK CD1A0	ENCLK CD1
R/W Attribute	R0,WX			R/W1	R/W1	R/W1	R/W1	R/W1
Protection Attribute	WPS							
Initial Value	000			1	1	1	1	1

bit	7	6	5	4	3	2	1	0
Field	Reserved							ENCLK HSSPI
R/W Attribute	R0,WX							R/W
Protection Attribute	WPS							
Initial Value	0000000							1

[bit31:29] Reserved: Reserved Bits

[bit28] ENCLKCD3B1: CD3B1 Clock Oscillation Enable Bit

This bit sets the CD3B1 clock.

Bit	Description
0	Setting '0' is prohibited.
1	Enable clock oscillation.

[bit27] ENCLKCD3B0: CD3B0 Clock Oscillation Enable Bit

This bit sets the CD3B0 clock.

Bit	Description
0	Setting '0' is prohibited.
1	Enable clock oscillation.

[bit26] ENCLKCD3A1: CD3A1 Clock Oscillation Enable Bit

This bit sets the CD3A1 clock.

Bit	Description
0	Setting '0' is prohibited.
1	Enable clock oscillation.

[bit25] ENCLKCD3A0: CD3A0 Clock Oscillation Enable Bit

This bit sets the CD3A0 clock.

Bit	Description
0	Setting '0' is prohibited.
1	Enable clock oscillation.

[bit24] ENCLKCD3: CD3 Clock Oscillation Enable Bit

This bit sets the CD3 clock.

Bit	Description
0	Setting '0' is prohibited.
1	Enable clock oscillation.

[bit23:21] Reserved: Reserved Bits

[bit20] ENCLKCD2B1: CD2B1 Clock Oscillation Enable Bit

This bit sets the CD2B1 clock.

Bit	Description
0	Setting '0' is prohibited.
1	Enable clock oscillation.

[bit19] ENCLKCD2B0: CD2B0 Clock Oscillation Enable Bit

This bit sets the CD2B0 clock.

Bit	Description
0	Setting '0' is prohibited.
1	Enable clock oscillation.

[bit18] ENCLKCD2A1: CD2A1 Clock Oscillation Enable Bit

This bit sets the CD2A1 clock.

Bit	Description
0	Setting '0' is prohibited.
1	Enable clock oscillation.

[bit17] ENCLKCD2A0: CD2A0 Clock Oscillation Enable Bit

This bit sets the CD2A0 clock.

Bit	Description
0	Setting '0' is prohibited.
1	Enable clock oscillation.

[bit16] ENCLKCD2: CD2 Clock Oscillation Enable Bit

This bit sets the CD2 clock.

Bit	Description
0	Setting '0' is prohibited.
1	Enable clock oscillation.

[bit15:13] Reserved: Reserved Bits

[bit12] ENCLKCD1B1: CD1B1 Clock Oscillation Enable Bit

This bit sets the CD1B1 clock.

Bit	Description
0	Setting '0' is prohibited.
1	Enable clock oscillation.

[bit11] ENCLKCD1B0: CD1B0 Clock Oscillation Enable Bit

This bit sets the CD1B0 clock.

Bit	Description
0	Setting '0' is prohibited.
1	Enable clock oscillation.

[bit10] ENCLKCD1A1: CD1A1 Clock Oscillation Enable Bit

This bit sets the CD1A1 clock.

Bit	Description
0	Setting '0' is prohibited.
1	Enable clock oscillation.

[bit9] ENCLKCD1A0: CD1A0 Clock Oscillation Enable Bit

This bit sets the CD1A0 clock.

Bit	Description
0	Setting '0' is prohibited.
1	Enable clock oscillation.

[bit8] ENCLKCD1: CD1 Clock Oscillation Enable Bit

This bit sets the CD1 clock.

Bit	Description
0	Setting '0' is prohibited.
1	Enable clock oscillation.

[bit7:1] Reserved: Reserved Bits

[bit0] ENCLKHSSPI: HSSPI Clock Oscillation Enable Bit

This bit sets the HSSPI clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

5.12.6. PSS Clock Source Enable Register 2(SYSC1_PSSCKER2)

The SYSC1_PSSCKER2 register sets whether to enable/disable oscillation of the internal operating clock.

bit	31							16
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	WPS							
Initial Value	00000000_00000000							

bit	15	14	13	12	11	10	9	8
Field	Reserved			ENCLKCD5B1	ENCLKCD5B0	ENCLKCD5A1	ENCLKCD5A0	ENCLKCD5
R/W Attribute	R0,WX			R/W1	R/W1	R/W1	R/W1	R/W1
Protection Attribute	WPS							
Initial Value	000			1	1	1	1	1

bit	7	6	5	4	3	2	1	0
Field	Reserved			ENCLKCD4B1	ENCLKCD4B0	ENCLKCD4A1	ENCLKCD4A0	ENCLKCD4
R/W Attribute	R0,WX			R/W1	R/W1	R/W1	R/W1	R/W1
Protection Attribute	WPS							
Initial Value	000			1	1	1	1	1

[bit31:13] Reserved: Reserved Bits

[bit12] ENCLKCD5B1: CD5B1 Clock Oscillation Enable Bit

This bit sets the CD5B1 clock.

Bit	Description
0	Setting '0' is prohibited.
1	Enable clock oscillation.

[bit11] ENCLKCD5B0: CD5B0 Clock Oscillation Enable Bit

This bit sets the CD5B0 clock.

Bit	Description
0	Setting '0' is prohibited.
1	Enable clock oscillation.

[bit10] ENCLKCD5A1: CD5A1 Clock Oscillation Enable Bit

This bit sets the CD5A1 clock.

Bit	Description
0	Setting '0' is prohibited.
1	Enable clock oscillation.

[bit9] ENCLKCD5A0: CD5A0 Clock Oscillation Enable Bit

This bit sets the CD5A0 clock.

Bit	Description
0	Setting '0' is prohibited.
1	Enable clock oscillation.

[bit8] ENCLKCD5: CD5 Clock Oscillation Enable Bit

This bit sets the CD5 clock.

Bit	Description
0	Setting '0' is prohibited.
1	Enable clock oscillation.

[bit7:5] Reserved: Reserved Bits

[bit4] ENCLKCD4B1: CD4B1 Clock Oscillation Enable Bit

This bit sets the CD4B1 clock.

Bit	Description
0	Setting '0' is prohibited.
1	Enable clock oscillation.

[bit3] ENCLKCD4B0: CD4B0 Clock Oscillation Enable Bit

This bit sets the CD4B0 clock.

Bit	Description
0	Setting '0' is prohibited.
1	Enable clock oscillation.

[bit2] ENCLKCD4A1: CD4A1 Clock Oscillation Enable Bit

This bit sets the CD4A1 clock.

Bit	Description
0	Setting '0' is prohibited.
1	Enable clock oscillation.

[bit1] ENCLKCD4A0: CD4A0 Clock Oscillation Enable Bit

This bit sets the CD4A0 clock.

Bit	Description
0	Setting '0' is prohibited.
1	Enable clock oscillation.

[bit0] ENCLKCD4: CD4 Clock Oscillation Enable Bit

This bit sets the CD4 clock.

Bit	Description
0	Setting '0' is prohibited.
1	Enable clock oscillation.

5.12.7. PSS Clock Divider Register 0 (SYSC1_PSSCKDIVR0)

The SYSC1_PSSCKDIVR0 register sets the division ratio of each internal operating clock.

bit	31	30	29	28	27	26	25	24
Field	Reserved					HPMDIV		
R/W Attribute	R0,WX					R/W		
Protection Attribute	WPS							
Initial Value	00000					000		

bit	23	22	21	20	19	18	17	16
Field	Reserved			TRCDIV				
R/W Attribute	R0,WX			R/W				
Protection Attribute	WPS							
Initial Value	000			00000				

bit	15	14	13	12	11	10	9	8
Field	Reserved		DBGDIV		Reserved		ATBDIV	
R/W Attribute	R0,WX		R/W		R0,WX		R/W	
Protection Attribute	WPS							
Initial Value	00		00		00		00	

bit	7	6	5	4	3	2	1	0
Field	Reserved			SYSDIV				
R/W Attribute	R0,WX			R/W				
Protection Attribute	WPS							
Initial Value	000			00000				

[bit31:27] Reserved: Reserved Bits

[bit26:24] HPMDIV: HPM Clock Divider Setting Bits

These bits set the division ratio of the HPM clock, DMA clock, and MEMC clock from the SYS clock.

bit26:24	Description
000	No division
001	Divided by 2
010	Divided by 3
011	Divided by 4
100	Divided by 5
101	Divided by 6
110	Divided by 7
111	Divided by 8

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

[bit23:21] Reserved: Reserved Bits

[bit20:16] TRCDIV: TRC Clock Divider Setting Bits

These bits set the division ratio of the TRC clock from the source clock (clock domain TRC).

bit20:16	Description
00000	No division
00001	Divided by 2
00010	Divided by 3
00011	Divided by 4
. . .	. . .
11101	Divided by 30
11110	Divided by 31
11111	Divided by 32

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

[bit15:14] Reserved: Reserved Bits

[bit13:12] DBGDIV: DBG Clock Divider Setting Bits

These bits set the division ratio of the DBG clock from the ATB clock.

bit13:12	Description
00	No division
01	Divided by 2
10	Divided by 4
11	Divided by 8

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

[bit11:10] Reserved: Reserved Bits

[bit9:8] ATBDIV [1:0]: ATB Clock Divider Setting Bits

These bits set the division ratio of the ATB clock from the SYS clock.

bit9:8	Description
00	No division
01	Divided by 2
10	Divided by 4
11	Divided by 8

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

[bit7:5] Reserved: Reserved Bits

[bit4:0] SYSDIV: SYS Clock Divider Setting Bits

These bits set the division ratio of the CPU0 clock from the source clock (clock domain 0).

bit4:0	Description
00000	No division
00001	Divided by 2
00010	Divided by 3
...	...
11101	Divided by 30
11110	Divided by 31
11111	Divided by 32

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

5.12.8. PSS Clock Divider Register 1 (SYSC1_PSSCKDIVR1)

The SYSC1_PSSCKDIVR1 register sets the division ratio of each internal operating clock.

bit	31	30	29	28	27	26	25	24
Field	HAPP1B1DIV				HAPP1B0DIV			
R/W Attribute	R/W				R/W			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	23	22	21	20	19	18	17	16
Field	HAPP0A1DIV				HAPP0A0DIV			
R/W Attribute	R/W				R/W			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	15	14	13	12	11	10	9	8
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	WPS							
Initial Value	00000000							

bit	7	6	5	4	3	2	1	0
Field	SYSC1DIV				Reserved	EXTBUSDIV		
R/W Attribute	R/W				R0,WX	R/W		
Protection Attribute	WPS							
Initial Value	0000				0	000		

[bit31:28] HAPP1B1DIV: HAPP1B1 Clock Divider Setting Bits

These bits set the division ratio of the HAPP1B1 clock from the HAPP1B0 clock.

bit31:28	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
0011	Divided by 4
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

[bit27:24] HAPP1B0DIV: HAPP1B0 Clock Divider Setting Bits

These bits set the division ratio of the HAPP1B0 clock from the HPM clock.

bit27:24	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
0011	Divided by 4
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

[bit23:20] HAPP0A1DIV: HAPP0A1 Clock Divider Setting Bits

These bits set the division ratio of the HAPP0A1 clock from the HAPP0A0 clock.

bit23:20	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
0011	Divided by 4
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

[bit19:16] HAPP0A0DIV: HAPP0A0 Clock Divider Setting Bits

These bits set the division ratio of the HAPP0A0 clock from the HPM clock.

bit19:16	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
0011	Divided by 4
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

[bit15:8] Reserved: Reserved Bits

[bit7:4] SYSC1DIV: SYSC1 Clock Divider Setting Bits

These bits set the division ratio of the SYSC1 clock from the HPM clock.

bit7:4	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
0011	Divided by 4
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

[bit3] Reserved: Reserved Bit

[bit2:0] EXTBUSDIV: EXTBUS Clock Divider Setting Bits

These bits set the division ratio of the EXTBUS clock from the HPM clock.

bit2:0	Description
000	No division
001	Divided by 2
010	Divided by 4
011	Divided by 8
100	Divided by 16
101	Divided by 32
110	Divided by 64
111	Divided by 128

5.12.9. PSS Clock Divider Register 2 (SYSC1_PSSCKDIVR2)

The SYSC1_PSSCKDIVR2 register sets the division ratio of each internal operating clock.

bit	31	30	29	28	27	26	25	24
Field	Reserved				LAPP1DIV			
R/W Attribute	R0,WX				R/W			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	23	22	21	20	19	18	17	16
Field	Reserved				LAPP0DIV			
R/W Attribute	R0,WX				R/W			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	15	14	13	12	11	10	9	8
Field	Reserved				LCP1DIV			
R/W Attribute	R0,WX				R/W			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	7	6	5	4	3	2	1	0
Field	LCP0DIV				Reserved		LCPDIV	
R/W Attribute	R/W				R0,WX		R/W	
Protection Attribute	WPS							
Initial Value	0000				00		00	

[bit31:28] Reserved: Reserved Bits

[bit27:24] LAPP1DIV: LAPP1 Clock Divider Setting Bits

These bits set the division ratio of the LAPP1 clock from the CPU clock.

bit27:24	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
0011	Divided by 4
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

[bit23:20] Reserved: Reserved Bits

[bit19:16] LAPP0DIV: LAPP0 Clock Divider Setting Bits

These bits set the division ratio of the LAPP0 clock from the CPU clock.

bit19:16	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
0011	Divided by 4
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

[bit15:12] Reserved: Reserved Bits

[bit11:8] LCP1DIV: LCP1 Clock Divider Setting Bits

These bits set the division ratio of the LCP1 clock from the CPU clock.

bit11:8	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
0011	Divided by 4
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

[bit7:4] LCP0DIV: LCP0 Clock Divider Setting Bits

These bits set the division ratio of the LCP0 clock from the CPU clock.

bit7:4	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
0011	Divided by 4
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

[bit3:2] Reserved: Reserved Bits

[bit1:0] LCPDIV: LCP Clock Divider Setting Bits

These bits set the division ratio of the LCP clock from the CPU clock.

bit1:0	Description
00	No division
01	Divided by 2
10	Divided by 4
11	Divided by 8

5.12.10. PSS Clock Divider Register 3 (SYSC1_PSSCKDIVR3)

The SYSC1_PSSCKDIVR3 register sets the division ratio of each internal operating clock.

bit	31	30	29	28	27	26	25	24
Field	Reserved			LAPP1ADIV				
R/W Attribute	R0,WX			R/W				
Protection Attribute	WPS							
Initial Value	000			00000				

bit	23	22	21	20	19	18	17	16
Field	Reserved			LAPP0ADIV				
R/W Attribute	R0,WX			R/W				
Protection Attribute	WPS							
Initial Value	000			00000				

bit	15	14	13	12	11	10	9	8
Field	Reserved			LCP1ADIV				
R/W Attribute	R0,WX			R/W				
Protection Attribute	WPS							
Initial Value	000			00000				

bit	7	6	5	4	3	2	1	0
Field	Reserved			LCP0ADIV				
R/W Attribute	R0,WX			R/W				
Protection Attribute	WPS							
Initial Value	000			00000				

[bit31:29] Reserved: Reserved Bits

[bit28:24] LAPP1ADIV: LAPP1A Clock Divider Setting Bits

These bits set the division ratio of the LAPP1A clock from the CPU/PLL0 clock.

bit28:24	Description
00000	No division
00001	Divided by 2
00010	Divided by 3
00011	Divided by 4
...	...
11101	Divided by 30
11110	Divided by 31
11111	Divided by 32

[bit23:21] Reserved: Reserved Bits

[bit20:16] LAPP0ADIV: LAPP0A Clock Divider Setting Bits

These bits set the division ratio of the LAPP0A clock from the CPU/PLL0 clock.

bit20:16	Description
00000	No division
00001	Divided by 2
00010	Divided by 3
00011	Divided by 4
...	...
11101	Divided by 30
11110	Divided by 31
11111	Divided by 32

[bit15:13] Reserved: Reserved Bits

[bit12:8] LCP1ADIV: LCP1A Clock Divider Setting Bits

These bits set the division ratio of the LCP1A clock from the CPU/PLL0 clock.

bit12:8	Description
00000	No division
00001	Divided by 2
00010	Divided by 3
00011	Divided by 4
...	...
11101	Divided by 30
11110	Divided by 31
11111	Divided by 32

[bit7:5] Reserved: Reserved Bits

[bit4:0] LCP0ADIV: LCP0A Clock Divider Setting Bits

These bits set the division ratio of the LCP0A clock from the CPU/PLL0 clock.

bit4:0	Description
00000	No division
00001	Divided by 2
00010	Divided by 3
00011	Divided by 4
...	...
11101	Divided by 30
11110	Divided by 31
11111	Divided by 32

5.12.11. PSS Clock Divider Register 4 (SYSC1_PSSCKDIVR4)

The SYSC1_PSSCKDIVR4 register sets the division ratio of each internal operating clock.

bit	31	8
Field	Reserved	
R/W Attribute	R0,WX	
Protection Attribute	WPS	
Initial Value	00000000_00000000_00000000	

bit	7	6	5	4	3	2	1	0
Field	Reserved			HSSPIDIV				
R/W Attribute	R0,WX			R/W				
Protection Attribute	WPS							
Initial Value	000			00000				

[bit31:5] Reserved: Reserved Bits

[bit4:0] HSSPIDIV: HSSPI Clock Divider Setting Bits

These bits set the division ratio of the HSSPI clock from the source clock (HSSPI clock domain).

bit4:0	Description
00000	No division
00001	Divided by 2
00010	Divided by 3
...	...
11101	Divided by 30
11110	Divided by 31
11111	Divided by 32

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

5.12.12. PSS Clock Divider Register 5 (SYSC1_PSSCKDIVR5)

The SYSC1_PSSCKDIVR5 register sets the division ratio of each internal operating clock.

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	WPS							
Initial Value	00000000							

bit	23	22	21	20	19	18	17	16
Field	CD1B1DIV				CD1B0DIV			
R/W Attribute	R/W				R/W			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	15	14	13	12	11	10	9	8
Field	CD1A1DIV				CD1A0DIV			
R/W Attribute	R/W				R/W			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	7	6	5	4	3	2	1	0
Field	Reserved			CD1DIV				
R/W Attribute	R0,WX			R/W				
Protection Attribute	WPS							
Initial Value	000			00000				

[bit31:24] Reserved: Reserved Bits

[bit23:20] CD1B1DIV: CD1B1 Clock Divider Setting Bits

These bits set the division ratio of the CD1B1 clock from the CD1B0 clock.

bit23:20	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

[bit19:16] CD1B0DIV: CD1B0 Clock Divider Setting Bits

These bits set the division ratio of the CD1B0 clock from the CD1 clock.

bit19:16	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

[bit15:12] CD1A1DIV: CD1A1 Clock Divider Setting Bits

These bits set the division ratio of the CD1A1 clock from the CD1A0 clock.

bit15:12	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

[bit11:8] CD1A0DIV: CD1A0 Clock Divider Setting Bits

These bits set the division ratio of the CD1A0 clock from the CD1 clock.

bit11:8	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

[bit7:5] Reserved: Reserved Bits

[bit4:0] CD1DIV: CD1 Clock Divider Setting Bits

These bits set the division ratio of the CD1 clock from the source clock (clock domain 1).

bit4:0	Description
00000	No division
00001	Divided by 2
00010	Divided by 3
...	...
11101	Divided by 30
11110	Divided by 31
11111	Divided by 32

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

5.12.13. PSS Clock Divider Register 6 (SYSC1_PSSCKDIVR6)

The SYSC1_PSSCKDIVR6 register sets the division ratio of each internal operating clock.

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	WPS							
Initial Value	00000000							

bit	23	22	21	20	19	18	17	16
Field	CD2B1DIV				CD2B0DIV			
R/W Attribute	R/W				R/W			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	15	14	13	12	11	10	9	8
Field	CD2A1DIV				CD2A0DIV			
R/W Attribute	R/W				R/W			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	7	6	5	4	3	2	1	0
Field	Reserved			CD2DIV				
R/W Attribute	R0,WX			R/W				
Protection Attribute	WPS							
Initial Value	000			00000				

[bit31:24] Reserved: Reserved Bits

[bit23:20] CD2B1DIV: CD2B1 Clock Divider Setting Bits

These bits set the division ratio of the CD2B1 clock from the CD2B0 clock.

bit23:20	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

[bit19:16] CD2B0DIV: CD2B0 Clock Divider Setting Bits

These bits set the division ratio of the CD2B0 clock from the CD2 clock.

bit19:16	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

[bit15:12] CD2A1DIV: CD2A1 Clock Divider Setting Bits

These bits set the division ratio of the CD2A1 clock from the CD2A0 clock.

bit15:12	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

[bit11:8] CD2A0DIV: CD2A0 Clock Divider Setting Bits

These bits set the division ratio of the CD2A0 clock from the CD2 clock.

bit11:8	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

[bit7:5] Reserved: Reserved Bits

[bit4:0] CD2DIV: CD2 Clock Divider Setting Bits

These bits set the division ratio of the CD2 clock from the source clock (clock domain 2).

bit4:0	Description
00000	No division
00001	Divided by 2
00010	Divided by 3
...	...
11101	Divided by 30
11110	Divided by 31
11111	Divided by 32

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

5.12.14. PSS Clock Divider Register 7 (SYSC1_PSSCKDIVR7)

The SYSC1_PSSCKDIVR7 register sets the division ratio of each internal operating clock.

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	WPS							
Initial Value	00000000							

bit	23	22	21	20	19	18	17	16
Field	CD3B1DIV				CD3B0DIV			
R/W Attribute	R/W				R/W			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	15	14	13	12	11	10	9	8
Field	CD3A1DIV				CD3A0DIV			
R/W Attribute	R/W				R/W			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	7	6	5	4	3	2	1	0
Field	Reserved			CD3DIV				
R/W Attribute	R0,WX			R/W				
Protection Attribute	WPS							
Initial Value	000			00000				

[bit31:24] Reserved: Reserved Bits

[bit23:20] CD3B1DIV: CD3B1 Clock Divider Setting Bits

These bits set the division ratio of the CD3B1 clock from the CD3B0 clock.

bit23:20	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

[bit19:16] CD3B0DIV: CD3B0 Clock Divider Setting Bits

These bits set the division ratio of the CD3B0 clock from the CD3 clock.

bit19:16	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

[bit15:12] CD3A1DIV: CD3A1 Clock Divider Setting Bits

These bits set the division ratio of the CD3A1 clock from the CD3A0 clock.

bit15:12	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

[bit11:8] CD3A0DIV: CD3A0 Clock Divider Setting Bits

These bits set the division ratio of the CD3A0 clock from the CD3 clock.

bit11:8	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

[bit7:5] Reserved: Reserved Bits

[bit4:0] CD3DIV: CD3 Clock Divider Setting Bits

These bits set the division ratio of the CD3 clock from the source clock (clock domain 3).

bit4:0	Description
00000	No division
00001	Divided by 2
00010	Divided by 3
...	...
11101	Divided by 30
11110	Divided by 31
11111	Divided by 32

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

5.12.15. PSS Clock Divider Register 8 (SYSC1_PSSCKDIVR8)

The SYSC1_PSSCKDIVR8 register sets the division ratio of each internal operating clock.

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	WPS							
Initial Value	00000000							

bit	23	22	21	20	19	18	17	16
Field	CD4B1DIV				CD4B0DIV			
R/W Attribute	R/W				R/W			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	15	14	13	12	11	10	9	8
Field	CD4A1DIV				CD4A0DIV			
R/W Attribute	R/W				R/W			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	7	6	5	4	3	2	1	0
Field	Reserved			CD4DIV				
R/W Attribute	R0,WX			R/W				
Protection Attribute	WPS							
Initial Value	000			00000				

[bit31:24] Reserved: Reserved Bits

[bit23:20] CD4B1DIV: CD4B1 Clock Divider Setting Bits

These bits set the division ratio of the CD4B1 clock from the CD4B0 clock.

bit23:20	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

[bit19:16] CD4B0DIV: CD4B0 Clock Divider Setting Bits

These bits set the division ratio of the CD4B0 clock from the CD4 clock.

bit19:16	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

[bit15:12] CD4A1DIV: CD4A1 Clock Divider Setting Bits

These bits set the division ratio of the CD4A1 clock from the CD4A0 clock.

bit15:12	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

[bit11:8] CD4A0DIV: CD4A0 Clock Divider Setting Bits

These bits set the division ratio of the CD4A0 clock from the CD4 clock.

bit11:8	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

[bit7:5] Reserved: Reserved Bits

[bit4:0] CD4DIV: CD4 Clock Divider Setting Bits

These bits set the division ratio of the CD4 clock from the source clock (clock domain 4).

bit4:0	Description
00000	No division
00001	Divided by 2
00010	Divided by 3
...	...
11101	Divided by 30
11110	Divided by 31
11111	Divided by 32

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

5.12.16. PSS Clock Divider Register 9 (SYSC1_PSSCKDIVR9)

The SYSC1_PSSCKDIVR9 register sets the division ratio of each internal operating clock.

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	WPS							
Initial Value	00000000							

bit	23	22	21	20	19	18	17	16
Field	CD5B1DIV				CD5B0DIV			
R/W Attribute	R/W				R/W			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	15	14	13	12	11	10	9	8
Field	CD5A1DIV				CD5A0DIV			
R/W Attribute	R/W				R/W			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	7	6	5	4	3	2	1	0
Field	Reserved			CD5DIV				
R/W Attribute	R0,WX			R/W				
Protection Attribute	WPS							
Initial Value	000			00000				

[bit31:24] Reserved: Reserved Bits

[bit23:20] CD5B1DIV: CD5B1 Clock Divider Setting Bits

These bits set the division ratio of the CD5B1 clock from the CD5B0 clock.

bit23:20	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

[bit19:16] CD5B0DIV: CD5B0 Clock Divider Setting Bits

These bits set the division ratio of the CD5B0 clock from the CD5 clock.

bit19:16	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

[bit15:12] CD5A1DIV: CD5A1 Clock Divider Setting Bits

These bits set the division ratio of the CD5A1 clock from the CD5A0 clock.

bit15:12	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

[bit11:8] CD5A0DIV: CD5A0 Clock Divider Setting Bits

These bits set the division ratio of the CD5A0 clock from the CD5 clock.

bit11:8	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

[bit7:5] Reserved: Reserved Bits

[bit4:0] CD5DIV: CD5 Clock Divider Setting Bits

These bits set the division ratio of the CD5 clock from the source clock (clock domain 5).

bit4:0	Description
00000	No division
00001	Divided by 2
00010	Divided by 3
...	...
11101	Divided by 30
11110	Divided by 31
11111	Divided by 32

Note:

- *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

5.12.17. PSS Profile Update Enable Register (SYSC1_PSSENR)

The SYSC1_PSSENR register sets the profile update enable.

bit	31							8
Field	Reserved							
R/W Attribute	R0, WX							
Protection Attribute	WPS							
Initial Value	00000000_00000000_00000000							

bit	7	6	5	4	3	2	1	0
Field	PSSEN1							
R/W Attribute	R,W							
Protection Attribute	WPS							
Initial Value	00000000							

[bit31:8] Reserved：Reserved Bits

[bit7:0] PSSEN1：PSS Profile Update Enable Bit

This bit is the PSS profile update enable register.

bit7:0	Description
During write operation	When "0xBA" is written: PSSEN is valid When "0x00" is written: PSSEN is invalid Other: Write operation is invalid
During read operation	When "0xBA" is written: "0xBA" is read Other: "0x00" is read

Notes:

- *This register is cleared to "0x00" by hardware after PSS profile (configuration of SYSC1 register) is updated.*
- *This register permits byte access only. Any other access is invalid. If the above is not complied with, a bus error occurs.*

5.13. APP Profile Register Group (SYSC1)

The registers in this group are APP profile control setting registers.

5.13.1. APP Clock Selection Register 0 (SYSC1_APPCKSELR0)

The SYSC1_APPCKSELR0 register indicates the set values of the clock source for the clock domain to be updated.

bit	31	30	29	28	27	26	25	24
Field	Reserved				HSSPICSL			
R/W Attribute	R0,WX				R,WX			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	23	22	21	20	19	18	17	16
Field	Reserved			LAPP1ACSL	Reserved			LAPP0ACSL
R/W Attribute	R0,WX			R,WX	R0,WX			R,WX
Protection Attribute	WPS							
Initial Value	000			0	000			0

bit	15	14	13	12	11	10	9	8
Field	Reserved			LCP1ACSL	Reserved			LCP0ACSL
R/W Attribute	R0,WX			R,WX	R0,WX			R,WX
Protection Attribute	WPS							
Initial Value	000			0	000			0

bit	7	6	5	4	3	2	1	0
Field	Reserved					CD0CSL		
R/W Attribute	R0,WX					R,WX		
Protection Attribute	WPS							
Initial Value	00000					000		

[bit31:28] Reserved: Reserved Bits

[bit27:24] HSSPICSL: HSSPI Clock Selection Bits

These bits indicate the set value of HSSPI clock to be updated.

bit27:24	Description
0000	High-speed CR clock
0001	Low-speed CR clock
0010	Main clock
0011	Sub clock (Do not use this setting)

bit27:24	Description
0100	PLL0 clock
0101	PLL1 clock
0110	PLL2 clock
0111	PLL3 clock
1000	SSCG PLL0 clock
1001	SSCG PLL1 clock
1010	SSCG PLL2 clock
1011	SSCG PLL3 clock
1100	Setting prohibited
1101	Setting prohibited
1110	Setting prohibited
1111	Clock fixed at "L"

[bit23:21] Reserved: Reserved Bits

[bit20] LAPP1ACSL: LAPP1A Clock Selection Bits

These bits indicate the set value of LAPP1A clock to be updated.

bit20	Description
0	CD0 clock
1	PLL0 clock

[bit19:17] Reserved: Reserved Bits

[bit16] LAPP0ACSL: LAPP0A Clock Selection Bits

These bits indicate the set value of LAPP0A clock to be updated.

bit16	Description
0	CD0 clock
1	PLL0 clock

[bit15:13] Reserved: Reserved Bits

[bit12] LCP1ACSL: LCP1A Clock Selection Bits

These bits indicate the set value of LCP1A clock to be updated.

bit12	Description
0	CD0 clock
1	PLL0 clock

[bit11:9] Reserved: Reserved Bits

[bit8] LCP0ACSL: LCP0A Clock Selection Bits

These bits indicate the set value of LCP0A clock to be updated.

bit8	Description
0	CD0 clock
1	PLL0 clock

[bit7:3] Reserved: Reserved Bits

[bit2:0] CD0CSL: Clock Domain 0 Clock Selection Bits

These bits indicate the set value of clock domain 0 to be updated.

bit2:0	Description
000	High-speed CR clock
001	Low-speed CR clock
010	Main clock
011	Sub clock (Do not use this setting)
100	PLL0 clock
101	SSCG0 clock
110	Setting prohibited
111	Clock fixed at "L"

5.13.2. APP Clock Selection Register 1 (SYSC1_APPCKSELR1)

The SYSC1_APPCKSELR1 register indicates the set values of the clock source for the clock domain to be updated.

bit	31	30	29	28	27	26	25	24
Field	Reserved				CD4CSL			
R/W Attribute	R0,WX				R,WX			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	23	22	21	20	19	18	17	16
Field	Reserved				CD3CSL			
R/W Attribute	R0,WX				R,WX			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	15	14	13	12	11	10	9	8
Field	Reserved				CD2CSL			
R/W Attribute	R0,WX				R,WX			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	7	6	5	4	3	2	1	0
Field	Reserved				CD1CSL			
R/W Attribute	R0,WX				R,WX			
Protection Attribute	WPS							
Initial Value	0000				0000			

[bit31:28] Reserved: Reserved Bits

[bit27:24] CD4CSL: Clock Domain 4 Clock Selection Bits

These bits indicate the set value of clock domain 4 to be updated.

bit27:24	Description
0000	High-speed CR clock
0001	Low-speed CR clock
0010	Main clock
0011	Sub clock (Do not use this setting)
0100	PLL0 clock
0101	PLL1 clock
0110	PLL2 clock
0111	PLL3 clock
1000	SSCG PLL0 clock

bit27:24	Description
1001	SSCG PLL1 clock
1010	SSCG PLL2 clock
1011	SSCG PLL3 clock
1100	Setting prohibited
1101	Setting prohibited
1110	Setting prohibited
1111	Clock fixed at "L"

[bit23:20] Reserved: Reserved Bits

[bit19:16] CD3CSL: Clock Domain 3 Clock Selection Bits

These bits indicate the set value of clock domain 3 to be updated.

bit19:16	Description
0000	High-speed CR clock
0001	Low-speed CR clock
0010	Main clock
0011	Sub clock (Do not use this setting)
0100	PLL0 clock
0101	PLL1 clock
0110	PLL2 clock
0111	PLL3 clock
1000	SSCG PLL0 clock
1001	SSCG PLL1 clock
1010	SSCG PLL2 clock
1011	SSCG PLL3 clock
1100	Setting prohibited
1101	Setting prohibited
1110	Setting prohibited
1111	Clock fixed at "L"

[bit15:12] Reserved: Reserved Bits

[bit11:8] CD2CSL: Clock Domain 2 Clock Selection Bits

These bits indicate the set value of clock domain 2 to be updated.

bit11:8	Description
0000	High-speed CR clock
0001	Low-speed CR clock
0010	Main clock
0011	Sub clock (Do not use this setting)
0100	PLL0 clock
0101	PLL1 clock

bit11:8	Description
0110	PLL2 clock
0111	PLL3 clock
1000	SSCG PLL0 clock
1001	SSCG PLL1 clock
1010	SSCG PLL2 clock
1011	SSCG PLL3 clock
1100	Setting prohibited
1101	Setting prohibited
1110	Setting prohibited
1111	Clock fixed at "L"

[bit7:4] Reserved: Reserved Bits

[bit3:0] CD1CSL: Clock Domain 1 Clock Selection Bits

These bits indicate the set value of clock domain 1 to be updated.

bit3:0	Description
0000	High-speed CR clock
0001	Low-speed CR clock
0010	Main clock
0011	Sub clock (Do not use this setting)
0100	PLL0 clock
0101	PLL1 clock
0110	PLL2 clock
0111	PLL3 clock
1000	SSCG PLL0 clock
1001	SSCG PLL1 clock
1010	SSCG PLL2 clock
1011	SSCG PLL3 clock
1100	Setting prohibited
1101	Setting prohibited
1110	Setting prohibited
1111	Clock fixed at "L"

5.13.3. APP Clock Selection Register 2 (SYSC1_APPCKSELR2)

The SYSC1_APPCKSELR2 register indicates the set values of the clock source for the clock domain to be updated.

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	Reserved															
R/W Attribute	R0,WX															
Protection Attribute	WPS															
Initial Value	00000000_00000000															

bit	15	14	13	12	11	10	9	8
Field	Reserved					TRCCSL		
R/W Attribute	R0,WX					R,WX		
Protection Attribute	WPS							
Initial Value	00000					000		

bit	7	6	5	4	3	2	1	0
Field	Reserved				CD5CSL			
R/W Attribute	R0,WX				R,WX			
Protection Attribute	WPS							
Initial Value	0000				0000			

[bit31:11] Reserved: Reserved Bits

[bit10:8] TRCCSL: TRC Clock Selection Bits

These bits select the source clock for TRC clock.

bit10:8	Description
000	High-speed CR clock
001	Low-speed CR clock
010	Main clock
011	Sub clock (Do not use this setting)
100	PLL0 clock
101	SSCG0 clock
110	PLL1 clock
111	Clock fixed at "L"

[bit7:4] Reserved: Reserved Bits

[bit3:0] CD5CSL: Clock Domain 5 Clock Selection Bits

These bits indicate the set value of clock domain 5 to be updated.

bit3:0	Description
0000	High-speed CR clock
0001	Low-speed CR clock
0010	Main clock
0011	Sub clock (Do not use this setting)
0100	PLL0 clock
0101	PLL1 clock
0110	PLL2 clock
0111	PLL3 clock
1000	SSCG PLL0 clock
1001	SSCG PLL1 clock
1010	SSCG PLL2 clock
1011	SSCG PLL3 clock
1100	Setting prohibited
1101	Setting prohibited
1110	Setting prohibited
1111	Clock fixed at "L"

5.13.4. APP Clock Source Enable Register0 (SYSC1_APPCKER0)

The SYSC1_APPCKER0 register indicates the value for setting whether to enable/disable oscillation of the internal operating clock to be updated.

bit	31	30	29	28	27	26	25	24
Field	ENCLK LAPP1A	ENCLK LAPP0A	ENCLK LCP1A	ENCLK LCP0A	ENCLK LAPP1	ENCLK LAPP0	ENCLK LCP1	ENCLK LCP0
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	WPS							
Initial Value	1	1	1	1	1	1	1	1

bit	23	22	21	20	19	18	17	16
Field	ENCLK LCP	Reserved	ENCLK LLPBM2	ENCLK LLPBM	ENCLK HAPP1B1	ENCLK HAPP1B0	ENCLK HAPP0A1	ENCLK HAPP0A0
R/W Attribute	R,WX	R0,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	WPS							
Initial Value	1	0	1	1	1	1	1	1

bit	15	14	13	12	11	10	9	8
Field	Reserved	ENCLK SYSC1	Reserved		ENCLK EXTBUS	ENCLK MEMC	ENCLK DMA	ENCLK HPM
R/W Attribute	R0,WX	R,WX	R0,WX		R,WX	R,WX	R,WX	R,WX
Protection Attribute	WPS							
Initial Value	0	1	00		1	1	1	1

bit	7	6	5	4	3	2	1	0
Field	ENCLK HPM2	ENCLK TRC	ENCLK DBG	ENCLK ATB	Reserved			ENCLK CPU0
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R0,WX			R,WX
Protection Attribute	WPS							
Initial Value	1	1	1	1	000			1

[bit31] ENCLKLAPP1A: LAPP1A Clock Oscillation Enable Bit

This bit indicates the set value of the LAPP1A clock to be updated.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit30] ENCLKLAPP0A: LAPP0A Clock Oscillation Enable Bit

This bit indicates the set value of the LAPP0A clock to be updated.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit29] ENCLKLCP1A: LCP1A Clock Oscillation Enable Bit

This bit indicates the set value of the LCP1A clock to be updated.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit28] ENCLKLCP0A: LCP0A Clock Oscillation Enable Bit

This bit indicates the set value of the LCP0A clock to be updated.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit27] ENCLKLAPP1: LAPP1 Clock Oscillation Enable Bit

This bit indicates the set value of the LAPP1 clock to be updated.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit26] ENCLKLAPP0: LAPP0 Clock Oscillation Enable Bit

This bit indicates the set value of the LAPP0 clock to be updated.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit25] ENCLKLCP1: LCP1 Clock Oscillation Enable Bit

This bit indicates the set value of the LCP1 clock to be updated.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit24] ENCLKLCP0: LCP0 Clock Oscillation Enable Bit

This bit indicates the set value of the LCP0 clock to be updated.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit23] ENCLKLCP: LCP Clock Oscillation Enable Bit

This bit indicates the set value of the LCP clock to be updated.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit22] Reserved: Reserved Bit

[bit21] ENCLKLLPBM2: LLPBM2 Clock Oscillation Enable Bit

This bit indicates the set value of the LLPBM2 clock to be updated.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit20] ENCLKLLPBM: LLPBM Clock Oscillation Enable Bit

This bit indicates the set value of the LLPBM clock to be updated.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit19] ENCLKHAPP1B1: HAPP1B1 Clock Oscillation Enable Bit

This bit indicates the set value of the HAPP1B1 clock to be updated.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit18] ENCLKHAPP1B0: HAPP1B0 Clock Oscillation Enable Bit

This bit indicates the set value of the HAPP1B0 clock to be updated.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit17] ENCLKHAPP0A1: HAPP0A1 Clock Oscillation Enable Bit

This bit indicates the set value of the HAPP0A1 clock to be updated.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit16] ENCLKHAPP0A0: HAPP0A0 Clock Oscillation Enable Bit

This bit indicates the set value of the HAPP0A0 clock to be updated.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit15] Reserved: Reserved Bit

[bit14] ENCLKSYSC1: ENCLKSYSC1 Clock Oscillation Enable Bit

This bit indicates the set value of the ENCLKSYSC1 clock to be updated.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit13:12] Reserved: Reserved Bits

[bit11] ENCLKEXTBUS: ENCLKEXTBUS Clock Oscillation Enable Bit

This bit indicates the set value of the ENCLKEXTBUS clock to be updated.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit10] ENCLKMEMC: ENCLKMEMC Clock Oscillation Enable Bit

This bit indicates the set value of the ENCLKMEMC clock to be updated.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit9] ENCLKDMA: ENCLKDMA Clock Oscillation Enable Bit

This bit indicates the set value of the ENCLKDMA clock to be updated.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit8] ENCLKHPM: ENCLKHPM Clock Oscillation Enable Bit

This bit indicates the set value of the ENCLKHPM clock to be updated.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit7] ENCLKHPM2: ENCLKHPM2 Clock Oscillation Enable Bit

This bit indicates the set value of the ENCLKHPM2 clock to be updated.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit6] ENCLKTRC: ENCLKTRC Clock Oscillation Enable Bit

This bit indicates the set value of the ENCLKTRC clock to be updated.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit5] ENCLKDBG: ENCLKDBG Clock Oscillation Enable Bit

This bit indicates the set value of the ENCLKDBG clock to be updated.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit4] ENCLKATB: ENCLKATB Clock Oscillation Enable Bit

This bit indicates the set value of the ENCLKATB clock to be updated.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit3:1] Reserved: Reserved Bits

[bit0] ENCLKCPU0: ENCLKCPU0 Clock Oscillation Enable Bit

This bit indicates the set value of the ENCLKCPU0 clock to be updated.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

5.13.5. APP Clock Source Enable Register1 (SYSC1_APPCKER1)

The SYSC1_APPCKER1 register indicates the value for setting whether to enable/disable oscillation of the internal operating clock to be updated.

bit	31	30	29	28	27	26	25	24
Field	Reserved			ENCLK CD3B1	ENCLK CD3B0	ENCLK CD3A1	ENCLK CD3A0	ENCLK CD3
R/W Attribute	R0,WX			R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	WPS							
Initial Value	000			1	1	1	1	1

bit	23	22	21	20	19	18	17	16
Field	Reserved			ENCLK CD2B1	ENCLK CD2B0	ENCLK CD2A1	ENCLK CD2A0	ENCLK CD2
R/W Attribute	R0,WX			R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	WPS							
Initial Value	000			1	1	1	1	1

bit	15	14	13	12	11	10	9	8
Field	Reserved			ENCLK CD1B1	ENCLK CD1B0	ENCLK CD1A1	ENCLK CD1A0	ENCLK CD1
R/W Attribute	R0,WX			R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	WPS							
Initial Value	000			1	1	1	1	1

bit	7	6	5	4	3	2	1	0
Field	Reserved							ENCLK HSSPI
R/W Attribute	R0,WX							R,WX
Protection Attribute	WPS							
Initial Value	0000000							1

[bit31:29] Reserved: Reserved Bits

[bit28] ENCLKCD3B1: CD3B1 Clock Oscillation Enable Bit

This bit indicates the set value of the CD3B1 clock to be updated.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit27] ENCLKCD3B0: CD3B0 Clock Oscillation Enable Bit

This bit indicates the set value of the CD3B0 clock to be updated.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit26] ENCLKCD3A1: CD3A1 Clock Oscillation Enable Bit

This bit indicates the set value of the CD3A1 clock to be updated.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit25] ENCLKCD3A0: CD3A0 Clock Oscillation Enable Bit

This bit indicates the set value of the CD3A0 clock to be updated.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit24] ENCLKCD3: CD3 Clock Oscillation Enable Bit

This bit indicates the set value of the CD3 clock to be updated.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit23:21] Reserved: Reserved Bits

[bit20] ENCLKCD2B1: CD2B1 Clock Oscillation Enable Bit

This bit indicates the set value of the CD2B1 clock to be updated.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit19] ENCLKCD2B0: CD2B0 Clock Oscillation Enable Bit

This bit indicates the set value of the CD2B0 clock to be updated.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit18] ENCLKCD2A1: CD2A1 Clock Oscillation Enable Bit

This bit indicates the set value of the CD2A1 clock to be updated.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit17] ENCLKCD2A0: CD2A0 Clock Oscillation Enable Bit

This bit indicates the set value of the CD2A0 clock to be updated.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit16] ENCLKCD2: CD2 Clock Oscillation Enable Bit

This bit indicates the set value of the CD2 clock to be updated.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit15:13] Reserved: Reserved Bits

[bit12] ENCLKCD1B1: CD1B1 Clock Oscillation Enable Bit

This bit indicates the set value of the CD1B1 clock to be updated.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit11] ENCLKCD1B0: CD1B0 Clock Oscillation Enable Bit

This bit indicates the set value of the CD1B0 clock to be updated.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit10] ENCLKCD1A1: CD1A1 Clock Oscillation Enable Bit

This bit indicates the set value of the CD1A1 clock to be updated.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit9] ENCLKCD1A0: CD1A0 Clock Oscillation Enable Bit

This bit indicates the set value of the CD1A0 clock to be updated.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit8] ENCLKCD1: CD1 Clock Oscillation Enable Bit

This bit indicates the set value of the CD1 clock to be updated.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit7:1] Reserved: Reserved Bits

[bit0] ENCLKHSSPI: ENCLKHSSPI Clock Oscillation Enable Bit

This bit indicates the set value of the ENCLKHSSPI clock to be updated.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

5.13.6. APP Clock Source Enable Register2 (SYSC1_APPCKER2)

The SYSC1_APPCKER2 register indicates the value for setting whether to enable/disable oscillation of the internal operating clock to be updated.

bit	31							16
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	WPS							
Initial Value	00000000_00000000							

bit	15	14	13	12	11	10	9	8
Field	Reserved			ENCLK CD5B1	ENCLK CD5B0	ENCLK CD5A1	ENCLK CD5A0	ENCLK CD5
R/W Attribute	R0,WX			R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	WPS							
Initial Value	000			1	1	1	1	1

bit	7	6	5	4	3	2	1	0
Field	Reserved			ENCLK CD4B1	ENCLK CD4B0	ENCLK CD4A1	ENCLK CD4A0	ENCLK CD4
R/W Attribute	R0,WX			R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	WPS							
Initial Value	000			1	1	1	1	1

[bit31:13] Reserved: Reserved Bits

[bit12] ENCLKCD5B1: CD5B1 Clock Oscillation Enable Bit

This bit indicates the set value of the CD5B1 clock to be updated.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit11] ENCLKCD5B0: CD5B0 Clock Oscillation Enable Bit

This bit indicates the set value of the CD5B0 clock to be updated.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit10] ENCLKCD5A1: CD5A1 Clock Oscillation Enable Bit

This bit indicates the set value of the CD5A1 clock to be updated.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit9] ENCLKCD5A0: CD5A0 Clock Oscillation Enable Bit

This bit indicates the set value of the CD5A0 clock to be updated.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit8] ENCLKCD5: CD5 Clock Oscillation Enable Bit

This bit indicates the set value of the CD5 clock to be updated.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit7:5] Reserved: Reserved Bits

[bit4] ENCLKCD4B1: CD4B1 Clock Oscillation Enable Bit

This bit indicates the set value of the CD4B1 clock to be updated.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit3] ENCLKCD4B0: CD4B0 Clock Oscillation Enable Bit

This bit indicates the set value of the CD4B0 clock to be updated.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit2] ENCLKCD4A1: CD4A1 Clock Oscillation Enable Bit

This bit indicates the set value of the CD4A1 clock to be updated.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit1] ENCLKCD4A0: CD4A0 Clock Oscillation Enable Bit

This bit indicates the set value of the CD4A0 clock to be updated.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit0] ENCLKCD4: CD4 Clock Oscillation Enable Bit

This bit indicates the set value of the CD4 clock to be updated.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

5.13.7. APP Clock Divider Register 0 (SYSC1_APPCKDIVR0)

The SYSC1_APPCKDIVR0 register indicates the set values of the division ratio of each internal operating clock to be updated.

bit	31	30	29	28	27	26	25	24
Field	Reserved					HPMDIV		
R/W Attribute	R0,WX					R,WX		
Protection Attribute	WPS							
Initial Value	00000					000		

bit	23	22	21	20	19	18	17	16
Field	Reserved			TRCDIV				
R/W Attribute	R0,WX			R,WX				
Protection Attribute	WPS							
Initial Value	000			00000				

bit	15	14	13	12	11	10	9	8
Field	Reserved		DBGDIV		Reserved		ATBDIV	
R/W Attribute	R0,WX		R,WX		R0,WX		R,WX	
Protection Attribute	WPS							
Initial Value	00		00		00		00	

bit	7	6	5	4	3	2	1	0
Field	Reserved			SYSDIV				
R/W Attribute	R0,WX			R,WX				
Protection Attribute	WPS							
Initial Value	000			00000				

[bit31:27] Reserved: Reserved Bits

[bit26:24] HPMDIV: HPM Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the HPM clock, DMA clock, or MEMC clock from the SYS clock to be updated.

bit26:24	Description
000	No division
001	Divided by 2
010	Divided by 3
011	Divided by 4
100	Divided by 5
101	Divided by 6
110	Divided by 7
111	Divided by 8

[bit23:21] Reserved: Reserved Bits

[bit20:16] TRCDIV: TRC Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the TRC clock from the source clock (clock domain TRC) to be updated.

bit20:16	Description
00000	No division
00001	Divided by 2
00010	Divided by 3
00011	Divided by 4
. . .	. . .
11101	Divided by 30
11110	Divided by 31
11111	Divided by 32

[bit15:14] Reserved: Reserved Bits

[bit13:12] DBGDIV: DBG Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the DBG clock from the ATB clock to be updated.

bit13:12	Description
00	No division
01	Divided by 2
10	Divided by 4
11	Divided by 8

[bit11:10] Reserved: Reserved Bits

[bit9:8] ATBDIV [1:0]: ATB Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the ATB clock from the SYS clock to be updated.

bit9:8	Description
00	No division
01	Divided by 2
10	Divided by 4
11	Divided by 8

[bit7:5] Reserved: Reserved Bits

[bit4:0] SYSDIV: SYS Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the CPU0 clock from the source clock (clock domain 0).

bit4:0	Description
00000	No division
00001	Divided by 2
00010	Divided by 3
...	...
11101	Divided by 30
11110	Divided by 31
11111	Divided by 32

5.13.8. APP Clock Divider Register 1 (SYSC1_APPCKDIVR1)

The SYSC1_APPCKDIVR1 register indicates the set values of the division ratio of each internal operating clock to be updated.

bit	31	30	29	28	27	26	25	24
Field	HAPP1B1DIV				HAPP1B0DIV			
R/W Attribute	R,WX				R,WX			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	23	22	21	20	19	18	17	16
Field	HAPP0A1DIV				HAPP0A0DIV			
R/W Attribute	R,WX				R,WX			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	15	14	13	12	11	10	9	8
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	WPS							
Initial Value	00000000							

bit	7	6	5	4	3	2	1	0
Field	SYSC1DIV				Reserved	EXTBUSDIV		
R/W Attribute	R,WX				R0,WX	R,WX		
Protection Attribute	WPS							
Initial Value	0000				0	000		

[bit31:28] HAPP1B1DIV: HAPP1B1 Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the HAPP1B1 clock from the HAPP1B0 clock to be updated.

bit31:28	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
0011	Divided by 4
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

[bit27:24] HAPP1B0DIV: HAPP1B0 Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the HAPP1B0 clock from the HPM clock to be updated.

bit27:24	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
0011	Divided by 4
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

[bit23:20] HAPP0A1DIV: HAPP0A1 Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the HAPP0A1 clock from the HAPP0A0 clock to be updated.

bit23:20	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
0011	Divided by 4
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

[bit19:16] HAPP0A0DIV: HAPP0A0 Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the HAPP0A0 clock from the HPM clock to be updated.

bit19:16	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
0011	Divided by 4
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

[bit15:8] Reserved: Reserved Bits

[bit7:4] SYSC1DIV: SYSC1 Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the SYSC1 clock from the HPM clock to be updated.

bit7:4	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
0011	Divided by 4
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

[bit3] Reserved: Reserved Bit

[bit2:0] EXTBUSDIV: EXTBUS Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the EXTBUS clock from the HPM clock to be updated.

bit2:0	Description
000	No division
001	Divided by 2
010	Divided by 4
011	Divided by 8
100	Divided by 16
101	Divided by 32
110	Divided by 64
111	Divided by 128

5.13.9. APP Clock Divider Register 2 (SYSC1_APPCKDIVR2)

The SYSC1_APPCKDIVR2 register indicates the set values of the division ratio of each internal operating clock to be updated.

bit	31	30	29	28	27	26	25	24
Field	Reserved				LAPP1DIV			
R/W Attribute	R0,WX				R,WX			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	23	22	21	20	19	18	17	16
Field	Reserved				LAPP0DIV			
R/W Attribute	R0,WX				R, WX			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	15	14	13	12	11	10	9	8
Field	Reserved				LCP1DIV			
R/W Attribute	R0,WX				R,WX			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	7	6	5	4	3	2	1	0
Field	LCP0DIV				Reserved		LCPDIV	
R/W Attribute	R,WX				R0,WX		R,WX	
Protection Attribute	WPS							
Initial Value	0000				00		00	

[bit31:28] Reserved: Reserved Bits

[bit27:24] LAPP1DIV: LAPP1 Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the LAPP1 clock from the CPU clock to be updated.

bit27:24	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
0011	Divided by 4
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

[bit23:20] Reserved: Reserved Bits

[bit19:16] LAPP0DIV: LAPP0 Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the LAPP0 clock from the CPU clock to be updated.

bit19:16	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
0011	Divided by 4
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

[bit15:12] Reserved: Reserved Bits

[bit11:8] LCP1DIV: LCP1 Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the LCP1 clock from the CPU clock to be updated.

bit11:8	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
0011	Divided by 4
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

[bit7:4] LCP0DIV: LCP0 Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the LCP0 clock from the CPU clock to be updated.

bit7:4	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
0011	Divided by 4
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

[bit3:2] Reserved: Reserved Bits

[bit1:0] LCPDIV: LCP Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the LCP clock from the CPU clock to be updated.

bit1:0	Description
00	No division
01	Divided by 2
10	Divided by 4
11	Divided by 8

5.13.10. APP Clock Divider Register 3 (SYSC1_APPCKDIVR3)

The SYSC1_APPCKDIVR3 register indicates the set values of the division ratio of each internal operating clock to be updated.

bit	31	30	29	28	27	26	25	24
Field	Reserved			LAPP1ADIV				
R/W Attribute	R0,WX			R,WX				
Protection Attribute	WPS							
Initial Value	000			00000				

bit	23	22	21	20	19	18	17	16
Field	Reserved			LAPP0ADIV				
R/W Attribute	R0,WX			R,WX				
Protection Attribute	WPS							
Initial Value	000			00000				

bit	15	14	13	12	11	10	9	8
Field	Reserved			LCP1ADIV				
R/W Attribute	R0,WX			R,WX				
Protection Attribute	WPS							
Initial Value	000			00000				

bit	7	6	5	4	3	2	1	0
Field	Reserved			LCP0ADIV				
R/W Attribute	R0,WX			R,WX				
Protection Attribute	WPS							
Initial Value	000			00000				

[bit31:29] Reserved: Reserved Bits

[bit28:24] LAPP1ADIV: LAPP1A Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the LAPP1A clock from the CPU/PLL0 clock to be updated.

bit28:24	Description
00000	No division
00001	Divided by 2
00010	Divided by 3
00011	Divided by 4
...	...
11101	Divided by 30
11110	Divided by 31
11111	Divided by 32

[bit23:21] Reserved: Reserved Bits

[bit20:16] LAPP0ADIV: LAPP0A Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the LAPP0A clock from the CPU/PLL0 clock to be updated.

bit20:16	Description
00000	No division
00001	Divided by 2
00010	Divided by 3
00011	Divided by 4
...	...
11101	Divided by 30
11110	Divided by 31
11111	Divided by 32

[bit15:13] Reserved: Reserved Bits

[bit12:8] LCP1ADIV: LCP1A Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the LCP1A clock from the CPU/PLL0 clock to be updated.

bit12:8	Description
00000	No division
00001	Divided by 2
00010	Divided by 3
00011	Divided by 4
...	...
11101	Divided by 30
11110	Divided by 31
11111	Divided by 32

[bit7:5] Reserved: Reserved Bits

[bit4:0] LCP0ADIV: LCP0A Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the LCP0A clock from the CPU/PLL0 clock to be updated.

bit4:0	Description
00000	No division
00001	Divided by 2
00010	Divided by 3
00011	Divided by 4
...	...
11101	Divided by 30
11110	Divided by 31
11111	Divided by 32

5.13.11. APP Clock Divider Register 4 (SYSC1_APPCKDIVR4)

The SYSC1_APPCKDIVR4 register indicates the set values of the division ratio of each internal operating clock to be updated.

bit	31	8
Field	Reserved	
R/W Attribute	R0,WX	
Protection Attribute	WPS	
Initial Value	00000000_00000000_00000000	

bit	7	6	5	4	3	2	1	0
Field	Reserved			HSSPIDIV				
R/W Attribute	R0,WX			R,WX				
Protection Attribute	WPS							
Initial Value	000			00000				

[bit31:5] Reserved: Reserved Bits

[bit4:0] HSSPIDIV: HSSPI Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the HSSPI clock from the source clock (HSSPI clock domain) to be updated.

bit4:0	Description
00000	No division
00001	Divided by 2
00010	Divided by 3
...	...
11101	Divided by 30
11110	Divided by 31
11111	Divided by 32

5.13.12. APP Clock Divider Register 5 (SYSC1_APPCKDIVR5)

The SYSC1_APPCKDIVR5 register indicates the set values of the division ratio of each internal operating clock to be updated.

bit	31	24
Field	Reserved	
R/W Attribute	R0,WX	
Protection Attribute	WPS	
Initial Value	00000000	

bit	23	22	21	20	19	18	17	16
Field	CD1B1DIV				CD1B0DIV			
R/W Attribute	R,WX				R,WX			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	15	14	13	12	11	10	9	8
Field	CD1A1DIV				CD1A0DIV			
R/W Attribute	R,WX				R,WX			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	7	6	5	4	3	2	1	0
Field	Reserved			CD1DIV				
R/W Attribute	R0,WX			R,WX				
Protection Attribute	WPS							
Initial Value	000			00000				

[bit31:24] Reserved: Reserved Bits

[bit23:20] CD1B1DIV: CD1B1 Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the CD1B1 clock from the CD1B0 clock to be updated.

bit23:20	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

[bit19:16] CD1B0DIV: CD1B0 Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the CD1B0 clock from the CD1 clock to be updated.

bit19:16	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

[bit15:12] CD1A1DIV: CD1A1 Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the CD1A1 clock from the CD1A0 clock to be updated.

bit15:12	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

[bit11:8] CD1A0DIV: CD1A0 Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the CD1A0 clock from the CD1 clock to be updated.

bit11:8	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

[bit7:5] Reserved: Reserved Bits

[bit4:0] CD1DIV: CD1 Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the CD1 clock from the source clock (clock domain 1) to be updated.

bit4:0	Description
00000	No division
00001	Divided by 2
00010	Divided by 3
...	...
11101	Divided by 30
11110	Divided by 31
11111	Divided by 32

5.13.13. APP Clock Divider Register 6 (SYSC1_APPCKDIVR6)

The SYSC1_APPCKDIVR6 register indicates the set values of the division ratio of each internal operating clock to be updated.

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	WPS							
Initial Value	00000000							

bit	23	22	21	20	19	18	17	16
Field	CD2B1DIV				CD2B0DIV			
R/W Attribute	R,WX				R,WX			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	15	14	13	12	11	10	9	8
Field	CD2A1DIV				CD2A0DIV			
R/W Attribute	R,WX				R,WX			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	7	6	5	4	3	2	1	0
Field	Reserved			CD2DIV				
R/W Attribute	R0,WX			R,WX				
Protection Attribute	WPS							
Initial Value	000			00000				

[bit31:24] Reserved: Reserved Bits

[bit23:20] CD2B1DIV: CD2B1 Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the CD2B1 clock from the CD2B0 clock to be updated.

bit23:20	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

[bit19:16] CD2B0DIV: CD2B0 Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the CD2B0 clock from the CD2 clock to be updated.

bit19:16	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

[bit15:12] CD2A1DIV: CD2A1 Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the CD2A1 clock from the CD2A0 clock to be updated.

bit15:12	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

[bit11:8] CD2A0DIV: CD2A0 Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the CD2A0 clock from the CD2 clock to be updated.

bit11:8	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

[bit7:5] Reserved: Reserved Bits

[bit4:0] CD2DIV: CD2 Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the CD2 clock from the source clock (clock domain 2) to be updated.

bit4:0	Description
00000	No division
00001	Divided by 2
00010	Divided by 3
...	...
11101	Divided by 30
11110	Divided by 31
11111	Divided by 32

5.13.14. APP Clock Divider Register 7 (SYSC1_APPCKDIVR7)

The SYSC1_APPCKDIVR7 register indicates the set values of the division ratio of each internal operating clock to be updated.

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	WPS							
Initial Value	00000000							

bit	23	22	21	20	19	18	17	16
Field	CD3B1DIV				CD3B0DIV			
R/W Attribute	R,WX				R,WX			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	15	14	13	12	11	10	9	8
Field	CD3A1DIV				CD3A0DIV			
R/W Attribute	R,WX				R,WX			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	7	6	5	4	3	2	1	0
Field	Reserved			CD3DIV				
R/W Attribute	R0,WX			R,WX				
Protection Attribute	WPS							
Initial Value	000			00000				

[bit31:24] Reserved: Reserved Bits

[bit23:20] CD3B1DIV: CD3B1 Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the CD3B1 clock from the CD3B0 clock to be updated.

bit23:20	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

[bit19:16] CD3B0DIV: CD3B0 Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the CD3B0 clock from the CD3 clock to be updated.

bit19:16	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

[bit15:12] CD3A1DIV: CD3A1 Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the CD3A1 clock from the CD3A0 clock to be updated.

bit15:12	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

[bit11:8] CD3A0DIV: CD3A0 Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the CD3A0 clock from the CD3 clock to be updated.

bit11:8	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

[bit7:5] Reserved: Reserved Bits

[bit4:0] CD3DIV: CD3 Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the CD3 clock from the source clock (clock domain 3) to be updated.

bit4:0	Description
00000	No division
00001	Divided by 2
00010	Divided by 3
...	...
11101	Divided by 30
11110	Divided by 31
11111	Divided by 32

5.13.15. APP Clock Divider Register 8 (SYSC1_APPCKDIVR8)

The SYSC1_APPCKDIVR8 register indicates the set values of the division ratio of each internal operating clock to be updated.

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	WPS							
Initial Value	00000000							

bit	23	22	21	20	19	18	17	16
Field	CD4B1DIV				CD4B0DIV			
R/W Attribute	R,WX				R,WX			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	15	14	13	12	11	10	9	8
Field	CD4A1DIV				CD4A0DIV			
R/W Attribute	R,WX				R,WX			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	7	6	5	4	3	2	1	0
Field	Reserved			CD4DIV				
R/W Attribute	R0,WX			R,WX				
Protection Attribute	WPS							
Initial Value	000			00000				

[bit31:24] Reserved: Reserved Bits

[bit23:20] CD4B1DIV: CD4B1 Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the CD4B1 clock from the CD4B0 clock to be updated.

bit23:20	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

[bit19:16] CD4B0DIV: CD4B0 Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the CD4B0 clock from the CD4 clock to be updated.

bit19:16	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

[bit15:12] CD4A1DIV: CD4A1 Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the CD4A1 clock from the CD4A0 clock to be updated.

bit15:12	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

[bit11:8] CD4A0DIV: CD4A0 Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the CD4A0 clock from the CD4 clock to be updated.

bit11:8	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

[bit7:5] Reserved: Reserved Bits

[bit4:0] CD4DIV: CD4 Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the CD4 clock from the source clock (clock domain 4) to be updated.

bit4:0	Description
00000	No division
00001	Divided by 2
00010	Divided by 3
...	...
11101	Divided by 30
11110	Divided by 31
11111	Divided by 32

5.13.16. APP Clock Divider Register 9 (SYSC1_APPCKDIVR9)

The SYSC1_APPCKDIVR9 register indicates the set values of the division ratio of each internal operating clock to be updated.

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	WPS							
Initial Value	00000000							

bit	23	22	21	20	19	18	17	16
Field	CD5B1DIV				CD5B0DIV			
R/W Attribute	R,WX				R,WX			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	15	14	13	12	11	10	9	8
Field	CD5A1DIV				CD5A0DIV			
R/W Attribute	R,WX				R,WX			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	7	6	5	4	3	2	1	0
Field	Reserved			CD5DIV				
R/W Attribute	R0,WX			R,WX				
Protection Attribute	WPS							
Initial Value	000			00000				

[bit31:24] Reserved: Reserved Bits

[bit23:20] CD5B1DIV: CD5B1 Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the CD5B1 clock from the CD5B0 clock to be updated.

bit23:20	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

[bit19:16] CD5B0DIV: CD5B0 Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the CD5B0 clock from the CD5 clock to be updated.

bit19:16	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

[bit15:12] CD5A1DIV: CD5A1 Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the CD5A1 clock from the CD5A0 clock to be updated.

bit15:12	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

[bit11:8] CD5A0DIV: CD5A0 Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the CD5A0 clock from the CD5 clock to be updated.

bit11:8	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

[bit7:5] Reserved: Reserved Bits

[bit4:0] CD5DIV: CD5 Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the CD5 clock from the source clock (clock domain 4) to be updated.

bit4:0	Description
00000	No division
00001	Divided by 2
00010	Divided by 3
...	...
11101	Divided by 30
11110	Divided by 31
11111	Divided by 32

5.14. STS Profile Register Group (SYSC1)

The registers in this group are the STS profile control setting registers.

5.14.1. STS Clock Selection Register 0 (SYSC1_STSCKSELR0)

The SYSC1_STSCKSELR0 register indicates the set values of the clock source for the clock domain.

bit	31	30	29	28	27	26	25	24
Field	HSSPICM				HSSPICSL			
R/W Attribute	R,WX				R,WX			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	23	22	21	20	19	18	17	16
Field	Reserved		LAPP1ACM	LAPP1ACSL	Reserved		LAPP0ACM	LAPP0ACSL
R/W Attribute	R0,WX		R,WX	R,WX	R0,WX		R,WX	R,WX
Protection Attribute	WPS							
Initial Value	00		0	0	00		0	0

bit	15	14	13	12	11	10	9	8
Field	Reserved		LCP1ACM	LCP1ACSL	Reserved		LCP0ACM	LCP0ACSL
R/W Attribute	R0,WX		R,WX	R,WX	R0,WX		R,WX	R,WX
Protection Attribute	WPS							
Initial Value	00		0	0	00		0	0

bit	7	6	5	4	3	2	1	0
Field	Reserved	CD0CM			Reserved	CD0CSL		
R/W Attribute	R0,WX	R,WX			R0,WX	R,WX		
Protection Attribute	WPS							
Initial Value	0	000			0	000		

[bit31:28] HSSPICM: HSSPI Clock Selection Status Bits

These bits indicate the selection status of HSSPI clock.

bit31:28	Description
0000	High-speed CR clock
0001	Low-speed CR clock
0010	Main clock
0011	Sub clock (Do not use this setting)
0100	PLL0 clock
0101	PLL1 clock
0110	PLL2 clock
0111	PLL3 clock

bit31:28	Description
1000	SSCG PLL0 clock
1001	SSCG PLL1 clock
1010	SSCG PLL2 clock
1011	SSCG PLL3 clock
1100	Setting prohibited
1101	Setting prohibited
1110	Setting prohibited
1111	Clock fixed at "L"

[bit27:24] HSSPICSL: HSSPI Clock Selection Bits

These bits indicate the set value of HSSPI clock.

bit27:24	Description
0000	High-speed CR clock
0001	Low-speed CR clock
0010	Main clock
0011	Sub clock (Do not use this setting)
0100	PLL0 clock
0101	PLL1 clock
0110	PLL2 clock
0111	PLL3 clock
1000	SSCG PLL0 clock
1001	SSCG PLL1 clock
1010	SSCG PLL2 clock
1011	SSCG PLL3 clock
1100	Setting prohibited
1101	Setting prohibited
1110	Setting prohibited
1111	Clock fixed at "L"

[bit23:22] Reserved: Reserved Bits

[bit21] LAPP1ACM: LAPP1A Clock Selection Status Bit

This bit indicates the selection status of LAPP1A clock.

bit21	Description
0	CD0 clock
1	PLL0 clock

[bit20] LAPP1ACSL: LAPP1A Clock Selection Bit

This bit indicates the set value of LAPP1A clock.

bit20	Description
0	CD0 clock
1	PLL0 clock

[bit19:18] Reserved: Reserved Bits

[bit17] LAPP0ACM: LAPP0A Clock Selection Status Bit

This bit indicates the selection status of LAPP0A clock.

bit17	Description
0	CD0 clock
1	PLL0 clock

[bit16] LAPP0ACSL: LAPP0A Clock Selection Bit

This bit indicates the set value of LAPP0A clock.

bit16	Description
0	CD0 clock
1	PLL0 clock

[bit15:14] Reserved: Reserved Bits

[bit13] LCP1ACM: LCP1A Clock Selection Status Bit

This bit indicates the selection status of LCP1A clock.

bit13	Description
0	CD0 clock
1	PLL0 clock

[bit12] LCP1ACSL: LCP1A Clock Selection Bit

This bit indicates the set value of LCP1A clock.

bit12	Description
0	CD0 clock
1	PLL0 clock

[bit11:10] Reserved: Reserved Bits

[bit9] LCP0ACM: LCP0A Clock Selection Status Bit

This bit indicates the selection status of LCP0A clock.

bit9	Description
0	CD0 clock
1	PLL0 clock

[bit8] LCP0ACSL: LCP0A Clock Selection Bit

This bit indicates the set value of LCP0A clock.

bit8	Description
0	CD0 clock
1	PLL0 clock

[bit7] Reserved: Reserved Bit

[bit6:4] CD0CM: Clock Domain 0 Clock Selection Status Bits

These bits indicate the selection status of clock domain 0.

bit6:4	Description
000	High-speed CR clock
001	Low-speed CR clock
010	Main clock
011	Sub clock (Do not use this setting)
100	PLL0 clock
101	SSCG0 clock
110	Setting prohibited
111	Clock fixed at "L"

[bit3] Reserved: Reserved Bit

[bit2:0] CD0CSL: Clock Domain 0 Clock Selection Bits

These bits indicate the set value of clock domain 0.

bit2:0	Description
000	High-speed CR clock
001	Low-speed CR clock
010	Main clock
011	Sub clock (Do not use this setting)
100	PLL0 clock
101	SSCG0 clock
110	Setting prohibited
111	Clock fixed at "L"

5.14.2. STS Clock Selection Register 1 (SYSC1_STSCKSELR1)

The SYSC1_STSCKSELR1 register indicates the set values of the clock source for the clock domain.

bit	31	30	29	28	27	26	25	24
Field	CD4CM				CD4CSL			
R/W Attribute	R,WX				R,WX			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	23	22	21	20	19	18	17	16
Field	CD3CM				CD3CSL			
R/W Attribute	R,WX				R,WX			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	15	14	13	12	11	10	9	8
Field	CD2CM				CD2CSL			
R/W Attribute	R,WX				R,WX			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	7	6	5	4	3	2	1	0
Field	CD1CM				CD1CSL			
R/W Attribute	R,WX				R,WX			
Protection Attribute	WPS							
Initial Value	0000				0000			

[bit31:28] CD4CM: Clock Domain 4 Clock Selection Status Bits

These bits indicate the selection status of clock domain 4.

bit31:28	Description
0000	High-speed CR clock
0001	Low-speed CR clock
0010	Main clock
0011	Sub clock (Do not use this setting)
0100	PLL0 clock
0101	PLL1 clock
0110	PLL2 clock
0111	PLL3 clock
1000	SSCG PLL0 clock
1001	SSCG PLL1 clock
1010	SSCG PLL2 clock
1011	SSCG PLL3 clock
1100	Setting prohibited
1101	Setting prohibited

bit31:28	Description
1110	Setting prohibited
1111	Clock fixed at "L"

[bit27:24] CD4CSL: Clock Domain 4 Clock Selection Bits

These bits indicate the set value of clock domain 4.

bit27:24	Description
0000	High-speed CR clock
0001	Low-speed CR clock
0010	Main clock
0011	Sub clock (Do not use this setting)
0100	PLL0 clock
0101	PLL1 clock
0110	PLL2 clock
0111	PLL3 clock
1000	SSCG PLL0 clock
1001	SSCG PLL1 clock
1010	SSCG PLL2 clock
1011	SSCG PLL3 clock
1100	Setting prohibited
1101	Setting prohibited
1110	Setting prohibited
1111	Clock fixed at "L"

[bit23:20] CD3CM: Clock Domain 3 Clock Selection Status Bits

These bits indicate the selection status of clock domain 3.

bit23:20	Description
0000	High-speed CR clock
0001	Low-speed CR clock
0010	Main clock
0011	Sub clock (Do not use this setting)
0100	PLL0 clock
0101	PLL1 clock
0110	PLL2 clock
0111	PLL3 clock
1000	SSCG PLL0 clock
1001	SSCG PLL1 clock
1010	SSCG PLL2 clock
1011	SSCG PLL3 clock
1100	Setting prohibited
1101	Setting prohibited
1110	Setting prohibited
1111	Clock fixed at "L"

[bit19:16] CD3CSL: Clock Domain 3 Clock Selection Bits

These bits indicate the set value of clock domain 3.

bit19:16	Description
0000	High-speed CR clock
0001	Low-speed CR clock
0010	Main clock
0011	Sub clock (Do not use this setting)
0100	PLL0 clock
0101	PLL1 clock
0110	PLL2 clock
0111	PLL3 clock
1000	SSCG PLL0 clock
1001	SSCG PLL1 clock
1010	SSCG PLL2 clock
1011	SSCG PLL3 clock
1100	Setting prohibited
1101	Setting prohibited
1110	Setting prohibited
1111	Clock fixed at "L"

[bit15:12] CD2CM: Clock Domain 2 Clock Selection Status Bits

These bits indicate the selection status of clock domain 2.

bit15:12	Description
0000	High-speed CR clock
0001	Low-speed CR clock
0010	Main clock
0011	Sub clock (Do not use this setting)
0100	PLL0 clock
0101	PLL1 clock
0110	PLL2 clock
0111	PLL3 clock
1000	SSCG PLL0 clock
1001	SSCG PLL1 clock
1010	SSCG PLL2 clock
1011	SSCG PLL3 clock
1100	Setting prohibited
1101	Setting prohibited
1110	Setting prohibited
1111	Clock fixed at "L"

[bit11:8] CD2CSL: Clock Domain 2 Clock Selection Bits

These bits indicate the set value of clock domain 2.

bit11:8	Description
0000	High-speed CR clock
0001	Low-speed CR clock
0010	Main clock
0011	Sub clock (Do not use this setting)
0100	PLL0 clock
0101	PLL1 clock
0110	PLL2 clock
0111	PLL3 clock
1000	SSCG PLL0 clock
1001	SSCG PLL1 clock
1010	SSCG PLL2 clock
1011	SSCG PLL3 clock
1100	Setting prohibited
1101	Setting prohibited
1110	Setting prohibited
1111	Clock fixed at "L"

[bit7:4] CD1CM: Clock Domain 1 Clock Selection Status Bits

These bits indicate the selection status of clock domain 1.

bit7:4	Description
0000	High-speed CR clock
0001	Low-speed CR clock
0010	Main clock
0011	Sub clock (Do not use this setting)
0100	PLL0 clock
0101	PLL1 clock
0110	PLL2 clock
0111	PLL3 clock
1000	SSCG PLL0 clock
1001	SSCG PLL1 clock
1010	SSCG PLL2 clock
1011	SSCG PLL3 clock
1100	Setting prohibited
1101	Setting prohibited
1110	Setting prohibited
1111	Clock fixed at "L"

[bit3:0] CD1CSL: Clock Domain 1 Clock Selection Bits

These bits indicate the set value of clock domain 1.

bit3:0	Description
0000	High-speed CR clock
0001	Low-speed CR clock
0010	Main clock
0011	Sub clock (Do not use this setting)
0100	PLL0 clock
0101	PLL1 clock
0110	PLL2 clock
0111	PLL3 clock
1000	SSCG PLL0 clock
1001	SSCG PLL1 clock
1010	SSCG PLL2 clock
1011	SSCG PLL3 clock
1100	Setting prohibited
1101	Setting prohibited
1110	Setting prohibited
1111	Clock fixed at "L"

5.14.3. STS Clock Selection Register 2 (SYSC1_STSCKSELR2)

The SYSC1_STSCKSELR2 register indicates the set values of the clock source for the clock domain.

bit	31							16
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	WPS							
Initial Value	00000000_00000000							

bit	15	14	13	12	11	10	9	8
Field	Reserved	TRCCM			Reserved	TRCCSL		
R/W Attribute	R0,WX	R,WX			R0,WX	R,WX		
Protection Attribute	WPS							
Initial Value	0	000			0	000		

bit	7	6	5	4	3	2	1	0
Field	CD5CM				CD5CSL			
R/W Attribute	R0,WX				R,WX			
Protection Attribute	WPS							
Initial Value	0000				0000			

[bit31:15] Reserved: Reserved Bits

[bit14:12] TRCCM: TRC Clock Selection Status Bits

These bits indicate the selection status of TRC clock.

bit14:12	Description
000	High-speed CR clock
001	Low-speed CR clock
010	Main clock
011	Sub clock (Do not use this setting)
100	PLL0 clock
101	SSCG0 clock
110	PLL1 clock
111	Clock fixed at "L"

[bit11] Reserved: Reserved Bit

[bit10:8] TRCCSL: TRC Clock Selection Bits

These bits select the source clock for TRC clock.

bit10:8	Description
000	High-speed CR clock
001	Low-speed CR clock
010	Main clock
011	Sub clock (Do not use this setting)
100	PLL0 clock
101	SSCG0 clock
110	PLL1 clock
111	Clock fixed at "L"

[bit7:4] CD5CM: Clock Domain 5 Clock Selection Status Bits

These bits indicate the selection status of clock domain 5.

bit7:4	Description
0000	High-speed CR clock
0001	Low-speed CR clock
0010	Main clock
0011	Sub clock (Do not use this setting)
0100	PLL0 clock
0101	PLL1 clock
0110	PLL2 clock
0111	PLL3 clock
1000	SSCG PLL0 clock
1001	SSCG PLL1 clock
1010	SSCG PLL2 clock
1011	SSCG PLL3 clock
1100	Setting prohibited
1101	Setting prohibited
1110	Setting prohibited
1111	Clock fixed at "L"

[bit3:0] CD5CSL: Clock Domain 5 Clock Selection Bits

These bits indicate the set value of clock domain 5.

bit3:0	Description
0000	High-speed CR clock
0001	Low-speed CR clock
0010	Main clock
0011	Sub clock (Do not use this setting)
0100	PLL0 clock
0101	PLL1 clock
0110	PLL2 clock
0111	PLL3 clock
1000	SSCG PLL0 clock
1001	SSCG PLL1 clock
1010	SSCG PLL2 clock
1011	SSCG PLL3 clock
1100	Setting prohibited
1101	Setting prohibited
1110	Setting prohibited
1111	Clock fixed at "L"

5.14.4. STS Clock Source Enable Register 0(SYSC1_STSCKER0)

The SYSC1_STSCKER0register indicates the value for setting whether to enable/disable oscillation of the internal operating clock.

bit	31	30	29	28	27	26	25	24
Field	ENCLK LAPP1A	ENCLK LAPP0A	ENCLK LCP1A	ENCLK LCP0A	ENCLK LAPP1	ENCLK LAPP0	ENCLK LCP1	ENCLK LCP0
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	WPS							
Initial Value	1	1	1	1	1	1	1	1

bit	23	22	21	20	19	18	17	16
Field	ENCLK LCP	Reserved	ENCLK LLPBM2	ENCLK LLPBM	ENCLK HAPP1B1	ENCLK HAPP1B0	ENCLK HAPP0A1	ENCLK HAPP0A0
R/W Attribute	R,WX	R0,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	WPS							
Initial Value	1	0	1	1	1	1	1	1

bit	15	14	13	12	11	10	9	8
Field	Reserved	ENCLK SYSC1	Reserved		ENCLK EXTBUS	ENCLK MEMC	ENCLK DMA	ENCLK HPM
R/W Attribute	R0,WX	R,WX	R0,WX		R,WX	R,WX	R,WX	R,WX
Protection Attribute	WPS							
Initial Value	0	1	00		1	1	1	1

bit	7	6	5	4	3	2	1	0
Field	ENCLK HPM2	ENCLK TRC	ENCLK DBG	ENCLK ATB	Reserved			ENCLK CPU0
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R0,WX			R,WX
Protection Attribute	WPS							
Initial Value	1	1	1	1	000			1

[bit31] ENCLKLAPP1A: LAPP1A Clock Oscillation Enable Bit

This bit indicates the set value of the LAPP1A clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit30] ENCLKLAPP0A: LAPP0A Clock Oscillation Enable Bit

This bit indicates the set value of the LAPP0A clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit29] ENCLKLCP1A: LCP1A Clock Oscillation Enable Bit

This bit indicates the set value of the LCP1A clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit28] ENCLKLCP0A: LCP0A Clock Oscillation Enable Bit

This bit indicates the set value of the LCP0A clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit27] ENCLKLAPP1: LAPP1 Clock Oscillation Enable Bit

This bit indicates the set value of the LAPP1 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit26] ENCLKLAPP0: LAPP0 Clock Oscillation Enable Bit

This bit indicates the set value of the LAPP0 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit25] ENCLKLCP1: LCP1 Clock Oscillation Enable Bit

This bit indicates the set value of the LCP1 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit24] ENCLKLCP0: LCP0 Clock Oscillation Enable Bit

This bit indicates the set value of the LCP0 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit23] ENCLKLCP: LCP Clock Oscillation Enable Bit

This bit indicates the set value of the LCP clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit22] Reserved: Reserved Bit

[bit21] ENCLKLLPBM2: LLPBM2 Clock Oscillation Enable Bit

This bit indicates the set value of the LLPBM2 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit20] ENCLKLLPBM: LLPBM Clock Oscillation Enable Bit

This bit indicates the set value of the LLPBM clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit19] ENCLKHAPP1B1: HAPP1B1 Clock Oscillation Enable Bit

This bit indicates the set value of the HAPP1B1 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit18] ENCLKHAPP1B0: HAPP1B0 Clock Oscillation Enable Bit

This bit indicates the set value of the HAPP1B0 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit17] ENCLKHAPP0A1: HAPP0A1 Clock Oscillation Enable Bit

This bit indicates the set value of the HAPP0A1 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit16] ENCLKHAPP0A0: HAPP0A0 Clock Oscillation Enable Bit

This bit indicates the set value of the HAPP0A0 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit15] Reserved: Reserved Bit

[bit14] ENCLKSYSC1: ENCLKSYSC1 Clock Oscillation Enable Bit

This bit indicates the set value of the ENCLKSYSC1 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit13:12] Reserved: Reserved Bits

[bit11] ENCLKEXTBUS: ENCLKEXTBUS Clock Oscillation Enable Bit

This bit indicates the set value of the ENCLKEXTBUS clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit10] ENCLKMEMC: ENCLKMEMC Clock Oscillation Enable Bit

This bit indicates the set value of the ENCLKMEMC clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit9] ENCLKDMA: ENCLKDMA Clock Oscillation Enable Bit

This bit indicates the set value of the ENCLKDMA clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit8] ENCLKHPM: ENCLKHPM Clock Oscillation Enable Bit

This bit indicates the set value of the ENCLKHPM clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit7] ENCLKHPM2: ENCLKHPM2 Clock Oscillation Enable Bit

This bit indicates the set value of the ENCLKHPM2 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit6] ENCLKTRC: ENCLKTRC Clock Oscillation Enable Bit

This bit indicates the set value of the ENCLKTRC clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit5] ENCLKDBG: ENCLKDBG Clock Oscillation Enable Bit

This bit indicates the set value of the ENCLKDBG clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit4] ENCLKATB: ENCLKATB Clock Oscillation Enable Bit

This bit indicates the set value of the ENCLKATB clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit3:1] Reserved: Reserved Bits

[bit0] ENCLKCPU0: ENCLKCPU0 Clock Oscillation Enable Bit

This bit indicates the set value of the ENCLKCPU0 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

5.14.5. STS Clock Source Enable Register 1(SYSC1_STSCKER1)

The SYSC1_STSCKER1register indicates the value for setting whether to enable/disable oscillation of the internal operating clock.

bit	31	30	29	28	27	26	25	24
Field	Reserved			ENCLK CD3B1	ENCLK CD3B0	ENCLK CD3A1	ENCLK CD3A0	ENCLK CD3
R/W Attribute	R0,WX			R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	WPS							
Initial Value	000			1	1	1	1	1

bit	23	22	21	20	19	18	17	16
Field	Reserved			ENCLK CD2B1	ENCLK CD2B0	ENCLK CD2A1	ENCLK CD2A0	ENCLK CD2
R/W Attribute	R0,WX			R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	WPS							
Initial Value	000			1	1	1	1	1

bit	15	14	13	12	11	10	9	8
Field	Reserved			ENCLK CD1B1	ENCLK CD1B0	ENCLK CD1A1	ENCLK CD1A0	ENCLK CD1
R/W Attribute	R0,WX			R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	WPS							
Initial Value	000			1	1	1	1	1

bit	7	6	5	4	3	2	1	0
Field	Reserved							ENCLK HSSPI
R/W Attribute	R0,WX							R,WX
Protection Attribute	WPS							
Initial Value	0000000							1

[bit31:29] Reserved: Reserved Bits

[bit28] ENCLKCD3B1: CD3B1 Clock Oscillation Enable Bit

This bit indicates the set value of the CD3B1 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit27] ENCLKCD3B0: CD3B1 Clock Oscillation Enable Bit

This bit indicates the set value of the CD3B1 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit26] ENCLKCD3A1: CD3A1 Clock Oscillation Enable Bit

This bit indicates the set value of the CD3A1 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit25] ENCLKCD3A0: CD3A0 Clock Oscillation Enable Bit

This bit indicates the set value of the CD3A0 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit24] ENCLKCD3: CD3 Clock Oscillation Enable Bit

This bit indicates the set value of the CD3 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit23:21] Reserved: Reserved Bits

[bit20] ENCLKCD2B1: CD2B1 Clock Oscillation Enable Bit

This bit indicates the set value of the CD2B1 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit19] ENCLKCD2B0: CD2B0 Clock Oscillation Enable Bit

This bit indicates the set value of the CD2B0 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit18] ENCLKCD2A1: CD2A1 Clock Oscillation Enable Bit

This bit indicates the set value of the CD2A1 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit17] ENCLKCD2A0: CD2A0 Clock Oscillation Enable Bit

This bit indicates the set value of the CD2A0 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit16] ENCLKCD2: CD2 Clock Oscillation Enable Bit

This bit indicates the set value of the CD2 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit15:13] Reserved: Reserved Bits

[bit12] ENCLKCD1B1: CD1B1 Clock Oscillation Enable Bit

This bit indicates the set value of the CD1B1 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit11] ENCLKCD1B0: CD1B0 Clock Oscillation Enable Bit

This bit indicates the set value of the CD1B0 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit10] ENCLKCD1A1: CD1A1 Clock Oscillation Enable Bit

This bit indicates the set value of the CD1A1 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit9] ENCLKCD1A0: CD1A0 Clock Oscillation Enable Bit

This bit indicates the set value of the CD1A0 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit8] ENCLKCD1: CD1 Clock Oscillation Enable Bit

This bit indicates the set value of the CD1 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit7:1] Reserved: Reserved Bits

[bit0] ENCLKHSSPI: HSSPI Clock Oscillation Enable Bit

This bit indicates the set value of the HSSPI clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

5.14.6. STS Clock Source Enable Register 2(SYSC1_STSCKER2)

The SYSC1_STSCKER2register indicates the value for setting whether to enable/disable oscillation of the internal operating clock.

bit	31							16
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	WPS							
Initial Value	00000000_00000000							

bit	15	14	13	12	11	10	9	8
Field	Reserved			ENCLK CD5B1	ENCLK CD5B0	ENCLK CD5A1	ENCLK CD5A0	ENCLK CD5
R/W Attribute	R0,WX			R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	WPS							
Initial Value	000			1	1	1	1	1

bit	7	6	5	4	3	2	1	0
Field	Reserved			ENCLK CD4B1	ENCLK CD4B0	ENCLK CD4A1	ENCLK CD4A0	ENCLK CD4
R/W Attribute	R0,WX			R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	WPS							
Initial Value	000			1	1	1	1	1

[bit31:13] Reserved: Reserved Bits

[bit12] ENCLKCD5B1: CD5B1 Clock Oscillation Enable Bit

This bit indicates the set value of the CD5B1 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit11] ENCLKCD5B0: CD5B0 Clock Oscillation Enable Bit

This bit indicates the set value of the CD5B0 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit10] ENCLKCD5A1: CD5A1 Clock Oscillation Enable Bit

This bit indicates the set value of the CD5A1 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit9] ENCLKCD5A0: CD5A0 Clock Oscillation Enable Bit

This bit indicates the set value of the CD5A0 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit8] ENCLKCD5: CD5 Clock Oscillation Enable Bit

This bit indicates the set value of the CD5 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit7:5] Reserved: Reserved Bits

[bit4] ENCLKCD4B1: CD4B1 Clock Oscillation Enable Bit

This bit indicates the set value of the CD4B1 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit3] ENCLKCD4B0: CD4B0 Clock Oscillation Enable Bit

This bit indicates the set value of the CD4B0 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit2] ENCLKCD4A1: CD4A1 Clock Oscillation Enable Bit

This bit indicates the set value of the CD4A1 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit1] ENCLKCD4A0: CD4A0 Clock Oscillation Enable Bit

This bit indicates the set value of the CD4A0 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit0] ENCLKCD4: CD4 Clock Oscillation Enable Bit

This bit indicates the set value of the CD4 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

5.14.7. STS Clock Divider Register 0 (SYSC1_STSCKDIVR0)

The SYSC1_STSCKDIVR0 register indicates the set values of the division ratio of each internal operating clock.

bit	31	30	29	28	27	26	25	24
Field	Reserved					HPMDIV		
R/W Attribute	R0,WX					R,WX		
Protection Attribute	WPS							
Initial Value	00000					000		

bit	23	22	21	20	19	18	17	16
Field	Reserved			TRCDIV				
R/W Attribute	R0,WX			R,WX				
Protection Attribute	WPS							
Initial Value	000			00000				

bit	15	14	13	12	11	10	9	8
Field	Reserved		DBGDIV		Reserved		ATBDIV	
R/W Attribute	R0,WX		R,WX		R0,WX		R,WX	
Protection Attribute	WPS							
Initial Value	00		00		00		00	

bit	7	6	5	4	3	2	1	0
Field	Reserved			SYSDIV				
R/W Attribute	R0,WX			R,WX				
Protection Attribute	WPS							
Initial Value	000			00000				

[bit31:27] Reserved: Reserved Bits

[bit26:24] HPMDIV: HPM Clock Frequency Divider Setting Bits

These bits indicate the set value of the division ratio of the HPM clock, DMA clock, or MEMC clock from the SYS clock.

bit26:24	Description
000	No division
001	Divided by 2
010	Divided by 3
011	Divided by 4
100	Divided by 5
101	Divided by 6
110	Divided by 7
111	Divided by 8

[bit23:21] Reserved: Reserved Bits

[bit20:16] TRCDIV: TRC Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the TRC clock from the source clock (clock domain TRC).

bit20:16	Description
00000	No division
00001	Divided by 2
00010	Divided by 3
00011	Divided by 4
. . .	. . .
11101	Divided by 30
11110	Divided by 31
11111	Divided by 32

[bit15:14] Reserved: Reserved Bits

[bit13:12] DBGDIV: DBG Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the DBG clock from the ATB clock.

bit13:12	Description
00	No division
01	Divided by 2
10	Divided by 4
11	Divided by 8

[bit11:10] Reserved: Reserved Bits

[bit9:8] ATBDIV [1:0]: ATB Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the ATB clock from the SYS clock.

bit9:8	Description
00	No division
01	Divided by 2
10	Divided by 4
11	Divided by 8

[bit7:5] Reserved: Reserved Bits

[bit4:0] SYSDIV: SYS Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the CPU0 clock or CPERI clock from the source clock (clock domain 0).

bit4:0	Description
00000	No division
00001	Divided by 2
00010	Divided by 3
...	...
11101	Divided by 30
11110	Divided by 31
11111	Divided by 32

5.14.8. STS Clock Divider Register 1 (SYSC1_STSCKDIVR1)

The SYSC1_STSCKDIVR1 register indicates the set values of the division ratio of each internal operating clock.

bit	31	30	29	28	27	26	25	24
Field	HAPP1B1DIV				HAPP1B0DIV			
R/W Attribute	R,WX				R,WX			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	23	22	21	20	19	18	17	16
Field	HAPP0A1DIV				HAPP0A0DIV			
R/W Attribute	R,WX				R,WX			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	15	14	13	12	11	10	9	8
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	WPS							
Initial Value	00000000							

bit	7	6	5	4	3	2	1	0
Field	SYSC1DIV				Reserved	EXTBUSDIV		
R/W Attribute	R,WX				R0,WX	R,WX		
Protection Attribute	WPS							
Initial Value	0000				0	000		

[bit31:28] HAPP1B1DIV: HAPP1B1 Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the HAPP1B1 clock from the HAPP1B0 clock.

bit31:28	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
0011	Divided by 4
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

[bit27:24] HAPP1B0DIV: HAPP1B0 Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the HAPP1B0 clock from the HPM clock.

bit27:24	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
0011	Divided by 4
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

[bit23:20] HAPP0A1DIV: HAPP0A1 Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the HAPP0A1 clock from the HAPP0A0 clock.

bit23:20	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
0011	Divided by 4
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

[bit19:16] HAPP0A0DIV: HAPP0A0 Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the HAPP0A0 clock from the HPM clock.

bit19:16	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
0011	Divided by 4
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

[bit15:8] Reserved: Reserved Bits

[bit7:4] SYSC1DIV: SYSC1 Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the SYSC1 clock from the HPM clock.

bit7:4	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
0011	Divided by 4
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

[bit3] Reserved: Reserved Bit

[bit2:0] EXTBUSDIV: EXTBUS Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the EXTBUS clock from the HPM clock.

bit2:0	Description
000	No division
001	Divided by 2
010	Divided by 4
011	Divided by 8
100	Divided by 16
101	Divided by 32
110	Divided by 64
111	Divided by 128

5.14.9. STS Clock Divider Register 2 (SYSC1_STSCKDIVR2)

The SYSC1_STSCKDIVR2 register indicates the set values of the division ratio of each internal operating clock.

bit	31	30	29	28	27	26	25	24
Field	Reserved				LAPP1DIV			
R/W Attribute	R0,WX				R,WX			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	23	22	21	20	19	18	17	16
Field	Reserved				LAPP0DIV			
R/W Attribute	R0,WX				R,WX			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	15	14	13	12	11	10	9	8
Field	Reserved				LCP1DIV			
R/W Attribute	R0,WX				R,WX			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	7	6	5	4	3	2	1	0
Field	LCP0DIV				Reserved		LCPDIV	
R/W Attribute	R,WX				R0,WX		R,WX	
Protection Attribute	WPS							
Initial Value	0000				00		00	

[bit31:28] Reserved: Reserved Bits

[bit27:24] LAPP1DIV: LAPP1 Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the LAPP1 clock from the CPU clock.

bit27:24	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
0011	Divided by 4
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

[bit23:20] Reserved: Reserved Bits

[bit19:16] LAPP0DIV: LAPP0 Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the LAPP0 clock from the CPU clock.

bit19:16	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
0011	Divided by 4
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

[bit15:12] Reserved: Reserved Bits

[bit11:8] LCP1DIV: LCP1 Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the LCP1 clock from the CPU clock.

bit11:8	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
0011	Divided by 4
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

[bit7:4] LCP0DIV: LCP0 Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the LCP0 clock from the CPU clock.

bit7:4	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
0011	Divided by 4
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

[bit3:2] Reserved: Reserved Bits

[bit1:0] LCPDIV: LCP Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the LCP clock from the CPU clock.

bit1:0	Description
00	No division
01	Divided by 2
10	Divided by 4
11	Divided by 8

5.14.10. STS Clock Divider Register 3 (SYSC1_STSCKDIVR3)

The SYSC1_STSCKDIVR3 register indicates the set values of the division ratio of each internal operating clock.

bit	31	30	29	28	27	26	25	24
Field	Reserved			LAPP1ADIV				
R/W Attribute	R0,WX			R,WX				
Protection Attribute	WPS							
Initial Value	000			00000				

bit	23	22	21	20	19	18	17	16
Field	Reserved			LAPP0ADIV				
R/W Attribute	R0,WX			R,WX				
Protection Attribute	WPS							
Initial Value	000			00000				

bit	15	14	13	12	11	10	9	8
Field	Reserved			LCP1ADIV				
R/W Attribute	R0,WX			R,WX				
Protection Attribute	WPS							
Initial Value	000			00000				

bit	7	6	5	4	3	2	1	0
Field	Reserved			LCP0ADIV				
R/W Attribute	R0,WX			R,WX				
Protection Attribute	WPS							
Initial Value	000			00000				

[bit31:29] Reserved: Reserved Bits

[bit28:24] LAPP1ADIV: LAPP1A Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the LAPP1A clock from the CPU/PLL0 clock.

bit28:24	Description
00000	No division
00001	Divided by 2
00010	Divided by 3
00011	Divided by 4
...	...
11101	Divided by 30
11110	Divided by 31
11111	Divided by 32

[bit23:21] Reserved: Reserved Bits

[bit20:16] LAPP0ADIV: LAPP0A Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the LAPP0A clock from the CPU/PLL0 clock.

bit20:16	Description
00000	No division
00001	Divided by 2
00010	Divided by 3
00011	Divided by 4
...	...
11101	Divided by 30
11110	Divided by 31
11111	Divided by 32

[bit15:13] Reserved: Reserved Bits

[bit12:8] LCP1ADIV: LCP1A Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the LCP1A clock from the CPU/PLL0 clock.

bit12:8	Description
00000	No division
00001	Divided by 2
00010	Divided by 3
00011	Divided by 4
...	...
11101	Divided by 30
11110	Divided by 31
11111	Divided by 32

[bit7:5] Reserved: Reserved Bits

[bit4:0] LCP0ADIV: LCP0A Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the LCP0A clock from the CPU/PLL0 clock.

bit4:0	Description
00000	No division
00001	Divided by 2
00010	Divided by 3
00011	Divided by 4
...	...
11101	Divided by 30
11110	Divided by 31
11111	Divided by 32

5.14.11. STS Clock Divider Register 4 (SYSC1_STSCKDIVR4)

The SYSC1_STSCKDIVR4 register indicates the set values of the division ratio of each internal operating clock.

bit	31	8
Field	Reserved	
R/W Attribute	R0,WX	
Protection Attribute	WPS	
Initial Value	00000000_00000000_00000000	

bit	7	6	5	4	3	2	1	0
Field	Reserved			HSSPIDIV				
R/W Attribute	R0,WX			R,WX				
Protection Attribute	WPS							
Initial Value	000			00000				

[bit31:5] Reserved: Reserved Bits

[bit4:0] HSSPIDIV: HSSPI Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the HSSPI clock from the source clock (HSSPI clock domain).

bit4:0	Description
00000	No division
00001	Divided by 2
00010	Divided by 3
...	...
11101	Divided by 30
11110	Divided by 31
11111	Divided by 32

5.14.12. STS Clock Divider Register 5 (SYSC1_STSCKDIVR5)

The SYSC1_STSCKDIVR5 register indicates the set values of the division ratio of each internal operating clock.

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	WPS							
Initial Value	00000000							

bit	23	22	21	20	19	18	17	16
Field	CD1B1DIV				CD1B0DIV			
R/W Attribute	R,WX				R,WX			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	15	14	13	12	11	10	9	8
Field	CD1A1DIV				CD1A0DIV			
R/W Attribute	R,WX				R,WX			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	7	6	5	4	3	2	1	0
Field	Reserved			CD1DIV				
R/W Attribute	R0,WX			R,WX				
Protection Attribute	WPS							
Initial Value	000			00000				

[bit31:24] Reserved: Reserved Bits

[bit23:20] CD1B1DIV: CD1B1 Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the CD1B1 clock from the CD1B0 clock.

bit23:20	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

[bit19:16] CD1B0DIV: CD1B0 Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the CD1B0 clock from the CD1 clock.

bit19:16	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

[bit15:12] CD1A1DIV: CD1A1 Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the CD1A1 clock from the CD1A0 clock.

bit15:12	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

[bit11:8] CD1A0DIV: CD1A0 Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the CD1A0 clock from the CD1 clock.

bit11:8	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

[bit7:5] Reserved: Reserved Bits

[bit4:0] CD1DIV: CD1 Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the CD1 clock from the source clock (clock domain 1).

bit4:0	Description
00000	No division
00001	Divided by 2
00010	Divided by 3
...	...
11101	Divided by 30
11110	Divided by 31
11111	Divided by 32

5.14.13. STS Clock Divider Register 6 (SYSC1_STSCKDIVR6)

The SYSC1_STSCKDIVR6 register indicates the set values of the division ratio of each internal operating clock.

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	WPS							
Initial Value	00000000							

bit	23	22	21	20	19	18	17	16
Field	CD2B1DIV				CD2B0DIV			
R/W Attribute	R,WX				R,WX			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	15	14	13	12	11	10	9	8
Field	CD2A1DIV				CD2A0DIV			
R/W Attribute	R,WX				R,WX			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	7	6	5	4	3	2	1	0
Field	Reserved			CD2DIV				
R/W Attribute	R0,WX			R,WX				
Protection Attribute	WPS							
Initial Value	000			00000				

[bit31:24] Reserved: Reserved Bits

[bit23:20] CD2B1DIV: CD2B1 Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the CD2B1 clock from the CD2B0 clock.

bit23:20	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

[bit19:16] CD2B0DIV: CD2B0 Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the CD2B0 clock from the CD2 clock.

bit19:16	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

[bit15:12] CD2A1DIV: CD2A1 Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the CD2A1 clock from the CD2A0 clock.

bit15:12	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

[bit11:8] CD2A0DIV: CD2A0 Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the CD2A0 clock from the CD2 clock.

bit11:8	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

[bit7:5] Reserved: Reserved Bits

[bit4:0] CD2DIV: CD2 Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the CD2 clock from the source clock (clock domain 2).

bit4:0	Description
00000	No division
00001	Divided by 2
00010	Divided by 3
...	...
11101	Divided by 30
11110	Divided by 31
11111	Divided by 32

5.14.14. STS Clock Divider Register 7 (SYSC1_STSCKDIVR7)

The SYSC1_STSCKDIVR7 register indicates the set values of the division ratio of each internal operating clock.

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	WPS							
Initial Value	00000000							

bit	23	22	21	20	19	18	17	16
Field	CD3B1DIV				CD3B0DIV			
R/W Attribute	R,WX				R,WX			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	15	14	13	12	11	10	9	8
Field	CD3A1DIV				CD3A0DIV			
R/W Attribute	R,WX				R,WX			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	7	6	5	4	3	2	1	0
Field	Reserved			CD3DIV				
R/W Attribute	R0,WX			R,WX				
Protection Attribute	WPS							
Initial Value	000			00000				

[bit31:24] Reserved: Reserved Bits

[bit23:20] CD3B1DIV: CD3B1 Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the CD3B1 clock from the CD3B0 clock.

bit23:20	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

[bit19:16] CD3B0DIV: CD3B0 Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the CD3B0 clock from the CD3 clock.

bit19:16	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

[bit15:12] CD3A1DIV: CD3A1 Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the CD3A1 clock from the CD3A0 clock.

bit15:12	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

[bit11:8] CD3A0DIV: CD3A0 Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the CD3A0 clock from the CD3 clock.

bit11:8	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

[bit7:5] Reserved: Reserved Bits

[bit4:0] CD3DIV: CD3 Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the CD3 clock from the source clock (clock domain 3).

bit4:0	Description
00000	No division
00001	Divided by 2
00010	Divided by 3
...	...
11101	Divided by 30
11110	Divided by 31
11111	Divided by 32

5.14.15. STS Clock Divider Register 8 (SYSC1_STSCKDIVR8)

The SYSC1_STSCKDIVR8 register indicates the set values of the division ratio of each internal operating clock.

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	WPS							
Initial Value	00000000							

bit	23	22	21	20	19	18	17	16
Field	CD4B1DIV				CD4B0DIV			
R/W Attribute	R,WX				R,WX			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	15	14	13	12	11	10	9	8
Field	CD4A1DIV				CD4A0DIV			
R/W Attribute	R,WX				R,WX			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	7	6	5	4	3	2	1	0
Field	Reserved			CD4DIV				
R/W Attribute	R0,WX			R,WX				
Protection Attribute	WPS							
Initial Value	000			00000				

[bit31:24] Reserved: Reserved Bits

[bit23:20] CD4B1DIV: CD4B1 Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the CD4B1 clock from the CD4B0 clock.

bit23:20	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

[bit19:16] CD4B0DIV: CD4B0 Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the CD4B0 clock from the CD4 clock.

bit19:16	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

[bit15:12] CD4A1DIV: CD4A1 Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the CD4A1 clock from the CD4A0 clock.

bit15:12	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

[bit11:8] CD4A0DIV: CD4A0 Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the CD4A0 clock from the CD4 clock.

bit11:8	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

[bit7:5] Reserved: Reserved Bits

[bit4:0] CD4DIV: CD4 Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the CD4 clock from the source clock (clock domain 4).

bit4:0	Description
00000	No division
00001	Divided by 2
00010	Divided by 3
...	...
11101	Divided by 30
11110	Divided by 31
11111	Divided by 32

5.14.16. STS Clock Divider Register 9 (SYSC1_STSCKDIVR9)

The SYSC1_STSCKDIVR9 register indicates the set values of the division ratio of each internal operating clock.

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	WPS							
Initial Value	00000000							

bit	23	22	21	20	19	18	17	16
Field	CD5B1DIV				CD5B0DIV			
R/W Attribute	R,WX				R,WX			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	15	14	13	12	11	10	9	8
Field	CD5A1DIV				CD5A0DIV			
R/W Attribute	R,WX				R,WX			
Protection Attribute	WPS							
Initial Value	0000				0000			

bit	7	6	5	4	3	2	1	0
Field	Reserved			CD5DIV				
R/W Attribute	R0,WX			R,WX				
Protection Attribute	WPS							
Initial Value	000			00000				

[bit31:24] Reserved : Reserved Bits

[bit23:20] CD5B1DIV : CD5B1 Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the CD5B1 clock of the CD5B0.

bit23:20	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

[bit19:16] CD5B0DIV：CD5B0 Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the CD5B0 clock of the CD5.

bit19:16	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

[bit15:12] CD5A1DIV：CD5A1 Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the CD5A1 clock of the CD5A0.

bit15:12	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

[bit11:8] CD5A0DIV：CD5A0 Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the CD5A0 clock of the CD5.

bit11:8	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

[bit7:5] Reserved：Reserved Bits

[bit4:0] CD5DIV：CD5 Clock Divider Setting Bits

These bits indicate the set value of the division ratio of the CD5 clock of the source clock (clock domain 5).

bit4:0	Description
00000	No division
00001	Divided by 2
00010	Divided by 3
...	...
11101	Divided by 30
11110	Divided by 31
11111	Divided by 32

6. Other

This section describes the precautions on use of low-power consumption.

6.1. Restrictions on Transitioning to the PSS

This section describes restrictions on transitioning to the PSS.

1. When transitioning to the PSS, create a 1:1:1:1 relationship among the CPU clock (CLK_CPU) , the Memory Configuration clock (CLK_MEMC) , the SCU clock (Fast CR clock) , and the MCUCH clock (CLK_SYSC0H) in PSS profile. The loss of this relationship among these clocks may cause abnormal operation.
2. When using in the following condition, Fast CR clock must be enabled.
   - Use PSS mode without PD2 shutdown
   - Use peripherals in PD6 at PSS mode shutdown

   Fast CR clock is enabled when it sets SYSC0_PSSCKSRER. CROSCEN=1.

6.2. Limitation when Changing the Setting of Regulator

Following shows the limitation when changing the setting of regulator.

Output voltage and regulator mode can not be changed while RUN mode.
PSS is only changeable, but mode and output voltage can not be changed simultaneously.

6.3. Restriction to use “Clock fixed at L” and changing clock division ratio

Do not execute a profile update which changes the division ratio when the source clock of belonging clock domain is selected to "Clock fixed at "L"". This restriction must be kept not only RUN to RUN profile update but also RUN to PSS and PSS to RUN.

Example:

NG

[PSS] Clock fixed at "L" divide 32	➔	[RUN] PLL0 clock divide 2

OK

[PSS] Clock fixed at "L" divide 2	➔	[RUN] PLL0 clock divide 2

NG

[RUN] Clock fixed at "L" divide 16	➔	[RUN] PLL0 clock divide 4

OK

[RUN] Clock fixed at "L" divide 16	➔	[RUN] PLL0 clock divide 16	➔	[RUN] PLL0 clock divide 4

"➔" means a profile update.

CHAPTER 7: Low-voltage Detection

This chapter explains the low-voltage detection.

1. **Overview**
2. **Configuration and Block Diagrams**
3. **Explanation of Operation**
4. **Setting Procedure Examples**
5. **Operation Examples**
6. **Precautions for Using This Device**

LVD-TXXPT03P01R01L06-E1-XX

1. Overview

This section provides an overview of the low-voltage detection.

1.1. Features

Internal Low-Voltage Detection

- Function, detection voltage:
  Please see (Product specification).
- Operation: Always active.
  The RAM contents after a power-on reset cannot be guaranteed.

Core Power Supply Low-Voltage Detection

The Core power supply low-voltage detection circuit has two channels.

- Function, detection voltage:
  Please see (Product specification).
- Operation: Always active.
  The RAM contents after a Core power supply low-voltage detection reset cannot be guaranteed.

3.3/5.0 V Power Supply Low-Voltage Detection

The 3.3/5.0 V power supply low-voltage detection circuit has two channels.

- Function: A 3.3/5.0 V power supply low-voltage detection reset or interrupt is generated upon the detection of a voltage within plus or minus the detection error value in volts from the preset detection voltage.
- Operation: Operation can be switched between the active and inactive states by user settings.
  The selection to generate an interrupt or reset is also available.
  Furthermore, the settings for RUN/PSS can be switched using a profile.
  In the case where a reset is generated, the reset is issued after all clocks are stopped in order to guarantee the RAM contents after an external low-voltage detection reset.
  For details, see the chapter "RESET."

The following registers are defined below as "LVD registers."

Table 7-1 LVD Registers

RUN low-voltage detection setting register (SYSC0_RUNLVDCFGR)
PSS low-voltage detection setting register (SYSC0_PSSLVDCFGR)
APP low-voltage detection setting register (SYSC0_APPLVDCFGR)
STS low-voltage detection setting register (SYSC0_STSLVDCFGR)

The following bits are defined below as "LVD operation select bits."

Table 7-2 LVD Operation Select Bits

LVDL1S	Core power supply low-voltage detection operation select bit ch1
LVDL2S	Core power supply low-voltage detection operation select bit ch2
LVDH1S	3.3/5.0 V power supply low-voltage detection operation select bit ch1
LVDH2S	3.3/5.0 V power supply low-voltage detection operation select bit ch2

The following bits are defined below as "LVD ready bits."

Table 7-3 LVD Ready Bits

LVDL1R	Core power supply low-voltage detection ready bit ch1
LVDL2R	Core power supply low-voltage detection ready bit ch2
LVDH1R	3.3/5.0 V power supply low-voltage detection ready bit ch1
LVDH2R	3.3/5.0 V power supply low-voltage detection ready bit ch2

The following bits are defined below as "LVD reset source bits."

Table 7-4 LVD Reset Source Bits

LVDL1E	Core power supply low-voltage detection operation enable bit ch1
LVDL2E	Core power supply low-voltage detection operation enable bit ch2
LVDH1E	3.3/5.0 V power supply low-voltage detection operation enable bit ch1
LVDH2E	3.3/5.0 V power supply low-voltage detection operation enable bit ch2

Notes:

- *In the TRAVEO™ T1G Family Hardware Manual Platform Part series, functions are mapped as follows:*
  Core power supply low-voltage detection ch1: 1.2V internal regulator voltage monitoring
  Core power supply low-voltage detection ch2: 1.2V external power supply monitoring
  3.3/5.0 V power supply external low-voltage detection ch1: 5.0 V external power supply monitoring
  3.3/5.0 V power supply external low-voltage detection ch2: 3.3 V external power supply monitoring
- *In the TRAVEO™ T1G Family Hardware Manual Platform Part series, a low-voltage detection interrupt corresponds to an NMI.*
- *The values for the LVD operation select bits vary depending on the chip.*
- *For example, if a register value is fixed, only the functions indicated by the register value can be used.*
- *Also, since operation in the initial state varies depending on the initial value, see explanation of registers.*

2. Configuration and Block Diagrams

This section explains block diagrams of the low-voltage detection.

Figure 2-1, Figure 2-2, and Figure 2-3 are configuration diagrams of the low-voltage detection.

Configuration Diagrams of the Low-Voltage Detection

Figure 2-1 Overall Diagram

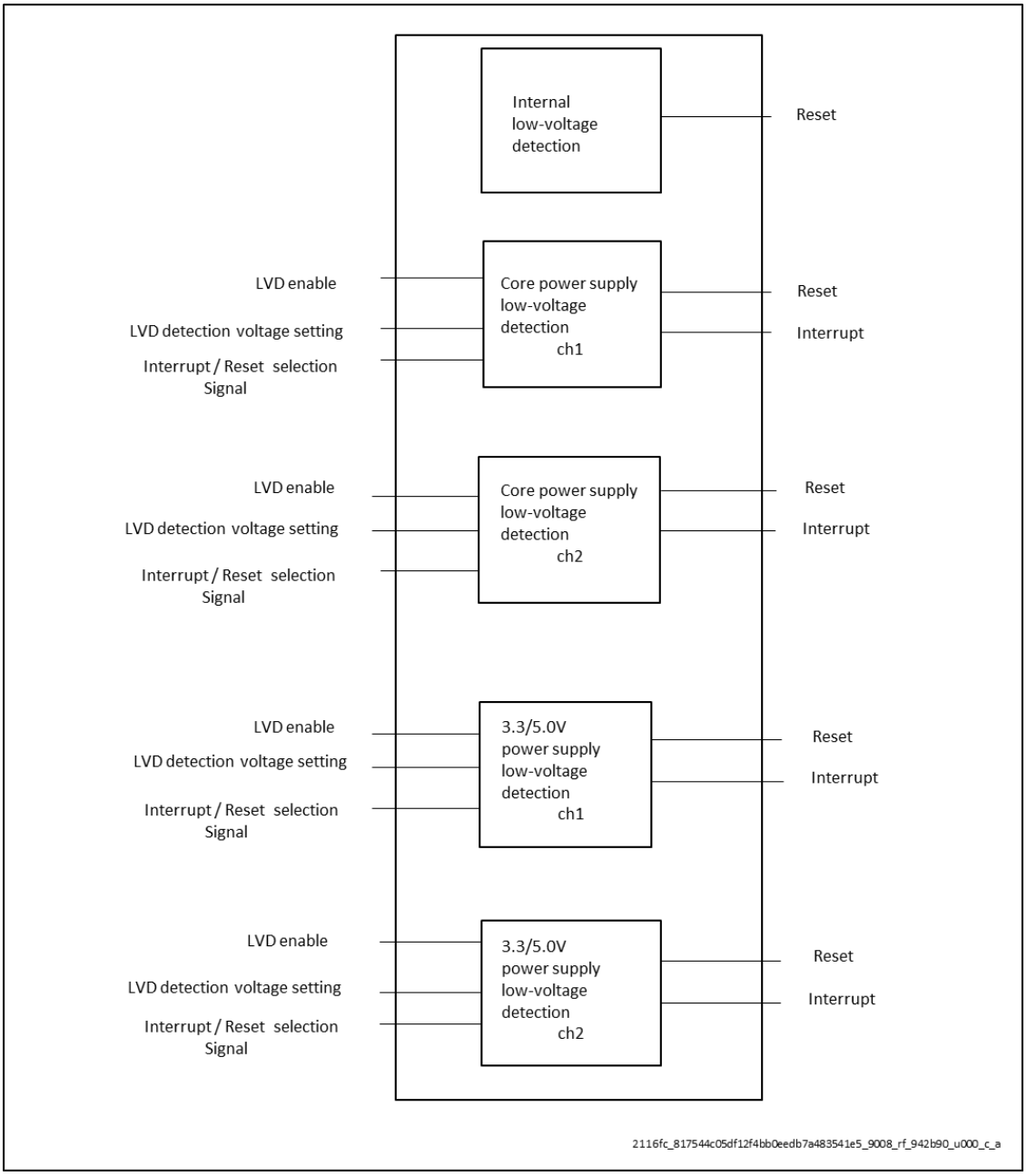


Figure 2-2 Internal Low-Voltage Detection

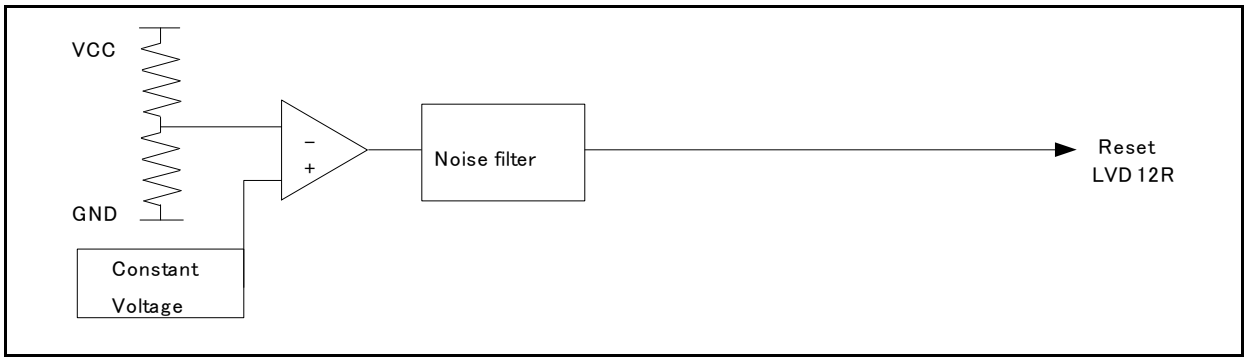
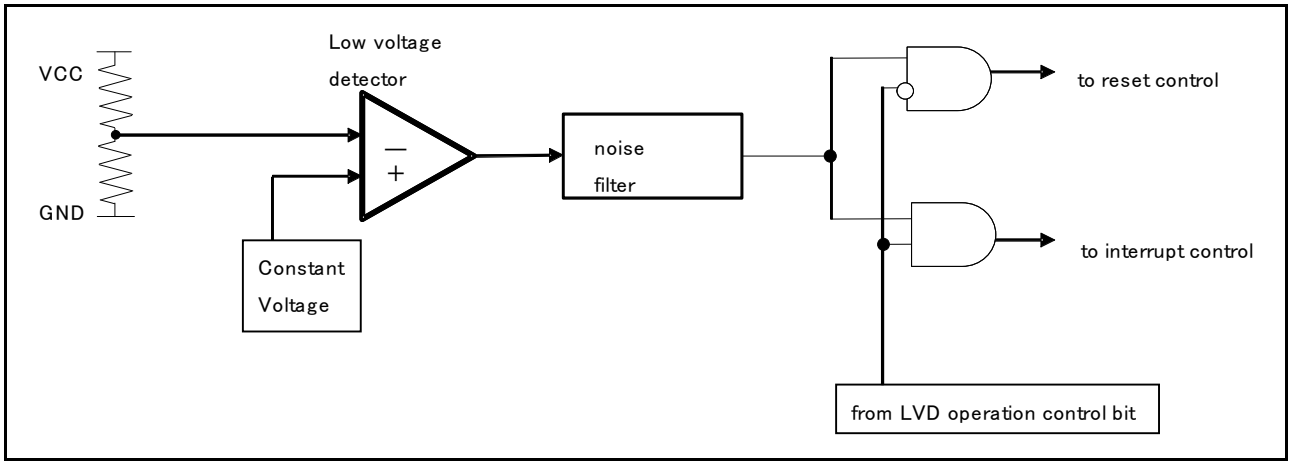


Figure 2-3 Core Power Supply Low-Voltage Detection, 3.3/5.0 V Power Supply Low-Voltage Detection



3. Explanation of Operation

This section explains the operation of the low-voltage detection.

Internal Low-Voltage Detection Circuit

The internal low-voltage detection circuit enters the monitoring state after power-on and remains in this state. It detects a low voltage when the internal supply voltage drops below the detection voltage. Upon detecting a low voltage, it generates a reset. The low-voltage detection is released when the internal supply voltage becomes higher than the release voltage.

Explanation of the Internal Low-Voltage Detection Operation

The low-voltage detection voltage is fixed. The circuit enters the monitoring state after power-on and remains in this state.

Internal Low-Voltage Detection Reset

An internal low-voltage reset is generated when the internal supply voltage drops below the set voltage. For details, see the chapter "RESET."

Core Power Supply Low-Voltage Detection Circuit, 3.3/5.0 V Power Supply Low-Voltage Detection Circuit

The 3.3/5.0 V power supply low-voltage detection circuit operates in the same way as the core power supply low-voltage detection circuit.

The low-voltage detection circuit enters the monitoring state after power-on. It detects a low voltage when the monitored supply voltage drops below the detection voltage. Either an interrupt or reset can be selected to be generated upon the detection of a low voltage. The low-voltage detection is released when the monitored supply voltage becomes higher than the release voltage.

Explanation of Operation

The available selections are whether to enable or disable the low-voltage detection, the low-voltage detection voltage, and whether to generate a reset or interrupt when a low voltage is detected. Executing the RUN/PSS profile changes the settings. After the execution of the profile, the low-voltage detection status register reflects the settings. After power-on or a change in its settings, the low-voltage detection circuit is given a stabilization wait time (about 260 us). During the stabilization wait time, the LVD ready bit becomes "0" and the detection result is masked. After the stabilization wait time has elapsed, the bit becomes "1" and supply voltage monitoring begins.

Interrupt Operation

If an interrupt has been selected to be generated upon the detection of a low voltage, the LVD interrupt source bit becomes "1" when the external supply voltage drops below the set voltage. At this point, an interrupt is generated.

Clearing an Interrupt Factor

To clear the low-voltage detection interrupt factor, write "1" to the LVD interrupt source clear bit. This clears the low-voltage detection interrupt factor. Note that if "1" is written to the LVD interrupt source

clear bit while the supply voltage is still below the preset detection voltage, the low-voltage detection interrupt factor is not cleared, so the interrupt request is retained.

Reset Operation

If a reset has been selected to be generated upon the detection of a low voltage, the LVD reset source bit becomes "1" and a reset is generated when the external supply voltage drops below the set voltage. For details, see the chapter "RESET."

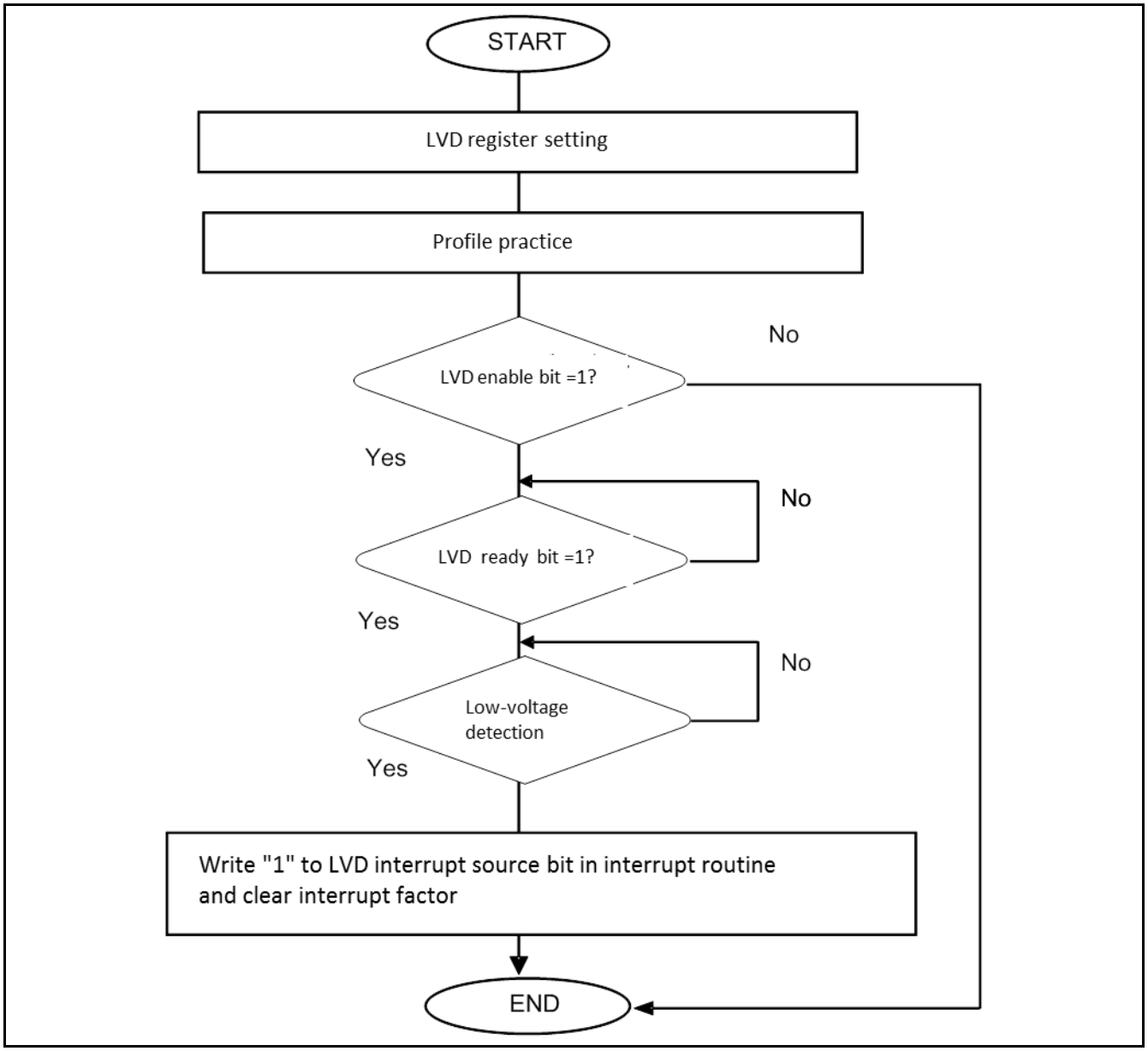
Reset will occur when LVD operation select bit is changed from "interrupt generation" to "reset generation" while interrupt is generated by low voltage detection circuit due to external supply voltage being lower than detection voltage.

4. Setting Procedure Examples

This section explains setting procedure examples of the low-voltage detection.

Setting the Low-Voltage Detection for Interrupts

Figure 4-1 Setting the Low-Voltage Detection for Interrupts

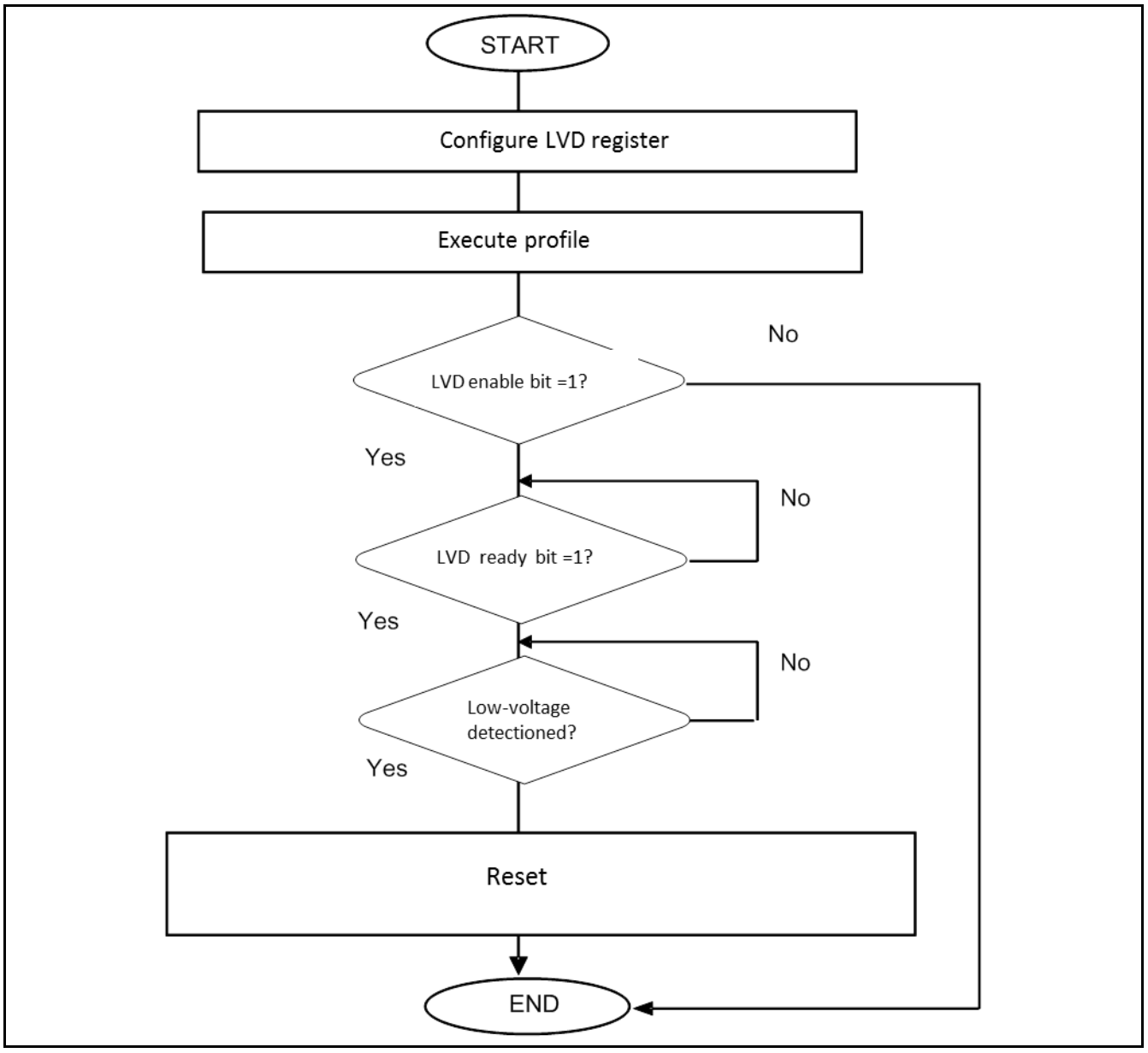


The execution of a profile applies the low-voltage detection settings made for the profile register. Low-voltage monitoring begins when the LVD ready bit becomes "1". Then, an interrupt is generated upon the detection of a low voltage.

To return from the interrupt, clear the interrupt factor by writing "1" to the LVD interrupt source clear bit at the end of the interrupt routine.

Setting the Low-Voltage Detection for Resets

Figure 4-2 Setting the Low-Voltage Detection for Resets



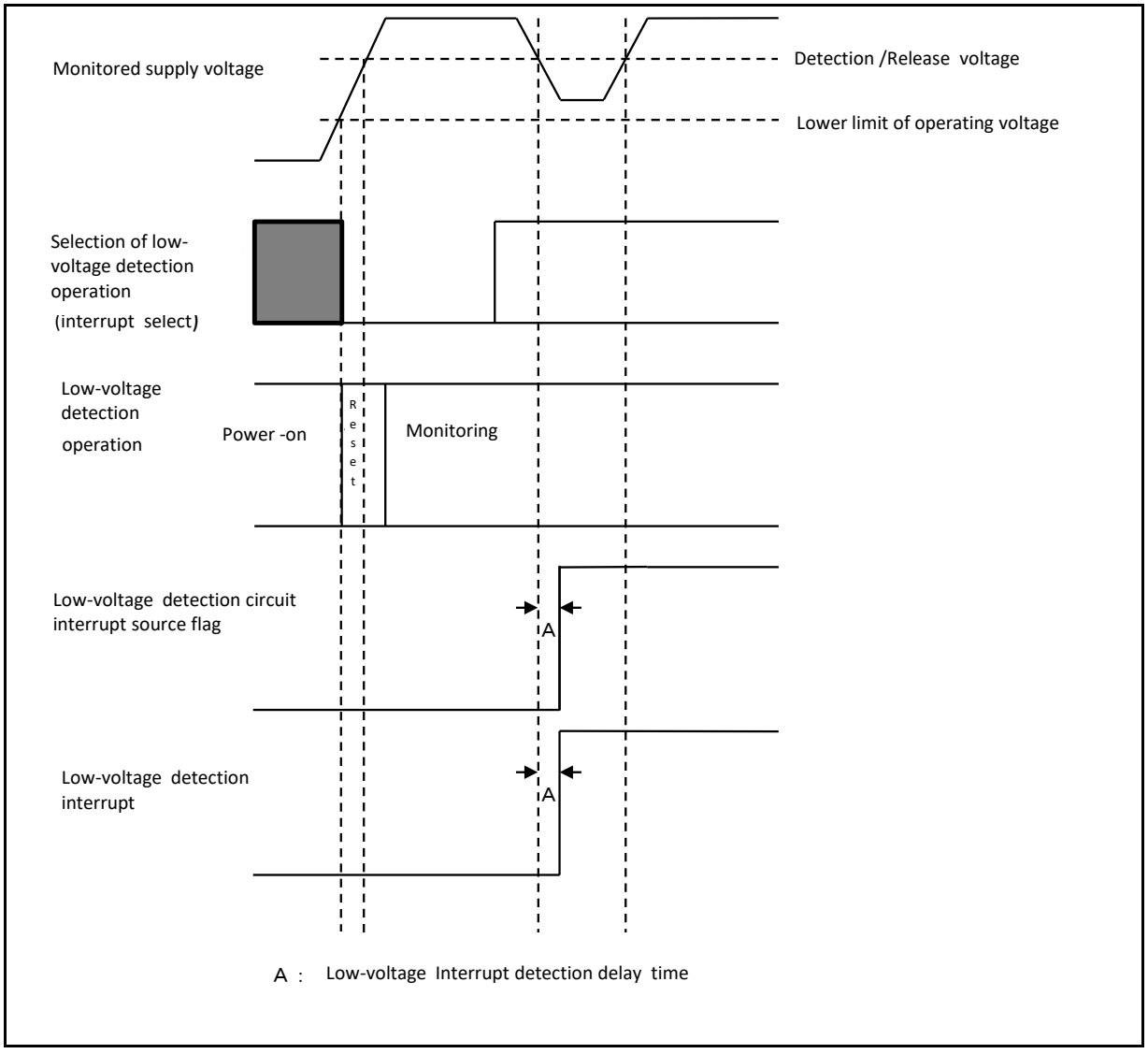
The execution of a profile applies the low-voltage detection settings made for the profile register. Low-voltage monitoring begins when the LVD ready bit becomes "1". Then, a reset is generated upon the detection of a low voltage.

5. Operation Examples

This section explains operation examples of the low-voltage detection.

If low-voltage detection interrupts are enabled, the detection flag is set and an interrupt is generated upon the detection of a low voltage, as shown in the following timing chart.

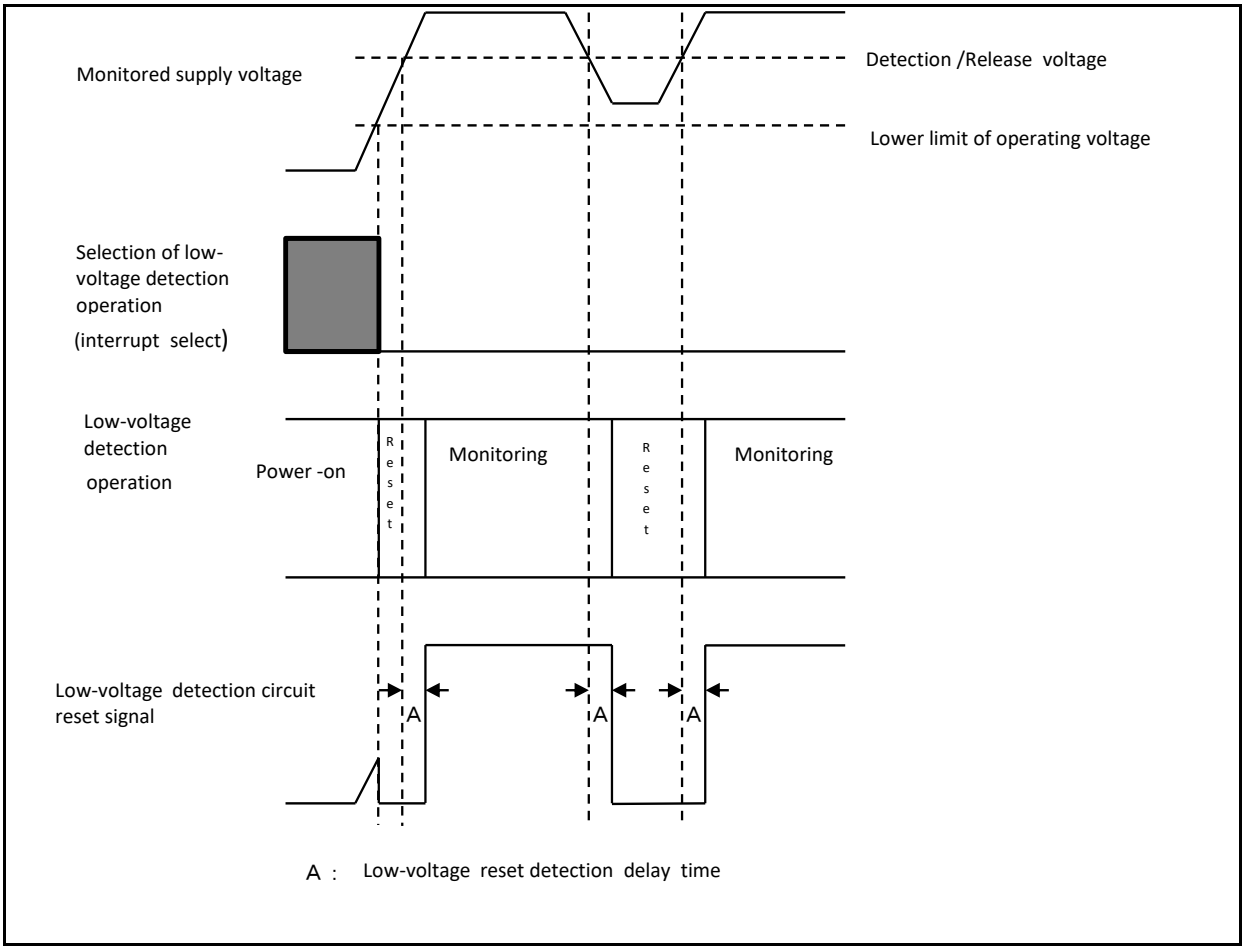
Figure 5-1 Timing Chart When Low-Voltage Detection Interrupts Are Enabled



When the detection circuit for resets is enabled

If the detection circuit for resets is enabled, a reset is generated upon the detection of a low voltage, as shown in the following timing chart.

Figure 5-2 Timing Chart When the Detection Circuit for Resets Is Enabled

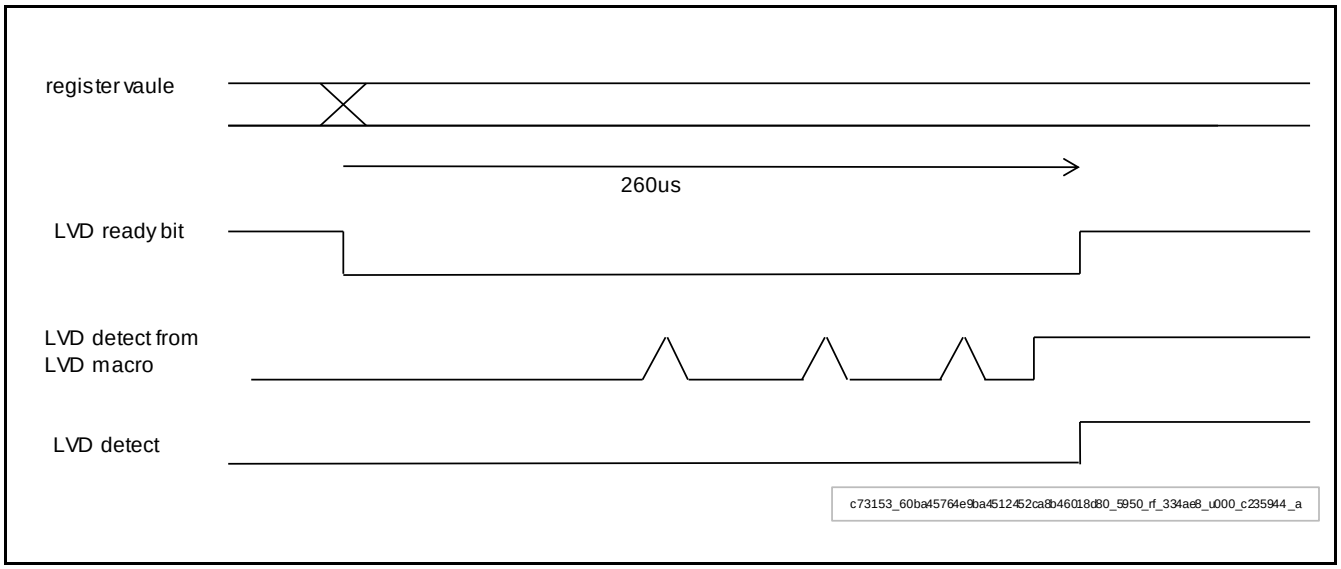


When the register setting value is changed

Set the LVD enable bit to 0 before changing the detection voltage.

If the register setting value has changed, the detection signal of the LVD macro is masked until the LVD ready bit becomes "1", as shown below.

Figure 5-3 Timing Chart When the Register Setting Value Is Changed



The change in the register value described here refers to the following bits.

LVDL1E	Core power supply low-voltage detection operation enable bit ch1
LVDL2E	Core power supply low-voltage detection operation enable bit ch2
LVDH1E :	3.3V/5.0V power supply low-voltage detection operation enable bit ch1
LVDH2E :	3.3V/5.0V power supply low-voltage detection operation enable bit ch2

6. Precautions for Using This Device

This section explains precautions on use of the low-voltage detection.

Operation during the PSS

- False detection during the PSS may occur in the following cases.
  - *Do not change the settings when CR is stopped in the PSS.*

- If a low-voltage detection reset is selected, a hard reset is generated when a low voltage is detected in the PSS. If an interrupt is selected, the PSS is released, and the system returns from the interrupt. For details on the hard reset, see the chapter on resets.

Masking of a Detection Result

- Following any low-voltage setting change, the detection result is masked for about 260 us.

CHAPTER 8: Clock Supervisor

This chapter shows the functions and operations of the clock supervisor.

1. **Overview**
2. **Configuration**
3. **Explanation of Operation**
4. **Setting Procedure Example**
5. **Registers**
6. **Precautions for Using This Device**

1. Overview

This device is equipped with several instances of two different types of clock supervisors. The used type of clock supervisor depends on the clock signal which is to be monitored.

1. Detector for stopped clock or abnormal frequency of the monitored clock.

This type of clock supervisor detects two possible errors in clock operation: a stopped clock or a clock operating in an abnormal frequency range. There are clock counters for both the monitored clock and the reference clock. Parameters for each counter define the frequency of the reference clock as well as the upper and lower limit for the frequency of the monitored clock. In case the dedicated frequency range comparator detects a stopped clock or a clock outside the frequency range, an abnormal state is signalled. Then, it depends on register settings and on the way, the monitored clock is used on the device, whether a reset or an interrupt is generated.

2. Detector for stopped clock only.

This type of clock supervisor detects a stopped clock operation for the monitored clock. When the rising edge of the monitored clock does not occur, the clock supervisor signals an abnormal state. In this case, a reset is generated.

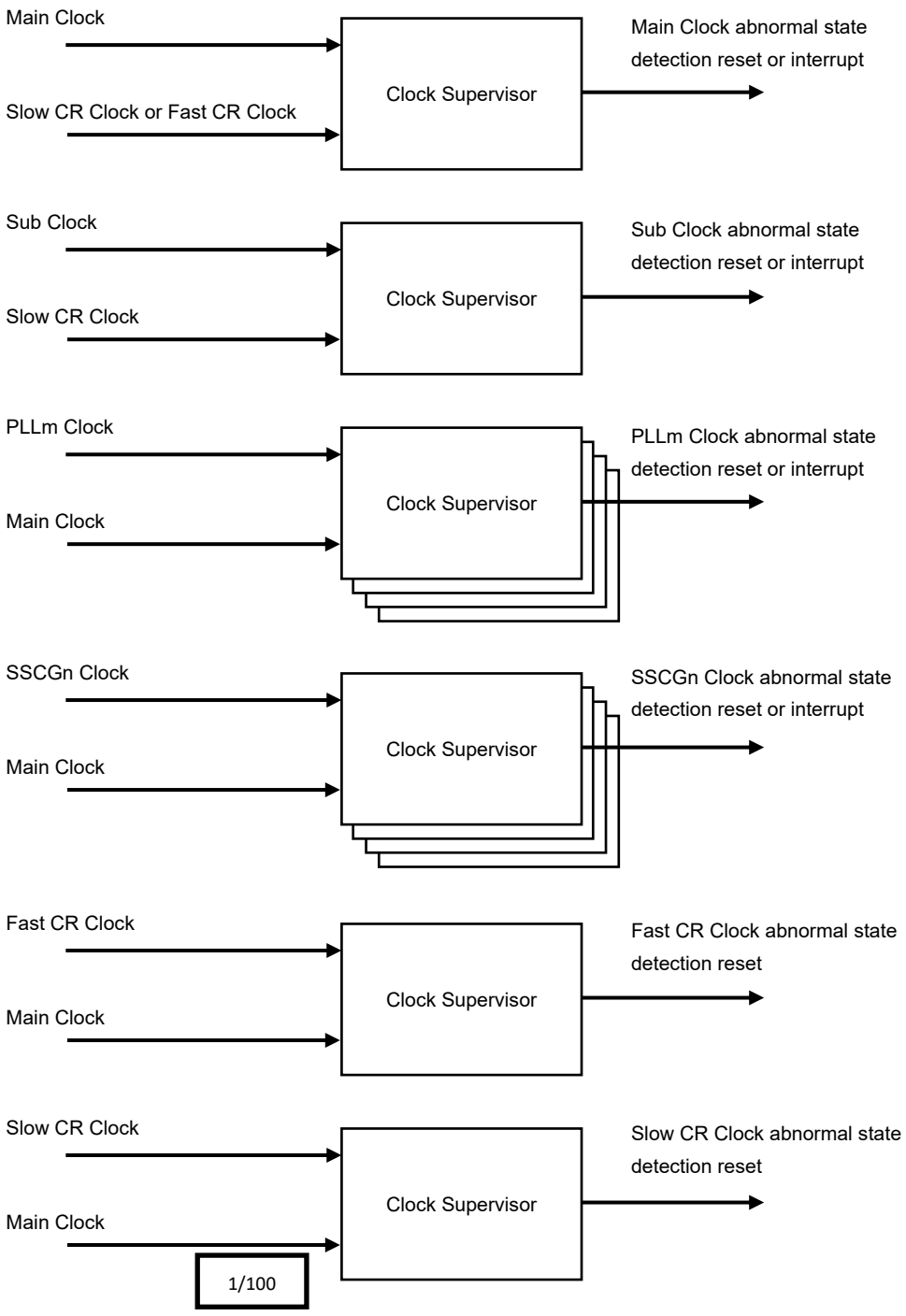
Table 1-1 Overview of Clock Supervisor Instances

Monitored Clock	Reference Clock	Clock Supervisor Type
Main clock	Selection from Fast CR clock or Slow CR clock	Detector for stopped clock or abnormal frequency of the monitored clock
Sub clock	Slow CR clock	
PLL0 clock	Main clock	
PLL1 clock	Main clock	
PLL2 clock	Main clock	
PLL3 clock	Main clock	
SSCG0 clock	Main clock	
SSCG1 clock	Main clock	
SSCG2 clock	Main clock	
SSCG3 clock	Main clock	
Fast CR clock	Main clock	Detector for stopped clock only.
Slow CR clock	Main clock divided by 100	

2. Configuration

The following figure is a block diagram of the clock supervisor.

Figure 2-1: Clock Supervisor Configuration



(m = 0, 1, 2, 3), (n = 0, 1, 2, 3)

Main Clock Supervisor

- Main clock supervisor can detect stop or frequency range abnormality of the main clock.
- The reference clock is the Fast CR clock or the Slow CR clock.
- Upon detecting an abnormality of the main clock, the supervisor generates a reset or interrupt.

Sub Clock Supervisor

- Sub clock supervisor can detect stop or frequency range abnormality of the sub clock.
- The reference clock is the Slow CR clock.
- Upon detecting an abnormality of the sub clock, the supervisor generates a reset or interrupt.

PLLm Clock Supervisor

- The PLLm clock supervisor can detect stop or frequency range abnormality of the PLLm clock.
- The reference clock is the main clock.
- Upon detecting an abnormality of the PLLm clock, the supervisor generates a reset or interrupt.

SSCG PLLn Clock Supervisor

- The SSCG clock supervisor can detect stop or frequency range abnormality of the SSCG clock.
- The reference clock is the main clock.
- Upon detecting an abnormality of the SSCG clock, the supervisor generates a reset or interrupt.

Fast CR Clock Supervisor

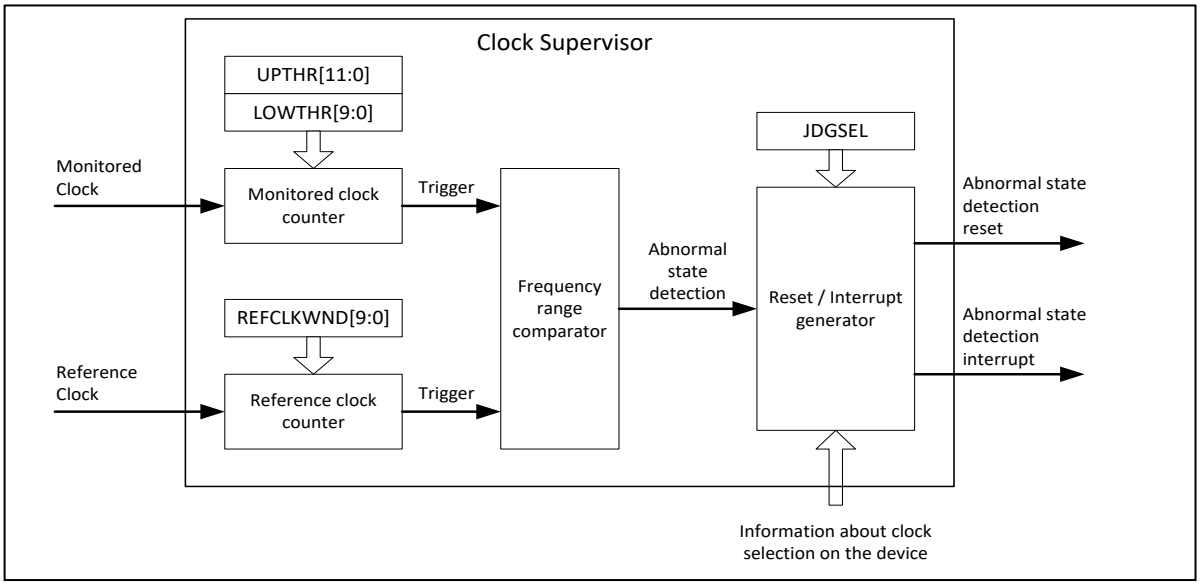
- The Fast CR clock supervisor can detect stop of the Fast CR clock.
- The reference clock is the main clock.
- Upon detecting an abnormality of the Fast CR clock, the supervisor generates a reset.

Slow CR Clock Supervisor

- The Slow CR clock supervisor can detect stop of the Slow CR clock.
- The reference clock uses the clock divided by 100 of the main clock.
- Upon detecting an abnormality of the Slow CR clock, the supervisor generates a reset.

2.1. Detector for Stopped Clock or Abnormal Frequency of the Monitored Clock

Figure 2-2: Block Diagram of Clock Supervisor for stopped clock or abnormal frequency of the monitored clock



Monitored Clock Counter

There are 2 down counters for counting with the monitored clock. The counters load the values that are set in the upper-limit threshold value bits (UPTHR[11:0]) and lower-limit threshold value bits (LOWTHR[9:0]), and count down. When a counter value reaches "0", the counter sends a trigger to the frequency range comparator to make a comparison. The counter reloads the register values and restarts counting down when the frequency comparator makes the comparison.

Reference Clock Counter

There is a down counter for counting with the reference clock. The counter loads the value that is set in the reference clock count period bits (REFCLKWND[9:0]), and counts down. When the counter value reaches "0", the counter sends a trigger to the frequency range comparator to make a comparison. The counter reloads the register value and restarts counting down when the frequency range comparator makes the comparison.

Frequency Range Comparator

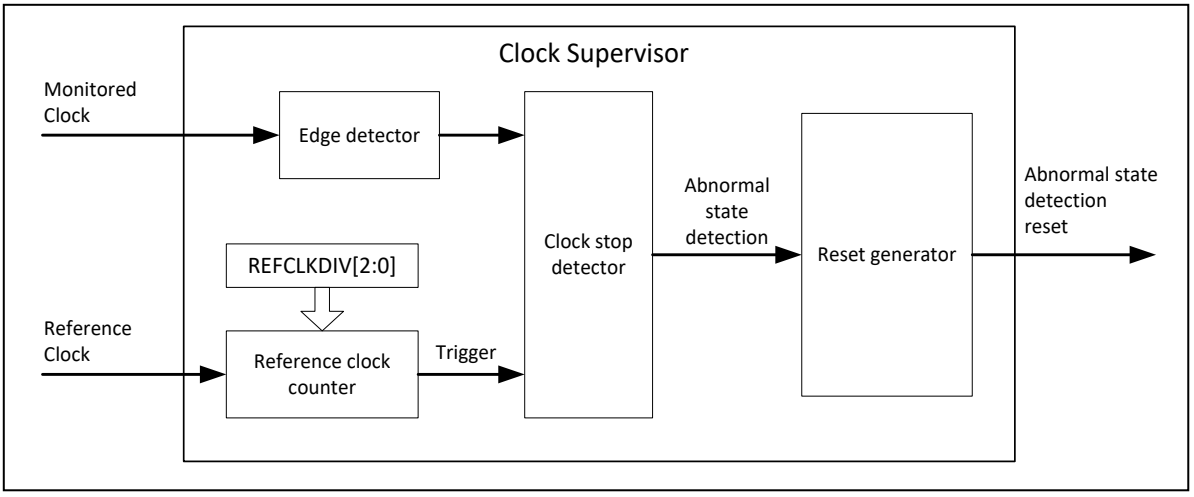
The frequency range comparator compares frequency ranges according to triggers from the monitored clock counters and reference clock counter. The lower-limit threshold value bits (LOWTHR[9:0]) and upper-limit threshold value bits (UPTHR[11:0]) of the monitored clock counters set a frequency range. The comparator judges a range outside this frequency range to be an abnormality.

Reset/Interrupt Generator

Upon detecting an abnormality of a clock, the reset/interrupt generator determines whether to generate a reset or interrupt according to the use status of the monitored clock.

2.2. Detector for Stopped Clock Only

Figure 2-3: Block Diagram of Detector for Stopped Clock Only



Reference Clock Counter

There is a counter for counting with the reference clock. The counter divides the reference clock according to the setting value of the reference clock divide setting (REFCLKDIV[2:0]),

Edge detection

The edge detection detects the rising edge of the monitored clock.

Clock stop Detection Judgment

The clock stop detection judgment judges the presence of rising edge of the monitored clock within the period of the clock that is divided by the reference clock. It judges as normal when it detects a rising edge.

It judges as abnormal when it cannot detect a rising edge.

Reset Generator

Upon detecting an abnormality of a clock, the reset generator generates a reset.

3. Explanation of Operation

This section explains the operation of the clock supervisor.

Main Clock Supervisor

- Main clock supervisor setting register 0 (SYSC_CSVMOCFGR0) and main clock supervisor setting register 1 (SYSC_CSVMOCFGR1) are used for settings.
- Reset generation or interrupt generation can be selected with the judgment selection bit (JDGSEL).
- Upon detecting an abnormality of the main clock and generating a reset, the main clock supervisor can confirm it with the main clock supervisor reset detection bit (CSVMOR) in the user reset factor register (SYSC_RSTCAUSEUR).
- Upon detecting an abnormality of the main clock and generating an interrupt, the main clock supervisor sets the MOSCIF bit in system error interrupt factor register 0 (SYSC0_SYSERRIR0).

Sub Clock Supervisor

- Sub clock supervisor setting register 0 (SYSC_CSVSOCFGR0) and sub clock supervisor setting register 1 (SYSC_CSVSOCFGR1) are used for settings.
- Reset generation or interrupt generation can be selected with the judgment selection bit (JDGSEL).
- Upon detecting an abnormality of the sub clock and generating a reset, the sub clock supervisor can confirm it with the sub clock supervisor reset detection bit (CSVSOR) in the user reset factor register (SYSC_RSTCAUSEUR).
- Upon detecting an abnormality of the sub clock and generating an interrupt, the sub clock supervisor sets the SOSCIF bit in system error interrupt factor register 0 (SYSC0_SYSERRIR0).

PLL Clock Supervisor

- PLLm clock supervisor setting register 0 (SYSC_CSVPLLmCFGR0) and PLLm clock supervisor setting register 1 (SYSC_CSVPLLmCFGR1) are used for settings.
- In case of PLL0, If PLL0 clock is selected for clock domain 0 or MCUC clock domain when abnormal state is detected by CSV, CSV generates a reset. CSV generates an interrupt under other conditions.
- In case of PLL1, PLL2 or PLL3, reset generation or interrupt generation can be selected with the judgment selection bit (JDGSEL).
- Upon detecting an abnormality of the PLL clock and generating a reset, the PLL clock supervisor can confirm it with the PLL clock supervisor reset detection bit (CSVPR) in the user reset factor register (SYSC_RSTCAUSEUR).
- Upon detecting an abnormality of the PLL clock and generating an interrupt, the PLL clock supervisor sets the PLLmIF bit in system error interrupt factor register 0 (SYSC0_SYSERRIR0).

SSCG PLL Clock Supervisor

- SSCG PLLn clock supervisor setting register 0 (SYSC_CSVSPnCFGR0) and SSCG PLLn clock supervisor setting register 1 (SYSC_CSVSPnCFGR1) are used for settings.
- In case of PLL0, If SSCG clock is selected for clock domain 0 or MCUC clock domain when abnormal state is detected by CSV, CSV generates a reset. CSV generates an interrupt under other conditions.
- In case of SSCG1, SSCG2 or SSCG3, reset generation or interrupt generation can be selected with the judgment selection bit (JDGSEL).
- Upon detecting an abnormality of the SSCG PLL clock and generating a reset, the SSCG PLL clock supervisor can confirm it with the SSCG PLL clock supervisor reset detection bit (CSVSR) in the user reset factor register (SYSC_RSTCAUSEUR).
- Upon detecting an abnormality of the SSCG PLL clock and generating an interrupt, the SSCG PLL clock supervisor sets the SSCGnIF bit in system error interrupt factor register 0 (SYSC0_SYSERRIR0).

Fast CR Clock Supervisor

- Fast CR clock supervisor setting register (SYSC_CSVFCRCFGR) is used for settings.
- Upon detecting an abnormality of the Fast CR clock and generating a reset, the Fast CR clock supervisor can confirm it with the Fast CR clock supervisor reset detection bit (CSVFCRR) in the user reset factor register (SYSC_RSTCAUSEUR).

Slow CR Clock Supervisor

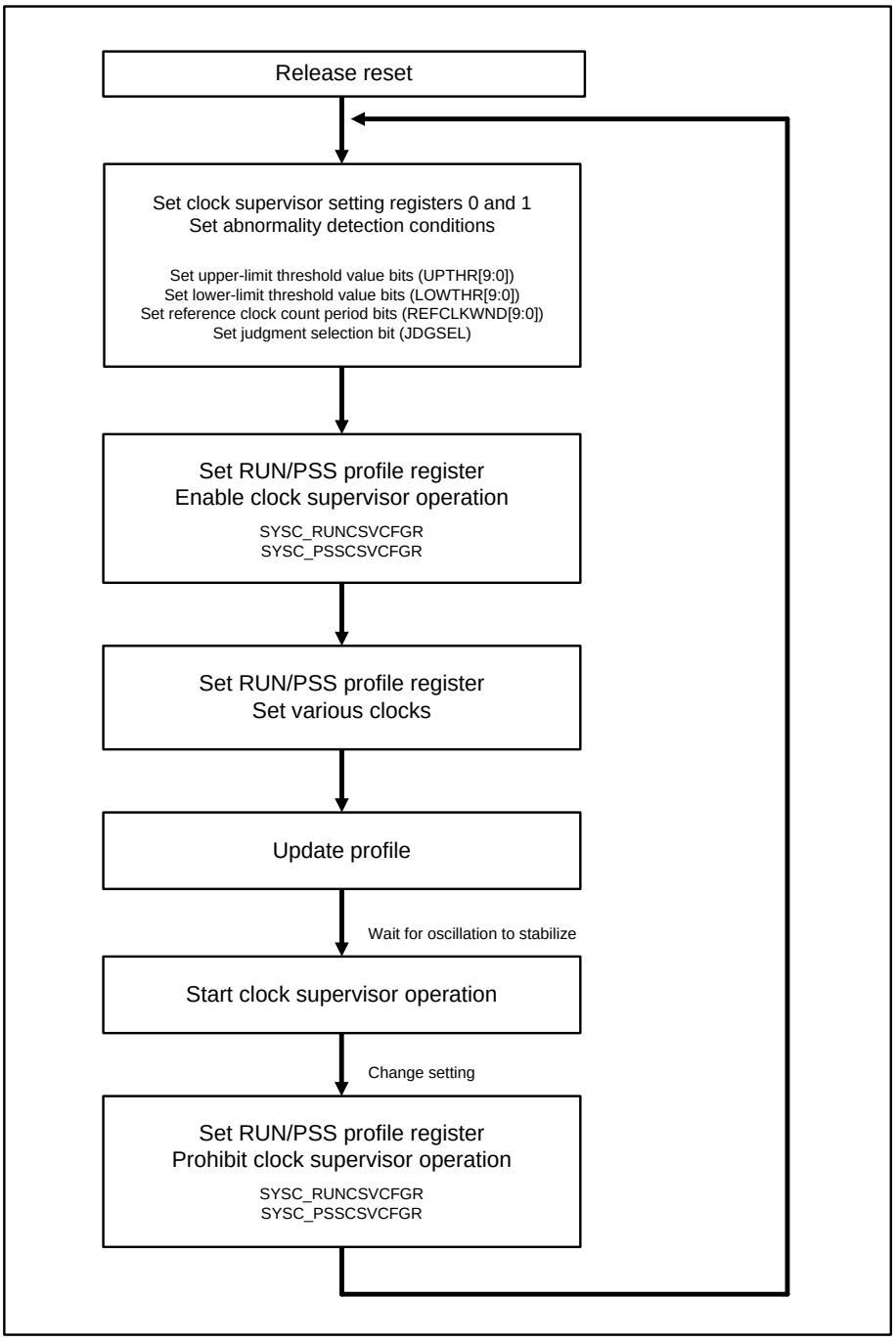
- Slow CR clock supervisor setting register (SYSC_CSVSCRCFGR) is used for settings.
- Upon detecting an abnormality of the Slow CR clock and generating a reset, the Slow CR clock supervisor can confirm it with the Slow CR clock supervisor reset detection bit (CSVSCRR) in the user reset factor register (SYSC_RSTCAUSEUR).

4. Setting Procedure Example

This section shows an example of the clock supervisor setting procedure.

The following figure shows an example of the clock supervisor setting procedure.

Figure 4-1 Example of Clock Supervisor Setting Procedure



Note:

- *For details on clock settings, see the following chapters: "CLOCK SYSTEM" and "LOW POWER CONSUMPTION."*

- After the hard reset is released, all clock supervisors are in the operation disabled state.
- Profile settings are used to enable/disable the main clock supervisor and PLL clock supervisor. Any violation in the profile settings is judged to be a profile error.
- For details on profile errors, see the following chapter: "LOW POWER CONSUMPTION."
- The clock supervisor does not operate until monitored clock and reference clock oscillation have stabilized.
- The clock supervisor can detect artificially the clock abnormality detection state by gating the monitored clock for self-testing.

5. Registers

This section lists the clock supervisor registers.

The clock supervisor registers are as follows:

These registers are subject to the SYSC0 protection register.

The response to any writing done to reserved bit is a bus error.

- RUN profile registers
- PSS profile registers
- APP profile registers
- STS profile registers
- CSV configuration register group

For details on the RUN profile registers, PSS profile registers, APP profile registers, and STS profile registers, see the following chapter: "LOW POWER CONSUMPTION."

Table 5-1 List of Clock Supervisor Registers (CSV Configuration Registers)

Abbreviated Register Name	Register Name	Reference
SYSC_CSVMOCFGR0	Main clock supervisor setting register 0	5.1
SYSC_CSVMOCFGR1	Main clock supervisor setting register 1	5.2
SYSC_CSVSOCFGR0	Sub clock supervisor setting register 0	5.3
SYSC_CSVSOCFGR1	Sub clock supervisor setting register 1	5.4
SYSC_CSVPLL0CFGR0	PLL0 clock supervisor setting register 0	5.5
SYSC_CSVPLL0CFGR1	PLL0 clock supervisor setting register 1	5.6
SYSC_CSVPLL1CFGR0	PLL1 clock supervisor setting register 0	5.5
SYSC_CSVPLL1CFGR1	PLL1 clock supervisor setting register 1	5.6
SYSC_CSVPLL2CFGR0	PLL2 clock supervisor setting register 0	5.5
SYSC_CSVPLL2CFGR1	PLL2 clock supervisor setting register 1	5.6
SYSC_CSVPLL3CFGR0	PLL3 clock supervisor setting register 0	5.5
SYSC_CSVPLL3CFGR1	PLL3 clock supervisor setting register 1	5.6
SYSC_CSVSP0CFGR0	SSCG PLL0 clock supervisor setting register 0	5.7
SYSC_CSVSP0CFGR1	SSCG PLL0 clock supervisor setting register 1	5.8
SYSC_CSVSP1CFGR0	SSCG PLL1 clock supervisor setting register 0	5.7
SYSC_CSVSP1CFGR1	SSCG PLL1 clock supervisor setting register 1	5.8
SYSC_CSVSP2CFGR0	SSCG PLL2 clock supervisor setting register 0	5.7
SYSC_CSVSP2CFGR1	SSCG PLL2 clock supervisor setting register 1	5.8
SYSC_CSVSP3CFGR0	SSCG PLL3 clock supervisor setting register 0	5.7
SYSC_CSVSP3CFGR1	SSCG PLL3 clock supervisor setting register 1	5.8
SYSC_CSVFCRCFGR	Fast CR clock supervisor setting register	5.9
SYSC_CSVSCRCFGR	Slow CR clock supervisor setting register	5.10
SYSC_CSVOUTER	Clock supervisor output enable register	5.11
SYSC_CSVTESTR	Clock supervisor test register	5.12

Table 5-2 Clock Supervisor Register Map (B/B)

Offset	Register Name/Initial Value
0x0000_0000	SYSC_CSVMOCFGR0 00000000_00000000_00000000_00000000
0x0000_0004	SYSC_CSVMOCFGR1 00000000_00000000_00000000_00000000
0x0000_0008	SYSC_CSVSOCFGR0 00000000_00000000_00000000_00000000
0x0000_000C	SYSC_CSVSOCFGR1 00000000_00000000_00000000_00000000
0x0000_0010	SYSC_CSVPLL0CFGR0 00000000_00000000_00000000_00000000
0x0000_0014	SYSC_CSVPLL0CFGR1 00000000_00000000_00000000_00000000
0x0000_0018	SYSC_CSVPLL1CFGR0 00000000_00000000_00000000_00000000
0x0000_001C	SYSC_CSVPLL1CFGR1 00000000_00000000_00000000_00000000
0x0000_0020	SYSC_CSVPLL2CFGR0 00000000_00000000_00000000_00000000
0x0000_0024	SYSC_CSVPLL2CFGR1 00000000_00000000_00000000_00000000
0x0000_0028	SYSC_CSVPLL3CFGR0 00000000_00000000_00000000_00000000
0x0000_002C	SYSC_CSVPLL3CFGR1 00000000_00000000_00000000_00000000
0x0000_0030	SYSC_CSVSP0CFGR0 00000000_00000000_00000000_00000000
0x0000_0034	SYSC_CSVSP0CFGR1 00000000_00000000_00000000_00000000
0x0000_0038	SYSC_CSVSP1CFGR0 00000000_00000000_00000000_00000000
0x0000_003C	SYSC_CSVSP1CFGR1 00000000_00000000_00000000_00000000
0x0000_0040	SYSC_CSVSP2CFGR0 00000000_00000000_00000000_00000000
0x0000_0044	SYSC_CSVSP2CFGR1 00000000_00000000_00000000_00000000
0x0000_0048	SYSC_CSVSP3CFGR0 00000000_00000000_00000000_00000000
0x0000_004C	SYSC_CSVSP3CFGR1 00000000_00000000_00000000_00000000
0x0000_0050	SYSC_CSVFCRCFGR 00000000_00000000_00000000_00000010
0x0000_0054	SYSC_CSVSCRCFGR 00000000_00000000_00000000_00000010

Offset	Register Name/Initial Value
0x0000_0058 to 0x0000_005C	Reserved
0x0000_0060	SYSC_CSVOUTER 00000000_00000000_00000000_00000000
0x0000_0064	SYSC_CSVTESTR 00000000_00000000_00000000_00000000

5.1. Main Clock Supervisor Setting Register 0 (SYSC_CSVMOCFGR0)

Main clock supervisor setting register 0 (SYSC_CSVMOCFGR0) sets the upper-limit threshold value and lower-limit threshold value of a frequency range.

Bit	31	30	29	28	27	26	25	24
Field	Reserved				UPTHR[11:8]			
R/W Attribute	R0,WX				R/W			
Protection Attribute	WPS							
Initial Value	0000				0000			

Bit	23	22	21	20	19	18	17	16
Field	UPTHR[7:0]							
R/W Attribute	R/W							
Protection Attribute	WPS							
Initial Value	00000000							

Bit	15	14	13	12	11	10	9	8
Field	Reserved						LOWTHR[9:8]	
R/W Attribute	R0,WX						R/W	
Protection Attribute	WPS							
Initial Value	000000						00	

Bit	7	6	5	4	3	2	1	0
Field	LOWTHR[7:0]							
R/W Attribute	R/W							
Protection Attribute	WPS							
Initial Value	00000000							

[bit31:28] Reserved: Reserved bits

[bit27:16] UPTHR[11:0]: Upper-limit threshold value bits

These bits set the clock upper-limit threshold value for comparison with the monitored clock count value. If the monitored clock counter value exceeds the upper-limit threshold value, the state is judged as abnormal.

[bit15:10] Reserved: Reserved bits

[bit9:0] LOWTHR[9:0]: Lower-limit threshold value bits

These bits set the clock lower-limit threshold value for comparison with the monitored clock count value. If the monitored clock counter value falls below the lower-limit threshold value, the state is judged as abnormal.

Notes:

- *This register must not be modified during operation of the main clock supervisor. The response to any writing done here is a bus error.*
- *For the lower-limit threshold value bits (LOWTHR), set a value larger than that obtained from dividing the reference clock cycle by the monitored clock cycle.*
- *Set the upper-limit threshold value bit (UPTHR) value to be greater than the lower-limit threshold value bit (LOWTHR) value.*

5.2. Main Clock Supervisor Setting Register 1 (SYSC_CSVMOCFGR1)

Main clock supervisor setting register 1 (SYSC_CSVMOCFGR1) can set a reference clock count period. It can also select whether to generate a reset or interrupt at the abnormal state detection time.

Bit	31	30	29	28	27	26	25	24
Field	Reserved							REFCLKSEL
R/W Attribute	R0,WX							R/W
Protection Attribute	WPS							
Initial Value	0000000							0

Bit	23	22	21	20	19	18	17	16
Field	Reserved							JDGSEL
R/W Attribute	R0,WX							R/W
Protection Attribute	WPS							
Initial Value	0000000							0

Bit	15	14	13	12	11	10	9	8
Field	Reserved						REFCLKWND[9:8]	
R/W Attribute	R0,WX						R/W	
Protection Attribute	WPS							
Initial Value	000000						00	

Bit	7	6	5	4	3	2	1	0
Field	REFCLKWND[7:0]							
R/W Attribute	R/W							
Protection Attribute	WPS							
Initial Value	00000000							

[bit31:25] Reserved: Reserved bits

[bit24] REFCLKSEL: Reference clock selection bit

The reference clock can be selected with this bit.

Bit	Description
0	Slow CR clock
1	Fast CR clock

[bit23:17] Reserved: Reserved bits

[bit16] JDGSEL: Judgment selection bit

This bit selects a reset condition when abnormal state is detected by CSV.

Bit	Description
0	Main clock is selected for clock domain 0 or for MCUC clock domain or for software watchdog timer.
1	Main clock is selected for clock domain 0 or for MCUC clock domain.

– *If it is not the above condition when an abnormal state is detected by CSV. CSV generates interrupt.*

[bit15:10] Reserved: Reserved bits

[bit9:0] REFCLKWND[9:0]: Reference clock Count period bits

These bits set the reference clock count value used to trigger a comparison by the frequency range comparator.

Note:

– *This register must not be modified during operation of the main clock supervisor. The response to any writing done here is a bus error.*

5.3. Sub Clock Supervisor Setting Register 0 (SYSC_CSVSOCFGR0)

Sub clock supervisor setting register 0 (SYSC_CSVSOCFGR0) sets the upper-limit threshold value and lower-limit threshold value of a frequency range.

Bit	31	30	29	28	27	26	25	24
Field	Reserved				UPTHR[11:8]			
R/W Attribute	R0,WX				R/W			
Protection Attribute	WPS							
Initial Value	0000				0000			

Bit	23	22	21	20	19	18	17	16
Field	UPTHR[7:0]							
R/W Attribute	R/W							
Protection Attribute	WPS							
Initial Value	00000000							

Bit	15	14	13	12	11	10	9	8
Field	Reserved						LOWTHR[9:8]	
R/W Attribute	R0,WX						R/W	
Protection Attribute	WPS							
Initial Value	000000						00	

Bit	7	6	5	4	3	2	1	0
Field	LOWTHR[7:0]							
R/W Attribute	R/W							
Protection Attribute	WPS							
Initial Value	00000000							

[bit31:28] Reserved: Reserved bits

[bit27:16] UPTHR[11:0]: Upper-limit threshold value bits

These bits set the clock upper-limit threshold value for comparison with the monitored clock count value. If the monitored clock counter value exceeds the upper-limit threshold value, the state is judged as abnormal.

[bit15:10] Reserved: Reserved bits

[bit9:0] LOWTHR[9:0]: Lower-limit threshold value bits

These bits set the clock lower-limit threshold value for comparison with the monitored clock count value. If the monitored clock counter value falls below the lower-limit threshold value, the state is judged as abnormal.

Notes:

- *This register must not be modified during operation of the sub clock supervisor. The response to any writing done here is a bus error.*
- *For the lower-limit threshold value bits (LOWTHR), set a value larger than that obtained from dividing the reference clock cycle by the monitored clock cycle.*
- *Set the upper-limit threshold value bit (UPTHR) value to be greater than the lower-limit threshold value bit (LOWTHR) value.*

5.4. Sub Clock Supervisor Setting Register 1 (SYSC_CSVSOCFGR1)

Sub clock supervisor setting register 1 (SYSC_CSVSOCFGR1) can set a reference clock count period. It can also select whether to generate a reset or interrupt at the abnormal state detection time.

Bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	WPS							
Initial Value	00000000							

Bit	23	22	21	20	19	18	17	16
Field	Reserved							JDGSEL
R/W Attribute	R0,WX							R/W
Protection Attribute	WPS							
Initial Value	0000000							0

Bit	15	14	13	12	11	10	9	8
Field	Reserved						REFCLKWND[9:8]	
R/W Attribute	R0,WX						R/W	
Protection Attribute	WPS							
Initial Value	000000						00	

Bit	7	6	5	4	3	2	1	0
Field	REFCLKWND[7:0]							
R/W Attribute	R/W							
Protection Attribute	WPS							
Initial Value	00000000							

[bit31:17] Reserved: Reserved bits

[bit16] JDGSEL: Judgment selection bit

This bit selects a reset condition when abnormal state is detected by CSV.

Bit	Description
0	Sub clock is selected for clock domain 0 or for MCUC clock domain or for software watchdog timer.
1	Sub clock is selected for clock domain 0 or for MCUC clock domain.

– *If it is not the above condition when an abnormal state is detected by CSV, CSV generates interrupt.*

[bit15:10] Reserved: Reserved bits

[bit9:0] REFCLKWND[9:0]: Reference clock count period bits

These bits set the reference clock cycle used to trigger a comparison by the frequency range comparator.

Note:

- *This register must not be modified during operation of the sub clock supervisor. The response to any writing done here is a bus error.*

5.5. PLLm Clock Supervisor Setting Register 0 (SYSC_CSVPLLmCFGR0)

PLLm clock supervisor setting register 0 (SYSC_CSVPLLmCFGR0) sets the upper-limit threshold value and lower-limit threshold value of a frequency range.

Bit	31	30	29	28	27	26	25	24
Field	Reserved				UPTHR[11:8]			
R/W Attribute	R0,WX				R/W			
Protection Attribute	WPS							
Initial Value	0000				0000			

Bit	23	22	21	20	19	18	17	16
Field	UPTHR[7:0]							
R/W Attribute	R/W							
Protection Attribute	WPS							
Initial Value	00000000							

Bit	15	14	13	12	11	10	9	8
Field	Reserved						LOWTHR[9:8]	
R/W Attribute	R0,WX						R/W	
Protection Attribute	WPS							
Initial Value	000000						00	

Bit	7	6	5	4	3	2	1	0
Field	LOWTHR[7:0]							
R/W Attribute	R/W							
Protection Attribute	WPS							
Initial Value	00000000							

[bit31:28] Reserved: Reserved bits

[bit27:16] UPTHR[11:0]: Upper-limit threshold value bits

These bits set the clock upper-limit threshold value for comparison with the monitored clock count value. If the monitored clock counter value exceeds the upper-limit threshold value, the state is judged as abnormal.

[bit15:10] Reserved: Reserved bits

[bit9:0] LOWTHR[9:0]: Lower-limit threshold value bits

These bits set the clock lower-limit threshold value for comparison with the monitored clock count value. If the monitored clock counter value falls below the lower-limit threshold value, the state is judged as abnormal.

Notes:

- *This register must not be modified during operation of the PLL clock supervisor. The response to any writing done here is a bus error.*
- *For the lower-limit threshold value bits (LOWTHR), set a value larger than that obtained from dividing the reference clock cycle by the monitored clock cycle.*
- *Set the upper-limit threshold value bit (UPTHR) value to be greater than the lower-limit threshold value bit (LOWTHR) value.*

5.6. PLLm Clock Supervisor Setting Register 1 (SYSC_CSVPLLmCFGR1)

PLLm clock supervisor setting register 1 (SYSC_CSVPLLmCFGR1) sets a reference clock count period.

Bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	WPS							
Initial Value	00000000							

Bit	23	22	21	20	19	18	17	16
Field	Reserved							JDGSEL
R/W Attribute	R0,WX							R/W
Protection Attribute	WPS							
Initial Value	0000000							0

Bit	15	14	13	12	11	10	9	8
Field	Reserved						REFCLKWND[9:8]	
R/W Attribute	R0,WX						R/W	
Protection Attribute	WPS							
Initial Value	000000						00	

Bit	7	6	5	4	3	2	1	0
Field	REFCLKWND[7:0]							
R/W Attribute	R/W							
Protection Attribute	WPS							
Initial Value	00000000							

[bit31:17] Reserved: Reserved bits

[bit16] JDGSEL: Judgment selection bit

This bit selects a reset condition when abnormal state is detected by CSV.

Bit	Description
0	Main clock is selected for clock domain 0 , MCUC clock domain or software watchdog timer.
1	Main clock is selected for clock domain 0 or MCUC clock domain

– *If it is not the above condition when an abnormal state is detected by CSV, CSV generates interrupt.*

[bit15:10] Reserved: Reserved bits

[bit9:0] REFCLKWND[9:0]: Reference clock count period bits

These bits set the reference clock count value used to trigger a comparison by the frequency range comparator.

Note:

- *This register must not be modified during operation of the PLL clock supervisor. The response to any writing done here is a bus error.*

5.7. SSCG PLLn Clock Supervisor Setting Register 0 (SYSC_CSVSPnCFGR0)

SSCG PLLn clock supervisor setting register 0 (SYSC_CSVSPnCFGR0) sets the upper-limit threshold value and lower-limit threshold value of a frequency range.

Bit	31	30	29	28	27	26	25	24
Field	Reserved				UPTHR[11:8]			
R/W Attribute	R0,WX				R/W			
Protection Attribute	WPS							
Initial Value	0000				0000			

Bit	23	22	21	20	19	18	17	16
Field	UPTHR[7:0]							
R/W Attribute	R/W							
Protection Attribute	WPS							
Initial Value	00000000							

Bit	15	14	13	12	11	10	9	8
Field	Reserved						LOWTHR[9:8]	
R/W Attribute	R0,WX						R/W	
Protection Attribute	WPS							
Initial Value	000000						00	

Bit	7	6	5	4	3	2	1	0
Field	LOWTHR[7:0]							
R/W Attribute	R/W							
Protection Attribute	WPS							
Initial Value	00000000							

[bit31:28] Reserved: Reserved bits

[bit27:16] UPTHR[11:0]: Upper-limit threshold value bits

These bits set the clock upper-limit threshold value for comparison with the monitored clock count value. If the monitored clock counter value exceeds the upper-limit threshold value, the state is judged as abnormal.

[bit15:10] Reserved: Reserved bits

[bit9:0] LOWTHR[9:0]: Lower-limit threshold value bits

These bits set the clock lower-limit threshold value for comparison with the monitored clock count value. If the monitored clock counter value falls below the lower-limit threshold value, the state is judged as abnormal.

Notes:

- *This register must not be modified during operation of the SSCG PLL clock supervisor. The response to any writing done here is a bus error.*
- *For the lower-limit threshold value bits (LOWTHR), set a value larger than that obtained from dividing the reference clock cycle by the monitored clock cycle.*
- *Set the upper-limit threshold value bit (UPTHR) value to be greater than the lower-limit threshold value bit (LOWTHR) value.*

5.8. SSCG PLLn Clock Supervisor Setting Register 1 (SYSC_CSVSPnCFGR1)

SSCG PLLn clock supervisor setting register 1 (SYSC_CSVSPnCFGR1) can set a reference clock count period. It can also select whether to generate a reset or interrupt at the abnormal state detection time.

Bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	WPS							
Initial Value	00000000							

Bit	23	22	21	20	19	18	17	16
Field	Reserved							JDGSEL
R/W Attribute	R0,WX							R/W
Protection Attribute	WPS							
Initial Value	0000000							0

Bit	15	14	13	12	11	10	9	8
Field	Reserved						REFCLKWND[9:8]	
R/W Attribute	R0,WX						R/W	
Protection Attribute	WPS							
Initial Value	000000						00	

Bit	7	6	5	4	3	2	1	0
Field	REFCLKWND[7:0]							
R/W Attribute	R/W							
Protection Attribute	WPS							
Initial Value	00000000							

[bit31:17] Reserved: Reserved bits

[bit16] JDGSEL: Judgment selection bit

This bit selects a reset condition when abnormal state is detected by CSV.

Bit	Description
0	Main clock is selected for clock domain 0 , MCUC clock domain or software watchdog timer.
1	Main clock is selected for clock domain 0 or MCUC clock domain

- *If it is not the above condition when an abnormal state is detected by CSV, CSV generates interrupt.*

[bit15:10] Reserved: Reserved bits

[bit9:0] REFCLKWND[9:0]: Reference clock count period bits

These bits set the reference clock cycle used to trigger a comparison by the frequency range comparator.

Note:

- *This register must not be modified during operation of the SSCG PLL clock supervisor. The response to any writing done here is a bus error.*

5.9. Fast CR Clock Supervisor Setting Register (SYSC_CSVFCRCFGR)

Fast CR clock supervisor setting register (SYSC_CSVFCRCFGR) can select a monitored clock division setting. It can also select a reference clock division setting.

Bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	WPS							
Initial Value	00000000							

Bit	23	22	21	20	19	18	17	16
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	WPS							
Initial Value	00000000							

Bit	15	14	13	12	11	10	9	8
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	WPS							
Initial Value	00000000							

Bit	7	6	5	4	3	2	1	0
Field	Reserved					REFCLKDIV[2:0]		
R/W Attribute	R0,WX					R/W		
Protection Attribute	WPS							
Initial Value	00000					010		

[bit31:3] Reserved: Reserved bits

[bit2:0] REFCLKDIV[2:0]: Reference clock division setting bits

The judgment trigger signal divided from the reference clock is input to the clock stop detection circuit. These bits set the division value of the judgment trigger signal.

Bits	Description
000	Divided by 4
001	Divided by 8
010	Divided by 16 (initial value)
011	Divided by 32
100	Divided by 64
others	Divided by 128

Note:

- *This register must not be modified during operation of the Fast CR clock supervisor. The response to any writing done here is a bus error.*

5.10. Slow CR Clock Supervisor Setting Register (SYSC_CSVSCRCFGR)

Slow CR clock supervisor setting register (SYSC_CSVSCRCFGR) can select a monitored clock division setting. It can also select a reference clock division setting.

Bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	WPS							
Initial Value	00000000							

Bit	23	22	21	20	19	18	17	16
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	WPS							
Initial Value	00000000							

Bit	15	14	13	12	11	10	9	8
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	WPS							
Initial Value	00000000							

Bit	7	6	5	4	3	2	1	0
Field	Reserved					REFCLKDIV[2:0]		
R/W Attribute	R0,WX					R/W		
Protection Attribute	WPS							
Initial Value	00000					010		

[bit31:3] Reserved: Reserved bits

[bit2:0] REFCLKDIV[2:0]: Reference clock division setting bits

The judgment trigger signal divided from the reference clock is input to the clock stop detection circuit. These bits set the division value of the judgment trigger signal.

Bits	Description
000	Divided by 4
001	Divided by 8
010	Divided by 16 (initial value)
011	Divided by 32
100	Divided by 64
others	Divided by 128

Note:

- *This register must not be modified during operation of the Slow CR clock supervisor. The response to any writing done here is a bus error.*

5.11. Clock Supervisor Output Enable Register (SYSC_CSVOUTER)

The clock supervisor output enable register (SYSC_CSVOUTER) enables output of the abnormality detection bit to a port.

Bit	31	8
Field	Reserved	
R/W Attribute	R0,WX	
Protection Attribute	WPS	
Initial Value	00000000 00000000 00000000	

Bit	7	6	5	4	3	2	1	0
Field	Reserved							OUTEN
R/W Attribute	R0,WX							R/W
Protection Attribute	WPS							
Initial Value	0000000							0

[bit31:1] Reserved: Reserved bits

[bit0] OUTEN: Clock supervisor output enable bit

This bit is the enable bit for outputting the abnormality detection bit to a port.

Bit	Description
0	Does not enable output of the abnormality detection bit from a port.
1	Enables output of the abnormality detection bit from a port.

For output ports, see the list of pin functions in "Overview" and the chapter of "I/O PORT."

Note:

- *This bit is cleared to "0" by to a power-on reset, an internal power supply low-voltage detection reset, the terminal INITX input reset, a RAM retain low-voltage detection reset and an illegal mode detection reset. It is not affected by other types of reset.*

5.12. Clock Supervisor Test Register (SYSC_CSVTESTR)

The clock supervisor test register (SYSC_CSVTESTR) can perform function tests using input clock gating.

Bit	31	30	29	28	27	26	25	24
Field	Reserved						SCRCLK GATE	FCRCLK GATE
R/W Attribute	R0,WX						R/W	
Protection Attribute	WPS							
Initial Value	000000						0	0

Bit	23	22	21	20	19	18	17	16
Field	Reserved				SP3CLK GATE	SP2CLK GATE	SP1CLK GATE	SP0CLK GATE
R/W Attribute	R0,WX				R/W			
Protection Attribute	WPS							
Initial Value	0000				0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	Reserved				PLL3CLK GATE	PLL2CLK GATE	PLL1CLK GATE	PLL0CLK GATE
R/W Attribute	R0,WX				R/W			
Protection Attribute	WPS							
Initial Value	0000				0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	Reserved				SO1CLK GATE	MO1CLK GATE	SO0CLK GATE	MO0CLK GATE
R/W Attribute	R0,WX				R/W			
Protection Attribute	WPS							
Initial Value	0000				0	0	0	0

[bit31:26] Reserved: Reserved bits

[bit25] SCRCLKGATE: SCR CR clock supervisor test bit

This bit can be used for function tests using input clock gating.

Bit	Description
0	Normal operation
1	Input clock gating (clock shutoff)

[bit24] FCRCLKGATE: FCR CR clock supervisor test bit

This bit can be used for function tests using input clock gating.

Bit	Description
0	Normal operation
1	Input clock gating (clock shutoff)

[bit23:20] Reserved: Reserved bits

[bit19] SP3CLKGATE: SSCG PLL3 clock supervisor test bit

This bit can be used for function tests using input clock gating.

Bit	Description
0	Normal operation
1	Input clock gating (clock shutoff)

[bit18] SP2CLKGATE: SSCG PLL2 clock supervisor test bit

This bit can be used for function tests using input clock gating.

Bit	Description
0	Normal operation
1	Input clock gating (clock shutoff)

[bit17] SP1CLKGATE: SSCG PLL1 clock supervisor test bit

This bit can be used for function tests using input clock gating.

Bit	Description
0	Normal operation
1	Input clock gating (clock shutoff)

[bit16] SP0CLKGATE: SSCG PLL0 clock supervisor test bit

This bit can be used for function tests using input clock gating.

Bit	Description
0	Normal operation
1	Input clock gating (clock shutoff)

[bit15:12] Reserved: Reserved bits

[bit11] PLL3CLKGATE: PLL3 clock supervisor test bit

This bit can be used for function tests using input clock gating.

Bit	Description
0	Normal operation
1	Input clock gating (clock shutoff)

[bit10] PLL2CLKGATE: PLL2 clock supervisor test bit

This bit can be used for function tests using input clock gating.

Bit	Description
0	Normal operation
1	Input clock gating (clock shutoff)

[bit9] PLL1CLKGATE: PLL1 clock supervisor test bit

This bit can be used for function tests using input clock gating.

Bit	Description
0	Normal operation
1	Input clock gating (clock shutoff)

[bit8] PLL0CLKGATE: PLL0 clock supervisor test bit

This bit can be used for function tests using input clock gating.

Bit	Description
0	Normal operation
1	Input clock gating (clock shutoff)

[bit7:4] Reserved: Reserved bits

[bit3] SO1CLKGATE: Sub clock 1 supervisor test bit

This bit can be used for function tests using input clock gating.

Bit	Description
0	Normal operation
1	Input clock gating (clock shutoff)

[bit2] MO1CLKGATE: Main clock 1 supervisor test bit

This bit can be used for function tests using input clock gating.

Bit	Description
0	Normal operation
1	Input clock gating (clock shutoff)

[bit1] SO0CLKGATE: Sub clock 0 supervisor test bit

This bit can be used for function tests using input clock gating.

Bit	Description
0	Normal operation
1	Input clock gating (clock shutoff)

[bit0] MO0CLKGATE: Main clock 0 supervisor test bit

This bit can be used for function tests using input clock gating.

Bit	Description
0	Normal operation
1	Input clock gating

6. Precautions for Using This Device

This section explains precautions for using the clock supervisor.

- Profile settings are used to enable/disable operations of the clock supervisor. For details on the profile settings, see the following chapter: "LOW POWER CONSUMPTION."
- To operate the clock supervisor, enable oscillation of the monitored clock and reference clock.
- To stop the clock supervisor, simultaneously stop oscillation of the monitored clock.
- After a reset is generated by the detection type clock supervisor of stop or frequency range abnormality of the clock, MCU will operate with FastCR clock. At this time, CSV cause clock must not be selected again. In addition, please disable the enable of the source clock oscillation of the clock that abnormality is detected.
- To protect hardware, clock supervisor setting registers 0 and 1 must not be modified during operation of the clock supervisor. To change either of them, do so only after disabling operation of the clock supervisor. The response to any writing done here is a bus error.

CHAPTER 9: Source Clock Timer

This chapter explains the functions and operations of the source clock timer.

1. **Overview**
2. **Configuration**
3. **Explanation of Operation**
4. **Setting Procedure Example**
5. **Registers**
6. **Precautions for Using this Device**

SCT-TXXPT03P01R01L07-E1-XX

1. Overview

This section provides an overview of the source clock timer.

The source clock timer is a timer for gating the clock output until the end of the clock oscillation stabilization wait time. After the clock oscillation stabilizes, this timer can be used as an ordinary timer. The 4 source clocks are the high-speed CR clock, low-speed CR clock, main clock, and sub clock, and they can be used separately.

High-Speed CR Clock Timer

The high-speed CR clock timer counts the internal high-speed CR clock. This timer is used to count time to wait for the high-speed CR clock oscillation to stabilize.

The high-speed CR clock timer has an oscillation stabilization wait time.

Low-Speed CR Clock Timer

The low-speed CR clock timer counts the internal low-speed CR clock. This timer is used to count time to wait for the low-speed CR clock oscillation to stabilize.

The low-speed CR clock timer has an oscillation stabilization wait time.

Main Clock Timer

The main clock timer counts the main clock. This timer is used to count time to wait for the main clock oscillation to stabilize.

The main clock timer has an initial value for the oscillation stabilization wait time.

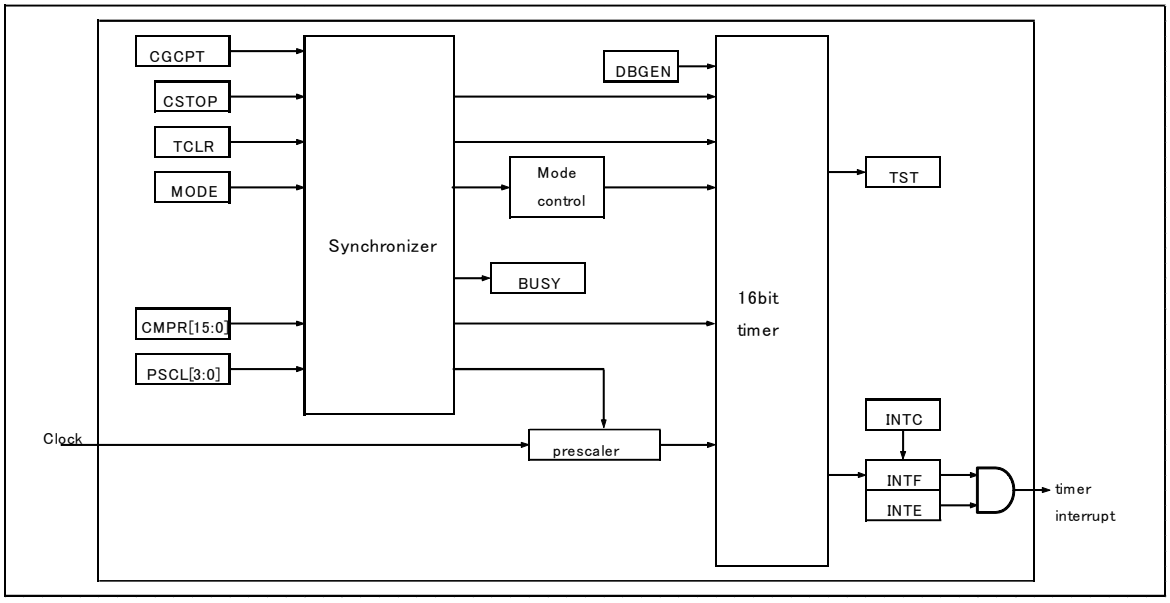
Sub Clock Timer

The sub clock timer counts the sub clock. This timer is used to count time to wait for the sub clock oscillation to stabilize. The sub clock timer has an initial value for the oscillation stabilization wait time.

2. Configuration

This section shows a block diagram of the source clock timer.

Figure 2-1 Block Diagram of Source Clock Timer



Setting Synchronization Block

- The setting synchronization block synchronizes the start of the source clock timer or the resetting of the timer. The timer setting capture bit (CGCPT) is set to "1" to acquire the contents of the timer settings.

Mode Control Block

- The mode control block controls the operation mode of the 16-bit timer.
- The 2 operation modes for the source clock timer are single-shot mode and continuous mode. One of the modes is selected with the mode control bit (MODE).
- Single-shot mode: This mode clears the value to "0x0000" and stops counting up when the count value of the 16-bit timer exceeds the compare value.
- Continuous mode: This mode clears the value to "0x0000" and restarts counting up when the count value of the 16-bit timer exceeds the compare value.
- The mode control bit (MODE) is used to select an operation mode.
- If the debug enable bit (DBGEN) is set to "1", the timer stops when operation stops at a breakpoint during debugging.

Prescaler

- The prescaler divides the input clock. For division, a power of 2 can be selected with the prescale bits (PSCL[3:0]). The range for the selection is from no division (divided by 1) to divided by 32768. The divided clock is supplied to the 16-bit timer.

16-Bit Timer

- The 16-bit timer is configured as a 16-bit counter and 16-bit comparator. This timer counts the clock supplied from the prescaler. The interrupt flag bit (INTF) is set to "1" and the count value is cleared to "0x0000" when the count value exceeds the set value for the compare value bits (CMPR[15:0]).

- In single-shot mode, the interrupt flag bit (INTF) is set to "1" only once. In continuous mode, "1" is set at a regular interval.

3. Explanation of Operation

This section explains the operations of the source clock timer.

Oscillation Stabilization Wait Operation

- Main, Fast-CR and Slow-CR source clock timers start the oscillation stabilization wait operation when a hard reset is released or oscillation is enabled. The source clock timers gate the clock output until the oscillation stabilization wait time ends.
- Sub source clock timer starts the oscillation stabilization wait operation when oscillation is enabled. The source clock timer gates the clock output until the oscillation stabilization wait time ends.
- If data is written to a trigger register, timer compare prescaler register, interrupt enable register, or interrupt clear register during the oscillation stabilization wait operation, release the sequence protection in privileged mode. If the above is not complied with, a bus error is returned.
- During the oscillation stabilization wait operation, the source clock timer operates in single-shot mode.

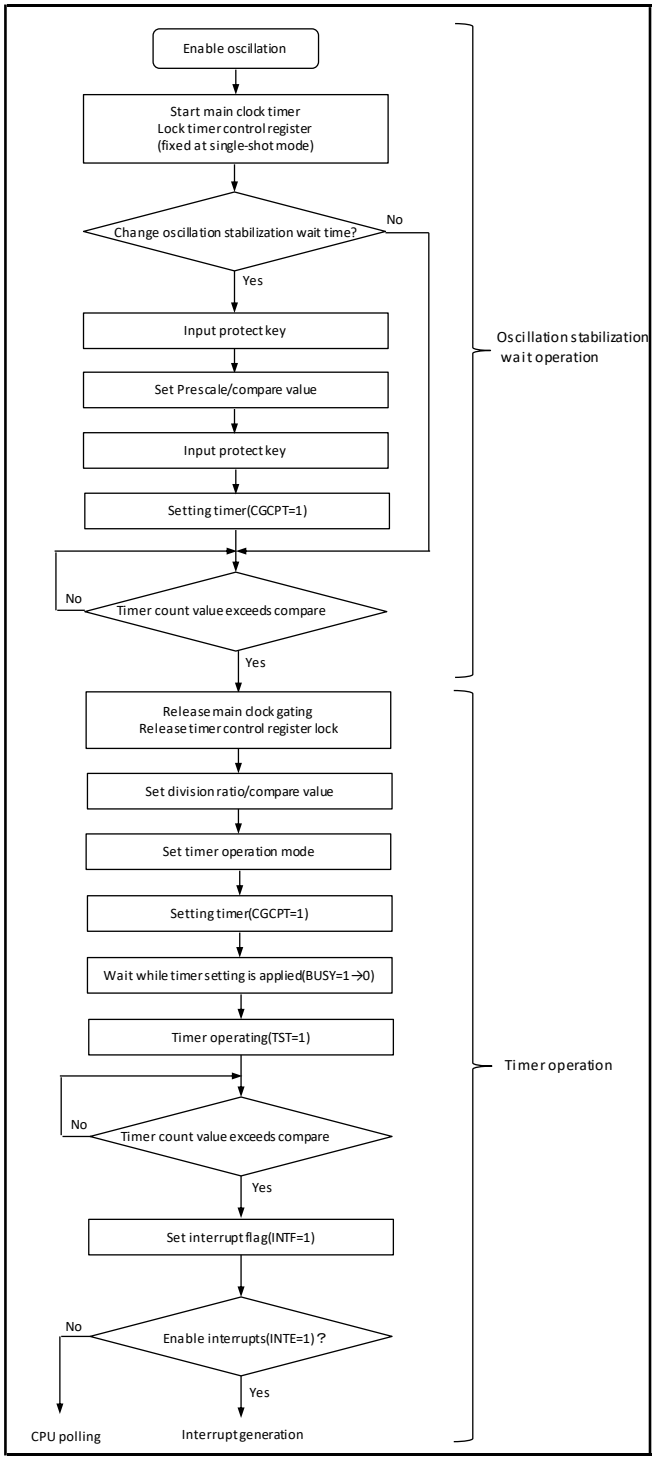
Timer Operation

- After the clock oscillation stabilizes, the source clock timer can be used as an ordinary timer.
- This timer can generate an interrupt within a certain period.

4. Setting Procedure Example

This section shows an example of the source clock timer setting procedure.

Figure 4-1 Example of Main Clock Timer Setting Procedure



5. Registers

This section lists the source clock timer registers.

List of High-Speed CR Clock Timer Registers

Table 5-1 List of High-Speed CR Clock Timer Registers

Abbreviated Register Name	Register Name	Reference
SYSC_FCRCTTRGR	High-speed CR clock timer trigger register	5.1
SYSC_FCRCTCNTR	High-speed CR clock timer control register	5.2
SYSC_FCRCTCPR	High-speed CR clock timer compare prescaler register	5.3
SYSC_FCRCTSTR	High-speed CR clock timer status register	5.4
SYSC_FCRCTINTER	High-speed CR clock timer interrupt enable register	5.5
SYSC_FCRCTICLR	High-speed CR clock timer interrupt clear register	5.6

List of Low-Speed CR Clock Timer Registers

Table 5-2 List of Low-Speed CR Clock Timer Registers

Abbreviated Register Name	Register Name	Reference
SYSC_SCRCTTRGR	Low-speed CR clock timer trigger register	5.7
SYSC_SCRCTCNTR	Low-speed CR clock timer control register	5.8
SYSC_SCRCTCPR	Low-speed CR clock timer compare prescaler register	5.9
SYSC_SCRCTSTR	Low-speed CR clock timer status register	5.10
SYSC_SCRCTINTER	Low-speed CR clock timer interrupt enable register	5.11
SYSC_SCRCTICLR	Low-speed CR clock timer interrupt clear register	5.12

List of Main Clock Timer Registers

Table 5-3 List of Main Clock Timer Registers

Abbreviated Register Name	Register Name	Reference
SYSC_MOCTTRGR	Main clock timer trigger register	5.13
SYSC_MOCTCNTR	Main clock timer control register	5.14
SYSC_MOCTCPR	Main clock timer compare prescaler register	5.15
SYSC_MOCTSTR	Main clock timer status register	5.16
SYSC_MOCTINTER	Main clock timer interrupt enable register	5.17
SYSC_MOCTICLR	Main clock timer interrupt clear register	5.18

List of Sub Clock Timer Registers

Table 5-4 List of Sub Clock Timer Registers

Abbreviated Register Name	Register Name	Reference
SYSC_SOCTTRGR	Sub clock timer trigger register	5.19
SYSC_SOCTCNTR	Sub clock timer control register	5.20
SYSC_SOCTCPR	Sub clock timer compare prescaler register	5.21
SYSC_SOCTSTR	Sub clock timer status register	5.22
SYSC_SOCTINTER	Sub clock timer interrupt enable register	5.23
SYSC_SOCTICLR	Sub clock timer interrupt clear register	5.24

Table 5-5 High-Speed CR Clock Timer Register Map

Offset	Register Name/Initial Value
0x0000_0000	SYSC_FCRCTTRGR 00000000_00000000_00000000_00000000
0x0000_0004	SYSC_FCRCTCNTR 00000000_00000000_00000000_00000000
0x0000_0008	SYSC_FCRCTCPR 00000000_00000110_00000000_00011110
0x0000_000C	SYSC_FCRCTSTR 00000000_00000000_00000000_00000000
0x0000_0010	SYSC_FCRCTINTER 00000000_00000000_00000000_00000000
0x0000_0014	SYSC_FCRCTICLR 00000000_00000000_00000000_00000000

Table 5-6 Low-Speed CR Clock Timer Register Map

Offset	Register Name/Initial Value
0x0000_0000	SYSC_SCRCTTRGR 00000000_00000000_00000000_00000000
0x0000_0004	SYSC_SCRCTCNTR 00000000_00000000_00000000_00000000
0x0000_0008	SYSC_SCRCTCPR 00000000_00000110_00000000_00000001
0x0000_000C	SYSC_SCRCTSTR 00000000_00000000_00000000_00000000
0x0000_0010	SYSC_SCRCTINTER 00000000_00000000_00000000_00000000
0x0000_0014	SYSC_SCRCTICLR 00000000_00000000_00000000_00000000

Table 5-7 Main Clock Timer Register Map

Offset	Register Name/Initial Value
0x0000_0000	SYSC_MOCTTRGR 00000000_00000000_00000000_00000000
0x0000_0004	SYSC_MOCTCNTR 00000000_00000000_00000000_00000000
0x0000_0008	SYSC_MOCTCPR 00000000_00000110_00000010_00000000
0x0000_000C	SYSC_MOCTSTR 00000000_00000000_00000000_00000000
0x0000_0010	SYSC_MOCTINTER 00000000_00000000_00000000_00000000
0x0000_0014	SYSC_MOCTICLR 00000000_00000000_00000000_00000000

Table 5-8 Sub Clock Timer Register Map

Offset	Register Name/Initial Value
0x0000_0000	SYSC_SOCTTRGR 00000000_00000000_00000000_00000000
0x0000_0004	SYSC_SOCTCNTR 00000000_00000000_00000000_00000000
0x0000_0008	SYSC_SOCTCPR 00000000_00000110_00000000_00000100
0x0000_000C	SYSC_SOCTSTR 00000000_00000000_00000000_00000000
0x0000_0010	SYSC_SOCTINTER 00000000_00000000_00000000_00000000
0x0000_0014	SYSC_SOCTICLR 00000000_00000000_00000000_00000000

The following tables list the initial values of the compare prescaler registers of the main clock timer and sub clock timer.

Main Oscillation Stabilization Wait Time (at 4 MHz)	Initial Value of SYSC_MOCTCPR
8.19 [ms]	00000000_00000110_00000010_00000000

Sub Oscillation Stabilization Wait Time (at 32 kHz)	Initial Value of SYSC_SOCTCPR
8.0 [ms]	00000000_00000110_00000000_00000100
16.0 [ms]	00000000_00000110_00000000_00001000

Notes:

- *The registers are protected by protection key setting register (SYSC0_PROTKEYR). For details on the protection, see the following chapter: "Low-power Consumption."*
- *Access to reserved address results bus error response.*

5.1. High-Speed CR Clock Timer Trigger Register (SYSC_FCRCTTRGR)

The high-speed CR clock timer trigger register (SYSC_FCRCTTRGR) is used to clear the count value of the timer, stop counting, and change the timer settings.

Bit	31	8
Field	Reserved	
R/W Attribute	R0,WX	
Protection Attribute	- (WPS)	
Initial Value	00000000_00000000_00000000	

bit	7	6	5	4	3	2	1	0
Field	Reserved					TCLR	CSTOP	CGCPT
R/W Attribute	R0,WX					R0,W	R0,W	R0,W
Protection Attribute	- (WPS)							
Initial Value	00000					0	0	0

The attribute during the oscillation stabilization wait operation is enclosed in parentheses ().

[bit31:3] Reserved: Reserved Bits

[bit2] TCLR: Timer Clear Bit

This bit clears the count value of the high-speed CR clock timer and stops the timer operation.

Bit	Description
0	No effect
1	Clear the timer count.

Note:

- *Setting the timer setting capture bit (CGCPT) to "1" at the same time as this bit is set to "1" clears the count value of the high-speed CR clock timer.*

[bit1] CSTOP: Count Stop Bit

This bit stops the counting of the high-speed CR clock timer.

Bit	Description
0	No effect
1	Stop the timer counting.

Note:

- *Setting the timer setting capture bit (CGCPT) to "1" at the same time as this bit is set to "1" stops the high-speed CR clock timer.*

[bit0] CGCPT: Timer Setting Capture Bit

This bit is used to change the high-speed CR clock timer settings and to start the timer counting. Setting this bit to "1" causes the high-speed CR clock timer to acquire the changed setting contents.

Bit	Description
0	No effect
1	Change the timer settings. / Start the timer counting.

5.2. High-Speed CR Clock Timer Control Register (SYSC_FCRCTCNTR)

The high-speed CR clock timer control register (SYSC_FCRCTCNTR) selects the operation mode of the timer and sets whether the counter operates or stops during debugging.

Bit	31	8
Field	Reserved	
R/W Attribute	R0,WX	
Protection Attribute	-	
Initial Value	00000000_00000000_00000000	

bit	7	6	5	4	3	2	1	0
Field	Reserved						DBGEN	MODE
R/W Attribute	R0,WX						R/W	R/W
Protection Attribute	-							
Initial Value	000000						0	0

[bit31:2] Reserved: Reserved Bits

[bit1] DBGEN: Debug Enable Bit

This bit sets the operation of the high-speed CR clock timer counter during debugging.

Bit	Description
0	Continue the count operation at the breakpoint.
1	Stop the count operation at the breakpoint.

Note:

- *This bit cannot be written during the oscillation stabilization wait operation. If writing is attempted, a bus error is returned.*

[bit0] MODE: Mode Control Bit

This bit selects the operation mode of the high-speed CR clock timer.

Single-shot mode clears the count and stops the count operation when the timer counter value exceeds the compare value (CMPR[15:0]).

Continuous mode clears the count and restarts the count operation when the timer counter value exceeds the compare value (CMPR[15:0]).

Bit	Description
0	Single-shot mode
1	Continuous mode

Notes:

- *Setting the timer setting capture bit (CGCPT) to "1" after this bit is set changes the operation mode of the high-speed CR clock timer.*
- *This bit cannot be written during the oscillation stabilization wait operation. If writing is attempted, a bus error is returned.*

5.3. High-Speed CR Clock Timer Compare Prescaler Register (SYSC_FCRCTCPR)

The high-speed CR clock timer compare prescaler register (SYSC_FCRCTCPR) selects the division ratio of the input clock and sets the compare value of the timer.

Bit	31	24
Field	Reserved	
R/W Attribute	R0,WX	
Protection Attribute	- (WPS)	
Initial Value	00000000	

bit	23	22	21	20	19	18	17	16
Field	Reserved				PSCL[3:0]			
R/W Attribute	R0,WX				R/W			
Protection Attribute	- (WPS)							
Initial Value	0000				0110			

bit	15	0
Field	CMPR[15:0]	
R/W Attribute	R/W	
Protection Attribute	- (WPS)	
Initial Value	00000000_00011110	

The attribute during the oscillation stabilization wait operation is enclosed in parentheses ().

[bit31:20] Reserved: Reserved Bits

[bit19:16] PSCL[3:0]: Prescale Bits

These bits select the division ratio of the input clock of the high-speed CR clock timer.

Bits				Description
0	0	0	0	No division (divided by 1)
0	0	0	1	Divided by 2
0	0	1	0	Divided by 4
0	0	1	1	Divided by 8
0	1	0	0	Divided by 16
0	1	0	1	Divided by 32
0	1	1	0	Divided by 64
0	1	1	1	Divided by 128
1	0	0	0	Divided by 256
1	0	0	1	Divided by 512
1	0	1	0	Divided by 1024
1	0	1	1	Divided by 2048
1	1	0	0	Divided by 4096
1	1	0	1	Divided by 8192
1	1	1	0	Divided by 16384
1	1	1	1	Divided by 32768

Note:

- *Setting the timer setting capture bit (CGCPT) to "1" after these bits are set changes the division ratio of the high-speed CR clock timer.*

[bit15:0] CMPR[15:0]: Compare Value Bits

These bits set the compare value of the high-speed CR clock timer.

The interrupt flag bit (INTF) of the high-speed CR clock timer is set to "1" when the count value of the high-speed CR clock timer exceeds the compare value.

Note:

- *Setting the timer setting capture bit (CGCPT) to "1" after these bits are set changes the compare value of the high-speed CR clock timer.*

Oscillation stabilization wait time of the high-speed CR clock

Prescale PSCL[3:0]	Compare Value CMPR[15:0]	Oscillation Stabilization Wait Time (High-Speed CR Clock Cycle x Prescale Value x Compare Value)
0110 (divided by 64)	0x001E(30)	Product specification

5.4. High-Speed CR Clock Timer Status Register (SYSC_FCRCTSTR)

The high-speed CR clock timer status register (SYSC_FCRCTSTR) indicates the update status of a timer reset, the operation status of the timer, and an interrupt flag.

bit	31							8
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	-							
Initial Value	00000000_00000000_00000000							

bit	7	6	5	4	3	2	1	0
Field	Reserved					BUSY	TST	INTF
R/W Attribute	R0,WX					R,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	00000					0	0	0

[bit31:3] Reserved: Reserved Bits

[bit2] BUSY: Setting Update Status Bit

This bit indicates the timer setting update status of the high-speed CR clock timer.

If the timer setting capture bit (CGCPT) is set to "1", this setting value remains "1" until the timer setting update is completed.

Bit	Description
0	Setting completed
1	Setting update in progress

Note:

- *Registers other than the interrupt enable register and interrupt clear register of the high-speed CR clock timer must not be changed when the read value of this bit is "1".*

[bit1] TST: Timer Status Bit

This bit indicates the operation status of the high-speed CR clock timer.

Bit	Description
0	Timer stopped
1	Timer operating

[bit0] INTF: Interrupt Flag Bit

This bit is the interrupt flag bit of the high-speed CR clock timer.

"1" is set when the high-speed CR clock timer value exceeds the set value in the compare bits (CMPR[15:0]). At this time, if the interrupt enable bit (INTE) is "1", the high-speed CR clock timer generates an interrupt request.

To clear the interrupt flag bit to "0", set the interrupt clear bit (INTC) to "1".

Bit	Description
0	No interrupt factor is detected.
1	An interrupt factor is detected.

Note:

- *This bit can be set only while the timer is operating. It cannot be set during the oscillation stabilization wait operation.*

5.5. High-Speed CR Clock Timer Interrupt Enable Register (SYSC_FCRCTINTER)

The high-speed CR clock timer interrupt enable register (SYSC_FCRCTINTER) sets whether to enable or disable interrupts of the timer.

bit	31	8
Field	Reserved	
R/W Attribute	R0,WX	
Protection Attribute	- (WPS)	
Initial Value	00000000_00000000_00000000	

bit	7	6	5	4	3	2	1	0
Field	Reserved							INTE
R/W Attribute	R0,WX							R/W
Protection Attribute	- (WPS)							
Initial Value	0000000							0

The attribute during the oscillation stabilization wait operation is enclosed in parentheses ().

[bit31:1] Reserved: Reserved Bits

[bit0] INTE: Interrupt Enable Bit

This bit sets whether to enable or disable interrupts of the high-speed CR clock timer.

Bit	Description
0	Disable interrupts.
1	Enable interrupts.

5.6. High-Speed CR Clock Timer Interrupt Clear Register (SYSC_FCRCTICLR)

The high-speed CR clock timer interrupt clear register (SYSC_FCRCTICLR) clears an interrupt flag to "0".

bit	31	8
Field	Reserved	
R/W Attribute	R0,WX	
Protection Attribute	- (WPS)	
Initial Value	00000000_00000000_00000000	

bit	7	6	5	4	3	2	1	0
Field	Reserved							INTC
R/W Attribute	R0,WX							R0,W
Protection Attribute	- (WPS)							
Initial Value	0000000							0

The attribute during the oscillation stabilization wait operation is enclosed in parentheses ().

[bit31:1] Reserved: Reserved Bits

[bit0] INTC: Interrupt Clear Bit

This bit clears the interrupt flag bit (INTF) of the high-speed CR clock timer to "0".

Bit	Description
0	No effect
1	Clear the interrupt flag bit (INTF) to "0".

5.7. Low-Speed CR Clock Timer Trigger Register (SYSC_SCRCTTRGR)

The low-speed CR clock timer trigger register (SYSC_SCRCTTRGR) is used to clear the count value of the timer, stop counting, and change the timer settings.

bit	31	8
Field	Reserved	
R/W Attribute	R0,WX	
Protection Attribute	- (WPS)	
Initial Value	00000000_00000000_00000000	

bit	7	6	5	4	3	2	1	0
Field	Reserved					TCLR	CSTOP	CGCPT
R/W Attribute	R0,WX					R0,W	R0,W	R0,W
Protection Attribute	- (WPS)							
Initial Value	00000					0	0	0

The attribute during the oscillation stabilization wait operation is enclosed in parentheses ().

[bit31:3] Reserved: Reserved Bits

[bit2] TCLR: Timer Clear Bit

This bit clears the count value of the low-speed CR clock timer and stops the timer operation.

Bit	Description
0	No effect
1	Clear the timer count.

Note:

- *Setting the timer setting capture bit (CGCPT) to "1" at the same time as this bit is set to "1" clears the count value of the low-speed CR clock timer.*

[bit1] CSTOP: Count Stop Bit

This bit stops the counting of the low-speed CR clock timer.

Bit	Description
0	No effect
1	Stop the timer counting.

Note:

- *Setting the timer setting capture bit (CGCPT) to "1" at the same time as this bit is set to "1" stops the low-speed CR clock timer.*

[bit0] CGCPT: Timer Setting Capture Bit

This bit is used to change the low-speed CR clock timer settings and to start the timer counting. Setting this bit to "1" causes the low-speed CR clock timer to acquire the changed setting contents.

Bit	Description
0	No effect
1	Change the timer settings. / Start the timer counting.

5.8. Low-Speed CR Clock Timer Control Register (SYSC_SCRCTCNTR)

The low-speed CR clock timer control register (SYSC_SCRCTCNTR) selects the operation mode of the timer and sets whether the counter operates or stops during debugging.

bit	31	8
Field	Reserved	
R/W Attribute	R0,WX	
Protection Attribute	-	
Initial Value	00000000_00000000_00000000	

bit	7	6	5	4	3	2	1	0
Field	Reserved						DBGEN	MODE
R/W Attribute	R0,WX						R/W	R/W
Protection Attribute	-							
Initial Value	000000						0	0

[bit31:2] Reserved: Reserved Bits

[bit1] DBGEN: Debug Enable Bit

This bit sets the operation of the low-speed CR clock timer counter during debugging.

Bit	Description
0	Continue the count operation at the breakpoint.
1	Stop the count operation at the breakpoint.

Note:

- *This bit cannot be written during the oscillation stabilization wait operation. If writing is attempted, a bus error is returned.*

[bit0] MODE: Mode Control Bit

This bit selects the operation mode of the low-speed CR clock timer.

Single-shot mode clears the count and stops the count operation when the timer counter value exceeds the compare value (CMPR[15:0]).

Continuous mode clears the count and restarts the count operation when the timer counter value exceeds the compare value (CMPR[15:0]).

Bit	Description
0	Single-shot mode
1	Continuous mode

Notes:

- *Setting the timer setting capture bit (CGCPT) to "1" after this bit is set changes the operation mode of the low-speed CR clock timer.*
- *This bit cannot be written during the oscillation stabilization wait operation. If writing is attempted, a bus error is returned.*

5.9. Low-Speed CR Clock Timer Compare Prescaler Register (SYSC_SCRCTCPR)

The low-speed CR clock timer compare prescaler register (SYSC_SCRCTCPR) selects the division ratio of the input clock and sets the compare value of the timer.

bit	31	24
Field	Reserved	
R/W Attribute	R0,WX	
Protection Attribute	- (WPS)	
Initial Value	00000000	

bit	23	22	21	20	19	18	17	16
Field	Reserved				PSCL[3:0]			
R/W Attribute	R0,WX				R/W			
Protection Attribute	- (WPS)							
Initial Value	0000				0110			

bit	15	0
Field	CMPR[15:0]	
R/W Attribute	R/W	
Protection Attribute	- (WPS)	
Initial Value	00000000_00000001	

The attribute during the oscillation stabilization wait operation is enclosed in parentheses ().

[bit31:20] Reserved: Reserved Bits

[bit19:16] PSCL[3:0]: Prescale Bits

These bits select the division ratio of the input clock of the low-speed CR clock timer.

Bits				Description
0	0	0	0	No division (divided by 1)
0	0	0	1	Divided by 2
0	0	1	0	Divided by 4
0	0	1	1	Divided by 8
0	1	0	0	Divided by 16
0	1	0	1	Divided by 32
0	1	1	0	Divided by 64
0	1	1	1	Divided by 128
1	0	0	0	Divided by 256
1	0	0	1	Divided by 512
1	0	1	0	Divided by 1024
1	0	1	1	Divided by 2048
1	1	0	0	Divided by 4096
1	1	0	1	Divided by 8192
1	1	1	0	Divided by 16384
1	1	1	1	Divided by 32768

Note:

- *Setting the timer setting capture bit (CGCPT) to "1" after these bits are set changes the division ratio of the low-speed CR clock timer.*

[bit15:0] CMPR[15:0]: Compare Value Bits

These bits set the compare value of the low-speed CR clock timer.

The interrupt flag bit (INTF) is set to "1" when the count value of the low-speed CR clock timer exceeds the compare value.

Note:

- *Setting the timer setting capture bit (CGCPT) to "1" after these bits are set changes the compare value of the low-speed CR clock timer.*

Oscillation stabilization wait time of the low-speed CR clock

Prescale PSCL[3:0]	Compare Value CMPR[15:0]	Oscillation Stabilization Wait Time (Low-Speed CR Clock Cycle x Prescale Value x Compare Value)
0110 (divided by 64)	0x0001(1)	Product specification

5.10. Low-Speed CR Clock Timer Status Register (SYSC_SCRCTSTR)

The low-speed CR clock timer status register (SYSC_SCRCTSTR) indicates the update status of a timer reset, the operation status of the timer, and an interrupt flag.

bit	31							8
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	-							
Initial Value	00000000_00000000_00000000							

bit	7	6	5	4	3	2	1	0
Field	Reserved					BUSY	TST	INTF
R/W Attribute	R0,WX					R,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	00000					0	0	0

[bit31:3] Reserved: Reserved Bits

[bit2] BUSY: Setting Update Status Bit

This bit indicates the setting update status of the low-speed CR clock timer.

If the timer setting capture bit (CGCPT) is set to "1", this setting value remains "1" until the timer setting update is completed.

Bit	Description
0	Setting completed
1	Setting update in progress

Note:

- *Registers other than the interrupt enable register and interrupt clear register of the low-speed CR clock timer must not be changed when the read value of this bit is "1".*

[bit1] TST: Timer Status Bit

This bit indicates the operation status of the low-speed CR clock timer.

Bit	Description
0	Timer stopped
1	Timer operating

[bit0] INTF: Interrupt Flag Bit

This bit is the interrupt flag bit of the low-speed CR clock timer.

"1" is set when the low-speed CR clock timer value exceeds the set value in the compare bits (CMPR[15:0]). At this time, if the interrupt enable bit (INTE) is "1", the low-speed CR clock timer generates an interrupt request.

To clear the interrupt flag bit to "0", set the interrupt clear bit (INTC) to "1".

Bit	Description
0	No interrupt factor is detected.
1	An interrupt factor is detected.

Note:

- *This bit can be set only while the timer is operating. It cannot be set during the oscillation stabilization wait operation.*

5.11. Low-Speed CR Clock Timer Interrupt Enable Register (SYSC_SCRCTINTER)

The low-speed CR clock timer interrupt enable register (SYSC_SCRCTINTER) sets whether to enable or disable interrupts of the timer.

bit	31	8
Field	Reserved	
R/W Attribute	R0,WX	
Protection Attribute	- (WPS)	
Initial Value	00000000_00000000_00000000	

bit	7	6	5	4	3	2	1	0
Field	Reserved							INTE
R/W Attribute	R0,WX							R/W
Protection Attribute	- (WPS)							
Initial Value	0000000							0

The attribute during the oscillation stabilization wait operation is enclosed in parentheses ().

[bit31:1] Reserved: Reserved Bits

[bit0] INTE: Interrupt Enable Bit

This bit sets whether to enable or disable interrupts of the low-speed CR clock timer.

Bit	Description
0	Disable interrupts.
1	Enable interrupts.

5.12. Low-Speed CR Clock Timer Interrupt Clear Register (SYSC_SCRCTICLR)

The low-speed CR clock timer interrupt clear register (SYSC_SCRCTICLR) clears an interrupt flag to "0".

bit	31	8
Field	Reserved	
R/W Attribute	R0,WX	
Protection Attribute	- (WPS)	
Initial Value	00000000_00000000_00000000	

bit	7	6	5	4	3	2	1	0
Field	Reserved							INTC
R/W Attribute	R0,WX							R0,W
Protection Attribute	- (WPS)							
Initial Value	0000000							0

The attribute during the oscillation stabilization wait operation is enclosed in parentheses ().

[bit31:1] Reserved: Reserved Bits

[bit0] INTC: Interrupt Clear Bit

This bit clears the interrupt flag bit (INTF) of the low-speed CR clock timer to "0".

Bit	Description
0	No effect
1	Clear the interrupt flag bit (INTF) to "0".

5.13. Main Clock Timer Trigger Register (SYSC_MOCTTRGR)

The main clock timer trigger register (SYSC_MOCTTRGR) is used to clear the count value of the timer, stop counting, and change the timer settings.

bit	31	8
Field	Reserved	
R/W Attribute	R0,WX	
Protection Attribute	- (WPS)	
Initial Value	00000000_00000000_00000000	

bit	7	6	5	4	3	2	1	0
Field	Reserved					TCLR	CSTOP	CGCPT
R/W Attribute	R0,WX					R0,W	R0,W	R0,W
Protection Attribute	- (WPS)							
Initial Value	00000					0	0	0

The attribute during the oscillation stabilization wait operation is enclosed in parentheses ().

[bit31:3] Reserved: Reserved Bits

[bit2] TCLR: Timer Clear Bit

This bit clears the count value of the main clock timer and stops the timer operation.

Bit	Description
0	No effect
1	Clear the timer count.

Note:

- *Setting the timer setting capture bit (CGCPT) to "1" at the same time as this bit is set to "1" clears the count value of the main clock timer.*

[bit1] CSTOP: Count Stop Bit

This bit stops the counting of the main clock timer.

Bit	Description
0	No effect
1	Stop the timer counting.

Note:

- *Setting the timer setting capture bit (CGCPT) to "1" at the same time as this bit is set to "1" stops the main clock timer.*

[bit0] CGCPT: Timer Setting Capture Bit

This bit is used to change the main clock timer settings and to start the timer counting. Setting this bit to "1" causes the main clock timer to acquire the changed setting contents.

Bit	Description
0	No effect
1	Change the timer settings. / Start the timer counting.

5.14. Main Clock Timer Control Register (SYSC_MOCTCNTR)

The main clock timer control register (SYSC_MOCTCNTR) selects the operation mode of the timer and sets whether the counter operates or stops during debugging.

bit	31	8
Field	Reserved	
R/W Attribute	R0,WX	
Protection Attribute	-	
Initial Value	00000000_00000000_00000000	

bit	7	6	5	4	3	2	1	0
Field	Reserved						DBGEN	MODE
R/W Attribute	R0,WX						R/W	R/W
Protection Attribute	-							
Initial Value	000000						0	0

[bit31:2] Reserved: Reserved Bits

[bit1] DBGEN: Debug Enable Bit

This bit sets the operation of the main clock timer counter during debugging.

Bit	Description
0	Continue the count operation at the breakpoint.
1	Stop the count operation at the breakpoint.

Note:

- *This bit cannot be written during the oscillation stabilization wait operation. If writing is attempted, a bus error is returned.*

[bit0] MODE: Mode Control Bit

This bit selects the operation mode of the main clock timer.

Single-shot mode clears the count and stops the count operation when the timer counter value exceeds the compare value (CMPR[15:0]).

Continuous mode clears the count and restarts the count operation when the timer counter value exceeds the compare value (CMPR[15:0]).

Bit	Description
0	Single-shot mode
1	Continuous mode

Notes:

- *Setting the timer setting capture bit (CGCPT) to "1" after this bit is set changes the operation mode of the main clock timer.*
- *This bit cannot be written during the oscillation stabilization wait operation. If writing is attempted, a bus error is returned.*

5.15. Main Clock Timer Compare Prescaler Register (SYSC_MOCTCPR)

The main clock timer compare prescaler register (SYSC_MOCTCPR) selects the division ratio of the input clock and sets the compare value of the timer.

bit	31	24
Field	Reserved	
R/W Attribute	R0,WX	
Protection Attribute	- (WPS)	
Initial Value	00000000	

bit	23	22	21	20	19	18	17	16
Field	Reserved				PSCL[3:0]			
R/W Attribute	R0,WX				R/W			
Protection Attribute	- (WPS)							
Initial Value	0000				0110			

bit	15	0
Field	CMPR[15:0]	
R/W Attribute	R/W	
Protection Attribute	- (WPS)	
Initial Value	00000010_00000000	

The attribute during the oscillation stabilization wait operation is enclosed in parentheses ().

[bit31:20] Reserved: Reserved Bits

[bit19:16] PSCL[3:0]: Prescale Bits

These bits select the division ratio of the input clock of the main clock timer.

Bits				Description
0	0	0	0	No division (divided by 1)
0	0	0	1	Divided by 2
0	0	1	0	Divided by 4
0	0	1	1	Divided by 8
0	1	0	0	Divided by 16
0	1	0	1	Divided by 32
0	1	1	0	Divided by 64
0	1	1	1	Divided by 128
1	0	0	0	Divided by 256
1	0	0	1	Divided by 512
1	0	1	0	Divided by 1024
1	0	1	1	Divided by 2048
1	1	0	0	Divided by 4096
1	1	0	1	Divided by 8192
1	1	1	0	Divided by 16384
1	1	1	1	Divided by 32768

Note:

- *Setting the timer setting capture bit (CGCPT) to "1" after these bits are set changes the division ratio of the main clock timer.*

[bit15:0] CMPR[15:0]: Compare Value Bits

These bits set the compare value of the main clock timer.

The interrupt flag bit (INTF) is set to "1" when the count value of the main clock timer exceeds the compare value.

Note:

- *Setting the timer setting capture bit (CGCPT) to "1" after these bits are set changes the compare value of the main clock timer.*

Oscillation stabilization wait time of the main clock

Prescale PSCL[3:0]	Compare Value CMPR[15:0]	Oscillation Stabilization Wait Time (Main Clock Cycle x Prescale Value x Compare Value)
0110 (divided by 64)	0x0200(512)	8.19 [ms]

5.16. Main Clock Timer Status Register (SYSC_MOCTSTR)

The main clock timer status register (SYSC_MOCTSTR) indicates the update status of a timer reset, the operation status of the timer, and an interrupt flag.

bit	31	8
Field	Reserved	
R/W Attribute	R0,WX	
Protection Attribute	-	
Initial Value	00000000_00000000_00000000	

bit	7	6	5	4	3	2	1	0
Field	Reserved					BUSY	TST	INTF
R/W Attribute	R0,WX					R,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	00000					0	0	0

[bit31:3] Reserved: Reserved Bits

[bit2] BUSY: Setting Update Status Bit

This bit indicates the setting update status of the main clock timer.

If the timer setting capture bit (CGCPT) is set to "1", this setting value remains "1" until the timer setting update is completed.

Bit	Description
0	Setting completed
1	Setting update in progress

Note:

- *Registers other than the interrupt enable register and interrupt clear register of the main clock timer must not be changed when the read value of this bit is "1".*

[bit1] TST: Timer Status Bit

This bit indicates the operation status of the main clock timer.

Bit	Description
0	Timer stopped
1	Timer operating

[bit0] INTF: Interrupt Flag Bit

This bit is the interrupt flag bit of the main clock timer.

"1" is set when the main clock timer value exceeds the set value in the compare bits (CMPR[15:0]). At this time, if the interrupt enable bit (INTE) is "1", the main clock timer generates an interrupt request.

To clear the interrupt flag bit to "0", set the interrupt clear bit (INTC) to "1".

Bit	Description
0	No interrupt factor is detected.
1	An interrupt factor is detected.

Note:

- *This bit can be set only while the timer is operating. It cannot be set during the oscillation stabilization wait operation.*

5.17. Main Clock Timer Interrupt Enable Register (SYSC_MOCTINTER)

The main clock timer interrupt enable register (SYSC_MOCTINTER) sets whether to enable or disable interrupts of the timer.

bit	31	8
Field	Reserved	
R/W Attribute	R0,WX	
Protection Attribute	- (WPS)	
Initial Value	00000000_00000000_00000000	

bit	7	6	5	4	3	2	1	0
Field	Reserved							INTE
R/W Attribute	R0,WX							R/W
Protection Attribute	- (WPS)							
Initial Value	0000000							0

The attribute during the oscillation stabilization wait operation is enclosed in parentheses ().

[bit31:1] Reserved: Reserved Bits

[bit0] INTE: Interrupt Enable Bit

This bit sets whether to enable or disable interrupts of the main clock timer.

Bit	Description
0	Disable interrupts.
1	Enable interrupts.

5.18. Main Clock Timer Interrupt Clear Register (SYSC_MOCTICLR)

The main clock timer interrupt clear register (SYSC_MOCTICLR) clears an interrupt flag.

bit	31	8
Field	Reserved	
R/W Attribute	R0,WX	
Protection Attribute	- (WPS)	
Initial Value	00000000_00000000_00000000	

bit	7	6	5	4	3	2	1	0
Field	Reserved							INTC
R/W Attribute	R0,WX							R0,W
Protection Attribute	- (WPS)							
Initial Value	0000000							0

The attribute during the oscillation stabilization wait operation is enclosed in parentheses ().

[bit31:1] Reserved: Reserved Bits

[bit0] INTC: Interrupt Clear Bit

This bit clears the interrupt flag bit (INTF) of the main clock timer to "0".

Bit	Description
0	No effect
1	Clear the interrupt flag bit (INTF) to "0".

5.19. Sub Clock Timer Trigger Register (SYSC_SOCTTRGR)

The sub clock timer trigger register (SYSC_SOCTTRGR) is used to clear the count value of the timer, stop counting, and change the timer settings.

bit	31	8
Field	Reserved	
R/W Attribute	R0,WX	
Protection Attribute	- (WPS)	
Initial Value	00000000_00000000_00000000	

bit	7	6	5	4	3	2	1	0
Field	Reserved					TCLR	CSTOP	CGCPT
R/W Attribute	R0,WX					R0,W	R0,W	R0,W
Protection Attribute	- (WPS)							
Initial Value	00000					0	0	0

The attribute during the oscillation stabilization wait operation is enclosed in parentheses ().

[bit31:3] Reserved: Reserved Bits

[bit2] TCLR: Timer Clear Bit

This bit clears the count value of the sub clock timer and stops the timer operation.

Bit	Description
0	No effect
1	Clear the timer count.

Note:

- *Setting the timer setting capture bit (CGCPT) to "1" at the same time as this bit is set to "1" clears the count value of the main clock timer.*

[bit1] CSTOP: Count Stop Bit

This bit stops the counting of the sub clock timer.

Bit	Description
0	No effect
1	Stop the timer counting.

Note:

- *Setting the timer setting capture bit (CGCPT) to "1" at the same time as this bit is set to "1" stops the main clock timer.*

[bit0] CGCPT: Timer Setting Capture Bit

This bit is used to change the sub clock timer settings and to start the timer counting. Setting this bit to "1" causes the sub clock timer to acquire the changed setting contents.

Bit	Description
0	No effect
1	Change the timer settings. / Start the timer counting.

5.20. Sub Clock Timer Control Register (SYSC_SOCTCNTR)

The sub clock timer control register (SYSC_SOCTCNTR) selects the operation mode of the timer and sets whether the counter operates or stops during debugging.

bit	31	8
Field	Reserved	
R/W Attribute	R0,WX	
Protection Attribute	-	
Initial Value	00000000_00000000_00000000	

bit	7	6	5	4	3	2	1	0
Field	Reserved						DBGEN	MODE
R/W Attribute	R0,WX						R/W	R/W
Protection Attribute	-							
Initial Value	000000						0	0

[bit31:2] Reserved: Reserved Bits

[bit1] DBGEN: Debug Enable Bit

This bit sets the operation of the sub clock timer counter during debugging.

Bit	Description
0	Continue the count operation at the breakpoint.
1	Stop the count operation at the breakpoint.

Note:

- *This bit cannot be written during the oscillation stabilization wait operation. If writing is attempted, a bus error is returned.*

[bit0] MODE: Mode Control Bit

This bit selects the operation mode of the sub clock timer.

Single-shot mode clears the count and stops the count operation when the timer counter value exceeds the compare value (CMPR[15:0]).

Continuous mode clears the count and restarts the count operation when the timer counter value exceeds the compare value (CMPR[15:0]).

Bit	Description
0	Single-shot mode
1	Continuous mode

Notes:

- *Setting the timer setting capture bit (CGCPT) to "1" after this bit is set changes the operation mode of the main clock timer.*
- *This bit cannot be written during the oscillation stabilization wait operation. If writing is attempted, a bus error is returned.*

5.21. Sub Clock Timer Compare Prescaler Register (SYSC_SOCTCPR)

The sub clock timer compare prescaler register (SYSC_SOCTCPR) selects the division ratio of the input clock and sets the compare value of the timer.

bit	31	24
Field	Reserved	
R/W Attribute	R0,WX	
Protection Attribute	- (WPS)	
Initial Value	00000000	

bit	23	22	21	20	19	18	17	16
Field	Reserved				PSCL[3:0]			
R/W Attribute	R0,WX				R/W			
Protection Attribute	- (WPS)							
Initial Value	0000				0110			

bit	15	0
Field	CMPR[15:0]	
R/W Attribute	R/W	
Protection Attribute	- (WPS)	
Initial Value	00000000_00000100	

The attribute during the oscillation stabilization wait operation is enclosed in parentheses ().

[bit31:20] Reserved: Reserved Bits

[bit19:16] PSCL[3:0]: Prescale Bits

These bits select the division ratio of the input clock of the sub clock timer.

Bits				Description
0	0	0	0	No division (divided by 1)
0	0	0	1	Divided by 2
0	0	1	0	Divided by 4
0	0	1	1	Divided by 8
0	1	0	0	Divided by 16
0	1	0	1	Divided by 32
0	1	1	0	Divided by 64
0	1	1	1	Divided by 128
1	0	0	0	Divided by 256
1	0	0	1	Divided by 512
1	0	1	0	Divided by 1024
1	0	1	1	Divided by 2048
1	1	0	0	Divided by 4096
1	1	0	1	Divided by 8192
1	1	1	0	Divided by 16384
1	1	1	1	Divided by 32768

Note:

- *Setting the timer setting capture bit (CGCPT) to "1" after these bits are set changes the division ratio of the main clock timer.*

[bit15:0] CMPR[15:0]: Compare Value Bits

These bits set the compare value of the sub clock timer.

The interrupt flag bit (INTF) is set to "1" when the count value of the sub clock timer exceeds the compare value.

Note:

- *Setting the timer setting capture bit (CGCPT) to "1" after these bits are set changes the compare value of the high-speed CR clock timer.*

Oscillation stabilization wait time of the sub clock

Prescale PSCL[3:0]	Compare Value CMPR[15:0]	Oscillation Stabilization Wait Time (Main Clock Cycle x Prescale Value x Compare Value)
0110 (divided by 64)	0x0004(4)	8.0[ms]

5.22. Sub Clock Timer Status Register (SYSC_SOCTSTR)

The sub clock timer status register (SYSC_SOCTSTR) indicates the update status of a timer reset, the operation status of the timer, and an interrupt flag.

bit	31							8
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	-							
Initial Value	00000000_00000000_00000000							

bit	7	6	5	4	3	2	1	0
Field	Reserved					BUSY	TST	INTF
R/W Attribute	R0,WX					R,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	00000					0	0	0

[bit31:3] Reserved: Reserved Bits

[bit2] BUSY: Setting Update Status Bit

This bit indicates the setting update status of the sub clock timer.

If the timer setting capture bit (CGCPT) is set to "1", this setting value remains "1" until the timer setting update is completed.

Bit	Description
0	Setting completed
1	Setting update in progress

Note:

- *Registers other than the interrupt enable register and interrupt clear register of the sub clock timer must not be changed when the read value of this bit is "1".*

[bit1] TST: Timer Status Bit

This bit indicates the operation status of the sub clock timer.

Bit	Description
0	Timer stopped
1	Timer operating

[bit0] INTF: Interrupt Flag Bit

This bit is the interrupt flag bit of the sub clock timer.

"1" is set when the sub clock timer value exceeds the set value in the compare bits (CMPR[15:0]). At this time, if the interrupt enable bit (INTE) is "1", the main clock timer generates an interrupt request.

To clear the interrupt flag bit to "0", set the interrupt clear bit (INTC) to "1".

Bit	Description
0	No interrupt factor is detected.
1	An interrupt factor is detected.

Note:

- *This bit can be set only while the timer is operating. It cannot be set during the oscillation stabilization wait operation.*

5.23. Sub Clock Timer Interrupt Enable Register (SYSC_SOCTINTER)

The sub clock timer interrupt enable register (SYSC_SOCTINTER) sets whether to enable or disable interrupts of the timer.

bit	31	8
Field	Reserved	
R/W Attribute	R0,WX	
Protection Attribute	- (WPS)	
Initial Value	00000000_00000000_00000000	

bit	7	6	5	4	3	2	1	0
Field	Reserved							INTE
R/W Attribute	R0,WX							R/W
Protection Attribute	- (WPS)							
Initial Value	0000000							0

The attribute during the oscillation stabilization wait operation is enclosed in parentheses ().

[bit31:1] Reserved: Reserved Bits

[bit0] INTE: Interrupt Enable Bit

This bit sets whether to enable or disable interrupts of the sub clock timer.

Bit	Description
0	Disable interrupts.
1	Enable interrupts.

5.24. Sub Clock Timer Interrupt Clear Register (SYSC_SOCTICLR)

The sub clock timer interrupt clear register (SYSC_SOCTICLR) clears an interrupt flag bit to "0".

bit	31	8
Field	Reserved	
R/W Attribute	R0,WX	
Protection Attribute	- (WPS)	
Initial Value	00000000_00000000_00000000	

bit	7	6	5	4	3	2	1	0
Field	Reserved							INTC
R/W Attribute	R0,WX							R0,W
Protection Attribute	- (WPS)							
Initial Value	0000000							0

The attribute during the oscillation stabilization wait operation is enclosed in parentheses ().

[bit31:1] Reserved: Reserved Bits

[bit0] INTC: Interrupt Clear Bit

This bit clears the interrupt flag bit (INTF) of the sub clock timer to "0".

Bit	Description
0	No effect
1	Clear the interrupt flag bit (INTF) to "0".

6. Precautions for Using this Device

This section explains precautions for using the source clock timer.

- When the oscillation stabilization wait time for each source clock changed, please set the value under standard value range. Refer to the datasheet for detail information for standard value of the oscillation stabilization wait time.

CHAPTER 10: Real Time Clock

This chapter explains the function and operation of the Real Time Clock.

.

1. Overview
2. Configuration and Block Diagram
3. Operation of the Real Time Clock
4. Registers
5. Usage Precaution

1. Overview

The Real Time Clock (RTC) consists of two modules, the Timer module and the Calibration module. This section lists the features of the Timer and Calibration modules.

The Real Time Clock (RTC) Timer module provides the current real time (HH/MM/SS) and the Calibration module calibrates Sub clock or Slow CR clock with respect to the Main oscillator clock.

Features of the RTC Timer Module

The features of the RTC Timer module are:

- It provides counting of real-time, comprising hour, minute and second counters
- It can operate in low-power mode
- Possible source clocks are:
  - Main oscillation (4 MHz)
  - Sub oscillation (32 kHz)
  - RC oscillation (to 100 kHz)
- It has a configurable Sub-second counter (RTC time base, normally half-second intervals) to calibrate the RTC for an applied source clock
- Precise source clock switching is synchronized with the Sub-second counter
- Interrupt sources are:
  - Half-second
  - One second
  - One minute
  - One hour
  - One day

Features of the Calibration Module

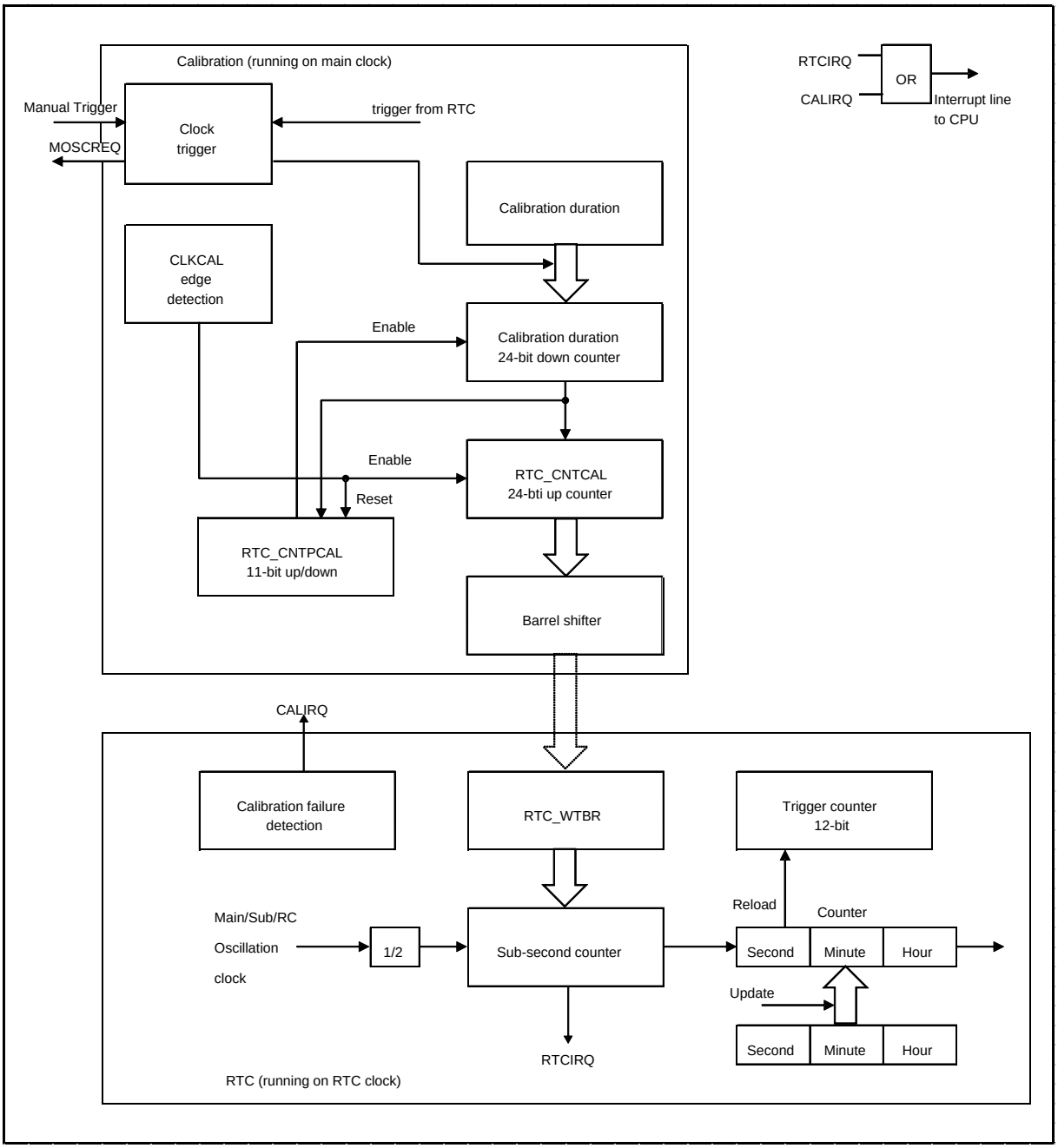
- The automatic cyclic calibration/measurement of Sub clock (CLK_SUB) or Slow CR clock (CLK_SRC) against Main clock (CLK_MAIN)
- The automatic correction of the RTC Sub-second counter value to keep accurate time
- Low-power modes due to Automatic wake-up/sleep of Main oscillation before/after Calibration phase
- An averaging method to prevent accumulation errors (Automatic modulation of Sub-second counter reload value)
- A configurable calibration interval for cyclic calibration (12-bit counter, counting seconds from RTC)
- A configurable calibration duration (i.e. 250 ms divided by 1/2/4/8/16)
- Automatic/Manual trigger for calibration
- Interrupt sources (to wake up the MCU from low-power mode):
  - Calibration done
  - Calibration failure (missing Main clock)
- Support for MCU debug mode

2. Configuration and Block Diagram

This section shows a block diagram of Real Time Clock.

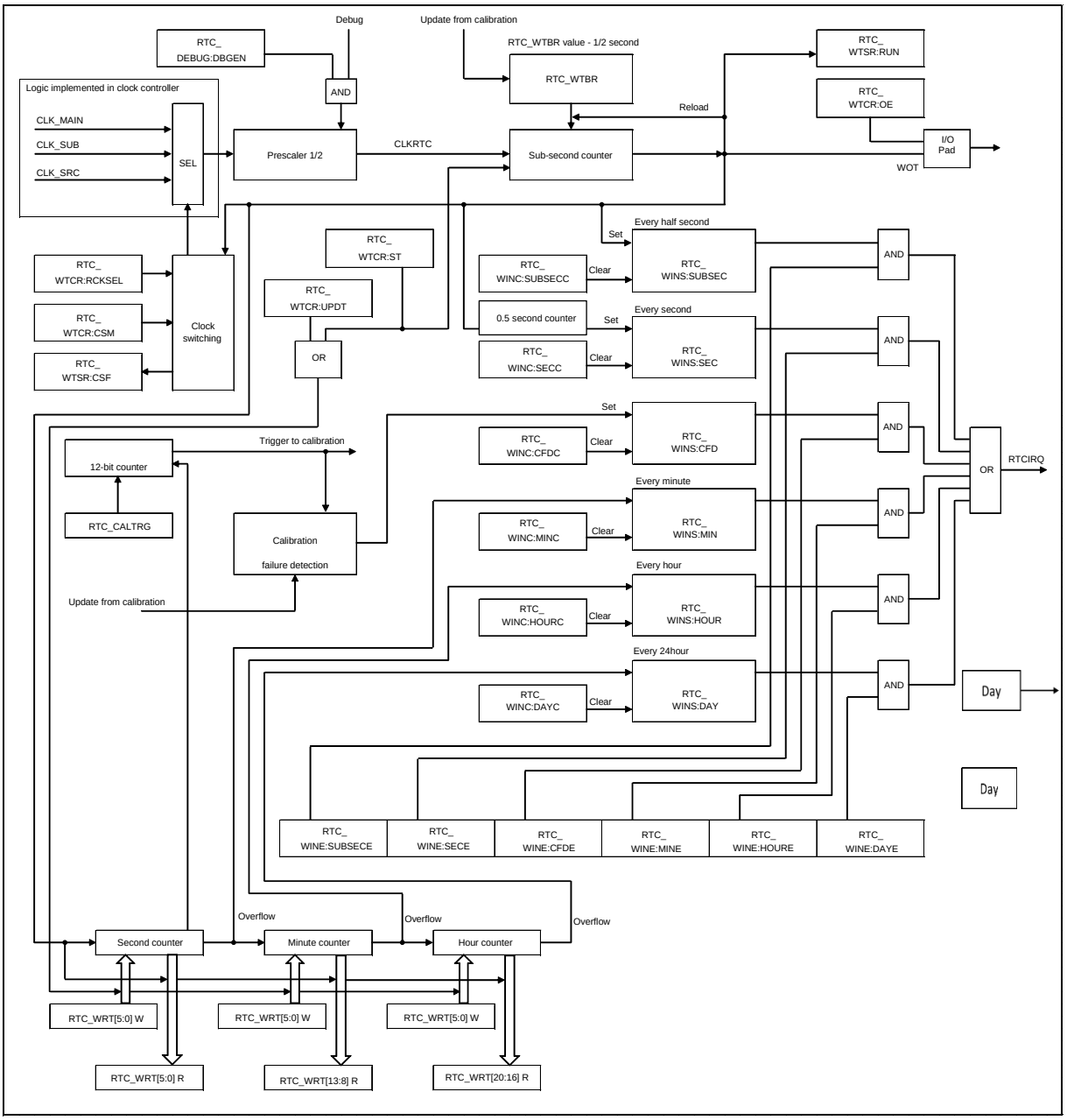
Block Diagram of Real Time Clock

Figure 2-1 Block Diagram of RTC



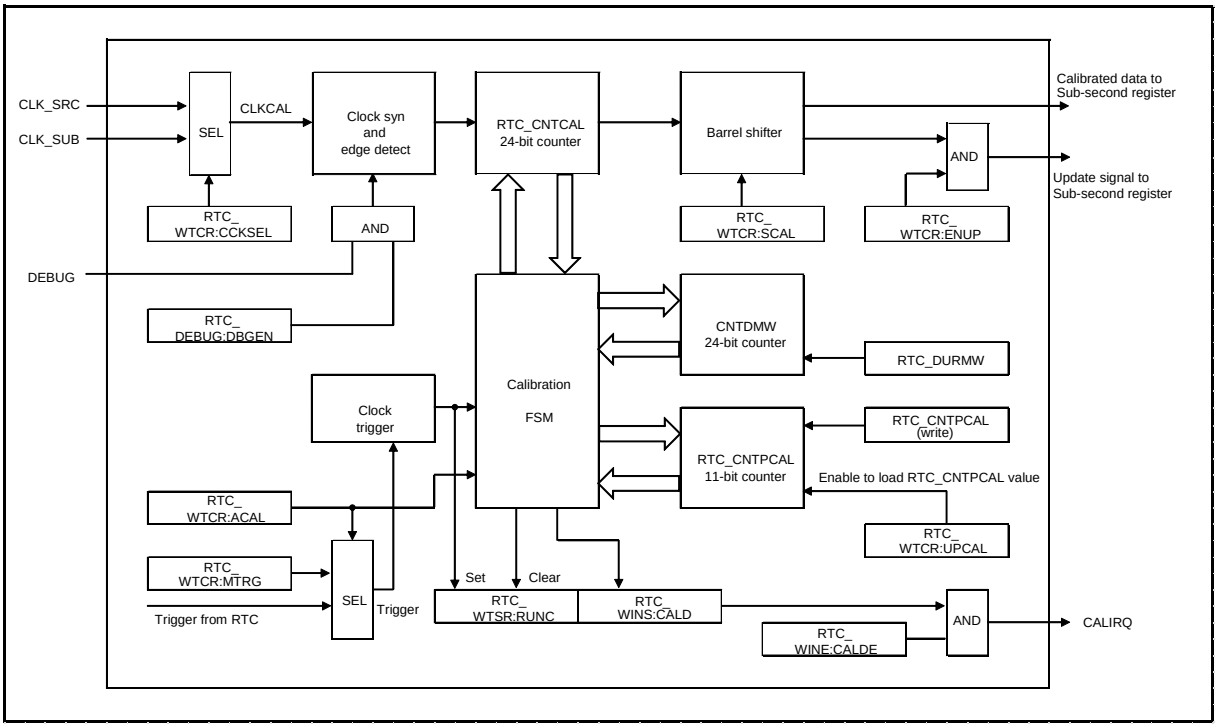
Block Diagram of RTC Configuration

Figure 2-2 RTC Timer Module Diagram



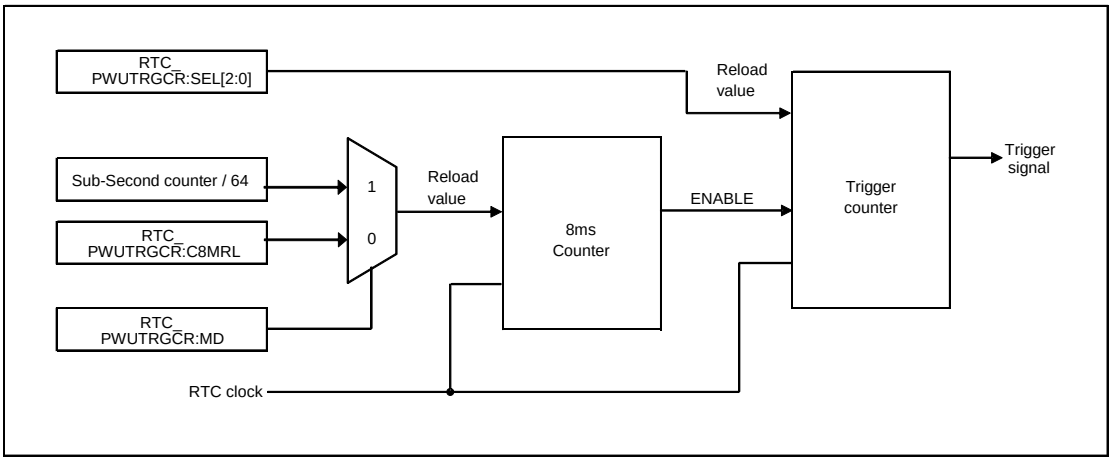
Calibration Configuration Diagram

Figure 2-3 Calibration Module Diagram



Block Diagram of Partial Wake Up Trigger

Figure 2-4 Partial Wake Up Trigger Module Diagram

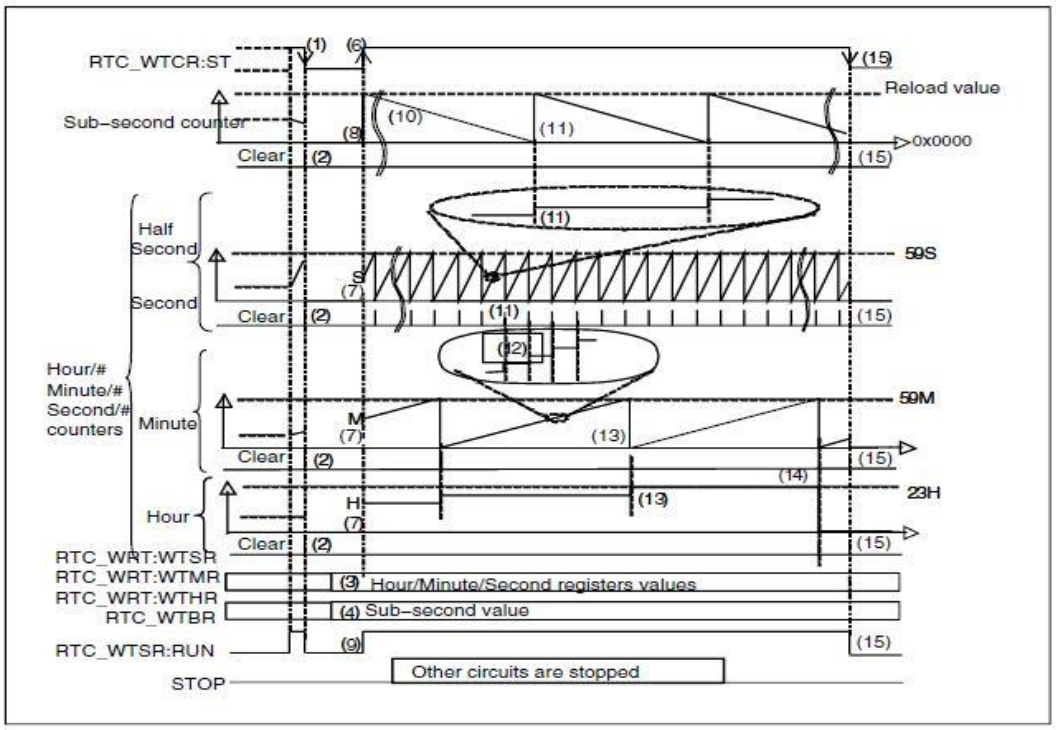


3. Operation of the Real Time Clock

This section describes the RTC Timer module and Calibration module operation.

RTC Timer Module Operation

Figure 3-1 RTC Timer Module Operation



The operation of the RTC Timer module, shown in Figure 3-1, is described as follows:

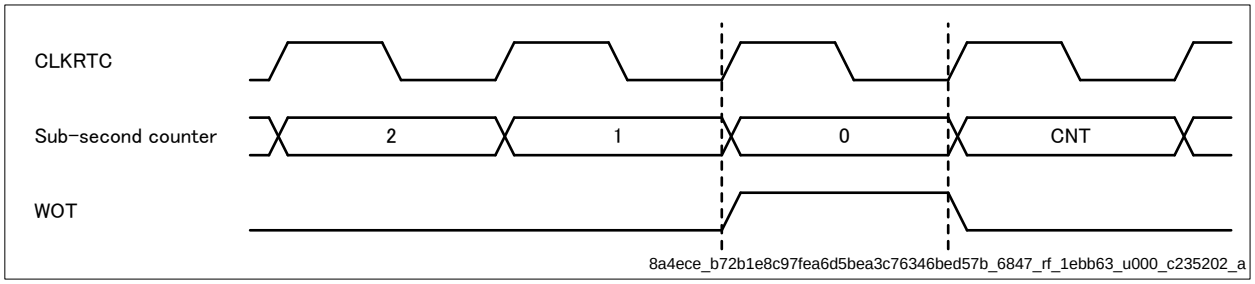
1. The software writes the start bit RTC_WTCR:ST to '1' and then '0'. (register initialization operation)
2. This resets the Sub-second counter to '0'. While RTC_WTCR:ST is '0' all counters for Sub-second, Second, Minute, and Hour are stopped.
3. The software writes Hour, Minute, and Second values to the Real Time Register, RTC_WRT.
4. The software writes the appropriate values to the Sub-Second Register, RTC_WTBR.
5. The interrupt request bits (RTC_WINS:SEC, RTC_WINS:MIN, RTC_WINS:HOUR, RTC_WINS:DAY, and RTC_WINS:SUBSEC) are cleared to '0', and the interrupt request enable bits (RTC_WINE:SECE, RTC_WINE:MINE, RTC_WINE:HOURE, RTC_WINE:DAYE and RTC_WINE:SUBSECE) are set to '1' i.e. interrupts enabled.
6. The software writes the start bit RTC_WTCR:ST to '1' to start the RTC counting operation.
7. This (RTC_WTCR:ST = '1') causes the values of the Real Time Register RTC_WRT to be loaded to the Hour/Minute/Second timers.
8. The value of the Sub-Second Register RTC_WTBR is loaded to the Sub-second counter.
9. With the second rising edge of CLKRTC the run flag RTC_WTSR:RUN is set to '1'.
10. The Sub-second counter begins counting on the CLKRTC. This could be the Main clock (CLK_MAIN) divided by 2 (4/2 MHz), Sub clock (CLK_SUB) divided by 2 (32.768/2 kHz) or Slow CR clock (CLK_SRC) divided by 2 (100/2 kHz).
11. When the Sub-second counter reaches '0', the value of the Sub-Second Register is loaded to the Sub-second counter, generating a sub-second interrupt request, i.e. the RTC_WINS:SUBSEC flag is set. For each other half-second interrupt, a one-second interrupt is generated, i.e. the RTC_WINS:SEC flag is set, and the Second counter is incremented.

12. When the Second counter reaches 59, it is cleared the next time it counts up, at which point the Minute counter counts up, generating a 1-minute interrupt request.
13. When the Minute counter reaches 59, it is cleared the next time it counts up, at which point the Hour counter counts up, generating a 1-hour interrupt request.
14. When the Hour counter reaches 23, it is cleared the next time it counts up, at which point a 1-day interrupt request is generated.
15. Writing RTC_WTCR:ST to '0' resets and stops the Sub-second counter and the Hour/Minute/Second counters.

Note:

- *To operate the RTC in low-power mode, the selected CLKRTC must remain enabled in the System Controller's Power Saving State (PSS) profile.*

Figure 3-2 WOT Pin Operation



Calibration Operation

The Calibration module provides automatic (and manual) RTC calibration. The module is able to operate in all run modes of the MCU if configured accordingly. The Main oscillation clock is used as an accurate time reference signal to measure the (normally) less accurate clock used for the RTC (CLKCAL). If the Main oscillator is switched off (e.g. during low-power mode), the Calibration module will temporarily switch it on, execute the calibration and switch it off afterwards to ensure low-current consumption.

Two counters operate during a calibration cycle to measure CLKCAL. The Calibration clock counter measures full clock cycles of CLKCAL and returns the value in RTC_CNTCAL. The second counter measures the period length of CLKCAL by counting Main clock cycles between two rising edges. This value is returned in RTC_CNTPCAL. Both counters operate during a configurable measurement window duration (RTC_DURMW), which is normally set up for 0.25 s. This allows the counted CLKCAL cycles to be directly used as the reload value for the Sub-second counter. During Automatic Calibration mode this value is transferred to the Sub-second counter by hardware means only, i.e. no software interaction is required.

Due to the cyclic and automatic re-calibration of the RTC Sub-second counter, any static and dynamic deviation of the RTC clock can be compensated for, resulting in very accurate RTC timing behavior. In the Calibration module, a novel method that positions the measurement window relative to CLKCAL actually modulates the Sub-second counter. This modulation can compensate for deviations that normally accumulate over time (caused by the limited resolution of the Sub-second counter to count only full CLKRTC cycles).

The following steps describe how to set up the Calibration module. The calibration operation is shown in Figure 3-3:

1. Configure the measurement window duration time for a calibration cycle in RTC_DURMW (normally 1,000,000 for 0.25 s and 4 MHz Main clock). During this time the clock to be calibrated (CLKCAL) is measured. The longer the measurement window, the higher the accuracy of the Calibration value i.e. the RTC timing.

2. Set the Calibration Value Scaling (RTC_WTCR:SCAL) with respect to the value set in the RTC_DURMW. For the normal measurement window of 0.25 s, scaling is not required (RTC_WTCR:SCAL must be set to '0'). Select the clock to be calibrated by configuring RTC_WTCR:CCKSEL.

3. Set RTC_WINE:CALDE to '0' or '1' to disable or enable interrupt generation after a calibration cycle has finished. This may wake up the MCU in low-power mode.

Automatic Calibration:

1. Write the Calibration trigger counter value to RTC_CALTRG to configure the time interval between two calibration cycles. For low-power mode, this also defines the intervals to wake up the Main oscillator.
2. Set RTC_WTCR:ENUP to '1' to enable the automatic transfer of the measured calibration value to the Sub-Second Register (RTC_WTBR) at the end of every calibration cycle.
3. Set RTC_WTCR:ACAL to '1', to enable Automatic Calibration. When the RTC is operating (RTC_WTSR:RUN) = '1', the Calibration trigger counter will run and generate cyclic calibration trigger events.
4. Set RTC_WINE:CFDE to '1' to enable interrupt generation (and MCU wake-up) in case of a calibration failure. An interrupt request will be generated when a calibration failure is detected if RTC_WTBR is not updated within one second after the hardware trigger. This happens when the Main clock is missing.

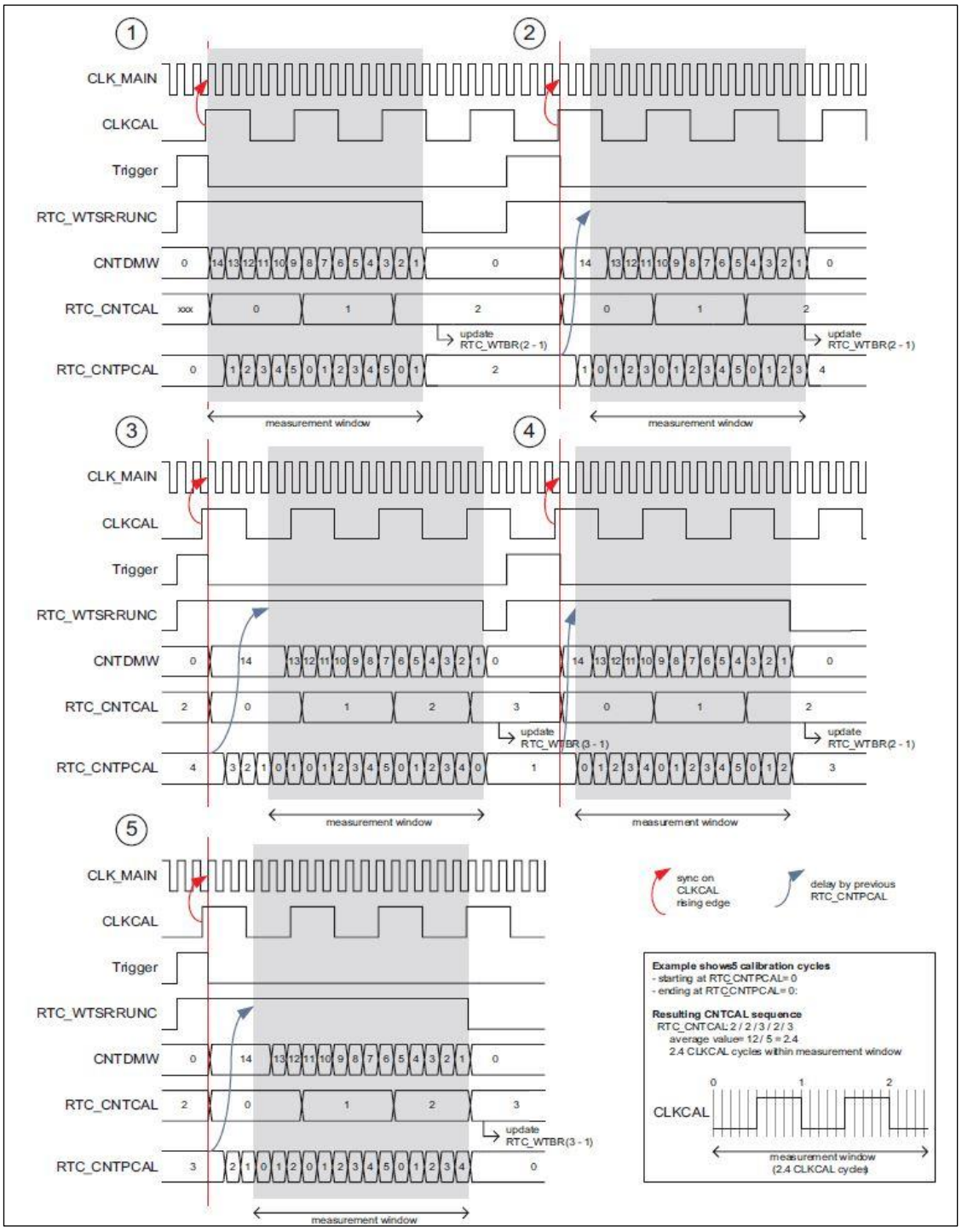
Manual Calibration:

1. Set RTC_WTCR:ACAL to '0' to disable Automatic Calibration and allow the calibration to be triggered by a Manual trigger.
2. Set RTC_WTCR:ENUP to '0' or '1' to disable or enable the updating of the Sub-Second Register (RTC_WTBR) with the calibration value at the end of the Manual Calibration.
3. Clear the Calibration done flag (RTC_WINS:CALD) by writing RTC_WINC:CALDC to '1'.
4. Set RTC_WTCR:MTRG to '1' - this triggers the calibration by software.
5. If RTC_WTCR:ENUP is disabled, read the RTC_CNTCAL value after a Calibration-done interrupt. If necessary, scale the read value according to the programmed Calibration duration value. Write the scaled value to the RTC_WTBR register.
6. If RTC_WTCR:ENUP is enabled, the Calibration module will scale the value according to the configuration in RTC_WTCR:SCAL and update.

Note:

- *For Automatic Calibration in low-power mode, ensure a valid stabilization time for CLK_MAIN oscillation is configured. Information about stabilization time can be found in CHAPTER 9.*

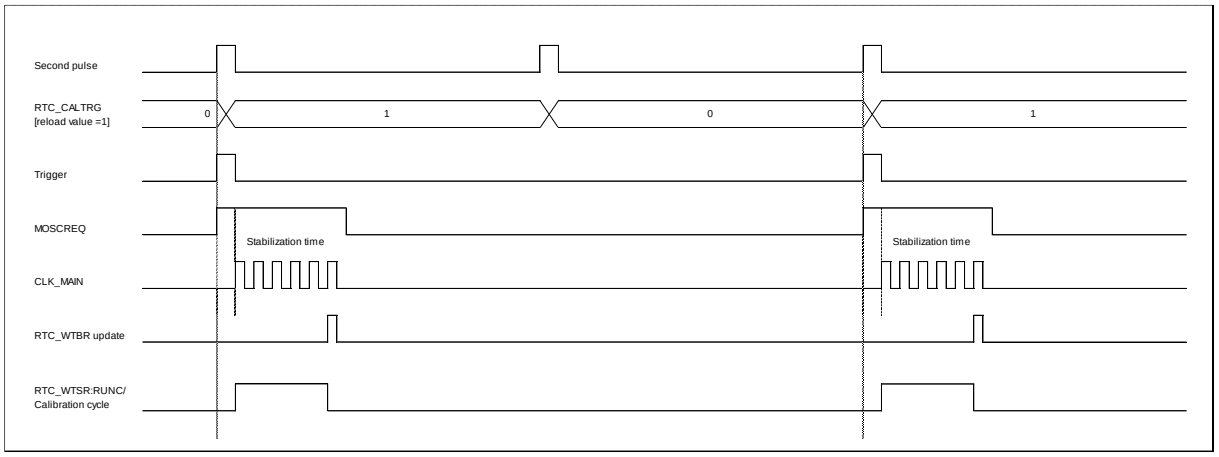
Figure 3-3 Calibration Operation



Note:

- *Simplified drawing without oscillation stabilization time.*

Figure 3-4 Cyclic Calibration



Cyclic Calibration

Cyclic Calibration, as shown in Figure 3-4, is done in Automatic Calibration mode. The Calibration trigger counter reloads the value configured in RTC_CALTRG and decrements on every second pulse from the RTC. It generates a calibration trigger when it expires. Then Main oscillator start request (MOSCREQ) is asserted to switch on the Main oscillation, which is available after specified stabilization time. The calibration now operates with the Main clock (CLK_MAIN). At the end of a Calibration cycle and with Automatic RTC_WTBR update enabled (RTC_WTCR:ENUP ='1'), RTC_WTBR will be updated with the measured (and scaled) value of RTC_CNTCAL. The Main oscillator start request (MOSCREQ) de-asserts at the end of the calibration to switch off the Main oscillator in low-power mode. As the Calibration trigger counter reloads after it expires, the Calibration operation repeats cyclically. If the Main clock is missing after MOSCREQ is asserted, a Calibration Failure Detection (CFD) interrupt is generated (if RTC_WINE:CFDE = '1'). The Calibration Trigger Register (RTC_CALTRG) value defines the duration between two calibrations.

4. Registers

This section describes the registers of real time clock.

Registers of the RTC

The following registers are available for the RTC:

Table 4-1 Registers of Real Time Clock

Abbreviated expressions	Register Name	Reference
RTC_WTCR	Timer Control Register	4.1
RTC_WTSR	Timer Status Register	4.2
RTC_WINS	Interrupt Status Register	4.3
RTC_WINE	Interrupt Enable Register	4.4
RTC_WINC	Interrupt Clear Register	4.5
RTC_WTBR	Sub-Second Register	4.6
RTC_WRT	Real Time Register	4.7
RTC_RTR1	Real Time Register1	4.8
RTC_CNTCAL	Calibration Clock Counter Register	4.9
RTC_CNTPCAL	Calibration Clock Period Counter Register	4.10
RTC_DURMW	Calibration Duration Register	4.11
RTC_CALTRG	Calibration Trigger Register	4.12
RTC_DEBUG	Debug Register	4.13
RTC_PWUTRGCR	Partial Wake Up Trigger Control Register	4.14
RTC_PWUTRGSR	Partial Wake Up Trigger Status Register	4.15

Memory Layout of the RTC Registers

Table 4-2 Memory Layout of RTC Registers

Offset	Register Name / Initial Value
0x0000_0000	RTC_WTCR 00000000_00000000_00000000_00000000
0x0000_0004	RTC_WTSR 00000000_00000000_00000000_00000000
0x0000_0008	RTC_WINS 00000000_00000000_00000000_00000000
0x0000_000C	RTC_WINE 00000000_00000000_00000000_00000000
0x0000_0010	RTC_WINC 00000000_00000000_00000000_00000000
0x0000_0014	RTC_WTBR 00000000_00000000_00000000_00000000
0x0000_0018	RTC_WRT 00000000_00000000_00000000_00000000
0x0000_001C	RTC_RTR1 00000000_00000000_00000000_00000000
0x0000_0020	RTC_CNTCAL 00000000_00000000_00000000_00000000
0x0000_0024	RTC_CNTPCAL 00000000_00000000_00000000_00000000
0x0000_0028	RTC_DURMW 00000000_00000000_00000000_00000000
0x0000_002C	RTC_CALTRG 00000000_00000000_00000000_00000000
0x0000_0030	RTC_DEBUG 00000000_00000000_00000000_00000000
0x0000_0034	RTC_PWUTRGCR 00000010_00000000_00000000_00000000
0x0000_0038	RTC_PWUTRGSR 00000000_00000000_00000000_00000000

Note:

- *All RTC registers and counters are initialized only by the PONR, INITX, IMR, LVDL1R, RVD, CSVMOR, CSVSOR and CSVSCRR.*

4.1. Timer Control Register (RTC_WTCR)

The Timer Control Register contains various control bits for the RTC including start, stop, mode and clock selection.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved							
ACCESS_TYPE	R0,WX							
PROT_TYPE	-							
INITIAL_VALUE	00000000							

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved							
ACCESS_TYPE	R0,WX							
PROT_TYPE	-							
INITIAL_VALUE	00000000							

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	UPCAL	SCAL[2:0]			CCKSEL	ENUP	MTRG	ACAL
ACCESS_TYPE	R,W	R,W			R/W	R,W	R0,W	R,W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved		RCKSEL[1:0]		CSM	UPDT	OE	ST
ACCESS_TYPE	R0,WX		R/W		R/W	R,W	R/W	R/W
PROT_TYPE	-							
INITIAL_VALUE	00		00		0	0	0	0

[bit31:16] Reserved

[bit15] UPCAL: Update calibration period counter

This bit is used to update the Calibration Period Counter (RTC_CNTPCAL). By setting this bit to '1', the RTC_CNTPCAL (write) register value is written to the RTC_CNTPCAL counter at the next Calibration trigger (which can be either Automatic or Manual).

Bit	Description
0	The update has been completed
1	Update the RTC_CNTPCAL counter with the value of the RTC_CNTPCAL (write) register

UPCAL is set by software and cleared by hardware after the completion of an update. Writing '0' by software will not effect this bit.

To make sure that update is done properly, the software should not change the RTC_CNTPCAL (write) register value before UPCAL is de-asserted by hardware.

Setting UPCAL to `1' has higher priority than setting it to `0' (in case both updates come at the same time).

[bit14:12] SCAL[2:0]:Scale calibrated value

These bits are used to scale the Calibration value in (RTC_CNTCAL), which is determined by the Calibration module, before it is transferred to the Sub-Second Register.

Bits	Description
000	No change
001	Multiply by 2
010	Multiply by 4
011	Multiply by 8
100	Multiply by 16
Others	Prohibited

SCAL should be configured w.r.t the value set in the corresponding Calibration Duration Register (RTC_DURMW).

For RTC_DURMW = (0.25s)/x, where x = {1,2,4,8,16}

Configure SCAL accordingly to match `x`

e.g. RTC_DURMW = (0.25s)/2 = 0.125s

set SCAL = `001`

Note:

- *Because of the internal clock synchronization, it takes certain period (several Main clock period and several bus clock period) that the written value is reflected to the bit.*

[bit11] CCKSEL: Clock select for calibration

This bit is used to select the Calibration clock (CLKCAL).

Bit	Description
0	Sub clock (CLK_SUB) is used as CLKCAL
1	Slow CR clock (CLK_SRC) is used as CLKCAL

[bit10] ENUP: Enable/Disable calibration value update

This bit is used to enable/disable the transfer of the calibration value in Calibration Clock Counter Register (RTC_CNTCAL) to the RTC Sub-Second Register (RTC_WTBR) after the completion of a calibration cycle.

Bit	Description
0	Calibration value (RTC_CNTCAL) update to RTC_WTBR is disabled
1	Calibration value (RTC_CNTCAL) update to RTC_WTBR is enabled

This bit is valid only when the Automatic Calibration mode is disabled (RTC_WTCR:ACAL is `0').

Multiple triggers during calibration are ignored. Reading this bit always returns '0'. This bit is cleared by the hardware at the start of calibration.

Note:

- *Because of the internal clock synchronization, it takes certain period (several Main clock period and several bus clock period) that the written value is reflected to the bit.*

[bit9] MTRG: Manual trigger for calibration

This bit is used to trigger a software calibration cycle.

Bit	Description
0	No trigger
1	Manual trigger

This bit is valid only when the Automatic Calibration mode is disabled (RTC_WTCR:ACAL is `0').

Multiple triggers during calibration are ignored.

Reading this bit always returns '0'. This bit is cleared by the hardware at the start of calibration.

[bit8] ACAL: Automatic calibration

This bit is used to enable/disable Automatic Calibration mode.

Bit	Description
0	Automatic Calibration disabled. A manual trigger (RTC_WTCR:MTRG) can be used to trigger calibration
1	Automatic Calibration enabled. The RTC will cyclically trigger the Automatic Calibration. The intervals are configured in the Calibration Trigger Register (RTC_CALTRG)

Switching ACAL will reset the ongoing Calibration.

In addition changing ACAL from '0' to '1' reloads the Calibration trigger counter with the RTC_CALTRG value

Notes:

- *For a detailed description of the Automatic RTC Calibration, refer to Section 3 'Operation of the Real Time Clock' and Section 5 'Usage Precaution'.*
- *Because of the internal clock synchronization, it takes certain period (several Main clock period and several bus clock period) that the written value is reflected to the bit.*

[bit7:6] Reserved

[bit5:4] RCKSEL[1:0]: Clock select for RTC

These bits are used to select the input clock for the RTC (CLKRTC).

Bits	Description
00	Main clock (CLK_MAIN/2) is selected as CLKRTC
01	Sub clock (CLK_SUB/2) is selected as CLKRTC
10	Slow CR clock (CLK_SRC/2) is selected as CLKRTC
11	Prohibited

Note:

- *It's necessary to write in register when RTC_WTSR:CLK_STS is 0. When CLK_STS is 1, the writing value is ignored.*

[bit3] CSM: Clock switching mode

This bit is used to configure the clock switching mode when RTC_WTCR:RCKSEL is changed.

Bit	Description
0	Precise switch
1	Immediate switch

Note:

- *It's necessary to write in register when RTC_WTSR:CLK_STS is 0. When CLK_STS is 1, the writing value is ignored.*

Precise switch - Clock switching is done when the Sub-second counter reloads from RTC_WTBR i.e. after half second.

Immediate switch - Clock switching is done immediately when RTC_WTCR:RCKSEL is changed.

Precise clock switching allows the setting of the new RTC_WTBR value w.r.t. the next CLKRTC. To guarantee correct functionality, the CPU must ensure that the new RTC_WTBR and new RCKSEL are set before the Sub-second counter reaches '0' (ideally use the half-second interrupt).

[bit2] UPDT: Update

This bit is used to update the time value. By setting it to '1', the Second/Minute/Hour register values are written to the Second/Minute/Hour counters respectively.

Bit	Description
0	The update has been completed
1	Update the Hour/Minute/Second counters with the values of the Hour/Minute/Second registers, respectively

UPDT is set by software and is cleared by hardware after the completion of an update. Writing '0' by software will not effect this bit.

Updating by the CPU has a higher priority than updating by hardware (in case both updates come at the same time). Refer to section 5 'Usage Precaution'.

[bit1] OE: Output enable

This bit is used to enable/disable the WOT external pin.

Bit	Description
0	The WOT external pin can be used as a General-purpose I/O or for another peripheral block
1	The WOT external pin serves as the output for the Sub-second counter

[bit0] ST: Start

This bit is used to start/stop the RTC.

Bit	Description
0	The Real Time Clock module stops operating, and the Sub-second counter and the Hour/Minute/Second counters are stopped
1	The settings of the Hour/Minute/Second registers are loaded to the Hour/Minute/Second counters, and the RTC module starts to operate

Notes:

- *When writing "1" to the start bit (ST) from RTC stop state (ST=0) (RTC operation start), do not write "1" to the update bit (UPDT) at the same time as the start bit. (While ST=0, writing "1" as byte immediate value to the ST bit and the UPDT bit at the same time is prohibited.)*
- *To write "1" to the update bit (UPDT), do it while RTC is working (ST=1).*
- *While the update bit (UPDT) is "1", writing "0" to the start bit (ST) (RTC stop) is prohibited.*

4.2. Timer Status Register (RTC_WTSR)

The Timer Status Register contains various status bits for the RTC.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved							
ACCESS_TYPE	R0,WX							
PROT_TYPE	-							
INITIAL_VALUE	00000000							

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved							
ACCESS_TYPE	R0,WX							
PROT_TYPE	-							
INITIAL_VALUE	00000000							

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved							RUNC
ACCESS_TYPE	R0,WX							R,WX
PROT_TYPE	-							
INITIAL_VALUE	0000000							0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	CLK_STS	Reserved					CSF	RUN
ACCESS_TYPE	R,WX	R0,WX					R,W	R,WX
PROT_TYPE	-							
INITIAL_VALUE	000000						0	0

[bit31:9] Reserved

[bit8] RUNC: RUN calibration

This bit indicates the status of the Calibration.

Bit	Description
0	The Calibration is inactive
1	The Calibration is in progress

This register bit is updated on CLK_MAIN. RUNC is set to '1' when Calibration is started by an Automatic or Manual trigger. RUNC is set to '0' when the Calibration Duration Counter expires or Calibration is reset.

[bit7] CLK_STS: Clock switching status

This bit indicates the status of the clock switching.

Bit	Description
0	The clock switching is inactive
1	The clock switching is in progress

[bit6:2] Reserved

[bit1] CSF: Clock switched flag

This bit indicates the status of switching the RTC input clock.

Bit	Description
0	The input clock for RTC has not been switched
1	The input clock for RTC has been switched

This bit is set by hardware and cleared by software.

CSF indicates the switching of the RTC input clock in precise mode(RTC_WTCR:CSM is '0'). When RTC_WTCR:RCKSEL is changed(i.e. the clock on which the RTC will run) in precise mode (RTC_WTCR:CSM is '0'), the clock source will be switched after the Sub-second counter expires. Even though reading RTC_WTCR:RCKSEL will give the updated configuration, the CSF flag will only be set at the actual internal clock switching.

To use this bit, it has to be first cleared. Then write the new clock setting to RTC_WTCR:RCKSEL and check if CSF is set to '1'.

Update by CPU has higher priority than update by hardware (in case both updates come at the same time).

[bit0] RUN: RUN

This bit indicates the status of the RTC Timer module.

Bit	Description
0	The RTC Timer module is inactive
1	The RTC Timer module is active

This register bit is updated on CLKRTC. RUN will go low at the second rising edge of the RTC clock after RTC_WTCR:ST has been set to '0'. It will rise again at the second rising edge of RTC clock after RTC_WTCR:ST has been set to '1'.

4.3. Interrupt Status Register (RTC_WINS)

The Interrupt Status Register contains various status bits that can trigger an interrupt.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved							
ACCESS_TYPE	R0,WX							
PROT_TYPE	-							
INITIAL_VALUE	00000000							

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved							
ACCESS_TYPE	R0,WX							
PROT_TYPE	-							
INITIAL_VALUE	00000000							

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved							
ACCESS_TYPE	R0,WX							
PROT_TYPE	-							
INITIAL_VALUE	00000000							

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	CALD	CFD	DAY	HOUR	MIN	SEC	SUBSEC
ACCESS_TYPE	R0,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:7] Reserved

[bit6] CALD: Calibration done

This bit is set when a Calibration cycle has finished. If the CALD interrupt is enabled (RTC_WINE:CALDE is `1'), an interrupt request is generated.

Bit	Description
0	Calibration is not triggered or in progress
1	Calibration is done

The Calibration Done flag will be set independently of the Calibration value update bit (RTC_WTCR:ENUP).

This bit is set to '1' at the Calibration Done event and is cleared by writing '1' to RTC_WINC:CALDC.

[bit5] CFD: Calibration failure detection

This bit is set when a calibration failure is detected. If the CFD interrupt is enabled (RTC_WINE:CFDE is `1'), an interrupt request is generated.

Bit	Description
0	No calibration failure
1	Calibration failure

The Calibration Failure Detection flag is set if the RTC_WTBR register is not updated by hardware within one second after an Automatic Calibration trigger event while the Calibration value update feature is enabled (RTC_WTCR:ENUP is `1').

The Calibration Failure Detection feature could also be used in Manual Calibration mode. This requires the issuing of the Manual trigger in time to ensure that the Calibration Duration Counter has elapsed before the next update of the Second counter.

The Calibration Failure Detection flag indicates that the Main clock is not available and calibration cannot be executed.

This bit is cleared by writing '1' to RTC_WINC:CFDC.

[bit4] DAY: Day flag

This bit is set when one day has elapsed. If the DAY interrupt is enabled (RTC_WINE:DAYE is `1'), an interrupt request is generated.

Bit	Description
0	No interrupt requests
1	The Hour counter has exceeded 24 hours

This bit is set to '1' at one-day intervals, i.e. when the Hour counter overflows. This bit is cleared by writing '1' to RTC_WINC:DAYC.

[bit3] HOUR: Hour flag

This bit is set when one hour has elapsed. If the HOUR interrupt is enabled (RTC_WINE:HOURE is `1'), an interrupt request is generated.

Bit	Description
0	No interrupt requests
1	The Hour counter has been updated

This bit is set to '1' at one-hour intervals, i.e. when the Minute counter overflows. This bit is cleared by writing '1' to RTC_WINC:HOURC.

[bit2] MIN: Minute flag

This bit is set when one minute has elapsed. If the MIN interrupt is enabled (RTC_WINE:MINE is `1'), an interrupt request is generated.

Bit	Description
0	No interrupt requests
1	The Minute counter has been updated

This bit is set to '1' at one-minute intervals, i.e. when the Second counter overflows. This bit is cleared by writing '1' to RTC_WINC:MINC.

[bit1] SEC: Second flag

This bit is set when one second has elapsed. If the SEC interrupt is enabled (i.e. RTC_WINE:SECE is `1'), an interrupt request is generated.

Bit	Description
0	No interrupt requests
1	The Second counter has been updated

This bit is set to '1' at one-second intervals. This bit is cleared by writing '1' to RTC_WINC:SECC.

[bit0] SUBSEC: Sub-second flag

This bit is set when one half second has elapsed. If the SUBSEC interrupt is enabled (RTC_WINE:SUBSECE is `1'), an interrupt request is generated.

Bit	Description
0	No interrupt requests
1	The Sub-second counter has expired

This bit is set to '1' at half-second intervals, i.e. when the Sub-second counter expires. This bit is cleared by writing '1' to RTC_WINC:SUBSECC.

4.4. Interrupt Enable Register (RTC_WINE)

The Interrupt Enable Register contains various enable/disable bits for the interrupts.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved							
ACCESS_TYPE	R0,WX							
PROT_TYPE	-							
INITIAL_VALUE	00000000							

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved							
ACCESS_TYPE	R0,WX							
PROT_TYPE	-							
INITIAL_VALUE	00000000							

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved							
ACCESS_TYPE	R0,WX							
PROT_TYPE	-							
INITIAL_VALUE	00000000							

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	CALDE	CFDE	DAYE	HOURE	MINE	SECE	SUB SECE
ACCESS_TYPE	R0,WX	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:7] Reserved

[bit6] CALDE: Calibration done interrupt enable

This bit is used to enable interrupt requests at Calibration done (RTC_WINS:CALD).

Bit	Description
0	Interrupt requests at Calibration Done are disabled
1	Interrupt requests at Calibration Done are enabled

[bit5] CFDE: Calibration failure detection interrupt enable

This bit is used to enable interrupt requests at Calibration Failure Detection (RTC_WINS:CFD).

Bit	Description
0	Interrupt requests at Calibration Failure Detection are disabled
1	Interrupt requests at Calibration Failure Detection are enabled

[bit4] DAYE: Day interrupt enable

This bit is used to enable interrupt requests at one-day (24 hour) intervals (RTC_WINS:DAY).

Bit	Description
0	Interrupt requests at one-day (24 hour) intervals are disabled
1	Interrupt requests at one-day (24 hour) intervals are enabled

[bit3] HOURE: Hour interrupt enable

This bit is used to enable interrupt requests at one-hour intervals (RTC_WINS:HOUR).

Bit	Description
0	Interrupt requests at one-hour intervals are disabled
1	Interrupt requests at one-hour intervals are enabled

[bit2] MINE: Minute interrupt enable

This bit is used to enable interrupt requests at one-minute intervals (RTC_WINS:MIN).

Bit	Description
0	Interrupt requests at one-minute intervals are disabled
1	Interrupt requests at one-minute intervals are enabled

[bit1] SECE: Second interrupt enable

This bit is used to enable interrupt requests at one-second intervals (RTC_WINS:SEC).

Bit	Description
0	Interrupt requests at one-second intervals are disabled
1	Interrupt requests at one-second intervals are enabled

[bit0] SUBSECE: Sub-second interrupt enable

This bit is used to enable interrupt requests at half-second intervals (RTC_WINS:SUBSEC).

Bit	Description
0	Interrupt requests at half-second intervals are disabled
1	Interrupt requests at half-second intervals are enabled

4.5. Interrupt Clear Register (RTC_WINC)

The Interrupt Clear Register contains various clear bits for the Interrupt Status Register (RTC_WINS).

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved							
ACCESS_TYPE	R0,WX							
PROT_TYPE	-							
INITIAL_VALUE	00000000							

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved							
ACCESS_TYPE	R0,WX							
PROT_TYPE	-							
INITIAL_VALUE	00000000							

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved							
ACCESS_TYPE	R0,WX							
PROT_TYPE	-							
INITIAL_VALUE	00000000							

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	CALDC	CFDC	DAYC	HOURC	MINC	SECC	SUB SECC
ACCESS_TYPE	R0,WX	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:7] Reserved

[bit6] CALDC: CALD interrupt clear

This bit is used to clear interrupt requests at Calibration Done.

Bit	Description
0	No effect
1	Clears RTC_WINS:CALD

Reading this bit always returns '0'.

[bit5] CFDC: CFD interrupt clear

This bit is used to clear interrupt requests at Calibration Failure Detection.

Bit	Description
0	No effect
1	Clears RTC_WINS:CFD

Reading this bit always returns '0'.

[bit4] DAYC: Day interrupt clear

This bit is used to clear the Day Interrupt flag.

Bit	Description
0	No effect
1	Clears RTC_WINS:DAY

Reading this bit always returns '0'.

[bit3] HOURC: Hour interrupt clear

This bit is used to clear the Hour Interrupt flag.

Bit	Description
0	No effect
1	Clears RTC_WINS:HOUR

Reading this bit always returns '0'.

[bit2] MINC: Minute interrupt clear

This bit is used to clear the Minute Interrupt flag.

Bit	Description
0	No effect
1	Clears RTC_WINS:MIN

Reading this bit always returns '0'.

[bit1] SECC: Second interrupt clear

This bit is used to clear the Second Interrupt flag.

Bit	Description
0	No effect
1	Clears RTC_WINS:SEC

Reading this bit always returns '0'.

[bit0] SUBSECC: Sub-second interrupt clear

This bit is used to clear interrupt requests at half-second interval.

Bit	Description
0	No effect
1	Clears RTC_WINS:SUBSEC

Reading this bit always returns '0'.

4.6. Sub-Second Register (RTC_WTBR)

The Sub-Second Register stores the reload value for the Sub-second counter that divides the CLKRTC to generate the RTC's time base. The reload value is usually set so that the Sub-second counter will count exactly half a second.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved							
ACCESS_TYPE	R0,WX							
PROT_TYPE	-							
INITIAL_VALUE	00000000							

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	WTBR[23:16]							
ACCESS_TYPE	R,W							
PROT_TYPE	-							
INITIAL_VALUE	00000000							

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	WTBR[15:8]							
ACCESS_TYPE	R,W							
PROT_TYPE	-							
INITIAL_VALUE	00000000							

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	WTBR[7:0]							
ACCESS_TYPE	R,W							
PROT_TYPE	-							
INITIAL_VALUE	00000000							

[bit31:24] Reserved

[bit23:0] WTBR[23:0]: Sub-second register

The Sub-Second Register holds the value to be reloaded to the Sub-second counter. When the Sub-second counter value becomes '0' the value of WTBR is reloaded to the Sub-second counter.

This register is also updated by the calibration circuit when RTC_WTCR:ENUP is set (even in low-power mode). Updating by the CPU has higher priority than updating by hardware (in case both updates come at the same time).

Note:

- *The required value depends on the RTC clock that is used (CLKRTC) and can be calculated with the formula on the following page.*

Table 4-3 Example Configuration of RTC_WTBR Register for Different Clock Configurations

	RTC_WTBR:WTBR[23:0]
Main oscillator, 4MHz	0x0F423F
RC oscillator, 100kHz	0x0061A7
Sub oscillator, 32.768kHz	0x001FFF

Calculation formula:

RTC_WTBR:WTBR[23:0] = fCLKRTC x 0.5 s – 1

fCLKRTC: CLKRTC frequency [Hz]

4.7. Real Time Register (RTC_WRT)

The Real Time Register (RTC_WRT) stores the time information i.e. Second/Minute/Hour values. Reading this register returns the counter values. This register is write accessible; however, the written data is loaded into the counters when the RTC is started (RTC_WTCR:ST set to '1') and each time the RTC_WTCR:UPDT bit is set to '1'.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved							
ACCESS_TYPE	R0,WX							
PROT_TYPE	-							
INITIAL_VALUE	00000000							

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved			WTHR[4:0]				
ACCESS_TYPE	R0,WX			R,W				
PROT_TYPE	-							
INITIAL_VALUE	000			00000				

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved		WTMR[5:0]					
ACCESS_TYPE	R0,WX		R,W					
PROT_TYPE	-							
INITIAL_VALUE	00		000000					

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved		WTSR[5:0]					
ACCESS_TYPE	R0,WX		R,W					
PROT_TYPE	-							
INITIAL_VALUE	00		000000					

[bit31:21] Reserved

[bit20:16] WTHR[4:0]: Hour register

The Hour bits are used to (initially) set the RTC Hour counter value and to read the actual Hour value.

The written value is transferred to the Hour counter by setting `1' to the update bit (RTC_WTCR:UPDT) or start bit (RTC_WTCR:ST).

Reading the Hour bits returns the latest Hour counter value, not the written value. The Hour counter value is saved to the hour bits every time the Sub-second counter expires, i.e. at intervals of half a second.

[bit15:14] Reserved

[bit13:8] WTMR[5:0]: Minute register

The minute bits are used to (initially) set the RTC Minute counter value and to read the actual Minute value.

The written value is transferred to the Minute counter by setting `1' to the update bit (RTC_WTCR:UPDT) or start bit (RTC_WTCR:ST).

Reading the Minute bits returns the latest Minute counter value, not the written value. The Minute counter value is saved to the minute bits every time the Sub-second counter expires, i.e. at intervals of half a second.

[bit7:6] Reserved

[bit5:0] WTSR[5:0]: Second register

The second bits are used to (initially) set the RTC Second counter value and read the actual Second value.

The written value is transferred to the Second counter by setting `1' to the update bit (RTC_WTCR:UPDT) or start bit (RTC_WTCR:ST).

Reading the Second bits returns the latest Second counter value, not the written value. The Second counter value is saved to the second bits every time the Sub-second counter expires, i.e. at intervals of half a second.

Note:

- *RTC_WRT is initialized only by the power-on reset, low-voltage reset and external resets, and not by any other reset except for unused bits.*

4.8. Real Time Clock Real Time Register1 (RTC_RTR1)

Real-time register (RTC_RTR1) is used to read and to update time information of the real time clock.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved							
ACCESS_TYPE	R0,WX							
PROT_TYPE	-							
INITIAL_VALUE	00000000							

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved							
ACCESS_TYPE	R0,WX							
PROT_TYPE	-							
INITIAL_VALUE	00000000							

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	WTDR[15:8]							
ACCESS_TYPE	R,W							
PROT_TYPE	-							
INITIAL_VALUE	00000000							

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	WTDR[7:0]							
ACCESS_TYPE	R,W							
PROT_TYPE	-							
INITIAL_VALUE	00000000							

[bit31:16] Reserved

[bit15:0] WTDR[15:0]: Date Bit

These bits show the count value of the day of the real time clock. The writing value is reflected in the count of the day of the real time clock when writing it.

Note:

- *These bits show day information of the real time clock by 16 bits. Do not access these bits with byte width because an appropriate value cannot be read.*

4.9. Calibration Clock Counter Register (RTC_CNTCAL)

The Calibration Clock Counter Register gives the value of counted CLKCAL cycles of a calibration cycle. The Calibration module controls the Calibration clock counter which is active during the measurement window duration. The counted value can be used to update the Sub-Second Register automatically or manually depending on configurations. Reading this register simply returns the counter value.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved							
ACCESS_TYPE	R0,WX							
PROT_TYPE	-							
INITIAL_VALUE	00000000							

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	CNTCAL[23:16]							
ACCESS_TYPE	R,WX							
PROT_TYPE	-							
INITIAL_VALUE	00000000							

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	CNTCAL[15:8]							
ACCESS_TYPE	R,WX							
PROT_TYPE	-							
INITIAL_VALUE	00000000							

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	CNTCAL[7:0]							
ACCESS_TYPE	R,WX							
PROT_TYPE	-							
INITIAL_VALUE	00000000							

[bit31:24] Reserved

[bit23:0] CNTCAL[23:0]: Calibration clock counter register

This register returns the value of the Calibration clock counter. The Calibration clock counter counts full clock cycles of the clock to be calibrated, i.e. CLKCAL. The counting is controlled by the Calibration Module and is active during the Calibration measurement window duration (RTC_DURMW).

This unscaled value can be used to manually update the Sub-Second Register. For the Automatic update of RTC_WTBR (RTC_WTCR:ENUP is set to `1'), the value is scaled according to the configured scaling (RTC_WTCR:SCAL) and written to RTC_WTBR. This register always returns the unscaled value.

The CNTCAL Register is updated with the Calibration clock counter value at the end of a calibration cycle i.e. when the measurement window duration has elapsed. The Calibration Done flag (RTC_WINS:CALD) and/or interrupt can be used to determine the end of a calibration cycle.

Notes:

- *For a manual update of the RTC_WTBR register, the software should scale the CNTCAL value w.r.t the configured measurement window duration (RTC_DURMW) and then subtract '1' before writing it to the Sub-Second Register. (Scaling Factor X CNTCAL value) - 1.*
- *For more details on the RTC_WTSR:SCAL value calculation, see the description of RTC_WTCR:SCAL*
- *For a detailed description of the Automatic RTC Calibration, refer to Sections 3 and 5.*

4.10. Calibration Clock Period Counter Register (RTC_CNTPCAL)

The Calibration Clock Period Counter Register gives the value of the last Calibration clock period of a Calibration cycle. The Calibration clock period counter measures the period length of CLKCAL by counting cycles of the Main clock. The Calibration module controls the Calibration clock period counter which is active during the measurement window duration. The counter is reset with each positive edge of CLKCAL. The counted value at the end of a calibration cycle is used to delay the start of the next calibration duration measurement window. Reading this register simply returns the counter value. Writing this register allows the setting of the Calibration clock period counter.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved							
ACCESS_TYPE	R0,WX							
PROT_TYPE	-							
INITIAL_VALUE	00000000							

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved							
ACCESS_TYPE	R0,WX							
PROT_TYPE	-							
INITIAL_VALUE	00000000							

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved					CNTPCAL[10:8]		
ACCESS_TYPE	R0,WX					R,W		
PROT_TYPE	-							
INITIAL_VALUE	00000					000		

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	CNTPCAL[7:0]							
ACCESS_TYPE	R,W							
PROT_TYPE	-							
INITIAL_VALUE	00000000							

[bit31:11] Reserved

[bit10:0] CNTPCAL[10:0]: Calibration clock period register

In this register the value to be loaded to the Calibration clock period counter before a calibration cycle is set and the value of the Calibration clock period counter after a calibration cycle is read.

The Calibration clock period counter counts the number of positive edges of CLK_MAIN between two positive edges of CLKCAL within a calibration measurement window duration. This is controlled by the Calibration Module during a calibration cycle. The counter value at the end of the calibration cycle is the remainder of the last CLKCAL period that was not counted by the Calibration clock counter (returned by RTC_CNTCAL). This value is then used for the next calibration cycle.

There are separate read and write registers.

CNTPCAL (read) register is updated with the Calibration clock period counter value at the end of a calibration cycle. The Calibration Done flag (RTC_WINS:CALD) and/or interrupt can be used to determine the end of a calibration cycle.

By setting '1' to RTC_WTCR:UPCAL, the CNTPCAL (write) register value is written to the CNTPCAL counter at the next calibration trigger (which can be either Automatic or Manual).

If RTC_WTCR:ACAL is changed during an ongoing calibration, the calibration will be stopped and the CNTPCAL counter will get an intermediate value. To make sure the value is not used for the next calibration cycle, software must update the CNTPCAL counter (e.g. by writing `0').

Note:

- *For a detailed description of the Automatic RTC Calibration, refer to Sections 3 and 5.*

4.11. Calibration Duration Register (RTC_DURMW)

The Calibration Duration Register stores the reload value of the Calibration Duration Counter (CNTDMW). This value is loaded into the Calibration Duration Counter at the calibration trigger. The value determines the calibration measurement window duration during which CLKCAL is measured.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved							
ACCESS_TYPE	R0,WX							
PROT_TYPE	-							
INITIAL_VALUE	00000000							

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	DURMW[23:16]							
ACCESS_TYPE	R/W							
PROT_TYPE	-							
INITIAL_VALUE	00000000							

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	DURMW[15:8]							
ACCESS_TYPE	R/W							
PROT_TYPE	-							
INITIAL_VALUE	00000000							

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	DURMW[7:0]							
ACCESS_TYPE	R/W							
PROT_TYPE	-							
INITIAL_VALUE	00000000							

[bit31:24] Reserved

[bit23:0] DURMW[23:0]: Calibration duration register

This register stores the reload value of the Calibration duration counter. The counter is controlled by the Calibration Module and is used to determine the measurement window. During the active measurement window CLKCAL is measured by the Calibration clock counter and Calibration clock period counter.

Notes:

1. RTC_DURMW should be programmed w.r.t. CLK_MAIN to achieve a measurement duration of exactly 0.25 s. This would result in a calibration value for CLKCAL, determined by the Calibration clock counter, that perfectly matches the required setting for the Sub-Second Register (RTC_WTBR) i.e. for half a second.

2. *Due to the fixed clock prescaler of 2 for the RTC clock (CLKRTC), measuring for 0.25 s returns a calibration value for 0.5 s. Shorter measurement window durations are possible, but should be a half, quarter, etc., of the normal duration of 0.25 s. By configuring the scaling feature accordingly (RTC_WTCR:SCAL), the RTC_CNTCAL value of shorter measurement windows can be scaled to match the Sub-second counter requirements for 0.5 s.*

Table 4-4 Example Configuration of RTC_DURMW Register for Different RTC_WTCR:SCAL Configurations at fCLK_MAIN, 4 MHz

RTC_WTCR:SCAL	RTC_DURMW:DURMW[23:0]
'000'	0x0F4240
'001'	0x07A120
'010'	0x03D090
'011'	0x01E848
'100'	0x00F424

Calculation Formula:

RTC_DURMW:DURMW[23:0] = fCLK_MAIN/(2^SCAL) x 0.25 s

where,

fCLK_MAIN: CLK_MAIN frequency [Hz]

4.12. Calibration Trigger Register (RTC_CALTRG)

The Calibration Trigger Register stores the reload value of the 12-bit Calibration trigger counter, which is used to generate cyclic trigger events for Automatic Calibration. This register defines the time interval in seconds between two consecutive calibration cycles (In fact, the actual time interval is RTC_CALTRG+1). In low-power mode, the counter cyclically triggers the Calibration Module to switch on the Main oscillation and execute a calibration cycle.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved							
ACCESS_TYPE	R0,WX							
PROT_TYPE	-							
INITIAL_VALUE	00000000							

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved							
ACCESS_TYPE	R0,WX							
PROT_TYPE	-							
INITIAL_VALUE	00000000							

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved				CALTRG[11:8]			
ACCESS_TYPE	R0,WX				R/W			
PROT_TYPE	-							
INITIAL_VALUE	0000				0000			

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	CALTRG[7:0]							
ACCESS_TYPE	R/W							
PROT_TYPE	-							
INITIAL_VALUE	00000000							

[bit31:12] Reserved

[bit11:0] CALTRG[11:0]: Calibration trigger register

This register stores the reload value of the 12-bit programmable Calibration trigger counter.

In Automatic Calibration mode (RTC_WTCR:ACAL is set to `1'), the Calibration trigger counter is decremented on each RTC Second interval. When the counter reaches zero, a calibration trigger is issued to the Calibration Module.

In Manual Calibration mode, this timer is not used.

Note:

- *While the actual time interval (in seconds) between two consecutive calibration cycles is RTC_CALTRG:CALTRG [11:0] +1, the register value is actually RTC_CALTRG :CALTRG [11:0] -1.*

4.13. Debug Register (RTC_DEBUG)

The Debug Register has configuration bits for debug functionality.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved							
ACCESS_TYPE	R0,WX							
PROT_TYPE	-							
INITIAL_VALUE	00000000							

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved							
ACCESS_TYPE	R0,WX							
PROT_TYPE	-							
INITIAL_VALUE	00000000							

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved							
ACCESS_TYPE	R0,WX							
PROT_TYPE	-							
INITIAL_VALUE	00000000							

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME								DBGEN
ACCESS_TYPE	R0,WX							R/W
PROT_TYPE	-							
INITIAL_VALUE	0000000							0

[bit31:1] Reserved

[bit0] DBGEN: Debug enable

This bit is used to enable/disable Debug mode for the RTC.

Bit	Description
0	Debug mode disabled
1	Debug mode enabled

When DBGEN is set to '1' and the processor is in the debug state, CLKRTC, CLKCAL, and CLK_MAIN are masked. Therefore the Sub-second counter and Calibration state machine are stopped.

For the definition of debug state, refer to Section 11.8 of the ARM® Cortex™-R5 Technical Reference Manual.

4.14. Partial Wake Up Trigger Control Register (RTC_PWUTRGCR)

This register controls the generation of the trigger signal used for Partial Wake Up.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved						C8MRL[17:16]	
ACCESS_TYPE	R0,WX						R/W	
PROT_TYPE	-							
INITIAL_VALUE	000000						10	

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	C8MRL[15:8]							
ACCESS_TYPE	R/W							
PROT_TYPE	-							
INITIAL_VALUE	00000000							

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	C8MRL[7:0]							
ACCESS_TYPE	R/W							
PROT_TYPE	-							
INITIAL_VALUE	00000000							

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved			MD	Reserved	SEL[2:0]		
ACCESS_TYPE	R0,WX			R/W	R0,WX	R/W		
PROT_TYPE	-							
INITIAL_VALUE	000			0	0	000		

[bit31:26] Reserved: Reservation Bit

[bit25:8] C8MRL[17:0]: Reload Value Setting Bit of 8ms Counter

C8MRL[17:0]	Reload Value Setting Bit of 8ms Counter
Writing	This bit is effective at MD= 0.It uses it to set 8ms counter with the manual operation. This bit sets the value that is reload to 8ms counter. 8ms counter is a down counter in 18 bits that works with the RTC clock. The count value is ALL "0" , reload when becoming it as for a set value of this bit. Please make the RTC clock correspond to the selected clock source and set the reload value.
Reading	The writing value can be read.

[bit7:5] Reserved: Reservation Bit

[bit4] MD: Reload Value Setting Bit to 8ms Counter

MD	Reload Value Setting Bit to 8ms Counter
Writing	The reload value to 8ms counter is selected. "0": RTC_PWUTRGCR.C8MRL[17:0] is selected. The reload setting value is decided in the manual operation. "1": The value in which the reload value to Sub-Second Counter is divided by 64 is selected. 0.25[s]÷32=7.8125[ms]≈8[ms] An appropriate value is set from a set value of Sub-Second register.
Reading	The writing value can be read.

[bit3] Reserved: Reservation Bit

[bit2:0] SEL[2:0]: Reload Value Setting Bit to Trigger Counter

SEL[2:0]	Reload Value Setting Bit to 8ms Counter
Writing	The reload value to the trigger counter that generates the trigger signal of Partial Wake Up is set. The interval time of Partial Wake Up can be set by setting the reload value. In the trigger counter, 8ms counter is 3bit decreased counter , saying that ALL '0'. The count value is reload, saying that ALL '0' as for a set value of this bit. The relation between a set value and the interval time is shown below. "000":8ms "001":16ms "010":24ms "011":32ms "100":40ms "101":48ms "110":56ms "111":64ms
Reading	The writing value can be read.

4.15. Partial Wake Up Trigger Status Register (RTC_PWUTRGSR)

This register shows the status about generation of the trigger signal used for Partial Wake Up.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved							
ACCESS_TYPE	R0,WX							
PROT_TYPE	-							
INITIAL_VALUE	00000000							

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved							
ACCESS_TYPE	R0,WX							
PROT_TYPE	-							
INITIAL_VALUE	00000000							

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved							
ACCESS_TYPE	R0,WX							
PROT_TYPE	-							
INITIAL_VALUE	00000000							

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved							BUSY
ACCESS_TYPE	R0,WX							R,WX
PROT_TYPE	-							
INITIAL_VALUE	0000000							0

[bit31:1] Reserved: Reservation Bit

[bit0] BUSY: Busy status

BUSY	Reload Value Setting Bit to 8ms Counter
Writing	Writing is invalid.
Reading	"0" : Update completion "1" : State of BUSY The transfer from the bus clock to the RTC clock is done when writing it in RTC_PWUTRGCR. It shows for "1" this bit while updating it by the clock transfer. The bus error is returned when writing it in RTC_PWUTRGCR in the state of BUSY.

5. Usage Precaution

This section describes the precautions to be considered when using the real time clock.

Cautions

- If '1' is written to the update bit (RTC_WTCR:UPDT) at the same time the update completes, the update bit (RTC_WTCR:UPDT) is set to '0'

- If the peripheral clock is stopped after updating the Hour/Minute/Second counters using the update bit (RTC_WTCR:UPDT), read the Hour/Minute/Second Registers to confirm that they have been updated before stopping the peripheral clock

- To reset the Hour, Minute, Second and Sub-second counters before starting the RTC
  - Write '1' to the start bit (RTC_WTCR:ST)
  - Wait until RUN (RTC_WTSR:RUN) is set to '1'
  - Write '0' to the start bit (RTC_WTCR:ST)

- The RTC module Sub-Second Register stores the reload value for the Sub-second counter. This value is reloaded after the Sub-second counter reaches '0'. When modifying the Sub-Second Register, ensure that a reload operation is not performed during the write operation. However, if a Sub-Second Register update is done immediately after an RTC second interrupt, there should be enough time to securely modify the registers until the next reload operation (next second interrupt) even if RTC_WTCR:ST is not set to '0' and the module is in operation

- If a reload has occurred during the updating of the Sub-Second Register (RTC_WTBR), an unexpected value may be reloaded to the Sub-second counter. Therefore, the Sub-Second Register (RTC_WTBR) should be updated with the start bit (RTC_WTCR:ST) set to '0'

- When updating the Hour/Minute/Second registers using the RTC_WTCR:ST bit, the following must be taken into account: The new value is written into the registers with the rising edge of the RUN bit. This RUN bit is clocked by the CLKRTC. To ensure that the update is done properly, write the new values into the registers, set RTC_WTCR:ST to '0', wait for the RUN bit to go low and then start the circuit again by setting RTC_WTCR:ST to '1'. RUN will go low at the second rising edge of the CLKRTC after RTC_WTCR:ST has been set to '0'. It will rise again at the second rising edge of CLKRTC after RTC_WTCR:ST has been set to '1'. If this operation is to be repeated several times directly after each other, wait for RUN to go high before setting RTC_WTCR:ST to low again

- If a carry operation occurs (e.g. Second counter overflows and Minute counter increments) while reading the Real Timer Register (RTC_WRT), inconsistent values may be read. To avoid this, the interrupts (RTC_WINS:SUBSEC, RTC_WINS:SEC, RTC_WINS:MIN, RTC_WINS:HOUR, RTC_WINS:DAY) should be used to read the time (HH/MM/SS)

- Register(RTC_WTSR:RUNC)value is changed asynchronously with the bus clock. Therefore register value should read more than once.

- Changing the below register during the operation of calibration is prohibited.
  1. RTC_WTCR.CCKSEL

2. RTC_CNTPCAL.CNTPCAL
3. RTC_DURMW.DURMW
4. RTC_CALTRG.CALTRG

- If a calibration function is used, there is limit to clock frequency of RTC clock and CLK_SYSC0H clock.
  RTC Clock * 2 < CLK_SYSC0H clock
  Consider the accuracy of clock.

CHAPTER 11: CR Calibration

This chapter explains CR calibration.

1. **Overview**
2. **Configuration**
3. **Explanation of Operation**
4. **Registers**

CRCALIB-TXXPT03P01R01L06-E1-XX

1. Overview

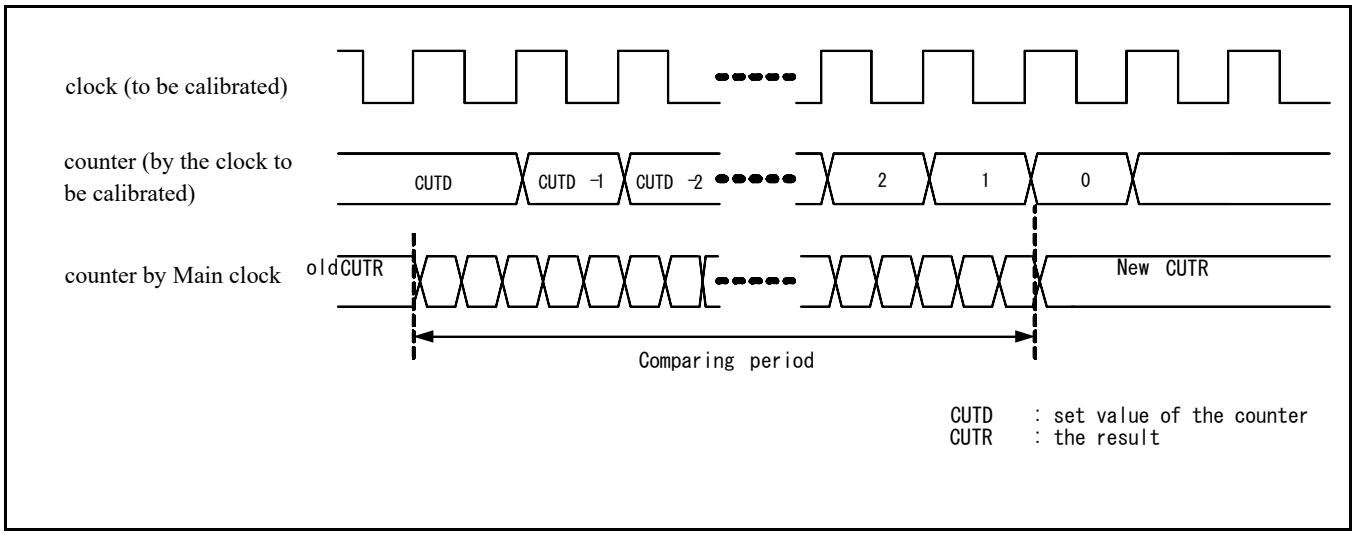
This section provides an overview of CR calibration.

The high-speed CR oscillator can correct frequencies by configuring trimming. A trimming value can be determined from the calculation of a count value for correcting the frequency of the high-speed CR oscillation circuit.

CR Oscillation Correction

Measurement of a clock error can be done by driving the counter driven by the main clock and measuring it for the set period of the counter driven by the CR clock (Figure 1-1).

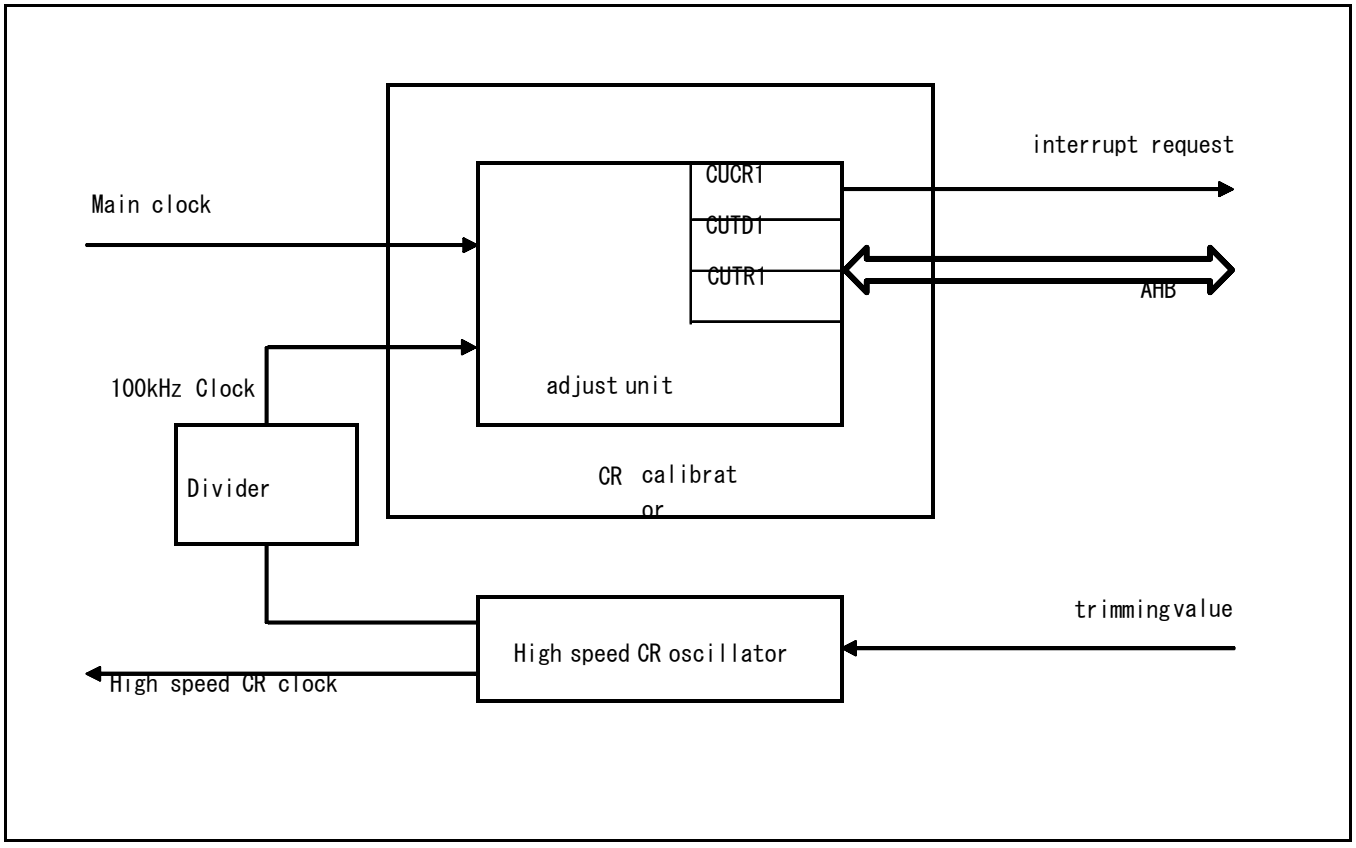
Figure 1-1 Comparison of Counters Driven by Each Clock



2. Configuration

This section explains the configuration for CR calibration.

Figure 2-1 CR Calibration Configuration Diagram



Note:

- *For details on the setting of a trimming value, see the description of the high-speed CR oscillation control register (SYSC_CRCNTR) in the following chapter: "Clock System".*

3. Explanation of Operation

This section explains the operation for CR calibration.

This explanation shows measurement of the CR clock frequency (error measurement). The measurement uses the following registers. For details, see Section 4.

- Correction unit control register (CUCR1)
  Controls the correction unit.
- CR clock timer data register 1 (CUTD1)
  Stores the set value of the measurement period. (CR clock driving)
- Main oscillation timer data register (CUTR1)
  Stores the main oscillation timer count results.

3.1. CR Clock Frequency Calculation

Frequency Calculation Procedure

1. Set the CR clock timer data register 1 (CUTD1).
2. Set "1" in the interrupt enable bit (CUCR1.INTEN).
3. Set "1" in the correction start bit (CUCR1.STRT).
4. Run a loop to wait for an interrupt.
5. Generate an interrupt.
6. Read the main oscillation timer data register 1 (CUTR1).
7. The ratio of "main clock frequency:CR clock frequency" can be calculated with "CUTR1:CUTD1".

3.2. CR Clock Frequency Correction

This section shows frequency correction procedures.

Perform frequency correction in the following 2 steps by using the TRC bits for low-resolution trimming and TRF bits for high-resolution trimming.

Step 1: TRC Frequency Trimming (Low-Resolution Adjustment)

1. Set 0b11111 in the TRF bits.
2. Set 0b00000 in the TRC bits to measure the frequency. Let the measured frequency be FC0.
3. Set 0b11111 in the TRC bits to measure the frequency. Let the measured frequency be FC31.
4. Calculate the TRC code according to the following formula.
   TC0=1/FC0, TC31=1/FC31
   CODE = (TC0 - 10*) / ((TC0 - TC31) / 31) (Round down to an integer.)
   *: 10[us] (Cycle time for 100kHz)
5. Use the code calculated by the above formula to measure the frequency.
   If the measured value is 100 kHz or higher, proceed to Step 2.
   If the measured value is 100kHz or lower, add 1 to the calculated code and repeat the step of 5.

Step 2: TRF Frequency Trimming (High-Resolution Adjustment)

1. Set the TRC code calculated in Step 1.
2. Set 0b00000 in the TRF bits to measure the frequency. Let the measured frequency be FF0.
3. Set 0b11111 in the TRF bits to measure the frequency. Let the measured frequency be FF31.
4. Calculate the TRF code according to the following formula.
   TF0 = 1 / FF0, TF31 = 1 / FF31
   CODE = (TF0 - 10) / ((TF0 - TF31) / 31) (Round down to an integer.)
   *: 10[us] (Cycle time for 100kHz)
5. Use the code calculated by the above formula to measure the frequency. Let the measured frequency be FA.
   If the measured value is 100 kHz or higher, perform the step of 11.
   If the measured value is 100 kHz or lower, perform the step of 12.
6. Subtract 1 from the calculated code and use it to measure the frequency. Let the measured frequency be FB.
   If | FA - 100kHz | < | FB - 100kHz | is true, set the calculated code and complete the correction.
   If | FA - 100kHz | > | FB - 100kHz | is true, replace the frequency of FB with FA and then repeat the step of 11.
7. Add 1 to the calculated code and use it to measure the frequency. Let the measured frequency be FB.
   If | 100kHz - FA | < | 100kHz - FB | is true, set the calculated code and complete the correction.
   If | 100kHz - FA | > | 100kHz - FB | is true, replace the frequency of FB with FA and then repeat the step of 12.

4. Registers

This section explains the CR calibration registers.

Table 4-1 lists the registers.

Table 4-1 Register List

Abbreviated Register Name	Register Name	Reference
CU_CUCR1	Correction unit control register 1	4.1
CU_CUTD1	CR oscillation timer register 1	4.2
CU_CUTR1	Main oscillation timer register 1	4.3
CU_CUCRC1	Correction unit control clear register 1	4.4

Table 4-2 shows a register map.

Table 4-2 Register Map

Offset	Register Name
0x0000_0000	CU_CUCR1 11111111_00000000
0x0000_0002	CU_CUTD1 11000011_01010000
0x0000_0004	CU_CUTR1 00000000_00000000_00000000_00000000
0x0000_0008	CU_CUCRC1 00000000_00000000_00000000_00000000

4.1. Correction Unit Control Register 1 (CU_CUCR1) (Calibration Unit Control Register 1)

This register sets the start of correction of the CR oscillation correction unit, and the clearing and enabling of interrupts.

Bit	15	14	13	12	11	10	9	8
Field	Reserved							
R/W Attribute	R1,WX							
Protection Attribute	-							
Initial Value	11111111							

Bit	7	6	5	4	3	2	1	0
Field	Reserved	Reserved		STRT	Reserved		INT	INTEN
R/W Attribute	R/W0	R0,WX		R,W	R0,WX		R,W	R/W
Protection Attribute	-							
Initial Value	0	00		0	00		0	0

[bit15:8] Reserved: Reserved Bits

[bit7] Reserved: Reserved Bit

Always write "0" to this bit. Normal operation is not guaranteed when "1" is written.

[bit6:5] Reserved: Reserved Bits

[bit4] STRT (calibration STaRT): Correction Start

This bit starts the 100 kHz drive counter generated from the main clock and CR clock. The INT bit is set to "1" at the completion of comparison.

If "0" is set, the comparison is aborted. Writing "1" during the comparison does not have any effect. The bit is cleared to "0" when the comparison is completed.

Bit	Description
0	Abort comparison.
1	Start comparison (during the comparison).

[bit3:2] Reserved: Reserved Bits

[bit1] INT (calibration INTerrupt): Interrupt

This bit is set to "1" at the completion of comparison. If the INTEN bit setting is "1", an interrupt is generated. Writing "0" clears it. If the INTC bit setting for CU_CUCRC1 register is "1", this bit will be cleared by "0".

[bit0] INTEN (calibration INTerrupt ENable): Enabling Interrupts

This bit sets whether to generate an interrupt for an INT bit setting.

Bit	Description
0	Disable interrupts.
1	Enable interrupts.

4.2. CR Clock Timer Data Register 1 (CU_CUTD1) (Calibration Unit Timer Data Register 1)

This register sets the drive period of the counter driven by the CR clock.

Bit	15	0
Field	TDD	
R/W Attribute	R/W	
Protection Attribute	-	
Initial Value	11000011_01010000	

[bit15:0] TDD (Timer Data Data): Timer Data

These bits set the comparison period with a number of CR clocks.

bit15:0	Description
	Number of CR Clocks
0x0000	65536
0x0001	1
0x0002	2
0x0003	3
…	…
0xC350	50000
…	…
0xFFFD	65533
0xFFFE	65534
0xFFFF	65535

4.3. Main Oscillation Timer Data Register 1 (CU_CUTR1) (Calibration Unit Timer Result Register 1)

This register displays the number of counters driven by the main clock during the set period of CUTD1.

Bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	-							
Initial Value	00000000							

Bit	23							0
Field	TDR							
R/W Attribute	R,WX							
Protection Attribute	-							
Initial Value	00000000_00000000_00000000							

[bit31:24] Reserved: Reserved Bits

[bit23:0] TDR (Timer Data Register): Timer Data

These bits display the number of counts that were counted in the comparison period. Read them after the comparison is completed.

If the value read during the comparison is undefined, writing with this value is invalid.

bit23:0	Description
	Number of Main Clock Counts
0x00_0000	0
0x00_0001	1
0x00_0002	2
0x00_0003	3
…	…
0xFF_FFFD	16777213
0xFF_FFFE	16777214
0xFF_FFFF	16777215

4.4. Correction Unit Control Clear Register 1 (CU_CUCRC1) (Calibration Unit Control Register Clear Register 1)

This register clears interrupts of the CR oscillation correction unit.

Bit	31	8
Field	Reserved	
R/W Attribute	R0,WX	
Protection Attribute	-	
Initial Value	00000000_00000000_00000000	

Bit	7	6	5	4	3	2	1	0
Field	Reserved						INTC	Reserved
R/W Attribute	R0,WX						R0,W	R0,WX
Protection Attribute	-							
Initial Value	000000						0	0

[bit31:2] Reserved: Reserved Bits

[bit1] INTC (INT Clear): Interrupt Clear

The read value is "0". If "1" is written, the INT bit in the CUCR1 register is cleared.

Bit	Description
0	No effect
1	Clear interrupts.

[bit0] Reserved: Reserved Bit

CHAPTER 12: Backup RAM Interface

This chapter provides an overview and explanation of the configuration, operation, and registers of the interface between Backup RAMs.

1. **Overview**
2. **Configuration**
3. **Explanation of Operation**
4. **Setting Procedure**
5. **Registers**
6. **Precautions for Using Device**

BURAM-TXXPT03P01R01L06-E1-XX

1. Overview

This section explains the features of the Backup RAM interface.

Target RAMs

- Backup RAM0: size is product specific
- Backup RAM1: size is product specific

RAMECC Function

RAMECC detects byte errors of up to 2 bits for data being read from and written to RAM. When RAMECC detects 1-bit errors, it corrects those errors.

- Interrupt function
  - RAMECC detects double-bit errors and generates RAM double-bit error interrupt signals.
  - RAMECC detects single-bit errors and generates RAM single-bit error interrupt signals.

- Test mode
  - Generation of pseudo-errors
  - ECC generation function disabled
  - ECC test function disabled
  - Access to the ECC area enabled

RAM Diagnosis Function

The RAM diagnosis function diagnoses and initializes the RAM. The following RAM diagnoses are selected and applied. (More than one can be selected.)

- Unique (as unique data, address information (below) is written in 52 bits containing the ECC bit.)
  - Address information

    ((address & 0x0000000F) << 48)
    + ((address & 0x000000FF) << 40)
    + ((address & 0x000000FF) << 32)
    + ((address & 0x000000FF) << 24)
    + ((address & 0x000000FF) << 16)
    + ((address & 0x000000FF) << 8)
    + (address & 0x000000FF)
- Checker
- March (performed with all "0" and then with all "1")

- Interrupt function
  - Generates an interrupt signal triggered by diagnosis end. (RAM diagnosis end interrupt)
  - Generates an interrupt signal when an error is detected. (RAM diagnosis error interrupt)

RAM Initialization

The following types of RAM initialization are selected and applied.

- Writing all "0"
- Writing all "1"

- Interrupt function
  - Generates an interrupt signal triggered by initialization end. (RAM initialization end interrupt)

Bus Error Response

- Unprivileged write access
- Writing in lock state
- Access violation to an unlocked register
- Access to powered-off RAM
- Access to undefined or reserved area

2. Configuration

This section explains the configuration of Backup RAM.

Figure 2-1 Backup RAM Interface Block Diagram

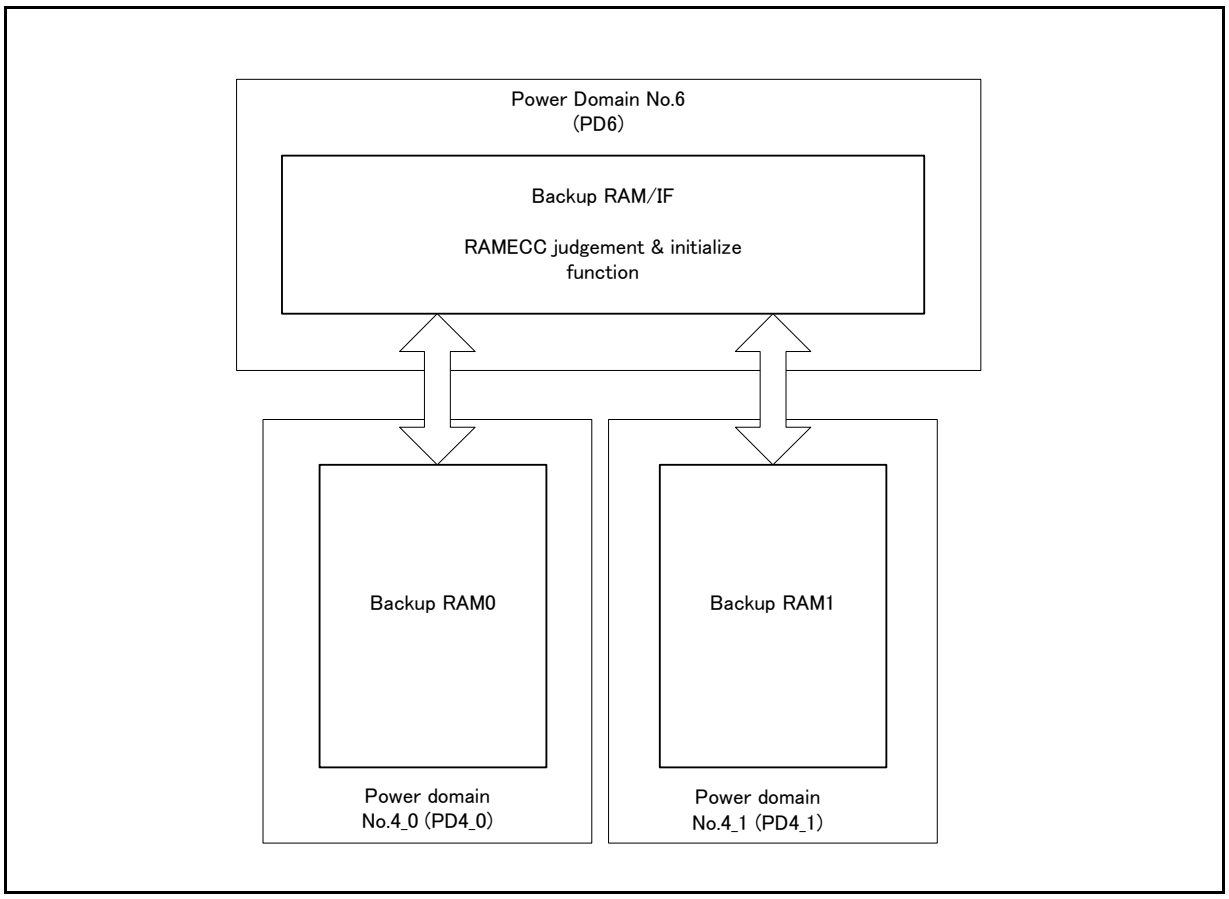


Figure 2-2 RAMECC Function Block Diagram

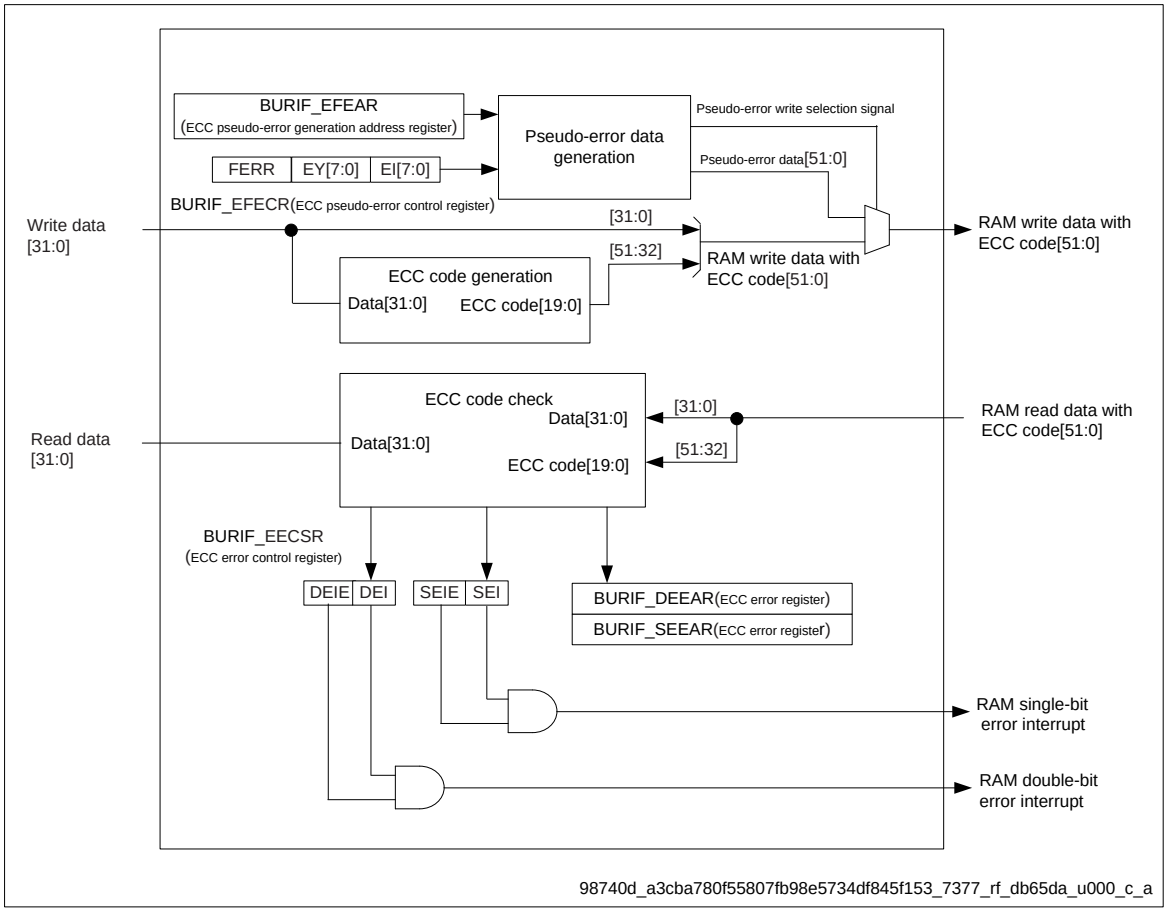
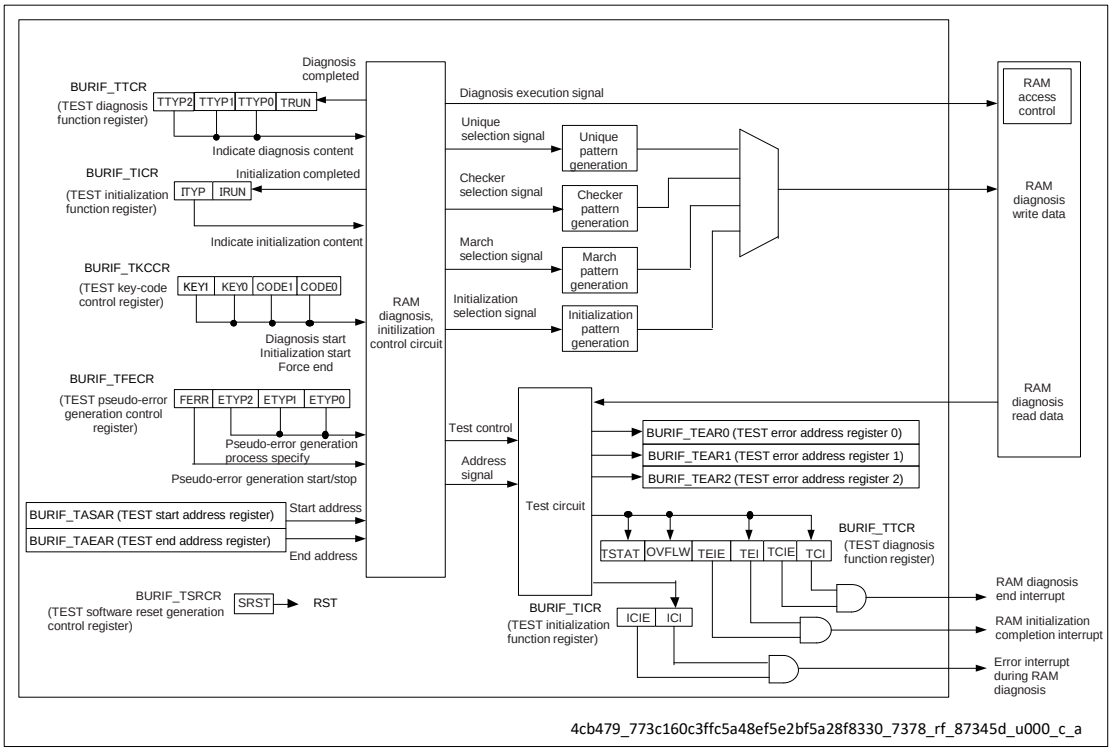


Figure 2-3 RAMECC Diagnosis Function Block Diagram



3. Explanation of Operation

This section explains the operation of the Backup RAM interface.

3.1. RAMECC Function
3.2. RAM Diagnosis
3.3. RAM Initialization
3.4. Required Number of Cycles
3.5. Bus Error Response

3.1. RAMECC Function

This section explains the RAMECC function.

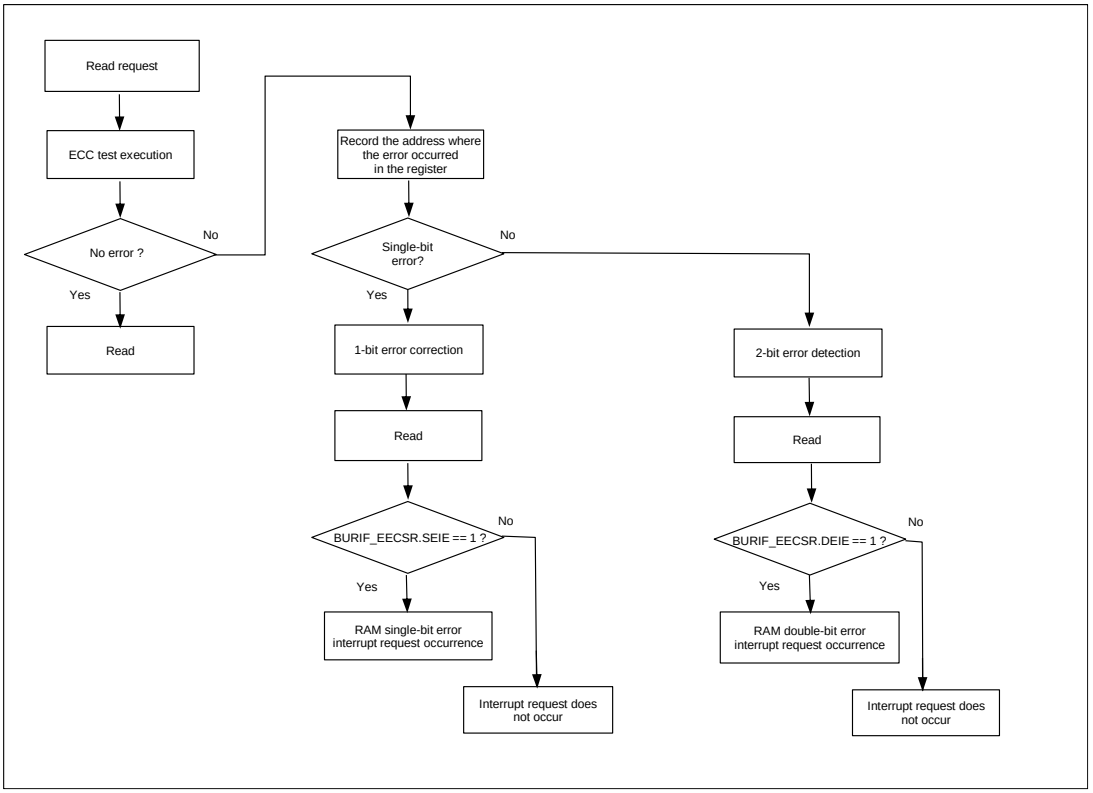
Generation of ECC

The RAMECC function is constantly running. (It does stop, however, when RAM diagnosis is being performed.) Upon the detection of a 1-bit error, the error occurrence address is held in the BURIF_SEEAR register. Upon the detection of a 2-bit error, the error occurrence address is held in the BURIF_DEEAR register. Error occurrence addresses that are currently held in the registers are not overwritten upon the subsequent detection of errors, and the initial values will be maintained.

The ECC code matrix records a set of 5 redundant bits for each byte as an ECC code.

RAM data [51:0]	[7:0]	[15:8]	[23:16]	[31:24]	[36:32]	[41:37]	[46:42]	[51:47]
data [31:0]	data [7:0]	data [15:8]	data [23:16]	data [31:24]	ECC [4:0]	ECC [9:5]	ECC [14:10]	ECC [19:15]

The operation flow chart is shown below.



Interrupt-Related Registers

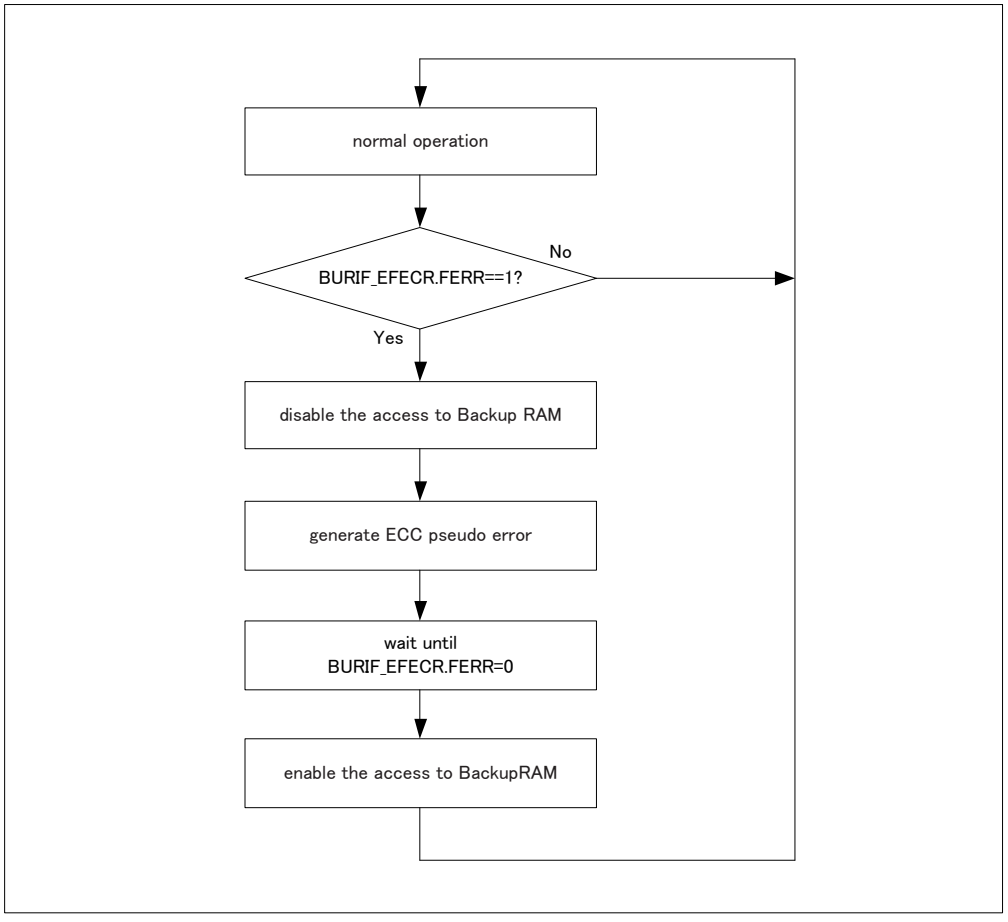
To generate an interrupt, write "1" to the interrupt generation enable bit (DEIE or SEIE) in the ECC error control register (BURIF_EECSR) in accordance with the purpose, and set the RAMECC interrupt vector. The interrupt request flag is not cleared automatically. For this reason, clear it by software using DEICLR or SEICLR of the event clear register (BURIF_EVENTCLR) before returning from the interrupt processing.

Test Mode

Pseudo-errors are generated for software debugging. ECC pseudo-errors are generated by the application of the following steps.

1. Specify the address for pseudo-error generation in the ECC pseudo-error generation address register (BURIF_EFEAR).
2. Set the byte and bit for pseudo-error generation in the ECC pseudo-error generation control register (BURIF_EFECR).
   (A) Specify the byte position of pseudo-error generation in EY[7:0]
   (B) Specify the bit position of pseudo-error generation in EI[7:0]
3. Write "1" to the FERR bit in the ECC pseudo-error generation control register (BURIF_EFECR).

The operation flow chart is shown below.



Erroneous data is deliberately written to the address specified by the ECC pseudo-error generation address register (BURIF_EFEAR) according to the contents of EY[7:0] and EI[7:0]. Data writing is followed by reading, at which point a pseudo-error is detected. The operation proceeds automatically after "1" is written to the FERR bit.

Note:

- *An error of 3 bits or more is detected as a single-bit error, which may result in an unintended correction. Accesses are blocked during ECC pseudo-error generation. Accesses should be attempted only after ECC pseudo-error processing has finished.*

Other test modes include the following functions.

- ECC generation function disabled
- ECC test function disabled
- Access to the ECC area enabled

To enable a test mode, use the ECC data path control register (BURIF_EDPCR) and the ECC test mode key code control register (BURIF_ECCTKCCR).

1. In the ECC test mode key code control register (BURIF_ECCTKCCR), write "1" to a desired test enable bit (TM_ESKPG, TM_ESKPC, or TM_EEACC), and then consecutively write 0x0, 0x1, 0x2, and 0x3 to the KEY bits in that order.
2. Check bits (TM_ESKPG, TM_ESKPC, and TM_EEACC) in the status register (BURIF_STATUS) to confirm that the setting of the ECC test mode key code control register (BURIF_ECCTKCCR) is correctly reflected.
3. Write "1" to the SKPG, SKPC, or EACC bit in the ECC data path control register (BURIF_EDPCR) to enable the test.

3.2. RAM Diagnosis

This section explains the RAM diagnosis.

RAM Diagnosis Procedure

RAM diagnosis consists of the following steps.

1. Unique (as unique data, address information (below) is written in 52 bits containing the ECC bit.)
   - Address information

   ((address & 0x0000000F) << 48)
   + ((address & 0x000000FF) << 40)
   + ((address & 0x000000FF) << 32)
   + ((address & 0x000000FF) << 24)
   + ((address & 0x000000FF) << 16)
   + ((address & 0x000000FF) << 8)
   + (address & 0x000000FF)
2. Checker
3. March (performed with all "0" and then with all "1")

Only the above order of steps is allowed. The diagnosis is performed according to the setting of the TTYP[2:0] bits in the TEST diagnosis function register (BURIF_TTCR). With the default setting, the unique and checker diagnoses are performed. The march diagnosis sets the RAM content to all "1" when finished. Other diagnoses leave some values in RAM. A read access without prior initialization or a write access results in an ECC error. Make sure to perform a read access after initialization or a write access.

The scope of RAM diagnosis is specified by the TEST start address register (BURIF_TASAR) and the TEST end address register (BURIF_TAEAR). A RAM diagnosis must be performed according to the following procedure.

1. Before starting a diagnosis, read TRUN in the TEST diagnosis function register (BURIF_TTCR) and IRUN in the TEST initialization function register (BURIF_TICR) to confirm that their values are "0".
   If BURIF_TTCR.TRUN or BURIF_TICR.IRUN is other than "0":
   - Wait until BURIF_TTCR.TRUN = 0, and then clear BURIF_TTCR.TCI.
   - Wait until BURIF_TICR.IRUN = 0, and then clear BURIF_TICR.ICI.

2. In the TEST key code control register (BURIF_TKCCR), consecutively write "02h", "42h", "82h", and "C2h," in that order, to start the diagnosis. When all the RAM diagnoses end, the TRUN bit of the TEST diagnosis function register (BURIF_TTCR) becomes "0" to exit the RAM diagnosis. The diagnosis results are retained in the TEST error address registers 0 to 2 (TEAR0 to TEAR2) and in the TEST diagnosis function register (BURIF_TTCR). The diagnosis data remains in the RAM. Moreover, RAM diagnosis is forcibly terminated by consecutively writing "00h", "40h", "80h", and "C0h," in that order, to the TEST key code

control register (BURIF_TKCCR). The RAM diagnosis is terminated even if it has not been completed. In this case, the diagnosis result is not guaranteed.

Interrupt-Related Registers

To generate an interrupt, write "1" to an interrupt generation enable bit (TEIE or TCIE) in the TEST diagnosis function register (BURIF_TTCR) in accordance with the purpose, and then set the RAMECC interrupt vector. The interrupt request flag is not cleared automatically. For this reason, clear it by software using TEICLR or TCICLR of the event clear register (BURIF_EVENTCLR) before returning from the interrupt processing.

Procedure for Generating a RAM Diagnosis Pseudo-Error

This function deliberately generates a pseudo-error for software debugging.

A RAM diagnosis pseudo-error is generated by applying the following steps.

1. Set the error type in the pseudo-error generation control register (BURIF_TFECR).
   (A) Set a diagnosis pattern in BURIF_TFECR.ETYP[2:0] for pseudo-error generation.
   (B) Specify a diagnosis pattern for pseudo-error generation by writing BURIF_TFECR.FERR = 1.
2. Set the start of diagnosis processing in the TEST diagnosis function register (BURIF_TTCR).
   (A) Set the diagnosis pattern to be applied in BURIF_TTCR.TTYP[2:0].
   (B) In the TEST key code control register (BURIF_TKCCR), consecutively write "02h", "42h", "82h", and "C2h," in that order, to set the diagnosis pattern using the operation start procedure.

3.3. RAM Initialization

This section explains RAM initialization.

Operation for RAM Initialization

RAM initialization is done in either of the following ways.

- Writing all "0" (default)
- Writing all "1"

This is specified by the ITYP bit in the TEST initialization function register (BURIF_TICR). The value corresponding to the write value is written in the ECC area. The scope of RAM initialization is specified by the TEST start address register (BURIF_TASAR) and the TEST end address register (BURIF_TAEAR). RAM initialization must be performed by applying the following procedure.

1. Before starting initialization, read TRUN in the TEST diagnosis function register (BURIF_TTCR) and IRUN in the TEST initialization function register (BURIF_TICR) to confirm that their values are "0".
   If BURIF_TTCR.TRUN or BURIF_TICR.IRUN is not "0":
   - Wait until BURIF_TTCR.TRUN = 0, and then clear BURIF_TTCR.TCI.
   - Wait until BURIF_TICR.IRUN = 0, and then clear BURIF_TICR.ICI.
2. Write to the TEST key code control register (BURIF_TKCCR) 4 times consecutively in the order of "01h", "41h", "81h", and then "C1h" to start the initialization. When RAM initialization ends, the IRUN bit in the TEST initialization function register (BURIF_TICR) goes to "0" and RAM initialization ends. RAM initialization is forcibly terminated by writing 4 times consecutively in the order of "00h", "40h", "80h", and then "C0h" in the TEST key code control register (BURIF_TKCCR). The RAM initialization is terminated even if it has not yet been completed. In this case, the initialization result is not guaranteed.

Interrupt-Related Registers

To generate an interrupt, write "1" to the interrupt generation enable bit (ICIE) in the TEST initialization function register (BURIF_TICR) in accordance with the purpose, and then set the interrupt vector. The interrupt request flag is not cleared automatically. For this reason, clear it by software using the event clear register (BURIF_EVENTCLR.ICICLR) before returning from the interrupt processing.

3.4. Required Number of Cycles

This section explains the required number of cycles.

In the example shown below, the number of cycles required for RAM diagnosis and initialization is calculated.

Backup RAM
16 K word address (64 KB)

RAM Diagnosis (Uniqueness)

For a 1-word address:

- Write (1 cycle)
- Read 1 (1 cycle)
- Read 2 (1 cycle)

The above processing is required for each word address (for 16K). The resulting number of cycles is as follows.

$$\underbrace{(\ 1\ }_{\text{write}} + \underbrace{1}_{\text{read1}} + \underbrace{1\ )}_{\text{read2}} \times \underbrace{16384}_{\text{16Kword}} + 1 = \underbrace{49153}_{\text{result}}$$

RAM Diagnosis (Checker)

For a 1-word address:

- Write 1 (1 cycle)
- Read 1 (1 cycle)

The above processing is required for each word address (for 16 K). In addition, 5 writes and 4 reads are required for a word address for the partial write function.

- Write 2 (1 x 5 cycles)
- Read 2 (2 x 4 cycles)

These cycles are required. Furthermore, the same processing is repeated with different data. This gives the total number of cycles for the following.

$$\underbrace{((\ 1\ }_{\text{write1}} + \underbrace{1\ )}_{\text{read1}} \times \underbrace{16384}_{\text{16Kword}} + 1 + \underbrace{5}_{\text{write2}} + \underbrace{8\ )}_{\text{read2}} \times \underbrace{2}_{\text{repeat}} = \underbrace{65564}_{\text{result}}$$

RAM Diagnosis (March)

Each word address requires 3 writes and 2 reads.

- Write (1 x 3 cycles)
- Read (2 x 2 cycles)

The above processing is required for each of the word addresses (16 K). Furthermore, the same processing is repeated with different data. This leads to the total number of cycles for the following.

$$\underset{\text{write}}{3} + \underset{\text{read1}}{4} \times \underset{\text{16Kword}}{16384} \times \underset{\text{repeat}}{2} = \underset{\text{result}}{229376}$$

RAM Initialization

For a 1-word address:

- Write (1 cycle)

The above processing is required for each word address (for 16K). The resulting number of cycles is as follows.

$$\underset{\text{write}}{1} \times \underset{\text{16Kword}}{16384} = \underset{\text{result}}{16384}$$

3.5. Bus Error Response

This section explains the bus error response.

The Backup RAM interface returns a bus error under the following conditions:

1. Unprivileged write access to the Backup RAM interface register
2. Write access to the Backup RAM interface register in the lock state (BURIF_STATUS.LOCKSTATUS=1)
3. Write access to a value other than the unlock and lock values in the BURIF_UNLOCK register
4. Write access other than 32-bit access to the BURIF_UNLOCK register
5. Access to powered-off Backup RAM
6. Write access to an undefined or reserved space

4. Setting Procedure

This section describes the setting procedure for the Backup RAM interface registers.

Figure 4-1 Register Setting Flow

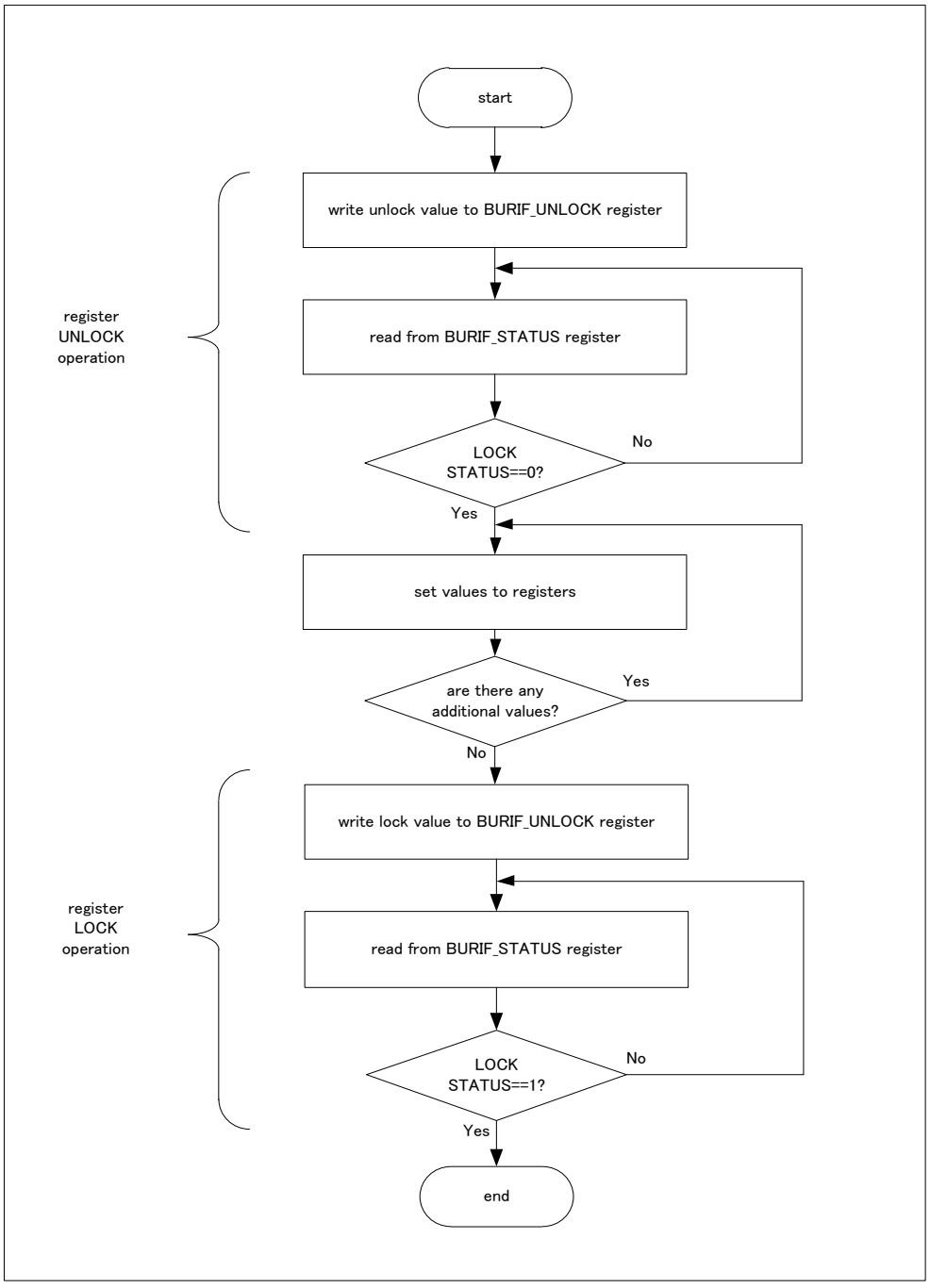


Figure 4-2 Setting Flow of ECC Test Mode Key Code Control Register

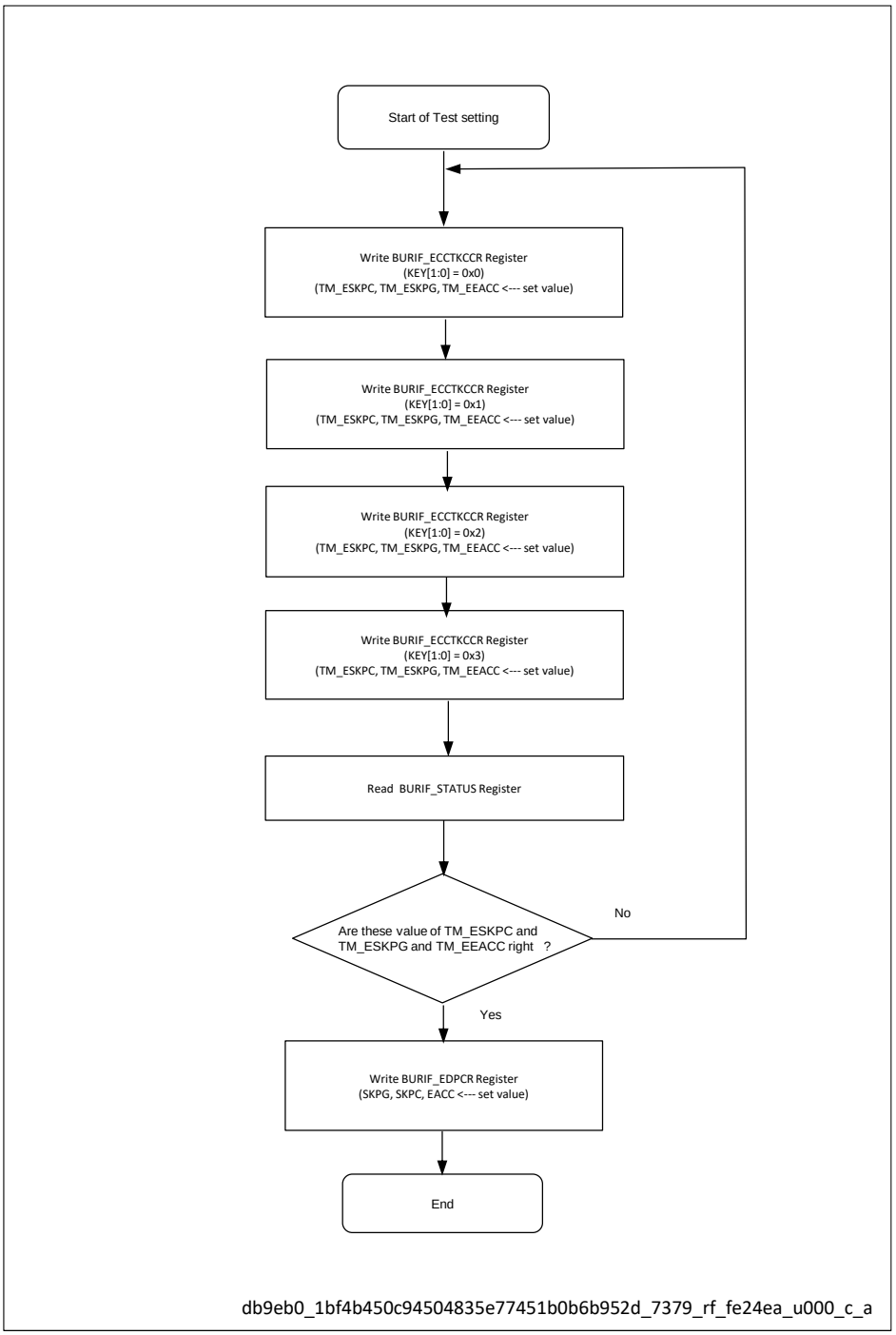


Figure 4-3 Setting Flow of RAM Self Diagnosis

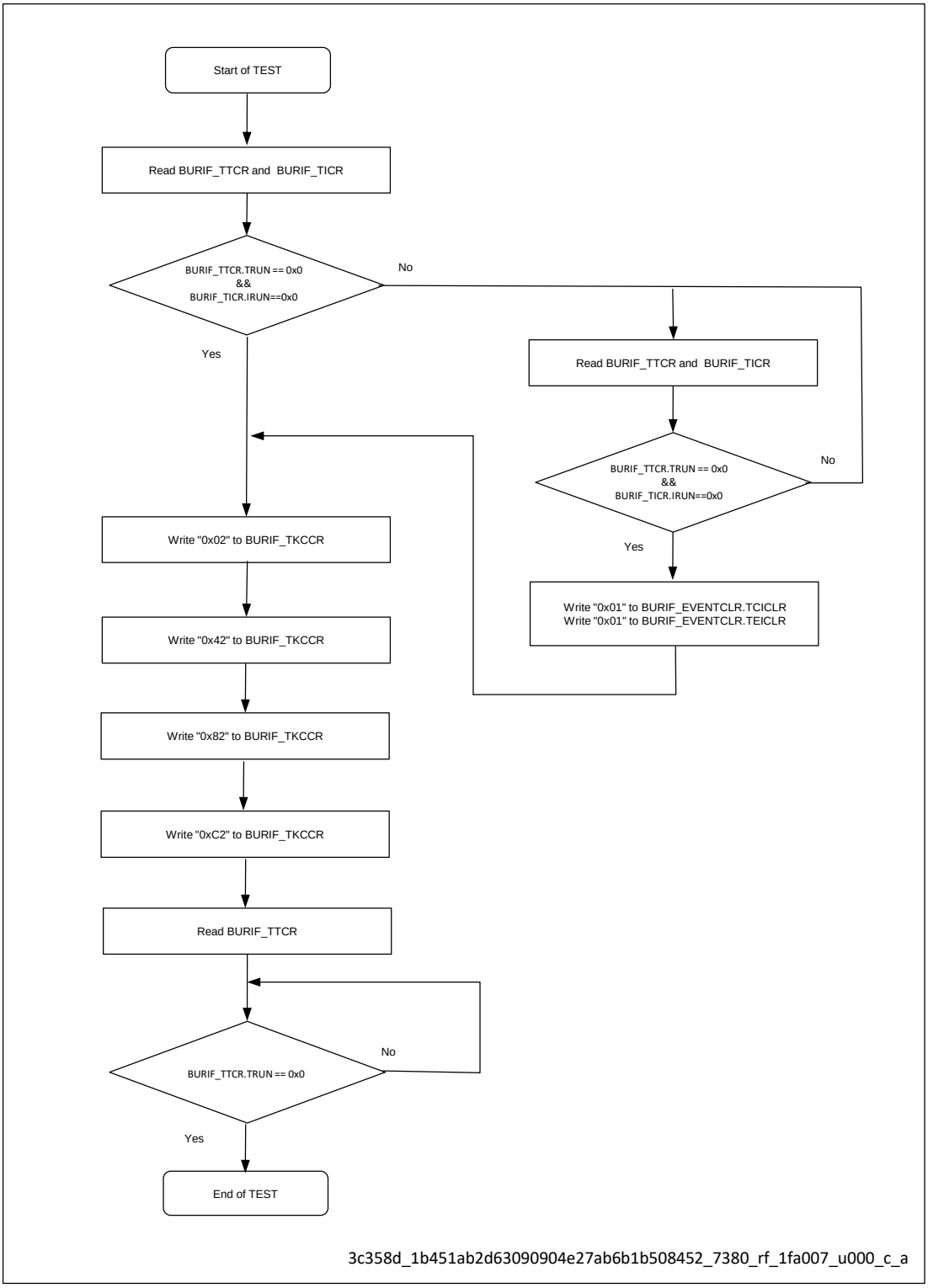
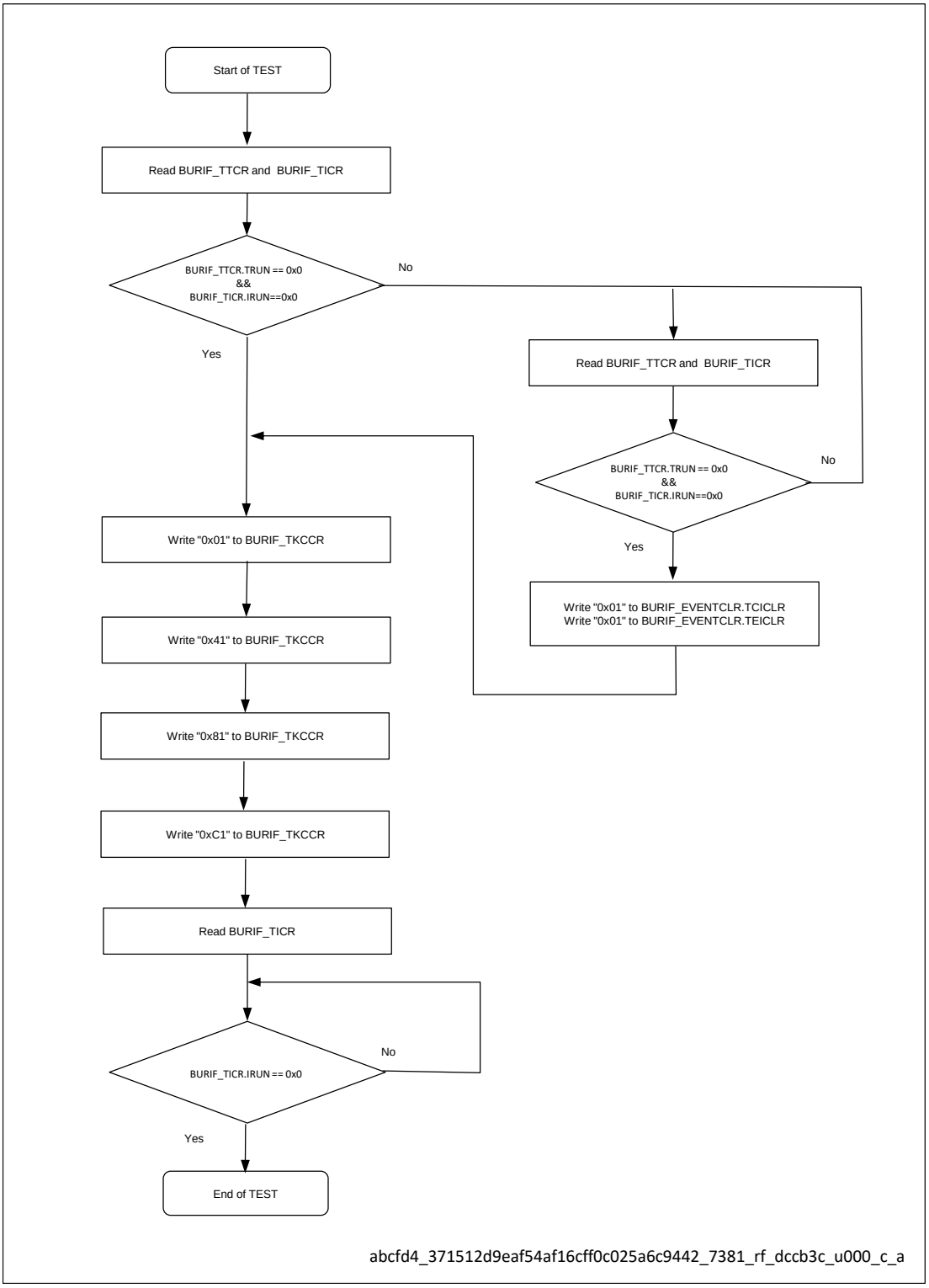


Figure 4-4 Setting Flow of RAM Initialization



5. Registers

This chapter provides a list of the Backup RAM interface registers.

Table 5-1 Backup RAM Interface Registers

Abbreviated Register Name	Register Name	Reference
BURIF_UNLOCK	BURAMIF unlock register	5.1
BURIF_STATUS	BURAMIF status register	5.2
BURIF_DEEAR	BURAMIF double-bit ECC error address register	5.3
BURIF_SEEAR	BURAMIF single-bit ECC error address register	5.4
BURIF_EFEAR	BURAMIF ECC pseudo-error generation address register	5.5
BURIF_EECSR	BURAMIF ECC error control register	5.6
BURIF_EFECR	BURAMIF ECC pseudo-error generation control register	5.7
BURIF_EDPCR	BURAMIF ECC data path control register	5.8
BURIF_ECCTKCCR	BURAMIF ECC test mode key code control register	5.9
BURIF_TEAR0	BURAMIF TEST error address register 0	5.10
BURIF_TEAR1	BURAMIF TEST error address register 1	5.11
BURIF_TEAR2	BURAMIF TEST error address register 2	5.12
BURIF_TASAR	BURAMIF TEST start address register	5.13
BURIF_TAEAR	BURAMIF TEST end address register	5.14
BURIF_TTCR	BURAMIF TEST diagnosis function register	5.15
BURIF_TICR	BURAMIF TEST initialization function register	5.16
BURIF_TFECR	BURAMIF TEST pseudo-error generation control register	5.18
BURIF_TKCCR	BURAMIF TEST key code control register	5.19
BURIF_TSRCR	BURAMIF TEST software reset generation control register	5.17
BURIF_EVENTCLR	BURAMIF event clear register	5.20

Table 5-2 Register Memory Layout

Address	+3	+2	+1	+0
0x0000_0000	BURIF_UNLOCK 00000000_00000000_00000000_00000000			
0x0000_0004	BURIF_STATUS 00000000_00000000_00000001_00000000			
0x0000_0008	BURIF_SEEAR 00000000_00000000		BURIF_DEEAR 00000000_00000000	
0x0000_000C	BURIF_EECSR 00000000	Reserved 00000000	BURIF_EFEAR 00000000_00000000	
0x0000_0010	BURIF_EDPCR 00000000	BURIF_EFECR 00000000_00000000_00000000		
0x0000_0014	Reserved 00000000	Reserved 00000000	Reserved 00000000	BURIF_ECCTKCCR 00000000
0x0000_0018	BURIF_TEAR0 00000000_00000000_00000000_00000000			
0x0000_001C	BURIF_TEAR1 00000000_00000000_00000000_00000000			
0x0000_0020	BURIF_TEAR2 00000000_00000000_00000000_00000000			
0x0000_0024	BURIF_TAEAR 0****111_11111111		BURIF_TASAR 00000000_00000000	
0x0000_0028	BURIF_TFECR 00000000	BURIF_TICR 00000000	BURIF_TTCR 00000000_00001100	
0x0000_002C	BURIF_TSRCR 00000000	Reserved 00000000	Reserved 00000000	BURIF_TKCCR 00000000
0x0000_0030	BURIF_EVENTCLR 00000000_00000000_00000000_00000000			

Notes:

- *An asterisk means that BURIF_TAEAR initial Value changes according to RAM size.*
  - 00000111_11111111 8KB
  - 00001111_11111111 16KB
  - 00011111_11111111 32KB
  - 00111111_11111111 64KB
  - 01111111_11111111 128KB

5.1. BURAMIF Unlock Register (BURIF_UNLOCK)

This register locks or unlocks write accesses to the Backup RAM interface registers. Writing the unlock value (0xACC55ECC) to this register enables write accesses to the registers. After setting the registers, writing the lock value (0x5ECCB10C) to this register disables write access to the registers. Writing to this register can be done only by privileged accesses in word units.

Bit	31	0
Field	UNLOCK[31:0]	
R/W Attribute	R0,W	
Protection Attribute	WP	
Initial Value	00000000_00000000_00000000_00000000	

[bit31:0] UNLOCK: Lock/Unlock Bits

UNLOCK	Lock/Unlock Bit
During Write Operation	These bits lock and unlock write accesses to the registers. 0xACC55ECC: LOCKSTATUS in BURIF_STATUS becomes "0", and write access to the Backup RAM interface registers is enabled. 0x5ECCB10C: LOCKSTATUS in BURIF_STATUS becomes "1", and write access to the Backup RAM interface registers is disabled. Other values: An error is generated. For details, see "Figure 4-1 Register Setting Flow"
During Read Operation	"0" is always read.

5.2. BURAMIF Status Register (BURIF_STATUS)

This register contains the LOCKSTATUS bit that indicates the lock/unlock status of the Backup RAM interface registers. This register also contains the status signals that indicate the ECC area access enable, ECC generation function disable, and ECC test function disable that are test modes.

Bit	31	8
Field	Reserved	LOCK STATUS
R/W Attribute	R0,W0	R,WX
Protection Attribute	-	
Initial Value	00000000_00000000_0000000	1

Bit	7	6	5	4	3	2	1	0
Field	Reserved					TM_ ESKPG	TM_ ESKPC	TM_ EEACC
R/W Attribute	R0,W0					R,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	00000					0	0	0

[bit31:9] Reserved: Reserved Bits

[bit8] LOCKSTATUS: Lock Status Bit

LOCK STATUS	Lock Status Bit
During Write Operation	No effect
During Read Operation	0: Backup RAM interface registers are unlocked. 1: Backup RAM interface registers are locked. (initial value)

[bit7:3] Reserved: Reserved Bits

[bit2] TM_ESKPG: ECC Generation Function Disable Status Bit

TM_ESKPG	ECC Generation Function Disable Status Bit
During Write Operation	No effect
During Read Operation	0: Disabling of the test-mode ECC generation function is not allowed (initial value). Writing "1" to BURIF_EDPCR.SKPG does not allow disabling of the test-mode ECC generation function. 1: Disabling of the test-mode ECC generation function is allowed. Writing "1" to BURIF_EDPCR.SKPG allows disabling of the test-mode ECC generation function.

[bit1] TM_ESKPC: ECC Test Function Disable Status Bit

TM_ESKPC	ECC Test Function Disable Status Bit
During Write Operation	No effect
During Read Operation	0: Disabling of the test-mode ECC test function is not allowed (initial value). Writing "1" to BURIF_EDPCR.SKPC does not allow disabling of the test-mode ECC test function. 1: Disabling of the test-mode ECC test function is allowed. Writing "1" to BURIF_EDPCR.SKPC allows disabling of the test-mode ECC test function.

[bit0] TM_EEACC: ECC Area Access Enable Status Bit

TM_EEACC	ECC Area Access Enable Status Bit
During Write Operation	No effect
During Read Operation	0: Access to the test-mode ECC area is not allowed (initial value). Writing "1" to BURIF_EDPCR.EACC does not enable access to the test-mode ECC area. 1: Access to test-mode ECC area is allowed. Writing "1" to BURIF_EDPCR.EACC enables access to the test-mode ECC area.

5.3. BURAMIF Double-Bit ECC Error Address Register (BURIF_DEEAR)

The double-bit ECC error address register holds the address of a 2-bit error detected by an ECC test of the Backup RAM.

Bit	15	14	13	12	11	10	9	8
Field	Reserved	ERR_ADDR[14:8]						
R/W Attribute	R0,W0	R,WX						
Protection Attribute	-							
Initial Value	0	0000000						

Bit	7	6	5	4	3	2	1	0
Field	ERR_ADDR[7:0]							
R/W Attribute	R,WX							
Protection Attribute	-							
Initial Value	00000000							

[bit15] Reserved: Reserved Bit

[bit14:0] ERR_ADDR: Double-Bit Error Occurrence Address Bits

These bits hold the address of a 2-bit error detected by the ECC test. The error occurrence address that is currently held in the register is not overwritten upon subsequent detections of the above condition, and the initial value will be maintained.

Notes:

- *ERR_ADDR is an offset in word units (RAM address).*
- *Obtain the absolute address by adding {ERR_ADDR << 2} to the base address.*
- *The effective address of ERR_ADDR changes according to RAM size.*
  - 64KB Effective address: ERR_ADDR[13:0]

5.4. BURAMIF Single-Bit ECC Error Address Register (BURIF_SEEAR)

The single-bit ECC error address register holds the address of a 1-bit error detected by the ECC test for Backup RAM.

Bit	15	14	13	12	11	10	9	8
Field	Reserved	ERR_ADDR[14:8]						
R/W Attribute	R0,W0	R,WX						
Protection Attribute	-							
Initial Value	0	0000000						

Bit	7	6	5	4	3	2	1	0
Field	ERR_ADDR [7:0]							
R/W Attribute	R,WX							
Protection Attribute	-							
Initial Value	00000000							

[bit15] Reserved: Reserved Bit

[bit14:0] ERR_ADDR: Single-Bit Error Occurrence Address Bits

These bits hold the address of a 1-bit error detected by the ECC test. The error occurrence address that is currently held in the register is not overwritten upon subsequent detections of the above condition, and the initial value will be maintained.

Notes:

- *ERR_ADDR is an offset in word units (RAM address).*
- *Obtain the absolute address by adding {ERR_ADDR << 2} to the base address.*
- *The effective address of ERR_ADDR changes according to RAM size.*
  - 64KB Effective address: ERR_ADDR[13:0]

5.5. BURAMIF ECC Pseudo-Error Generation Address Register (BURIF_EFEAR)

The ECC pseudo-error generation address register specifies the address that generates a pseudo-error for Backup RAM. Writing to this register is possible only when privileged access is available and BURIF_STATUS.LOCKSTATUS is "0".

Bit	15	14	13	12	11	10	9	8
Field	Reserved	ERR_ADDR[14:8]						
R/W Attribute	R0,W0	R/W						
Protection Attribute	WP							
Initial Value	0	0000000						

Bit	7	6	5	4	3	2	1	0
Field	ERR_ADDR[7:0]							
R/W Attribute	R/W							
Protection Attribute	WP							
Initial Value	00000000							

[bit15] Reserved: Reserved Bit

[bit14:0] ERR_ADDR: Pseudo-Single-Bit Error Occurrence Address Setting Bits

These bits specify the address that generates a pseudo ECC error generation for Backup RAM. When BURIF_EFECR.FERR = 1, these bits generate a write access to the address and deliberately include an error in the data to be written according to the BURIF_EFECR setting, which generates an ECC error.

Notes:

- *ERR_ADDR is an offset in word units (RAM address).*
- *Obtain the absolute address by adding {ERR_ADDR << 2} to the base address.*
- *The effective address of ERR_ADDR changes according to RAM size.*
  - 64KB Effective address: ERR_ADDR[13:0]

5.6. BURAMIF ECC Error Control Register (BURIF_EECSR)

This register indicates whether 1-bit error correction or 2-bit error detection has occurred during an ECC test for Backup RAM. This register specifies whether interrupts by 2-bit error detection are enabled. Writing to this register is possible only when privileged access is available and BURIF_STATUS.LOCKSTATUS is "0".

Bit	7	6	5	4	3	2	1	0
Field	Reserved				DEIE	DEI	SEIE	SEI
R/W Attribute	R0,W0				R/W	R,W	R/W	R,W
Protection Attribute	WP							
Initial Value	0000				0	0	0	0

[bit7:4] Reserved: Reserved Bits

[bit3] DEIE: Double-Bit-Error-Caused Interrupt Enable Bit

DEIE	Double-Bit-Error-Caused Interrupt Enable Bit
During Write Operation	0: Disable interrupts. 1: Enable Interrupts.
During Read Operation	The value set in this bit is read.

[bit2] DEI: Double-Bit Error Occurrence Bit

DEI	Double-Bit Error Occurrence Bit
During Write Operation	0: Clear this bit. 1: No effect.
During Read Operation	0: Double-bit error has not occurred. 1: Double-bit error has occurred.

[bit1] SEIE: Single-Bit-Error-Caused Interrupt Enable Bit

SEIE	Single-Bit-Error-Caused Interrupt Enable Bit
During Write Operation	0: Disable interrupts. 1: Enable Interrupts.
During Read Operation	The value set in this bit is read.

[bit0] SEI: Single-Bit Error Occurrence Bit

SEI	Single-Bit Error Occurrence Bit
During Write Operation	0: Clear this bit. 1: No effect.
During Read Operation	0: Single-bit error has not occurred. 1: Single-bit error has occurred.

5.7. BURAMIF ECC Pseudo-Error Generation Control Register (BURIF_EFECR)

The ECC pseudo-error generation control register specifies the content of the Backup RAM pseudo-error to be generated in the form of a generation byte and generation bit. Writing to this register is possible only when privileged access is available and BURIF_STATUS.LOCKSTATUS is "0".

Bit	23	22	21	20	19	18	17	16
Field	Reserved							FERR
R/W Attribute	R0,W0							R/W
Protection Attribute	WP							
Initial Value	0000000							0

Bit	15	14	13	12	11	10	9	8
Field	EY[7:0]							
R/W Attribute	R/W							
Protection Attribute	WP							
Initial Value	00000000							

Bit	7	6	5	4	3	2	1	0
Field	EI[7:0]							
R/W Attribute	R/W							
Protection Attribute	WP							
Initial Value	00000000							

[bit23:17] Reserved: Reserved Bits

[bit16] FERR: Pseudo-Error Generation Enable Bit

This bit enables the generation of pseudo-ECC-errors.

FERR	Pseudo-Error Generation Enable Bit
During Write Operation	0: No effect. When generation of pseudo-ECC-errors is completed, this bit is automatically cleared to "0" by the hardware. 1: Enable generation of pseudo-ECC-errors.
During Read Operation	0: Generation of pseudo-ECC-errors is the disabled status. (Normal operation) 1: Generation of pseudo-ECC-errors is the enabled status.

[bit15:8] EY7 to EY0: Pseudo-Error Generation Byte Setting Bits

These bits specify the byte position of the pseudo-ECC-error generation in Backup RAM.

EY7 to EY0	Target Bytes in RAM
EY0	RAM data [7:0]
EY1	RAM data [15:8]
EY2	RAM data [23:16]
EY3	RAM data [31:24]
EY4	RAM data [36:32]
EY5	RAM data [41:37]
EY6	RAM data [46:42]
EY7	RAM data [51:47]

For example, if EY2 = 1 while the other bits are "0", the target byte for pseudo-error generation is RAM data [23:16], and the other data will not have errors. If EY2 = EY3 =1 while the other bits are "0", the target bytes for pseudo-error generation is RAM data [31:16].

[bit7:0] EI7 to EI0: Pseudo-Error Generation Bit Setting Bits

These bits specify the bit position of the pseudo-ECC-error generation in Backup RAM.

EI7 to EI0	Target bits in RAM
EI0	[0]
EI1	[1]
EI2	[2]
EI3	[3]
EI4	[4]
EI5	[5]
EI6	[6]
EI7	[7]

For example, if EY2 = 1 and EI4 = 1 while the other bits are "0", the target bit for pseudo-error generation is RAM data [20], and this single-bit error will be corrected. If EY2 = 1 and EI4 = EI7 = 1 while the other bits are "0", the target bits for pseudo-error generation are RAM data [23] and RAM data [20], and a double-bit error will be detected. If EY2 = EY3 = 1 and EI4 = 1 while the other bits are "0", the target bits for pseudo-error generation are RAM data [28] and RAM data [20], and single-bit errors in these bytes will be corrected.

5.8. BURAMIF ECC Data Path Control Register (BURIF_EDPCR)

The ECC data path control register controls disabling of the ECC generation function, disabling of the ECC test function, and enabling of access to the ECC area. Writing to this register is possible only when privileged access is available and BURIF_STATUS.LOCKSTATUS is "0".

Bit	7	6	5	4	3	2	1	0
Field	Reserved					SKPG	SKPC	EACC
R/W Attribute	R0,W0					R,W	R,W	R,W
Protection Attribute	WP							
Initial Value	00000					0	0	0

[bit7:3] Reserved: Reserved Bits

[bit2] SKPG: ECC Generation Function Disable Bit

This bit allows disabling of the ECC generation function.

SKPG	ECC Generation Function Disable Bit
During Write Operation	0: Use the ECC generation function. (Normal operation) 1: Do not use the ECC generation function.
During Read Operation	The value set in this bit is read.

This bit is effective only when BURIF_STATUS.TM_ESKPG = "H". When it is "L", writing is also disabled.

[bit1] SKPC: ECC Test Function Disable Bit

This bit allows disabling of the ECC test function.

SKPC	ECC Test Function Disable Bit
During Write Operation	0: Use the ECC test function. (Normal operation) 1: Do not use the ECC test function.
During Read Operation	The value set in this bit is read.

This bit is effective only when BURIF_STATUS.TM_ESKPC = "H". When it is "L", writing is also disabled.

[bit0] EACC: ECC Area Access Enable Bit

This bit enables access to the ECC area.

EACC	ECC Area Access Enable Bit
During Write Operation	0: The access area is the data area. (Normal operation) 1: The access area is the ECC area.
During Read Operation	The value set in this bit is read.

This bit is effective only when BURIF_STATUS.TM_EEACC = "H". When it is "L", writing is also disabled.

5.9. BURAMIF ECC Test Mode Key Code Control Register (BURIF_ECCTKCCR)

The ECC test mode key code control register is used to enable ECC pseudo-error generation for Backup RAM, access to the ECC area, and disabling of the ECC generation test function. Writing to this register is possible only when privileged access is available and BURIF_STATUS.LOCKSTATUS is "0".

Bit	7	6	5	4	3	2	1	0
Field	KEY[1:0]		Reserved			TM_ ESKPG	TM_ ESKPC	TM_ EEACC
R/W Attribute	R0,W		R0,W0			R0,W	R0,W	R0,W
Protection Attribute	WP							
Initial Value	00		000			0	0	0

[bit7:6] KEY: Key Code Control

KEY	Key Code Control Bit
During Write Operation	These bits control the key code. Writing values to these bits in the order of 0b00, 0b01, 0b10, and 0b11 enables the setting values of TM_ESKPG, TM_ESKPC, and TM_EEACC. For details on the procedure for setting these bits, see ","
During Read Operation	"0" is always read.

[bit5:3] Reserved: Reserved Bits

[bit2] TM_ESKPG: ECC Generation Function Skip Enable Bit

TM_ESKPG	ECC Generation Function Skip Enable Bit
During Write Operation	0: Enable the ECC generation function, regardless of the BURIF_EDPCR.SKPG value. Moreover, writing to BURIF_EDPCR.SKPG by software is disabled. 1: Enable skipping of the ECC generation function. Moreover, BURIF_EDPCR.SKPG= 1 enables skipping of the ECC generation function.
During Read Operation	"0" is always read.

[bit1] TM_ESKPC: ECC Test Function Skip Enable Bit

TM_ESKPC	ECC Test Function Skip Enable Bit
During Write Operation	0: Disable skipping of the ECC test function, regardless of the BURIF_EDPCR.SKPC value. Moreover, writing to BURIF_EDPCR.SKPC by software is disabled. 1: Enable skipping of the ECC test function. Moreover, BURIF_EDPCR.SKPC= 1 enables skipping of the ECC test function.
During Read Operation	"0" is always read.

[bit0] TM_EEACC: ECC Area Access Enable Bit

TM_EEACC	ECC Area Access Enable Bit
During Write Operation	0: Access only the DATA area, regardless of the BURIF_EDPCR.EACC value. Moreover, writing to BURIF_EDPCR.EACC by software is disabled. 1: Enable access to the ECC area. Moreover, BURIF_EDPCR.EACC= 1 enables access to the ECC area.
During Read Operation	"0" is always read.

Note:

- *Set the same values in the TM_ESKPG, TM_ESKPC and TM_EEACC bits while KEY bit operation is being performed. For details on the procedure for setting these bits, see "Figure 4-2 Setting Flow of ECC Test Mode Key Code Control Register"*

5.10. BURAMIF TEST Error Address Register 0 (BURIF_TEAR0)

TEST error address register 0 contains the address at which a RAM diagnosis error has occurred in Backup RAM. This register also contains the source flag indicating the diagnosis pattern in which the error has occurred.

Bit	31	30	29	28	27	26	25	24
Field	TER[2:0]			Reserved				
R/W Attribute	R,WX			R0,W0				
Protection Attribute	-							
Initial Value	000			00000				

Bit	23	22	21	20	19	18	17	16
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial Value	00000000							

Bit	15	14	13	12	11	10	9	8
Field	Reserved	ERR_ADDR[14:8]						
R/W Attribute	R0,W0	R,WX						
Protection Attribute	-							
Initial Value	0	0000000						

Bit	7	6	5	4	3	2	1	0
Field	ERR_ADDR[7:0]							
R/W Attribute	R,WX							
Protection Attribute	-							
Initial Value	00000000							

[bit31:29] TER: Diagnosis Error Factor Indication

TER	Diagnosis Error Factor Indication Bit
During Write Operation	Writing is disabled.
During Read Operation	These bits contain the diagnosis pattern in which an error has occurred during RAM diagnosis. ERR_ADDR[14:0] is effective only when any of these bits is set to "1". 0b001: Error occurred in the march diagnosis 0b010: Error occurred in the checker diagnosis 0b100: Error occurred in the unique diagnosis 0b000: No error occurred A RAM diagnosis start instruction triggers the hardware to initialize (clear to "000") these bits. Note: When any of these bits becomes "1", even if an error occurs while another diagnosis is being performed, the corresponding error source bit does not become "1".

[bit23:15] Reserved: Reserved Bits

[bit14:0] ERR_ADDR: Error Occurrence Address

ERR_ADDR	Error Occurrence Address Bit
During Write Operation	Writing is disabled.
During Read Operation	These bits contain the address at which an error occurred during RAM diagnosis. These bits are effective only when TER is other than "000". A RAM diagnosis start instruction triggers the hardware to initialize (clear to "0b0000000000000000") these bits.

Notes:

- *When any of the TER bits becomes "1", even if an error occurs while another diagnosis is being performed, the corresponding error source bit does not become "1".*
- *ERR_ADDR is an offset address in units of words (RAM address).*
- *To calculate the absolute address, add {ERR_ADDR << 2} to the base address.*
- *The effective address of ERR_ADDR changes according to RAM size.*
  - 64KB Effective address: ERR_ADDR[13:0]

5.11. BURAMIF TEST Error Address Register 1 (BURIF_TEAR1)

TEST error address register 1 contains the address of an error that has occurred during Backup RAM diagnosis only if the address differs from that in BURIF_TEAR0. This register also contains the source flag indicating the diagnosis pattern in which the error has occurred.

Bit	31	30	29	28	27	26	25	24
Field	TER[2:0]			Reserved				
R/W Attribute	R,WX			R0,W0				
Protection Attribute	-							
Initial Value	000			00000				

Bit	23	22	21	20	19	18	17	16
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial Value	00000000							

Bit	15	14	13	12	11	10	9	8
Field	Reserved	ERR_ADDR[14:8]						
R/W Attribute	R0,W0	R,WX						
Protection Attribute	-							
Initial Value	0	0000000						

Bit	7	6	5	4	3	2	1	0
Field	ERR_ADDR[7:0]							
R/W Attribute	R,WX							
Protection Attribute	-							
Initial Value	00000000							

[bit31:29] TER: Diagnosis Error Factor Indication

TER	Diagnosis Error Factor Indication Bit
During Write Operation	Writing is disabled.
During Read Operation	These bits contain the diagnosis pattern in which an error has occurred during RAM diagnosis. ERR_ADDR[14:0] is effective only when any of these bits is set to "1". 0b001: Error occurred in the march diagnosis 0b010: Error occurred in the checker diagnosis 0b100: Error occurred in the unique diagnosis 0b000: No error occurred A RAM diagnosis start instruction triggers the hardware to initialize (clear to "000") these bits. Note: When any of these bits becomes "1", even if an error occurs while another diagnosis is being performed, the corresponding error source bit does not become "1".

[bit23:15] Reserved: Reserved Bits

[bit14:0] ERR_ADDR: Error Occurrence Address

ERR_ADDR	Error Occurrence Address Bit
During Write Operation	Writing is disabled.
During Read Operation	These bits contain the address at which an error occurred during RAM diagnosis. These bits are effective only when TER is not "000". A RAM diagnosis start instruction triggers the hardware to initialize (clear to "0b000000000000000") these bits.

Notes:

- *When any of the TER bits becomes "1", even if an error occurs while another diagnosis is being performed, the corresponding error source bit does not become "1".*
- *ERR_ADDR is an offset address in units of words (RAM address).*
- *To calculate the absolute address, add {ERR_ADDR << 2} to the base address.*
- *The effective address of ERR_ADDR changes according to RAM size.*
  - 64KB Effective address: ERR_ADDR[13:0]

5.12. BURAMIF TEST Error Address Register 2 (BURIF_TEAR2)

TEST error address register 2 contains the address of an error that has occurred during Backup RAM diagnosis only if the address differs from those in BURIF_TEAR0 and BURIF_TEAR1. This register also contains the source flag indicating the diagnosis pattern in which the error has occurred.

Bit	31	30	29	28	27	26	25	24
Field	TER[2:0]			Reserved				
R/W Attribute	R,WX			R0,W0				
Protection Attribute	-							
Initial Value	000			00000				

Bit	23	22	21	20	19	18	17	16
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial Value	00000000							

Bit	15	14	13	12	11	10	9	8
Field	Reserved	ERR_ADDR[14:8]						
R/W Attribute	R0,WX	R,WX						
Protection Attribute	-							
Initial Value	0	0000000						

Bit	7	6	5	4	3	2	1	0
Field	ERR_ADDR[7:0]							
R/W Attribute	R,WX							
Protection Attribute	-							
Initial Value	00000000							

[bit31:29] TER: Diagnosis Error Factor Indication

TER	Diagnosis Error Factor Indication Bit
During Write Operation	Writing is disabled.
During Read Operation	These bits contain the diagnosis pattern in which an error has occurred during RAM diagnosis. ERR_ADDR[14:0] is effective only when any of these bits is set to "1". 0b001: Error occurred in the march diagnosis 0b010: Error occurred in the checker diagnosis 0b100: Error occurred in the unique diagnosis 0b000: No error occurred A RAM diagnosis start instruction triggers the hardware to initialize (clear to "000") these bits. Note: When any of these bits becomes "1", even if an error occurs while another diagnosis is being performed, the corresponding error source bit does not become "1".

[bit23:15] Reserved: Reserved Bits

[bit14:0] ERR_ADDR: Error Occurrence Address

ERR_ADDR	Error Occurrence Address Bit
During Write Operation	Writing is disabled.
During Read Operation	These bits contain the address at which an error occurred during RAM diagnosis. These bits are effective only when TER is not "000". A RAM diagnosis start instruction triggers the hardware to initialize (clear to "0b0000000000000000") these bits.

Notes:

- *When any of the TER bits becomes "1", even if an error occurs while another diagnosis is being performed, the corresponding error source bit does not become "1".*
- *ERR_ADDR is an offset address in units of words (RAM address).*
- *To calculate the absolute address, add {ERR_ADDR << 2} to the base address.*
- *The effective address of ERR_ADDR changes according to RAM size.*
  - 64KB Effective address: ERR_ADDR[13:0]

5.13. BURAMIF TEST Start Address Register (BURIF_TASAR)

The TEST start address register specifies the start address of RAM diagnosis and the initialization of Backup RAM. Writing to this register is possible only when privileged access is available and BURIF_STATUS.LOCKSTATUS is "0".

Bit	15	14	13	12	11	10	9	8
Field	Reserved	SADDR[14:8]						
R/W Attribute	R0,W0	R/W						
Protection Attribute	-	WP						
Initial Value	0	0000000						

Bit	7	6	5	4	3	2	1	0
Field	SADDR[7:0]							
R/W Attribute	R/W							
Protection Attribute	WP							
Initial Value	00000000							

[bit15] Reserved: Reserved Bit

[bit14:0] SADDR: RAM Diagnosis Start Address

SADDR	RAM Diagnosis Start Address Bit
During Write Operation	The start address for RAM diagnosis and initialization can be set.
During Read Operation	The start address for RAM diagnosis and initialization being set can be read.

Notes:

- *SADDR is an offset address in units of words (RAM address).*
- *To calculate the absolute address, add {SADDR << 2} to the base address.*
- *A value outside the Backup RAM area and a value such that BURIF_TASAR.SADDR > BURIF_TAEAR.EADDR cannot be set.*
- *The effective address of SADDR changes according to RAM size.*
  - 64KB Effective address: SADDR[13:0]

5.14. BURAMIF TEST End Address Register (BURIF_TAEAR)

The TEST end address register specifies the end address for RAM diagnosis and the initialization of Backup RAM. Writing to this register is possible only when privileged access is available and BURIF_STATUS.LOCKSTATUS is "0".

Bit	15	14	13	12	11	10	9	8
Field	Reserved	EADDR[14:8]						
R/W Attribute	R0,W0	R/W						
Protection Attribute	-	WP						
Initial Value	0	****111						

Bit	7	6	5	4	3	2	1	0
Field	EADDR[7:0]							
R/W Attribute	R/W							
Protection Attribute	WP							
Initial Value	11111111							

[bit15] Reserved: Reserved Bit

[bit14:0] EADDR: RAM Diagnosis End Address

EADDR	RAM Diagnosis End Address Bit
During Write Operation	The end address for RAM diagnosis and initialization can be set.
During Read Operation	The end address for RAM diagnosis and initialization being set can be read.

Notes:

- *EADDR is an offset in word units (RAM address).*
- *Obtain the absolute address by adding {EADDR << 2} to the base address.*
- *A value outside the Backup RAM area and a value such that BURIF_TASAR.SADDR > BURIF_TAEAR.EADDR cannot be set.*
- *The effective address of EADDR changes according to RAM size.*
- *The "*" display of the initial value means that the initial value of EADDR changes according to RAM size.*
  - 64KB Effective address: EADDR[13:0]
- *EADDR initial Value changes according to RAM size.*
  - 00111111_11111111 64KB

5.15. BURAMIF TEST Diagnosis Function Register (BURIF_TTCR)

The TEST diagnosis function register contains the specification of the RAM diagnosis content of Backup RAM, the diagnosis result, and the diagnosis status. Writing to this register is possible only when privileged access is available and BURIF_STATUS.LOCKSTATUS is "0".

Bit	15	14	13	12	11	10	9	8
Field	Reserved						TSTAT	OVFLW
R/W Attribute	R0,W0						R,WX	R,WX
Protection Attribute	WP							
Initial Value	000000						0	0

Bit	7	6	5	4	3	2	1	0
Field	TEIE	TEI	TCIE	TCI	TTYP			TRUN
R/W Attribute	R/W	R,W	R/W	R,W	R/W			R,WX
Protection Attribute	WP							
Initial Value	0	0	0	0	110			0

[bit15:10] Reserved: Reserved Bits

[bit9] TSTAT: RAM Diagnosis Error Detection

TSTAT	RAM Diagnosis Error Detection Bit
During Write Operation	No effect
During Read Operation	0: No error has been detected during RAM diagnosis. 1: An error has been detected during RAM diagnosis. A RAM diagnosis start instruction triggers the hardware to initialize (clear to "0") this bit.

[bit8] OVFLW: RAM Diagnosis Error Overflow

OVFLW	RAM Diagnosis Error Overflow Bit
During Write Operation	No effect
During Read Operation	0: RAM diagnosis errors have occurred at 3 or fewer addresses. 1: RAM diagnosis errors have occurred at 4 or more addresses. A RAM diagnosis start instruction triggers the hardware to initialize (clear to "0") this bit.

[bit7] TEIE: Enable Error Occurrence Interrupt during Diagnosis

TEIE	Diagnosis-Time Error Generation Interrupt Enable Bit
During Write Operation	0: Disable interrupts triggered by RAM diagnosis errors. 1: Enable interrupts triggered by RAM diagnosis errors. An interrupt occurs for TTCR.TEI=1.
During Read Operation	The enable/disable setting for error interrupts during diagnosis can be read.

[bit6] TEI: Error Occurrence during Diagnosis

TEI	Diagnosis-Time Error Generation Bit
During Write Operation	0: Clear this bit. 1: No effect.
During Read Operation	0: No error has occurred during RAM diagnosis. 1: An error has occurred during RAM diagnosis.

[bit5] TCIE: Diagnosis End Factor Interrupt Enable

TCIE	Diagnosis End Factor Interrupt Enable Bit
During Write Operation	0: Disable interrupts triggered by RAM diagnosis end. 1: Enable interrupts triggered by RAM diagnosis end. An interrupt occurs for TTCR.TCI= 1.
During Read Operation	The enable/disable setting for diagnosis end factor interrupts can be read.

[bit4] TCI: Diagnosis End

TCI	Diagnosis End Bit
During Write Operation	0: Clear this bit. 1: No effect.
During Read Operation	0: The diagnosis has not completed or has been stopped. 1: RAM diagnosis has been completed. This bit is not set for forcible stop by the key code.

[bit3:1] TTYP: RAM Diagnosis Contents Specification

TTYP	RAM Diagnosis Contents Specification Bit
During Write Operation	0bxx1: March diagnosis is performed. 0bxx0: March diagnosis is not performed. 0bx1x: Checker diagnosis is performed. 0bx0x: Checker diagnosis is not performed. 0b1xx: Unique diagnosis is performed. 0b0xx: Unique diagnosis is not performed. RAM diagnoses are performed in the following order. Unique diagnosis (The address itself is used as the data.) Checker diagnosis March diagnosis (all "0" and then all "1")
During Read Operation	The setting value for the RAM diagnosis type to be performed can be read.

[bit0] TRUN: RAM Diagnosis Operation Status

TRUN	RAM Diagnosis Operation Status Bit
During Write Operation	No effect
During Read Operation	0: RAM diagnosis is not in progress. 1: RAM diagnosis is in progress.

Note:

- *Set this register before starting RAM diagnosis.*

5.16. BURAMIF TEST Initialization Function Register (BURIF_TICR)

The TEST initialization function register contains the specification of the RAM initialization content for the Backup RAM, the initialization result, and the initialization status. Writing to this register is possible only when privileged access is available and BURIF_STATUS.LOCKSTATUS is "0".

Bit	7	6	5	4	3	2	1	0
Field	Reserved				ICIE	ICI	ITYP	IRUN
R/W Attribute	R0,W0				R/W	R,W	R/W	R,WX
Protection Attribute	WP							
Initial Value	0000				0	0	0	0

[bit7:4] Reserved: Reserved Bits

[bit3] ICIE: RAM Initialization End Factor Interrupt Enable

ICIE	RAM Initialization End Factor Interrupt Enable Bit
During Write Operation	0: Disable interrupts triggered by RAM initialization end. 1: Enable interrupts triggered by RAM initialization end.
During Read Operation	The enable/disable setting of initialization end factor interrupts can be read.

[bit2] ICI: RAM Initialization End

ICI	RAM Initialization End Bit
During Write Operation	0: Clear this bit. 1: No effect.
During Read Operation	0: Initialization has not been completed or has not been started. 1: RAM initialization has been completed. This bit is not set for forcible stop by the key code.

[bit1] ITYP: RAM Initialization Contents Specification

ITYP	RAM Initialization Contents Specification Bit
During Write Operation	0: Initialize with all "0's". 1: Initialize with all "1's".
During Read Operation	The initialization content specification setting can be read.

[bit0] IRUN: RAM Initialization Operation Status

IRUN	RAM Initialization Operation Status Bit
During Write Operation	No effect
During Read Operation	0: Initialization is not in progress. 1: Initialization is in progress.

5.17. BURAMIF TEST Software Reset Generation Control Register (BURIF_TSRCR)

The TEST software reset generation control register specifies the generation of software reset for initializing the internal circuit of the RAM diagnosis of Backup RAM. Writing to this register is possible only when privileged access is available and BURIF_STATUS.LOCKSTATUS is "0".

Bit	7	6	5	4	3	2	1	0
Field	SRST	Reserved						
R/W Attribute	R0,W	R0,W0						
Protection Attribute	WP							
Initial Value	0	0000000						

[bit7] SRST: Software Reset

SRST	Software Reset Bit
During Write Operation	0: No effect 1: Reset all circuits related to RAM diagnosis except for this register.
During Read Operation	"0" is always read.

[bit6:0] Reserved: Reserved Bits

5.18. BURAMIF TEST Pseudo-Error Generation Control Register (BURIF_TFECR)

The TEST pseudo-error generation control register generates pseudo-errors during the RAM diagnosis of Backup RAM. This register can specify the RAM diagnosis in which the error is generated. Writing to this register is possible only when privileged access is available and BURIF_STATUS.LOCKSTATUS is "0".

Bit	7	6	5	4	3	2	1	0
Field	Reserved				FERR	ETYP[2:0]		
R/W Attribute	R0,W0				R/W	R/W		
Protection Attribute	WP							
Initial Value	0000				0	000		

[bit7:4] Reserved: Reserved Bits

[bit3] FERR: RAM Diagnosis Pseudo-Error Generation Enable

FERR	RAM Diagnosis Pseudo-Error Generation Enable Bit
During Write Operation	0: Disable generation of pseudo-errors. (Normal operation) 1: Enable generation of pseudo-errors. Erroneous data is deliberately written in accordance with the ETYP setting.
During Read Operation	The enable/disable setting for the generation of pseudo-errors can be read.

[bit2:0] ETYP: Pseudo-Error Generation Processing Specification

ETYP	Pseudo-Error Generation Processing Specification Bit
During Write Operation	0bxx1: Pseudo-errors are generated during march diagnosis. 0bx1x: Pseudo-errors are generated during checker diagnosis. 0b1xx: Pseudo-errors are generated during unique diagnosis. 0b000: Generates no pseudo error.
During Read Operation	This bit can read the setting value for the diagnosis in which the pseudo-error is generated.

5.19. BURAMIF TEST Key Code Control Register (BURIF_TKCCR)

The TEST key code control register is used to start RAM diagnosis and initialization of Backup RAM, and to execute their forced termination. Writing to this register is possible only when privileged access is available and BURIF_STATUS.LOCKSTATUS is "0".

Bit	7	6	5	4	3	2	1	0
Field	KEY[1:0]		Reserved				CODE[1:0]	
R/W Attribute	R0,W		R0,W0				R/W	
Protection Attribute	WP							
Initial Value	00		0000				00	

[bit7:6] KEY: Key Code Control

KEY	Key Code Control Bit
During Write Operation	These bits control the key code. The operation specified in the CODE bits is performed by writing in the order of 0b00, 0b01, 0b10, and 0b11. If access is made to a register other than the key code control register during a key code operation, repeat the operation from the beginning. For details on the procedure for setting these bits, see "Figure 4-3 Setting Flow of RAM Self Diagnosis" and "Figure 4-4 Setting Flow of RAM Initialization".
During Read Operation	"0" is always read.

[bit5:2] Reserved: Reserved Bits

[bit1:0] CODE: Operation Specification

CODE	Operation Instruction Bit
During Write Operation	0b00: Forced termination 0b01: Activation of initialization 0b10: Activation of diagnosis 0b11: Setting prohibited
During Read Operation	The setting value for the operation instruction can be read.

Notes:

- *Set the same value for the CODE bits when KEY bits operation is in progress.*
- *Setting the value 0b11 to the CODE bits is prohibited.*

5.20. BURAMIF Event Clear Register (BURIF_EVENTCLR)

This register is used for clearing events relating to ECC error detection, RAM diagnosis, and RAM initialization of Backup RAM. Writing to this register is possible only when privileged access is available and BURIF_STATUS.LOCKSTATUS is "0".

Bit	31	30	29	28	27	26	25	24
Field	Reserved					DEICLR	Reserved	SEICLR
R/W Attribute	R0,W0					R0,W	R0,W0	R0,W
Protection Attribute	WP							
Initial Value	00000					0	0	0

Bit	23	22	21	20	19	18	17	16
Field	Reserved					ICICLR	Reserved	
R/W Attribute	R0,W0					R0,W	R0,W0	
Protection Attribute	WP							
Initial Value	00000					0	00	

Bit	15	14	13	12	11	10	9	8
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	WP							
Initial Value	00000000							

Bit	7	6	5	4	3	2	1	0
Field	Reserved	TEICLR	Reserved	TCICLR	Reserved			
R/W Attribute	R0,W0	R0,W	R0,W0	R0,W	R0,W0			
Protection Attribute	WP							
Initial Value	0	0	0	0	0000			

[bit31:27] Reserved: Reserved Bits

[bit26] DEICLR: Double-Bit Error Occurrence Clear Bit

DEICLR	Double-Bit Error Occurrence Clear Bit
During Write Operation	1: Clear the BURIF_EECSR. DEI bit. 0: No effect.
During Read Operation	"0" is always read.

[bit25] Reserved: Reserved Bit

[bit24] SEICLR: Single-Bit Error Occurrence Clear Bit

SEICLR	Single-Bit Error Occurrence Clear Bit
During Write Operation	1: Clear the BURIF_EECSR. SEI bit. 0: No effect.
During Read Operation	"0" is always read.

[bit23:19] Reserved: Reserved Bits

[bit18] ICICLR: RAM Initialization End Clear Bit

ICICLR	RAM Initialization End Clear Bit
During Write Operation	1: Clear the BURIF_TICR. ICI bit. 0: No effect.
During Read Operation	"0" is always read.

[bit18:7] Reserved: Reserved Bits

[bit6] TEICLR: Error Occurrence during Diagnosis Clear Bit

TEICLR	Error Occurrence During Diagnostic Bit
During Write Operation	1: Clear the BURIF_TTCR. TEI bit. 0: No effect.
During Read Operation	"0" is always read.

[bit5] Reserved: Reserved Bit

[bit4] TCICLR: Diagnosis End Clear Bit

TCICLR	Diagnosis End Bit
During Write Operation	1: Clear the BURIF_TTCR. TCI bit. 0: No effect.
During Read Operation	"0" is always read.

[bit3:0] Reserved: Reserved Bits

6. Precautions for Using Device

This section explains precautions for using the Backup RAM.

- Power domain control for Backup RAM depends on profile setting. To detail profile setting information, please refer to the Chapter "LOW-POWER CONSUMPTION".
- When Backup RAM power restores, the completion for the profile setting needs to be confirmed through profile update status register, or start to access Backup RAM after confirmed profile update completion interrupt register.
- Backup RAM can configure SLEEP setting during transfer to PSS. To detail SLEEP control information, please refer to the Chapter "LOW-POWER CONSUMPTION".
- When Backup RAM returns from SLEEP setting, start to access Backup RAM after confirmed that PSS had transferred to RUN status through the device status register of the system status register.

CHAPTER 13: External Interrupt

This chapter explains the functions and operations of external interrupts.

1. **Overview**
2. **Configuration**
3. **Explanation of Operation**
4. **Setting Procedure Example**
5. **Registers**

EXTIRQ-TXXPT03P01R01L05-E1-XX

1. Overview

This section provides an overview of external interrupts.

External Interrupt Functions

The external interrupt functions detect signal input to the external interrupt pin and generate interrupt requests.

- External interrupt requests available at 5 levels (H, L, rising edge, falling edge, and both (rising and falling) edges)
- External interrupt supports (product specification) pin
- Noise filter bypass possible
- Non-maskable interrupt supported
- DMA supports to (product specification) channels

2. Configuration

This section explains the configuration for external interrupts.

Figure 2-1 External Interrupt Block Diagram

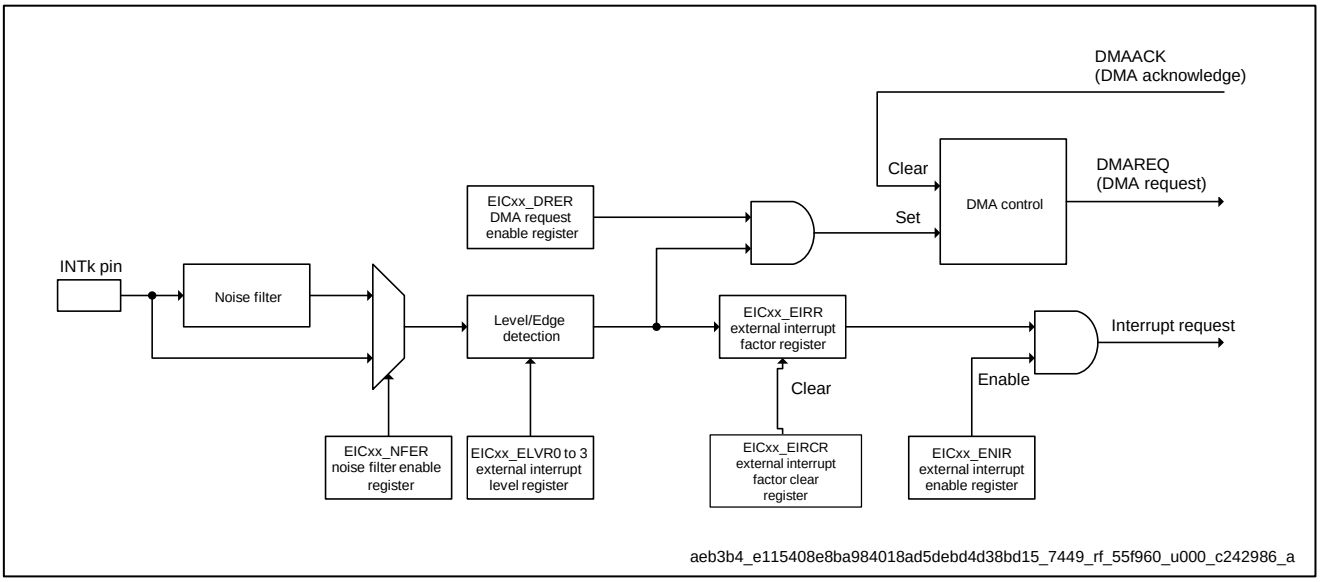
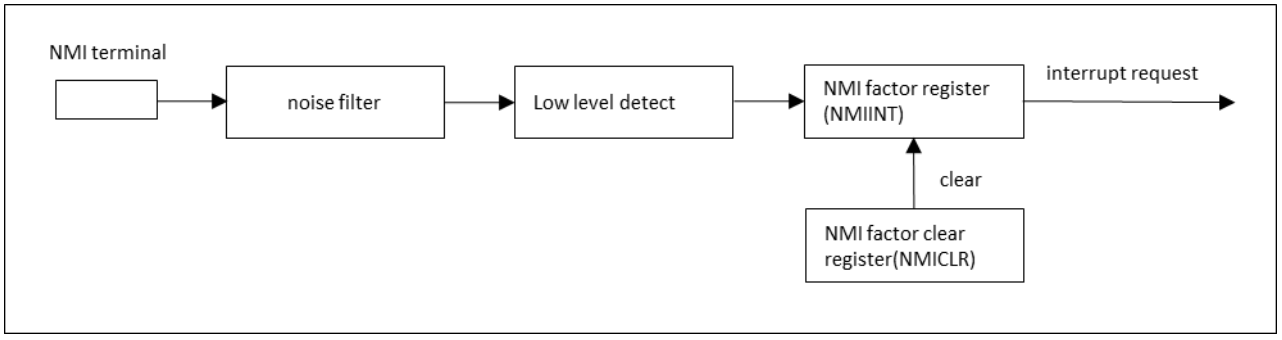


Figure 2-2 NMI (Non-Maskable Interrupt) Block Diagram



3. Explanation of Operation

This section explains the operations of external interrupts.

Clearing of Interrupt Flag

Clear the interrupt flag EICxx_EIRR:ER in the interrupt handler to use it as an external interrupt. If it is not cleared after the first interrupt handler is completed, the same handler is executed again.

Even after the interrupt flag is cleared, if level detection is specified as an event input, the flag is set again as long as the input pin keeps it at the active level. In such cases, clear the requested external factor or interrupt enable bit. Clearing by software has priority over the hardware settings.

Noise Filter

The noise filter removes noise from the signals input through the INT pin.

External Interrupt Request Level

If edge detection is specified as an event input and the noise filter is enabled, a rule applies to the pulse width of an input signal to be recognized as an input edge. For the minimum value of the pulse width, see the TRAVEO™ T1G Family Hardware Manual Platform Part data sheet.

If level detection is specified as an event input and the specified level is input, the interrupt flag retains even after a change of the input signal to the inactive level. See Figure 3-2. Clear the interrupt flag to clear the request.

Figure 3-1 Clearing of Interrupt Factor Register at Level Setting Time

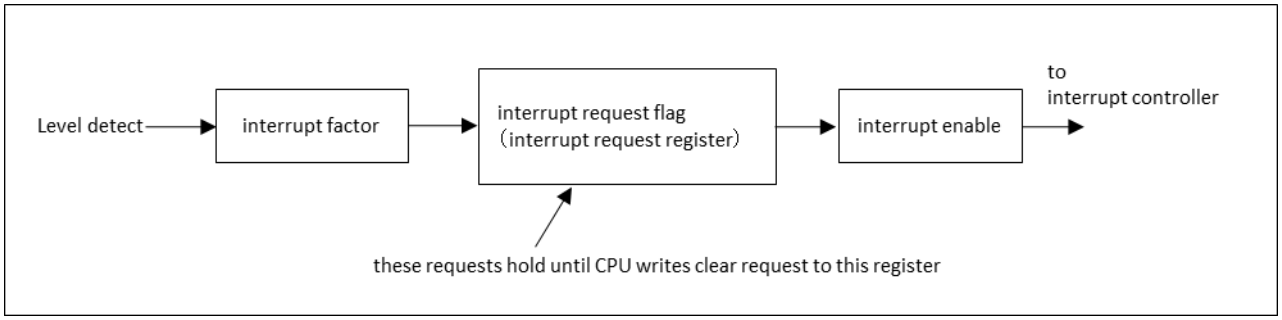


Figure 3-2 Interrupt Factor and Interrupt Request to Interrupt Controller while Interrupts Are Enabled

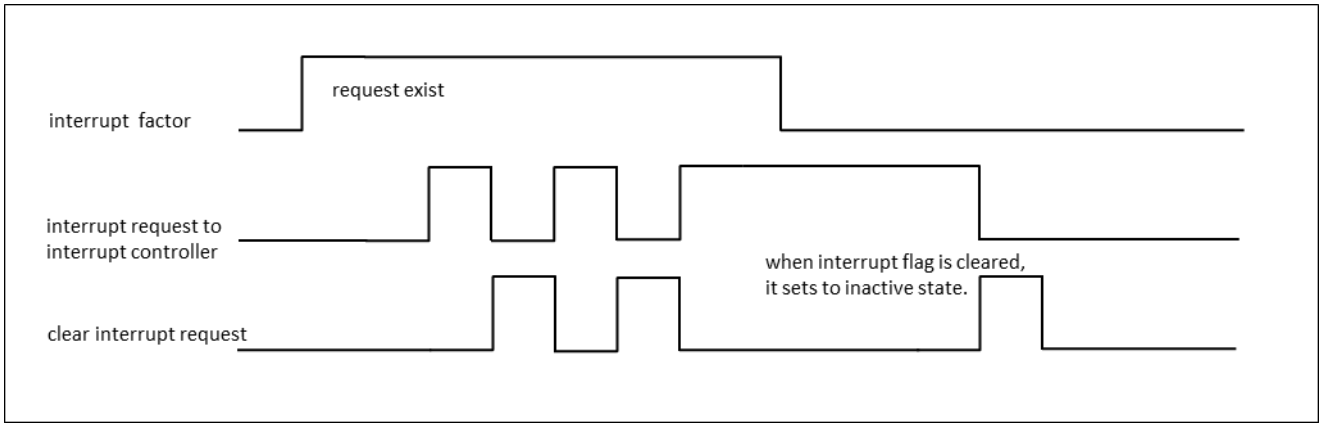
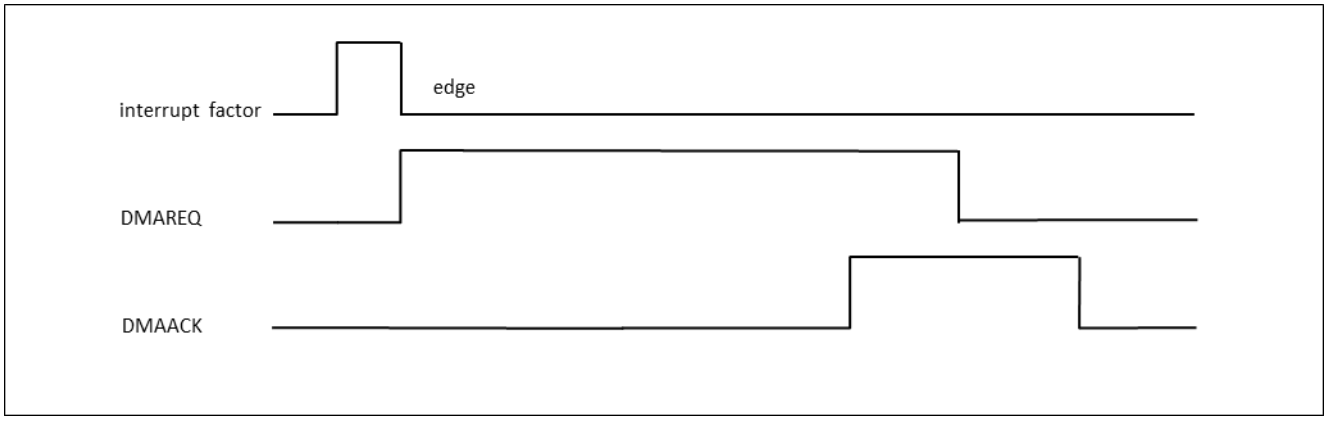


Figure 3-3 Interrupt Factor and DMAREQ to DMA while DMA Is Enabled



4. Setting Procedure Example

This section shows an example of the external interrupt setting procedure.

External Interrupt Programming Procedure

Use the following procedure to configure the existing registers in the external interrupt block.

1. Set disable in the bits that are the target of the external interrupt enable register (EICxx_ENIR).
2. Set the bits that are the target of an external interrupt level setting register (EICxx_ELVR0 to 3) or the noise filter enable register (EICxx_NFER).
3. Confirm write and read the bits that are the target of an external interrupt level setting register (EICxx_ELVR0 to 3) or the noise filter enable register (EICxx_NFER).
4. Clear the bits that are the target of the external interrupt factor register (EICxx_EIRR).
5. Set enable in the bits that are the target of the external interrupt enable register (EICxx_ENIR).

Notes:

- *The enable register must be disabled before the registers in this module are configured. Also, be sure to clear the factor register before enabling the enable register.*
- *This prevents an interrupt factor from being incorrectly generated when the registers are configured or when interrupts are enabled.*

5. Registers

This section lists the registers.

Register List

The following table lists the external interrupt and NMI controller registers.

The additional character xx in the register name indicates the unit.

EIC00 supports channels 0 to 31.

EIC01 supports channels 32 to 63.

EIC02 supports channels 64 to 95.

EIC03 supports channels 96 to 127.

Table 5-1 List of External Interrupt/NMI Controller Registers

Abbreviated Register Name	Register Name	Reference
EICxx_ENIR	External interrupt enable register	5.1
EICxx_ENISR	External interrupt enable set register	5.2
EICxx_ENICR	External interrupt enable clear register	5.3
EICxx_EIRR	External interrupt factor register	5.4
EICxx_EIRCR	External interrupt factor clear register	5.5
EICxx_NFER	Noise filter enable register	5.6
EICxx_NFESR	Noise filter enable set register	5.7
EICxx_NFECR	Noise filter enable clear register	5.8
EICxx_ELVR0 to 3	External interrupt level register	5.9
EICxx_NMIR	Non-maskable interrupt register	5.10
EICxx_DRER	DMA request enable register	5.11
EICxx_DRESR	DMA request enable set register	5.12
EICxx_DRECR	DMA request enable clear register	5.13
EICxx_DRFR	DMA request flag register	5.14

Table 5-2 External Interrupt Register Memory Map shows the register address map.

Table 5-2 External Interrupt Register Memory Map

Offset	+3	+2	+1	+0
0x00000000	EICxx_ENIR 00000000_00000000_00000000_00000000			
0x00000004	EICxx_ENISR 00000000_00000000_00000000_00000000			
0x00000008	EICxx_ENICR 00000000 00000000 00000000 00000000			
0x0000000C	EICxx_EIRR XXXXXXXX_XXXXXXXX_XXXXXXXX_XXXXXXXX			
0x00000010	EICxx_EIRCR 00000000_00000000_00000000_00000000			
0x00000014	EICxx_NFER 00000000_00000000_00000000_00000000			

Offset	+3	+2	+1	+0
0x00000018	EICxx_NFESR 00000000_00000000_00000000_00000000			
0x0000001C	EICxx_NFECR 00000000_00000000_00000000_00000000			
0x00000020	EICxx_ELVR0 00000000_00000000_00000000_00000000			
0x00000024	EICxx_ELVR1 00000000_00000000_00000000_00000000			
0x00000028	EICxx_ELVR2 00000000_00000000_00000000_00000000			
0x0000002C	EICxx_ELVR3 00000000_00000000_00000000_00000000			
0x00000030	EICxx_NMIR 00000000_00000000_00000000_00000000			
0x00000034	EICxx_DRER 00000000_00000000_00000000_00000000			
0x00000038	EICxx_DRESR 00000000_00000000_00000000_00000000			
0x0000003C	EICxx_DRECR 00000000_00000000_00000000_00000000			
0x00000040	EICxx_DRFR XXXXXXXX_XXXXXXXX_XXXXXXXX_XXXXXXXX			

Note:

- "X" indicates that the initial register value after reset is undefined.

5.1. External Interrupt Enable Register (EICxx_ENIR)

This register is used for mask control of external interrupt source output.

bit	31	30	29	28	27	26	25	24
Field	EN31	EN30	EN29	EN28	EN27	EN26	EN25	EN24
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	EN23	EN22	EN21	EN20	EN19	EN18	EN17	EN16
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	EN15	EN14	EN13	EN12	EN11	EN10	EN9	EN8
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	EN7	EN6	EN5	EN4	EN3	EN2	EN1	EN0
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	WP							
Initial Value	0	0	0	0	0	0	0	0

[bit31:0] EN31 to EN0: External Interrupt Enable Bits

These bits set enable for external interrupts of each channel.

Bit	Description
0	Disable generation of external interrupt factors.
1	Enable generation of external interrupt factors.

Note:

- *This register can be set/cleared not only by writing "1" or "0" directly to the EICxx_ENIR register but also by writing "1" to the EICxx_ENISR/EICxx_ENICR register.*

5.2. External Interrupt Enable Set Register (EICxx_ENISR)

This register is used for set control of the external interrupt enable register.

bit	31	30	29	28	27	26	25	24
Field	ENS31	ENS30	ENS29	ENS28	ENS27	ENS26	ENS25	ENS24
R/W Attribute	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W
Protection Attribute	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	ENS23	ENS22	ENS21	ENS20	ENS19	ENS18	ENS17	ENS16
R/W Attribute	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W
Protection Attribute	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	ENS15	ENS14	ENS13	ENS12	ENS11	ENS10	ENS9	ENS8
R/W Attribute	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W
Protection Attribute	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	ENS7	ENS6	ENS5	ENS4	ENS3	ENS2	ENS1	ENS0
R/W Attribute	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W
Protection Attribute	WP							
Initial Value	0	0	0	0	0	0	0	0

[bit31:0] ENS31 to ENS0: External Interrupt Enable Set Bits

These bits configure the set control of the EICxx_ENIR register.

ENS31 to ENS0	Description
During write operation	When "0" is written: Invalid When "1" is written: Set the EICxx_ENIR:ENn bit to "1".
During read operation	The read value is always "0".

5.3. External Interrupt Enable Clear Register (EICxx_ENICR)

This register is used for the clear control of the external interrupt enable register.

bit	31	30	29	28	27	26	25	24
Field	ENC31	ENC30	ENC29	ENC28	ENC27	ENC26	ENC25	ENC24
R/W Attribute	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W
Protection Attribute	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	ENC23	ENC22	ENC21	ENC20	ENC19	ENC18	ENC17	ENC16
R/W Attribute	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W
Protection Attribute	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	ENC15	ENC14	ENC13	ENC12	ENC11	ENC10	ENC9	ENC8
R/W Attribute	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W
Protection Attribute	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	ENC7	ENC6	ENC5	ENC4	ENC3	ENC2	ENC1	ENC0
R/W Attribute	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W
Protection Attribute	WP							
Initial Value	0	0	0	0	0	0	0	0

[bit31:0] ENC31 to ENC0: External Interrupt Enable Clear Bits

These bits configure the clear control of the EICxx_ENIR register.

ENC31 to ENC0	Description
During write operation	When "0" is written: Invalid When "1" is written: Clear the EICxx_ENIR:ENn bit to "0".
During read operation	The read value is always "0".

5.4. External Interrupt Factor Register (EICxx_EIRR)

This register indicates the status when a pin detects an external interrupt factor.

bit	31	30	29	28	27	26	25	24
Field	ER31	ER30	ER29	ER28	ER27	ER26	ER25	ER24
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	WP							
Initial Value	X	X	X	X	X	X	X	X

bit	23	22	21	20	19	18	17	16
Field	ER23	ER22	ER21	ER20	ER19	ER18	ER17	ER16
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	WP							
Initial Value	X	X	X	X	X	X	X	X

bit	15	14	13	12	11	10	9	8
Field	ER15	ER14	ER13	ER12	ER11	ER10	ER9	ER8
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	WP							
Initial Value	X	X	X	X	X	X	X	X

bit	7	6	5	4	3	2	1	0
Field	ER7	ER6	ER5	ER4	ER3	ER2	ER1	ER0
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	WP							
Initial Value	X	X	X	X	X	X	X	X

[bit31:0] ER31 to ER0: External Interrupt Factor Detection Bits

These bits retain the detection of an external interrupt request.

ER31 to ER0	Description
0	No external interrupt factor is detected.
1	An external interrupt factor is detected.

5.5. External Interrupt Factor Clear Register (EICxx_EIRCR)

This register is used for the clear control of the external interrupt factor register.

bit	31	30	29	28	27	26	25	24
Field	ERC31	ERC30	ERC29	ERC28	ERC27	ERC26	ERC25	ERC24
R/W Attribute	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W
Protection Attribute	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	ERC23	ERC22	ERC21	ERC20	ERC19	ERC18	ERC17	ERC16
R/W Attribute	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W
Protection Attribute	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	ERC15	ERC14	ERC13	ERC12	ERC11	ERC10	ERC9	ERC8
R/W Attribute	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W
Protection Attribute	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	ERC7	ERC6	ERC5	ERC4	ERC3	ERC2	ERC1	ERC0
R/W Attribute	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W
Protection Attribute	WP							
Initial Value	0	0	0	0	0	0	0	0

[bit31:0] ERC31 to ERC0: External Interrupt Factor Clear Bits

These bits configure the clear control of the EICxx_EIRR register.

ERC31 to ERC0	Description
During write operation	When "0" is written: Invalid When "1" is written: Clear the EICxx_EIRR:ERn bit to "0".
During read operation	The read value is always "0".

5.6. Noise Filter Enable Register (EICxx_NFER)

This register can set whether to use the noise filter for a corresponding external interrupt factor.

bit	31	30	29	28	27	26	25	24
Field	NFE31	NFE30	NFE29	NFE28	NFE27	NFE26	NFE25	NFE24
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	NFE23	NFE22	NFE21	NFE20	NFE19	NFE18	NFE17	NFE16
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	NFE15	NFE14	NFE13	NFE12	NFE11	NFE10	NFE9	NFE8
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	NFE7	NFE6	NFE5	NFE4	NFE3	NFE2	NFE1	NFE0
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	WP							
Initial Value	0	0	0	0	0	0	0	0

[bit31:0] NFE31 to NFE0: Noise Filter Enable Bits

These bits configure the noise filter control of each external interrupt.

NFE31 to NFE0	Description
0	Disable the noise filter.
1	Enable the noise filter.

Note:

- *This register can be set/cleared not only by writing "1" or "0" directly to the EICxx_NFER register but also by writing "1" to the EICxx_NFESR/EICxx_NFECR register.*

5.7. Noise Filter Enable Set Register (EICxx_NFESR)

This register is used for the set control of the noise filter enable register.

bit	31	30	29	28	27	26	25	24
Field	NFES31	NFES30	NFES29	NFES28	NFES27	NFES26	NFES25	NFES24
R/W Attribute	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W
Protection Attribute	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	NFES23	NFES22	NFES21	NFES20	NFES19	NFES18	NFES17	NFES16
R/W Attribute	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W
Protection Attribute	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	NFES15	NFES14	NFES13	NFES12	NFES11	NFES10	NFES9	NFES8
R/W Attribute	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W
Protection Attribute	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	NFES7	NFES6	NFES5	NFES4	NFES3	NFES2	NFES1	NFES0
R/W Attribute	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W
Protection Attribute	WP							
Initial Value	0	0	0	0	0	0	0	0

[bit31:0] NFES31 to NFES0: Noise Filter Enable Set Bits

These bits configure the set control of the EICxx_NFER register.

NFES31 to NFES0	Description
During write operation	When "0" is written: Invalid When "1" is written: Set the EICxx_NFER:NFEn bit to "1".
During read operation	The read value is always "0".

5.8. Noise Filter Enable Clear Register (EICxx_NFECR)

This register is used for the clear control of the noise filter enable register.

bit	31	30	29	28	27	26	25	24
Field	NFEC31	NFEC30	NFEC29	NFEC28	NFEC27	NFEC26	NFEC25	NFEC24
R/W Attribute	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W
Protection Attribute	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	NFEC23	NFEC22	NFEC21	NFEC20	NFEC19	NFEC18	NFEC17	NFEC16
R/W Attribute	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W
Protection Attribute	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	NFEC15	NFEC14	NFEC13	NFEC12	NFEC11	NFEC10	NFEC9	NFEC8
R/W Attribute	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W
Protection Attribute	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	NFEC7	NFEC6	NFEC5	NFEC4	NFEC3	NFEC2	NFEC1	NFEC0
R/W Attribute	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W
Protection Attribute	WP							
Initial Value	0	0	0	0	0	0	0	0

[bit31:0] NFEC31 to NFEC0: Noise Filter Enable Clear Bits

These bits configure the clear control of the EICxx_NFER register.

NFEC31 to NFEC0	Description
During write operation	When "0" is written: Invalid When "1" is written: Clear the EICxx_NFER:NFEn bit to "0".
During read operation	The read value is always "0".

5.9. External Interrupt Level Registers (EICxx_ELVR0 to 3)

These registers select the level and edge of the signals to be detected as external interrupt requests.

bit	31	30	29	28	27	26	25	24
Field	Reserved	LC7	LB7	LA7	Reserved	LC6	LB6	LA6
R/W Attribute	R0, WX	R/W	R/W	R/W	R0, WX	R/W	R/W	R/W
Protection Attribute	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	Reserved	LC5	LB5	LA5	Reserved	LC4	LB4	LA4
R/W Attribute	R0, WX	R/W	R/W	R/W	R0, WX	R/W	R/W	R/W
Protection Attribute	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	Reserved	LC3	LB3	LA3	Reserved	LC2	LB2	LA2
R/W Attribute	R0, WX	R/W	R/W	R/W	R0, WX	R/W	R/W	R/W
Protection Attribute	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	Reserved	LC1	LB1	LA1	Reserved	LC0	LB0	LA0
R/W Attribute	R0, WX	R/W	R/W	R/W	R0, WX	R/W	R/W	R/W
Protection Attribute	WP							
Initial Value	0	0	0	0	0	0	0	0

[bit31:0] LC7 to LC0/LB7 to LB0/LA7 to LA0: Level Selection Bits for External Interrupt Request Detection

These bits select the level and edge of the signals to be detected as external interrupt requests.

EICxx_ELVRm.LCn	EICxx_ELVRm.LBn	EICxx_ELVRm.LAn	Description
0	0	0	Input of the L level to the pin
0	0	1	Input of the H level to the pin
0	1	0	Input of a rising edge to the pin
0	1	1	Input of a falling edge to the pin
1	X	X	Both edges (rising edge and falling edge)

* m=0 to 3, n=7 to 0

Setting		Description
EICxx_ELVR3	LC/LB/LA7	INT31 interrupt level setting
EICxx_ELVR3	LC/LB/LA6	INT30 interrupt level setting
EICxx_ELVR3	LC/LB/LA5	INT29 interrupt level setting
EICxx_ELVR3	LC/LB/LA4	INT28 interrupt level setting
EICxx_ELVR3	LC/LB/LA3	INT27 interrupt level setting

Setting		Description
EICxx_ELVR3	LC/LB/LA2	INT26 interrupt level setting
EICxx_ELVR3	LC/LB/LA1	INT25 interrupt level setting
EICxx_ELVR3	LC/LB/LA0	INT24 interrupt level setting
EICxx_ELVR2	LC/LB/LA7	INT23 interrupt level setting
EICxx_ELVR2	LC/LB/LA6	INT22 interrupt level setting
EICxx_ELVR2	LC/LB/LA5	INT21 interrupt level setting
EICxx_ELVR2	LC/LB/LA4	INT20 interrupt level setting
EICxx_ELVR2	LC/LB/LA3	INT19 interrupt level setting
EICxx_ELVR2	LC/LB/LA2	INT18 interrupt level setting
EICxx_ELVR2	LC/LB/LA1	INT17 interrupt level setting
EICxx_ELVR2	LC/LB/LA0	INT16 interrupt level setting
EICxx_ELVR1	LC/LB/LA7	INT15 interrupt level setting
EICxx_ELVR1	LC/LB/LA6	INT14 interrupt level setting
EICxx_ELVR1	LC/LB/LA5	INT13 interrupt level setting
EICxx_ELVR1	LC/LB/LA4	INT12 interrupt level setting
EICxx_ELVR1	LC/LB/LA3	INT11 interrupt level setting
EICxx_ELVR1	LC/LB/LA2	INT10 interrupt level setting
EICxx_ELVR1	LC/LB/LA1	INT9 interrupt level setting
EICxx_ELVR1	LC/LB/LA0	INT8 interrupt level setting
EICxx_ELVR0	LC/LB/LA7	INT7 interrupt level setting
EICxx_ELVR0	LC/LB/LA6	INT6 interrupt level setting
EICxx_ELVR0	LC/LB/LA5	INT5 interrupt level setting
EICxx_ELVR0	LC/LB/LA4	INT4 interrupt level setting
EICxx_ELVR0	LC/LB/LA3	INT3 interrupt level setting
EICxx_ELVR0	LC/LB/LA2	INT2 interrupt level setting
EICxx_ELVR0	LC/LB/LA1	INT1 interrupt level setting
EICxx_ELVR0	LC/LB/LA0	INT0 interrupt level setting

5.10. Non-Maskable Interrupt Register (EICxx_NMIR)

This register configures the non-maskable interrupt register.

bit	31	16
Field	Reserved	
R/W Attribute	R0, WX	
Protection Attribute	WP	
Initial Value	00000000_00000000	

bit	15	14	13	12	11	10	9	8
Field	Reserved							NMICLR
R/W Attribute	R0, WX							R0, W
Protection Attribute	WP							
Initial Value	0000000							0

bit	7	6	5	4	3	2	1	0
Field	Reserved							NMIINT
R/W Attribute	R0, WX							R, WX
Protection Attribute	WP							
Initial Value	0000000							0

[bit31:9] Reserved: Reserved Bits

[bit8] NMICLR: Non-Maskable Interrupt Clear Bit

This bit configures the clear control of non-maskable interrupts.

NMICLR	Description
During write operation	When "0" is written: Invalid When "1" is written: Clear the EICxx_NMIR:NMIINT bit to "0".
During read operation	The read value is always "0".

[bit7:1] Reserved: Reserved Bits

[bit0] NMIINT: Non-Maskable Interrupt Request Detection Bit

This bit retains the detection of a non-maskable interrupt request.

Bit	Description
0	No non-maskable interrupt request is detected.
1	A non-maskable interrupt request is detected.

5.11. DMA Request Enable Register (EICxx_DRER)

This register enables DMA for external interrupt requests.

bit	31	30	29	28	27	26	25	24
Field	DRE31	DRE30	DRE29	DRE28	DRE27	DRE26	DRE25	DRE24
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	DRE23	DRE22	DRE21	DRE20	DRE19	DRE18	DRE17	DRE16
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	DRE15	DRE14	DRE13	DRE12	DRE11	DRE10	DRE9	DRE8
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	DRE7	DRE6	DRE5	DRE4	DRE3	DRE2	DRE1	DRE0
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	WP							
Initial Value	0	0	0	0	0	0	0	0

[bit31:0] DRE31 to DRE0: DMA Request Enable Bits

These bits set enable for DMA requests for external interrupts.

Bit	Description
0	Disable DMA.
1	Enable DMA.

Note:

- *This register can be set/cleared not only by writing "1" or "0" directly to the EICxx_DRER register but also by writing "1" to the EICxx_DRESR/EICxx_DRECR register.*

5.12. DMA Request Enable Set Register (EICxx_DRESR)

This register is used for the set control of the DMA request enable register.

bit	31	30	29	28	27	26	25	24
Field	DRES31	DRES30	DRES29	DRES28	DRES27	DRES26	DRES25	DRES24
R/W Attribute	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W
Protection Attribute	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	DRES23	DRES22	DRES21	DRES20	DRES19	DRES18	DRES17	DRES16
R/W Attribute	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W
Protection Attribute	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	DRES15	DRES14	DRES13	DRES12	DRES11	DRES10	DRES9	DRES8
R/W Attribute	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W
Protection Attribute	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	DRES7	DRES6	DRES5	DRES4	DRES3	DRES2	DRES1	DRES0
R/W Attribute	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W
Protection Attribute	WP							
Initial Value	0	0	0	0	0	0	0	0

[bit31:0] DRES31 to DRES0: DMA Request Enable Set Bits

These bits configure the set control of the EICxx_DRER register.

DRES31 to DRES0	Description
During write operation	When "0" is written: Invalid When "1" is written: Set the EICxx_DRER:DREn bit to "1".
During read operation	The read value is always "0".

5.13. DMA Request Enable Clear Register (EICxx_DRECR)

This register is used for the clear control of the DMA request enable register.

bit	31	30	29	28	27	26	25	24
Field	DREC31	DREC30	DREC29	DREC28	DREC27	DREC26	DREC25	DREC24
R/W Attribute	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W
Protection Attribute	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	DREC23	DREC22	DREC21	DREC20	DREC19	DREC18	DREC17	DREC16
R/W Attribute	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W
Protection Attribute	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	DREC15	DREC14	DREC13	DREC12	DREC11	DREC10	DREC9	DREC8
R/W Attribute	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W
Protection Attribute	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	DREC7	DREC6	DREC5	DREC4	DREC3	DREC2	DREC1	DREC0
R/W Attribute	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W	R0, W
Protection Attribute	WP							
Initial Value	0	0	0	0	0	0	0	0

[bit31:0] DREC31 to DREC0: DMA Request Enable Clear Bits

These bits configure the clear control of the EICxx_DRER register.

DREC31 to DREC0	Description
During write operation	When "0" is written: Invalid When "1" is written: Clear the EICxx_DRER:DREn to "0".
During read operation	The read value is always "0".

5.14. DMA Request Flag Register (EICxx_DRFR)

This register indicates the status about when a DMA request is detected.

bit	31	30	29	28	27	26	25	24
Field	DRF31	DRF30	DRF29	DRF28	DRF27	DRF26	DRF25	DRF24
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	WP							
Initial Value	X	X	X	X	X	X	X	X

bit	23	22	21	20	19	18	17	16
Field	DRF23	DRF22	DRF21	DRF20	DRF19	DRF18	DRF17	DRF16
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	WP							
Initial Value	X	X	X	X	X	X	X	X

bit	15	14	13	12	11	10	9	8
Field	DRF15	DRF14	DRF13	DRF12	DRF11	DRF10	DRF9	DRF8
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	WP							
Initial Value	X	X	X	X	X	X	X	X

bit	7	6	5	4	3	2	1	0
Field	DRF7	DRF6	DRF5	DRF4	DRF3	DRF2	DRF1	DRF0
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	WP							
Initial Value	X	X	X	X	X	X	X	X

[bit31:0] DRF31 to DRF0: DMA Request Detection Bits

These bits retain the detection of a DMA request.

DRF31 to DRF0	Description
0	No DMA request is detected.
1	A DMA request is detected.

Notes:

- *Depending on the response from DMA, the EICxx_DRFR register bits are cleared.*
- *An EICxx_DRFR register bit is set when EICxx_DRER is enabled and a pin detects an interrupt event.*

CHAPTER 14: Hardware Watchdog Timer

This section describes the Hardware watchdog timer.

1. **Overview**
2. **Configuration and Block Diagram**
3. **Explanation of Operation**
4. **Setting Procedure Example**
5. **Operation Examples**
6. **Register List**
7. **Precautions for Using This Device**

HWDT-TXXPT03P01R01L08-E1-XX

1. Overview

This section provides an overview of the Hardware watchdog timer.

The Hardware watchdog timer is positioned in the MCU configuration group, used for detecting a runaway state caused by a user program, also the Hardware watchdog timer is used to monitor control of the entire system performed by the CPU.

Hardware Watchdog Timer Features

- Detecting a runaway state caused by a user program generates a reset request. (*a)
- Clearing the hard reset starts the hardware watchdog timer. (*a)
- The watchdog counter source clock is selected from 2 types. (*a)
  (High-speed CR clock (default) or low-speed CR clock)
- The BootROM program is used to set registers. (*a)
- It operates as a 32-bit watchdog counter.
- Different initializations between soft reset and hard reset can be applied.
- Soft reset clears only the watchdog counter.
- Hard reset initializes the watchdog counter, register setting, and internal circuits.
- It monitors the watchdog counter clear protection trigger sequence.
- The window watchdog function is implemented.
- Different operation settings are possible for each device state (RUN and PSS).
- It monitors the watchdog register write protection sequence.
- The majority circuits for bits that affect important functions are implemented.
- It can generate the watchdog reset request or watchdog interrupt request (NMI) in response to a watchdog error.
- It can generate a prior warning interrupt request before the reset request or interrupt request (NMI).
- It provides the function for outputting the watchdog counter monitor bit for the MCU output pin so that the watchdog timer counter can be observed from the outside. (Support target of this function is different depends on product specification. For details, see "Output of the Watchdog Counter Monitor Bit for MCU Output Pin" in 3 Explanation of Operation.)
- The watchdog counter can be stopped in the debugging state of the processor.

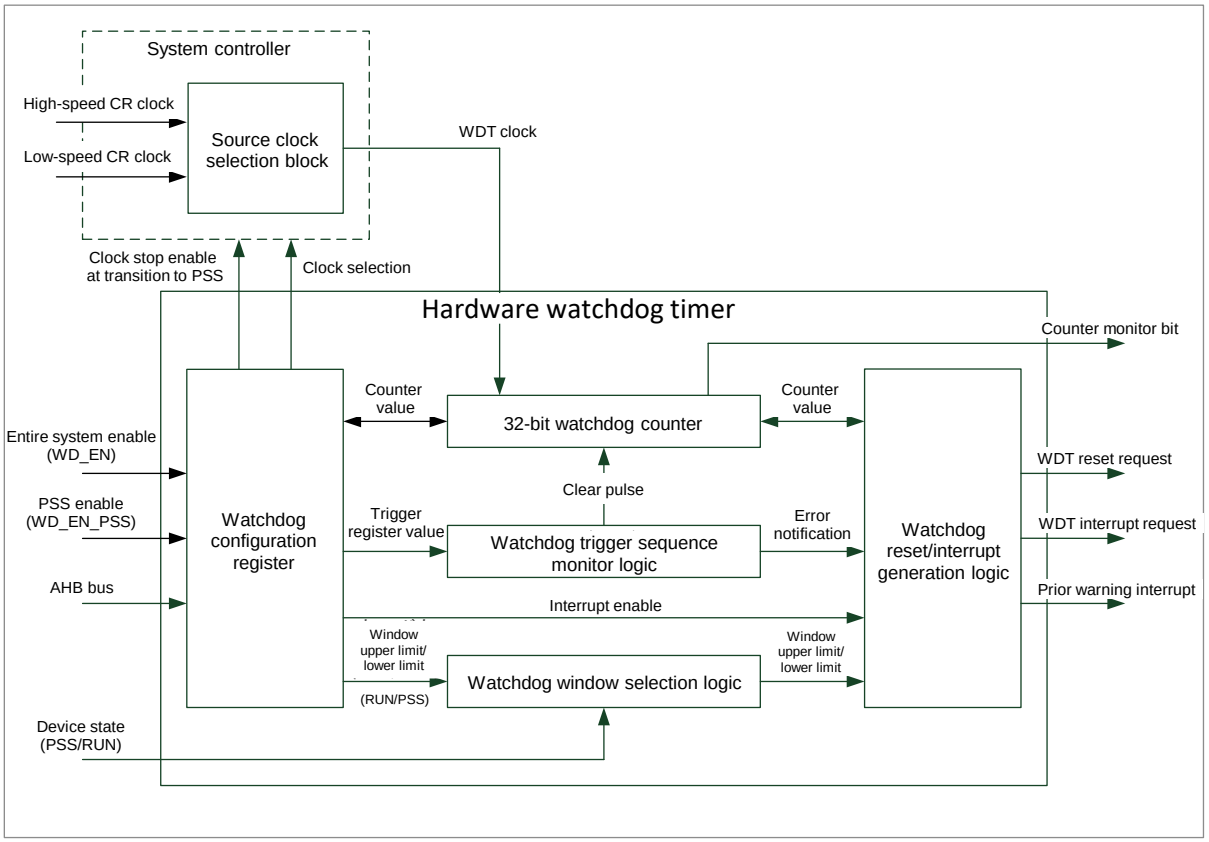
Notes:

- The notation (*a) indicates a feature that is specific to the hardware watchdog timer compared with the software watchdog.
- A watchdog error is a factor that causes the watchdog reset request or watchdog interrupt request (NMI).
- The use of the watchdog interrupt request (NMI) is prohibited except when used as a test function. (This is prohibited because if such use is allowed, ***security*** will be compromised during the execution of an application.)
- Figure 3-6 shows the operation of the test function of the Hardware watchdog.

2. Configuration and Block Diagram

This section describes the block diagrams of the Hardware watchdog timer.

Figure 2-1 Hardware Watchdog Timer Block Diagram



- Watchdog configuration register:

  In this block, the set value of each register is stored. I/O operations for information including the counter value for read operation, trigger register value necessary for clearing the counter, and window upper limit/lower limit values are performed.

- 32-bit watchdog counter:

  This is the block of the 32-bit up counter.

  It provides the function for outputting the watchdog counter monitor bit for the MCU output pin.

- Watchdog window selection logic:

  This block fetches the window upper limit value and lower limit value for each device state (RUN and PSS) and selects a window setting according to the device state transition.

- Watchdog reset/interrupt generation logic:

  This block generates the watchdog reset request, watchdog interrupt request (NMI), and prior warning interrupt request.

3. Explanation of Operation

This section describes operation of the Hardware watchdog timer.

Hardware Watchdog Timer Functions

The Hardware watchdog timer implements the following functions.

Reset Request Generation when Hardware Failure Causes Runaway State

The hardware watchdog timer performs monitoring to assure that there is no runaway state caused by a user program. The timer starts monitoring when it is activated by clearing a hard reset. If the watchdog counter is not cleared in the range between the upper limit value and the lower limit value of the window specified beforehand, this state is judged to be a runaway state caused by a user program. In this case, to the entire MCU, a reset request or an interrupt request (NMI) is generated.

Note:

- *Detectable runaway states caused by hardware failures include, for example, one in which the BootROM program goes out of control as a result of a memory bit error after a hard reset is cleared.*

Hardware Watchdog Timer Start by Clearing Hard Reset

Clearing a hard reset starts the hardware watchdog timer. Specifically, operation starts by using the clock generated by the high-speed CR oscillation circuit immediately after the hardware watchdog is canceled.

Hardware Watchdog Timer Control by CPU

The hardware watchdog timer monitors control by CPU over the entire system. The number of hardware watchdog timers to be mounted in the MCU is 1.

Watchdog Counter Source Clock Selection (2 Types)

The hardware watchdog timer sets values in the CLKSEL bits in the hardware watchdog configuration register (HWDG_CFG) to select a source clock for the watchdog counter from among 2 types of source clocks. Specifically, it selects a source clock from the high-speed CR clock or low-speed CR clock.

(The initial setting is for the high-speed CR clock.)

Selection of a source clock is a clock request to the system controller. Source clock switching is performed on the clock system side of the system controller. For details on the procedure, see "7. Precautions for Using This Device Switching of watchdog counter source clock."

Register Setting by BootROM Program

Hardware watchdog timer registers can be set by using the BootROM program.

32-Bit Watchdog Counter

The Hardware watchdog timer operates by using the 32-bit watchdog counter (up counter) (the initial value is 0x0000_0000).

The watchdog counter operates when all the following conditions are met.

1. The device is operating in user mode.
2. The watchdog counter has not reached the window upper limit value.
3. The device state is RUN and the WDENRUN bit is "1".
   (See "6.15 Hardware Watchdog Configuration Register (HWDG_CFG).")
4. The device state is PSS and the WDENPSS bit is "1".
   (See "6.15 Hardware Watchdog Configuration Register (HWDG_CFG).")

Note:

- *When the watchdog counter reaches the window upper limit value, the count operation stops.*

The following Table 3-1 shows the relationship between the window upper limit value and count time.

Table 3-1 Relationship between Hardware Window Upper Limit Value of Watchdog Timer and Count Time

Input Clock Frequency	Window Upper Limit Value	Count Time	Remarks
8MHz	0x0100_0000	About 2.1 s	It operates at the initial value of the Hardware watchdog upper RUN setting register (HWDG_RUNULS).
12MHz	0x0100_0000	About 1.4 s	
8MHz	0x8000_0000	About 268 s	It operates at the initial value of the Hardware watchdog upper PSS setting register (HWDG_PSSULS).
12MHz	0x8000_0000	About 178 s	

Note:

- *The above table is an example involving high-speed CR clock frequencies of 8 MHz and 12 MHz.*

Different Initializations between Soft Reset and Hard Reset

For the Hardware watchdog timer, different initializations between soft reset and hard reset can be applied. The following Table 3-2 Ranges of Initialization for Soft Reset and Hard Reset shows the ranges of initialization for soft reset and hard reset.

Table 3-2 Ranges of Initialization for Soft Reset and Hard Reset

Conditions	Reset Name	Range of Initialization
1	Soft reset	Watchdog counter
2	Hard reset	Watchdog counter and settings for all registers

Watchdog Counter Clear Protection Trigger Sequence

The Hardware watchdog timer monitors the watchdog counter clear protection trigger sequence to clear the watchdog counter.

The watchdog counter clear protection trigger sequence must meet the following conditions.

- Write data to the Hardware watchdog trigger 1 register (HWDG_TRG1) after writing data to the Hardware watchdog trigger 0 register (HWDG_TRG0).
- Do not execute the watchdog counter clear protection trigger sequence when the LOCK bit in the Hardware watchdog configuration register (HWDG_CFG) is "0".
- The value to be written to the Hardware watchdog trigger 0 register (HWDG_TRG0) must match the Hardware watchdog trigger 0 configuration register (HWDG_TRG0CFG).
- The value to be written to the Hardware watchdog trigger 1 register (HWDG_TRG1) must match the Hardware watchdog trigger 1 configuration register (HWDG_TRG1CFG).
- The watchdog counter must already have reached the window lower limit value when the watchdog counter clear protection trigger sequence is completed.

Note:

- *For details on the watchdog counter clear protection trigger sequence, see Figure 3-2.*

Window Watchdog Function

The window watchdog function is a function for setting a range in which the watchdog counter value can be cleared with upper limit and lower limit values. The following shows the range in which the watchdog counter can be cleared for 2 set values.

(Window lower limit value) ≤ (watchdog counter) < (window upper limit value)

For example, suppose that a runaway state occurs as a result of a user program and the watchdog counter clear protection trigger sequence is executed continuously. In this case, you can set a value other than 0x00000000 as a window lower limit value to detect the continuous watchdog counter clear as an abnormal operation.

The window setting for RUN is defined with the following 2 registers.

Hardware watchdog lower limit RUN current register (HWDG_RUNLLC)

Hardware watchdog upper limit RUN current register (HWDG_RUNULC)

The window setting for PSS is defined with the following 2 registers.

Hardware watchdog lower limit PSS current register (HWDG_PSSLLC)

Hardware watchdog upper limit PSS current register (HWDG_PSSULC)

Different Operation Settings for Each Device State (RUN and PSS)

The Hardware watchdog timer can make different operation settings for each device state (RUN and PSS). The operation setting is switched to the RUN or PSS operation setting according to the device state transition.

RUN operation setting

- When the LOCK bit and WDENRUN bit in the Hardware watchdog configuration register (HWDG_CFG) are set to "1", operation is performed by using the window setting for RUN.

- When the watchdog counter is not cleared in the range between the window upper limit value and lower limit value, or there is a violation in the watchdog counter clear protection trigger sequence, a watchdog error is detected. The watchdog reset request or watchdog interrupt request (NMI) is generated for this watchdog error. (For details on watchdog errors, see "7 Precautions for Using This Device Watchdog Error.")

PSS operation setting

- The window setting is switched to the window setting for PSS when the device state transitions from RUN to PSS. When the WDENPSS bit in the Hardware watchdog configuration register (HWDG_CFG) is set to "1", operation is performed by using the window setting for PSS.

- When the device state transitions from PSS to RUN, the enable bit is switched from the WDENPSS bit in the Hardware watchdog configuration register (HWDG_CFG) to the WDENRUN bit in the same register. However, the window setting is not switched immediately. Use of the window setting for PSS continues until the watchdog counter clear protection trigger sequence is executed.

Refer to Figure 3-4 and Figure 3-5 for the details on the watchdog operation in RUN and PSS, respectively.

Notes:

- *Certainly set the same value in upper limit registers and lower limit registers.*
- *Set the same value following two lower limit registers.*
  - Hardware watchdog lower limit PSS setting register (HWDG_PSSLLS)
  - Hardware watchdog lower limit RUN setting register (HWDG_RUNLLS)
- *Set the same value following two upper limit registers*
  - Hardware watchdog upper limit PSS setting register (HWDG_PSSULS)
  - Hardware watchdog upper limit RUN setting register (HWDG_RUNULS)
- *The limitation is needed, when device state change from RUN to PSS. Because the watchdog timer continues to use the window setting for PSS until the watchdog counter clear protection trigger sequence is performed. When WDENPSS bit is "0", the watchdog timer perform the same operation. So it needs same limitation.*

Figure 3-4 is shown about the hardware watchdog operation in RUN. Figure 3-5 is shown about the hardware watchdog operation in PSS.

Please refer to Figure 3-5.

Watchdog Register Write Protection Sequence

The Hardware watchdog timer monitors the watchdog register write protection sequence to write a setting value in the register.

The watchdog register write protection sequence must meet the following conditions.

- It writes a setting value in the register after writing data in the Hardware watchdog protection register (HWDG_PROT).
- It writes data in the Hardware watchdog protection register (HWDG_PROT) in privileged mode.
- The value written in the Hardware watchdog protection register (HWDG_PROT) must match the lock release key for register write protection 0xEDAC_CE55.
- It does not write a setting value in any register in another module after writing data in the Hardware watchdog protection register (HWDG_PROT).
- It does not write data in the Hardware watchdog protection register (HWDG_PROT) twice in a row.
- It writes a value in the register when the LOCK bit in the Hardware watchdog configuration register (HWDG_CFG) is "0" and the mode is privileged mode.

Note:

- *For details on the watchdog register write protection sequence, see Figure 3-1 Setting Procedure for Watchdog Register Write Protection Sequence.*

Table 3-3 Key Value for Register Programming

Key Value	Target Register	Purpose
0xEDAC_CE55	HWDG_PROT:KEY	Releasing protection lock for register write

Notes:

- *The watchdog register write protection sequence is unnecessary for the Hardware watchdog trigger 0/1 register (HWDG_TRG0 and HWDG_TRG1).*
- *Setting the LOCK bit in the Hardware watchdog configuration register (HWDG_CFG) to "1" prevents the following registers from being rewritten.*
  - Hardware watchdog interrupt configuration register (HWDG_INT)
  - Hardware watchdog trigger 0 configuration register (HWDG_TRG0CFG)
  - Hardware watchdog trigger 1 configuration register (HWDG_TRG1CFG)
  - Hardware watchdog lower limit RUN setting register (HWDG_RUNLLS)
  - Hardware watchdog upper limit RUN setting register (HWDG_RUNULS)
  - Hardware watchdog lower limit PSS setting register (HWDG_PSSLLS)
  - Hardware watchdog upper limit PSS setting register (HWDG_PSSULS)
  - Hardware watchdog reset delay counter register (HWDG_RSTDLY)
  - Hardware watchdog configuration register (HWDG_CFG)

Majority Circuits for Bits That Affect Important Functions

The Hardware watchdog timer has majority circuits for bits that affect important functions. The majority circuit consists of 3 FFs. When the bit in an FF bit is inverted due to noise or because of another factor, the correct value can be selected by majority decision among the 3 FFs.

The bits that are equipped with majority circuits are as follows.

- RSTEN bit (*1)
- LOCK bit (*2)

Notes:

- *The notation (*1) indicates a bit in the Hardware watchdog interrupt configuration register (HWDG_INT).*
- *The notation (*2) indicates a bit in the Hardware watchdog configuration register (HWDG_CFG).*

Generation of Watchdog Reset Request or Watchdog Interrupt Request (NMI)

The Hardware watchdog timer generates the watchdog reset request or watchdog interrupt request (NMI) when detecting a watchdog error. Use the watchdog interrupt request (NMI) as a test function. (For example, you can use the watchdog interrupt request (NMI) to set a break point at the beginning of an interrupt handler.)

Also, the Hardware watchdog timer can generate the prior warning interrupt request before the watchdog reset request (or watchdog interrupt request (NMI)). (For example, you can use the prior warning interrupt request so that the processor can save important data to the memory before the watchdog reset request is generated.)

The Hardware watchdog timer generates these requests by using the following procedure.

- When a watchdog error occurs, the IRQFLAG bit in the Hardware watchdog interrupt configuration register (HWDG_INT) is set. At this time, if the IRQEN bit in the Hardware watchdog interrupt configuration register (HWDG_INT) is "1", the prior warning interrupt request is generated.
- As many cycles as are set in the Hardware watchdog reset delay counter register (HWDG_RSTDLY) are inserted between the prior warning interrupt request and the watchdog reset interrupt request or watchdog interrupt request (NMI) as a delay time.
- Once the prior warning interrupt request is generated, the watchdog counter cannot be cleared with the watchdog counter clear protection trigger sequence. Also, if a new watchdog error occurs, it is ignored.
- After the elapse of the time corresponding to the number of cycles for delay time set in the Hardware watchdog reset delay counter register (HWDG_RSTDLY), the watchdog reset request or watchdog interrupt request (NMI) is generated. If the Hardware watchdog reset delay counter register (HWDG_RSTDLY) is 0x0000, no delay time occurs.

Notes:

- *When you generate the prior warning interrupt request, set the Hardware watchdog reset delay counter register (HWDG_RSTDLY) to a value other than 0x0000.*

- *The use of the watchdog interrupt request (NMI) is prohibited except when used as a test function. (This is prohibited because if such use is allowed, safety will be compromised during the execution of an application.) Figure 3-6 shows the operation of the test function of the Hardware watchdog.*
- *Insertion of a number of cycles equal to that for the delay time set in the Hardware watchdog reset delay counter register (HWDG_RSTDLY) is valid only for 1 watchdog reset request or watchdog interrupt request (NMI). (It remains invalid until the next hard reset occurs.)*

Output of the Watchdog Counter Monitor Bit for MCU Output Pin

The Hardware watchdog timer outputs the watchdog counter monitor bit to the outside so that watchdog counter operation can be monitored from the outside. You can select any 1 bit in the watchdog counter (32 bits) by setting the OBSSEL[4:0] bits in the Hardware watchdog configuration register (HWDG_CFG).

Support of this function is different depends on product specification.

Watchdog Counter Stop in Debugging State of Processor

The watchdog counter stops when the processor enters the debugging state. When the processor returns from the debugging state, the watchdog counter resumes operation from the point where it had stopped.

Note:

- *For the definition of the debugging state of the processor, see the material from ARM(R) (CortexTM-R5 Revision:r1p2 Technical Reference Manual (ARM DDI 0460D)).*

Differences between Software Watchdog and Hardware Watchdog

The following Table 3-4 Differences between Software Watchdog and Hardware Watchdog shows the main differences between the software watchdog and hardware watchdog.

Table 3-4 Differences between Software Watchdog and Hardware Watchdog

Item	Software Watchdog	Hardware Watchdog
Watchdog Counter	32-bit up counter	Same as on left
Different Initializations between Soft Reset and Hard Reset	Supported	Same as on left
Trigger Sequence for Watchdog Counter Clear	Supported	Same as on left
Window Watchdog Function	Supported	Same as on left
Different Operation Settings for Each Device State (RUN and PSS)	Supported	Same as on left
Watchdog Counter Source Clock Selection	4 types - High-speed CR clock - Low-speed CR clock - Sub clock - Main clock	2 types - High-speed CR clock - Low-speed CR clock
Watchdog Register Write Protection Sequence	Supported (However, prevention of rewriting of the window setting by the LOCK bit is not supported.)	Same as on left (However, prevention of rewriting of the window setting by the LOCK bit is supported.)
Majority Circuits for Bits That Affect Important Functions	Supported (4 target bits)	Same as on left (2 target bits)
Reset Request or Interrupt Request (NMI) Generation	Supported	Same as on left
Prior Warning Interrupt Request Generation	Supported	Same as on left
Output of Watchdog Counter Monitor Bit for MCU Output Pin	Supported	Supported
Watchdog Counter Stop in the Debugging State of the Processor	Supported	Same as on left
Watchdog Error Conditions	5 types	Same as on left
Bus Error Response Conditions	8 types	Same as on left
Watchdog Timer Start	Register write by a user program	Hard reset clear
Register Setting Value Write	User program	BootROM program
Watchdog Counter Enable Bit Control	Register write by a user program	Constantly operation (Fixed hardware)

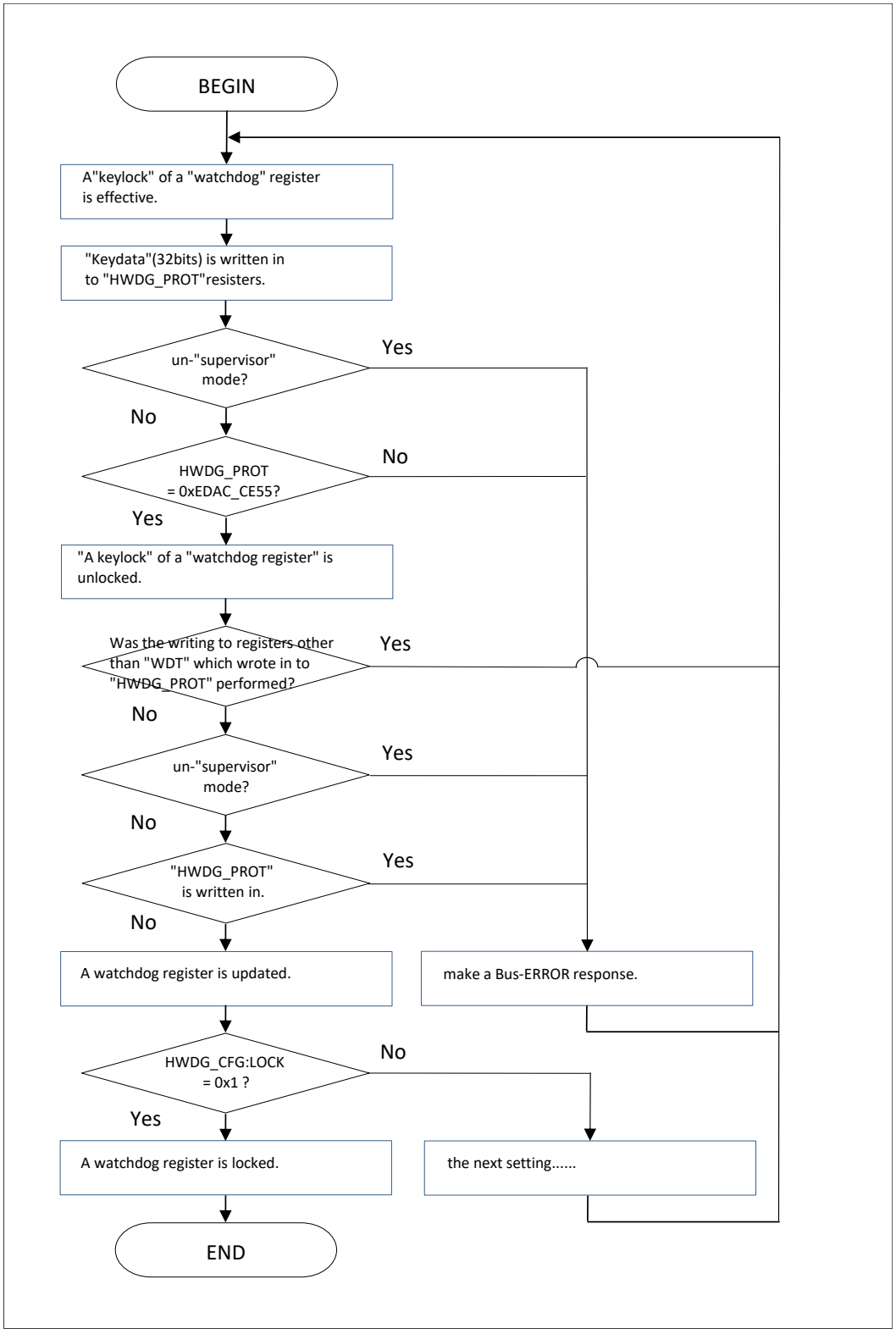
Notes:

- *Unlike the case of the software watchdog timer, for the hardware watchdog timer, the BootROM program sets register during the initial operation setting. (Write and read operation from the CPU can also be performed.)*
- *The following register setting values are determined by a marker in the flash memory (BootROM marker).*
  - HWDG_INT
  - HWDG_TRG0CFG
  - HWDG_TRG1CFG
  - HWDG_RUNLLS
  - HWDG_RUNULS
  - HWDG_PSSLLS
  - HWDG_PSSULS
  - HWDG_RSTDLY
  - HWDG_CFG

For details on the registers, see "6.Register List"

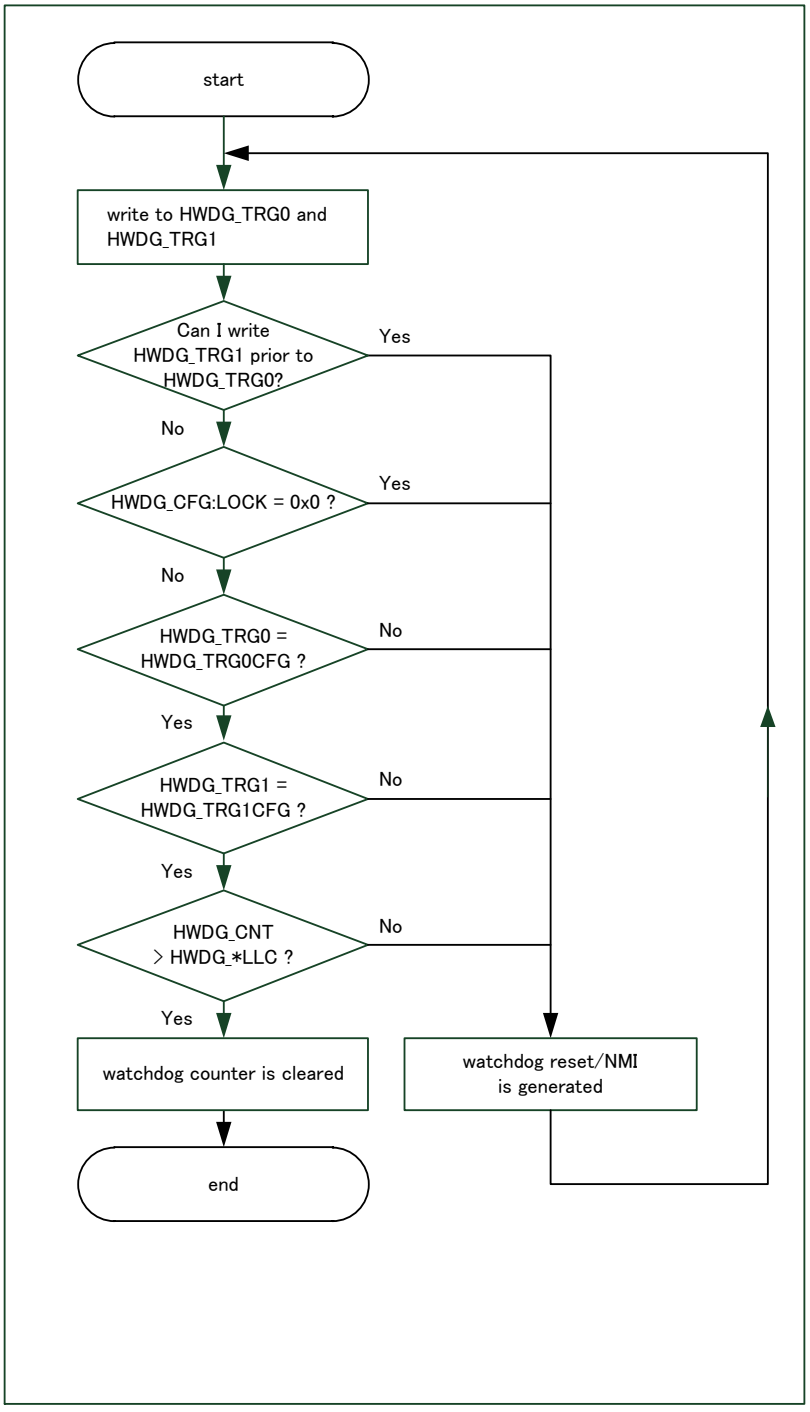
The following figure shows the watchdog register write protection sequence setting procedure.

Figure 3-1 Setting Procedure for Watchdog Register Write Protection Sequence



The following figure shows the watchdog counter clear protection trigger sequence setting procedure.

Figure 3-2 Setting Procedure for Watchdog Counter Clear Protection Trigger Sequence

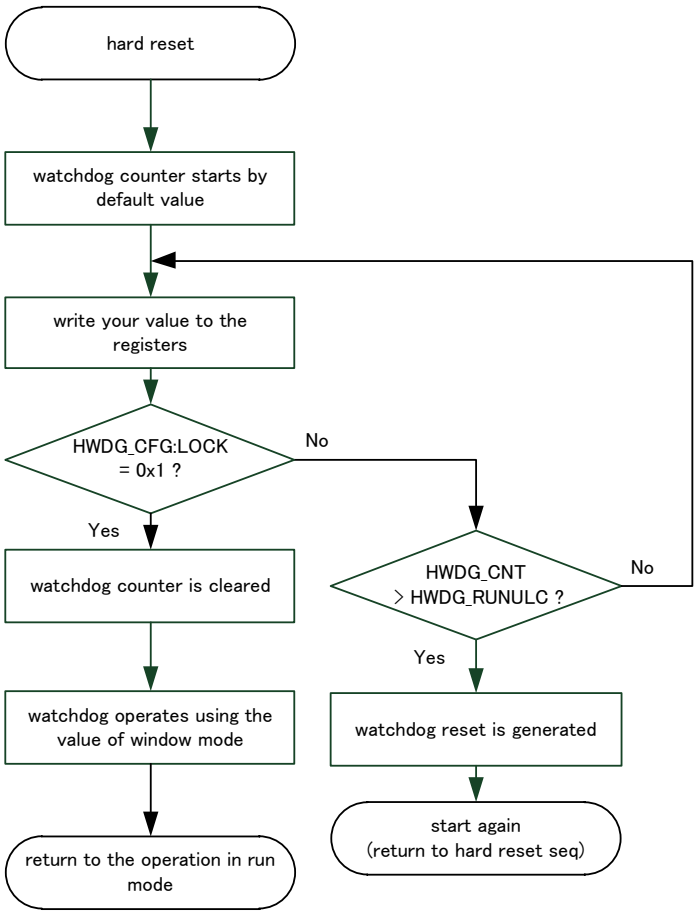


Note:

- *The notation * in Figure 3-2 Setting Procedure for Watchdog Counter Clear Protection Trigger Sequence is replaced with device state RUN or PSS.*

The following figure shows the Hardware watchdog timer startup procedure.

Figure 3-3 Startup of Hardware Watchdog Timer



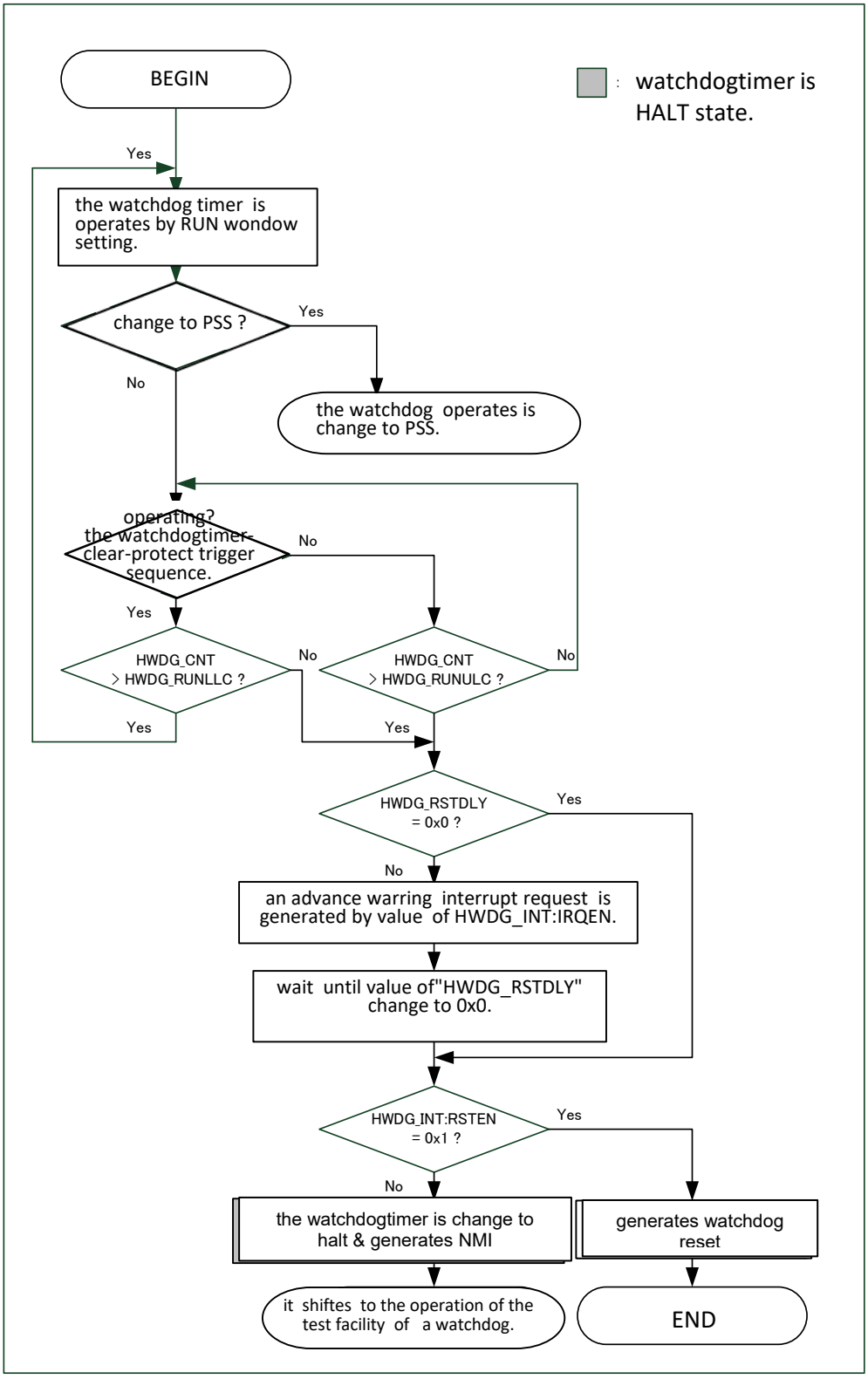
1. The hard reset initializes the hardware watchdog timer. Then, immediately after the hard reset is cleared, the operating state is indicated.
2. The default window setting indicates the window lower limit value (= 0x0000_0000) and window upper limit value (= 0x0100_0000).
3. You can change the register setting as long as the LOCK bit in the hardware watchdog configuration register (HWDG_CFG) is "0". However, complete the setting change within the time it takes to reach the window upper limit value.
4. Setting the LOCK bit in the hardware watchdog configuration register (HWDG_CFG) to "1" clears the watchdog counter.

Note:

- *The hardware watchdog timer operates by using the default window setting regardless of the set values of the registers and the device state (RUN or PSS) as long as the LOCK bit in the hardware watchdog configuration register (HWDG_CFG) is "0".*

The following figure shows the Hardware watchdog operation in RUN.

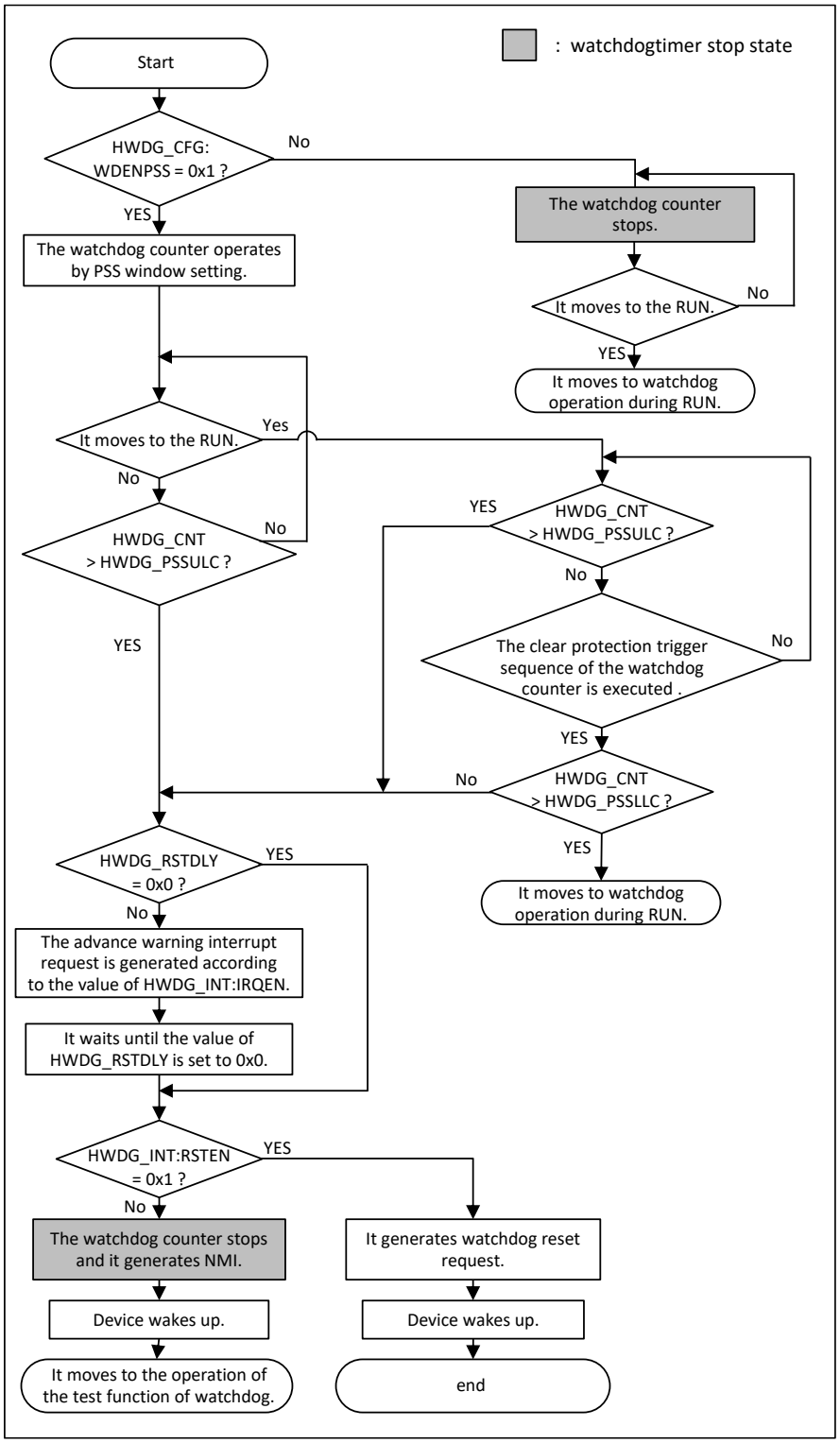
Figure 3-4 Hardware Watchdog Operation in RUN



1. The watchdog counter in RUN always operates.
2. When the device state transitions from RUN to PSS, the processing transitions to the operation of the watchdog in PSS. For details, see Figure 3-5.
3. During the operation of the watchdog counter, execute the watchdog counter clear protection trigger sequence within the range between the window upper limit value and lower limit value that have been specified beforehand. Do this to clear the watchdog counter periodically.
4. If a runaway state caused by a user program occurs and prevents periodic clearing, the watchdog counter reaches the window upper limit value. Then, processing transitions to the flow that generates the watchdog reset request or watchdog interrupt request (NMI).
5. According to the value of the IRQEN bit in the Hardware watchdog interrupt configuration register (HWDG_INT), the prior warning interrupt request is generated. At the same time, as many cycles as are set for the delay time in the Hardware watchdog reset delay counter register (HWDG_RSTDLY) are inserted.
6. After the elapse of the time corresponding to the number of cycles for the delay time, the watchdog reset request or watchdog interrupt request (NMI) is generated according to the value of the RSTEN bit in the Hardware watchdog interrupt configuration register (HWDG_INT). When the watchdog interrupt request (NMI) is generated, the processing transitions to the operation of the test function of the watchdog. For details, see Figure 3-6.

The following figure shows the Hardware watchdog operation in PSS.

Figure 3-5 Hardware Watchdog Operation in PSS



1. The watchdog counter in PSS operates or stops according to the value of the WDENPSS bit in the Hardware watchdog configuration register (HWDG_CFG).

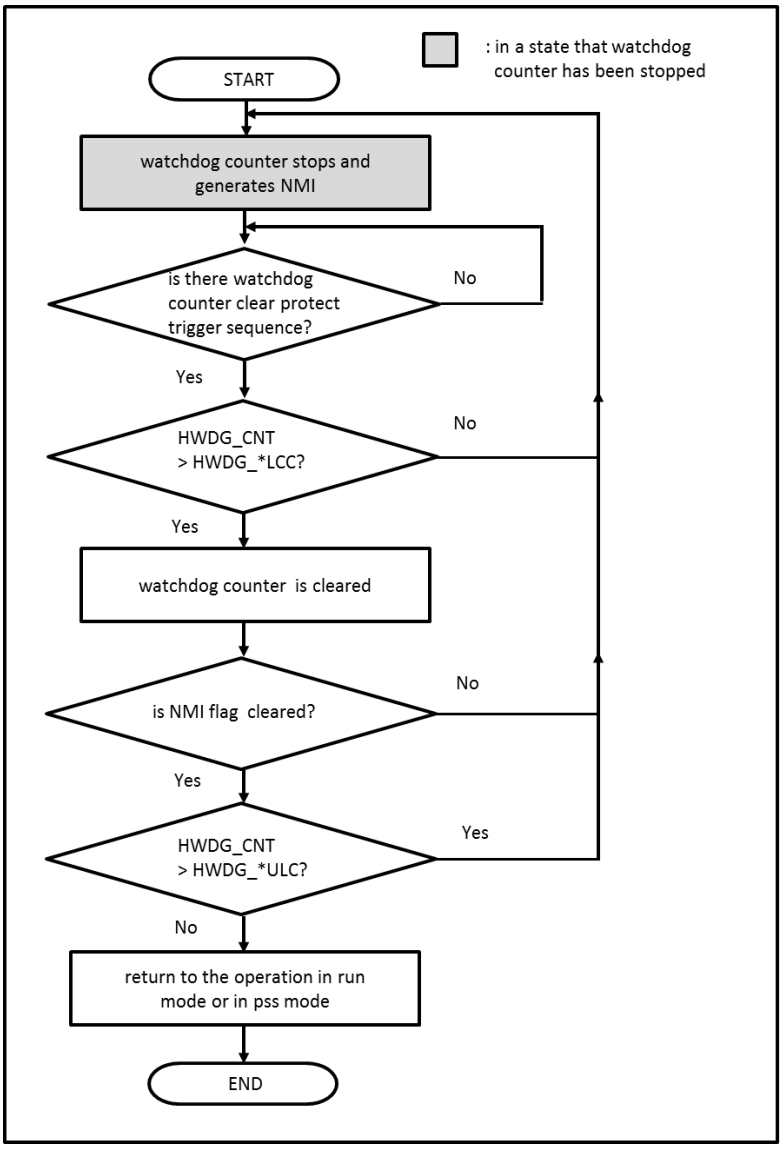
2. When the device state transitions from PSS to RUN, the watchdog counter operation is started immediately. However, the PSS window setting continues to be used until the watchdog counter clear protection trigger sequence is executed.
3. During the operation of the watchdog counter, execute the watchdog counter clear protection trigger sequence within the range between the window upper limit value and lower limit value that have been specified beforehand. Continuously clearing the watchdog counter needs to be done periodically.
4. If a runaway state caused by a user program and prevents periodic clearing, the watchdog counter reaches the window upper limit value. Then, processing transitions to the flow that generates the watchdog reset request or watchdog interrupt request (NMI).
5. According to the value of the IRQEN bit in the Hardware watchdog interrupt configuration register (HWDG_INT), the prior warning interrupt request is generated. At the same time, as many cycles as are set for the delay time in the Hardware watchdog reset delay counter register (HWDG_RSTDLY) are inserted.
6. After the elapse of the time corresponding to the number of cycles for the delay time, the watchdog reset request or watchdog interrupt request (NMI) is generated according to the value of the RSTEN bit in the Hardware watchdog interrupt configuration register (HWDG_INT). When the watchdog interrupt request (NMI) is generated, the processing transitions to the operation of the test function of the watchdog. For details, see Figure 3-6.

Note:

- *If the watchdog counter clear protection trigger sequence is executed during operation with the PSS window setting, this means that the device state has already returned from PSS to RUN.*

The following figure shows the operation of the test function of the Hardware watchdog.

Figure 3-6 Operation of Hardware Watchdog Test Function



1. The watchdog counter is in the stop state.
2. If the watchdog counter clear protection trigger sequence is executed in the stop state of the watchdog counter and the window lower limit value has already been reached at this time, the watchdog counter is cleared. (If this occurs outside the counter range, no special change occurs.)
3. When the NMIFLAG bit in the Hardware watchdog interrupt configuration register (HWDG_INT) is cleared, the watchdog interrupt request (NMI) is cleared.
4. When the watchdog counter has not reached the window upper limit value, processing returns to operation in RUN or PSS. When it has already reached the window upper limit value, no special change occurs.

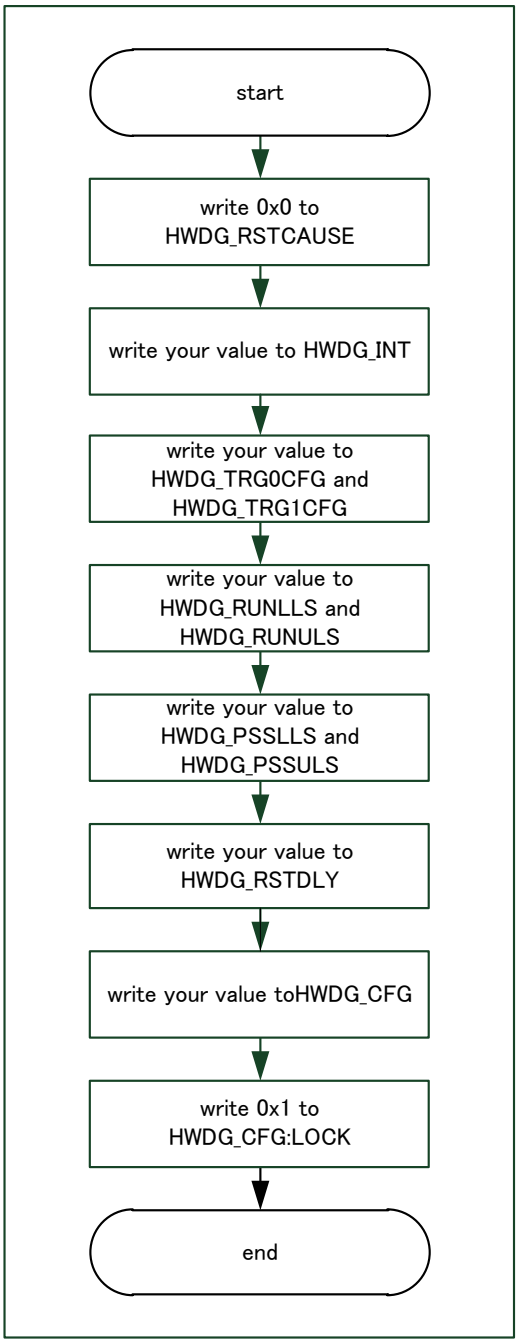
Note:

The notation * in Figure 3-6 means "RUN" or "PSS".

4. Setting Procedure Example

This section explains an example of a procedure for setting the Hardware watchdog timer.

Figure 4-1 Example of Hardware Watchdog Timer Register Setting Procedure



Note:

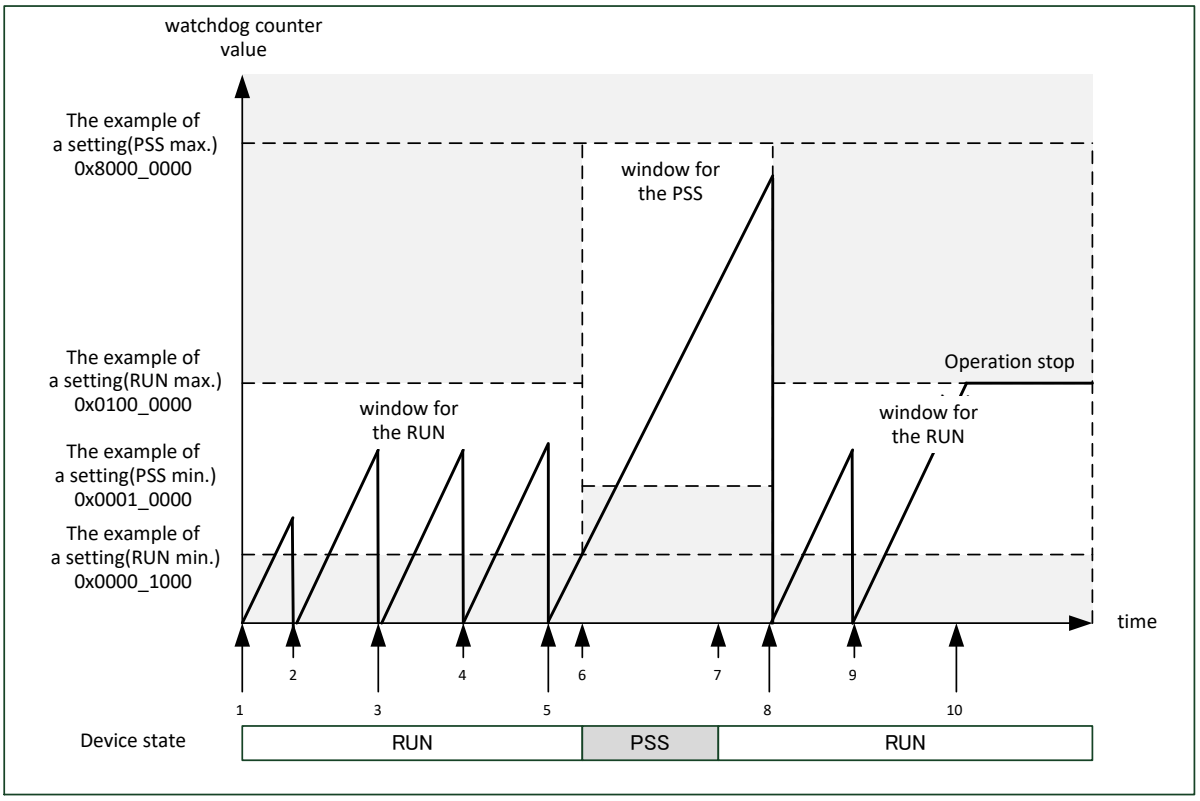
- *In Figure 4-1, the setting order can be changed except for the step of setting the LOCK bit in the Hardware watchdog configuration register (HWDG_CFG) to "1".*

- *Hardware watchdog timer is configured by boot program, and this is determined by value of BootROM marker. For detail of boot program and BootROM marker, see BootROM Software Interface chapter.*
  Therefore, user program has to execute only "write 0x0 to HWDG_RSTCAUSE".

5. Operation Examples

This section describes examples of operating the Hardware watchdog timer.

Figure 5-1 Example of Operating Hardware Watchdog Timer (When Operation in PSS Is Enabled)

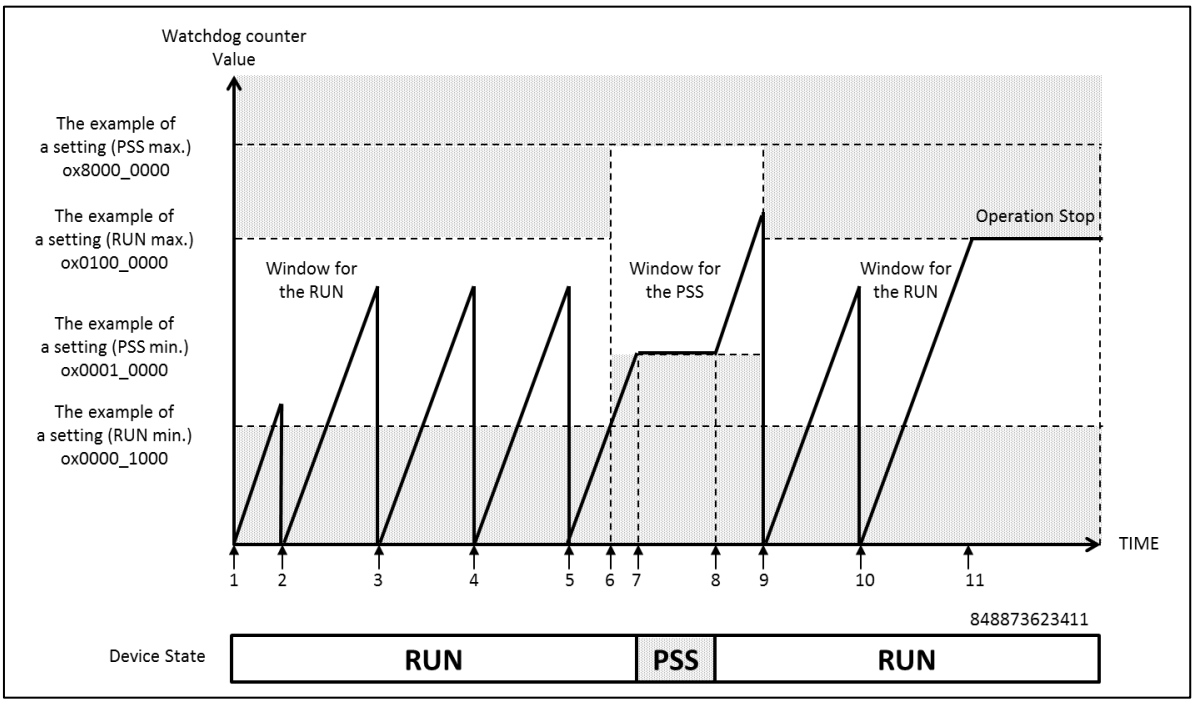


1. After power-on, the hardware watchdog timer starts operation. (The initial value of the window upper limit is 0x01000000.)
2. After writing setting values in the registers, the LOCK bit in the hardware watchdog configuration register (HWDG_CFG) is set to"1". At this time, the watchdog counter is automatically cleared, and it starts up-counting from 0x00000000.
3. The watchdog counter clear protection trigger sequence is executed, and the watchdog counter is cleared.
4. The watchdog counter clear protection trigger sequence is executed, and the watchdog counter is cleared.
5. The watchdog counter clear protection trigger sequence is executed, and the watchdog counter is cleared.
6. The device state transitions from RUN to PSS. At this time, switching to the PSS window setting occurs.
7. The device state returns from PSS to RUN. At this time, the switching to the RUN window setting does not occur immediately.
8. The watchdog counter clear protection trigger sequence is executed, and the watchdog counter is cleared. At this time, the switching to the RUN window setting occurs.
9. The watchdog counter clear protection trigger sequence is executed, and the watchdog counter is cleared.
10. When the watchdog counter reaches the window upper limit value, the watchdog reset request or watchdog interrupt request (NMI) is generated.

Notes:

- *The window setting is not immediately switched to the window setting for RUN when the device state is returned from PSS to RUN.*
- *Before the transition from RUN to PSS, be sure to clear the watchdog counter.*

Figure 5-2 Example of Operating Hardware Watchdog Timer (When Operation in PSS Is Disabled)



1. After power-on, the hardware watchdog timer starts operation. (The initial value of the window upper limit is 0x0100_0000.)
2. After writing setting values in the registers, the LOCK bit in the hardware watchdog configuration register (HWDG_CFG) is set to"1". At this time, the watchdog counter is automatically cleared, and it starts up-counting from 0x00000000.
3. The watchdog counter clear protection trigger sequence is executed, and the watchdog counter is cleared.
4. The watchdog counter clear protection trigger sequence is executed, and the watchdog counter is cleared.
5. The watchdog counter clear protection trigger sequence is executed, and the watchdog counter is cleared.
6. Switching to the PSS window setting occurs.
7. The device state transitions from RUN to PSS. At this time, the Hardware watchdog timer stops operation.
8. The device state returns from PSS to RUN. At this time, the Hardware watchdog timer immediately starts operation. The switching to the RUN window setting does not occur immediately.
9. The watchdog counter clear protection trigger sequence is executed, and the watchdog counter is cleared. At this time, the switching to the RUN window setting occurs.
10. The watchdog counter clear protection trigger sequence is executed, and the watchdog counter is cleared.
11. When the watchdog counter reaches the window upper limit value, the watchdog reset request or watchdog interrupt request (NMI) is generated.

Notes:

- *The window setting is not immediately switched to the window setting for RUN when the device state is returned from PSS to RUN.*
- *After the device state returned from PSS to RUN, the watchdog counter is cleared. At this time, the switching to the RUN window setting occurs.*
- *Certainly set the same value in upper limit registers and lower limit registers.*
- *Set the same value following two lower limit registers.*

  - Hardware watchdog lower limit PSS setting register (HWDG_PSSLLS)
  - Hardware watchdog lower limit RUN setting register (HWDG_RUNLLS)
- *Set the same value following two upper limit registers*
  - Hardware watchdog upper limit PSS setting register (HWDG_PSSULS)
  - Hardware watchdog upper limit RUN setting register (HWDG_RUNULS)
- *The limitation is needed, when device state change from PSS to RUN. Because the window setting don't change immediately. When device state change from RUN to PSS, clear the watchdog counter.*

6. Register List

This section explains the registers of the Hardware watchdog timer.

Table 6-1 List of Hardware Watchdog Timer Registers

Abbreviated Register Name	Register Name	Reference
HWDG_PROT	Hardware watchdog protection register	6.1
HWDG_CNT	Hardware watchdog counter register	6.2
HWDG_RSTCAUSE	Hardware watchdog reset source register	6.3
HWDG_TRG0	Hardware watchdog trigger 0 register	6.4
HWDG_TRG1	Hardware watchdog trigger 1 register	6.5
HWDG_INT	Hardware watchdog interrupt configuration register	6.6
HWDG_INTCLR	Hardware watchdog interrupt clear register	6.7
HWDG_TRG0CFG	Hardware watchdog trigger 0 configuration register	6.8
HWDG_TRG1CFG	Hardware watchdog trigger 1 configuration register	6.9
HWDG_RUNLLS	Hardware watchdog lower limit RUN setting register	6.10
HWDG_RUNULS	Hardware watchdog upper limit RUN setting register	6.11
HWDG_PSSLLS	Hardware watchdog lower limit PSS setting register	6.12
HWDG_PSSULS	Hardware watchdog upper limit PSS setting register	6.13
HWDG_RSTDLY	Hardware watchdog reset delay counter register	6.14
HWDG_CFG	Hardware watchdog configuration register	6.15
HWDG_RUNLLC	Hardware watchdog lower limit RUN current register	6.16
HWDG_RUNULC	Hardware watchdog upper limit RUN current register	6.17
HWDG_PSSLLC	Hardware watchdog lower limit PSS current register	6.18
HWDG_PSSULC	Hardware watchdog upper limit PSS current register	6.19

Table 6-2 Register Memory Layout of Hardware Watchdog Timer

Offset	Register Name			
	+3	+2	+1	+0
0x0000_0000	HWDG_PROT 00000000_00000000_00000000_00000000			
0x0000_0004	-	-	-	-
0x0000_0008	HWDG_CNT 00000000_00000000_00000000_00000000			
0x0000_000C	HWDG_RSTCAUSE 00000000_00000000_00000000_000XXXXX			
0x0000_0010	HWDG_TRG0 00000000_00000000_00000000_00000000			
0x0000_0014	-	-	-	-
0x0000_0018	HWDG_TRG1 00000000_00000000_00000000_00000000			
0x0000_001C	-	-	-	-
0x0000_0020	HWDG_INT 00000000_00000010_00000000_00000000			
0x0000_0024	HWDG_INTCLR 00000000_00000000_00000000_00000000			
0x0000_0028	-	-	-	-

Offset	Register Name			
	+3	+2	+1	+0
0x0000_002C	HWDG_TRG0CFG 00000000_00000000_00000000_00000000			
0x0000_0030	HWDG_TRG1CFG 00000000_00000000_00000000_00000000			
0x0000_0034	HWDG_RUNLLS 00000000_00000000_00000000_00000000			
0x0000_0038	HWDG_RUNULS 00000001_00000000_00000000_00000000			
0x0000_003C	HWDG_PSSLLS 00000000_00000000_00000000_00000000			
0x0000_0040	HWDG_PSSULS 10000000_00000000_00000000_00000000			
0x0000_0044	HWDG_RSTDLY 00000000_00000000_00000000_00000000			
0x0000_0048	HWDG_CFG 00000000_00000000_00000000_00000XXX			
0x0000_004C	HWDG_RUNLLC 00000000_00000000_00000000_00000000			
0x0000_0050	HWDG_RUNULC 00000001_00000000_00000000_00000000			
0x0000_0054	HWDG_PSSLLC 00000000_00000000_00000000_00000000			
0x0000_0058	HWDG_PSSULC 10000000_00000000_00000000_00000000			

Note:

- *"X" indicates an undefined value.*

6.1. Hardware Watchdog Protection Register (HWDG_PROT)

This register is used to execute the watchdog register write protection sequence. Write the correct key (0xEDAC_CE55) to this register before performing write access to each register. (However, the Hardware watchdog trigger 0/1 register (HWDG_TRG0 and HWDG_TRG1) are excluded.) Writing the correct key releases the write protection lock for subsequent register access. Write access to each register enables write protection lock for the registers. (Read access to each register does not affect the lock.) For write access to this register, you must write 32-bit data.

Bit	31 … 0
Field	KEY[31:0]
R/W Attribute	R, W
Protection Attribute	WP
Initial Value	00000000_00000000_00000000_00000000

[bit31:0] KEY[31:0]: Protection Bits

KEY[31:0]	Description
During write operation	When 0xEDAC_CE55 is written: Protection lock for register write is released. When a value other than 0xEDAC_CE55 is written: Protection lock for register write is not released.
During read operation	When 0xFFFF_FFFF is read: Protection lock for register write has been released. When 0x0000_0000 is read: Protection lock for register write has been enabled.

Notes:

- *An AHB transfer error response is generated when write access to this register is performed under any of the conditions below. (This error response invokes the CPU exception handler.)*
  - A wrong key is written to this register.
  - Non-32-bit data is written to this register.
  - Data is written to this register twice in a row.
- *For details on the watchdog register write protection sequence, see Figure 3-1.*
- *A protect key will be locked again by writing to the address where is in the same group area (MCU Config Group), however protect key will not be locked by writing to no protect target register (HWDG_TRG0/1) in Hardware Watchdog.*

6.2. Hardware Watchdog Counter Register (HWDG_CNT)

This register indicates the current up count value of the watchdog counter.

Bit	31	0
Field	WDGCNT[31:0]	
R/W Attribute	R, WX	
Protection Attribute	-	
Initial Value	00000000_00000000_00000000_00000000	

[bit31:0] WDGCNT[31:0]: Watchdog Counter Bits

WDGCNT[31:0]	Description
During write operation	Invalid
During read operation	Returns the current watchdog counter value. The watchdog counter value is sampled by using the bus clock.

Note:

- *An AHB transfer error response is generated where there is write access to this register.*

6.3. Hardware Watchdog Reset Factor Register (HWDG_RSTCAUSE)

When performing write access to this register, follow the watchdog register write protection sequence. This register is the status register that indicates the factor of the watchdog reset request or watchdog interrupt request (NMI). This register is not initialized by reset. After a watchdog reset, the value is maintained. To check the factor of the watchdog reset request, read this register.

Bit	31	8
Field	Reserved	
R/W Attribute	R0, WX	
Protection Attribute	WPS	
Initial Value	00000000_00000000_00000000	

Bit	7	5	4	3	2	1	0
Field	Reserved		RST CAUSE4	RST CAUSE3	RST CAUSE2	RST CAUSE1	RST CAUSE0
R/W Attribute	R0, WX		R, W	R, W	R, W	R, W	R, W
Protection Attribute	WPS						
Initial Value	000		X	X	X	X	X

[bit31:5] Reserved: Reserved Bits

[bit4] RSTCAUSE4: Reset Factor Bit 4

If the watchdog counter clear protection trigger sequence is executed when the LOCK bit in the Hardware watchdog configuration register (HWDG_CFG) is "0", this bit is set to "1".

RSTCAUSE4	Description
During write operation	When "0" is written: This bit cleared When "1" is written: Invalid
During read operation	When "0" is read: Indicating no detection of a reset/NMI factor When "1" is read: Indicating detection of a reset/NMI factor

[bit3] RSTCAUSE3: Reset Factor Bit 3

If the watchdog counter clear protection trigger sequence is executed before the watchdog counter reaches the window lower limit value, this bit is set to "1".

RSTCAUSE3	Description
During write operation	When "0" is written: This bit cleared When "1" is written: Invalid
During read operation	When "0" is read: Indicating no detection of a reset/NMI factor When "1" is read: Indicating detection of a reset/NMI factor

[bit2] RSTCAUSE2: Reset Factor Bit 2

When the watchdog counter reaches the window upper limit value, this bit is set to "1".

RSTCAUSE2	Description
During write operation	When "0" is written: This bit cleared When "1" is written: Invalid
During read operation	When "0" is read: Indicating no detection of a reset/NMI factor When "1" is read: Indicating detection of a reset/NMI factor

When the prior warning interrupt request is used, RSTCAUSE2 cannot be cleared during the period from the occurrence of a prior warning interrupt until a watchdog reset or watchdog interrupt (NMI).

[bit1] RSTCAUSE1: Reset Factor Bit 1

When there is a violation of the watchdog counter clear protection trigger sequence, this bit is set to "1". For details, see Figure 3-2.

.

RSTCAUSE1	Description
During write operation	When "0" is written: This bit cleared When "1" is written: Invalid
During read operation	When "0" is read: Indicating no detection of a reset/NMI factor When "1" is read: Indicating detection of a reset/NMI factor

[bit0] RSTCAUSE0: Reset Factor Bit 0

If the value written in the Hardware watchdog trigger 0/1 register (HWDG_TRG0 or HWDG_TRG1) does not match a proper value, this bit is set to "1".

RSTCAUSE0	Description
During write operation	When "0" is written: This bit cleared When "1" is written: Invalid
During read operation	When "0" is read: Indicating no detection of a reset/NMI factor When "1" is read: Indicating detection of a reset/NMI factor

Notes:

- *When using the Hardware watchdog timer, be sure to clear this register.*
- *The value in this register becomes valid when the watchdog reset request is generated.*

6.4. Hardware Watchdog Trigger 0 Register (HWDG_TRG0)

When performing write access to this register, you do not need to follow the watchdog register write protection sequence. This register is used to execute the watchdog counter clear protection trigger sequence. Write the value defined in the Hardware watchdog trigger 0 configuration register (HWDG_TRG0CFG) to this register. When a value other than the one in the Hardware watchdog trigger 0 configuration register (HWDG_TRG0CFG) is written, the watchdog reset request or watchdog interrupt request (NMI) is generated.

Bit	31 ... 8
Field	Reserved
R/W Attribute	R0, WX
Protection Attribute	-
Initial Value	00000000_00000000_00000000

Bit	7	6	5	4	3	2	1	0
Field	WDGTRG0[7:0]							
R/W Attribute	R0, W							
Protection Attribute	-							
Initial Value	00000000							

[bit31:8] Reserved: Reserved Bits

[bit7:0] WDGTRG0[7:0]: Watchdog Trigger 0 Bits

These bits are used to execute the watchdog counter clear protection trigger sequence to clear the watchdog counter.

WDGTRG0[7:0]	Description
During write operation	When the HWDG_TRG0CFG value is written: 1 condition for executing the watchdog counter clear protection trigger sequence is met. When a value other than the HWDG_TRG0CFG value is written: A watchdog error is generated.
During read operation	Reads 0b00000000.

Note:

- *Figure 3-2 shows the flow of the watchdog counter clear protection trigger sequence.*

6.5. Hardware Watchdog Trigger 1 Register (HWDG_TRG1)

When performing write access to this register, you do not need to follow the watchdog register write protection sequence. This register is used to execute the watchdog counter clear protection trigger sequence. Write the value defined in the Hardware watchdog trigger 1 configuration register (HWDG_TRG1CFG) to this register. When a value other than the one in the Hardware watchdog trigger 1 configuration register (HWDG_TRG1CFG) is written, the watchdog reset request or watchdog interrupt request (NMI) is generated.

Bit	31 ... 8
Field	Reserved
R/W Attribute	R0, WX
Protection Attribute	-
Initial Value	00000000_00000000_00000000

Bit	7	6	5	4	3	2	1	0
Field	WDGTRG1[7:0]							
R/W Attribute	R0, W							
Protection Attribute	-							
Initial Value	00000000							

[bit31:8] Reserved: Reserved Bits

[bit7:0] WDGTRG1[7:0]: Watchdog Trigger 1 Bits

These bits are used to execute the watchdog counter clear protection trigger sequence to clear the watchdog counter.

WDGTRG1[7:0]	Description
During write operation	When the HWDG_TRG1CFG value is written: 1 condition for executing the watchdog counter clear protection trigger sequence is met. When a value other than the HWDG_TRG1CFG value is written: A watchdog error is generated.
During read operation	Reads 0b00000000.

Note:

- *Figure 3-2 shows the flow of the watchdog counter clear protection trigger sequence.*

6.6. Hardware Watchdog Interrupt Configuration Register (HWDG_INT)

When performing write access to this register, follow the watchdog register write protection sequence. This register is used to make settings related to the watchdog resetrequest (NMI) and prior warning interrupt request. This register also includes interrupt status flags.

Bit	31	18	17	16
Field	Reserved		RSTEN	IRQEN
R/W Attribute	R0, WX		R/W	R/W
Protection Attribute	WPS			
Initial Value	000000_00000000		1	0

Bit	15	2	1	0
Field	Reserved		NMI FLAG	IRQ FLAG
R/W Attribute	R0, WX		R, WX	R, WX
Protection Attribute	WPS			
Initial Value	000000_00000000		0	0

[bit31:18] Reserved: Reserved Bits

[bit17] RSTEN: Reset/NMI Enable Bit

This bit is used to generate either the watchdog reset request or watchdog interrupt request (NMI) in response to a watchdog error. Prohibiting setting this bit to "0" except for the purpose of using it as a test function. (If this bit is set to "0" during application execution, security is compromised.) This bit is equipped with a majority circuit with 3 FFs as a measure against a bit invert caused by the effects of noise or another factor.

RSTEN	Description
During write operation	When "0" is written: The watchdog interrupt request (NMI) is generated. When "1" is written: The watchdog reset request is generated.
During read operation	Set value read

[bit16] IRQEN: Prior warning Interrupt Enable Bit

This bit is used to enable generation of the prior warning interrupt request.

IRQEN	Description
During write operation	When "0" is written: The prior warning interrupt request is not generated. When "1" is written: The prior warning interrupt request is generated.
During read operation	Set value read

[bit15:2] Reserved: Reserved Bits

[bit1] NMIFLAG: NMI Flag

This bit is set by a watchdog error when the RSTEN bit in the Hardware watchdog interrupt configuration register (HWDG_INT) is "0". When the RSTEN bit in the Hardware watchdog interrupt clear register (HWDG_INT) is "1", the watchdog reset request is generated. You can clear this bit by writing "1" to the NMICLR bit in the Hardware watchdog interrupt configuration register (HWDG_INTCLR).

NMIFLAG	Description
During write operation	Invalid
During read operation	When "0" is read: No detection of a watchdog error (NMI) is indicated. When "1" is read: Detection of a watchdog error (NMI) is indicated.

[bit0] IRQFLAG: IRQ Flag

This bit is set by a watchdog error. The prior warning interrupt is generated when the IRQEN bit in the Hardware watchdog interrupt clear register (HWDG_INT) is "1". You can clear this bit by writing "1" to the IRQCLR bit in the Hardware watchdog interrupt configuration register (HWDG_INTCLR).

IRQFLAG	Description
During write operation	Invalid
During read operation	When "0" is read: No detection of a watchdog error (IRQ) is indicated. When "1" is read: Detection of a watchdog error (IRQ) is indicated.

Notes:

- *For details on the watchdog interrupt request (NMI), see "3 Explanation of Operation Generation of watchdog reset request or watchdog interrupt request (NMI)."*
- *For details on watchdog errors, see "7 Precautions for Using This Device Watchdog Error."*

6.7. Hardware Watchdog Interrupt Clear Register (HWDG_INTCLR)

When performing write access to this register, follow the watchdog register write protection sequence. This register is used to clear the NMI flag and IRQ flag in the Hardware watchdog interrupt configuration register (HWDG_INT). You can write data to this register even after the LOCK bit in the Hardware watchdog configuration register (HWDG_CFG) is set.

Bit	31	16
Field	Reserved	
R/W Attribute	R0, WX	
Protection Attribute	WPS	
Initial Value	00000000_00000000	

Bit	15	4	3	2	1	0
Field	Reserved				NMI CLR	IRQ CLR
R/W Attribute	R0, WX				R0, W	R0, W
Protection Attribute	WPS					
Initial Value	000000_00000000				0	0

[bit31:2] Reserved: Reserved Bits

[bit1] NMICLR: NMI Clear Bit

This bit is used to clear the NMIFLAG bit in the Hardware watchdog interrupt configuration register (HWDG_INT).

NMICLR	Description
During write operation	When "0" is written: Invalid When "1" is written: The NMI flag is cleared.
During read operation	"0" is read.

You must clear the NMI after waiting for watchdog counter clear (0x0000_0000). Check the watchdog counter value with HWDG_CNT.

[bit0] IRQCLR: Prior warning Interrupt Clear Bit

This bit is used to clear the IRQFLAG bit in the Hardware watchdog interrupt configuration register (HWDG_INT).

IRQCLR	Description
During write operation	When "0" is written: Invalid When "1" is written: The IRQ flag is cleared.
During read operation	"0" is read.

6.8. Hardware Watchdog Trigger 0 Configuration Register (HWDG_TRG0CFG)

When performing write access to this register, follow the watchdog register write protection sequence. This register is used to define a valid value to be written to the Hardware watchdog trigger 0 register (HWDG_TRG0) when the watchdog counter clear protection trigger sequence is performed.

Bit	31 ... 8
Field	Reserved
R/W Attribute	R0, WX
Protection Attribute	WPS
Initial Value	00000000_00000000_00000000

Bit	7	6	5	4	3	2	1	0
Field	WDGTRG0CFG[7:0]							
R/W Attribute	R/W							
Protection Attribute	WPS							
Initial Value	00000000							

[bit31:8] Reserved: Reserved Bits

[bit7:0] WDGTRG0CFG[7:0]: Watchdog Trigger 0 Configuration Bits

These bits are used to define a value to be written to the Hardware watchdog trigger 0 register (HWDG_TRG0) for execution of the watchdog counter clear protection trigger sequence.

WDGTRG0CFG[7:0]	Description
During write operation	Set value written
During read operation	Set value read

6.9. Hardware Watchdog Trigger 1 Configuration Register (HWDG_TRG1CFG)

When performing write access to this register, follow the watchdog register write protection sequence. This register is used to define a valid value to be written to the Hardware watchdog trigger 1 register (HWDG_TRG1) when the watchdog counter clear protection trigger sequence is performed.

Bit	31	8
Field	Reserved	
R/W Attribute	R0, WX	
Protection Attribute	WPS	
Initial Value	00000000_00000000_00000000	

Bit	7	6	5	4	3	2	1	0
Field	WDGTRG1CFG[7:0]							
R/W Attribute	R/W							
Protection Attribute	WPS							
Initial Value	00000000							

[bit31:8] Reserved: Reserved Bits

[bit7:0] WDGTRG1CFG[7:0]: Watchdog Trigger 1 Configuration Bits

These bits are used to define a value to be written to the Hardware watchdog trigger 1 register (HWDG_TRG1) for execution of the watchdog counter clear protection trigger sequence.

WDGTRG1CFG[7:0]	Description
During write operation	Set value written
During read operation	Set value read

6.10. Hardware Watchdog Lower Limit RUN Setting Register (HWDG_RUNLLS)

When performing write access to this register, follow the watchdog register write protection sequence. A setting value of the window lower limit value for RUN is written to this register. However, this register value is different from the window lower limit value for RUN that is actually used. For details on how to reflect this register value in the window lower limit value for RUN actually used, see "6.16 Hardware Watchdog Lower Limit RUN Current Register (HWDG_RUNLLC)." You cannot change the set value of this register after the LOCK bit in the Hardware watchdog configuration register (HWDG_CFG) is set.

Bit	31	0
Field	WDGRUNLLS[31:0]	
R/W Attribute	R/W	
Protection Attribute	WPS	
Initial Value	00000000_00000000_00000000_00000000	

[bit31:0] WDGRUNLLS[31:0]: Window Lower Limit Value for RUN Set Bits

These bits are used to define the window lower limit value for RUN. The reading of these bits returns the set value regardless of the Hardware watchdog configuration register (HWDG_CFG) LOCK bit value.

WDGRUNLLS[31:0]	Description
During write operation	When "0" is written to all 32 bits: The window function is disabled. When any value except "0" is written: The window function is enabled.
During read operation	Set value read

Note:

- *This register value is different from the window lower limit value that is actually used. The window lower limit value actually used is the value in the Hardware watchdog lower limit RUN current register (HWDG_RUNLLC).*

6.11. Hardware Watchdog Upper Limit RUN Setting Register (HWDG_RUNULS)

When performing write access to this register, follow the watchdog register write protection sequence. A setting value of the window upper limit value for RUN is written to this register. However, this register value is different from the window upper limit value for RUN that is actually used. For details on how to reflect this register value in the window upper limit value for RUN actually used, see "6.17 Hardware Watchdog Upper Limit RUN Current Register (HWDG_RUNULC)." You cannot change the set value of this register after the LOCK bit in the Hardware watchdog configuration register (HWDG_CFG) is set.

Bit	31	0
Field	WDGRUNULS[31:0]	
R/W Attribute	R/W	
Protection Attribute	WPS	
Initial Value	00000001_00000000_00000000_00000000	

[bit31:0] WDGRUNULS[31:0]: Window Upper Limit Value for RUN Set Bits

These bits are used to define the window upper limit value for RUN. The reading of these bits returns the set value regardless of the Hardware watchdog configuration register (HWDG_CFG) LOCK bit value.

WDGRUNULS[31:0]	Description
During write operation	Set value written
During read operation	Set value read

Note:

- *This register value is different from the window upper limit value that is actually used. The window upper limit value actually used is the value in the Hardware watchdog upper limit RUN current register (HWDG_RUNULC).*

6.12. Hardware Watchdog Lower Limit PSS Setting Register (HWDG_PSSLLS)

When performing write access to this register, follow the watchdog register write protection sequence. A setting value of the window lower limit value for PSS is written to this register. However, this register value is different from the window lower limit value for PSS that is actually used. For details on how to reflect this register value in the window lower limit value for PSS actually used, see "6.18 Hardware Watchdog Lower Limit PSS Current Register (HWDG_PSSLLC)." You cannot change the set value of this register after the LOCK bit in the Hardware watchdog configuration register (HWDG_CFG) is set.

Bit	31 … 0
Field	WDGPSSLLS[31:0]
R/W Attribute	R/W
Protection Attribute	WPS
Initial Value	00000000_00000000_00000000_00000000

[bit31:0] WDGPSSLLS[31:0]: Window Lower Limit Value for PSS Set Bits

These bits are used to define the window lower limit value for PSS. The reading of these bits returns the set value regardless of the Hardware watchdog configuration register (HWDG_CFG) LOCK bit value.

WDGPSSLLS[31:0]	Description
During write operation	When "0" is written to all 32 bits: The window function is disabled. When any value except "0" is written: The window function is enabled.
During read operation	Set value read

Note:

- *This register value is different from the window lower limit value that is actually used. The window lower limit value actually used is the value in the Hardware watchdog lower limit PSS current register (HWDG_PSSLLC).*

6.13. Hardware Watchdog Upper Limit PSS Setting Register (HWDG_PSSULS)

When performing write access to this register, follow the watchdog register write protection sequence. A setting value of the window upper limit value for PSS is written to this register. However, this register value is different from the window upper limit value for PSS that is actually used. For details on how to reflect this register value in the window upper limit value for PSS actually used, see "6.19 Hardware Watchdog Upper Limit PSS Current Register (HWDG_PSSULC)." You cannot change the set value of this register after the LOCK bit in the Hardware watchdog configuration register (HWDG_CFG) is set.

Bit	31	0
Field	WDGPSSULS[31:0]	
R/W Attribute	R/W	
Protection Attribute	WPS	
Initial Value	10000000_00000000_00000000_00000000	

[bit31:0] WDGPSSULS[31:0]: Window Upper Limit Value for PSS Set Bits

These bits are used to define the window upper limit value for PSS. The reading of these bits returns the set value regardless of the Hardware watchdog configuration register (HWDG_CFG) LOCK bit value.

WDGPSSULS[31:0]	Description
During write operation	Set value written
During read operation	Set value read

Note:

- *This register value is different from the window upper limit value that is actually used. The window upper limit value actually used is the value in the Hardware watchdog upper limit PSS current register(HWDG_PSSULC).*

6.14. Hardware Watchdog Reset Delay Counter Register (HWDG_RSTDLY)

When performing write access to this register, follow the watchdog register write protection sequence. This register is used to set the number of cycles for the delay time from the occurrence of a watchdog error to the watchdog reset request or watchdog interrupt request (NMI). The reference clock for the delay time is the source clock of the watchdog counter. When the IRQEN bit in the Hardware watchdog interrupt configuration register (HWDG_INT) is "1", this register indicates the number of cycles for the delay time from the prior warning interrupt request to the watchdog reset request or watchdog interrupt request (NMI).

Bit	bit31	bit16
Field	Reserved	
R/W Attribute	R0, WX	
Protection Attribute	WPS	
Initial Value	00000000_00000000	

Bit	bit15	bit0
Field	WDGRSTDLY[15:0]	
R/W Attribute	R0, W	
Protection Attribute	WPS	
Initial Value	00000000_00000000	

[bit31:16] Reserved: Reserved Bits

[bit15:0] WDGRSTDLY[15:0]: Reset/NMI Delay Counter Bits

These bits define the number of cycles for the delay time that is to be inserted before generation of the watchdog reset request or watchdog interrupt request (NMI). The reference clock for this delay time is the source clock of the watchdog counter.

WDGRSTDLY[15:0]	Description
During write operation	Set value written
During read operation	0x0000 is read.

6.15. Hardware Watchdog Configuration Register (HWDG_CFG)

When performing write access to this register, follow the watchdog register write protection sequence. This register is used to make the operation settings of the Hardware watchdog timer.

Bit	bit31	bit25	bit24
Field	Reserved		LOCK
R/W Attribute	R0, WX		R, W
Protection Attribute	WPS		
Initial Value	0000000		0

Bit	bit23	bit21	bit20	bit19	bit18	bit17	bit16
Field	Reserved		OBSSEL[4:0]				
R/W Attribute	R0, WX		R/W				
Protection Attribute	WPS						
Initial Value	000		00000				

Bit	bit15	bit10	bit9	bit8
Field	Reserved		CLKSEL[1:0]	
R/W Attribute	R0, WX		R/W	
Protection Attribute	WPS			
Initial Value	000000		00	

Bit	bit7	bit3	bit2	bit1	bit0
Field	Reserved		ALLOWS TOP CLK	WDEN PSS	WDEN RUN
R/W Attribute	R0, WX		R, WX	R, WX	R, WX
Protection Attribute	WPS				
Initial Value	00000		X	X	X

[bit31:25] Reserved: Reserved Bits

[bit24] LOCK: Lock Bit

You can write data to this bit only once. This register is used to prevent rewriting of the set values of various registers. When this bit is "0", you can rewrite the setting values of various registers. When this bit is set to "1" the watchdog counter is automatically cleared. This bit is equipped with a majority circuit with 3 FFs as a measure against a bit invert caused by the effects of noise or another factor.

This bit is set up by BootROM software.

LOCK	Description
During write operation	When "0" is written: Invalid When "1" is written: The set values of the registers are locked.
During read operation	When "0" is read: The lock for the register set values is disabled. When "1" is read: The lock for the register set values is enabled.

[bit23:21] Reserved: Reserved Bits

[bit20:16] OBSSEL[4:0]: Watchdog Counter Monitor Bit Output Selection Bits

These bits are used for selecting any 1 bit in the watchdog counter (32 bits) as the output of the watchdog counter monitor bit.(Support target of this function is different depends on product specification. For details, see "Output of the Watchdog Counter Monitor Bit for MCU Output Pin" in 3 Explanation of Operation.)

OBSSEL[4:0]	Description
During write operation	When 0b00000 is written: Bit 0 is selected for the monitor output. When 0b00001 is written: Bit 1 is selected for the monitor output. When 0b00010 is written: Bit 2 is selected for the monitor output. · · · When 0b11111 is written: Bit 31 is selected for the monitor output.
During read operation	Set value read

Note:

- *If products do not support "Output of watchdog counter monitor bit for MCU output pin", any data written to these bits does not affect operation.*

[bit15:10] Reserved: Reserved Bits

[bit9:8] CLKSEL[1:0]: Clock Selection Bits

These bits are used to select the source clock for the watchdog counter. When activated, the watchdog counter starts its operation as the high-speed CR clock. For details on clock switching, see "7 Precautions for Using This Device Switching of watchdog counter source clock.

CLKSEL[1:0]	Description
During write operation	When 0bX0 is written: The high-speed CR clock is selected. When 0bX1 is written: The low-speed CR clock is selected. Writing data in CLKSEL[1] bits does not affect operation.
During read operation	Set value read

[bit7:3] Reserved: Reserved Bits

[bit2] ALLOWSTOPCLK: Clock Stop for PSS Enable Bit

This bit is used to enable transition to PSS that involves the source clock stop of the watchdog counter. This bit is valid only when the WDENPSS bit in the Hardware watchdog configuration register (HWDG_CFG) is "1". (This bit is fixed by hardware, "0" is always read.)

This bit indicates the inverted value of the PSS enable signal.

See This section describes the block diagrams of the Hardware watchdog timer.

ALLOWSTOPCLK	Description
During write operation	Invalid
During read operation	When "0" is read: Watchdog clock stop in PSS is disabled. When "1" is read: Watchdog clock stop in PSS is enabled.

[bit1] WDENPSS: Watchdog Counter for PSS Enable Bit

This bit is used to enable the watchdog counter in PSS. This bit is enabled when the LOCK bit in the Hardware watchdog configuration register (HWDG_CFG) is set to "1". (This bit is fixed by hardware, "0" is always read.)

WDENPSS	Description
During write operation	Invalid
During read operation	When "0" is read: The watchdog counter in PSS is disabled. When "1" is read: The watchdog counter in PSS is enabled.

[bit0] WDENRUN: Watchdog Counter for RUN Enable Bit

This bit is used to enable the watchdog counter in RUN. This bit is enabled when the LOCK bit in the Hardware watchdog configuration register (HWDG_CFG) is set to "1".

This bit indicates the value of the entire system enable signal.

See This section describes the block diagrams of the Hardware watchdog timer.

WDENRUN	Description
During write operation	Invalid
During read operation	When "0" is read: The watchdog counter in RUN is disabled. When "1" is read: The watchdog counter in RUN is enabled.

Note:

- *When the ALLOWSTOPCLK bit in the Hardware watchdog configuration register (HWDG_CFG) is "1", transition to PSS that involves clock stop is enabled. However, in case of transition from RUN to PSS during reset delay counter operation, the reset delay counter is stopped until a return from PSS to RUN occurs.*

6.16. Hardware Watchdog Lower Limit RUN Current Register (HWDG_RUNLLC)

This register is used to read the window lower limit value for RUN that is actually used. This register fetches the set value in the Hardware watchdog lower limit RUN setting register (HWDG_RUNLLS) when the LOCK bit in the Hardware watchdog configuration register (HWDG_CFG) is set. You cannot change the set value of this register after the LOCK bit in the Hardware watchdog configuration register (HWDG_CFG) is set.

Bit	bit31 ... bit0
Field	WDGRUNLLC[31:0]
R/W Attribute	R, WX
Protection Attribute	-
Initial Value	00000000_00000000_00000000_00000000

[bit31:0] WDGRUNLLC[31:0]: Window Lower Limit for RUN Current Bits

These bits are used to define the window lower limit value for RUN. The reading of these bits returns the initial value, until the Hardware watchdog configuration register (HWDG_CFG) LOCK bit is set.

WDGRUNLLC[31:0]	Description
During write operation	Invalid
During read operation	Set value read

Note:

- *An AHB transfer error response is generated where there is write access to this register.*

6.17. Hardware Watchdog Upper Limit RUN Current Register (HWDG_RUNULC)

This register is used to read the window upper limit value for RUN that is actually used. This register fetches the set value in the Hardware watchdog upper limit RUN setting register (HWDG_RUNULS) when the LOCK bit in the Hardware watchdog configuration register (HWDG_CFG) is set. Also, you cannot change the set value of this register after the LOCK bit in the Hardware watchdog configuration register (HWDG_CFG) is set.

Bit	bit31	bit0
Field	WDGRUNULC[31:0]	
R/W Attribute	R, WX	
Protection Attribute	-	
Initial Value	00000001_00000000_00000000_00000000	

[bit31:0] WDGRUNULC[31:0]: Window Upper Limit for RUN Current Bits

These bits are used to define the window upper limit value for RUN. The reading of these bits returns the initial value, until the Hardware watchdog configuration register (HWDG_CFG) LOCK bit is set.

WDGRUNULC[31:0]	Description
During write operation	Invalid
During read operation	Set value read

Note:

- *An AHB transfer error response is generated where there is write access to this register.*

6.18. Hardware Watchdog Lower Limit PSS Current Register (HWDG_PSSLLC)

This register is used to read the window lower limit value for PSS that is actually used. This register fetches the set value in the Hardware watchdog lower limit PSS setting register (HWDG_PSSLLS) when the LOCK bit in the Hardware watchdog configuration register (HWDG_CFG) is set. You cannot change the set value of this register after the LOCK bit in the Hardware watchdog configuration register (HWDG_CFG) is set.

Bit	bit31	bit0
Field	WDGPSSLLC[31:0]	
R/W Attribute	R, WX	
Protection Attribute	-	
Initial Value	00000000_00000000_00000000_00000000	

[bit31:0] WDGPSSLLC[31:0]: Window Lower Limit for PSS Current Bits

These bits are used to define the window lower limit value for PSS. The reading of these bits returns the initial value, until the Hardware watchdog configuration register (HWDG_CFG) LOCK bit is set.

WDGPSSLLC[31:0]	Description
During write operation	Invalid
During read operation	Set value read

Note:

- *An AHB transfer error response is generated where there is write access to this register.*

6.19. Hardware Watchdog Upper Limit PSS Current Register (HWDG_PSSULC)

This register is used to read the window upper limit value for PSS that is actually used. This register fetches the set value in the Hardware watchdog upper limit PSS setting register (HWDG_PSSULS) when the LOCK bit in the Hardware watchdog configuration register (HWDG_CFG) is set. You cannot change the set value of this register after the LOCK bit in the Hardware watchdog configuration register (HWDG_CFG) is set.

Bit	bit31	bit0
Field	WDGPSSULC[31:0]	
R/W Attribute	R, WX	
Protection Attribute	-	
Initial Value	10000000_00000000_00000000_00000000	

[bit31:0] WDGPSSULC[31:0]: Window Upper Limit for PSS Current Bits

These bits are used to define the window upper limit value for PSS. The reading of these bits returns the initial value, until the Hardware watchdog configuration register (HWDG_CFG) LOCK bit is set.

WDGPSSULC[31:0]	Description
During write operation	Invalid
During read operation	Set value read

Note:

- *Write access to this register generates an AHB transfer error response.*

7. Precautions for Using This Device

This section describes precautions on the use of the Hardware watchdog timer.

Watchdog Window Setting

For the Hardware watchdog timer, you can make 2 types of window setting, the setting for RUN and setting for PSS. The user must make an appropriate window setting for each device state.

When the device transitions from PSS to RUN, the Hardware watchdog timer continues operation using the PSS window setting until the watchdog counter clear protection trigger sequence is executed. The watchdog counter clear protection trigger sequence switches the window setting to the RUN window setting. You can control the watchdog function in each device state with the WDENRUN bit and WDENPSS bit in the Hardware watchdog configuration register (HWDG_CFG).

Notes:

- *When WDENPSS bit is "0", the watchdog timer perform the same operation. So certainly set the same value in upper limit registers and lower limit registers.*
- *Set the same value following two upper limit registers.*
  - Hardware watchdog lower limit PSS setting register (HWDG_PSSLLS)
  - Hardware watchdog lower limit RUN setting register (HWDG_RUNLLS)
- *Set the same value following two upper limit registers.*
  - Hardware watchdog upper limit PSS setting register (HWDG_PSSULS)
  - Hardware watchdog upper limit RUN setting register (HWDG_RUNULS)

Switching of Watchdog Counter Source Clock

The system controller controls the switching of the watchdog counter source clock since the source clock control of the watchdog counter is a part of the system setting information (profile). To change the watchdog counter source clock, follow the sequence below.

1. Setting the CLKSEL bit in the Hardware watchdog configuration register (HWDG_CFG) sets the new source clock of the watchdog counter. Then, writing data to the following registers finalizes the window setting.
2. Hardware watchdog lower limit RUN setting register (HWDG_RUNLLS)
3. Hardware watchdog upper limit RUN setting register (HWDG_RUNULS)
4. Hardware watchdog lower limit PSS setting register (HWDG_PSSLLS)
5. Hardware watchdog upper limit PSS setting register (HWDG_PSSULS)
6. Setting the LOCK bit in the Hardware watchdog configuration register (HWDG_CFG) to "1" locks rewriting of the register set values.
7. Writing 0xAB to the RUN profile update trigger register of the system controller (SYSC0_TRGRUNCNTR) executes RUN profile update. At this point in time, the source clock of the watchdog counter is switched to the new source clock selected by the CLKSEL bit of the Hardware watchdog configuration register (HWDG_CFG).

Watchdog Error

A watchdog error is generated under the following conditions. This watchdog error generates the watchdog reset request/interrupt request (NMI).

1. An incorrect value is written to the Hardware watchdog trigger 0/1 register (HWDG_TRG0 or HWDG_TRG1).
2. The watchdog counter clear protection trigger sequence is violated.
3. The watchdog counter clear protection trigger sequence is not executed before the watchdog counter reaches the window upper limit value.
4. The watchdog counter clear protection trigger sequence is executed before the watchdog counter reaches the window lower limit value.
5. The watchdog counter clear protection trigger sequence is executed while the LOCK bit in the Hardware watchdog configuration register (HWDG_CFG) is still "0".

Factors for Bus Error Responses (Data Abort)

A bus error response is generated in the following cases.

- Access (read or write) to an address where no register exists
- Violation of the watchdog register write protection sequence
- Write access to a register with the read-only attribute (indicated as R, WX)
- Write access to a register after the LOCK bit in the Hardware watchdog configuration register (HWDG_CFG) is set to "1"
- Write access with an incorrect value to the Hardware watchdog protection register (HWDG_PROT)
- Byte/Half-word write access to the Hardware watchdog protection register (HWDG_PROT)
- Write access to a register in non-privileged mode
  - (Except for the Hardware watchdog trigger 0/1 register (HWDG_TRG0 and HWDG_TRG1))

Watchdog Timer Operation in Standby State of Processor

The Hardware watchdog timer operates using the RUN window setting in the following case.

- Without a valid key being written to the PSS profile update enable register (SYSC0_PSSENR.PSSEN0 and SYSC1_PSSENR.PSSEN1), the processor transitions to the standby state by executing a wait for interrupt (WFI) instruction.

CHAPTER 15: Software Watchdog Timer

This section describes the Software watchdog timer.

1. **Overview**
2. **Configuration and Block Diagram**
3. **Explanation of Operation**
4. **Setting Procedure Example**
5. **Operation Examples**
6. **Register List**
7. **Precautions for Using This Device**

SWDT-TXXPT03P01R01L08-E1-XX

1. Overview

This section provides an overview of the Software watchdog timer.

The Software watchdog timer is positioned in the SYSC1 group, used for detecting a runaway state caused by a user program, also the Software watchdog timer monitors user program by the CPU.

Software Watchdog Timer Features

- Detecting a runaway state of the software generates a reset request. (*a)
- A user program starts the software watchdog timer. (*a)

- The watchdog counter source clock is selected from 4 types. (*a)
  (High-speed CR clock (default), low-speed CR clock, sub clock, or main clock)
- A user program is used to set registers. (*a)
- It operates as a 32-bit watchdog counter.
- Different initializations between soft reset and hard reset can be applied.
- Soft reset clears only the watchdog counter.
- Hard reset initializes the watchdog counter, register setting, and internal circuits.
- It monitors the watchdog counter clear protection trigger sequence.
- The window watchdog function is implemented.
- Different operation settings are possible for each device state (RUN and PSS).
- It monitors the watchdog register write protection sequence.
- The majority circuits for bits that affect important functions are implemented.

- It can generate the watchdog reset request or watchdog interrupt request (NMI) in response to a watchdog error.
- It can generate a prior warning interrupt request before the reset request or interrupt request (NMI).
- It provides the function for outputting the watchdog counter monitor bit for the MCU output pin so that the watchdog timer counter can be observed from the outside. (Support target of this function is different depends on product specification. For details, see “Output of the Watchdog Counter Monitor Bit for MCU Output Pin” in 3 Explanation of Operation.
- The watchdog counter can be stopped in the debugging state of the processor.

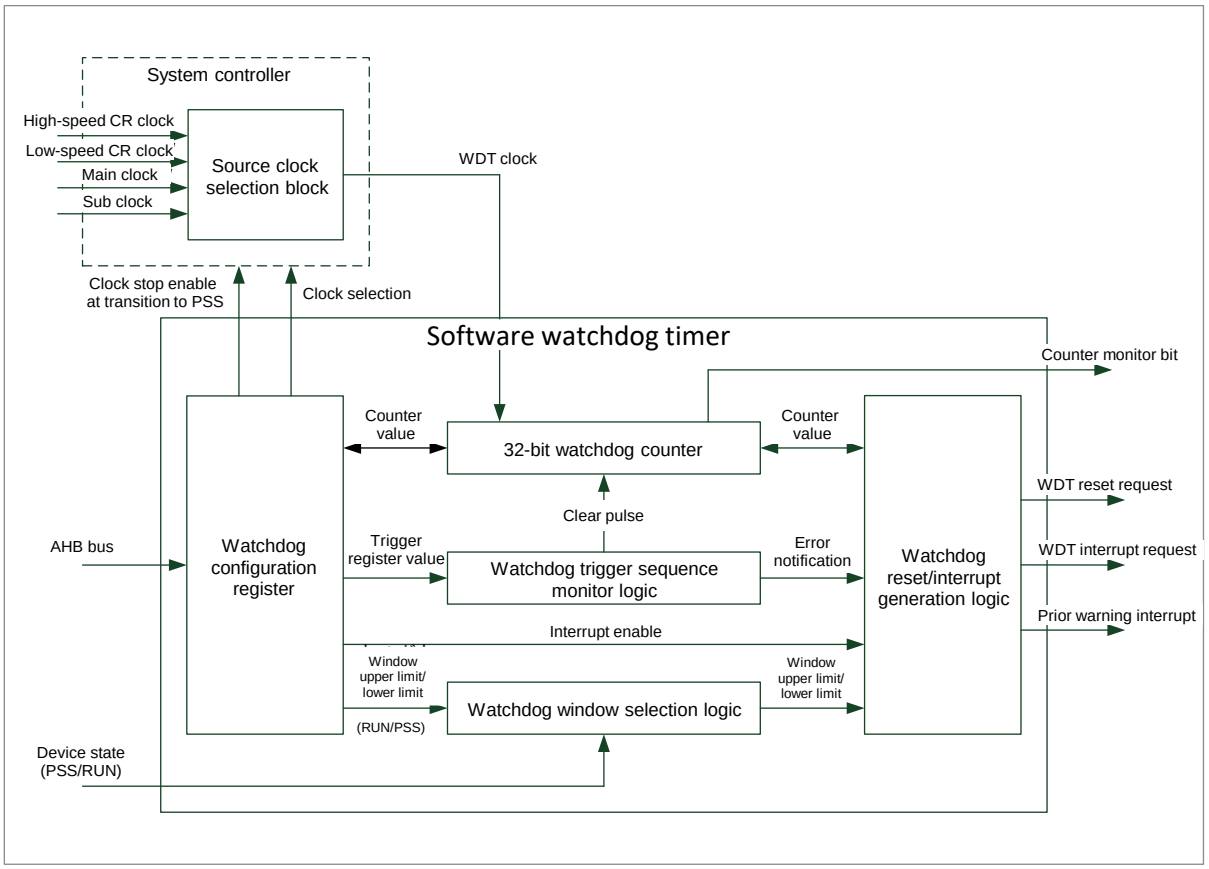
Notes:

- The notation (*a) indicates a feature that is specific to the software watchdog timer compared with the hardware watchdog.
- A watchdog error is a factor that causes the watchdog reset request or watchdog interrupt request (NMI).

2. Configuration and Block Diagram

This section describes the block diagrams of the Software watchdog timer.

Figure 2-1 Software Watchdog Timer Block Diagram



- Watchdog configuration register:

  In this block, the set value of each register is stored. I/O operations for information including the counter value for read operation, trigger register value necessary for clearing the counter, and window upper limit/lower limit values are performed.

- 32-bit watchdog counter:

  This is the block of the 32-bit up counter.

  It provides the function for outputting the watchdog counter monitor bit for the MCU output pin.

- Watchdog window selection logic:

  This block fetches the window upper limit value and lower limit value for each device state (RUN and PSS) and selects a window setting according to the device state transition.

- Watchdog reset/interrupt generation logic:

  This block generates the watchdog reset request, watchdog interrupt request (NMI), and prior warning interrupt request.

3. Explanation of Operation

This section describes operation of the Software watchdog timer.

Software Watchdog Timer Functions

The Software watchdog timer implements the following functions.

Reset Request Generation by Detection of Runaway State of Software

The software watchdog timer monitors user program execution by the CPU. The timer starts the monitoring when it is activated by a user program. If the watchdog counter is not cleared within the range between the upper limit value and the lower limit value of the window specified beforehand, this state is judged to be a runaway state of a user program. In this case, a reset request or an interrupt request (NMI) is generated.

Note:

- *For example, after the watchdog timer is activated by a user program, it can detect a software runaway state in which a user program goes out of control as a result of an abnormal change in the program counter (PC).*

Software Watchdog Timer Start by User Program

A user program starts the software watchdog timer. Specifically, operation is started by setting the LOCK bit in the software watchdog configuration register (SWDG_CFG).

Software Watchdog Timer Control by CPU

The software watchdog timer monitors control by CPU over the entire system. The number of software watchdog timers to be mounted in the MCU is 1.

Watchdog Counter Source Clock Selection (4 Types)

The software watchdog timer sets values in the CLKSEL[1:0] bits in the software watchdog configuration register (SWDG_CFG) to select a source clock for the watchdog counter from among 4 types of source clocks. Specifically, it selects a source clock from the high-speed CR clock, low-speed CR clock, sub clock, or main clock. (The initial setting is for the high-speed CR clock.)

Selection of a source clock is a clock request to the system controller. Source clock switching is performed on the clock system side of the system controller. For details on the procedure, see "7. Precautions for Using This Device Switching of watchdog counter source clock."

Register Setting by User Program

Use a user program to set software watchdog timer registers.

32-Bit Watchdog Counter

The Software watchdog timer operates by using the 32-bit watchdog counter (up counter) (the initial value is 0x0000_0000).

The watchdog counter operates when all the following conditions are met.

1. The device is operating in user mode.
2. The watchdog counter has not reached the window upper limit value.
3. The device state is RUN and the WDENRUN bit is "1".
   (See "6.15 Software Watchdog Configuration Register (SWDG_CFG).")
4. The device state is PSS and the WDENPSS bit is "1".
   (See "6.15 Software Watchdog Configuration Register (SWDG_CFG).")

Note:

- *When the watchdog counter reaches the window upper limit value, the count operation stops.*

The following Table 3-1 shows the relationship between the window upper limit value and count time.

Table 3-1 Relationship between Software Window Upper Limit Value of Watchdog Timer and Count Time

Input Clock Frequency	Window Upper Limit Value	Count Time	Remarks
8MHz	0x0100_0000	About 2.1 s	It operates at the initial value of the Software watchdog upper RUN setting register (SWDG_RUNULS).
12MHz	0x0100_0000	About 1.4 s	
8MHz	0x8000_0000	About 268 s	It operates at the initial value of the Software watchdog upper PSS setting register (SWDG_PSSULS).
12MHz	0x8000_0000	About 178 s	

Note:

- *The above table is an example involving high-speed CR clock frequencies of 8 MHz and 12 MHz.*

Different Initializations between Soft Reset and Hard Reset

For the Software watchdog timer, different initializations between soft reset and hard reset can be applied. The following Table 3-2 Ranges of Initialization for Soft Reset and Hard Reset shows the ranges of initialization for soft reset and hard reset.

Table 3-2 Ranges of Initialization for Soft Reset and Hard Reset

Conditions	Reset Name	Range of Initialization
1	Soft reset	Watchdog counter
2	Hard reset	Watchdog counter and settings for all registers

Watchdog Counter Clear Protection Trigger Sequence

The Software watchdog timer monitors the watchdog counter clear protection trigger sequence to clear the watchdog counter.

The watchdog counter clear protection trigger sequence must meet the following conditions.

- Write data to the Software watchdog trigger 1 register (SWDG_TRG1) after writing data to the Software watchdog trigger 0 register (SWDG_TRG0).
- Do not execute the watchdog counter clear protection trigger sequence when the LOCK bit in the Software watchdog configuration register (SWDG_CFG) is "0".

- The value to be written to the Software watchdog trigger 0 register (SWDG_TRG0) must match the Software watchdog trigger 0 configuration register (SWDG_TRG0CFG).
- The value to be written to the Software watchdog trigger 1 register (SWDG_TRG1) must match the Software watchdog trigger 1 configuration register (SWDG_TRG1CFG).
- The watchdog counter must already have reached the window lower limit value when the watchdog counter clear protection trigger sequence is completed.

Note:

- *For details on the watchdog counter clear protection trigger sequence, see Figure 3-2.*

Window Watchdog Function

The window watchdog function is a function for setting a range in which the watchdog counter value can be cleared with upper limit and lower limit values. The following shows the range in which the watchdog counter can be cleared for 2 set values.

(Window lower limit value) ≤ (watchdog counter) < (window upper limit value)

For example, suppose that a runaway state of a user program occurs and the watchdog counter clear protection trigger sequence is executed continuously. In this case, you can set a value other than 0x00000000 as a window lower limit value to detect the continuous watchdog counter clear as an abnormal operation.

The window setting for RUN is defined with the following 2 registers.

Software watchdog lower limit RUN current register (SWDG_RUNLLC)

Software watchdog upper limit RUN current register (SWDG_RUNULC)

The window setting for PSS is defined with the following 2 registers.

Software watchdog lower limit PSS current register (SWDG_PSSLLC)

Software watchdog upper limit PSS current register (SWDG_PSSULC)

Different Operation Settings for Each Device State (RUN and PSS)

The Software watchdog timer can make different operation settings for each device state (RUN and PSS). The operation setting is switched to the RUN or PSS operation setting according to the device state transition.

RUN operation setting

- When the LOCK bit and WDENRUN bit in the Software watchdog configuration register (SWDG_CFG) are set to "1", operation is performed by using the window setting for RUN.

- When the watchdog counter is not cleared in the range between the window upper limit value and lower limit value, or there is a violation in the watchdog counter clear protection trigger sequence, a watchdog error is detected. The watchdog reset request or watchdog interrupt request (NMI) is generated for this watchdog error. (For details on watchdog errors, see "7 Precautions for Using This Device Watchdog Error.")

PSS operation setting

- The window setting is switched to the window setting for PSS when the device state transitions from RUN to PSS. When the WDENPSS bit in the Software watchdog configuration register (SWDG_CFG) is set to "1", operation is performed by using the window setting for PSS.

- When the device state transitions from PSS to RUN, the enable bit is switched from the WDENPSS bit in the Software watchdog configuration register (SWDG_CFG) to the WDENRUN bit in the same register. However, the window setting is not switched immediately. Use of the window setting for PSS continues until the watchdog counter clear protection trigger sequence is executed.

Refer to Figure 3-4 and Figure 3-5 for the details on the watchdog operation in RUN and PSS, respectively.

Notes:

- *Certainly set the same value in upper limit registers and lower limit registers.*
- *Set the same value following two upper limit registers.*
  - Software watchdog lower limit PSS setting register (SWDG_PSSLLS)
  - Software watchdog lower limit RUN setting register (SWDG_RUNLLS)
- *Set the same value following two upper limit registers*
  - Software watchdog upper limit PSS setting register (SWDG_PSSULS)
  - Software watchdog upper limit RUN setting register (SWDG_RUNULS)
- *The limitation is needed, when device state change from RUN to PSS. Because the watchdog timer continues to use the window setting for PSS until the watchdog counter clear protection trigger sequence is performed. When WDENPSS bit is "0", the watchdog timer perform the same operation. So it needs same limitation.*

Figure 3-4 is shown about the software watchdog operation in RUN. Figure 3-5 is shown about the software watchdog operation in PSS.

Please refer to Figure 3-5.

Watchdog Register Write Protection Sequence

The Software watchdog timer monitors the watchdog register write protection sequence to write a setting value in the register.

The watchdog register write protection sequence must meet the following conditions.

- It writes a setting value in the register after writing data in the Software watchdog protection register (SWDG_PROT).
- It writes data in the Software watchdog protection register (SWDG_PROT) in privileged mode.
- The value written in the Software watchdog protection register (SWDG_PROT) must match the lock release key for register write protection 0xEDAC_CE55.
- It does not write a setting value in any register in another module after writing data in the Software watchdog protection register (SWDG_PROT).
- It does not write data in the Software watchdog protection register (SWDG_PROT) twice in a row.
- It writes a value in the register when the LOCK bit in the Software watchdog configuration register (SWDG_CFG) is "0" and the mode is privileged mode.

Note:

- *For details on the watchdog register write protection sequence, see Figure 3-1 Setting Procedure for Watchdog Register Write Protection Sequence.*

Table 3-3 Key Value for Register Programming

Key Value	Target Register	Purpose
0xEDAC_CE55	SWDG_PROT:KEY	Releasing protection lock for register write

Notes:

- *The watchdog register write protection sequence is unnecessary for the Software watchdog trigger 0/1 register (SWDG_TRG0 and SWDG_TRG1).*
- *Setting the LOCK bit in the Software watchdog configuration register (SWDG_CFG) to "1" prevents the following registers from being rewritten.*
  - Software watchdog interrupt configuration register (SWDG_INT)
  - Software watchdog trigger 0 configuration register (SWDG_TRG0CFG)
  - Software watchdog trigger 1 configuration register (SWDG_TRG1CFG)
  - Software watchdog reset delay counter register (SWDG_RSTDLY)
  - Software watchdog configuration register (SWDG_CFG)

Majority Circuits for Bits That Affect Important Functions

The Software watchdog timer has majority circuits for bits that affect important functions. The majority circuit consists of 3 FFs. When the bit in an FF bit is inverted due to noise or because of another factor, the correct value can be selected by majority decision among the 3 FFs.

The bits that are equipped with majority circuits are as follows.

- RSTEN bit (*1)
- LOCK bit (*2)
- WDENPSS bit (*2)
- WDENRUN bit (*2)

Notes:

- *The notation (*1) indicates a bit in the Software watchdog interrupt configuration register (SWDG_INT).*
- *The notation (*2) indicates a bit in the Software watchdog configuration register (SWDG_CFG).*

Generation of Watchdog Reset Request or Watchdog Interrupt Request (NMI)

The Software watchdog timer generates the watchdog reset request or watchdog interrupt request (NMI) when detecting a watchdog error. Use the watchdog interrupt request (NMI) as a test function. (For example, you can use the watchdog interrupt request (NMI) to set a break point at the beginning of an interrupt handler.)

Also, the Software watchdog timer can generate the prior warning interrupt request before the watchdog reset request (or watchdog interrupt request (NMI)). (For example, you can use the prior warning interrupt request so that the processor can save important data to the memory before the watchdog reset request is generated.)

The Software watchdog timer generates these requests by using the following procedure.

- When a watchdog error occurs, the IRQFLAG bit in the Software watchdog interrupt configuration register (SWDG_INT) is set. At this time, if the IRQEN bit in the Software watchdog interrupt configuration register (SWDG_INT) is "1", the prior warning interrupt request is generated.
- As many cycles as are set in the Software watchdog reset delay counter register (SWDG_RSTDLY) are inserted between the prior warning interrupt request and the watchdog reset interrupt request or watchdog interrupt request (NMI) as a delay time.
- Once the prior warning interrupt request is generated, the watchdog counter cannot be cleared with the watchdog counter clear protection trigger sequence. Also, if a new watchdog error occurs, it is ignored.
- After the elapse of the time corresponding to the number of cycles for delay time set in the Software watchdog reset delay counter register (SWDG_RSTDLY), the watchdog reset request or watchdog interrupt request (NMI) is generated. If the Software watchdog reset delay counter register (SWDG_RSTDLY) is 0x0000, no delay time occurs.

Notes:

- *When you generate the prior warning interrupt request, set the Software watchdog reset delay counter register (SWDG_RSTDLY) to a value other than 0x0000.*
- *The use of the watchdog interrupt request (NMI) is prohibited except when used as a test function. (This is prohibited because if such use is allowed, security will be compromised during the execution of an application.) Figure 3-6 shows the operation of the test function of the Software watchdog.*
- *Insertion of a number of cycles equal to that for the delay time set in the Software watchdog reset delay counter register (SWDG_RSTDLY) is valid only for 1 watchdog reset request or watchdog interrupt request (NMI). (It remains invalid until the next hard reset occurs.)*

Output of the Watchdog Counter Monitor Bit for MCU Output Pin

The Software watchdog timer outputs the watchdog counter monitor bit to the outside so that watchdog counter operation can be monitored from the outside. You can select any 1 bit in the watchdog counter (32 bits) by setting the OBSSEL[4:0] bits in the Software watchdog configuration register (SWDG_CFG).

Support of this function is different depends on product specification.

Watchdog Counter Stop in Debugging State of Processor

The watchdog counter stops when the processor enters the debugging state. When the processor returns from the debugging state, the watchdog counter resumes operation from the point where it had stopped.

Note:

- *For the definition of the debugging state of the processor, see the material from ARM(R) (CortexTM-R5 Revision:r1p2 Technical Reference Manual (ARM DDI 0460D)).*

Differences between Software Watchdog and Hardware Watchdog

The following Table 3-4 Differences between Software Watchdog and Hardware Watchdog shows the main differences between the software watchdog and hardware watchdog.

Table 3-4 Differences between Software Watchdog and Hardware Watchdog

Item	Software Watchdog	Hardware Watchdog
Watchdog Counter	32-bit up counter	Same as on left
Different Initializations between Soft Reset and Hard Reset	Supported	Same as on left
Trigger Sequence for Watchdog Counter Clear	Supported	Same as on left
Window Watchdog Function	Supported	Same as on left
Different Operation Settings for Each Device State (RUN and PSS)	Supported	Same as on left
Watchdog Counter Source Clock Selection	4 types - High-speed CR clock - Low-speed CR clock - Sub clock - Main clock	2 types - High-speed CR clock - Low-speed CR clock
Watchdog Register Write Protection Sequence	Supported (However, prevention of rewriting of the window setting by the LOCK bit is not supported.)	Same as on left (However, prevention of rewriting of the window setting by the LOCK bit is supported.)
Majority Circuits for Bits That Affect Important Functions	Supported (4 target bits)	Same as on left (2 target bits)
Reset Request or Interrupt Request (NMI) Generation	Supported	Same as on left
Prior Warning Interrupt Request Generation	Supported	Same as on left
Output of Watchdog Counter Monitor Bit for MCU Output Pin	Supported	Supported
Watchdog Counter Stop in the Debugging State of the Processor	Supported	Same as on left
Watchdog Error Conditions	5 types	Same as on left
Bus Error Response Conditions	8 types	Same as on left
Watchdog Timer Start	Register write by a user program	Hard reset clear
Register Setting Value Write	User program	BootROM program

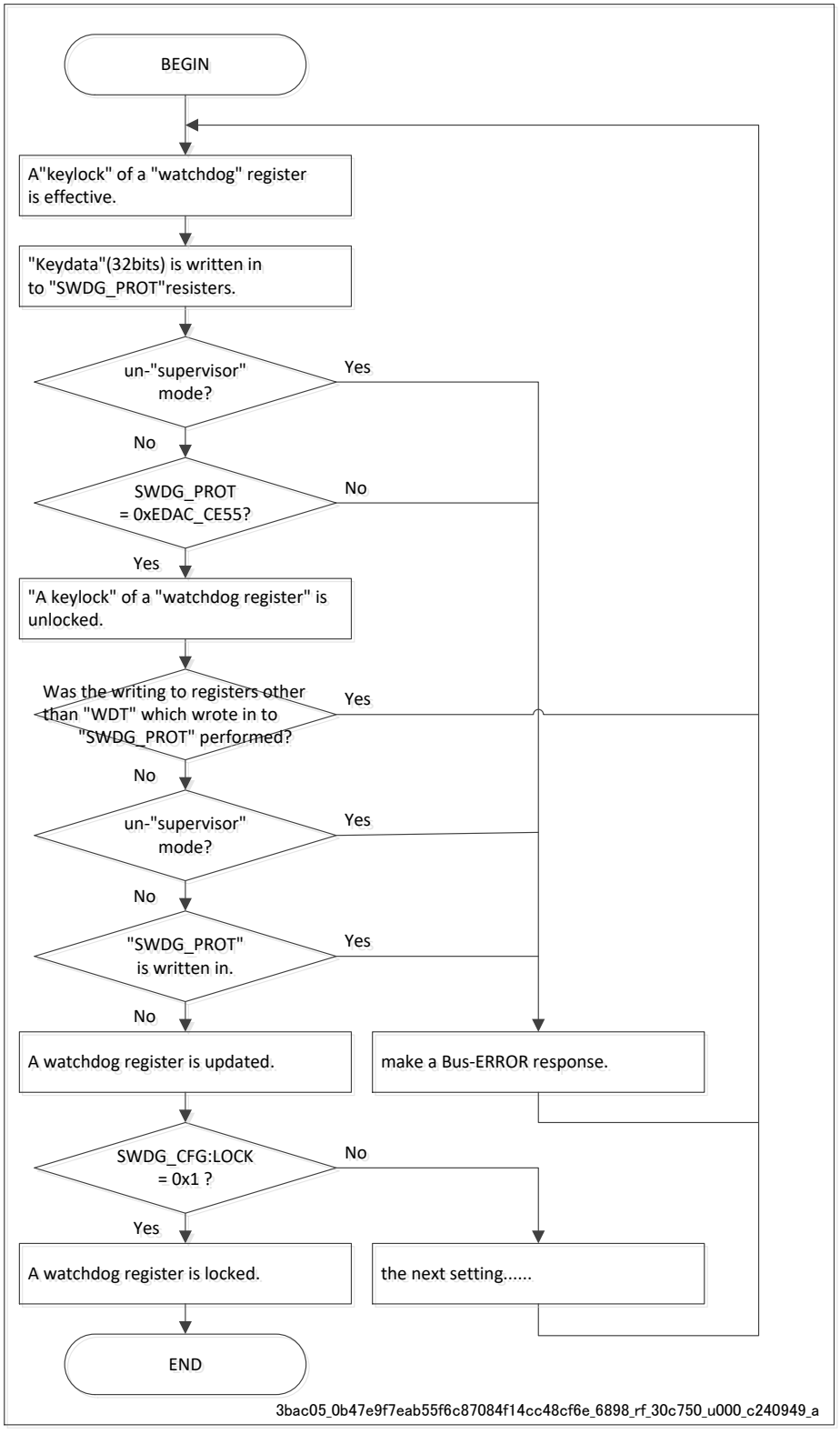
Item	Software Watchdog	Hardware Watchdog
Watchdog Counter Enable Bit Control	Register write by a user program	Constantly operation (Fixed hardware)

Note:

- *Unlike the case of the software watchdog timer, for the hardware watchdog timer, the BootROM program sets registers during the initial operation setting. (Write and read operation from the CPU can also be performed.)*

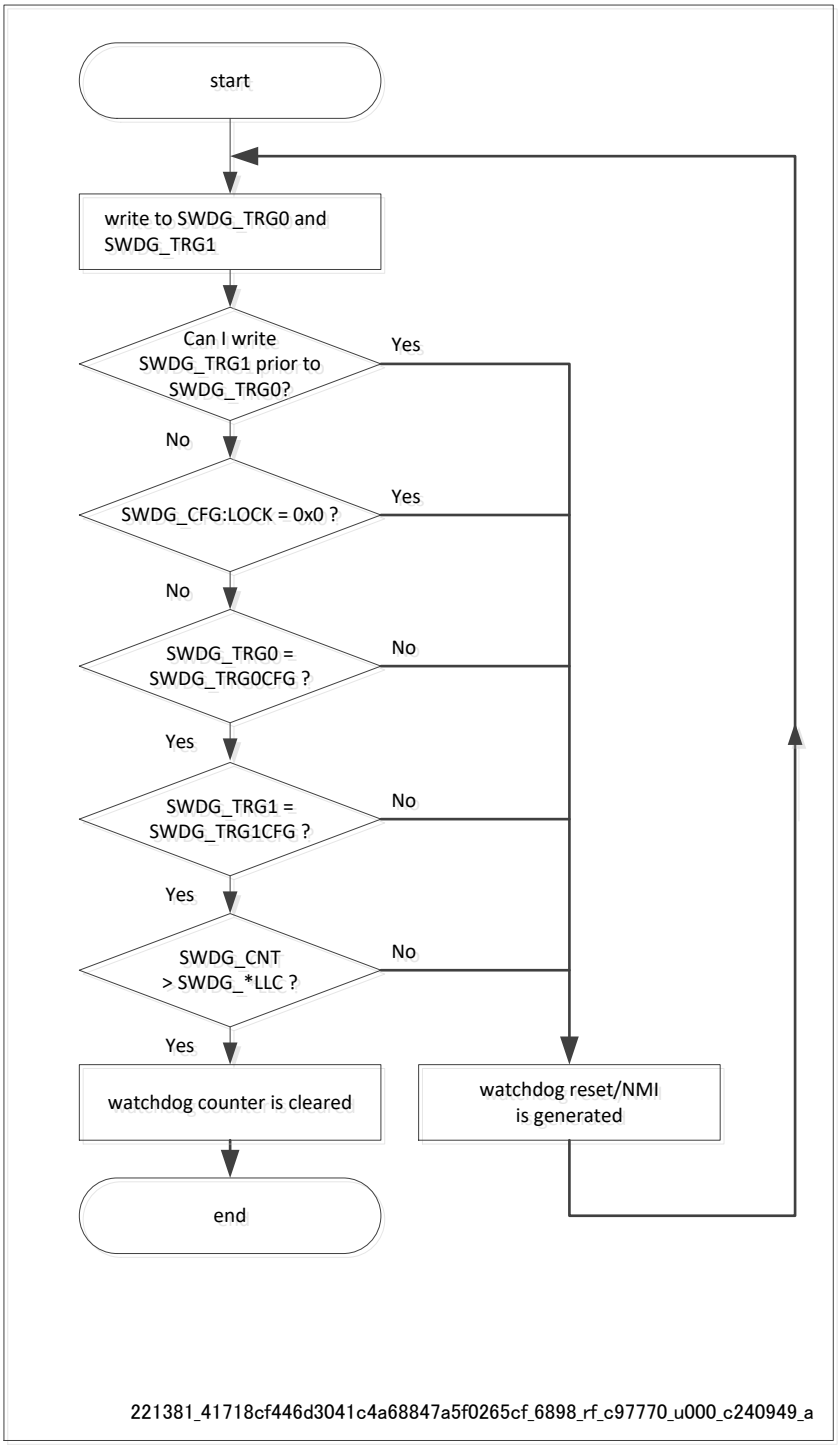
The following figure shows the watchdog register write protection sequence setting procedure.

Figure 3-1 Setting Procedure for Watchdog Register Write Protection Sequence



The following figure shows the watchdog counter clear protection trigger sequence setting procedure.

Figure 3-2 Setting Procedure for Watchdog Counter Clear Protection Trigger Sequence

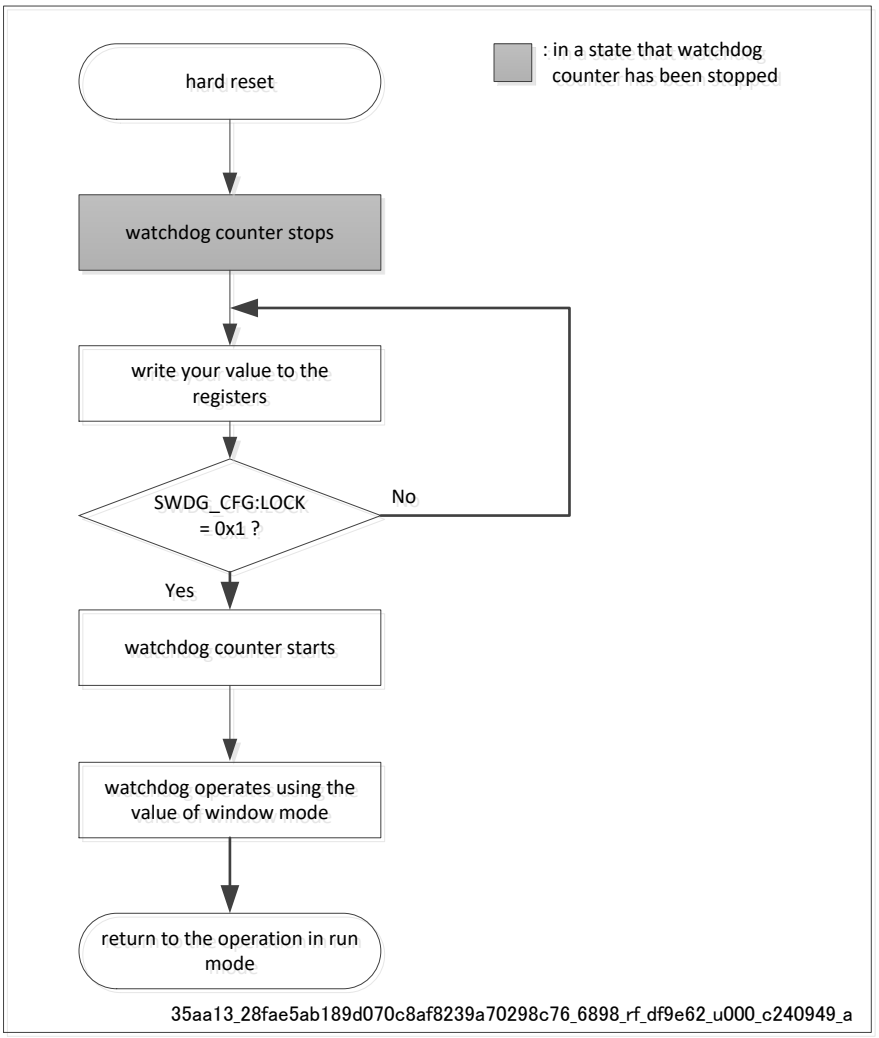


Note:

- *The notation * in Figure 3-2 Setting Procedure for Watchdog Counter Clear Protection Trigger Sequence is replaced with device state RUN or PSS.*

The following figure shows the Software watchdog timer startup procedure.

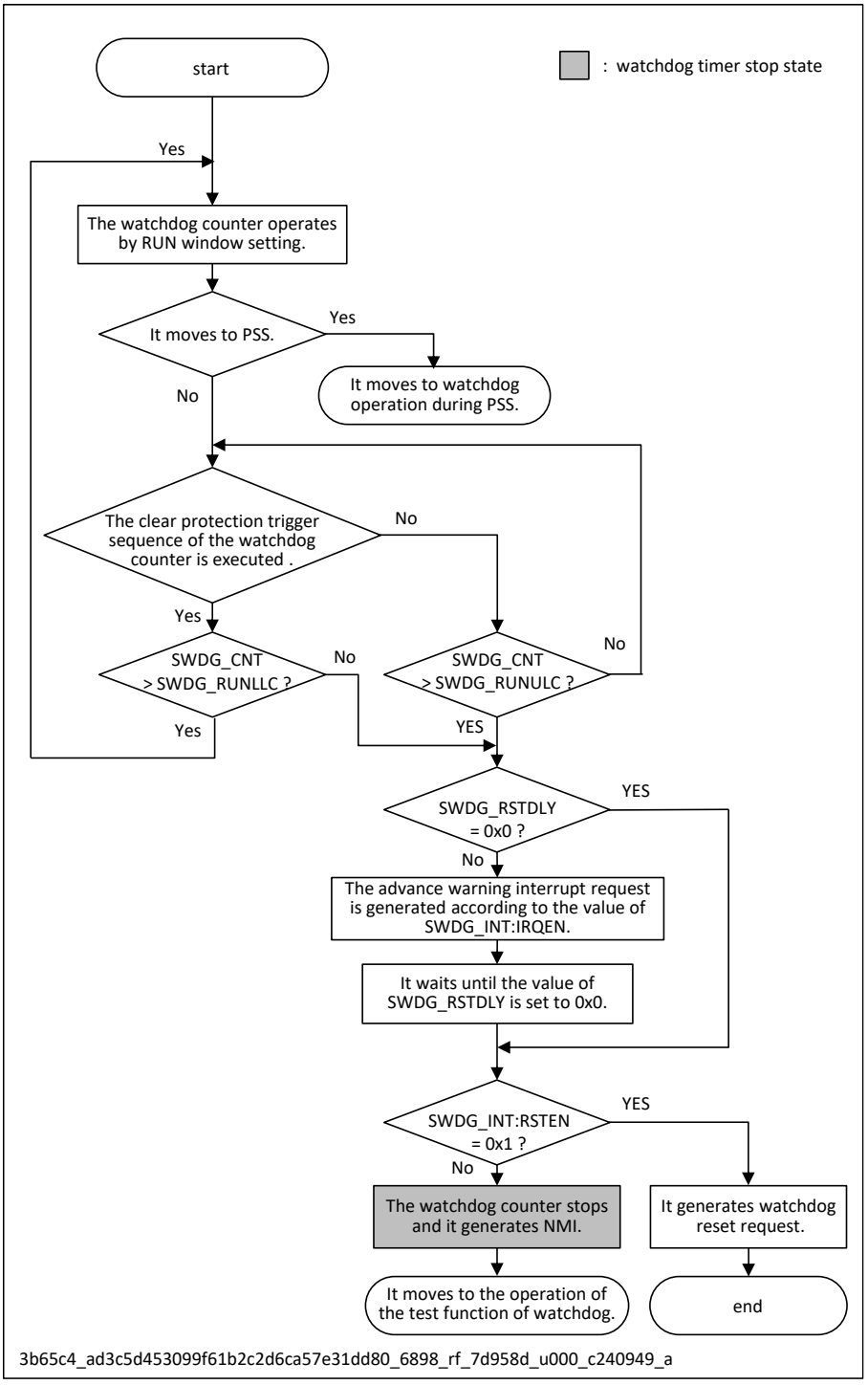
Figure 3-3 Startup of Software Watchdog Timer



1. The hard reset initializes the software watchdog timer. Then, immediately after the hard reset is cleared, the stop state is indicated.
2. You can change the register setting as long as the LOCK bit in the software watchdog configuration register (SWDG_CFG) is "0".
3. Setting "1" in the LOCK bit in the software watchdog configuration register (SWDG_CFG) causes the watchdog counter to start up-counting.

The following figure shows the Software watchdog operation in RUN.

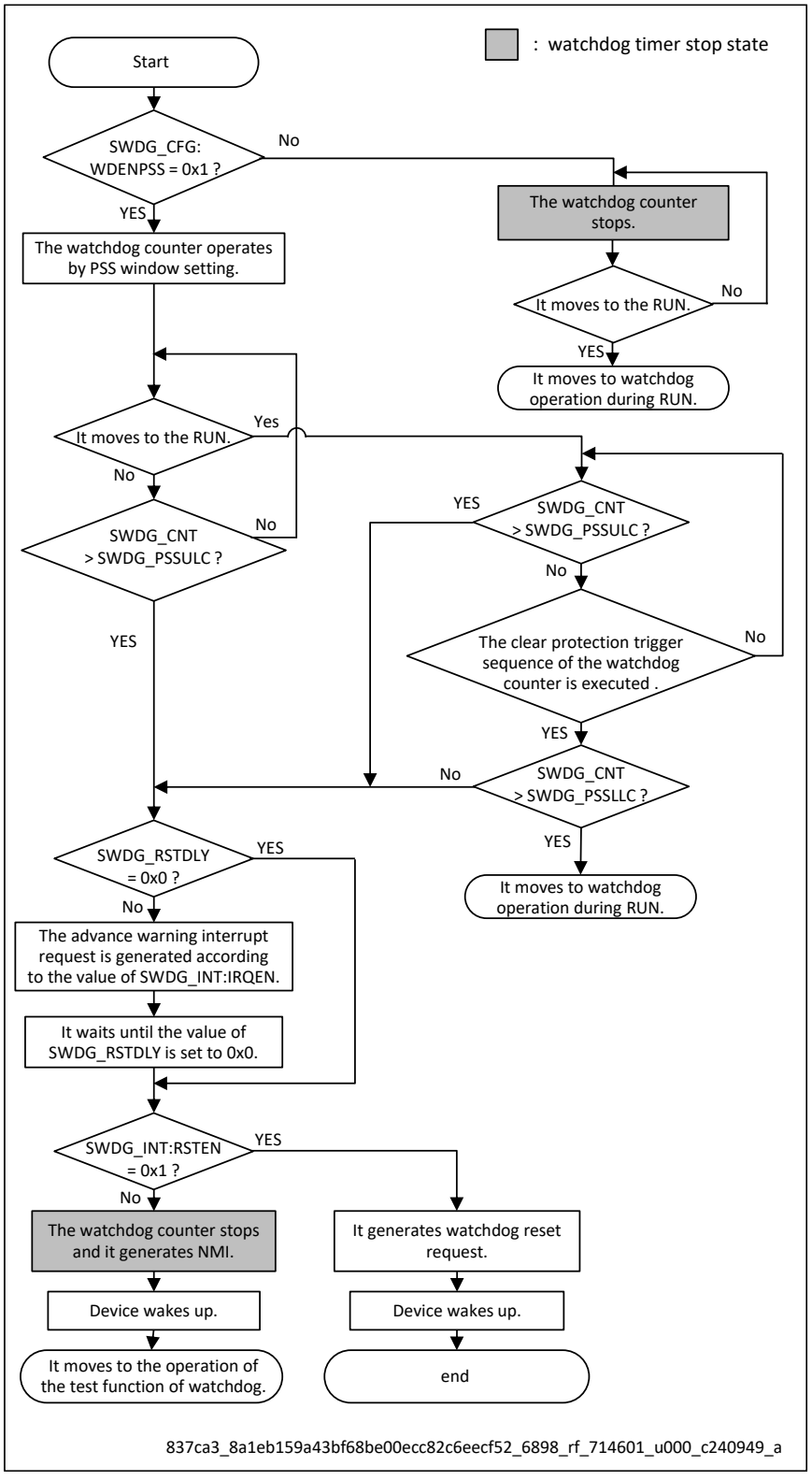
Figure 3-4 Software Watchdog Operation in RUN



1. The watchdog counter in RUN always operates.
2. When the device state transitions from RUN to PSS, the processing transitions to the operation of the watchdog in PSS. For details, see Figure 3-5.
3. During the operation of the watchdog counter, execute the watchdog counter clear protection trigger sequence within the range between the window upper limit value and lower limit value that have been specified beforehand. Do this to clear the watchdog counter periodically.
4. If a runaway state of the software occurs and prevents periodic clearing, the watchdog counter reaches the window upper limit value. Then, processing transitions to the flow that generates the watchdog reset request or watchdog interrupt request (NMI).
5. According to the value of the IRQEN bit in the Software watchdog interrupt configuration register (SWDG_INT), the prior warning interrupt request is generated. At the same time, as many cycles as are set for the delay time in the Software watchdog reset delay counter register (SWDG_RSTDLY) are inserted.
6. After the elapse of the time corresponding to the number of cycles for the delay time, the watchdog reset request or watchdog interrupt request (NMI) is generated according to the value of the RSTEN bit in the Software watchdog interrupt configuration register (SWDG_INT). When the watchdog interrupt request (NMI) is generated, the processing transitions to the operation of the test function of the watchdog. For details, see Figure 3-6.

The following figure shows the Software watchdog operation in PSS.

Figure 3-5 Software Watchdog Operation in PSS



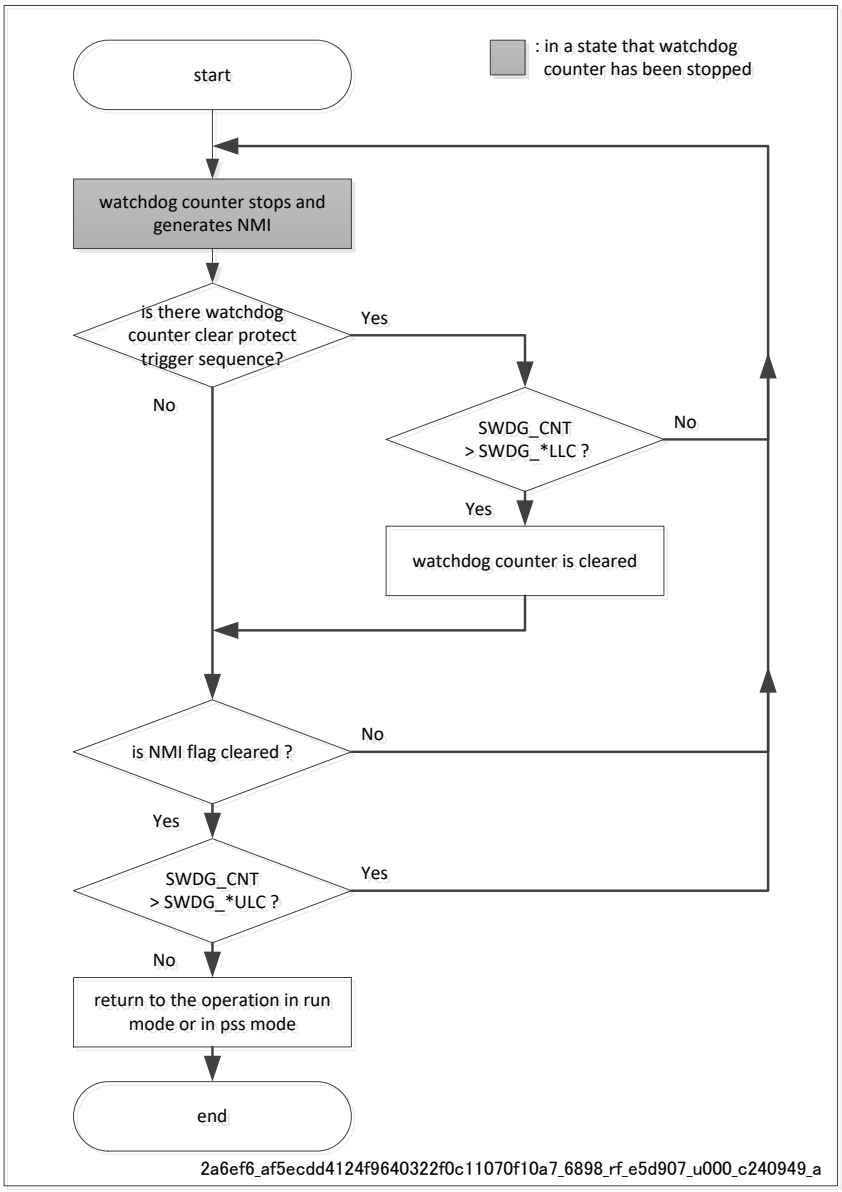
1. The watchdog counter in PSS operates or stops according to the value of the WDENPSS bit in the Software watchdog configuration register (SWDG_CFG).
2. When the device state transitions from PSS to RUN, the watchdog counter operation is started immediately. However, the PSS window setting continues to be used until the watchdog counter clear protection trigger sequence is executed.
3. During the operation of the watchdog counter, execute the watchdog counter clear protection trigger sequence within the range between the window upper limit value and lower limit value that have been specified beforehand. Continuously clearing the watchdog counter needs to be done periodically.
4. If a runaway state of the software occurs and prevents periodic clearing, the watchdog counter reaches the window upper limit value. Then, processing transitions to the flow that generates the watchdog reset request or watchdog interrupt request (NMI).
5. According to the value of the IRQEN bit in the Software watchdog interrupt configuration register (SWDG_INT), the prior warning interrupt request is generated. At the same time, as many cycles as are set for the delay time in the Software watchdog reset delay counter register (SWDG_RSTDLY) are inserted.
6. After the elapse of the time corresponding to the number of cycles for the delay time, the watchdog reset request or watchdog interrupt request (NMI) is generated according to the value of the RSTEN bit in the Software watchdog interrupt configuration register (SWDG_INT). When the watchdog interrupt request (NMI) is generated, the processing transitions to the operation of the test function of the watchdog. For details, see Figure 3-6.

Note:

- *If the watchdog counter clear protection trigger sequence is executed during operation with the PSS window setting, this means that the device state has already returned from PSS to RUN.*

The following figure shows the operation of the test function of the Software watchdog.

Figure 3-6 Operation of Software Watchdog Test Function



1. The watchdog counter is in the stop state.
2. If the watchdog counter clear protection trigger sequence is executed in the stop state of the watchdog counter and the window lower limit value has already been reached at this time, the watchdog counter is cleared. (If this occurs outside the counter range, no special change occurs.)
3. When the NMIFLAG bit in the Software watchdog interrupt configuration register (SWDG_INT) is cleared, the watchdog interrupt request (NMI) is cleared.
4. When the watchdog counter has not reached the window upper limit value, processing returns to operation in RUN or PSS. When it has already reached the window upper limit value, no special change occurs.

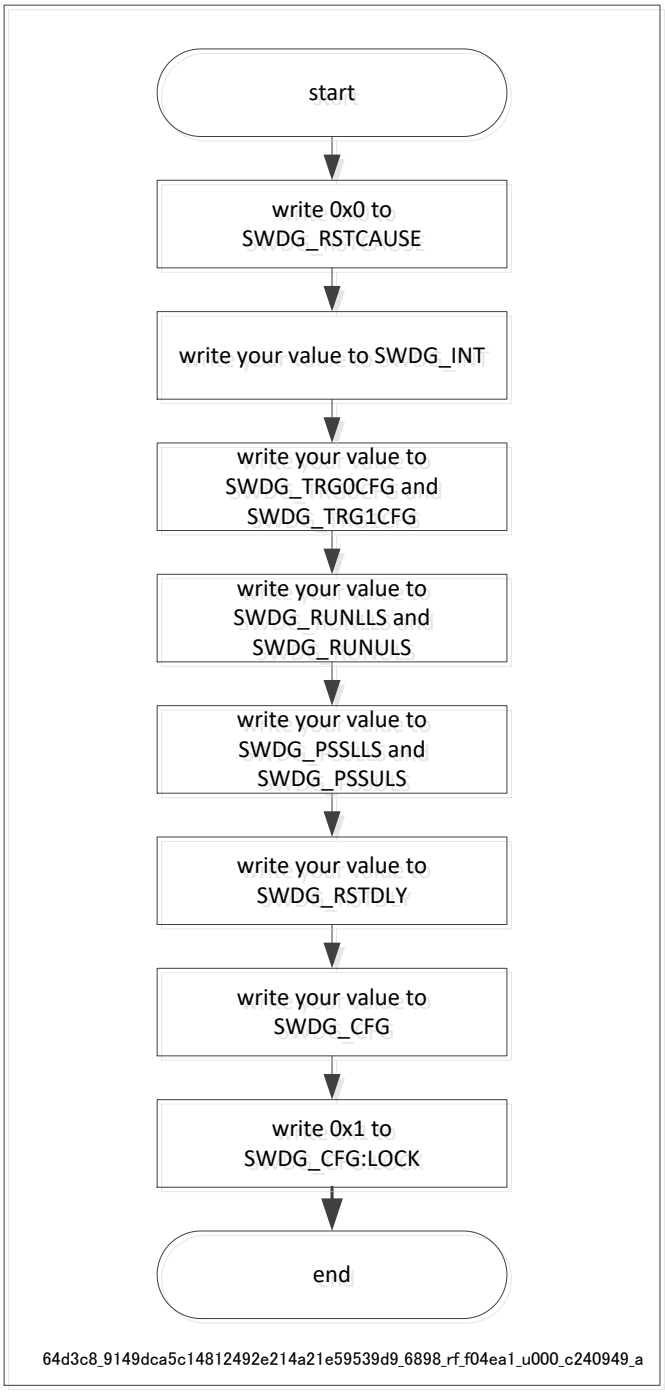
Note:

The notation * in Figure 3-6 means "RUN" or "PSS".

4. Setting Procedure Example

This section explains an example of a procedure for setting the Software watchdog timer.

Figure 4-1 Example of Software Watchdog Timer Register Setting Procedure



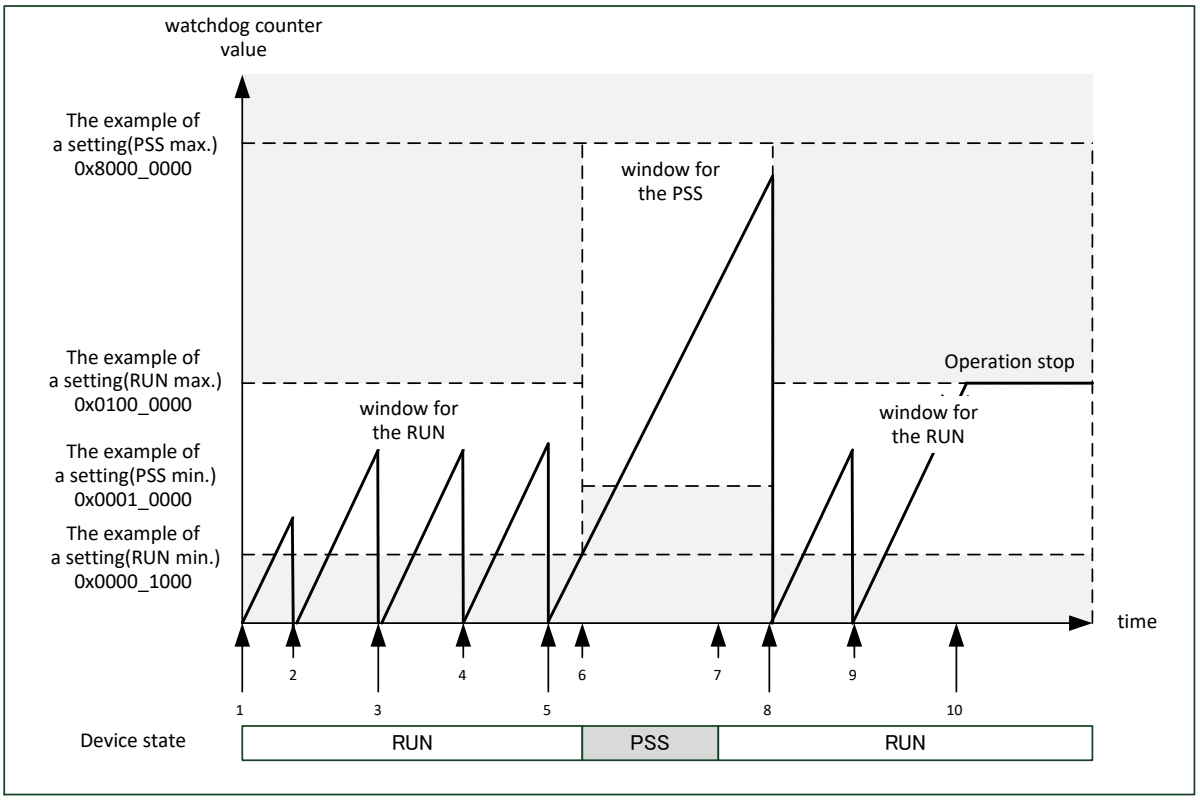
Note:

- *In Figure 4-1, the setting order can be changed except for the step of setting the LOCK bit in the Software watchdog configuration register (SWDG_CFG) to "1".*

5. Operation Examples

This section describes examples of operating the Software watchdog timer.

Figure 5-1 Example of Operating Software Watchdog Timer (When Operation in PSS Is Enabled)

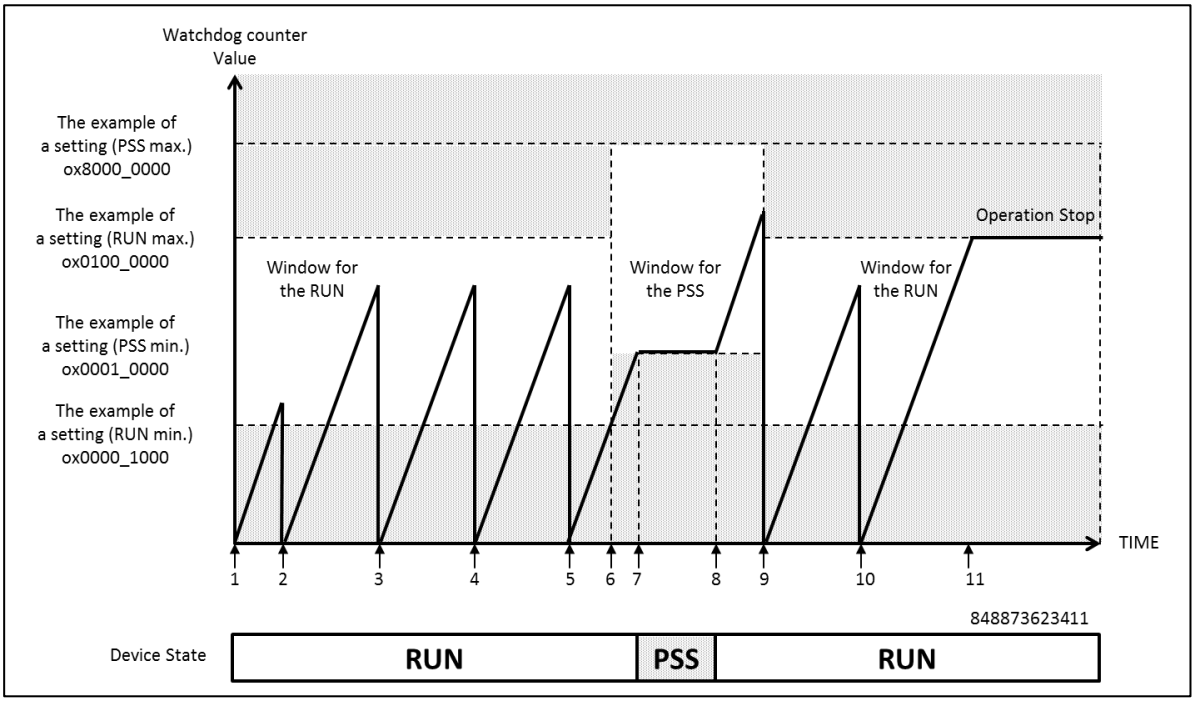


1. Before starting the software watchdog timer, the setting values of the registers are written.
2. The LOCK bit in the software watchdog configuration register (SWDG_CFG) is set to "1" to start the software watchdog timer. At this time, it starts up-counting from 0x00000000.
3. The watchdog counter clear protection trigger sequence is executed, and the watchdog counter is cleared.
4. The watchdog counter clear protection trigger sequence is executed, and the watchdog counter is cleared.
5. The watchdog counter clear protection trigger sequence is executed, and the watchdog counter is cleared.
6. The device state transitions from RUN to PSS. At this time, switching to the PSS window setting occurs.
7. The device state returns from PSS to RUN. At this time, the switching to the RUN window setting does not occur immediately.
8. The watchdog counter clear protection trigger sequence is executed, and the watchdog counter is cleared. At this time, the switching to the RUN window setting occurs.
9. The watchdog counter clear protection trigger sequence is executed, and the watchdog counter is cleared.
10. When the watchdog counter reaches the window upper limit value, the watchdog reset request or watchdog interrupt request (NMI) is generated.

Notes:

- *The window setting is not immediately switched to the window setting for RUN when the device state is returned from PSS to RUN.*
- *Before the transition from RUN to PSS, be sure to clear the watchdog counter.*

Figure 5-2 Example of Operating Software Watchdog Timer (When Operation in PSS Is Disabled)



1. Before starting the software watchdog timer, the setting values of the registers are written.
2. The LOCK bit in the software watchdog configuration register (SWDG_CFG) is set to "1" to start the software watchdog timer. At this time, it starts up-counting from 0x00000000.
3. The watchdog counter clear protection trigger sequence is executed, and the watchdog counter is cleared.
4. The watchdog counter clear protection trigger sequence is executed, and the watchdog counter is cleared.
5. The watchdog counter clear protection trigger sequence is executed, and the watchdog counter is cleared.
6. Switching to the PSS window setting occurs.
7. The device state transitions from RUN to PSS. At this time, the Software watchdog timer stops operation.
8. The device state returns from PSS to RUN. At this time, the Software watchdog timer immediately starts operation. The switching to the RUN window setting does not occur immediately.
9. The watchdog counter clear protection trigger sequence is executed, and the watchdog counter is cleared. At this time, the switching to the RUN window setting occurs.
10. The watchdog counter clear protection trigger sequence is executed, and the watchdog counter is cleared.
11. When the watchdog counter reaches the window upper limit value, the watchdog reset request or watchdog interrupt request (NMI) is generated.

Notes:

- *The window setting is not immediately switched to the window setting for RUN when the device state is returned from PSS to RUN.*
- *After the device state returned from PSS to RUN, the watchdog counter is cleared. At this time, the switching to the RUN window setting occurs.*

- *Certainly set the same value in upper limit registers and lower limit registers.*
- *Set the same value following two lower limit registers.*
  - Software watchdog lower limit PSS setting register (SWDG_PSSLLS)
  - Software watchdog lower limit RUN setting register (SWDG_RUNLLS)
- *Set the same value following two upper limit registers*
  - Software watchdog upper limit PSS setting register (SWDG_PSSULS)
  - Software watchdog upper limit RUN setting register (SWDG_RUNULS)
- *The limitation is needed, when device state change from PSS to RUN. Because the window setting don't change immediately. When device state change from RUN to PSS, clear the watchdog counter.*

6. Register List

This section explains the registers of the Software watchdog timer.

Table 6-1 List of Software Watchdog Timer Registers

Abbreviated Register Name	Register Name	Reference
SWDG_PROT	Software watchdog protection register	6.1
SWDG_CNT	Software watchdog counter register	6.2
SWDG_RSTCAUSE	Software watchdog reset source register	6.3
SWDG_TRG0	Software watchdog trigger 0 register	6.4
SWDG_TRG1	Software watchdog trigger 1 register	6.5
SWDG_INT	Software watchdog interrupt configuration register	6.6
SWDG_INTCLR	Software watchdog interrupt clear register	6.7
SWDG_TRG0CFG	Software watchdog trigger 0 configuration register	6.8
SWDG_TRG1CFG	Software watchdog trigger 1 configuration register	6.9
SWDG_RUNLLS	Software watchdog lower limit RUN setting register	6.10
SWDG_RUNULS	Software watchdog upper limit RUN setting register	6.11
SWDG_PSSLLS	Software watchdog lower limit PSS setting register	6.12
SWDG_PSSULS	Software watchdog upper limit PSS setting register	6.13
SWDG_RSTDLY	Software watchdog reset delay counter register	6.14
SWDG_CFG	Software watchdog configuration register	6.15
SWDG_RUNLLC	Software watchdog lower limit RUN current register	6.16
SWDG_RUNULC	Software watchdog upper limit RUN current register	6.17
SWDG_PSSLLC	Software watchdog lower limit PSS current register	6.18
SWDG_PSSULC	Software watchdog upper limit PSS current register	6.19

Note:

- *The notation n indicates the CPU number of a monitored (0). Since the software watchdog timer is assigned to each of the monitored CPUs, for each type of these registers, there are as many registers as there are CPUs.*

Table 6-2 Register Memory Layout of Software Watchdog Timer

Offset	Register Name			
	+3	+2	+1	+0
0x0000_0000	SWDG_PROT 00000000_00000000_00000000_00000000			
0x0000_0004	-	-	-	-
0x0000_0008	SWDG_CNT 00000000_00000000_00000000_00000000			
0x0000_000C	SWDG_RSTCAUSE 00000000_00000000_00000000_000XXXXX			
0x0000_0010	SWDG_TRG0 00000000_00000000_00000000_00000000			
0x0000_0014	-	-	-	-
0x0000_0018	SWDG_TRG1 00000000_00000000_00000000_00000000			

Offset	Register Name			
	+3	+2	+1	+0
0x0000_001C	-	-	-	-
0x0000_0020	SWDG_INT 00000000_00000010_00000000_00000000			
0x0000_0024	SWDG_INTCLR 00000000_00000000_00000000_00000000			
0x0000_0028	-	-	-	-
0x0000_002C	SWDG_TRG0CFG 00000000_00000000_00000000_00000000			
0x0000_0030	SWDG_TRG1CFG 00000000_00000000_00000000_00000000			
0x0000_0034	SWDG_RUNLLS 00000000_00000000_00000000_00000000			
0x0000_0038	SWDG_RUNULS 00000001_00000000_00000000_00000000			
0x0000_003C	SWDG_PSSLLS 00000000_00000000_00000000_00000000			
0x0000_0040	SWDG_PSSULS 10000000_00000000_00000000_00000000			
0x0000_0044	SWDG_RSTDLY 00000000_00000000_00000000_00000000			
0x0000_0048	SWDG_CFG 00000000_00000000_00000000_00000011			
0x0000_004C	SWDG_RUNLLC 00000000_00000000_00000000_00000000			
0x0000_0050	SWDG_RUNULC 00000001_00000000_00000000_00000000			
0x0000_0054	SWDG_PSSLLC 00000000_00000000_00000000_00000000			
0x0000_0058	SWDG_PSSULC 10000000_00000000_00000000_00000000			

Notes:

- *"X" indicates an undefined value.*

6.1. Software Watchdog Protection Register (SWDG_PROT)

This register is used to execute the watchdog register write protection sequence. Write the correct key (0xEDAC_CE55) to this register before performing write access to each register. (However, the Software watchdog trigger 0/1 register (SWDG_TRG0 and SWDG_TRG1) are excluded.) Writing the correct key releases the write protection lock for subsequent register access. Write access to each register enables write protection lock for the registers. (Read access to each register does not affect the lock.) For write access to this register, you must write 32-bit data.

Bit	31 … 0
Field	KEY[31:0]
R/W Attribute	R, W
Protection Attribute	WP
Initial Value	00000000_00000000_00000000_00000000

[bit31:0] KEY[31:0]: Protection Bits

KEY[31:0]	Description
During write operation	When 0xEDAC_CE55 is written: Protection lock for register write is released. When a value other than 0xEDAC_CE55 is written: Protection lock for register write is not released.
During read operation	When 0xFFFF_FFFF is read: Protection lock for register write has been released. When 0x0000_0000 is read: Protection lock for register write has been enabled.

Notes:

- *An AHB transfer error response is generated when write access to this register is performed under any of the conditions below. (This error response invokes the CPU exception handler.)*
  - A wrong key is written to this register.
  - Non-32-bit data is written to this register.
  - Data is written to this register twice in a row.

- *For details on the watchdog register write protection sequence, see Figure 3-1.*
- *A protect key will be locked again by writing to the address where is in the same group area(SYSC1 Group), however protect key will not be locked by writing to no protect target register (SWDG_TRG0/1) in Software Watchdog.*

6.2. Software Watchdog Counter Register (SWDG_CNT)

This register indicates the current up count value of the watchdog counter.

Bit	31	0
Field	WDGCNT[31:0]	
R/W Attribute	R, WX	
Protection Attribute	-	
Initial Value	00000000_00000000_00000000_00000000	

[bit31:0] WDGCNT[31:0]: Watchdog Counter Bits

WDGCNT[31:0]	Description
During write operation	Invalid
During read operation	Returns the current watchdog counter value. The watchdog counter value is sampled by using the bus clock.

Note:

- *An AHB transfer error response is generated where there is write access to this register.*

6.3. Software Watchdog Reset Factor Register (SWDG_RSTCAUSE)

When performing write access to this register, follow the watchdog register write protection sequence. This register is the status register that indicates the factor of the watchdog reset request or watchdog interrupt request (NMI). This register is not initialized by reset. After a watchdog reset, the value is maintained. To check the factor of the watchdog reset request, read this register.

Bit	31	8
Field	Reserved	
R/W Attribute	R0, WX	
Protection Attribute	WPS	
Initial Value	00000000_00000000_00000000	

Bit	7 5	4	3	2	1	0
Field	Reserved	RST CAUSE4	RST CAUSE3	RST CAUSE2	RST CAUSE1	RST CAUSE0
R/W Attribute	R0, WX	R, W	R, W	R, W	R, W	R, W
Protection Attribute	WPS					
Initial Value	000	X	X	X	X	X

[bit31:5] Reserved: Reserved Bits

[bit4] RSTCAUSE4: Reset Factor Bit 4

If the watchdog counter clear protection trigger sequence is executed when the LOCK bit in the Software watchdog configuration register (SWDG_CFG) is "0", this bit is set to "1".

RSTCAUSE4	Description
During write operation	When "0" is written: This bit cleared When "1" is written: Invalid
During read operation	When "0" is read: Indicating no detection of a reset/NMI factor When "1" is read: Indicating detection of a reset/NMI factor

[bit3] RSTCAUSE3: Reset Factor Bit 3

If the watchdog counter clear protection trigger sequence is executed before the watchdog counter reaches the window lower limit value, this bit is set to "1".

RSTCAUSE3	Description
During write operation	When "0" is written: This bit cleared When "1" is written: Invalid
During read operation	When "0" is read: Indicating no detection of a reset/NMI factor When "1" is read: Indicating detection of a reset/NMI factor

[bit2] RSTCAUSE2: Reset Factor Bit 2

When the watchdog counter reaches the window upper limit value, this bit is set to "1".

RSTCAUSE2	Description
During write operation	When "0" is written: This bit cleared When "1" is written: Invalid
During read operation	When "0" is read: Indicating no detection of a reset/NMI factor When "1" is read: Indicating detection of a reset/NMI factor

When the prior warning interrupt request is used, RSTCAUSE2 cannot be cleared during the period from the occurrence of a prior warning interrupt until a watchdog reset or watchdog interrupt (NMI).

[bit1] RSTCAUSE1: Reset Factor Bit 1

When there is a violation of the watchdog counter clear protection trigger sequence, this bit is set to "1". For details, see Figure 3-2.

.

RSTCAUSE1	Description
During write operation	When "0" is written: This bit cleared When "1" is written: Invalid
During read operation	When "0" is read: Indicating no detection of a reset/NMI factor When "1" is read: Indicating detection of a reset/NMI factor

[bit0] RSTCAUSE0: Reset Factor Bit 0

If the value written in the Software watchdog trigger 0/1 register (SWDG_TRG0 or SWDG_TRG1) does not match a proper value, this bit is set to "1".

RSTCAUSE0	Description
During write operation	When "0" is written: This bit cleared When "1" is written: Invalid
During read operation	When "0" is read: Indicating no detection of a reset/NMI factor When "1" is read: Indicating detection of a reset/NMI factor

Notes:

- *When using the Software watchdog timer, be sure to clear this register.*
- *The value in this register becomes valid when the watchdog reset request is generated.*

6.4. Software Watchdog Trigger 0 Register (SWDG_TRG0)

When performing write access to this register, you do not need to follow the watchdog register write protection sequence. This register is used to execute the watchdog counter clear protection trigger sequence. Write the value defined in the Software watchdog trigger 0 configuration register (SWDG_TRG0CFG) to this register. When a value other than the one in the Software watchdog trigger 0 configuration register (SWDG_TRG0CFG) is written, the watchdog reset request or watchdog interrupt request (NMI) is generated.

Bit	31	8
Field	Reserved	
R/W Attribute	R0, WX	
Protection Attribute	-	
Initial Value	00000000_00000000_00000000	

Bit	7	6	5	4	3	2	1	0
Field	WDGTRG0[7:0]							
R/W Attribute	R0, W							
Protection Attribute	-							
Initial Value	00000000							

[bit31:8] Reserved: Reserved Bits

[bit7:0] WDGTRG0[7:0]: Watchdog Trigger 0 Bits

These bits are used to execute the watchdog counter clear protection trigger sequence to clear the watchdog counter.

WDGTRG0[7:0]	Description
During write operation	When the SWDG_TRG0CFG value is written: 1 condition for executing the watchdog counter clear protection trigger sequence is met. When a value other than the SWDG_TRG0CFG value is written: A watchdog error is generated.
During read operation	Reads 0b00000000.

Note:

- *Figure 3-2 shows the flow of the watchdog counter clear protection trigger sequence.*

6.5. Software Watchdog Trigger 1 Register (SWDG_TRG1)

When performing write access to this register, you do not need to follow the watchdog register write protection sequence. This register is used to execute the watchdog counter clear protection trigger sequence. Write the value defined in the Software watchdog trigger 1 configuration register (SWDG_TRG1CFG) to this register. When a value other than the one in the Software watchdog trigger 1 configuration register (SWDG_TRG1CFG) is written, the watchdog reset request or watchdog interrupt request (NMI) is generated.

Bit	31	8
Field	Reserved	
R/W Attribute	R0, WX	
Protection Attribute	-	
Initial Value	00000000_00000000_00000000	

Bit	7	6	5	4	3	2	1	0
Field	WDGTRG1[7:0]							
R/W Attribute	R0, W							
Protection Attribute	-							
Initial Value	00000000							

[bit31:8] Reserved: Reserved Bits

[bit7:0] WDGTRG1[7:0]: Watchdog Trigger 1 Bits

These bits are used to execute the watchdog counter clear protection trigger sequence to clear the watchdog counter.

WDGTRG1[7:0]	Description
During write operation	When the SWDG_TRG1CFG value is written: 1 condition for executing the watchdog counter clear protection trigger sequence is met. When a value other than the SWDG_TRG1CFG value is written: A watchdog error is generated.
During read operation	Reads 0b00000000.

Note:

- *Figure 3-2 shows the flow of the watchdog counter clear protection trigger sequence.*

6.6. Software Watchdog Interrupt Configuration Register (SWDG_INT)

When performing write access to this register, follow the watchdog register write protection sequence. This register is used to make settings related to the watchdog reset request (NMI) and prior warning interrupt request. This register also includes interrupt status flags.

Bit	31	18	17	16
Field	Reserved		RSTEN	IRQEN
R/W Attribute	R0, WX		R/W	R/W
Protection Attribute	WPS			
Initial Value	000000_00000000		1	0

Bit	15	2	1	0
Field	Reserved		NMI FLAG	IRQ FLAG
R/W Attribute	R0, WX		R, WX	R, WX
Protection Attribute	WPS			
Initial Value	000000_00000000		0	0

[bit31:18] Reserved: Reserved Bits

[bit17] RSTEN: Reset/NMI Enable Bit

This bit is used to generate either the watchdog reset request or watchdog interrupt request (NMI) in response to a watchdog error. Prohibiting setting this bit to "0" except for the purpose of using it as a test function. (If this bit is set to "0" during application execution, security is compromised.) This bit is equipped with a majority circuit with 3 FFs as a measure against a bit invert caused by the effects of noise or another factor.

RSTEN	Description
During write operation	When "0" is written: The watchdog interrupt request (NMI) is generated. When "1" is written: The watchdog reset request is generated.
During read operation	Set value read

[bit16] IRQEN: Prior warning Interrupt Enable Bit

This bit is used to enable generation of the prior warning interrupt request.

IRQEN	Description
During write operation	When "0" is written: The prior warning interrupt request is not generated. When "1" is written: The prior warning interrupt request is generated.
During read operation	Set value read

[bit15:2] Reserved: Reserved Bits

[bit1] NMIFLAG: NMI Flag

This bit is set by a watchdog error when the RSTEN bit in the Software watchdog interrupt configuration register (SWDG_INT) is "0". When the RSTEN bit in the Software watchdog interrupt clear register (SWDG_INT) is "1", the watchdog reset request is generated. You can clear this bit by writing "1" to the NMICLR bit in the Software watchdog interrupt configuration register (SWDG_INTCLR).

NMIFLAG	Description
During write operation	Invalid
During read operation	When "0" is read: No detection of a watchdog error (NMI) is indicated. When "1" is read: Detection of a watchdog error (NMI) is indicated.

[bit0] IRQFLAG: IRQ Flag

This bit is set by a watchdog error. The prior warning interrupt is generated when the IRQEN bit in the Software watchdog interrupt clear register (SWDG_INT) is "1". You can clear this bit by writing "1" to the IRQCLR bit in the Software watchdog interrupt configuration register (SWDG_INTCLR).

IRQFLAG	Description
During write operation	Invalid
During read operation	When "0" is read: No detection of a watchdog error (IRQ) is indicated. When "1" is read: Detection of a watchdog error (IRQ) is indicated.

Notes:

- *For details on the watchdog interrupt request (NMI), see "3 Explanation of Operation Generation of watchdog reset request or watchdog interrupt request (NMI)."*
- *For details on watchdog errors, see "7 Precautions for Using This Device Watchdog Error."*

6.7. Software Watchdog Interrupt Clear Register (SWDG_INTCLR)

When performing write access to this register, follow the watchdog register write protection sequence. This register is used to clear the NMI flag and IRQ flag in the Software watchdog interrupt configuration register (SWDG_INT). You can write data to this register even after the LOCK bit in the Software watchdog configuration register (SWDG_CFG) is set.

Bit	31	16
Field	Reserved	
R/W Attribute	R0, WX	
Protection Attribute	WPS	
Initial Value	00000000_00000000	

Bit	15	4	3	2	1	0
Field	Reserved				NMI CLR	IRQ CLR
R/W Attribute	R0, WX				R0, W	R0, W
Protection Attribute	WPS					
Initial Value	000000_00000000				0	0

[bit31:2] Reserved: Reserved Bits

[bit1] NMICLR: NMI Clear Bit

This bit is used to clear the NMIFLAG bit in the Software watchdog interrupt configuration register (SWDG_INT).

NMICLR	Description
During write operation	When "0" is written: Invalid When "1" is written: The NMI flag is cleared.
During read operation	"0" is read.

You must clear the NMI after waiting for watchdog counter clear (0x0000_0000). Check the watchdog counter value with SWDG_CNT.

[bit0] IRQCLR: Prior warning Interrupt Clear Bit

This bit is used to clear the IRQFLAG bit in the Software watchdog interrupt configuration register (SWDG_INT).

IRQCLR	Description
During write operation	When "0" is written: Invalid When "1" is written: The IRQ flag is cleared.
During read operation	"0" is read.

6.8. Software Watchdog Trigger 0 Configuration Register (SWDG_TRG0CFG)

When performing write access to this register, follow the watchdog register write protection sequence. This register is used to define a valid value to be written to the Software watchdog trigger 0 register (SWDG_TRG0) when the watchdog counter clear protection trigger sequence is performed.

Bit	31	8
Field	Reserved	
R/W Attribute	R0, WX	
Protection Attribute	WPS	
Initial Value	00000000_00000000_00000000	

Bit	7	6	5	4	3	2	1	0
Field	WDGTRG0CFG[7:0]							
R/W Attribute	R/W							
Protection Attribute	WPS							
Initial Value	00000000							

[bit31:8] Reserved: Reserved Bits

[bit7:0] WDGTRG0CFG[7:0]: Watchdog Trigger 0 Configuration Bits

These bits are used to define a value to be written to the Software watchdog trigger 0 register (SWDG_TRG0) for execution of the watchdog counter clear protection trigger sequence.

WDGTRG0CFG[7:0]	Description
During write operation	Set value written
During read operation	Set value read

6.9. Software Watchdog Trigger 1 Configuration Register (SWDG_TRG1CFG)

When performing write access to this register, follow the watchdog register write protection sequence. This register is used to define a valid value to be written to the Software watchdog trigger 1 register (SWDG_TRG1) when the watchdog counter clear protection trigger sequence is performed.

Bit	31 ... 8
Field	Reserved
R/W Attribute	R0, WX
Protection Attribute	WPS
Initial Value	00000000_00000000_00000000

Bit	7	6	5	4	3	2	1	0
Field	WDGTRG1CFG[7:0]							
R/W Attribute	R/W							
Protection Attribute	WPS							
Initial Value	00000000							

[bit31:8] Reserved: Reserved Bits

[bit7:0] WDGTRG1CFG[7:0]: Watchdog Trigger 1 Configuration Bits

These bits are used to define a value to be written to the Software watchdog trigger 1 register (SWDG_TRG1) for execution of the watchdog counter clear protection trigger sequence.

WDGTRG1CFG[7:0]	Description
During write operation	Set value written
During read operation	Set value read

6.10. Software Watchdog Lower Limit RUN Setting Register (SWDG_RUNLLS)

When performing write access to this register, follow the watchdog register write protection sequence. A setting value of the window lower limit value for RUN is written to this register. However, this register value is different from the window lower limit value for RUN that is actually used. For details on how to reflect this register value in the window lower limit value for RUN actually used, see "6.16 Software Watchdog Lower Limit RUN Current Register (SWDG_RUNLLC)." You can change the set value of this register even after the LOCK bit in the Software watchdog configuration register (SWDG_CFG) is set.

Bit	31	0
Field	WDGRUNLLS[31:0]	
R/W Attribute	R/W	
Protection Attribute	WPS	
Initial Value	00000000_00000000_00000000_00000000	

[bit31:0] WDGRUNLLS[31:0]: Window Lower Limit Value for RUN Set Bits

These bits are used to define the window lower limit value for RUN. The reading of these bits returns the set value regardless of the Software watchdog configuration register (SWDG_CFG) LOCK bit value.

WDGRUNLLS[31:0]	Description
During write operation	When "0" is written to all 32 bits: The window function is disabled. When any value except "0" is written: The window function is enabled.
During read operation	Set value read

Note:

- *This register value is different from the window lower limit value that is actually used. The window lower limit value actually used is the value in the Software watchdog lower limit RUN current register (SWDG_RUNLLC).*

6.11. Software Watchdog Upper Limit RUN Setting Register (SWDG_RUNULS)

When performing write access to this register, follow the watchdog register write protection sequence. A setting value of the window upper limit value for RUN is written to this register. However, this register value is different from the window upper limit value for RUN that is actually used. For details on how to reflect this register value in the window upper limit value for RUN actually used, see "6.17 Software Watchdog Upper Limit RUN Current Register (SWDG_RUNULC)" You can change the set value of this register even after the LOCK bit in the Software watchdog configuration register (SWDG_CFG) is set.

Bit	31	0
Field	WDGRUNULS[31:0]	
R/W Attribute	R/W	
Protection Attribute	WPS	
Initial Value	00000001_00000000_00000000_00000000	

[bit31:0] WDGRUNULS[31:0]: Window Upper Limit Value for RUN Set Bits

These bits are used to define the window upper limit value for RUN. The reading of these bits returns the set value regardless of the Software watchdog configuration register (SWDG_CFG) LOCK bit value.

WDGRUNULS[31:0]	Description
During write operation	Set value written
During read operation	Set value read

Note:

- *This register value is different from the window upper limit value that is actually used. The window upper limit value actually used is the value in the Software watchdog upper limit RUN current register (SWDG_RUNULC).*

6.12. Software Watchdog Lower Limit PSS Setting Register (SWDG_PSSLLS)

When performing write access to this register, follow the watchdog register write protection sequence. A setting value of the window lower limit value for PSS is written to this register. However, this register value is different from the window lower limit value for PSS that is actually used. For details on how to reflect this register value in the window lower limit value for PSS actually used, see "6.18 Software Watchdog Lower Limit PSS Current Register (SWDG_PSSLLC)." You can change the set value of this register even after the LOCK bit in the Software watchdog configuration register (SWDG_CFG) is set.

Bit	31	0
Field	WDGPSSLLS[31:0]	
R/W Attribute	R/W	
Protection Attribute	WPS	
Initial Value	00000000_00000000_00000000_00000000	

[bit31:0] WDGPSSLLS[31:0]: Window Lower Limit Value for PSS Set Bits

These bits are used to define the window lower limit value for PSS. The reading of these bits returns the set value regardless of the Software watchdog configuration register (SWDG_CFG) LOCK bit value.

WDGPSSLLS[31:0]	Description
During write operation	When "0" is written to all 32 bits: The window function is disabled. When any value except "0" is written: The window function is enabled.
During read operation	Set value read

Note:

- *This register value is different from the window lower limit value that is actually used. The window lower limit value actually used is the value in the Software watchdog lower limit PSS current register (SWDG_PSSLLC).*

6.13. Software Watchdog Upper Limit PSS Setting Register (SWDG_PSSULS)

When performing write access to this register, follow the watchdog register write protection sequence. A setting value of the window upper limit value for PSS is written to this register. However, this register value is different from the window upper limit value for PSS that is actually used. For details on how to reflect this register value in the window upper limit value for PSS actually used, see "6.19 Software Watchdog Upper Limit PSS Current Register (SWDG_PSSULC)." You can change the set value of this register even after the LOCK bit in the Software watchdog configuration register (SWDG_CFG) is set.

Bit	31	0
Field	WDGPSSULS[31:0]	
R/W Attribute	R/W	
Protection Attribute	WPS	
Initial Value	10000000_00000000_00000000_00000000	

[bit31:0] WDGPSSULS[31:0]: Window Upper Limit Value for PSS Set Bits

These bits are used to define the window upper limit value for PSS. The reading of these bits returns the set value regardless of the Software watchdog configuration register (SWDG_CFG) LOCK bit value.

WDGPSSULS[31:0]	Description
During write operation	Set value written
During read operation	Set value read

Note:

- *This register value is different from the window upper limit value that is actually used. The window upper limit value actually used is the value in the Software watchdog upper limit PSS current register(SWDG_PSSULC).*

6.14. Software Watchdog Reset Delay Counter Register (SWDG_RSTDLY)

When performing write access to this register, follow the watchdog register write protection sequence. This register is used to set the number of cycles for the delay time from the occurrence of a watchdog error to the watchdog reset request or watchdog interrupt request (NMI). The reference clock for the delay time is the source clock of the watchdog counter. When the IRQEN bit in the Software watchdog interrupt configuration register (SWDG_INT) is "1", this register indicates the number of cycles for the delay time from the prior warning interrupt request to the watchdog reset request or watchdog interrupt request (NMI).

Bit	bit31	bit16
Field	Reserved	
R/W Attribute	R0, WX	
Protection Attribute	WPS	
Initial Value	00000000_00000000	

Bit	bit15	bit0
Field	WDGRSTDLY[15:0]	
R/W Attribute	R0, W	
Protection Attribute	WPS	
Initial Value	00000000_00000000	

[bit31:16] Reserved: Reserved Bits

[bit15:0] WDGRSTDLY[15:0]: Reset/NMI Delay Counter Bits

These bits define the number of cycles for the delay time that is to be inserted before generation of the watchdog reset request or watchdog interrupt request (NMI). The reference clock for this delay time is the source clock of the watchdog counter.

WDGRSTDLY[15:0]	Description
During write operation	Set value written
During read operation	0x0000 is read.

6.15. Software Watchdog Configuration Register (SWDG_CFG)

When performing write access to this register, follow the watchdog register write protection sequence. This register is used to make the operation settings of the Software watchdog timer.

Bit	bit31	bit25	bit24
Field	Reserved		LOCK
R/W Attribute	R0, WX		R, W
Protection Attribute	WPS		
Initial Value	0000000		0

Bit	bit23	bit21	bit20	bit19	bit18	bit17	bit16
Field	Reserved		OBSSEL[4:0]				
R/W Attribute	R0, WX		R/W				
Protection Attribute	WPS						
Initial Value	000		00000				

Bit	bit15	bit10	bit9	bit8
Field	Reserved		CLKSEL[1:0]	
R/W Attribute	R0, WX		R/W	
Protection Attribute	WPS			
Initial Value	000000		00	

Bit	bit7	bit4	bit3	bit2	bit1	bit0
Field	Reserved			ALLOWSTOPCLK	WDENPSS	WDENRUN
R/W Attribute	R0, WX			R/W	R/W	R/W
Protection Attribute	WPS					
Initial Value	00000			0	1	1

[bit31:25] Reserved: Reserved Bits

[bit24] LOCK: Lock Bit

You can write data to this bit only once. This register is used to prevent rewriting of the set values of various registers. When this bit is "0", you can rewrite the setting values of various registers. When this bit is set to "1" the watchdog counter is automatically cleared. This bit is equipped with a majority circuit with 3 FFs as a measure against a bit invert caused by the effects of noise or another factor.

LOCK	Description
During write operation	When "0" is written: Invalid When "1" is written: The set values of the registers are locked.
During read operation	When "0" is read: The lock for the register set values is disabled. When "1" is read: The lock for the register set values is enabled.

[bit23:21] Reserved: Reserved Bits

[bit20:16] OBSSEL[4:0]: Watchdog Counter Monitor Bit Output Selection Bits

These bits are used for selecting any 1 bit in the watchdog counter (32 bits) as the output of the watchdog counter monitor bit.(Support target of this function is different depends on product specification. For details, see “Output of the Watchdog Counter Monitor Bit for MCU Output Pin” in 3 Explanation of Operation.

OBSSEL[4:0]	Description
During write operation	When 0b00000 is written: Bit 0 is selected for the monitor output. When 0b00001 is written: Bit 1 is selected for the monitor output. When 0b00010 is written: Bit 2 is selected for the monitor output. ・ ・ ・ When 0b11111 is written: Bit 31 is selected for the monitor output.
During read operation	Set value read

Note:

- *If products do not support "Output of watchdog counter monitor bit for MCU output pin", Any data written to these bits does not affect operation.*

[bit15:10] Reserved: Reserved Bits

[bit9:8] CLKSEL[1:0]: Clock Selection Bits

These bits are used to select the source clock for the watchdog counter. When activated, the watchdog counter starts its operation as the high-speed CR clock. For details on clock switching, see "7 Precautions for Using This Device Switching of watchdog counter source clock."

CLKSEL[1:0]	Description
During write operation	When 0b00 is written: The high-speed CR clock is selected. When 0b01 is written: The low-speed CR clock is selected. When 0b10 is written: The sub clock is selected. When 0b11 is written: The main clock is selected.
During read operation	Set value read

[bit7:3] Reserved: Reserved Bits

[bit2] ALLOWSTOPCLK: Clock Stop for PSS Enable Bit

This bit is used to enable transition to PSS that involves the source clock stop of the watchdog counter. This bit is valid only when the WDENPSS bit in the Software watchdog configuration register (SWDG_CFG) is "1".

ALLOWSTOPCLK	Description
During write operation	When "0" is written: Watchdog clock stop in PSS is disabled. When "1" is written: Watchdog clock stop in PSS is enabled.
During read operation	Set value read

[bit1] WDENPSS: Watchdog Counter for PSS Enable Bit

This bit is used to enable the watchdog counter in PSS. This bit is enabled when the LOCK bit in the Software watchdog configuration register (SWDG_CFG) is set to "1".

This bit is equipped with a majority circuit with 3 FFs as a measure against a bit invert caused by the effects of noise or another factor.

WDENPSS	Description
During write operation	When "0" is written: The watchdog counter in PSS is disabled. When "1" is written: The watchdog counter in PSS is enabled.
During read operation	Set value read

[bit0] WDENRUN: Watchdog Counter for RUN Enable Bit

This bit is used to enable the watchdog counter in RUN. This bit is enabled when the LOCK bit in the Software watchdog configuration register (SWDG_CFG) is set to "1".

This bit is equipped with a majority circuit with 3 FFs as a measure against a bit invert caused by the effects of noise or another factor.

WDENRUN	Description
During write operation	When "0" is written: The watchdog counter in RUN is disabled. When "1" is written: The watchdog counter in RUN is enabled.
During read operation	Set value read

Note:

- *When the ALLOWSTOPCLK bit in the Software watchdog configuration register (SWDG_CFG) is "1", transition to PSS that involves clock stop is enabled. However, in case of transition from RUN to PSS during reset delay counter operation, the reset delay counter is stopped until a return from PSS to RUN occurs.*

6.16. Software Watchdog Lower Limit RUN Current Register (SWDG_RUNLLC)

This register is used to read the window lower limit value for RUN that is actually used. This register fetches the set value in the Software watchdog lower limit RUN setting register (SWDG_RUNLLS) when the LOCK bit in the Software watchdog configuration register (SWDG_CFG) is set. You can change the set value of this register even after the LOCK bit in the Software watchdog configuration register (SWDG_CFG) is set. In this case, this register fetches the set value of the software watchdog lower limit RUN setting register (SWDG_RUNLLS) when the watchdog counter clear protection sequence is executed.

Bit	bit31	bit0
Field	WDGRUNLLC[31:0]	
R/W Attribute	R, WX	
Protection Attribute	-	
Initial Value	00000000_00000000_00000000_00000000	

[bit31:0] WDGRUNLLC[31:0]: Window Lower Limit for RUN Current Bits

These bits are used to define the window lower limit value for RUN. The reading of these bits returns the initial value, until the Software watchdog configuration register (SWDG_CFG) LOCK bit is set.

WDGRUNLLC[31:0]	Description
During write operation	Invalid
During read operation	Set value read

Note:

- *An AHB transfer error response is generated where there is write access to this register.*

6.17. Software Watchdog Upper Limit RUN Current Register (SWDG_RUNULC)

This register is used to read the window upper limit value for RUN that is actually used. This register fetches the set value in the Software watchdog upper limit RUN setting register (SWDG_RUNULS) when the LOCK bit in the Software watchdog configuration register (SWDG_CFG) is set. You can change the set value of this register even after the LOCK bit in the Software watchdog configuration register (SWDG_CFG) is set. In this case, this register fetches the set value of the software watchdog upper limit RUN setting register (SWDG_RUNULS) when the watchdog counter clear protection trigger sequence is executed.

Bit	bit31	bit0
Field	WDGRUNULC[31:0]	
R/W Attribute	R, WX	
Protection Attribute	-	
Initial Value	00000001_00000000_00000000_00000000	

[bit31:0] WDGRUNULC[31:0]: Window Upper Limit for RUN Current Bits

These bits are used to define the window upper limit value for RUN. The reading of these bits returns the initial value, until the Software watchdog configuration register (SWDG_CFG) LOCK bit is set.

WDGRUNULC[31:0]	Description
During write operation	Invalid
During read operation	Set value read

Note:

- *An AHB transfer error response is generated where there is write access to this register.*

6.18. Software Watchdog Lower Limit PSS Current Register (SWDG_PSSLLC)

This register is used to read the window lower limit value for PSS that is actually used. This register fetches the set value in the Software watchdog lower limit PSS setting register (SWDG_PSSLLS) when the LOCK bit in the Software watchdog configuration register (SWDG_CFG) is set. You can change the set value of this register even after the LOCK bit in the Software watchdog configuration register (SWDG_CFG) is set. In this case, this register fetches the set value of the software watchdog lower limit PSS setting register (SWDG_PSSLLS) when the watchdog counter clear protection trigger sequence is executed.

Bit	bit31	bit0
Field	WDGPSSLLC[31:0]	
R/W Attribute	R, WX	
Protection Attribute	-	
Initial Value	00000000_00000000_00000000_00000000	

[bit31:0] WDGPSSLLC[31:0]: Window Lower Limit for PSS Current Bits

These bits are used to define the window lower limit value for PSS. The reading of these bits returns the initial value, until the Software watchdog configuration register (SWDG_CFG) LOCK bit is set.

WDGPSSLLC[31:0]	Description
During write operation	Invalid
During read operation	Set value read

Note:

- *An AHB transfer error response is generated where there is write access to this register.*

6.19. Software Watchdog Upper Limit PSS Current Register (SWDG_PSSULC)

This register is used to read the window upper limit value for PSS that is actually used. This register fetches the set value in the Software watchdog upper limit PSS setting register (SWDG_PSSULS) when the LOCK bit in the Software watchdog configuration register (SWDG_CFG) is set. You can change the set value of this register even after the LOCK bit in the Software watchdog configuration register (SWDG_CFG) is set. In this case, this register fetches the set value of the software watchdog upper limit PSS setting register (SWDG_PSSULS) when the watchdog counter clear protection trigger sequence is executed.

Bit	bit31	bit0
Field	WDGPSSULC[31:0]	
R/W Attribute	R, WX	
Protection Attribute	-	
Initial Value	10000000_00000000_00000000_00000000	

[bit31:0] WDGPSSULC[31:0]: Window Upper Limit for PSS Current Bits

These bits are used to define the window upper limit value for PSS. The reading of these bits returns the initial value, until the Software watchdog configuration register (SWDG_CFG) LOCK bit is set.

WDGPSSULC[31:0]	Description
During write operation	Invalid
During read operation	Set value read

Note:

- *Write access to this register generates an AHB transfer error response.*

7. Precautions for Using This Device

This section describes precautions on the use of the Software watchdog timer.

Watchdog Window Setting

For the Software watchdog timer, you can make 2 types of window setting, the setting for RUN and setting for PSS. The user must make an appropriate window setting for each device state.

When the device transitions from PSS to RUN, the Software watchdog timer continues operation using the PSS window setting until the watchdog counter clear protection trigger sequence is executed. The watchdog counter clear protection trigger sequence switches the window setting to the RUN window setting. You can control the watchdog function in each device state with the WDENRUN bit and WDENPSS bit in the Software watchdog configuration register (SWDG_CFG).

Only PSS operation is not recommend at Software watchdog timer.

Because of Software watchdog timer can clear the counter under working the timer itself.

Software watchdog timer cannot clear the counter under CPU stopping.

Notes:

- *When WDENPSS bit is "0", the watchdog timer perform the same operation. So certainly set the same value in upper limit registers and lower limit registers.*
- *Set the same value following two upper limit registers.*
  - Software watchdog lower limit PSS setting register (SWDG_PSSLLS)
  - Software watchdog lower limit RUN setting register (SWDG_RUNLLS)
- *Set the same value following two upper limit registers.*
  - Software watchdog upper limit PSS setting register (SWDG_PSSULS)
  - Software watchdog upper limit RUN setting register (SWDG_RUNULS)

Switching of Watchdog Counter Source Clock

The system controller controls the switching of the watchdog counter source clock since the source clock control of the watchdog counter is a part of the system setting information (profile). To change the watchdog counter source clock, follow the sequence below.

1. Writing data to the RUN profile source clock enable register of the system controller (SYSC0_RUNCKSRER) enables oscillation of the new source clock of the watchdog. Then, writing 0xAB to the RUN profile update trigger register of the system controller (SYSC0_TRGRUNCNTR) executes RUN profile update.
2. Setting the CLKSEL bit in the Software watchdog configuration register (SWDG_CFG) sets the new source clock of the watchdog counter. Then, writing data to the following registers finalizes the window setting.
3. Software watchdog lower limit RUN setting register (SWDG_RUNLLS)
4. Software watchdog upper limit RUN setting register (SWDG_RUNULS)
5. Software watchdog lower limit PSS setting register (SWDG_PSSLLS)
6. Software watchdog upper limit PSS setting register (SWDG_PSSULS)
7. Setting the LOCK bit in the Software watchdog configuration register (SWDG_CFG) to "1" locks rewriting of the register set values.
8. Writing 0xAB to the RUN profile update trigger register of the system controller (SYSC0_TRGRUNCNTR) executes RUN profile update. At this point in time, the source clock of the watchdog counter is switched to the new source clock selected by the CLKSEL bit of the Software watchdog configuration register (SWDG_CFG).

Note:

- *When the oscillation of the new source clock is stable, Step 1 is unnecessary.*

Watchdog Error

A watchdog error is generated under the following conditions. This watchdog error generates the watchdog reset request/interrupt request (NMI).

1. An incorrect value is written to the Software watchdog trigger 0/1 register (SWDG_TRG0 or SWDG_TRG1).
2. The watchdog counter clear protection trigger sequence is violated.
3. The watchdog counter clear protection trigger sequence is not executed before the watchdog counter reaches the window upper limit value.
4. The watchdog counter clear protection trigger sequence is executed before the watchdog counter reaches the window lower limit value.
5. The watchdog counter clear protection trigger sequence is executed while the LOCK bit in the Software watchdog configuration register (SWDG_CFG) is still "0".

Factors for Bus Error Responses (Data Abort)

A bus error response is generated in the following cases.

- Access (read or write) to an address where no register exists
- Violation of the watchdog register write protection sequence
- Write access to a register with the read-only attribute (indicated as R, WX)
- Write access to a register after the LOCK bit in the Software watchdog configuration register (SWDG_CFG) is set to "1"
- Write access with an incorrect value to the Software watchdog protection register (SWDG_PROT)
- Byte/Half-word write access to the Software watchdog protection register (SWDG_PROT)
- Write access to a register in non-privileged mode
  - (Except for the Software watchdog trigger 0/1 register (SWDG_TRG0 and SWDG_TRG1))

Watchdog Timer Operation in Standby State of Processor

The Software watchdog timer operates using the RUN window setting in the following case.

- Without a valid key being written to the PSS profile update enable register (SYSC0_PSSENR.PSSEN0 and SYSC1_PSSENR.PSSEN1), the processor transitions to the standby state by executing a wait for interrupt (WFI) instruction.

CHAPTER 16: TCRAM Interface

This chapter provides an overview of the interface between the Cortex-R5F BTCM port and SRAM, and describes their configuration, operation, and registers.

1. **Overview**
2. **Configuration**
3. **Explanation of Operation**
4. **Setting Procedure Examples**
5. **Registers**

TCRAM-TXXPT03P01R01L06-E1-XX

1. Overview

This section describes the features of the TCRAM interface.

Features

The TCRAM interface has the following features.

- Connection between the Cortex-R5F BTCM port and SRAM (RAM size is Product specific).

- Wait control function
  The Cortex-R5F BTCM port has a signal for waiting for data read (BxTCWAIT).

  The TCRAM interface sets and controls the number of cycles of the wait signal (for data write, there is no wait control).

- Generation of a bus error response
  The TCRAM interface generates an error when non-privilege access or unsupported access to the configuration register is attempted.

  For details on unsupported access, see "Generation of a bus error response" in "3. Explanation of Operation."

- ECC error insertion function
  The Cortex-R5F performs an ECC check.

  The TCRAM interface can insert an error in data read from the TCRAM for the ECC function test performed by the Cortex-R5F.

- Direct write/read test function of the ECC area
  A function to write an arbitrary value to the ECC area is implemented for test.

  Also, a function that captures a read value before error correction and stores it to the register is implemented.

- RAM diagnosis and initialization function
  Diagnosis and initialization are performed for the TCRAM. One or more methods from the following can be selected to perform diagnosis.

  - Unique (as unique data, address information (below), Upper 7 bits are ECC data.)

    Address Information
    ((address & 0x0000007F) << 32)
    + ((address & 0x000000FF) << 24)
    + ((address & 0x000000FF) << 16)
    + ((address & 0x000000FF) << 8)
    + (address & 0x000000FF)

  - Checker
  - March (performed in the order of all "0's" to all "1's")

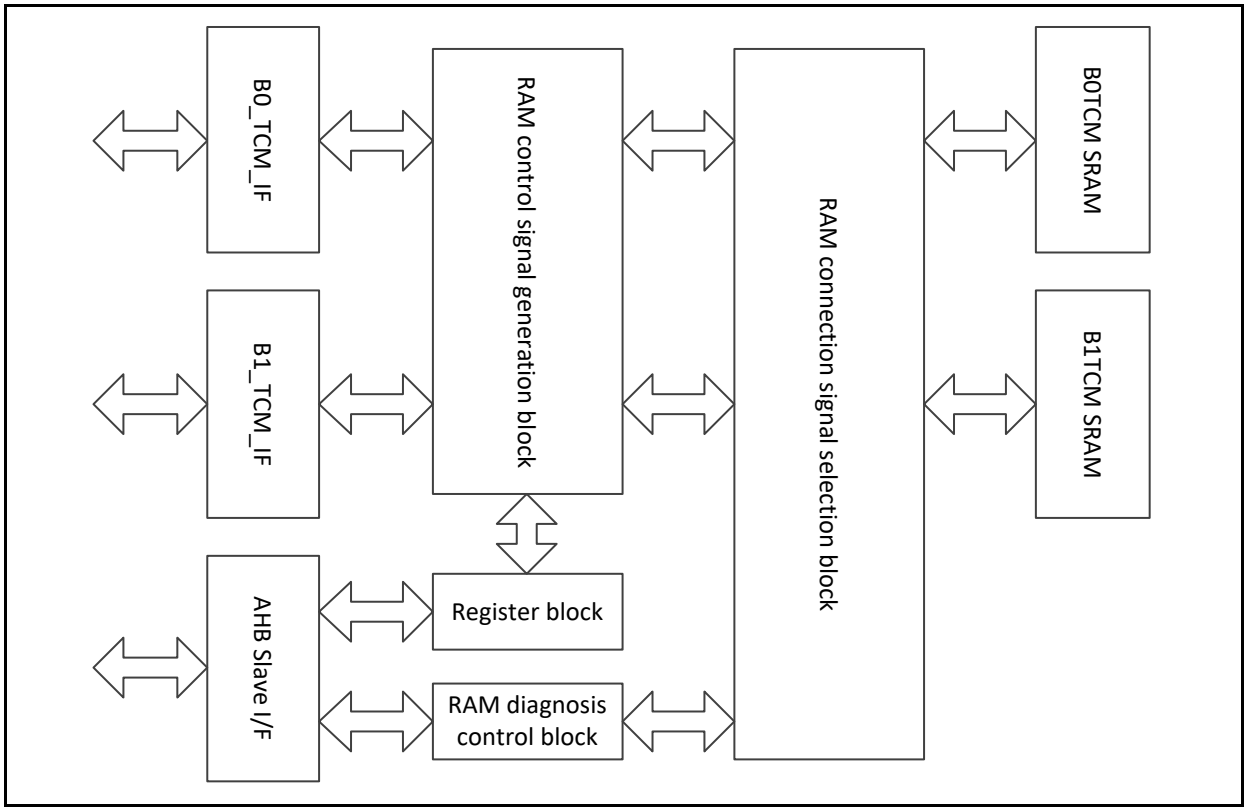
  One of the following can be selected as initialization.

  - all "0" is written
  - all "1" is written

2. Configuration

Figure 2-1 is a block diagram of the TCRAM interface that uses B0TCM and B1TCM.

Figure 2-1 TCRAM Interface Block Diagram



B0_TCM_IF

Interface for the Cortex-R5F B0TCM port

B1_TCM_IF

Interface for the Cortex-R5F B1TCM port

AHB Slave I/F

AHB Slave interface, which is used as a bus for configuration

Register Block

Includes the registers of the TCRAM interface.

The registers are common to the B0TCM and B1TCM, and set values are reflected in both ports.

RAM Control Signal Generation Block

Generates control signals of the TCRAM from signals input at the Cortex-R5F BTCM port.

RAM Diagnosis Control Block

Controls diagnosis and initialization of the SRAM.

RAM Connection Signal Selection Block

Selects the BTCM port or RAM diagnosis control block as a path for connecting to the SRAM.

3. Explanation of Operation

This section describes the operation of the TCRAM interface.

Wait Control

The wait status can be configured by setting TRCFGn_TCMCFG0:DWAIT via the configuration bus when privilege access is allowed and the TRCFGn_TCMCFG0:LOCKSTATUS bit is "0".

The number of waits that can be set is from 0 to 3, and the default set value is the least number of waits (0 waits).

The asserted BxTCWAIT signal indicates the wait status.

The Cortex-R5F is kept waiting for acquisition of data read from the SRAM while the BxTCWAIT signal is asserted. There is no wait regarding write operation.

Generation of a Bus Error Response

The TCRAM interface returns an error response to the configuration bus under the following conditions.

- Non-privilege write access to a TCRAM interface register
- Write access to a TCRAM interface register that is not the TRCFGn_TCMUNLOCK register in the locked state
- Write access to the TRCFGn_TCMUNLOCK register with the unlock value or any value except the lock value
- Write access that is not 32-bit access to the TRCFGn_TCMUNLOCK register
- Write access to a reserved space
- Read or write access to a space that is not used as a register space
- Write access when all the write target bits are reserved or read-only bits

ECC Check

Since the following access to an uninitialized area of the TCRAM causes an ECC error, 32-bit initialization is necessary before using it.

- 8- or 16-bit write access
- Read access

The Cortex-R5F performs an ECC check for the TCRAM in units of 32 bits. For 32- or 64-bit write operation, add ECC to data and then write it to the TCRAM. For 8- or 16-bit write operation, perform read-modify-write access so that the correct ECC can be generated. An ECC check is also performed during read operation in read-modify-write access.

ECC Error Insertion

This interface has a function to insert an error into data that is read from the SRAM so that the 32-bit ECC function of the Cortex-R5F can be tested.

When the TRCFGn_TCMCFG0:LOCKSTATUS bit is "0", this function sets "1" in a bit in the configuration register that causes an ECC error in the RAM.

To generate an ECC error by data area destruction, set the TRCFGn_TCMCFG1:ERRBIT[31:0] bits, and to generate one by ECC area destruction, set the TRCFGn_TCMCFG0:ERRECC[6:0] bits.

For read from the TCRAM the read data is XOR-ed with TRCFGn_TCMCFG1:ERRBIT[31:0], and the ECC data is XOR-ed with TRCFGn_TCMCFG0:ERRECC[6:0].

Note:

- *During normal operation, keep TRCFGn_TCMCFG1:ERRBIT[31:0] and TRCFGn_TCMCFG0:ERRECC[6:0] set to "0".*

Direct Write/Read Function of the ECC Area

A direct write/read function of the ECC data storage area is implemented for the SRAM test.

By using this function, you can write arbitrary data into the ECC area of the SRAM, and read the data before ECC error correction and the ECC data.

This function is a testing function. To prevent incorrect setting during the execution of a user program, the direct access enable bit and direct write access enable bit are protected by their relevant lock bits.

In the locked state, write access to the relevant registers is ignored, while read access is not restricted.

Direct Read Function

A function that captures data before ECC error correction read from the TCRAM and its ECC data into a register

When the direct access enable bit is set (TRCFGn_ECCDEN:DEN = 1), reading the TCRAM has this function store the read data and ECC data in a register.

The direct access enable bit is set or cleared under the following conditions.

[Setting condition]

- When TRCFGn_TCMCFG0:LOCKSTATUS is "0", 0b10011110 is written in bit[23:16] in the TRCFGn_ECCDEN register (DEN, DENUNLOCK) in a state in which privilege access is possible.

[Clearing conditions]

- TRCFGn_TCMCFG0:LOCKSTATUS becomes "1".
- A value that is not 0b10011110 is written in bit[23:16] in the TRCFGn_ECCDEN register (DEN, DENUNLOCK) in a state in which privilege access is possible.

For 32-bit read operation, only 32 bits in the data register and their corresponding 32 bits in ECC data are updated, and the data register and ECC data bits for unread data are not updated.

Since registers are used commonly for the B0TCM and B1TCM ports, when read access via the B0TCM and B1TCM ports occurs at the same time, data on the B0TCM port side is stored in the register.

Direct Write Function

This function writes an arbitrary value set in a register to the ECC data area of the TCRAM instead of writing the ECC calculated from the written data.

When the direct write access enable bit is set (TRCFGn_ECCDW:DWEN = 1), and write operation is performed to the TCRAM, this function writes the value in the ECCDW_PARITY[13:0] bits in the TRCFGn_ECCDW register.

For 32-bit write access, only the ECC bits for the word to be written are updated and the ECC bits for the unwritten word are not updated.

The direct write access enable bit is cleared under the following conditions.

[Setting condition]

- When TRCFGn_ECCDEN:DEN is "1", 0b10110101 is written in bit[31:24] in the TRCFGn_ECCDW register (DWEN, DWENUNLOCK) in a state in which privilege access is possible.

[Clearing conditions]

- TRCFGn_ECCDEN:DEN becomes "0".
- A value that is not 0b10110101 is written in bit[31:24] in the TRCFGn_ECCDW register (DWEN, DWENUNLOCK) in a state in which privilege access is possible.

Note:

- *Before using this function, set the stack pointer and make other settings in an area that is not the area subject to the direct access.*

RAM Diagnosis and Initialization Function

RAM Diagnosis

Select RAM diagnosis items to perform for the TCRAM from the following. (Multiple items can be selected.)

- Unique (as unique data, address information (below), Upper 7 bits are ECC data.)
  Address Information
  ((address & 0x0000007F) << 32)
  + ((address & 0x000000FF) << 24)
  + ((address & 0x000000FF) << 16)
  + ((address & 0x000000FF) << 8)
  + (address & 0x000000FF)
- Checker
- March (performed in the order of all "0's" to all "1's")

The RAM diagnosis is performed only in the following sequence.

1. Unique
2. Checker
3. March

Diagnosis items to be performed are selected according to the setting of the TTYP[2:0] bits in the TEST diagnosis function register (TRCFGn_TTCR). (Unique and Checker are selected by default.)

The range for Diagnosis can be defined with TEST start address register (TRCFGn_TASAR) and TEST end address register (TRCFGn_TAEAR).

Diagnosis is initiated with the key code control by software.

The following shows the RAM diagnosis start procedure.

1. Wait until TRUN bit of TEST diagnosis function register (TRCFGn_TTCR) and IRUN bit of TEST initialization function register both become "0" by reading these bits.
2. Clear TRCFGn_TTCR:TCI and TRCFGn_TICR:ICI.
   To clear TRCFGn_TTCR:TCI and TRCFGn_TICR:ICI, write "1" to the corresponding bit of TRCFGn_TSCR.
3. To the TEST key code control register (TRCFGn_TKCCR), write 0x02, 0x42, 0x82, 0xC2 in a row in this sequence to TEST keycode control register to start Diagnosis.
4. Wait until Diagnosis finishes (TRCFGn_TTCR:TCI=1).

Notes:

- *When you use this function, all the data bits of the RAM contain "1" after the March diagnosis, or some values after diagnosis other than it.*
- *If data on the RAM is read before initialization or 32- or 64-bit write access, an ECC error occurs. Therefore, be sure to perform initialization or write access.*
- *Also, it is prohibited to access the RAM when it is under diagnosis.*

RAM Initialization

For the TCRAM initialization operation, you can select either of the following by the specification of the ITYP bit in the TEST initialization function register (TRCFGn_TICR).

- Writing all "0's" (default)
- Writing all "1's"

To the ECC area, the value corresponding to the written value is written.

You can specify a range for performing RAM initialization with the TEST start address register (TRCFGn_TASAR) and TEST end address register (TRCFGn_TAEAR).

The following shows the RAM initialization start procedure.

1. Wait until TRUN bit of TEST diagnosis function register (TRCFGn_TTCR) and IRUN bit of TEST initialization function register both become "0" by reading these bits.
2. Clear TRCFGn_TTCR:TCI and TRCFGn_TICR:ICI.
   To clear TRCFGn_TTCR:TCI and TRCFGn_TICR:ICI, write "1" to the corresponding bit of TRCFGn_TSCR.
3. To the TEST key code control register (TRCFGn_TKCCR), write
   0x01, 0x41, 0x81, and 0xC1
   in this sequence in a row to start initialization.
4. Wait until Diagnosis finishes (TRCFGn_TICR:ICI=1).

Note:

- *It is prohibited to access the RAM when it is being initialized.*

Forcibly Stopping RAM Diagnosis and Initialization

You can forcibly stop the execution of diagnosis and initialization of the TCRAM.

The following shows the procedure for the forcible stop.

1. During the diagnosis or initialization, to the TEST key code control register (TRCFGn_TKCCR), write 0x00, 0x40, 0x80, and 0xC0 in this sequence in a row.

2. Wait until TRCFGn_TTCR:TRUN becomes "0" for diagnosis, and wait until TRCFGn_TICR:IRUN is "0" for initialization.

Operation when a RAM Diagnosis Error Occurs

If an error occurs during RAM diagnosis, the contents of the diagnosis in which the error occurs and the address where it occurs are stored in the TRCFGn_TEAR0 to 2 registers.

When 4 or more errors occur, TRCFGn_TTCR:OVFLW is set to "1".

RAM Diagnosis Interrupt

You can generate an interrupt for RAM diagnosis and initialization.

Table 3-1 shows interrupt factors, generation flag bits, and enable bits.

Table 3-1 Correspondence between Interrupt Factors, Generation Flag Bits, and Enable Bits

Interrupt Factor	Generation Flag Bit	Enable Bit
Diagnosis end interrupt	TRCFGn_TTCR:TCI	TRCFGn_TTCR:TCIE
Diagnosis error interrupt	TRCFGn_TTCR:TEI	TRCFGn_TTCR:TEIE
Initialization end interrupt	TRCFGn_TICR:ICI	TRCFGn_TICR:ICIE

The TCRAM interface output 1 interrupt signal collectively indicates the above 3 types of interrupts.

You can check what type of interrupt has occurred by reading the TCI and TEI bits in TRCFGn_TTCR and the ICI bit in TRCFGn_TICR.

(By reading data at the offset address 0x0000_0040 in units of words, you can check it by 1-time read access.)

Generation of a RAM Diagnosis Pseudo Error

With this function, you can generate a pseudo error on purpose for debugging software.

The operation for generating a pseudo error is set as the following procedure.

1. Selecting an error type by setting the TEST pseudo error generation control register (TRCFGn_TFECR)
   - Set a diagnosis pattern that generates a pseudo error in TRCFGn_TFECR:ETYP[2:0].
   - Identify a diagnosis pattern that generates a pseudo error by writing "1" in TRCFGn_TFECR:FERR.
2. Starting RAM diagnosis (For the procedure see "Starting RAM diagnosis.")

4. Setting Procedure Examples

Figure 4-1 shows a procedure for register setting of the TCRAM interface,Figure 4-2 shows a procedure for direct read access, Figure 4-3 shows a procedure for direct write access, and Figure 4-4 shows a procedure for starting RAM diagnosis and initialization.

Figure 4-1 Register Setting Flow

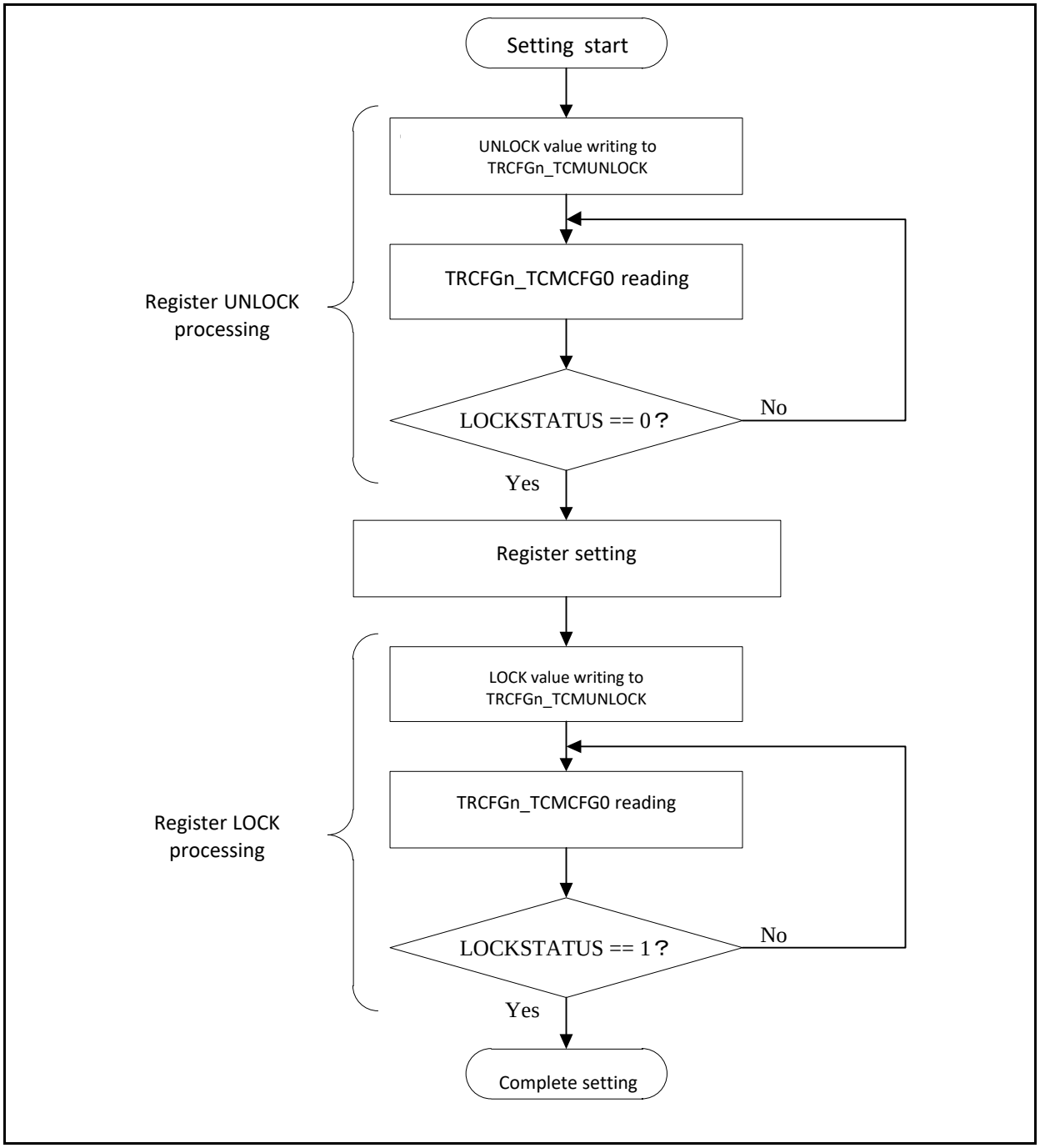
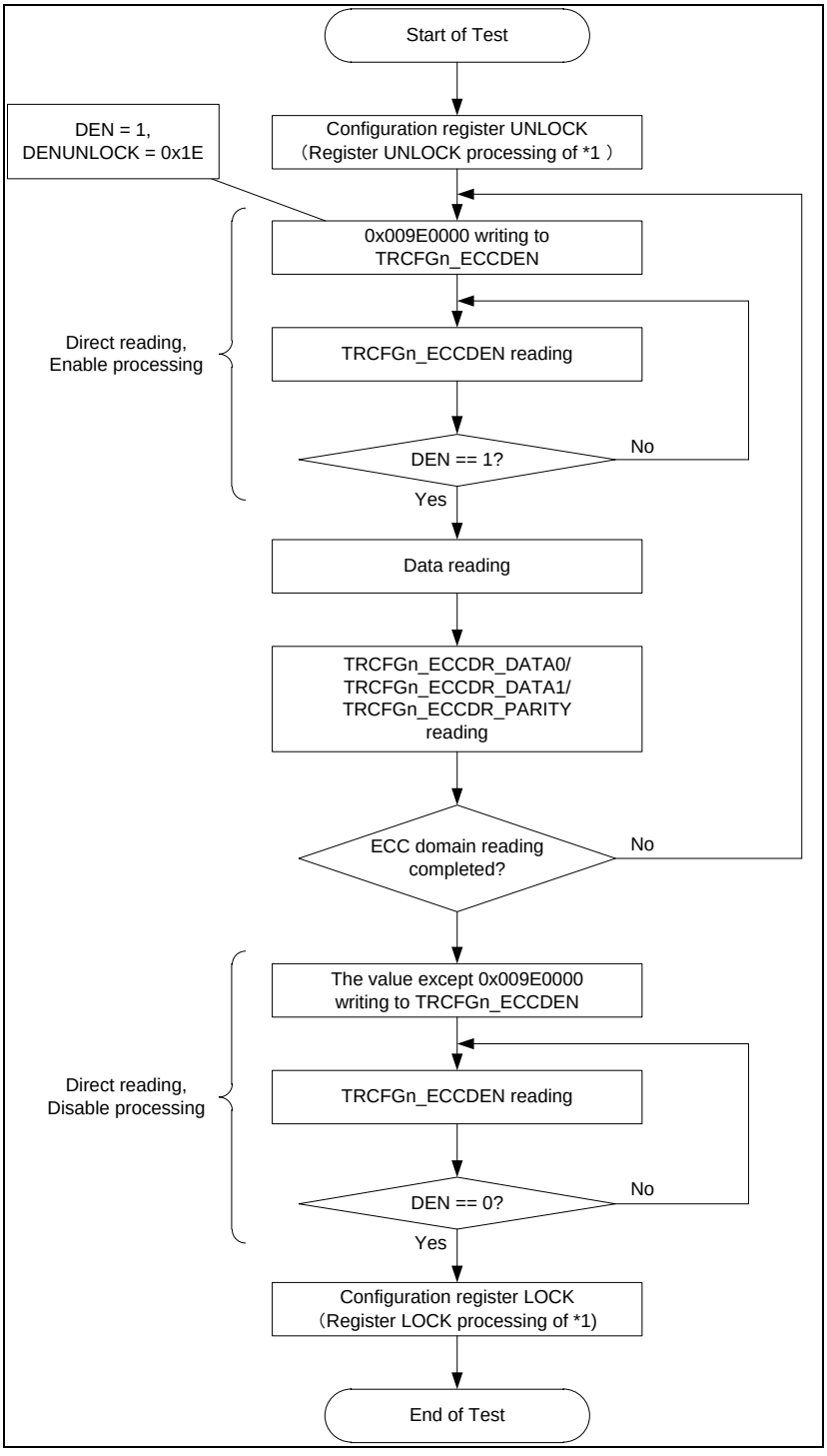
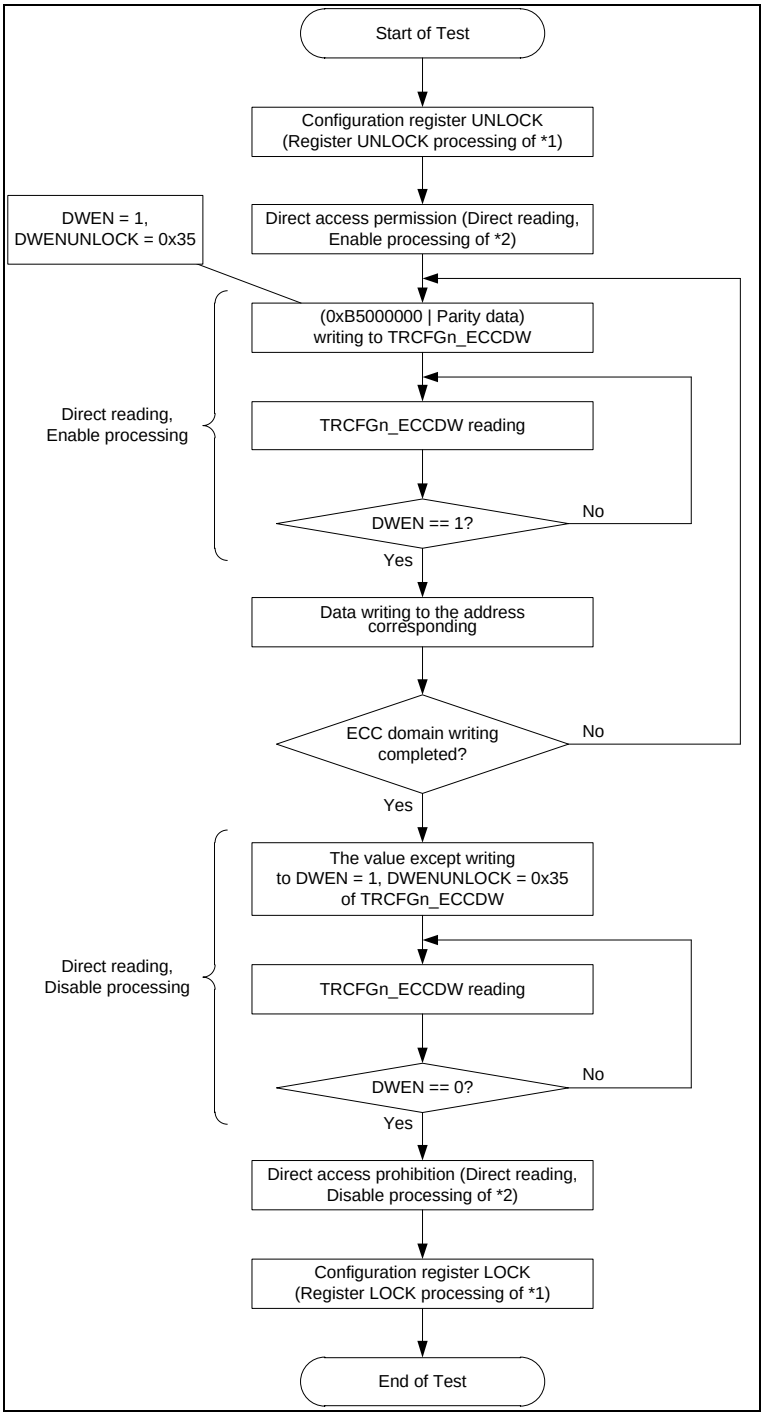


Figure 4-2 Direct Read Access Flow



*1 See Figure 4-1.

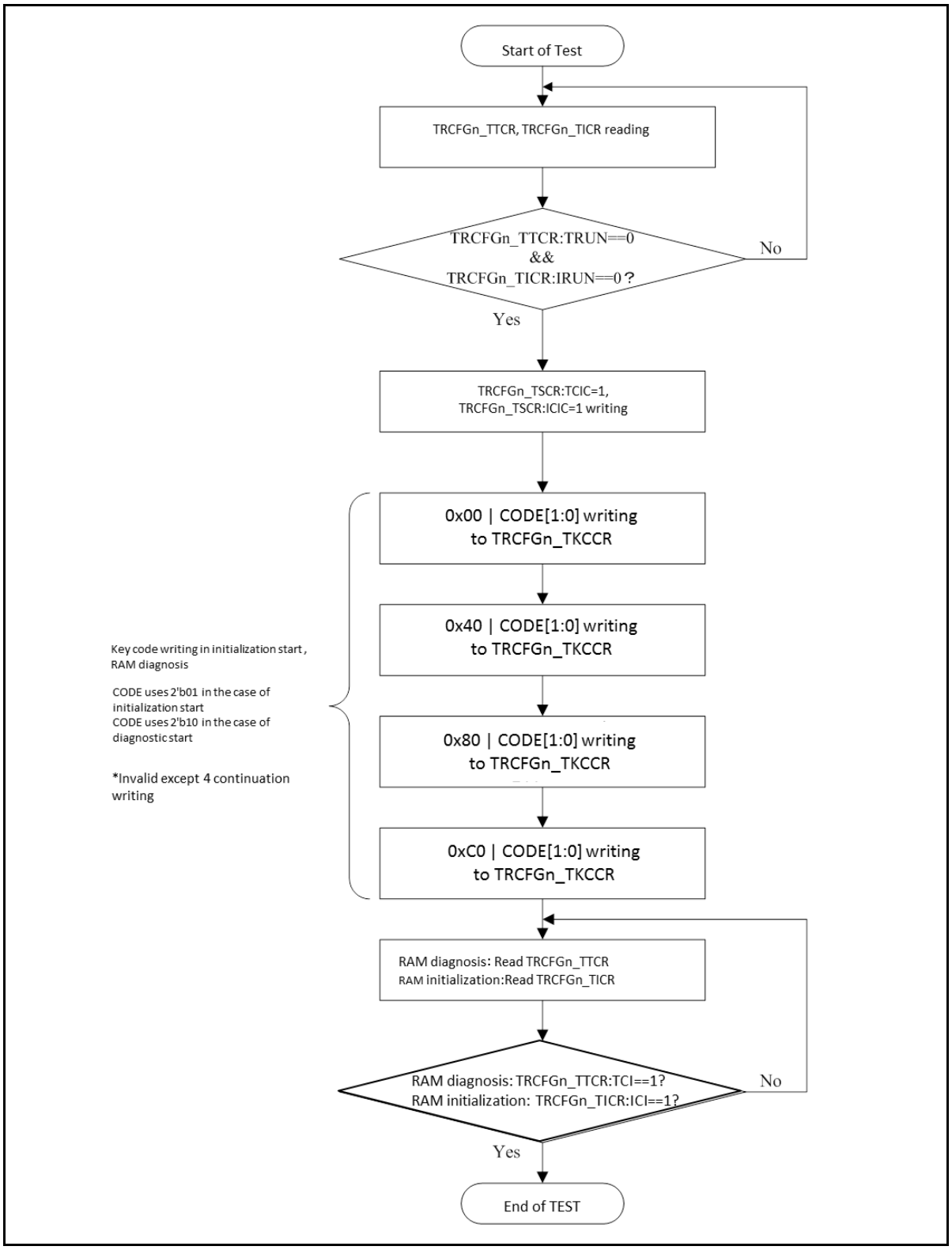
Figure 4-3 Direct Write Access Flow



*1 See Figure 4-1.

*2 See Figure 4-2.

Figure 4-4 RAM Diagnosis and Initialization Start Flow



5. Registers

This section provides a list of the TCRAM interface registers.

When you use the B1TCM port, register setting affects both B0TCM and B1TCM.

Table 5-1 List of TCRAM Interface Registers

Abbreviated Register Name	Register Name	Reference
TRCFGn_TCMCFG0	Configuration register 0	5.1
TRCFGn_TCMCFG1	Configuration register 1	5.2
TRCFGn_TCMUNLOCK	Unlock register	5.3
TRCFGn_ECCDEN	ECC direct access enable register	5.4
TRCFGn_ECCDR_PARITY	ECC direct read parity register	5.5
TRCFGn_ECCDR_DATA0	ECC direct read data register 0	5.6
TRCFGn_ECCDR_DATA1	ECC direct read data register 1	5.7
TRCFGn_ECCDW	ECC direct write register	5.8
TRCFGn_TEAR0	TEST error address register 0	5.9
TRCFGn_TEAR1	TEST error address register 1	5.10
TRCFGn_TEAR2	TEST error address register 2	5.11
TRCFGn_TAEAR	TEST end address register	5.12
TRCFGn_TASAR	TEST start address register	5.13
TRCFGn_TFECR	TEST pseudo error generation control register	5.14
TRCFGn_TICR	TEST initialization function register	5.15
TRCFGn_TTCR	TEST diagnosis function register	5.16
TRCFGn_TSRCR	TEST soft reset generation control register	5.17
TRCFGn_TKCCR	TEST key code control register	5.18
TRCFGn_TSCR	TEST status clear register	5.19

The register memory layout of TCRAM is shown in Table 5-2.

Table 5-2 Register Memory Layout

Offset	Register Name/Initial Value
0x0000_0000	TRCFGn_TCMCFG0 00000000_00000000_00000001_00000000
0x0000_0004	TRCFGn_TCMCFG1 00000000_00000000_00000000_00000000
0x0000_0008	TRCFGn_TCMUNLOCK 00000000_00000000_00000000_00000000
0x0000_000C	Reserved 00000000_00000000_00000000_00000000
0x0000_0010	TRCFGn_ECCDEN 00000000_00000000_00000000_00000000
0x0000_0014	TRCFGn_ECCDR_PARITY 00000000_00000000_00000000_00000000
0x0000_0018	TRCFGn_ECCDR_DATA0 00000000_00000000_00000000_00000000
0x0000_001C	TRCFGn_ECCDR_DATA1 00000000_00000000_00000000_00000000
0x0000_0020	Reserved 00000000_00000000_00000000_00000000
0x0000_0024	TRCFGn_ECCDW 00000000_00000000_00000000_00000000
0x0000_0028	Reserved 00000000_00000000_00000000_00000000
0x0000_002C	Reserved 00000000_00000000_00000000_00000000
0x0000_0030	TRCFGn_TEAR0 00000000_00000000_00000000_00000000
0x0000_0034	TRCFGn_TEAR1 00000000_00000000_00000000_00000000
0x0000_0038	TRCFGn_TEAR2 00000000_00000000_00000000_00000000
0x0000_003C	TRCFGn_TAEAR 01111111_11111111
0x0000_003E	TRCFGn_TASAR 00000000_00000000
0x0000_0040	TRCFGn_TFECR 00000000
0x0000_0041	TRCFGn_TICR 00000000
0x0000_0042	TRCFGn_TTCR 00000000_00001100
0x0000_0044	TRCFGn_TSRCR 00000000

Offset	Register Name/Initial Value
0x0000_0045	TRCFGn_TSCR 00000000
0x0000_0046	Reserved 00000000
0x0000_0047	TRCFGn_TKCCR 00000000
0x0000_0048 to 0x0000_03FF	Reserved

Notes:

- *The [bit25:24] bits in TRCFGn_TCMCFG0 may not operate with the above described default number of waits depending on the TCRAM interface clock frequency, the access time of the mounted SRAM, and other conditions.*
- *If necessary, change the number of waits to the least number of waits that allows operation.*

5.1. TCRAM IF Configuration Register 0 (TRCFGn_TCMCFG0)

This register has 7 bits for ERRECC data for the BTCM port, 2 bits for setting the number of waits, and the LOCKSTATUS bit that indicates the locked state or unlocked state of the TCRAM IF configuration register.

You can perform the write operation for this register only when privilege access is possible and TRCFGn_TCMCFG0:LOCKSTATUS is "0".

Bit	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24
Field	Reserved						DWAIT[1:0]	
R/W Attribute	R0,WX						R/W	
Protection Attribute	WPS							
Initial Value	000000						00	

Bit	bit 23	bit 22	bit 21	bit 20	bit 19	bit 18	bit 17	bit 16
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	WPS							
Initial Value	00000000							

Bit	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
Field	Reserved							LOCKSTATUS
R/W Attribute	R0,WX							R,WX
Protection Attribute	WPS							
Initial Value	0000000							1

Bit	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Field	Reserved	ERRECC[6:0]						
R/W Attribute	R0,WX	R/W						
Protection Attribute	-	WPS						
Initial Value	0	0000000						

[bit31:26] Reserved: Reserved Bits

[bit25:24] DWAIT: Number of Data Wait Bits

DWAIT	Number of Data Wait Bits
During Write Operation	The number of read waits for the BTCM port (0 to 3) is set.
During Read Operation	The number of read waits for the BTCM port that has been set can be read.

[bit23:9] Reserved: Reserved Bits

[bit8] LOCKSTATUS: Lock Status Bit

LOCKSTATUS	Lock Status Bit
During Write Operation	No effect
During Read Operation	0: The TCRAM interface registers are in the unlocked state. 1: The TCRAM interface registers are in the locked state. (Initial value)

[bit7] Reserved: Reserved Bit

[bit6:0] ERRECC: ECC Data Error Insertion Bits

ERRECC	ECC Data Error Insertion Bits
During Write Operation	The data resulting from the XOR operation of these bits and ECC data read from the SRAM is input to the Cortex-R5F BTCM port. In the case of 64-bit read, data whose higher 7 bits and lower 7 bits are XOR-ed with these bits is input to the BTCM port.
During Read Operation	Set value read

5.2. TCRAM IF Configuration Register 1 (TRCFGn_TCMCFG1)

This register is used to insert an error in data read from the TCRAM for the purpose of testing the ECC function of the Cortex-R5F BTCM port.

You can perform the write operation for this register only when privilege access is possible and TRCFGn_TCMCFG0:LOCKSTATUS is "0".

Bit	bit 31 ... bit 0
Field	ERRBIT[31:0]
R/W Attribute	R/W
Protection Attribute	WPS
Initial Value	00000000_00000000_00000000_00000000

[bit31:0] ERRBIT: ECC Error insertion Bits

ERRBIT	ECC Error Insertion Bits
During Write Operation	The data resulting from the XOR operation of these bits and data read from the SRAM is input to the Cortex-R5F BTCM port. In the case of 64-bit read, data whose higher word and lower word are XOR-ed with these bits is input to the BTCM port.
During Read Operation	Set value read

5.3. TCRAM IF Unlock Register (TRCFGn_TCMUNLOCK)

This register locks or unlocks write access to the TCRAM interface registers.

Writing the correct unlock value (0xACC55ECC) to this register enables write access to the registers.

After setting registers, writing the correct lock value (0x5ECCB10C) to this register disables write access to the registers.

You must access this register in units of words, and can access it only with privilege access.

Bit	bit 31	bit 0
Field	UNLOCK[31:0]	
R/W Attribute	R0,W	
Protection Attribute	WP	
Initial Value	00000000_00000000_00000000_00000000	

[bit31:0] UNLOCK: Lock/Unlock Bits

UNLOCK	Lock/Unlock Bit
During Write Operation	Locks and unlocks write accesses to the registers. 0xACC55ECC: TRCFGn_TCMCFG0:LOCKSTATUS becomes "0" and write access to the TCRAM interface registers is enabled. 0x5ECCB10C: TRCFGn_TCMCFG0:LOCKSTATUS becomes "1" and write access to the TCRAM interface registers is disabled. Other values: An error is generated. For details on the register setting procedure, see Figure 4-1.
During Read Operation	"0" is always read.

5.4. TCRAM IF ECC Direct Access Enable Register (TRCFGn_ECCDEN)

This register enables/disables the direct access function for testing the TCRAM ECC area.

Also, it can set the unlock bits for the direct access enable/disable bit for preventing incorrect writing.

You can perform the write operation for this register only when privilege access is possible and TRCFGn_TCMCFG0:LOCKSTATUS is "0".

Bit	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	WPS							
Initial Value	00000000							

Bit	bit 23	bit 22	bit 21	bit 20	bit 19	bit 18	bit 17	bit 16
Field	DEN	DENUNLOCK[6:0]						
R/W Attribute	R,W	R0,W						
Protection Attribute	WPS							
Initial Value	0	0000000						

Bit	bit 15	bit 0
Field	Reserved	
R/W Attribute	R0,WX	
Protection Attribute	WPS	
Initial Value	00000000_00000000	

[bit31:24] Reserved: Reserved Bits

[bit23] DEN: Direct Access Enable

DEN	Direct Access Enable Bit
During Write Operation	0: Disable direct access. 1: Enable direct access when this bit and the DENUNLOCK bits make up the value of 0b10011110. With a value other than the above, direct access is disabled. For the detailed setting procedure of these bits, see Figure 4-2.
During Read Operation	0: Direct access has not been enabled. 1: Direct access has been enabled.

[bit22:16] DENUNLOCK: Direct Access Enable Unlock

DENUNLOCK	Direct Access Enable Unlock Bits
During Write Operation	0b0011110: Enables direct access when the direct access enable bit (DEN) is set to "1" at the same time. When DEN is "0", direct access is disabled. A value other than 0b0011110: The direct access enable bit becomes "0".
During Read Operation	"0" is always read.

[bit15:0] Reserved: Reserved Bits

5.5. TCRAM IF ECC Direct Read Parity Register (TRCFGn_ECCDR_PARITY)

This register includes the ECCDR_PARITY bits, which display ECC data that is read when direct access is enabled.

The displayed ECC data is the value that is read by the last read access.

Bit	bit 31							bit 16
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	-							
Initial Value	00000000_00000000							

Bit	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
Field	Reserved		ECCDR_PARITY[13:8]					
R/W Attribute	R0,WX		R,WX					
Protection Attribute	-							
Initial Value	00		000000					

Bit	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Field	ECCDR_PARITY[7:0]							
R/W Attribute	R,WX							
Protection Attribute	-							
Initial Value	00000000							

[bit31:14] Reserved: Reserved Bits

[bit13:0] ECCDR_PARITY: Direct Read Parity Data

ECCDR_PARITY	Direct Read Parity Data Bits
During Write Operation	Writing data to these bits causes a bus error.
During Read Operation	Displays the ECC data that is last read while the TRCFGn_ECCDEN:DEN bit is set. When the ECC error insertion function is enabled (TRCFGn_TCMCFG0:ERRECC is not "0"), the ECC data modified with TRCFGn_TCMCFG0:ERRECC is captured. This register is updated at B0TCM read or B1TCM read.

5.6. TCRAM IF ECC Direct Read Data Register 0 (TRCFGn_ECCDR_DATA0)

This register displays data [31:0] that is read when direct access is enabled.

The displayed data is the value that is read by the last read access.

Bit	bit 31	bit 0
Field	ECCDR_DATA0[31:0]	
R/W Attribute	R,WX	
Protection Attribute	-	
Initial Value	00000000_00000000_00000000_00000000	

[bit31:0] ECCDR_DATA0: Direct Read Data 0

ECCDR_DATA0	Direct Read Data 0 Bits
During Write Operation	Writing data to these bits causes a bus error.
During Read Operation	Displays the data [31:0] that is last read while the TRCFGn_ECCDEN:DEN bit is set. When the ECC error insertion function is enabled (TRCFGn_TCMCFG1 is not "0"), the data [31:0] modified with TRCFGn_TCMCFG1 is captured. This register is updated at B0TCM read or B1TCM read.

5.7. TCRAM IF ECC Direct Read Data Register 1 (TRCFGn_ECCDR_DATA1)

This register displays data [63:32] that is read when direct access is enabled.

The displayed data is the value that is read by the last read access.

Bit	bit 31	bit 0
Field	ECCDR_DATA1[31:0]	
R/W Attribute	R,WX	
Protection Attribute	-	
Initial Value	00000000_00000000_00000000_00000000	

[bit31:0] ECCDR_DATA1: Direct Read Data 1

ECCDR_DATA1	Direct Read Data 1 Bits
During Write Operation	Writing data to these bits causes a bus error.
During Read Operation	Displays the data [63:32] that is last read while the TRCFGn_ECCDEN:DEN bit is set. When the ECC error insertion function is enabled (TRCFGn_TCMCFG1 is not "0"), the data [63:32] modified with TRCFGn_TCMCFG1 is captured. This register is updated at B0TCM read or B1TCM read.

5.8. TCRAM IF ECC Direct Write Register (TRCFGn_ECCDW)

This register has a direct write access enable bit, direct write access enable unlock bits, and direct write ECC data bits.

To enable the direct write access enable bit, you must write "1" in the DWEN bit and the unlock value in the DWENUNLOCK bits when TRCFGn_ECCDEN.DEN is "1".

You can perform the write operation for this register only when privilege access is possible and TRCFGn_TCMCFG0:LOCKSTATUS is "0".

Bit	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24
Field	DWEN	DWENUNLOCK[6:0]						
R/W Attribute	R,W	R0,W						
Protection Attribute	WPS							
Initial Value	0	0000000						

Bit	bit 23	bit 22	bit 21	bit 20	bit 19	bit 18	bit 17	bit 16
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	WPS							
Initial Value	00000000							

Bit	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
Field	Reserved		ECCDW_PARITY[13:8]					
R/W Attribute	R0,WX		R/W					
Protection Attribute	WPS							
Initial Value	00		000000					

Bit	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Field	ECCDW_PARITY[7:0]							
R/W Attribute	R/W							
Protection Attribute	WPS							
Initial Value	00000000							

[bit31] DWEN: Direct Write Access Enable

DWEN	Direct Write Access Enable Bit
During Write Operation	0: Disable direct write access. 1: Enable direct write access when this bit and the DWENUNLOCK bits make up the value of 0b10110101. With a value other than the above, direct write access is disabled. For the detailed setting procedure of these bits, see Figure 4-3.
During Read Operation	0: Direct write access has not been enabled. 1: Direct write access has been enabled.

[bit30:24] DWENUNLOCK: Direct Write Access Enable Unlock

DWEN UNLOCK	Direct Write Access Enable Unlock Bits
During Write Operation	0b0110101: Enable direct write access (DWEN = 1) when the direct access enable bit (DWEN) is set to "1" at the same time. When DWEN is "0", direct write access is disabled. A value other than 0b0110101: The direct write access enable bit is "0".
During Read Operation	"0" is read.

[bit23:14] Reserved: Reserved Bits

[bit13:0] ECCDW_PARITY: Direct Write Parity Data

ECCDW_ PARITY	Direct Write Parity Data Bits
During Write Operation	Write the value you want to set in the ECC area of the SRAM. When the DWEN bit is "1", the value of these bits is written in the ECC area of the SRAM instead of the ECC data that is calculated by the BTCM. These bits are used for write access of both B0TCM and B1TCM.
During Read Operation	Reads the value set in these bits.

5.9. TCRAM IF TEST Error Address Register 0 (TRCFGn_TEAR0)

This register holds the address for when an error occurs during RAM diagnosis.

Also, this register holds the source flag indicating the diagnosis pattern that has caused this error.

Bit	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24
Field	TER[2:0]			Reserved				
R/W Attribute	R,WX			R0,WX				
Protection Attribute	-							
Initial Value	000			00000				

Bit	bit 23	bit 22	bit 21	bit 20	bit 19	bit 18	bit 17	bit 16
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	-							
Initial Value	00000000							

Bit	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
Field	Reserved	ERR_ADDR[14:8]						
R/W Attribute	R0,WX	R,WX						
Protection Attribute	-							
Initial Value	0	0000000						

Bit	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Field	ERR_ADDR[7:0]							
R/W Attribute	R,WX							
Protection Attribute	-							
Initial Value	00000000							

[bit31:29]TER: Diagnosis Error Source identification

TER	Diagnosis Error Source Identification Bits
During Write Operation	Writing data to these bits causes a bus error.
During Read Operation	Holds the diagnosis pattern when an error occurs during RAM diagnosis. ERR_ADDR[14:0] is effective only when any of these bits is set to "1". 0b001: Error occurred in the march diagnosis 0b010: Error occurred in the checker diagnosis 0b100: Error occurred in the unique diagnosis 0b000: No error occurred A RAM diagnosis start instruction triggers the hardware to initialize (clear to 0b000) these bits.

[bit28:15] Reserved: Reserved Bits

[bit14:0] ERR_ADDR: Error Occurrence Address

ERR_ADDR	Error Occurrence Address Bit
During Write Operation	Writing data to these bits causes a bus error.
During Read Operation	Holds the address when an error occurs during RAM diagnosis. These bits indicate a valid value only when TER is not 0b000. Triggered by the specification of RAM diagnosis start, the hardware initializes these bits (clears them to 0b0000000_00000000).

Notes:

- *When any of the TER bits becomes "1", even if an error occurs during another diagnosis item, the corresponding error source bit does not become "1".*
- *ERR_ADDR is an offset address in units of words (RAM address).*
- *To calculate the absolute address, add {ERR_ADDR << 2} to the base address.*

5.10. TCRAM IF TEST Error Address Register 1 (TRCFGn_TEAR1)

This register holds the address for when an error occurs during RAM diagnosis at an address that is different from the one held in TRCFGn_TEAR0.

Also, this register holds the source flag indicating the diagnosis pattern that has caused this error.

Bit	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24
Field	TER[2:0]			Reserved				
R/W Attribute	R,WX			R0,WX				
Protection Attribute	-							
Initial Value	000			00000				

Bit	bit 23	bit 22	bit 21	bit 20	bit 19	bit 18	bit 17	bit 16
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	-							
Initial Value	00000000							

Bit	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
Field	Reserved	ERR_ADDR[14:8]						
R/W Attribute	R0,WX	R,WX						
Protection Attribute	-							
Initial Value	0	0000000						

Bit	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Field	ERR_ADDR[7:0]							
R/W Attribute	R,WX							
Protection Attribute	-							
Initial Value	00000000							

[bit31:29]TER: Diagnosis Error Source identification

TER	Diagnosis Error Source Identification Bits
During Write Operation	Writing data to these bits causes a bus error.
During Read Operation	Holds the diagnosis pattern when an error occurs during RAM diagnosis. ERR_ADDR[14:0] is effective only when any of these bits is set to "1". 0b001: Error occurred in the march diagnosis 0b010: Error occurred in the checker diagnosis 0b100: Error occurred in the unique diagnosis 0b000: No error occurred A RAM diagnosis start instruction triggers the hardware to initialize (clear to 0b000) these bits. Note: When any of these bits becomes "1", even if an error occurs during another diagnosis item, the corresponding error source bit does not become "1".

[bit28:15] Reserved: Reserved Bits

[bit14:0] ERR_ADDR: Error Occurrence Address

ERR_ADDR	Error Occurrence Address Bit
During Write Operation	Writing data to these bits causes a bus error.
During Read Operation	Holds the address when an error occurs during RAM diagnosis. These bits are effective only when TER is not 0b000. A RAM diagnosis start instruction triggers the hardware to initialize (clear to 0b0000000_00000000) these bits.

Notes:

- *When any of the TER bits becomes "1", even if an error occurs during another diagnosis item, the corresponding error source bit does not become "1".*
- *ERR_ADDR is an offset address in units of words (RAM address).*
- *To calculate the absolute address, add {ERR_ADDR << 2} to the base address.*

5.11. TCRAM IF TEST Error Address Register 2 (TRCFGn_TEAR2)

This register holds the address for when an error occurs during RAM diagnosis at an address that is different from the one held in TRCFGn_TEAR0 and TRCFGn_TEAR1.

Also, this register holds the source flag indicating the diagnosis pattern that has caused this error.

Bit	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24
Field	TER[2:0]			Reserved				
R/W Attribute	R,WX			R0,WX				
Protection Attribute	-							
Initial Value	000			00000				

Bit	bit 23	bit 22	bit 21	bit 20	bit 19	bit 18	bit 17	bit 16
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	-							
Initial Value	00000000							

Bit	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
Field	Reserved	ERR_ADDR[14:8]						
R/W Attribute	R0,WX	R,WX						
Protection Attribute	-							
Initial Value	0	0000000						

Bit	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Field	ERR_ADDR[7:0]							
R/W Attribute	R,WX							
Protection Attribute	-							
Initial Value	00000000							

[bit31:29]TER: Diagnosis Error Source identification

TER	Diagnosis Error Source Identification Bits
During Write Operation	Writing data to these bits causes a bus error.
During Read Operation	Holds the diagnosis pattern when an error occurs during RAM diagnosis. ERR_ADDR[14:0] is effective only when any of these bits is set to "1". 0b001: Error occurred in the march diagnosis 0b010: Error occurred in the checker diagnosis 0b100: Error occurred in the unique diagnosis 0b000: No error occurred A RAM diagnosis start instruction triggers the hardware to initialize (clear to 0b000) these bits. Note: When any of these bits becomes "1", even if an error occurs during another diagnosis item, the corresponding error source bit does not become "1".

[bit28:15] Reserved: Reserved Bits

[bit14:0] ERR_ADDR: Error Occurrence Address

ERR_ADDR	Error Occurrence Address Bit
During Write Operation	Writing data to these bits causes a bus error.
During Read Operation	Holds the address when an error occurs during RAM diagnosis. These bits are effective only when TER is not 0b000. A RAM diagnosis start instruction triggers the hardware to initialize (clear to 0b0b0000000_00000000) these bits.

Notes:

- *When any of the TER bits becomes "1", even if an error occurs during another diagnosis item, the corresponding error source bit does not become "1".*
- *ERR_ADDR is an offset address in units of words (RAM address).*
- *To calculate the absolute address, add {ERR_ADDR << 2} to the base address.*

5.12. TCRAM IF TEST End Address Register (TRCFGn_TAEAR)

This register specifies the end address for performing RAM diagnosis or initialization.

You can perform the write operation for this register only when privilege access is possible and TRCFGn_TCMCFG0:LOCKSTATUS is "0".

Bit	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
Field	Reserved	EADDR[14:8]						
R/W Attribute	R0,WX	R/W						
Protection Attribute	WPS							
Initial Value	0	1111111						

Bit	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Field	EADDR[7:0]							
R/W Attribute	R/W							
Protection Attribute	WPS							
Initial Value	11111111							

[bit15] Reserved: Reserved Bit

[bit14:0] EADDR: RAM Diagnosis End Address

EADDR	RAM Diagnosis End Address Bits
During Write Operation	You can set an address at which RAM diagnosis or initialization operation ends. A word address of the RAM is set.
During Read Operation	You can read the address at which specified RAM diagnosis or initialization operation ends.

Notes:

- *EADDR is an offset in word units (RAM address).*
- *Obtain the absolute address by adding {EADDR << 2} to the base address.*

5.13. TCRAM IF TEST Start Address Register (TRCFGn_TASAR)

This register specifies the start address for performing RAM diagnosis or initialization.

You can perform the write operation for this register only when privilege access is possible and TRCFGn_TCMCFG0:LOCKSTATUS is "0".

Bit	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
Field	Reserved	SADDR[14:8]						
R/W Attribute	R0,WX	R/W						
Protection Attribute	WPS							
Initial Value	0	0000000						

Bit	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Field	SADDR[7:0]							
R/W Attribute	R/W							
Protection Attribute	WPS							
Initial Value	00000000							

[bit15] Reserved: Reserved Bit

[bit14:0] SADDR: RAM Diagnosis Start Address

SADDR	RAM Diagnosis Start Address Bits
During Write Operation	You can set an address at which RAM diagnosis or initialization operation starts. A word address of the RAM is set.
During Read Operation	You can read the address at which specified RAM diagnosis or initialization operation starts.

Notes:

- *SADDR is an offset address in units of words (RAM address).*
- *To calculate the absolute address, add {SADDR << 2} to the base address.*

5.14. TCRAM IF TEST Pseudo Error Generation Control Register (TRCFGn_TFECR)

This register specifies the RAM diagnosis operation in which a pseudo error is to be generated.

You can perform the write operation for this register only when privilege access is possible and TRCFGn_TCMCFG0:LOCKSTATUS is "0".

Bit	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Field	Reserved				FERR	ETYP[2:0]		
R/W Attribute	R0,WX				R/W	R/W		
Protection Attribute	WPS							
Initial Value	0000				0	000		

[bit7:4] Reserved: Reserved Bits

[bit3]FERR: RAM Diagnosis Pseudo-Error Generation Enable

FERR	RAM Diagnosis Pseudo-Error Generation Enable Bit
During Write Operation	0: Disable generation of pseudo-errors. (Normal operation) 1: Enable generation of pseudo-errors. Erroneous data is deliberately written in accordance with the ETYP setting.
During Read Operation	The enable/disable setting for generation of pseudo-errors can be read.

[bit2:0] ETYP: Pseudo-Error Generation Processing Specification

ETYP	Pseudo-Error Generation Processing Specification Bits
During Write Operation	0bxx1: Generate pseudo-errors during march diagnosis. 0bx1x: Generate pseudo-errors during checker diagnosis. 0b1xx: Generate pseudo-errors during unique diagnosis. 0b000: Generate no pseudo error.
During Read Operation	This register can read the setting value for the diagnosis in which the pseudo-error is generated.

5.15. TCRAM IF TEST Initialization Function Register (TRCFGn_TICR)

This register specifies contents of RAM initialization, and holds the result and state of the initialization.

You can perform the write operation for this register only when privilege access is possible and TRCFGn_TCMCFG0:LOCKSTATUS is "0".

Bit	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Field	Reserved				ICIE	ICI	ITYP	IRUN
R/W Attribute	R0,WX				R/W	R,WX	R/W	R,WX
Protection Attribute	WPS							
Initial Value	0000				0	0	0	0

[bit7:4] Reserved: Reserved Bits

[bit3] ICIE: RAM Initialization End Source Interrupt Enable

ICIE	RAM Initialization End Source Interrupt Enable Bit
During Write Operation	0: Disable interrupts triggered by RAM initialization end. 1: Enable interrupts triggered by RAM initialization end.
During Read Operation	You can read the setting for enabling or disabling the initialization end source interrupt.

[bit2] ICI: RAM Initialization Completion

ICI	RAM Initialization End Bit
During Write Operation	No effect
During Read Operation	0: Initialization has not been completed or has not been started. 1: RAM initialization has been completed. This bit is not set for forcible stop by the key code.

[bit1] ITYP: RAM Initialization Contents Specification

ITYP	RAM Initialization Contents Specification Bit
During Write Operation	0: Initialize with all "0's". 1: Initialize with all "1's".
During Read Operation	You can read the setting of the initialization contents specification.

[bit0] IRUN: RAM Initialization Operation Status

IRUN	RAM Initialization Operation Status Bit
During Write Operation	No effect
During Read Operation	0: Initialization is not in progress. 1: Initialization is in progress.

5.16. TCRAM IF TEST Diagnosis Function Register (TRCFGn_TTCR)

This register specifies the contents of RAM diagnosis, and holds the result and state of the diagnosis.

You can perform the write operation for this register only when privilege access is possible and TRCFGn_TCMCFG0:LOCKSTATUS is "0".

Bit	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
Field	Reserved						TSTAT	OVFLW
R/W Attribute	R0,WX						R,WX	R,WX
Protection Attribute	WPS							
Initial Value	000000						0	0

Bit	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Field	TEIE	TEI	TCIE	TCI	TTYP			TRUN
R/W Attribute	R/W	R,WX	R/W	R,WX	R/W			R,WX
Protection Attribute	WPS							
Initial Value	0	0	0	0	110			0

[bit15:10] Reserved: Reserved Bits

[bit9] TSTAT: RAM Diagnosis Error Detection

TSTAT	RAM Diagnosis Error Detection Bit
During Write Operation	No effect
During Read Operation	0: No error has been detected in RAM diagnosis. 1: An error has been detected in RAM diagnosis. A RAM diagnosis start instruction triggers the hardware to initialize (clear to "0") this bit.

[bit8] OVFLW: RAM Diagnosis Error Overflow

OVFLW	RAM Diagnosis Error Overflow Bit
During Write Operation	No effect
During Read Operation	0: RAM diagnosis errors have occurred at 3 addresses or less. 1: RAM diagnosis errors have occurred at 4 addresses or more. A RAM diagnosis start instruction triggers the hardware to initialize (clear to "0") this bit.

[bit7] TEIE: Enable Error Occurrence Interrupt during Diagnosis

TEIE	Diagnosis-Time Error Generation Interrupt Enable Bit
During Write Operation	0: Disable interrupts triggered by RAM diagnosis errors. 1: Enable interrupts triggered by RAM diagnosis errors. When TRCFGn_TTCR:TEI is "1", "H" is output to the interrupt signal.
During Read Operation	You can read the setting for enabling or disabling the error interrupts during diagnosis.

[bit6] TEI: Diagnosis-Time Error Generation

TEI	Diagnosis-Time Error Generation Bit
During Write Operation	No effect
During Read Operation	0: TRCFGn_TTCR:TSTAT = 0 when RAM diagnosis ends. 1: TRCFGn_TTCR:TSTAT = 1 when RAM diagnosis ends.

[bit5] TCIE: Diagnosis End Source Interrupt Enable

TCIE	Diagnosis End Source Interrupt Enable Bit
During Write Operation	0: Disable interrupts triggered by RAM diagnosis end. 1: Enable interrupts triggered by RAM diagnosis end. When TRCFGn_TTCR:TCI is "1", "H" is output to the interrupt signal.
During Read Operation	You can read the setting for enabling or disabling the diagnosis end source interrupt.

[bit4] TCI: Diagnosis End

TCI	Diagnosis End Bit
During Write Operation	No effect
During Read Operation	0: The diagnosis has not completed or has been stopped. 1: RAM diagnosis has been completed. This bit is not set for forcible stop by the key code.

[bit3:1] TTYP: RAM Diagnosis Contents Specification

TTYP	RAM Diagnosis Contents Specification Bits
During Write Operation	0bxx1: Perform march diagnosis. 0bxx0: Do not perform march diagnosis. 0bx1x: Perform checker diagnosis. 0bx0x: Do not perform checker diagnosis. 0b1xx: Perform unique diagnosis. 0b0xx: Do not perform unique diagnosis. RAM diagnoses are performed in the following order. Unique diagnosis (The address itself is used as the data.) Checker diagnosis March diagnosis (all "0" and then all "1")
During Read Operation	You can read the set value for the types to be executed in RAM diagnosis.

[bit0] TRUN: RAM Diagnosis Operation Status

TRUN	RAM Diagnosis Operation Status Bit
During Write Operation	No effect
During Read Operation	0: RAM diagnosis is not in progress. 1: RAM diagnosis is in progress.

Note:

- *Be sure to make the setting for this register before starting RAM diagnosis.*

5.17. TCRAM IF TEST Soft Reset Generation Control Register (TRCFGn_TSRCR)

This register resets the entire circuit related to RAM diagnosis except this register itself.

You can perform the write operation for this register only when privilege access is possible and TRCFGn_TCMCFG0:LOCKSTATUS is "0".

Bit	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Field	SRST	Reserved						
R/W Attribute	R0,W	R0,WX						
Protection Attribute	WPS							
Initial Value	0	0000000						

[bit7] SRST: Software Reset

SRST	Software Reset Bit
During Write Operation	0: No effect 1: Reset all circuits that are related to RAM diagnosis except for this register.
During Read Operation	"0" is always read.

[bit6:0] Reserved: Reserved Bits

5.18. TCRAM IF TEST Key Code Control Register (TRCFGn_TKCCR)

This register is used to start or to forcibly stop RAM diagnosis or initialization.

You can perform the write operation for this register only when privilege access is possible and TRCFGn_TCMCFG0:LOCKSTATUS is "0".

Bit	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Field	KEY[1:0]		Reserved				CODE[1:0]	
R/W Attribute	R0,W		R0,WX				R/W	
Protection Attribute	WPS							
Initial Value	00		0000				00	

[bit7:6] KEY: Key Code Control

KEY	Key Code Control Bits
During Write Operation	These bits control the key code. The operation specified in the CODE bits is performed by writing in the order of 0b00, 0b01, 0b10, and 0b11. In addition, if you have accessed another RAMTEST (TRCFGn_TEAR0, TRCFGn_TEAR1, TRCFGn_TEAR2, TRCFGn_TAEAR, TRCFGn_TASAR, TRCFGn_TFECR, TRCFGn_TICR, TRCFGn_TTCR, TRCFGn_TSRCR, TRCFGn_TSCR) register or have performed an operation other than that described above (read operation or a series of write operations other than the above) during key code operation, start the operation over again. For the detailed setting procedure of these bits, see Figure 4-4.
During Read Operation	"0" is always read.

[bit5:2] Reserved: Reserved Bits

[bit1:0] CODE: Operation Specification

CODE	Operation Specification Bits
During Write Operation	0b00: Forcibly terminate 0b01: Activate initialization 0b10: Activate diagnosis 0b11: Setting prohibited
During Read Operation	The set value of operation specification can be read.

Notes:

- *During KEY bits operation, set the same value in the CODE bits.*
- *Setting the value 0b11 to the CODE bits is prohibited*

5.19. TCRAM IF TEST Status Clear Register (TRCFGn_TSCR)

This register is used to clear the RAM diagnosis and initialization status bits.

You can perform the write operation for this register only when privilege access is possible and TRCFGn_TCMCFG0:LOCKSTATUS is "0".

Bit	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Field	Reserved	TEIC	Reserved	TCIC	Reserved	ICIC	Reserved	
R/W Attribute	R0,WX	R0,W	R0,WX	R0,W	R0,WX	R0,W	R0,WX	
Protection Attribute	WPS							
Initial Value	0	0	0	0	0	0	00	

[bit7] Reserved: Reserved Bit

[bit6] TEIC: Diagnosis-Time Error Generation Clear

TEIC	Diagnosis-Time Error Generation Clear Bit
During Write Operation	0: No effect 1: Clear the TRCFGn_TTCR:TEI bit.
During Read Operation	"0" is always read.

[bit5] Reserved: Reserved Bit

[bit4] TCIC: Diagnosis End Clear

TCIC	Diagnosis End Clear Bit
During Write Operation	0: No effect 1: Clear the TRCFGn_TTCR:TCI bit.
During Read Operation	"0" is always read.

[bit3] Reserved: Reserved Bit

[bit2] ICIC: RAM Initialization End Clear

ICIC	RAM Initialization End Clear Bit
During Write Operation	0: No effect 1: Clear the TRCFGn_TICR:ICI bit.
During Read Operation	"0" is always read.

[bit1:0] Reserved: Reserved Bits

CHAPTER 17: TCFLASH

This chapter explains the functions and operations of TCFLASH.

1. **Overview**
2. **Configuration**
3. **Explanation of Operation**
4. **Setting Procedure Examples**
5. **Registers**
6. **Others**
7. **Support for Dual Port Flash Macro**

1. Overview

This section explains the overview of TCFLASH.

Overview of TCFLASH

- TCFLASH is a flash memory to mainly store programs. If this LSI is in user mode, it is mapped to 2 regions, the TCM region and the AXI region. If accessed via the TCM region, TCFLASH is treated as an L1 memory on the ARM(R) architecture. Therefore, non-cacheable and low-latency access is possible. If TCFLASH is accessed via the AXI region, it is treated as an L2 memory in the context of ARM architecture.
- TCFLASH can be extended to up to 16 MB, logically. However, only 8 MB of it can be accessed via the TCM region. If it is accessed via the AXI region, 16 MB of the entire region can be accessed.
- Programming or erasing in TCFLASH is performed if a program sequence is sent when it is accessed via the AXI region. Programming or erasing in TCFLASH cannot be performed when it is accessed via the TCM region.
- Programming with access via the AXI region has the highest priority in TCFLASH. With the default setting, reading priority is toggled between access via the AXI region and access via the TCM region every 16 accesses.
- In TCFLASH, set priority to allow reading with access via the TCM region. However, reading has lower priority than programming.
- Data can be read from the CPU in units of 8 bits/16 bits/32 bits/64 bits.
- If ECC is enabled, programming from the CPU is possible only in 16/32-bit. If ECC is disabled, programming from the CPU is possible in 8-/16-/32-bit.
- Non-aligned read for the access size is supported. However, non-aligned programming for the access size is not supported. If non-aligned programing for the access size is detected, TCFLASH responds by issuing a bus error.
- Wait cycle count that is inserted during data access can be specified.
- Flash memory reads data with a width (product specification) times as wide as 64 bits. If there are consecutive addresses, 64-bit data can be used with a frequency (product specification) times as high as the flash clock frequency.
- The TCM has (product specification) buffers to reduce penalties for non-consecutive addresses resulting from branch instructions or the like.
- ECC is supported with the same calculation formula as that for the ARM Cortex(TM)-R5F core, realizing 1-bit error correction and 2-bit error detection.
- The ECC check for reading via the TCM region is performed with logic in the ARM Cortex-R5F core.
- ECC logic check can be performed by inserting an error. Error injection is valid for both reading via the TCM region and reading via the AXI region.
- Register setting values are protected using a protection key.
- Programming and erasing are possible only when access is in privileged mode.
- The flash security function is implemented. For details, see the security specifications.
- Bus error response is sent when access is to the reserved area or detect 1-bit ECC error via TCM region(*1).
  *1:Partially ECC supported. For more detail, See " CHAPTER17, 6. Others, 1-bit ECC error is detected".
- The flash security function enables the function for protection from programming and erasing in sector units.
- Parallel programmer provides the function for reading, programing, and erasing data in flash memory.
- ECC support for the AXI ports for FLASH 1-bit error correction and 2-bit error detection (SEC-DED).

- ECC support for the TCM ports for FLASH 1-bit error detection(*1) and 2-bit error detection (SED-DED) at default setting.
  *1 For more detail, See "CHAPTER17, 6. Others, 1-bit ECC error is detected"

2. Configuration

This section explains the configuration within TCFLASH.

2.1. Block Diagrams

- There is 1 TCFLASH per 1 Cortex-R5F core.
- When Cortex-R5F core accesses via the TCM region, TCFLASH is handled as a read-only L1 memory. On the other hand, when Cortex-R5F core accesses via the AXI region, TCFLASH is handled as an L2 memory that can both write and read.

Figure 2-1 Position of TCFLASH

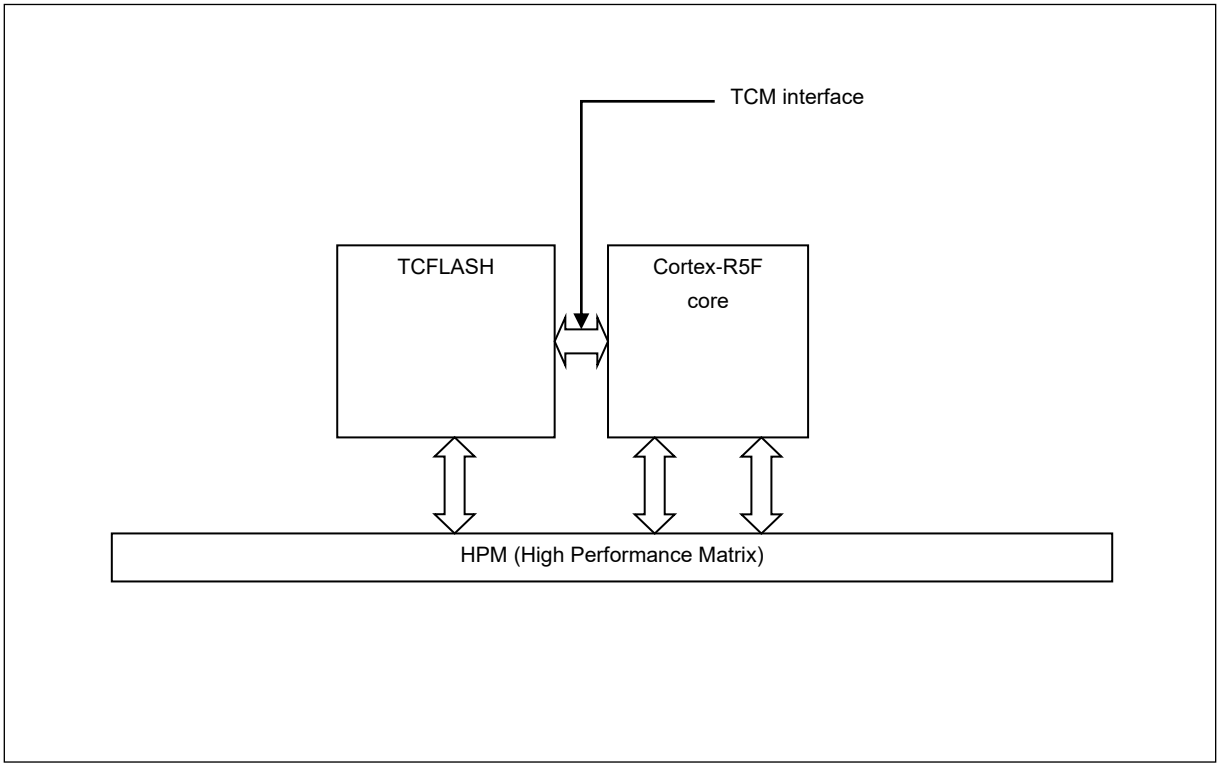
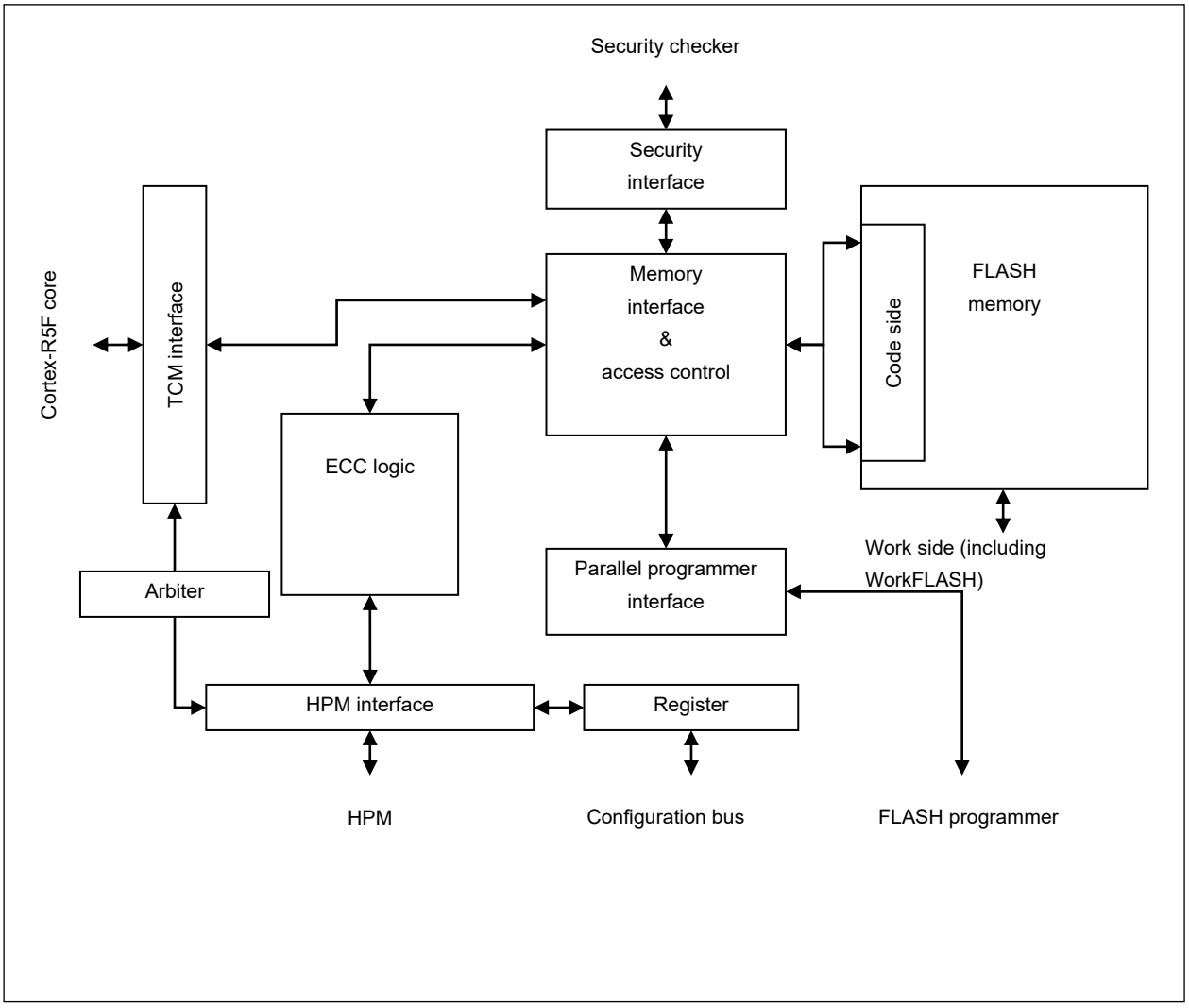


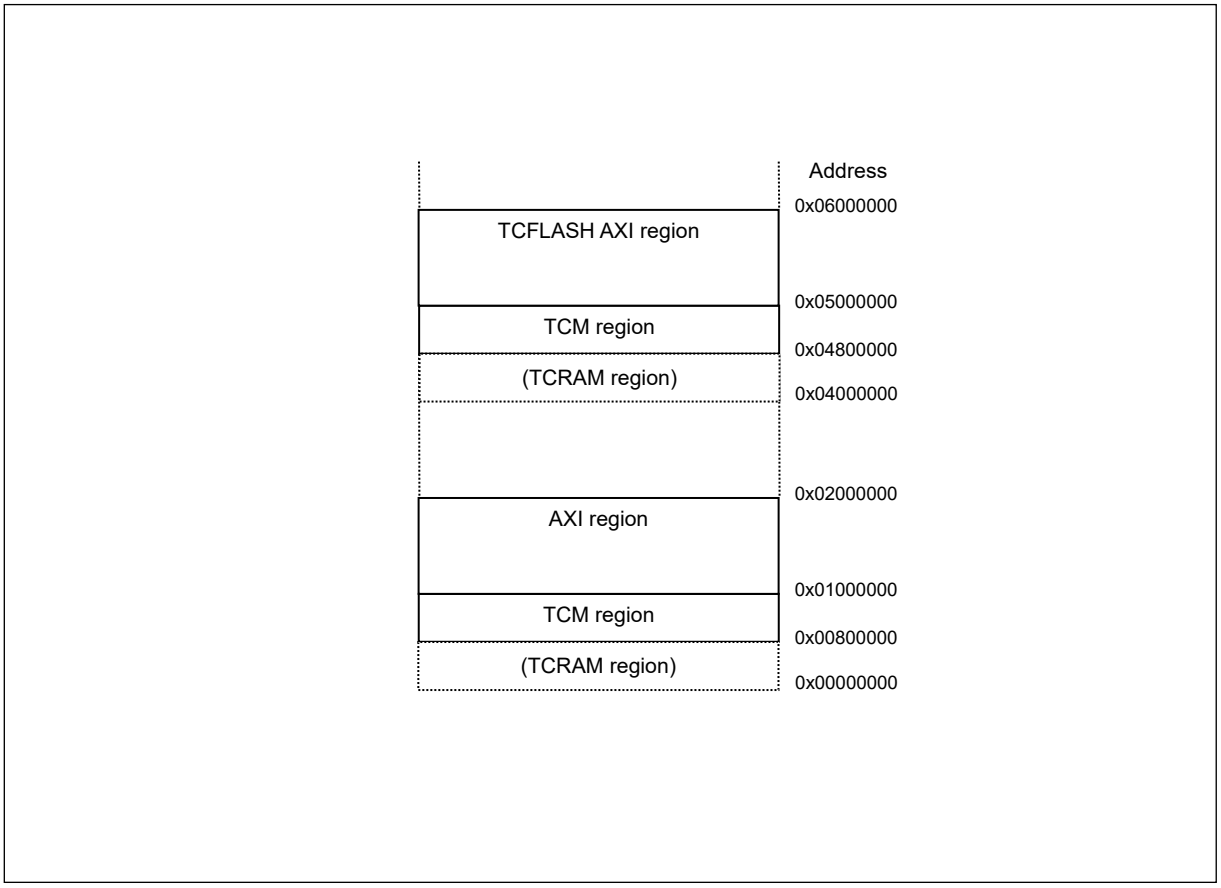
Figure 2-2 Configuration of TCFLASH



2.2. Address/Sector Map of TCFLASH

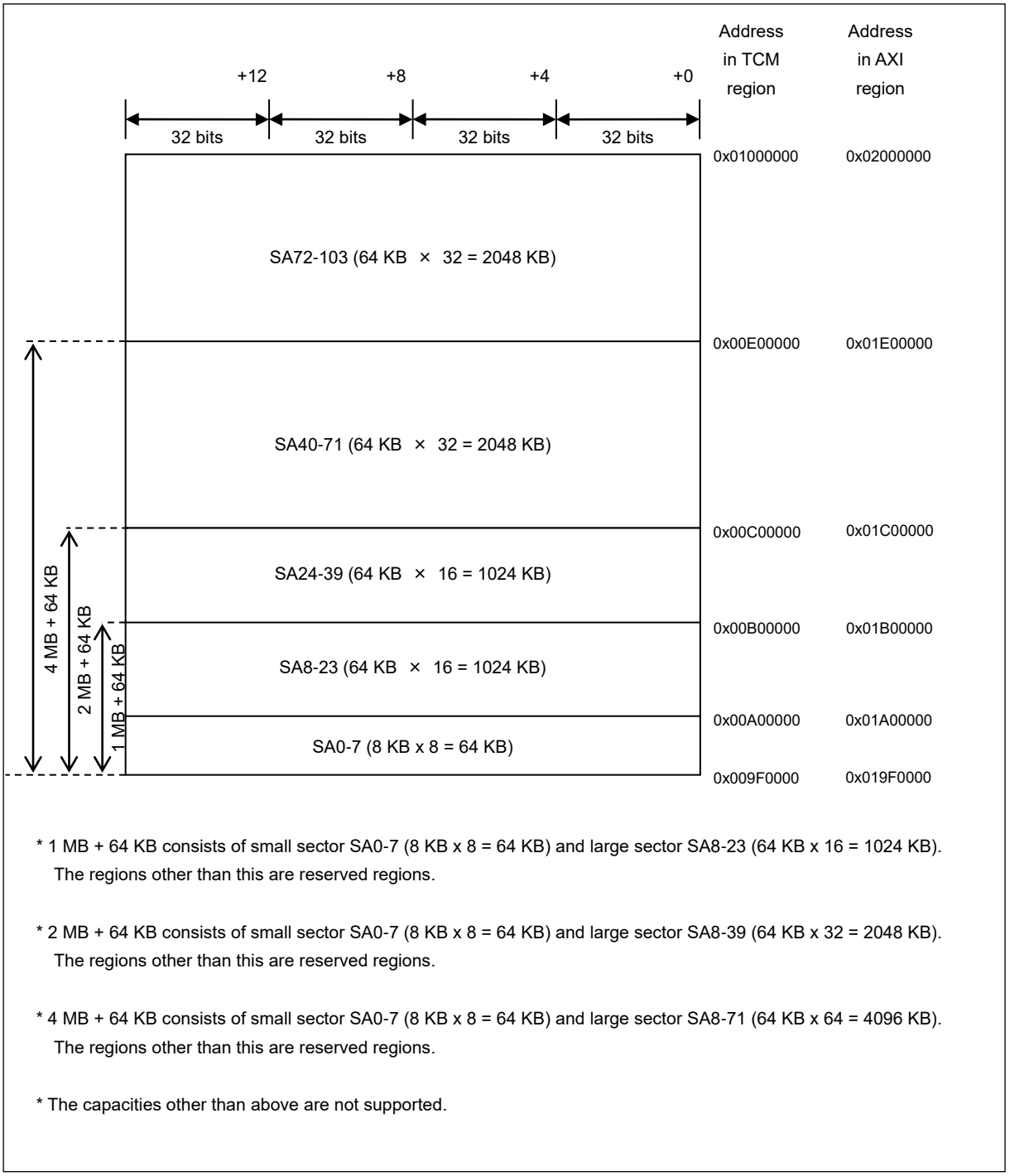
A TCFLASH exists for CPU core and is placed on the memory map by the program as shown in Figure 2-3.

Figure 2-3 Position of TCFLASH on the Memory Map



The size of the TCM region for each TCFLASH is 8 MB. The size of the AXI region is 16 MB. The sector placement for the respective region is shown in Figure 2-4 and Figure 2-5.

Figure 2-4 Address/Sector Map of TCFLASH



* 1 MB + 64 KB consists of small sector SA0-7 (8 KB x 8 = 64 KB) and large sector SA8-23 (64 KB x 16 = 1024 KB). The regions other than this are reserved regions.

* 2 MB + 64 KB consists of small sector SA0-7 (8 KB x 8 = 64 KB) and large sector SA8-39 (64 KB x 32 = 2048 KB). The regions other than this are reserved regions.

* 4 MB + 64 KB consists of small sector SA0-7 (8 KB x 8 = 64 KB) and large sector SA8-71 (64 KB x 64 = 4096 KB). The regions other than this are reserved regions.

* The capacities other than above are not supported.

Figure 2-5 TCFLASH Address Map for Multiple Flash Memories

	Address Within TCM Region	Address With AXI Region
	0x00C00000	0x01C00000
SA39	0x00BF0000	0x01BF0000
SA38	0x00BE0000	0x01BE0000
SA37	0x00BD0000	0x01BD0000
SA36	0x00BC0000	0x01BC0000
SA35	0x00BB0000	0x01BB0000
SA34	0x00BA0000	0x01BA0000
SA33	0x00B90000	0x01B90000
SA32	0x00B80000	0x01B80000
SA31	0x00B70000	0x01B70000
SA30	0x00B60000	0x01B60000
SA29	0x00B50000	0x01B50000
SA28	0x00B40000	0x01B40000
SA27	0x00B30000	0x01B30000
SA26	0x00B20000	0x01B20000
SA25	0x00B10000	0x01B10000
SA24	0x00B00000	0x01B00000
SA23	0x00AF0000	0x01AF0000
SA22	0x00AE0000	0x01AE0000
SA21	0x00AD0000	0x01AD0000
SA20	0x00AC0000	0x01AC0000
SA19	0x00AB0000	0x01AB0000
SA18	0x00AA0000	0x01AA0000
SA17	0x00A90000	0x01A90000
SA16	0x00A80000	0x01A80000
SA15	0x00A70000	0x01A70000
SA14	0x00A60000	0x01A60000
SA13	0x00A50000	0x01A50000
SA12	0x00A40000	0x01A40000
SA11	0x00A30000	0x01A30000
SA10	0x00A20000	0x01A20000
SA9	0x00A10000	0x01A10000
SA8	0x00A00000	0x01A00000
SA7	0x009FE000	0x019FE000
SA6	0x009FC000	0x019FC000
SA5	0x009FA000	0x019FA000
SA4	0x009F8000	0x019F8000
SA3	0x009F6000	0x019F6000
SA2	0x009F4000	0x019F4000
SA1	0x009F2000	0x019F2000
SA0	0x009F0000	0x019F0000

	Address Within TCM Region	Address With AXI Region
	0x00E00000	0x01E00000
SA71	0x00DF0000	0x01DF0000
SA70	0x00DE0000	0x01DE0000
SA69	0x00DD0000	0x01DD0000
SA68	0x00DC0000	0x01DC0000
SA67	0x00DB0000	0x01DB0000
SA66	0x00DA0000	0x01DA0000
SA65	0x00D90000	0x01D90000
SA64	0x00D80000	0x01D80000
SA63	0x00D70000	0x01D70000
SA62	0x00D60000	0x01D60000
SA61	0x00D50000	0x01D50000
SA60	0x00D40000	0x01D40000
SA59	0x00D30000	0x01D30000
SA58	0x00D20000	0x01D20000
SA57	0x00D10000	0x01D10000
SA56	0x00D00000	0x01D00000
SA55	0x00CF0000	0x01CF0000
SA54	0x00CE0000	0x01CE0000
SA53	0x00CD0000	0x01CD0000
SA52	0x00CC0000	0x01CC0000
SA51	0x00cB0000	0x01CB0000
SA50	0x00cA0000	0x01CA0000
SA49	0x00C90000	0x01C90000
SA48	0x00C80000	0x01C80000
SA47	0x00C70000	0x01C70000
SA46	0x00C60000	0x01C60000
SA45	0x00C50000	0x01C50000
SA44	0x00C40000	0x01C40000
SA43	0x00C30000	0x01C30000
SA42	0x00C20000	0x01C20000
SA41	0x00C10000	0x01C10000
SA40	0x00C00000	0x01C00000

Figure 2-6 Sector Placement of TCFLASH in Parallel Programmer Mode

	Address
	0xC00000
SA39	0xBF0000
SA38	0xBE0000
SA37	0xBD0000
SA36	0xBC0000
SA35	0xBB0000
SA34	0xBA0000
SA33	0xB90000
SA32	0xB80000
SA31	0xB70000
SA30	0xB60000
SA29	0xB50000
SA28	0xB40000
SA27	0xB30000
SA26	0xB20000
SA25	0xB10000
SA24	0xB00000
SA23	0xAF0000
SA22	0xAE0000
SA21	0xAD0000
SA20	0xAC0000
SA19	0xAB0000
SA18	0xAA0000
SA17	0xA90000
SA16	0xA80000
SA15	0xA70000
SA14	0xA60000
SA13	0xA50000
SA12	0xA40000
SA11	0xA30000
SA10	0xA20000
SA9	0xA10000
SA8	0xA00000
SA7	0x9FE000
SA6	0x9FC000
SA5	0x9FA000
SA4	0x9F8000
SA3	0x9F6000
SA2	0x9F4000
SA1	0x9F2000
SA0	0x9F0000

	Address
	0xE00000
SA71	0xDF0000
SA70	0xDE0000
SA69	0xDD0000
SA68	0xDC0000
SA67	0xDB0000
SA66	0xDA0000
SA65	0xD90000
SA64	0xD80000
SA63	0xD70000
SA62	0xD60000
SA61	0xD50000
SA60	0xD40000
SA59	0xD30000
SA58	0xD20000
SA57	0xD10000
SA56	0xD00000
SA55	0xCF0000
SA54	0xCE0000
SA53	0xCD0000
SA52	0xCC0000
SA51	0xcB0000
SA50	0xcA0000
SA49	0xC90000
SA48	0xC80000
SA47	0xC70000
SA46	0xC60000
SA45	0xC50000
SA44	0xC40000
SA43	0xC30000
SA42	0xC20000
SA41	0xC10000
SA40	0xC00000

3. Explanation of Operation

This section explains the operation of TCFLASH.

3.1. Operation Mode of TCFLASH

If this LSI is in user mode, the CPU or other bus master can access TCFLASH. A Cortex-R5F core can access TCFLASH that is connected to itself, via the TCM interface or via AXI. It can access TCFLASH that is connected to another Cortex-R5F core, via AXI Bus masters (non-Cortex-R5F core) such as DMAC can access TCFLASH via AXI.

In user mode, programming or erasing in TCFLASH via AXI interface can be performed. However, programming or erasing via the TCM interface cannot be performed. Programming and erasing are performed by executing a program access sequence to start an automatic algorithm. ECC is generated in programming by using the TCFLASH interface. (*1)

In the serial writer mode or the parallel writer mode, the Flash can be read/programmed/erased according to security setting.*1: Do not perform the read operation via the TCM interface if you have run the program access sequence via AXI. Otherwise, a bus error occurs and an undefined value is output.

3.2. Programming and Erasing

If this LSI is in user mode, programming and erasing in TCFLASH are performed by writing the programming access sequence in the flash memory via the AXI region.

- For the programming access sequence, see "3.5.1 Command Sequence." For the programming procedure, see "4.4 Programming Procedure." For macro erase in non-user mode, restriction of the erasing order exists. For details on the sequence, see the security specifications.
- Overwriting of the same data bit is prohibited. (Even overwriting of 1 -> 0, 0 -> 0, 1 -> 1 is also prohibited.)
- If ECC is enabled, only programming in 16-/32-bit is possible. If ECC is disabled, programming in 8-/16 -/32-bit is possible.
- After erase operation is completed, all the values of the flash memory cells in the target sectors are "1".

If this LSI is in parallel programmer mode or serial programmer mode, and if the flash security is turned on, programming or erasing in TCFLASH is restricted as described in the security specifications. To remove the restriction, you must set the flash security to "off". For the procedure for this setting, see CHAPTER: Security.

3.3. TCM Buffer

This section explains the TCM buffer which allows efficient access from the TCM.

When TCFLASH is accessed from the TCM, the TCM buffers allow the access time to be reduced. The previous flash memory read data and its address are stored in the TCM buffer. If data accessed later matches the address of TCM buffer, the data is read directly from TCM buffer. Compared to FLASH macro, access with high throughput and low latency is possible. (product specification) buffers are assembled.

A single buffer consists of the following.

- Buffer enable bit (TCFCFG_TCMBUF0 to 15 :BVALID)
- Buffered address (TCFCFG_TCMBUF0 to 15 :BUFADD[31:0])
- Buffered data (TCFCFG_TCMBUF0 to 15 :BUFDATA[127:0])
- Buffered parity (TCFCFG_TCMBUF0 to 15 :ECC_PARITY[15:0])

Only one region can be set. The following can be set:

- Enable/Disable setting (TCFCFG_BRAT :RGEN)
- Buffer number of region (TCFCFG_BRCFG :REGION_END_BUF[15:0]): 0 to 15 can be set.
- Range of bufferable sectors (TCFCFG_BRCFG :SSEC[7:0]/ESEC[7:0]): SA0 to SA71 can be set.
- Selection policy for words to be buffered (TCFCFG_BRAT :AM)

TCFCFG_TCMBUF	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Buffer enable	BVALID	BVALID	BVALID	BVALID	BVALID	BVALID	BVALID	BVALID	BVALID	BVALID	BVALID	BVALID	BVALID	BVALID	BVALID	BVALID
Buffered address	BUFADD[31:0]	BUFADD[31:0]	BUFADD[31:0]	BUFADD[31:0]	BUFADD[31:0]	BUFADD[31:0]	BUFADD[31:0]	BUFADD[31:0]	BUFADD[31:0]	BUFADD[31:0]	BUFADD[31:0]	BUFADD[31:0]	BUFADD[31:0]	BUFADD[31:0]	BUFADD[31:0]	BUFADD[31:0]
Buffered Data	BUFDATA[127:0]	BUFDATA[127:0]	BUFDATA[127:0]	BUFDATA[127:0]	BUFDATA[127:0]	BUFDATA[127:0]	BUFDATA[127:0]	BUFDATA[127:0]	BUFDATA[127:0]	BUFDATA[127:0]	BUFDATA[127:0]	BUFDATA[127:0]	BUFDATA[127:0]	BUFDATA[127:0]	BUFDATA[127:0]	BUFDATA[127:0]
Buffered parity data	ECCDR_PARITY[15:0]	ECCDR_PARITY[15:0]	ECCDR_PARITY[15:0]	ECCDR_PARITY[15:0]	ECCDR_PARITY[15:0]	ECCDR_PARITY[15:0]	ECCDR_PARITY[15:0]	ECCDR_PARITY[15:0]	ECCDR_PARITY[15:0]	ECCDR_PARITY[15:0]	ECCDR_PARITY[15:0]	ECCDR_PARITY[15:0]	ECCDR_PARITY[15:0]	ECCDR_PARITY[15:0]	ECCDR_PARITY[15:0]	ECCDR_PARITY[15:0]

3.4. Arbitration of TCM and AXI

If access via the AXI region conflicts with access via the TCM region, arbitration is performed as follows.

- If the WE bit in the TCFCFG_FCFGR register is "1", programming TCFLASH is enabled. In this case, programming via the AXI region has the highest priority for processing.
- The read operation via the AXI region has equal priority with that via the TCM region, both of which are processed accordingly. TCFLASH alternates the priority of the AXI region and the TCM region after each 16 read operations. For example, if the first 16 read operations via the AXI region are processed on a priority basis, the next 16 read operations via the TCM region are processed on a priority basis.
- In either case, priority does not shift during burst access.

3.5. Automatic Algorithms

Programming and erasing can be performed by sending the program access sequence to flash memory to start an automatic algorithm. The available commands in the automatic algorithm include reset, read, program, macro erase, and sector erase. For the sector erase command, it is possible to control the suspension and resumption of its execution.

3.5.1. Command Sequence

To start an automatic algorithm, perform operation to program given data to a given address once to six times consecutively, depending on the command type.

Table 3-1 Command Sequence List

Command	Program Count	1st cycle		2nd cycle		3rd cycle		4th cycle		5th cycle		6th cycle		7th cycle	
		Address	Data	Address	Data	Address	Data	Address	Data	Address	Data	Address	Data	Address	Data
Reset	1	CA0	0xF0	-	-	-	-	-	-	-	-	-	-		
Read	1	RA	RD	-	-	-	-	-	-	-	-	-	-		
Program	4	CA0	0xAA	CA1	0x55	CA0	0xA0	PA	PD	-	-	-	-		
Program (64-bit)	5	CA0	0xAA	CA1	0x55	CA0	0xAC	PA	PD64_0	any	PD64_1	-	-		
Macro Erase	6	CA0	0xAA	CA1	0x55	CA0	0x80	CA0	0xAA	CA1	0x55	CA0	0x10		
Sector Erase	6	CA0	0xAA	CA1	0x55	CA0	0x80	CA0	0xAA	CA1	0x55	SA	0x30		
Multiple Sectors Erase	6 + x	CA0	0xAA	CA1	0x55	CA0	0x80	CA0	0xAA	CA1	0x55	SA0	0xE0	SA1	0x30
Sector Erase Suspend	1	SA	0xB0	-	-	-	-	-	-	-	-	-	-		
Sector Erase Resume	1	SA	0x30	-	-	-	-	-	-	-	-	-	-		

Table 3-2 Address in the Command

Operation Mode	Code (Table 3-1)	Offset in AXI Region
User mode	CA0	0x?????AA8
	CA1	0x?????554

- Specify the values shown in Table 3-2 for "CA0" and "CA1" in Table 3-1.
- Specify the value of the lower 32 bits of 64-bit program data for "PD64_0" in Table 3-1. Specify the value of the upper 32 bits of 64-bit program data for "PD64_1".
- Except for the program data which is represented as "PD PD64_0 PD64_1" in the table, the upper 24 bits (bit31 to bit8) of data to be programmed to start an automatic algorithm are ignored.
- "?????" of Table 3-2 is an arbitrary value that specifies the address range used by the flash memory on which to execute the command. For the memory map, see "2.2 Address/Sector Map of TCFLASH."

- The address PA given upon the 4th write cycle is the address in which program data PD is written.
- The programming address PA for programming must be a value which is programming size aligned.
- The address SA given upon the 6th program cycle of the sector erase command and 1st cycle of sector erase suspend/resume commnad, and the address SA0, SA1 given upon 6th, 7th cycle of multi sector erase command indicates the address of the sector to be erased. SA is specified in the same format as PA.
- Flash memory is reset and transitions to read mode if an invalid address or data set is programmed as a command or if commands are programmed in the wrong order.
- The read operation from flash memory is possible even when the command sequence is being programmed. The automatic algorithm starts upon completion of the last cycle of the program sequence.
- Do not specify the sector address of which sector protection is enabled for any of CA0, CA1, SA, SA0, SA1, and PA.

- Do not change the setting of the sector protection register for TCFLASH (TCFCFG0_C*SWP) while writing the command sequence.
- Flash memory must not be read-accessed until the status register READ changes to "1" after the sector erase suspend command is issued.
- The 4 MB model has 2 FLASH macros(SA0 to 39 and SA40 to 71), and care should be taken regarding the following.
  - Macro erasure should be issued separately for 2 FLASH macros.
  - Multiple sector erasure cannot be executed over 2 FLASH macro (SA39 and SA40).

Parallel Programmer Mode

- Commands supported in parallel programmer mode differ from those in the command sequence stated above. For details on how to read, program, and erase in parallel programmer mode, see the Parallel Programmer Specifications.

3.6. Automatic Algorithm Execution Status

The hardware sequence flag which had been implemented so far is not included. The detailed status of flash memory during the execution of an automatic algorithm can be confirmed by reading the TCFCFG_FSTAT : TCFLASH status register to check the value of each bit. If flash memory is read during the execution of automatic algorithm, it causes a bus error response.

Table 3-3 Bit Assignment of TCFLASH Status Register

Bit No.	7	6	5	4	3	2	1	0
Flag Name	CERS	PGMS	ESPS	ERSEC	SERS	READ	HANG	RDY

The TCFCFG_FSTAT : TCFLASH status register consists of the following bits: CERS, PGMS, ESPS, ERSEC, SERS, READ, HANG, and RDY. Each bit indicates the following status:

Table 3-4 Values of Register TCFLASH Status Register

State		CERS	PGMS	ESPS	ERSEC	SERS	READ	HANG	RDY
Reset		0	0	0	0	0	0	0	0
Command		0	0	0	0	0	1	0	1
Program		0	1	0	0	0	0	0	0
Macro erase		1	0	0	0	0	0	0	0
Sector erase		0	0	0	0	1	0	0	0
Interruption of sector erasure	Sector read during erasure	0	0	1	1	1	1	0	1
	Non-target sector read during erasure	0	0	1	0	1	1	0	1
Hang up 1	Program	0	1	0	0	0	0	1	0
	Macro erase	1	0	0	0	0	0	1	0
	Sector erase	0	0	0	0	1	0	1	0

- CERS (Chip ERase Status) bit

  This bit indicates the macro erase status.

  Any command can be received until macro erase is completed.

- PGMS (ProGraM Status) bit

  This bit indicates the programming status.

- ESPS (Erase SusPend Status) bit

  This bit indicates the sector erase suspend status.

  Reading and programming are possible for sectors other than erase targets.

  Reading and programming are not possible sectors that are erase targets.

  Sector erase resume command resumes sector erase.

- ERSEC (ERase Suspend SECtor status) bit

  This bit indicates the status in which a target sector read is being executed while sector erase is suspended.

  At this time, the data read from flash memory is not the correct data.

■ SERS (Sector ERase Status) bit

This bit indicates the sector erase status.

During sector erase, sector erase suspend commands can be received.

■ READ (READ rdy) bit

This bit indicates the readable status of TCFLASH.

■ HANG (HANG up 1 status) bit

This bit indicates the Hang up 1 status.

Hang up 1 status indicates that 2 of the following events occurred.

1. When an attempt is made to overwrite with value 1 an address to which the value of 0 has been written
2. When programming, macro erase, and sector erase do not complete within the time limit

In the Hang up 1 status, the status can be recovered by writing command reset.

■ RDY (RDY) bit

This bit indicates the programmable/erasable status of TCFLASH/WorkFLASH.

When WorkFLASH is programmable/erasable, this bit becomes "0".

The TCFLASH status can be confirmed with READ and RDY bits.

1. READ=0, RDY=0: Reset (TCFLASH: Waiting for ready)
2. READ=0, RDY=1: Reading TCFLASH (TCFLASH: Waiting for ready)
3. READ=1, RDY=1: NOP (TCFLASH: Readable/programmable/erasable)
4. READ=1, RDY=0: WorkFLASH programming/erasing (TCFLASH: Readable only)

3.7. Inserting Wait Cycles

You can insert wait cycles when flash memory is accessed by setting the FAWC bit in the TCFCFG_FCFGR register.

If the system clock frequency is low, you can even set the FAWC bit to "0". The wait cycle count set for the FAWC bit is applied when flash memory is accessed for both reading and programming. The value set for the FAWC bit is valid until the setting is changed. The value of the FAWC bit is "3" after reset.

3.8. ECC Generation and Check

TCFLASH is capable of 1-bit error detection and correction and 2-bit error detection by adding the 8-bit error check code (ECC: Error Check Code) per 64 bits.

The ECC logic used in TCFLASH performs ECC generation during write-access and syndrome check during read-access in the same way as with 64-bit ECC in the ARM Cortex-R5F core.

During write-access, 8-bit ECC is generated for the 64-bit data, which is programmed along with the data. For the procedure for adding ECC to program data, see "4.4 Programming Procedure."

During read-access via the AXI space, a syndrome is calculated to determine any of "No error," "1-bit error," or "2-bit-or-more error." If the result is "2-bit-or-more error," error correction cannot be performed. If the content of flash memory is erased, an error is never detected. During read-access via the TCM space, TCFLASH does not detect or correct any errors. TCFLASH directly sends data and the ECC bit that are read from flash memory to the Cortex-R5F core. Error detection/correction for the data and ECC bit read via the TCM space is performed in the Cortex-R5F core.

The ECC movement in TCM port is based on ECC setting inside the CPU.

The contents described from 3.8.1 to 3.8.3 are the operations performed in TCFLASH when it is read-accessed via the AXI space.

To test the check function, the ECC logic has the function to insert an error into the data and ECC that are read from flash memory. Error injection is performed during read-access both via the TCM space and via the AXI space.

ECC is enabled/disabled by setting the ECCOFF bit in the TCFCFG_FECCCTRL register.

3.8.1. ECC Generation

When programming, the 8-bit check bit CB[7:0] is generated from the 64-bit program data D[63:0] according to the calculation formulas given in Table 3-5 ECC Calculation Formulas. The generated check bit is XORed with 0xf3 and then programmed in flash memory.

Table 3-5 ECC Calculation Formulas

		Data Bit																														
		63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
CB0	XOR	1	0	1	1	0	1	0	0	1	1	0	1	0	0	0	1	1	0	1	1	0	1	0	0	1	1	0	1	0	0	0	1
CB1	XOR	0	0	0	1	0	1	0	1	0	1	0	1	0	1	1	1	0	0	0	1	0	1	0	1	0	1	0	1	0	1	1	1
CB2	XNOR	1	0	1	0	0	1	1	0	1	0	0	1	1	0	0	1	1	0	1	0	0	1	1	0	1	0	0	1	1	0	0	1
CB3	XNOR	0	0	1	1	1	0	0	0	1	1	1	0	0	0	1	1	0	0	1	1	1	0	0	0	1	1	1	0	0	0	1	1
CB4	XOR	1	1	0	0	0	0	0	0	1	1	1	1	1	1	0	0	1	1	0	0	0	0	0	0	1	1	1	1	1	1	0	0
CB5	XOR	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
CB6	XOR	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
CB7	XOR	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0

		Data Bit																														
		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CB0	XOR	0	1	0	0	1	0	1	1	0	0	1	0	1	1	1	0	0	1	0	0	1	0	1	1	0	0	1	0	1	1	1	0
CB1	XOR	0	0	0	1	0	1	0	1	0	1	0	1	0	1	1	1	0	0	0	1	0	1	0	1	0	1	0	1	0	1	1	1
CB2	XNOR	1	0	1	0	0	1	1	0	1	0	0	1	1	0	0	1	1	0	1	0	0	1	1	0	1	0	0	1	1	0	0	1
CB3	XNOR	0	0	1	1	1	0	0	0	1	1	1	0	0	0	1	1	0	0	1	1	1	0	0	0	1	1	1	0	0	0	1	1
CB4	XOR	1	1	0	0	0	0	0	0	1	1	1	1	1	1	0	0	1	1	0	0	0	0	0	0	1	1	1	1	1	1	0	0
CB5	XOR	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
CB6	XOR	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
CB7	XOR	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

For check bits, all data bits of which value is "1" in the table above are calculated using XOR or XNOR. Example:

CB0 = D[63]^D[61]^D[60]^D[58]^D[55]^D[54]^D[52]^D[48]^D[47]^D[45]^D[44]^D[42]^D[39]
^D[38]^D[36]^D[32]^D[30]^D[27]^D[25]^D[24]^D[21]^D[19]^D[18]^D[17]^D[14]^D[11]^D[9]
^D[8]^D[5]^D[3]^D[2]^D[1] ;

CB2 = ~ (D[63]^D[61]^D[58]^D[57]^D[55]^D[52]^D[51]^D[48]^D[47]^D[45]^D[42]^D[41]^D[39]
^D[36]^D[35]^D[32]^D[31]^D[29]^D[26]^D[25]^D[23]^D[20]^D[19]^D[16]^D[15]^D[13]^D[10]
^D[9]^D[7]^D[4]^D[3]^D[0]) ;

3.8.2. Calculating a Syndrome

When read-accessing, the 8-bit check bits CB[7:0] are calculated from the data D[63:0] which is read from flash memory in accordance with the calculation formulas given in Table 3-5. The calculated check bits are used together with the check bits EDOR[7:0] that are read from the flash memory to generate a syndrome S[7:0] in accordance with the calculation formulas given in Table 3-6.

Table 3-6 Syndrome Calculation Formulas

Bit	Calculation Formula
S[7]	~ (CB[7] ^ EDOR[7])
S[6]	~ (CB[6] ^ EDOR[6])
S[5]	~(CB[5] ^ EDOR[5])
S[4]	~(CB[4] ^ EDOR[4])
S[3]	(CB[3] ^ EDOR[3])
S[2]	(CB[2] ^ EDOR[2])
S[1]	~(CB[1] ^ EDOR[1])
S[0]	~ (CB[0] ^ EDOR[0])

3.8.3. Detecting Errors

Based on the value of the calculated syndrome S[7:0], a decision is made: "no error detected," "1-bit error detected," "2-bit error detected," and "error of 3 bits or greater detected." Table 3-7 shows the relationship between syndrome values and decision results. The meanings of symbols used in the table are as follows:

- "+" : No error is detected.
- "C[n] (0≤n≤7) " : 1-bit error is detected. The value of the check bit CB[n] is erroneous.
- "[m] (0≤m≤63) : 1-bit error is detected. The value of the data bit D[m] is erroneous.
- "T" : 2-bit error is detected. It cannot be corrected.
- "M" : 3-bit-or-more error is detected. It cannot be corrected.

The 1-bit errors can be detected and corrected in all cases. The 2-bit errors can be detected in all cases, but cannot be corrected. As indicated in Table 3-7, errors of 3 bits or greater may be detected in some cases. However, they cannot be detected in all cases. Errors of 3 bits or greater cannot be corrected, either.

Table 3-7 Meanings of Syndrome Values

		Syndrome[7:4]															
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
Syndrome[3:0]	**0**	+	C4	C5	T	C6	T	T	62	C7	T	T	46	T	M	M	T
	1	C0	T	T	14	T	M	M	T	T	M	M	T	M	T	T	30
	2	C1	T	T	M	T	34	56	T	T	50	40	T	M	T	T	M
	3	T	18	8	T	M	T	T	M	M	T	T	M	T	2	24	T
	4	C2	T	T	15	T	35	57	T	T	51	41	T	M	T	T	31
	5	T	19	9	T	M	T	T	63	M	T	T	47	T	3	25	T
	6	T	20	10	T	M	T	T	M	M	T	T	M	T	4	26	T
	7	M	T	T	M	T	36	58	T	T	52	42	T	M	T	T	M
	8	C3	T	T	M	T	37	59	T	T	53	43	T	M	T	T	M
	9	T	21	11	T	M	T	T	M	M	T	T	M	T	5	27	T
	A	T	22	12	T	33	T	T	M	49	T	T	M	T	6	28	T
	B	17	T	T	M	T	38	60	T	T	54	44	T	1	T	T	M
	C	T	23	13	T	M	T	T	M	M	T	T	M	T	7	29	T
	D	M	T	T	M	T	39	61	T	T	55	45	T	M	T	T	M
	E	16	T	T	M	T	M	M	T	T	M	M	T	0	T	T	M
	F	T	M	M	T	32	T	T	M	48	T	T	M	T	M	M	T

3.9. Interrupt

TCFLASH can generate an interrupt request in the following case.

- When a 1-bit error (correctable) is detected with the ECC logic
  The SECINT bit in the TCFCFG_FSECIR register is "1". In this case, if the SECIE bit of the same register is "1", TCFLASH generates an interrupt request.

3.10. Bus Error Response

TCFLASH generates a bus error response in the following cases.

- When an uncorrectable error is detected. (A 2-bit-or-more error is detected with the ECC check.)
- When a program or erase operation in the non-privileged state is attempted
- When a program or erase operation is attempted while the WE bit in the TCFCFG_FCFGR register is "0"
- When programming is attempted from the TCM region
- When 8-bit programming is attempted while ECC is enabled
- When programming an address that is not boundary aligned with the program size is attempted
- When read-access is attempted via the TCM/AXI interface during execution of the automatic algorithm
- When access is attempted to the reserved area in the TCM space, reserved area in the AXI space, or reserved area in the register space
- When a change of setting is attempted by programming registers in the non-privileged state
- When programming of the 2nd and subsequent cycle in the TCFCFG_FECCCTRL register is attempted
- When a change of settings is attempted in a manner deviating from the unlock sequence. Specifically, the following cases are applicable.
  - When 2 consecutive write-accesses are attempted for the TCFCFG_FCPROTKEY
  - When programming of the wrong protection key is attempted for the TCFCFG_FCPROTKEY
  - When write-access of the protected register is attempted by a bus master other than the one that unlocked programming of that protected register
- When write-access to the read-only register is attempted
- When programing/erasing is attempted for the sectors protected with security
- When software reset is issued for Flash of which program is not enabled
- When TCM buffer register (TCFCFG_TCMBUF is programmed in a way that does not meet the following conditions:
  - TCFLASH TCM Buffer Region Attribute Register:TCFCFG_BRAT:AM=10
  - 32/64-bit access

4. Setting Procedure Examples

This section explains the TCFLASH setting procedures.

4.1. Setting Wait Cycle Count

If the operating frequency of the system is higher than the maximum operating frequency of the flash memory, it is necessary to insert wait cycles when accessing the flash memory, by setting an appropriate value in FAWC[1:0] in the TCFCFG_FCFGR register.

You can calculate the value to set for FAWC[1:0] according to the following formula.

FAWC[1:0] = CEILING (System operating frequency/Maximum operating frequency of flash memory) - 1

CEILING() in the above formula represents a function that rounds up the decimal part of the argument to make it an integer.

The maximum operating frequency of flash memory built in this LSI is (product specification) MHz. Therefore, if the system operating frequency is (product specification) MHz, for example, the setting value for FAWC[1:0] should be "01" as calculated using the above formula.

4.2. Reading Flash Memory State/Transition to Reset State

Program the read/reset command sequence in flash memory. If execution of the automatic algorithm terminates abnormally (due to a timeout, for example), the state of flash memory can be initialized. Because the state of flash memory is read/reset immediately after this LSI is reset, you do not need to program the read/reset command sequence in flash memory.

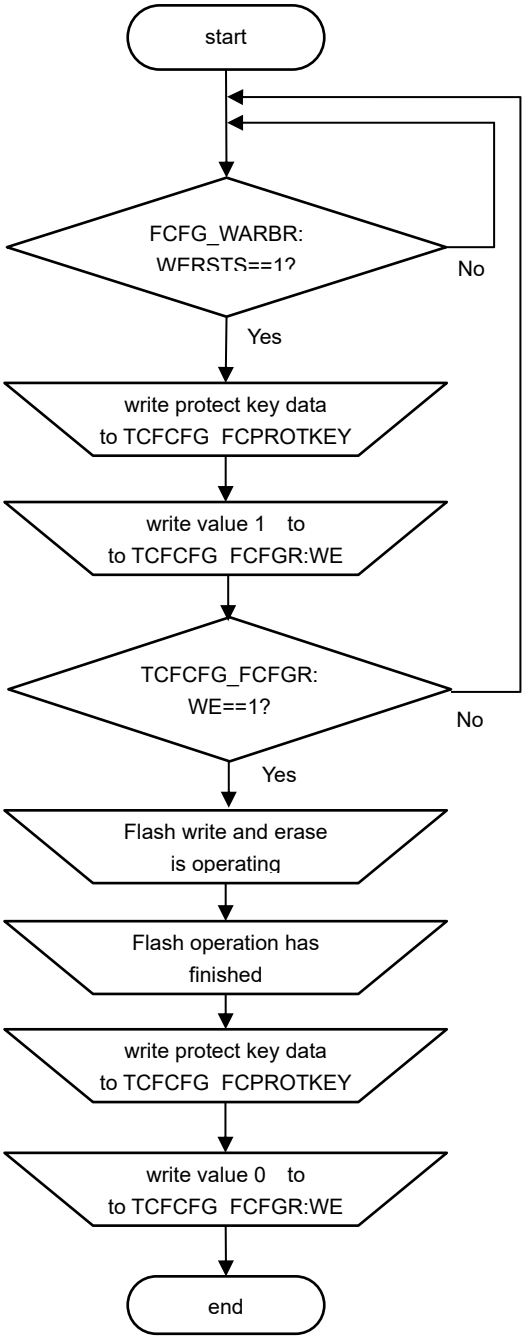
Successful program or erase operation by flash memory is not affected by programming of the read/reset command sequence. In addition, a suspended sector erase cannot be canceled by programming the read/reset command sequence.

4.3. Enable Programming

To program/erase data in TCFLASH, program the enable bit of TCFLASH has to be set. In addition, TCFLASH and WorkFLASH cannot be programmed/erased at the same time. Therefore, the flash interface (TCFLASH, WorkFLASH) includes the arbitration function.

To program from TCFLASH, the program enable bit has to be enabled using the following procedure.

Figure 4-1 Procedure to Set Flash Memory Program Enable Bit



Write enable setting can be done using program enable release (FCFG_WARBR:WERINT) interrupt, instead of the monitoring program enable release status (FCFG_WARBR:WERSTS) bit.

To confirm the completion of Flash program/erase, check that the Programming/erasing ready bit (TCFCFG_FSTAT:RDY) has become “1”.

When programming/erasing is not required, to allow programming/erasing to Flash using WorkFLASH interface or SHE, set program enable (TCFCFG_FCFGR:WE) bit to "0".

When program enable is requested simultaneously, programming is enabled in accordance with the priority below:

1. SHE
2. WorkFLASH
3. TCFLASH

4.4. Programming Procedure

To program data in flash memory, write the program command sequence to flash memory. The destination address must be aligned with the data size to be programmed.

Overwriting of the same data bit is prohibited. (Even overwriting 1 -> 0, 0 -> 0, 1 -> 1 is prohibited as well.)

The cell value of flash memory cannot be changed from "0" to "1" by programming. Therefore, if "1" is written in the cell in which "0" has already been written, flash memory is indicated as being in the hung up state.

During execution of program operation, all the command sequences additionally written in flash memory are ignored.

If a bus error response is received while the command sequence is being written, issue software reset by writing "1" bit to the SWFRST bit to initialize the macro (for flash memory to be ready to accept a new command rather than erasing the macro).

If this LSI is reset during execution of program operation or flash memory is reset because "1" is written to the SWFRST bit in the TCFCFG_FCFGR register, contents of the cell programmed after reset cannot be assured.

When ECC is disabled and programming is performed in 8-/16-/32-bit, the data is directly written in the address specified in flash memory. When ECC is enabled, follow the procedure below to program in 16-/32-bit.

Flash security function blocks the programming of data in protected sectors. For details, see the security specifications.

- When ECC is enabled and programming is performed in 32-bit (64-bit program command is used.) (See Table 3-1.)

  1st time: Command sequence for 64-bit program

    1-1: Address = CA0, Data = 0xAA

    1-2: Address＝CA1, Data = 0x55

    1-3: Address＝CA0, Data = 0xAC

    1-4: Address＝PA(Lower:0**), Data = PD64_0([31:0])

    1-5: Address＝PA(Lower:1**), Data = PD64_1([63:32])

- When ECC is enabled and program is performed in 32-bit (Program command is used.) (See Table 3-1.)

  1st time: Command sequence for 32-bit program

    1-1: Address＝CA0, Data = 0xAA

    1-2: Address＝CA1, Data = 0x55

    1-3: Address＝CA0, Data = 0xA0

    1-4: Address＝PA(Lower:0**), Data = PD32_0([31:0])

  2nd time: Command sequence for 32-bit program

    2-1: Address＝CA0, Data = 0xAA

    2-2: Address＝CA1, Data = 0x55

2-3: Address＝CA0, Data = 0xA0

2-4: Address＝PA (Lower:1**), Data = PD32_1([63:32])

- When ECC is enabled and program is performed in 16-bit (Program command is used.) (See Table 3-1.)

1st time: Command sequence for 16-bit program

1-1: Address＝CA0, Data = 0xAA

1-2: Address＝CA1, Data = 0x55

1-3: Address＝CA0, Data = 0xA0

1-4: Address＝PA (Lower:00*), Data = PD16_0([15:0])

2nd time: Command sequence for 16-bit program

2-1: Address＝CA0, Data = 0xAA

2-2: Address＝CA1, Data = 0x55

2-3: Address＝CA0, Data = 0xA0

2-4: Address＝PA (Lower:01*), Data = PD16_1([31:16])

3rd time: Command sequence for 16-bit program

3-1: Address＝CA0, Data = 0xAA

3-2: Address＝CA1, Data = 0x55

3-3: Address＝CA0, Data = 0xA0

3-4: Address＝PA(Lower:10*), Data = PD16_2([47:32])

4th time: Command sequence for 16-bit program

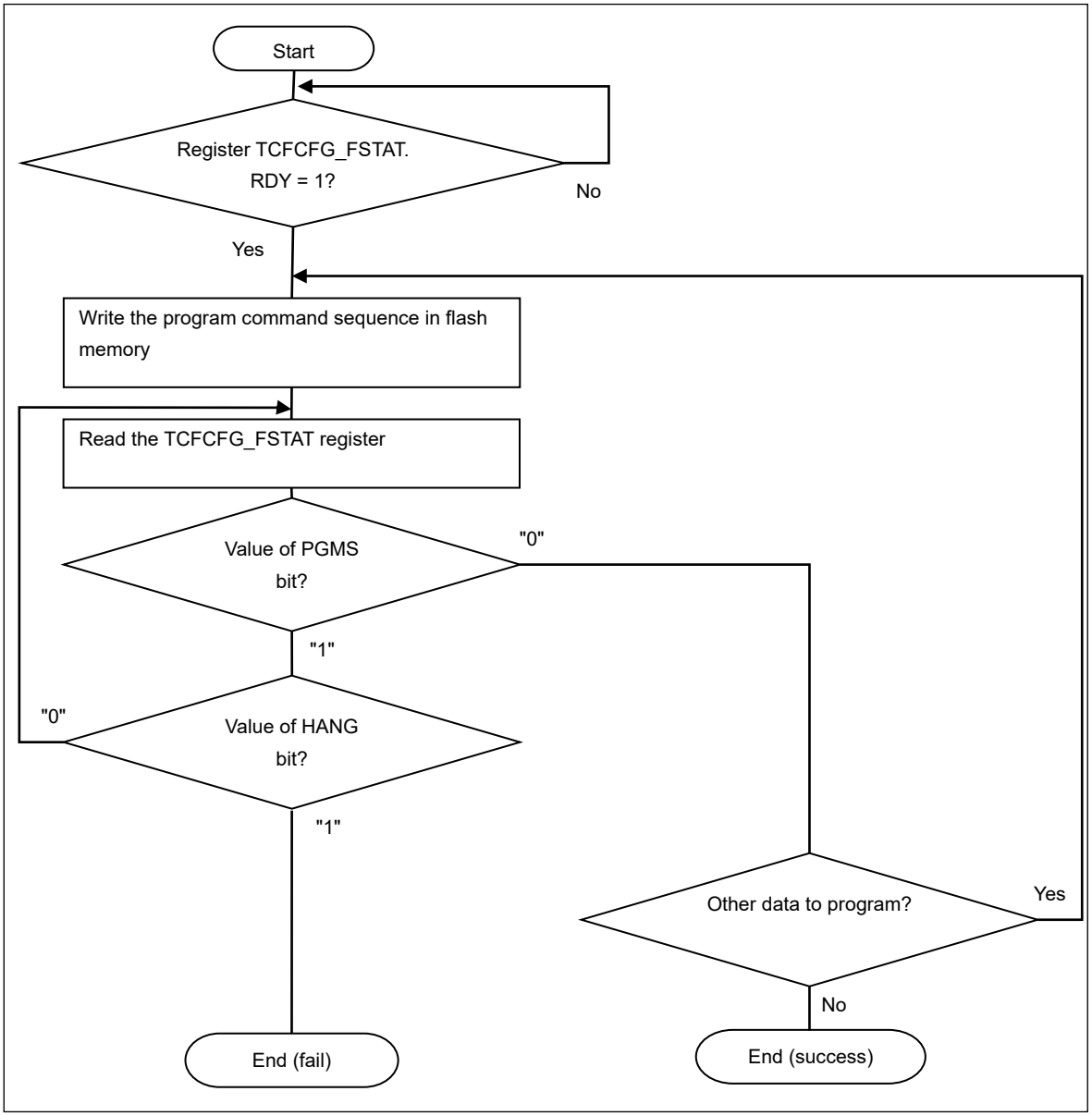
4-1: Address＝CA0, Data = 0xAA

4-2: Address＝CA1, Data = 0x55

4-3: Address＝CA0, Data = 0xA0

4-4: Address＝PA (Lower:11*), Data = PD16_3([63:48])

Figure 4-2 Programming Procedure



The time it takes for HANG to become “1” is not equal to the Flash max writing time defined in datasheet. For example, when max writing time is defined as 400us in datasheet, it does not indicate HANG is set 400us after writing the data. HANG is designed so that the time for it to set is relatively longer than the max writing time defined in the datasheet. Hence, do not design the program that expects HANG=1 after the max writing time defined in datasheet (the same thing applies to sector erasing).

4.5. Macro Erasure Procedure

Write the macro erase command sequence in flash memory. If the last cycle of programming completes in a total of 6 cycles of programming that comprise the sequence, flash memory starts the macro erase operation. After the macro erase operation completes, the values of all the memory cells in the targeted flash memory are "1".

Since 4 MB model has 2 FLASH macros, macro erasure has to be performed for each FLASH macro.

During execution of the macro erase operation, the write operation of any command sequence is ignored in flash memory.

If this LSI is reset during execution of the macro erasure operation or flash memory is reset because "1" is written to the SWFRST bit in the TCFCFG_FCFGR register, the contents of the flash memory after reset cannot be assured.

The execution status of macro erasure can be confirmed with the value of the CERS bit. In addition, the timing limit excess for macro erasure can be checked with the value of HANG bit.

If there is at least 1 sector protected with flash security function, macro erasure cannot be performed. For details, see the security specifications.

The time it takes for HANG to become “1” is not equal to the Flash macro max erasing time defined in datasheet. For example, when max macro erasing time is defined as 5000ms in datasheet, it does not indicate HANG is set 5000ms after the initiation of the macro erase operation. HANG is designed so that the time for it to set is relatively longer than the max macro erasing time defined in the datasheet. Hence, do not design the program that expects HANG=1 after the max macro erasing time defined in datasheet .

4.6. Procedure for Sector Erase

You can erase data in units of sectors by writing the sector erase command sequence in the flash memory. The sector to which the address specified in the command sequence belongs is to be erased.

The command sequence for multiple sectors erasure can erase multiple sectors. However, in the case of 4 MB model, multiple sectors cannot be erased for 2 FLASH macros (SA39 and SA40) together.

After erase operation completes, all the values of the flash memory cells in the target sectors are "1".

The execution status of sector erasure can be confirmed with the value of the SERS bit of the status register. In addition, the timing limit excess for sector erasure can be checked with the value of HANG bit.

Figure 4-3 and Figure 4-4 show a sector erase flow example.

The flash security function does not allow erasure of protected sectors. For details, see the security specifications.

The time it takes for HANG to become “1” is not equal to the Flash sector max erasing time defined in datasheet. For example, when max sector erasing time is defined as 900ms in datasheet, it does not indicate HANG is set 900ms after the initiation of the sector erase operation. HANG is designed so that the time for it to set is relatively longer than the max sector erasing time defined in the datasheet. Hence, do not design the program that expects HANG=1 after the max sector erasing time defined in datasheet .

Figure 4-3 Flow Example When a Single Sector is Erased

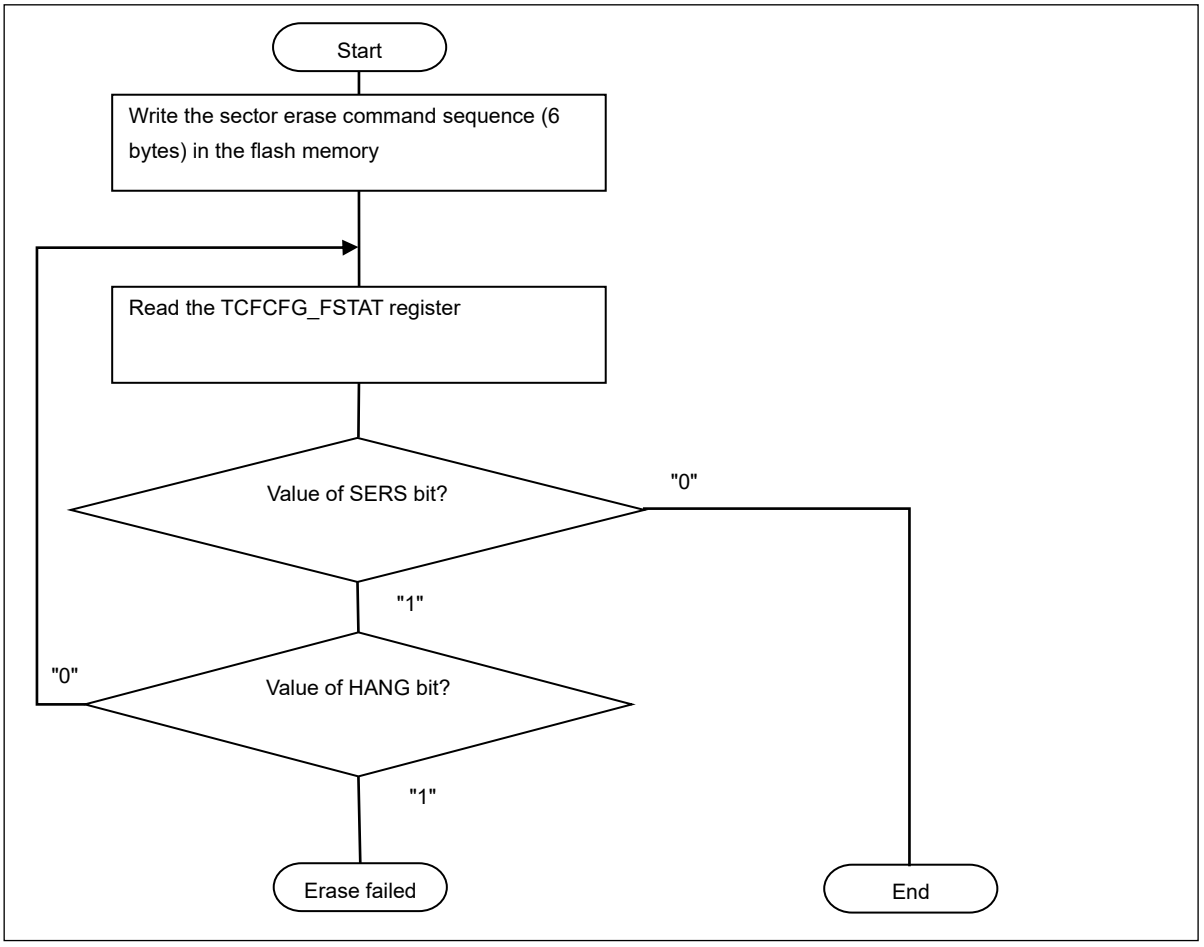
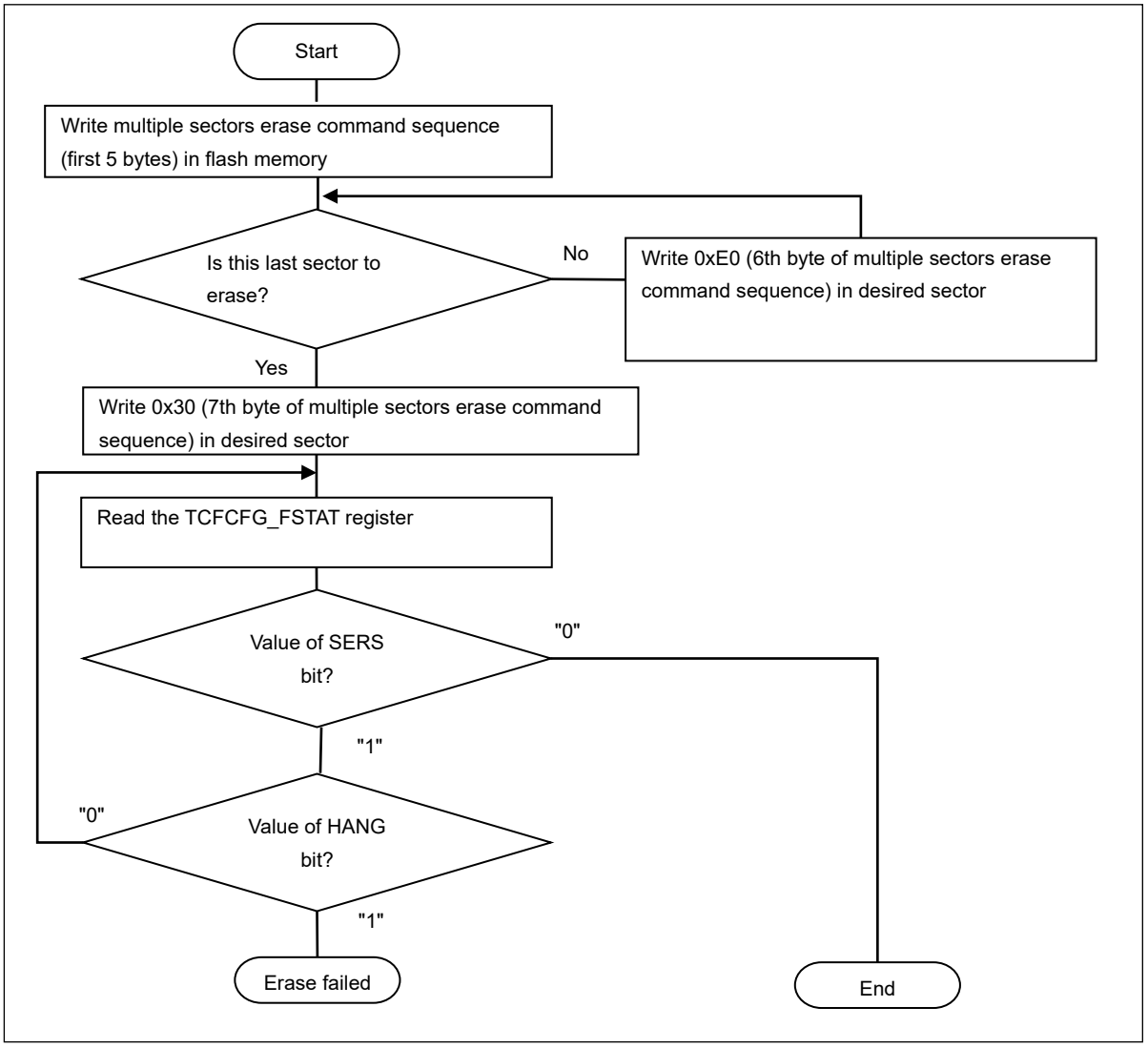


Figure 4-4 Flow Example When Multiple Sectors are Erased



4.6.1. Sector Erase Suspend

Write the sector erase suspend command sequence in the flash memory that performs the sector erase operation that you want to suspend. At this time, specify the address that belongs to the target sector for which sector erase is to be suspended.

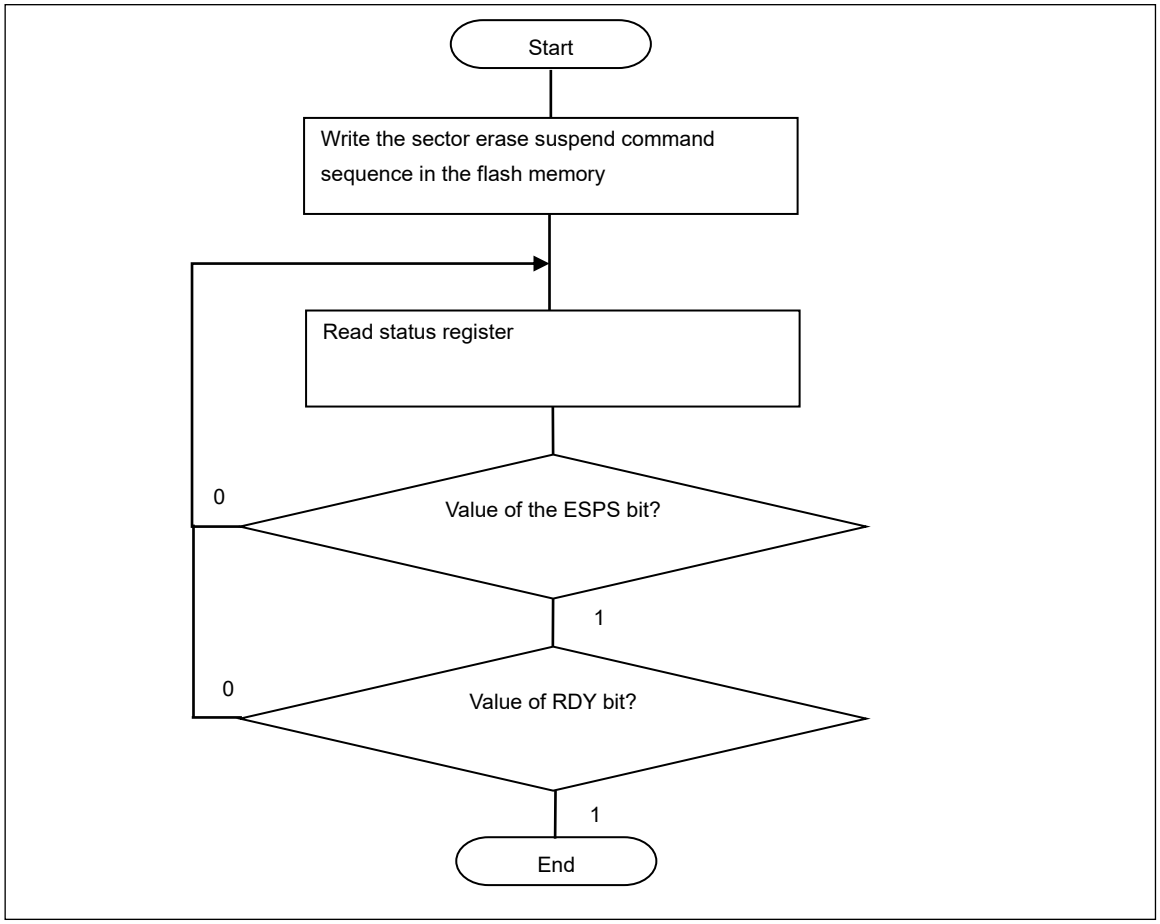
The sector erase suspend command sequence is valid only when flash memory is in the sector erase state. The sector erase suspend command sequence is ignored by the flash memory if the programming or macro erase operation is being performed or the flash memory is already in the sector erase suspend state.

Sector erase suspension can be confirmed by the TCFCFG_FSTAT:ESPS bit and TCFCFG_FSTAT:RDY bit both being "1". Therefore, before suspending sector erase to program/read data to TCFLASH, it is necessary to confirm the value for these bits.

Do not write a command sequence other than the read/reset or sector erase resume command in the flash memory in the sector erase suspend state.

Figure 4-5 shows a flow example of sector erase suspend.

Figure 4-5 Flow of Sector Erase Suspend Process (Example)



4.6.2. Sector Erase Resume

Write the sector erase resume command sequence in the flash memory that performs sector erase operation that you want to resume. At this time, specify the address that belongs to the target sector for which sector erase is suspended.

The sector erase resume command sequence is ignored by the flash memory if the sector erase is not suspended.

5. Registers

This section explains the registers in TCFLASH.

Register List

The register areas in TCFLASH are located in the peripheral functions area, each of which has a size of 1 KB.

Table 5-1 Register Areas in TCFLASH

Unit No.	Offset	Remarks
0	0x000	TCFLASH that is connected to the Cortex-R5F core #0
1	0x400	TCFLASH that is connected to the Cortex-R5F core #1. (*1)
2	0x800	TCFLASH that is connected to the Cortex-R5F core #2. (*2)
3	0xC00	TCFLASH that is connected to the Cortex-R5F core #3. (*3)

(*1) Reserved area for models in which Cortex-R5F core#1 is not assembled.
(*2) Reserved area for models in which Cortex-R5F core#2 is not assembled.
(*3) Reserved area for models in which Cortex-R5F core#3 is not assembled.

Table 5-2 shows the location of registers in the register area.

Table 5-2 Register Map

Abbreviated Register Name	Description	Reference
TCFCFG_FCPROTKEY	TCFLASH configuration protection key register	5.1
TCFCFG_FCFGR	TCFLASH configuration register	5.2
TCFCFG_FECCCTRL	TCFLASH ECC control register	5.3
TCFCFG_FDATEIR	TCFLASH data bit error injection register	5.4
TCFCFG_FECCEIR	TCFLASH ECC bit error injection register	5.5
TCFCFG_FICTRL	TCFLASH interrupt control register	5.6
TCFCFG_FSTAT	TCFLASH status register	5.7
TCFCFG_FSECIR	TCFLASH SEC interrupt register	5.8
TCFCFG_FECCEAR	TCFLASH ECC error address register	5.9
TCFCFG_FUCEDIR	TCFLASH uncorrectable error detection interrupt register	5.10
TCFCFG_FUCEAR	TCFLASH uncorrectable error address register	5.11
TCFCFG_BRCFG	TCM buffer region configuration register	5.12
TCFCFG_BRAT	TCM buffer region attribute register	5.13
TCFCFG_TCMBUF0 to 15	TCM Buffer0 to Buffer15 256-bit register	5.14

Table 5-3 Register Map

Offset	Abbreviated Register Name	Description
0x000	TCFCFG_FCPROTKEY 00000000_00000000_00000000_00000000	TCFLASH configuration protection key register
0x004	-	Reserved area
0x008	TCFCFG_FCFGR 00000000_00000000_00000000_00000011	TCFLASH configuration register
0x00C	-	Reserved area
0x010	TCFCFG_FECCCTRL 00000000_00000000_00000000_00000000	TCFLASH ECC control register
0x014	-	Reserved area
0x018	-	Reserved area (moved for 64-bit migration)
0x01C	TCFCFG_FECCEIR 00000000_00000000_00000000_00000000	TCFLASH ECC bit error injection register
0x020	TCFCFG_FICTRL 00000000_00000000_00000000_00000000	TCFLASH interrupt control register
0x024	-	Reserved area
0x028	TCFCFG_FDATEIR (lower32) 00000000_00000000_00000000_00000000	TCFLASH data bit error injection register Lower 32 bits
0x02c	TCFCFG_FDATEIR (upper32) 00000000_00000000_00000000_00000000	TCFLASH data bit error injection register Upper 32 bits
0x030 to 0x34	-	Reserved area
0x038	TCFCFG_FSTAT 00000000_00000000_00000000_00000000	TCFLASH status register
0x03C to 0x4C	-	Reserved area
0x050	TCFCFG_FSECIR 00000000_00000000_00000000_00000000	TCFLASH SEC interrupt register
0x054	TCFCFG_FECCEAR 00000000_00000000_00000000_00000000	TCFLASH ECC error address register
0x058 to 0x07C	-	Reserved area
0x080	TCFCFG_FUCEDIR 00000000_00000000_00000000_00000000	TCFLASH uncorrectable error detection interrupt register
0x084	TCFCFG_FUCEAR 00000000_00000000_00000000_00000000	TCFLASH uncorrectable error address register
0x088 to 0x08C	-	Reserved area
0x090	TCFCFG_BRCFG 00000000_00010000_00000000_11111111	TCM buffer region configuration register
0x094	-	Reserved area
0x098	TCFCFG_BRAT 00000000_00000000_00000000_00000001	TCM buffer region attribute register
0x09C to 0x0FC	-	Reserved area

Offset	Abbreviated Register Name	Description
0x100 to 0x11C 0x120 to 0x13C 0x140 to 0x15C 0x160 to 0x17C 0x180 to 0x19C 0x1A0 to 0x1BC 0x1C0 to 0x1DC 0x1E0 to 0x1FC 0x200 to 0x21C 0x220 to 0x23C 0x240 to 0x25C 0x260 to 0x27C 0x280 to 0x29C 0x2A0 to 0x2BC 0x2C0 to 0x2DC 0x2E0 to 0x2FC	TCFCFG_TCMBUF0 TCFCFG_TCMBUF1 TCFCFG_TCMBUF2 TCFCFG_TCMBUF3 TCFCFG_TCMBUF4 TCFCFG_TCMBUF5 TCFCFG_TCMBUF6 TCFCFG_TCMBUF7 TCFCFG_TCMBUF8 TCFCFG_TCMBUF9 TCFCFG_TCMBUF10 TCFCFG_TCMBUF11 TCFCFG_TCMBUF12 TCFCFG_TCMBUF13 TCFCFG_TCMBUF14 TCFCFG_TCMBUF15 128-bit ALL0	TCM Buffer0 to Buffer15 256-bit register
The following registers are those listed in the security specifications. These are noted here because the memory area is in the TCFLASH area, and actual read and program are performed through TCFLASH.For details, see the security specifications.		
0x3C0	TCFCFG0_SECSTAT	**Security State Register**
0x3C4	TCFCFG0_SER	**Security Enable Register**
0x3C8	TCFCFG0_SSR	**Security Scope Register**
0x3CC	TCFCFG0_CEER	**Chip Erase Enable Register**
0x3D0	TCFCFG0_SOER	**Security Overwrite Enable Register**
0x3D4	TCFCFG0_SWPOER	**Sector Write Permission Overwrite Enable Register**
0x3D8	TCFCFG0_WSWP	**Work Flash Sector Write Permissions**
0x3DC	TCFCFG0_C0SWP	**Code Flash Sector Write Permissions (Small Sectors)**
0x3E0	TCFCFG0_C1SWP	**Code Flash Sector Write Permissions (Large Sectors)**
0x3E4	TCFCFG0_C2SWP	
0x3E8	TCFCFG0_C3SWP	

5.1. TCFLASH Configuration Protection Key Register: TCFCFG_FCPROTKEY (TCFLASH Configuration Protection Key Register)

TCFLASH configuration protection key register (TCFCFG_FCPROTKEY) is used to protect the registers listed below from unintended writing.

TCFLASH configuration register (TCFCFG_FCFGR)

TCFLASH ECC control register (TCFCFG_FECCCTRL)

TCFLASH data bit error injection register (TCFCFG_FDATEIR)

TCFLASH ECC bit error injection register (TCFCFG_FECCEIR)

TCFLASH TCM buffer region configuration register (TCFCFG_BRCFG)

TCFLASH TCM buffer region attribute register(TCFCFG_BRAT)

TCFLASH TCM buffer register (TCFCFG_TCMBUF0 to 15)

Before programming these registers, you must write the correct value for the TCFLASH configuration protection key (0xCF61F1A5) to the TCFLASH configuration protection key register in the same unit to unlock programing to these registers.

Programming to the above registers is locked again by programming to the addresses in the TCFLASH internal register area.

If the value of the configuration protection key is wrong or an invalid procedure is used to program the above registers, such that programming is attempted in these registers without unlocking programming, TCFLASH generates a bus error response. For access with a CPU, a data abort exception is generated due to the bus error response.

Be sure to program in 32-bit to the TCFLASH configuration protection key register.

bit	31	0
Field	FCPROTKEY[31:0]	
R/W Attribute	R, W	
Protection Attribute	WP	
Initial Value	00000000_00000000_00000000_00000000	

[bit31:0] FCPROTKEY[31:0]: Configuration Protection Key

By writing a correct configuration protection key value (0xCF61F1A5) to the register, programming for the following is unlocked to set a new value: TCFLASH configuration register (TCFCFG_FCFGR), TCFLASH ECC control register (TCFCFG_FECCCTRL), TCFLASH data bit error insertion register (TCFCFG_FDATEIR), TCFLASH ECC bit error insertion register (TCFCFG_FECCEIR), TCFLASH TCM buffer region configuration register (TCFCFG_BRCFG), TCFLASH TCM buffer region attribute register (TCFCFG_BRAT), and TCFLASH TCM buffer register (TCFCFG_TCMBUF0 to 15).

When programming to these registers is unlocked, 0xFFFFFFFF is output from this register. When programming to these registers is locked, 0x00000000 is output from this register.

5.2. TCFLASH Configuration Register: TCFCFG_FCFGR (TCFLASH Configuration Register)

TCFLASH configuration register has the following functions:

- Resets flash memory
- Determines whether access via the TCM port or that via the AXI has higher priority.
- Enables programming of flash memory
- Specifies the wait cycle count when flash memory is accessed

This register setting can be changed only when the correct configuration protection key value is written to the TCFLASH configuration protection key register and programming to this register is unlocked.

bit	31	8
Field	Reserved	
R/W Attribute	R0, WX	
Protection Attribute	WPS	
Initial Value	00000000_00000000_00000000	

bit	7	6	5	4	3	2	1	0
Field	Reserved	SWFRST	TCMPR	WE	Reserved		FAWC[1:0]	
R/W Attribute	R0, WX	R0, W	R/W	R,W	R0, WX		R/W	
Protection Attribute	WPS							
Initial Value	0	0	0	0	00		11	

[bit31:7] Reserved: Reserved Bits

[bit 6] SWFRST: Software Reset

SWFRST	Description
0	There is no effect on operation.
1	Flash memory is reset.

When the WorkFLASH interface or SHE obtains program enable, writing "1" in this bit does not reset flash memory. When no program is enabled, writing "1" in this bit causes a bus error.

If "1" is written in this bit at the same time that program enable is obtained, flash memory reset is not issued. It causes a bus error response. To issue a reset to flash memory, it is necessary to obtain program enable before writing "1" in this bit. For the flow for obtaining for program enable, see 4.3 Enable Programming.

[bit5] TCMPR: TCM Priority Enable

Determine whether read access via TCM or read access via AXI has higher priority.

(AXI write access has the highest priority)

TCMPR	Description
0	Toggle the priority between the AXI access and TCM access every 16 accesses.
1	Give priority to the TCM access.

[bit4] WE: Program Enable

It sets enable/disable programming to/erasing of flash memory. If programming of the command sequence to flash memory is detected while this bit is "0", TCFLASH generates a bus error response.

Perform the program/erase operation in FLASH by starting the automatic algorithm after setting the program enable bit at WE. After the program/erase operation in FLASH completes, set the program disable bit.

When the WorkFLASH interface or SHE obtains program enable, program enable can not be set. In this case, writing "1" cannot be reflected, and the read value becomes "0". For the configuration method for program enable, see 4.3 Enable Programming.

WE	Description
0	Programming command sequence is prohibited.
1	Programming command sequence is permitted.

[bit3:2] Reserved: Reserved Bits

[bit1:0] FAWC[1:0]: Flash Wait Control

These bits set the wait cycle count when flash memory is accessed. Set the appropriate value according to the operating frequency of the flash memory and flash memory access time.

FAWC[1:0]	Description
00	Do not insert wait cycle.
01	Insert 1 wait cycle.
10	Insert 2 wait cycles.
11	Insert 3 wait cycles.

Note:

- *Avoid TCFlash access via AXI bus while changing Flash wait configuration (FAWC[1:0]) 0 -> X (X=1,2,3)*

5.3. TCFLASH ECC Control Register: TCFCFG_FECCCTRL (TCFLASH ECC Control Register)

This register is used to control the operation of the ECC logic. This register setting can be changed only when the correct configuration protection key value is written to the TCFLASH configuration protection key register and programming to this register is unlocked. You can change the setting value of this register only once. If a 2nd and subsequent write access is attempted to this register, TCFLASH generates a bus error response.

bit	31	8
Field	Reserved	
R/W Attribute	R0, WX	
Protection Attribute	WPS	
Initial Value	00000000_00000000_00000000	

bit	7	6	5	4	3	2	1	0
Field	Reserved							ECCOFF
R/W Attribute	R0, WX							R/W
Protection Attribute	WPS							
Initial Value	0000000							0

[bit31:1] Reserved: Reserved Bits

[bit 0] ECCOFF: ECC Off

This bit enables or disables operation of the ECC logic for AXI accesses. ECC generation and check for TCM accesses are controlled by the settings in the Cortex-R5F core.

ECCOFF	Description
0	Perform ECC generation and check for AXI accesses.
1	Do not perform ECC generation and check for AXI accesses.

5.4. TCFLASH Interrupt Control Register: TCFCFG_FICTRL (TCFLASH Interrupt Control Register)

This register allows enabling interrupt request generation for each factor and clearing interrupt factors.

This register is not supported.

Always write "0" to RDYIE and HANGIE bit.

bit	31							16
Field	Reserved							
R/W Attribute	R0, WX							
Protection Attribute	WP							
Initial Value	00000000_00000000							

bit	15	14	13	12	11	10	9	8
Field	Reserved						HANGIC	RDYIC
R/W Attribute	R0, WX						R0, W0	R0, W0
Protection Attribute	WP							
Initial Value	000000						0	0

bit	7	6	5	4	3	2	1	0
Field	Reserved						HANGIE	RDYIE
R/W Attribute	R0, WX						R/W0	R/W0
Protection Attribute	WP							
Initial Value	000000						0	0

[bit31:10] Reserved: Reserved Bits

[bit 9] HANGIC: Hang Interrupt Clear

HANGIC	Description
0	Writing "0" to this bit has no meaning.
1	Writing "1" to this bit clears the HANGINT bit in the TCFCFG_FSTAT register.

Note:

- *Do not use this function.*

[bit 8] RDYIC: Ready Interrupt Clear

RDYIC	Description
0	Writing "0" to this bit has no meaning.
1	Writing "1" to this bit clears the RDYINT bit in the TCFCFG_FSTAT register.

Note:

- *Do not use this function.*

[bit7:2] Reserved: Reserved Bits

[bit 1] HANGIE: Hang Interrupt Enable

HANGIE	Description
0	Disable the generation of hang interrupt requests.
1	Enable the generation of hang interrupt requests.

Note:

- *Do not use this function.*

[bit 0] RDYIE: Programming/Erasing Ready Interrupt Enable

RDYIE	Description
0	Disable the generation of programming/erasing ready interrupt requests.
1	Enable the generation of programming/erasing ready interrupt requests.

Note:

- *Do not use this function.*

5.5. TCFLASH Data Bit Error Injection Register: TCFCFG_FDATEIR (TCFLASH Data Bit Error Injection Register)

The TCFLASH data bit error injection register is used to test operation of the ECC logic by injecting an error to the data bit read from the flash memory.

This register setting can be changed only when the correct configuration protection key value is written to the TCFLASH configuration protection key register and programming to this register is unlocked.

bit	63	0
Field	FDATEIR[63:0]	
R/W Attribute	R/W	
Protection Attribute	WPS	
Initial Value	00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000	

[bit63:0] FDATEIR[63:0]: Data Bit Error Injection Point

The exclusive OR of the value in this field and the value of the data bits that are read from the flash memory is sent to the ECC inspection logic.

FDATEIR[i] (63≥i≥0)	Description
0	The bits at the same position as that of the data that is read from the flash memory are sent to the ECC inspection logic as they are.
1	The bits at the same position as that of the data that is read from the Flash memory are sent to the ECC inspection logic after being inverted.

5.6. TCFLASH ECC Bit Error Insertion Register: TCFCFG_FECCEIR (TCFLASH ECC Bit Error Injection Register)

The TCFLASH ECC bit error injection register is used to test operation of the ECC logic by injecting an error into the ECC bit from the flash memory.

This register setting can be changed only when the correct configuration protection key value is written to the TCFLASH configuration protection key register and programming to this register is unlocked.

bit	31	30	29	28	27	26	25	24
Field	Reserved						LMASK[1:0]	
R/W Attribute	R0, WX						R/W	
Protection Attribute	WPS							
Initial Value	000000						00	

bit	23	8
Field	Reserved	
R/W Attribute	R0, WX	
Protection Attribute	WPS	
Initial Value	00000000_00000000	

bit	7	6	5	4	3	2	1	0
Field	FECCEIR[7:0]							
R/W Attribute	R/W							
Protection Attribute	WPS							
Initial Value	00000000							

[bit31:26] Reserved: Reserved Bits

[bit25:24] LMASK[1:0]: Error Injection Lane Mask

These bits specify whether to insert an error in the data read from the flash memory.

LMASK[i] (1≥i≥0)	Description
0	Insert an error between bit(64*(i+1)-1) and bit(64*i) of the data read from the flash memory, as specified by the TCFCFG_FDATEIR register. Similarly, insert an error between bit(8*(i+1)-1) and bit(8*i) of the ECC data read from the flash memory, as specified by the TCFCFG_FECCEIR register.
1	Do not insert an error between bit (64*(i+1)-1) and (64*i) of the data read from the flash memory. Also, do not insert an error between bit (8*(i+1)-1) and bit (8*i) of the ECC data read from the flash memory.

LMASK[0] decides whether an error inserted in data ending with 0x0 to 7 and ECC data should be masked. LMASK[1] decides whether an error inserted in data ending with 0x8 to F and ECC data should be masked.

[bit23:8] Reserved: Reserved Bits

[bit7:0] FECCEIR[7:0]: ECC Bit Error Injection Point

The exclusive OR of the value in this field and the value of the ECC bits that are read from the flash memory is sent to the ECC inspection logic.

FECCEIR[i] (7≥i≥0)	Description
0	Directly send the bits with the same position as the ECC data being read from the flash memory, to the ECC inspection logic.
1	Send the bits with the same position as the ECC data being read from the flash memory, to the ECC inspection logic after inverting their values.

5.7. TCFLASH Status Register: TCFCFG_FSTAT (TCFLASH Status Register)

This register is a read-only register that retains the state of the flash memory and individual interrupt factors. Programming to this register returns a bus error.

bit	31	16
Field	Reserved	
R/W attribute	R0, WX	
Protection Attribute	-	
Initial Value	00000000_00000000	

bit	15	14	13	12	11	10	9	8
Field	Reserved						HANGINT	RDYINT
R/W Attribute	R0, WX						R, WX	R, WX
Protection Attribute	-							
Initial Value	000000						0	0

bit	7	6	5	4	3	2	1	0
Field	CERS	PGMS	ESPS	ERSEC	SERS	READ	HANG	RDY
R/W attribute	R, WX	R, WX	R, WX	R, WX	R, WX	R, WX	R, WX	R, WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

[bit31:10] Reserved: Reserved Bits

[bit 9] HANGINT: Hang Up Interrupt

This bit indicates that a hung up interrupt request is generated because transition of the flash memory to the hung up 1 state is detected.

This bit is cleared by writing "1" to the HANGIC bit in the TCFCFG_FICTRL register.

HANGINT	Description
0	No hang up interrupt request is generated.
1	A hang up interrupt request is generated.

Note:

- *Do not use this function. Use polling instead of interrupt. Use HANG(bit-1) bit.*

[bit 8] RDYINT: Ready INTerrupt

This bit indicates that a ready interrupt request has been generated because a change of the flash memory to the ready state has been detected. This bit is cleared by writing "1" to the RDYIC bit in the TCFCFG_FICTRL register for the same memory in the same unit.

RDYINT	Description
0	No programming/erasing ready interrupt request is generated.
1	A programming/erasing ready interrupt request is generated.

Notes:

- *Initial value depends on the boot configuration.*
- *Do not use this function. Use polling instead of interrupt. Use RDY(bit-0) bit.*

[bit 7] CERS: Chip ERase Status

This bit shows the TCFLASH macro erase status.

CERS	Description
0	TCFLASH does not perform macro erase.
1	TCFLASH performs macro erase.

[bit 6] PGMS: ProGraM Status

This bit shows the TCFLASH programming status.

PGMS	Description
0	No programming to TCFLASH.
1	Programming to TCFLASH.

[bit 5] ESPS: Erase SusPend Status

This bit shows the TCFLASH sector erase suspend status.

ESPS	Description
0	TCFLASH does not suspend sector erase.
1	TCFLASH received sector erase suspend command.

This bit indicates the receipt of the sector erase suspend command. Therefore, the suspension of sector erase can be confirmed with the TCFCFG_FSTAT:ESPS bit becoming "1" and the TCFCFG_FSTAT:RDY bit becoming "1". Therefore, before suspending sector erase to program/read data to TCFLASH, it is necessary to confirm the value for these bits.

[bit 4] ERSEC: ERase Suspend SECtor status

This bit indicates the target sector read execution status while TCFLASH sector erase is suspended.

ERSEC	Description
0	There is no target sector read while TCFLASH sector erase is suspended.
1	There is target sector read while TCFLASH sector erase is suspended.

[bit 3] SERS: Sector ERase Status

This bit indicates the TCFLASH sector erase status.

SERS	Description
0	TCFLASH does not perform sector erase.
1	TCFLASH performs sector erase.

[bit 2] READ: Reading ready

This bit indicates whether TCFLASH is in the reading ready state. TCFLASH can start new read command execution in the reading ready state.

READ	Description
0	Unable to enter read command to TCFLASH
1	Able to enter read command to TCFLASH

Note:

- *Initial value depends on the boot configuration.*

[bit 1] HANG: Hang up

This bit indicates whether the flash memory is in hang up 1 state. The flash memory changes to hang up 1 state in the following cases.

- When write access with "1" is attempted to a cell with the value of "0"
- When execution of the automatic algorithm has not completed within the given time

HANG	Description
0	The flash memory is not in hang up 1 state.
1	The flash memory is in hang up 1 state.

[bit 0] RDY: Programming/erasing Ready

This bit indicates whether the flash memory is in the programming/erasing ready state. The flash memory can start command execution in the programming/erasing ready state.

RDY	Description
0	Indicate that the flash memory is performing program or erase operation using the automatic algorithm.
1	Indicate that the flash memory completes program or erase operation using the automatic algorithm and is ready to start the next command.

You can confirm whether TCFLASH is in the programming/erasing ready state with the RDY bit. The reading ready state can be checked with the READ bit.

Note:

- *Initial value depends on the boot configuration.*

5.8. TCFLASH SEC Interrupt Register: TCFCFG_FSECIR (TCFLASH SEC Interrupt Register)

This register holds status flags, enable bits, and clear bits concerning 1-bit error correction interrupt.

bit	31	30	29	28	27	26	25	24
Field	SYN							
R/W Attribute	R, WX							
Protection Attribute	WP							
Initial Value	00000000							

bit	23	22	21	20	19	18	17	16
Field	Reserved							SECINT
R/W Attribute	R0, WX							R, WX
Protection Attribute	WP							
Initial Value	0000000							0

bit	15	14	13	12	11	10	9	8
Field	Reserved							SECIC
R/W Attribute	R0, WX							R0, W
Protection Attribute	WP							
Initial Value	0000000							0

bit	7	6	5	4	3	2	1	0
Field	Reserved							SECIE
R/W Attribute	R0, WX							R/W
Protection Attribute	WP							
Initial Value	0000000							0

[bit31:24] SYN: Syndrome

This bit holds syndrome when a 1-bit error is detected. If a 1-bit error is detected in both the upper 64 bits and lower 64 bits when 128-bit read access is performed, syndrome for the error detected in the lower 64 bits is stored. Use the value of bit3 in the TCFCFG_FECCEAR register to determine whether the syndrome stored in this field is detected in the upper 64-bits or lower 64 bits.

[bit23:17] Reserved: Reserved Bits

[bit 16] SECINT: 1-Bit Error Correction Interrupt

This bit indicates whether there is a 1-bit error correction interrupt request. A 1-bit error correction interrupt request is generated if a 1-bit error is detected with the ECC inspection during reading, and is corrected. Flash ECC is generated and detected for every 64bits. Note that if a read operation with the size of is 32-bit wide or less is performed, and an ECC error exists in other byte lanes than the read address, this bit is still set. This is a read-only bit. This bit is cleared by writing "1" to the SECIC bit in the same register.

SECINT	Description
0	No 1-bit error correction interrupt request is generated.
1	A 1-bit error correction interrupt request is generated.

[bit15:9] Reserved: Reserved Bits

[bit 8] SECIC: 1-Bit Error Correction Interrupt Clear

SECIC	Description
0	Writing "0" to this bit has no meaning.
1	Writing "1" to this bit clears the SECINT bit in the same register.

[bit7:1] Reserved: Reserved Bits

[bit 0] SECIE: 1-bit Error Correction Interrupt Enable

SECIE	Description
0	Disable generation of 1-bit error correction interrupt request.
1	Enable generation of 1-bit error correction interrupt request.

5.9. TCFLASH ECC Error Address Register: TCFCFG_FECCEAR (TCFLASH ECC Error Address Register)

This register retains the address at which a 1-bit error is detected during reading. If a 1-bit error is detected multiple times, the register retains the address of the error detected last.

This register is read-only. Programming to this register returns a bus error.

bit	31	0
Field	FECCEAR[31:0]	
R/W Attribute	R, WX	
Protection Attribute	-	
Initial Value	00000000_00000000_00000000_00000000	

[bit31:0] FECCEAR[31:0]: Error Address

These bits hold an address (word address) at which a 1-bit error is detected in the ECC check when read accessed. If an error is detected multiple times, they retain the address of the error detected last. Flash ECC is generated and detected for every 64bits. Note that if a read operation with the size of is 32-bit wide or less is performed, and an ECC error exists in other byte lanes than the read address, the address of the read operation is retained in this register.

If a 1-bit error is detected in both the upper 64 bits and lower 64 bits when 128-bit read access is performed, the address of the lower 64 bits is stored.

5.10. TCFLASH Uncorrectable Error Detection Interrupt Register: TCFCFG_FUCEDIR (TCFLASH Un-Correctable Error Detection Interrupt Register)

This register holds status flags and clear bits concerning the uncorrectable error detection interrupt.

bit	31	30	29	28	27	26	25	24
Field	SYN							
R/W Attribute	R, WX							
Protection Attribute	WP							
Initial Value	00000000							

bit	23	22	21	20	19	18	17	16
Field	Reserved							UCEDINT
R/W Attribute	R0, WX							R, WX
Protection Attribute	WP							
Initial Value	0000000							0

bit	15	14	13	12	11	10	9	8
Field	Reserved							UCEDIC
R/W Attribute	R0, WX							R0, W
Protection Attribute	WP							
Initial Value	0000000							0

bit	7	6	5	4	3	2	1	0
Field	Reserved							
R/W Attribute	R0, WX							
Protection Attribute	WP							
Initial Value	00000000							

[bit31:24] SYN: Syndrome

This bit holds syndrome when a 2-bit error is detected. If an uncorrectable error is detected in both the upper 64 bits and lower 64 bits when 128-bit read access is performed, syndrome for the error detected in the lower 64 bits is stored. Use the value of bit3 in the TCFCFG_FUCEAR register to determine whether the stored syndrome is for an error that occurred in the upper 64 bits or an error that occurred in the lower 64 bits.

[bit23:17]: Reserved Bits

[bit16] UCEDINT: Uncorrectable Error Detection Interrupt

This bit indicates whether an uncorrectable error detection interrupt request exists. An uncorrectable error detection interrupt request is generated if an uncorrectable error is detected in the ECC check at read access. This bit is read-only. The value written in this bit has no meaning. This bit is cleared by writing "1" to the UCEDIC bit in this register.

UCEDINT	Description
0	Indicate that an uncorrectable error detection interrupt request is not generated.
1	Indicate that an uncorrectable error detection interrupt request is generated.

[bit15:9] Reserved: Reserved Bits

[bit8] UCEDIC: Uncorrectable Error Detection Interrupt Clear

UCEDIC	Description
0	Writing "0" to this bit has no meaning.
1	Writing "1" to this bit clears the UCEDINT bit in this register.

"0" is always read from this bit.

[bit7:0] Reserved: Reserved Bits

5.11. TCFLASH Uncorrectable Error Address Register: TCFCFG_FUCEAR (TCFLASH Un-Correctable Error Address Register)

This register holds an address (word address) at which an uncorrectable error is detected when read accessed. If multiple uncorrectable errors are detected, it holds the address at which the error is detected last. If an uncorrectable error is detected in both the upper 64 bits and lower 64 bits when 128-bit read access is performed, the address of the lower 64 bits is stored.

This register is read-only. Programming to this register returns a bus error.

bit	31	0
Field	UCEA[31:0]	
R/W Attribute	R, WX	
Protection Attribute	-	
Initial Value	00000000_00000000_00000000_00000000	

[bit 31:0] UCEA[31:0]: Uncorrectable Error Address

These bits indicate the address at which an uncorrectable error is detected in the ECC check at read access. If multiple errors are detected, they hold the address at which the error is detected last.

5.12. TCFLASH TCM Buffer Region Configuration Register: TCFCFG_BRCFG (TCFLASH Buffer Region Configuration Register)

This register configures the range of the TCM buffer region. When the content of this register is changed, it is necessary to intentionally clear the entire TCM buffer region.

bit	31															16
Field	REGION_END_BUF[15:0]															
R/W Attribute	R/W															
Protection Attribute	WPS															
Initial Value	00000000_00010000															

bit	15															8
Field	SSEC[7:0]															
R/W Attribute	R/W															
Protection Attribute	WPS															
Initial Value	00000000															

bit	7					5										0
Field	ESEC[7:0]															
R/W Attribute	R/W															
Protection Attribute	WPS															
Initial Value	11111111															

[bit31:16] REGION_END_BUF[15:0]: TCM Buffer Word Number of Region End

These bits set the TCM buffer word number of this region end. The first of this region end is set to "0". If REGION_END_BUF is set to "8", it uses up to TCFCFG_TCMBUF0 to TCFCFG_TCMBUF7.

[bit15:8] SSEC[7:0]: Starting Bufferable Sector Number

These bits set the starting sector number that can be buffered by this TCM buffer region. This sector is included in the bufferable sectors. (Big and small sectors shall not be differentiated, but shall be set using numbers.) Set SSEC≦ESEC.

[bit7:0] ESEC[7:0]: Ending Bufferable Sector Number

These bits set the ending sector number that can be buffered by this TCM buffer region. This sector is included in the bufferable sectors. (Big and small sectors shall not be differentiated, but shall be set using numbers.) When a sector number larger than the number mounted is set, the largest mounted sector number is used. Set SSEC≦ESEC.

5.13. TCFLASH TCM Buffer Region Attribute Register: TCFCFG_BRAT (TCFLASH Buffer Region Attribution Register)

This register configures the attributes of the TCM buffer region. When the content of this register is changed, it is necessary to intentionally clear the TCM buffer region in the region.

bit	31	24
Field	Reserved	
R/W Attribute	R0, WX	
Protection Attribute	WPS	
Initial Value	00000000	

bit	23	16
Field	Reserved	VCLR
R/W Attribute	R0, WX	R0, W
Protection Attribute	WPS	
Initial Value	0000000	0

bit	15	9	8
Field	Reserved	AM[1:0]	
R/W Attribute	R0, WX	R/W	
Protection Attribute	WPS		
Initial Value	000000	00	

bit	7	0
Field	Reserved	RGEN
R/W Attribute	R0, WX	R/W
Protection Attribute	WPS	
Initial Value	0000000	1

[bit31:17] Reserved: Reserved Bits

[bit 16] VCLR: TCM Buffer Region Intentional Clear

VCLR	Description
0	Writing "0" to this bit has no meaning.
1	Writing "1" to this bit performs the following operations. Instruct to clear the entire relevant TCM buffer region. All the buffers become invalid and the buffer pointer returns to its beginning.

[bit15:10] Reserved: Reserved Bits

[bit 9:8] AM[1:0]: Buffer Policy for TCM Buffer Region

AM[1:0]	Description
00	Buffer all the data that enters the access wait state. Although this consumes a considerable amount of buffers, if a short loop occurs frequently, TCM throughput can be expected to be higher than FLASH macro theoretical throughput.
01	Buffer the first 2 pieces of data in the non-consecutive address. This is targeting the TCM throughput that is close to FLASH macro theoretical throughput.
10	Lock the buffer region. The buffer contents can be accessed only by directly writing to the buffer register and are not changed by any other means. In the TCM buffer region, in which this lock mode is set, TCFCFG_BRCFG:SSEC[7:0] and TCFCFG_BRCFG:ESEC[7:0] have no meaning. Intentional care for coherency is required.
11	Reserved

[bit7:1] Reserved: Reserved Bits

[bit 0] RGEN: TCM Buffer Region Enable

RGEN	Description
0	The relevant buffer region does not function.
1	The relevant buffer region functions.

5.14. TCFLASH TCM Buffer Register: TCFCFG_TCMBUF0 to 15 (TCFLASH Buffer0 to 15)

This register reads/writes the TCM buffer itself. It is a 256-bit register. There are 16 registers. Programming/erasing FLASH clears the TCM buffer register. Use 32-/64-bit access for programming of the TCM buffer register only when TCFCFG_BRAT:AM = 10. Other programming causes a bus error response. Program when TCFCFG_BRAT:RGEN = 0 (TCM buffer is disabled).

When TCFCFG_BRAT:AM = 00/01, the data buffered with TCM can be read. When TCFCFG_BRAT:AM = 10 (Lock mode), a user can intentionally set the TCM buffer content.

bit	127 … 0
Field	BUFDATA[127:0]
R/W Attribute	R,W
Protection Attribute	WPS
Initial Value	ALL0

bit	239 … 161	160	159 … 152	151 … 132	131 … 128
Field	Reserved	BVALID	Reserved	BUFADDH[23:4]	BUFADDL[3:0]
R/W Attribute	R0, WX	R,W	R0, WX	R,W	R0, WX
Protection Attribute	WPS			WPS	
Initial Value	ALL0			0x800000	

bit	255 … 240
Field	ECC_PARITY[15:0]
R/W Attribute	R,W
Protection Attribute	WPS
Initial Value	ALL0

[bit127:0] BUFDATA[127:0]: Data Value of TCM Buffer

These bits are data to be buffered. If TCFCFG_BRAT:AM is in lock mode, these bits do not function unless they are intentionally set. It is necessary to rewrite when TCM is not read. In addition, do not set the same address to multiple buffers.

Do not program if not in lock mode.

[bit131:128] BUFADDL[3:0]: Buffer Address Lower Bit

These are the lower 4 bits that are implicitly 0 bits because the buffer address is 128-bit data.

[bit151:132] BUFADDH[23:4]: Buffer Address Upper Bit

These are the upper 20 bits of the buffer address. These bits indicate the address that uses buffering. To use in lock mode, do not set the same address to multiple buffers.

Note:

- *See the following table for the initial value of BUFADDH[23:4] and BUFADDL[3:0].*

TCFLASH memory size	Initial value of BUFADDH[23:4] and BUFADDL[3:0]
1MB	0x900000
2MB	0x800000
4MB	0x800000

[bit159:152] Reserved: Reserved Bits

[bit160] BVALID: Buffer Valid

This bit indicates that the buffer address and data are valid.

[bit239:161] Reserved: Reserved Bits

If 32-bit programming is performed in [bit 223:192], it causes a bus error response.

[bit255:240] ECC_PARITY: Parity Value of TCM Buffer

This is the parity value of buffered data. To use in lock mode, this must be set by using software for the calculation.

6. Others

This section explains precautions on the use of TCFLASH.

Handling of Values Read from Reserved Bits

"0" is read from the reserved bits that exist in registers in TCFLASH. However, do not allow the values read from the reserved bits to have meanings when programming, from the standpoint of software compatibility with the future products.

Reset during Execution of Program or Erase Operation

If this LSI is reset during execution of the program operation, the contents of the target address are shown as undefined. If it is reset during execution of the sector erase operation, the contents of the target sector are shown as undefined. In these cases, following steps should be done.

1. erase the sector.
2. re-program the sector

Instruction Fetch from Flash Memory

During execution of the program or erase operation, you cannot read data from TCFLASH. Therefore, you must copy the required data or program from TCFLASH to RAM before starting the program or erase operation so that you do not need to read them while the operation is being executed.

Wait for Reset Completion from Software

Flash memory is reset by writing "1" to the SWFRST bit in the TCFCFG_FCFGR register. If the flash memory is reset, be sure to wait for reset completion by monitoring the RDY bit in the TCFCFG_FSTAT register in the unit that executes reset before accessing the memory.

To generate software reset, the program enable (TCFCFG_FCFGR:WE) bit must be "1". Software reset cannot be generated without program enable. If software reset cannot be generated, it causes a bus error response.

Suspend Command

After the suspend command is issued, do not read flash memory until the RDY bit in the status register becomes "1".

When 1-bit ECC error is detected

When the CPU accesses the TCFLASH via TCM and detected 1-bit ECC error with the default setting, it tries to re-write the 1-bit-error-corrected data to the same address by the hardware. But TCFlash cannot be written. So it results to the data abort.

Note:

TCFLASH Status Register Bits

Since some status information of flash memory is common to Code side (TCFLASH) and Work side (WorkFLASH) in Dual Port FLASH macro,

TCFLASH status register bits listed below are affected by WorkFLASH operation in addition to TCFLASH operation:

- *TCFCFG_FSTAT.ESPS*
- *TCFCFG_FSTAT.ERSEC*
- *TCFCFG_FSTAT.RDY*
- *TCFCFG_FSTAT.HANG*
- *TCFCFG_FSTAT.RDYINT*
- *TCFCFG_FSTAT.HANGINT*

Use these bits only when program/erase is performed in TCFLASH.

7. Support for Dual Port Flash Macro

This section explains restrictions when using a dual port flash macro as a flash macro.

Difference from the Case where Single Port Flash Macro is Used

- TCFLASH uses ports on the code side of the dual port flash.
- Since the programming/erasing circuit is shared with the Work side, some operation at the Work side affects the Code side.

Software control that keeps the foregoing in mind is necessary.

Whether the Code Side and Work Side of a Dual Port Macro can be Executed at the Same Time

	Code:Read/NOP Work:Read/NOP	Code:Read/NOP Work:PGM/ERS	Code:PGM/ERS Work:Read/NOP	Code:PGM/ERS Work:PGM/ERS
Operation enabled/disabled	OK	OK	OK	NG

Access of Dual Port to Code and Work Sides

The flash destinations that each interface accesses are described.

Interfaces	TCFLASH	WorkFLASH	SHE
Accessed Area	Code	Work	Work

CHAPTER 18: WorkFLASH

This chapter explains WorkFLASH.

1. **Overview**
2. **Configuration**
3. **Explanation of Operation**
4. **Setting Procedure**
5. **Registers**
6. **Other**
7. **Secure Hardware Extension (SHE) Support**
8. **Dual Port FLASH Macro Support**

WORKFLASH-TXXPT03P01R01L07-E1-XX

1. Overview

This section provides an overview of WorkFLASH.

WorkFLASH is rewritable, non-volatile data memory with a built-in MCU.

Some features of WorkFLASH are listed below.

- Enables reading in units of 8/16/32/64 bits.
- Provides an ECC (SEC/DED) function. (64bit ECC)
- Enables switching between the enabled and disabled states of the ECC function through a register setting.
- Enables switching between ECC-enabled access and ECC-disabled access by using two mirror areas.
- Enables writing in units of 64 bits when ECC is enabled. In any attempt to write with a width of less than 64 bits when ECC is enabled, WorkFLASH reads the corresponding 64-bit data first and then replaces the bit portion to be written. After that, it creates 64-bit data, writes the data to be written, and writes the ECC generated from the 64 bits.
- Enables writing in units of 8/16/32/64 bits when ECC is disabled.

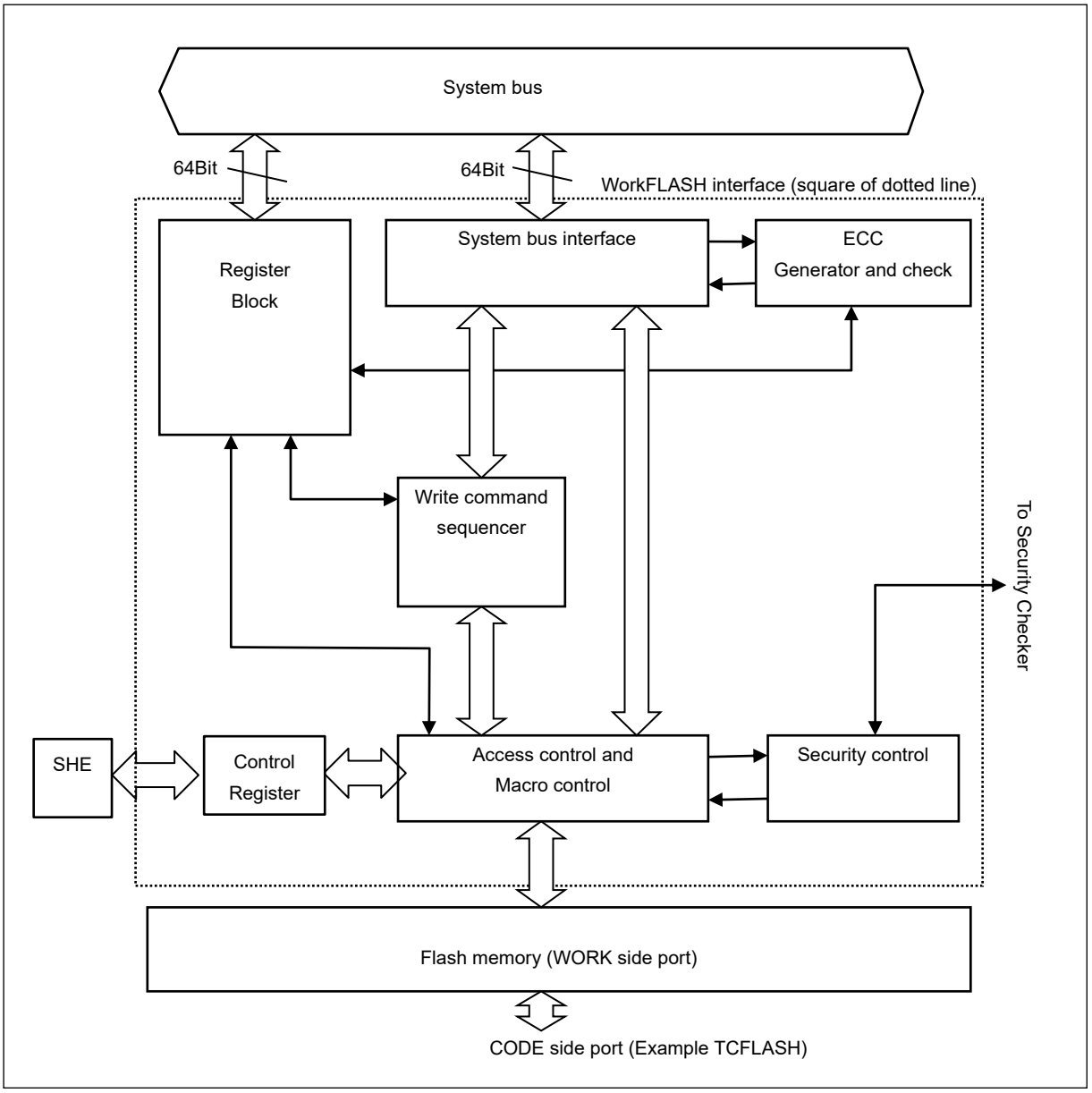
- Equipped with an ECC check circuit test function with error injection.
- Enables writing from not only a CPU but also the DMAC.
- Enables writing from an unprivileged mode.
- Equipped with a control register setting protection function using a key code.
- Equipped with a Flash security function for data protection.
- Equipped with a bus error response function for access to reserved areas.
- Enables interrupt generation with writing/erasing completed.
- The Flash security function has a function to protect against writing and erasing in units of sectors. CPU can execute instructions on WorkFLASH in writing or erasing TCFLASH.

2. Configuration

This section explains the configuration within WorkFLASH.

2.1. Block Diagrams

Figure 2-1 Block Diagram of WorkFLASH



2.2. Address/Sector Maps in User Mode

This section explains WorkFLASH address/sector maps in user mode.

In the products in this series, 48 KB/112 KB/240 KB WorkFLASH is mounted depending on the model. (In this model, (product specification) KB WorkFLASH is mounted.)

WorkFLASH has the addresses placed on the memory map in a 4-MB area between 0x0E000000 and 0x0E3FFFFF. As shown in Figure 2-2, this area is further divided into four mirror areas.

Mirror area 1 is an area provided for access to WorkFLASH via ECC logic.

When WorkFLASH is read via mirror area 1 with ECC enabled for the unit to be read, an ECC check is performed. If the check detects a 1-bit error, the error is corrected. If it detects an error of 2 or more bits, a bus error response is made. However, in cases where ECC is disabled, the ECC check, error detection, and correction are not performed. In these cases, even if there is an error, no action such as error correction and bus error response is taken for it. When writing to WorkFLASH via mirror area 1 with ECC enabled for the unit to be written, an ECC is generated and written. WorkFLASH can generate an ECC for only 64-bit data, so for any attempt to write 8/16/32 bits via mirror area 1 with ECC enabled, WorkFLASH reads the corresponding 64-bit data and then replaces the relevant 8/16/32-bit data to create new 64-bit data. Based on this, an ECC is generated and written at the same time.

Mirror area 2 is entirely a reserved area. For this reason, WorkFLASH makes a bus error response for any attempt to read and write or erase data in mirror area 2.

Mirror area 3 is an area provided for access to WorkFLASH by bypassing ECC logic. If WorkFLASH is read via mirror area 3, the ECC check and error detection are not performed. Therefore, even if an error occurs, there is neither the 1-bit error correction nor the bus error response associated with the detection of an error of 2 or more bits. If writing is performed via mirror area 3, no ECC is generated and written regardless of whether ECC is enabled or disabled.

Mirror area 4 is a read-only area that is set up for reading with Normal memory attribute access. If writing is attempted, WorkFLASH makes a bus error response.

For reading and writing, the access attributes for each mirror area must be set in the MPU. See Figure 2-2.

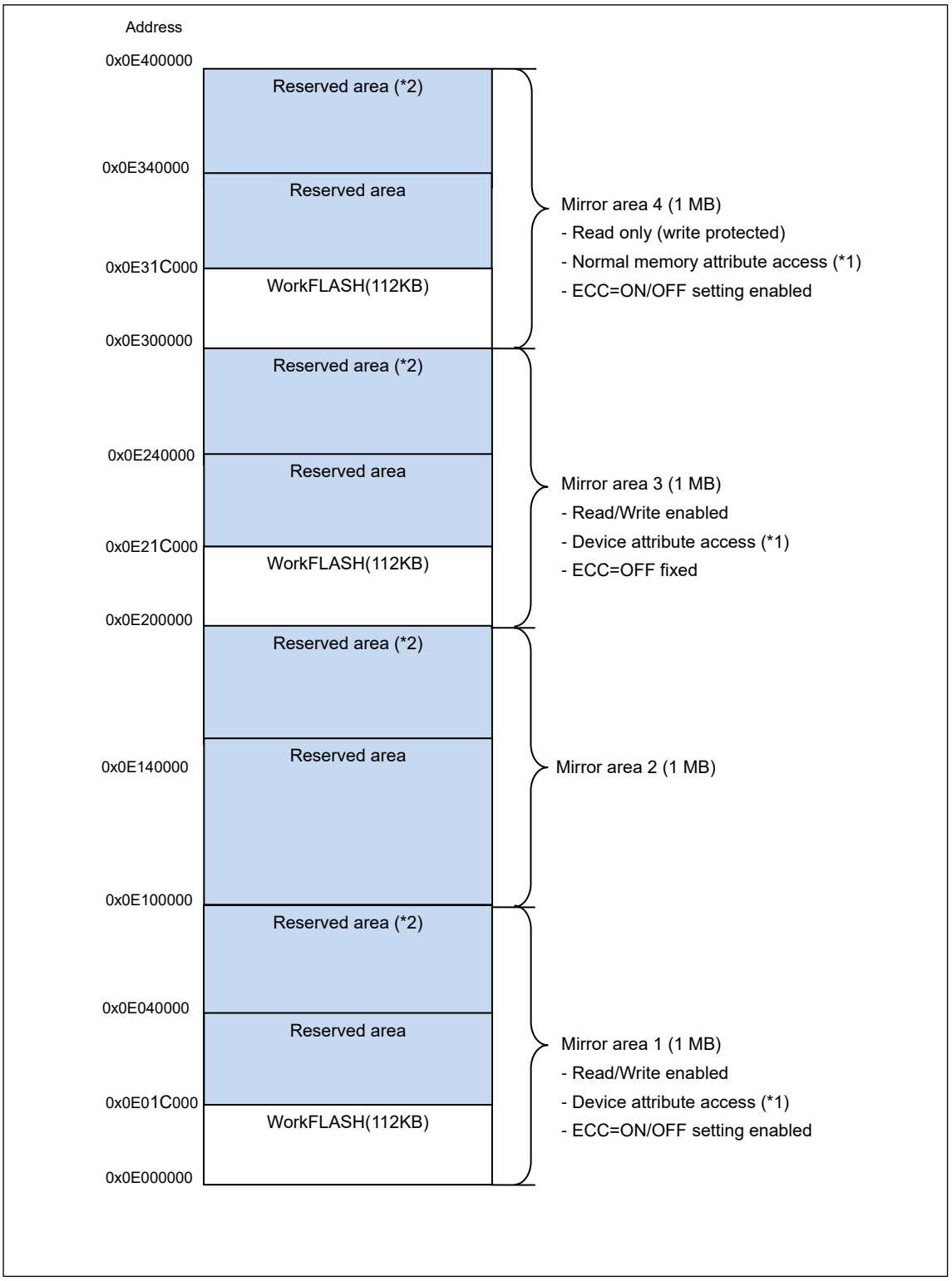
A single WorkFLASH has an area of 48 KB (or 112 KB or 240 KB depending on the assembly). The 48-KB area consists of 6 small sectors (or 14 or 30 sectors depending on the assembly), and the size of each small sector is 8 KB. These sectors are placed as shown in Figure 2-3. Their places in an overall address map are shown in Figure 2-2.

The following table shows the purpose for each area.

Area Number	Read Operation	Write Operation
Mirror area 1	Normal read	Write the value with ECC calculated
Mirror area 2	Reserved area (access prohibited)	Reserved area (access prohibited)
Mirror area 3	Normal read	Write the value without ECC
Mirror area 4	Normal memory attribute read	Access prohibited

* In order to perform read and write operation as shown above, the access attribute has to be configured for each mirror area by MPU.

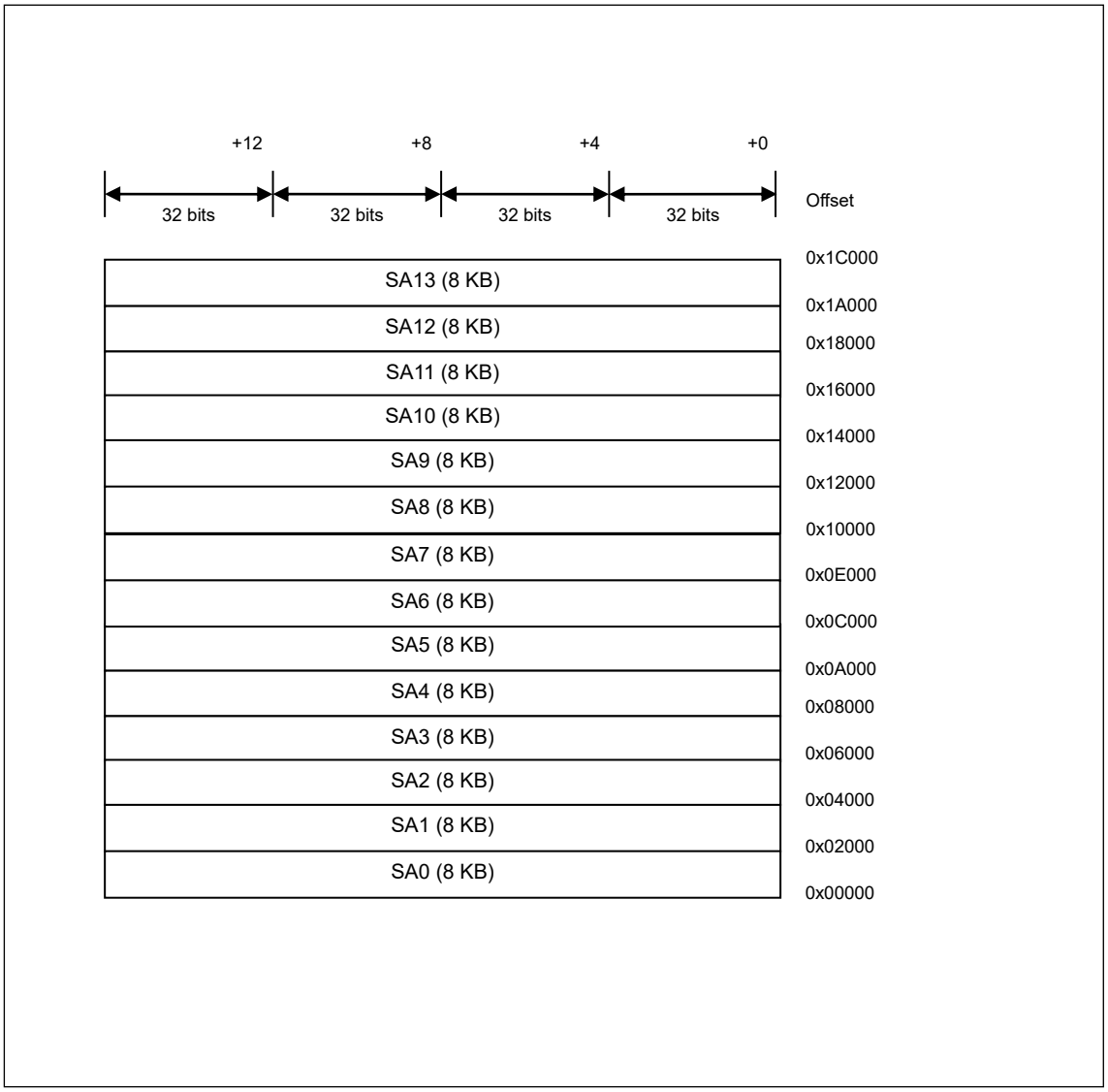
Figure 2-2 Address Map of WorkFLASH in User Mode



(*1) Access attributes for each mirror area must be set in the MPU.

(*2) Accessing this region results in a bus error, but the reserved area access bit (WFCFG_BERR:RESA) is not set.

Figure 2-3 Sector Configuration of WorkFLASH in User Mode



3. Explanation of Operation

This section explains the operation of WorkFLASH.

3.1. Read

Reading from WorkFLASH can be done with a size of 8 bits/16 bits/32 bits/64 bits.

However, if reading is attempted while the command sequencer is operating, WorkFLASH makes a bus error response.

If reading is attempted from an existing reserved area inside mirror areas 1 to 4, WorkFLASH makes a bus error response. Thus, any attempt by software to read a reserved area causes a data abort to occur.

If the ECCOFF bit in the WFCFG_ECR register is "0", an ECC check is performed during reading via mirror area 1 or 4. If the check finds a 1-bit error, the error is corrected, and the reading itself ends normally. The detection of the 1-bit error can be posted with an interrupt. If the check detects a 2-bit error, WorkFLASH makes a bus error response.

No ECC check is performed for reading via mirror area 3. The value of the ECCOFF bit in this WFCFG_ECR register in such cases has no significance.

3.2. Write and Erase

When data is written to an address that is not in a reserved area of mirror area 1 or mirror area 3, the WorkFLASH hardware automatically sends the programming access sequence of the write command to Flash memory.

Overwriting of the same data bit is prohibited. (Even overwriting of 1 -> 0, 0 -> 0, or 1 -> 1 is also prohibited.)

After erase operation is completed, all the values of the flash memory cells in the target sectors are "1". The Flash security function does not allow writing and erasing of protected sectors. For details, see the security specifications. If any of these sectors is written to or erased, a bus error response is made.

If the ECCOFF bit in the WFCFG_ECR register is "0", WorkFLASH generates and writes an ECC for 64-bit writing via mirror area 1. In any attempt to write with a width of less than 64 bits, WorkFLASH reads the corresponding 64-bit data first and then replaces the bit portion to be written. After that, it creates 64-bit data, write the data to be written, and writes the ECC generated from the 64 bits.

WFCFG_WSR :ST[1:0]	00	01	01	10	00
Mirror area 1 (ECCOFF=0) 8/16/32-bit Write	IDLE	64-bit data Read	Write command being issued	Write command completion wait	IDLE
Mirror area 1 (ECCOFF=0) 64-bit Write	IDLE	-	Write command being issued	Write command completion wait	IDLE

If the ECCOFF bit in the WFCFG_ECR register is "1", writing is performed without reading and replacement of data, even when attempting to write in 8/16/32/64 bits to mirror area 1. An ECC is not generated and written.

For mirror area 3, writing with a size of 8/16/32/64 bits is possible. However, in writing via mirror area 3, no ECC is generated and written.

WFCFG_WSR :ST[1:0]	00	01	10	00
Mirror area 1 (ECCOFF=1) 8/16/32/64-bit Write	IDLE	Write command being issued	Write command completion wait	IDLE
Mirror area 3 8/16/32/64-bit Write	IDLE	Write command being issued	Write command completion wait	IDLE

Normally, for writing with an ECC and with a size of 32 bits or less, first write the 64-bit part by using mirror area 3. After that, write the last bit portion together with the ECC, by writing in mirror area 1. For the writing procedure, see "4.3 Writing Procedure."

Example: Writing [15:0] in area 3 -> Writing [31:16] in area 3 -> Writing [63:32] in area 1

An erase operation is executed with the ERS[7:0] bits in the WFCFG_SEQCM register specifying the sector to erase, and the OPC[2:0] bits specifying the erase operation.

The suspend and resume operations for sector erase are also specified in the OPC[2:0] bits to suspend and resume sector erase.

The command sequencer can handle one write operation at a time. If another write request is received during execution of the write command sequence, a bus error response is made. While the write command sequence is not being executed, the value of the ST[1:0] bits in the WFCFG_WSR register is "00".

If a hang up is detected when the command sequencer is operating during writing or sector erase, the command sequencer automatically performs a reset for the FLASH macro to restore operation. Furthermore, the OPC[1:0] bits can also execute the read/reset command.

3.3. Transfer of Write Data with DMA

WorkFLASH enables writing by DMA transfer.

When the DMAEN bit in the WFCFG_WCR register is "1", a DMA transfer request is generated each time that the write command sequencer becomes idle (the ST[1:0] bit of the WFCFG_WSR register is "00").

The command sequencer can use DMA transfer in block mode. However, for DMA transfer of data to the command sequencer, use block mode DMA transfer with a block size of 64 bits.

Notes:

- *If DMA transfer stops because of an error, follow the processing flow below.*

  1. *The detection of an error in WorkFLASH cancels a DMA transfer, but a WorkFLASH DMA transfer request is asserted again.*
     (Any write data that is written immediately before the error is detected is not guaranteed after the error is detected.)
  2. *A WorkFLASH interrupt request is asserted.*
  3. *From the CPU, write DMAEN="0" (DMA transfer release) in the WFCFG_WCR register.*
  4. *After step 3, WorkFLASH DMA transfer requests are disabled (initialized state).*
     (To perform a DMA transfer after step 4, set and perform it from the beginning.)

3.4. Inserting a Wait Cycle

Since WorkFLASH and TCFLASH use the same Dual Port Flash memory, the FAWC bit setting in the TCFCFG_FCFGR register can insert a wait cycle at the Flash memory access time. (See "4.1 Setting Wait Cycle Count" and "5.2 TCFLASH Configuration Register: TCFCFG_FCFGR" in the TCFLASH specifications.)

3.5. ECC Generation and Check

WorkFLASH adds an 8-bit error check code (ECC: Error Check Code) for every 64 bits, so that it can detect and correct 1-bit errors and detect 2-bit errors.

The ECC logic of WorkFLASH generates an ECC during writing and performs a syndrome check during reading, by the same means as the ECC made for a unit of 64 bits in the ARM(R) Cortex(TM)-R5F core.

During writing, an 8-bit ECC is generated for 64-bit data and written together with the data. For details on how to write data with an ECC added, see "4.3 Writing Procedure."

During reading, a syndrome is calculated for the following decision: "no error detected," "1-bit error detected," or "error of 2 or more bits detected." If the result is "error of 2 or more bits detected," the error cannot be corrected. If the Flash memory contents are erased, errors are never detected.

To test the check function, the ECC logic has a function to insert an error into the data and ECC read from Flash memory.

ECC can be turned on/off with the setting of the ECCOFF bit in the WFCFG_ECR register.

3.5.1. ECC Generation

During writing, the 8-bit check bits CB[7:0] are generated from the 64-bit write data D[63:0] according to the calculation formula shown in Table 3-1 ECC Calculation Formula. The generated check bits are exclusively ORed with 0xf3 and then written to Flash memory.

Table 3-1 ECC Calculation Formula

		Data Bit																													
		63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
CB0	XOR	1	0	1	1	0	1	0	0	1	1	0	1	0	0	0	1	1	0	1	1	0	1	0	0	1	1	0	1	0	0	0	1
CB1	XOR	0	0	0	1	0	1	0	1	0	1	0	1	0	1	1	1	0	0	0	1	0	1	0	1	0	1	0	1	0	1	1	1
CB2	XNOR	1	0	1	0	0	1	1	0	1	0	0	1	1	0	0	1	1	0	1	0	0	1	1	0	1	0	0	1	1	0	0	1
CB3	XNOR	0	0	1	1	1	0	0	0	1	1	1	0	0	0	1	1	0	0	1	1	1	0	0	0	1	1	1	0	0	0	1	1
CB4	XOR	1	1	0	0	0	0	0	0	1	1	1	1	1	1	0	0	1	1	0	0	0	0	0	0	1	1	1	1	1	1	0	0
CB5	XOR	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
CB6	XOR	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
CB7	XOR	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0

		Data Bit																													
		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CB0	XOR	0	1	0	0	1	0	1	1	0	0	1	0	1	1	1	0	0	1	0	0	1	0	1	1	0	0	1	0	1	1	1	0
CB1	XOR	0	0	0	1	0	1	0	1	0	1	0	1	0	1	1	1	0	0	0	1	0	1	0	1	0	1	0	1	0	1	1	1
CB2	XNOR	1	0	1	0	0	1	1	0	1	0	0	1	1	0	0	1	1	0	1	0	0	1	1	0	1	0	0	1	1	0	0	1
CB3	XNOR	0	0	1	1	1	0	0	0	1	1	1	0	0	0	1	1	0	0	1	1	1	0	0	0	1	1	1	0	0	0	1	1
CB4	XOR	1	1	0	0	0	0	0	0	1	1	1	1	1	1	0	0	1	1	0	0	0	0	0	0	1	1	1	1	1	1	0	0
CB5	XOR	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
CB6	XOR	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
CB7	XOR	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

3.5.2. Syndrome Calculation

During reading, the 8-bit check bits CB[7:0] are calculated from the data [63:0] read from Flash memory, according to the calculation formulas shown in Table 3-1. The calculated check bits are used together with the check bits EDOR[7:0] read from Flash memory to generate a syndrome S[7:0] according to the calculation formulas shown in Table 3-2.

Table 3-2 Syndrome Calculation Formulas

Bit	Calculation Formula
S[7]	~ (CB[7] ^ EDOR[7])
S[6]	~ (CB[6] ^ EDOR[6])
S[5]	~(CB[5] ^ EDOR[5])
S[4]	~(CB[4] ^ EDOR[4])
S[3]	(CB[3] ^ EDOR[3])
S[2]	(CB[2] ^ EDOR[2])
S[1]	~(CB[1] ^ EDOR[1])
S[0]	~ (CB[0] ^ EDOR[0])

3.5.3. Error Detection

Based on the calculated value of syndrome S[7:0], the following decision is made: "no error detected," "1-bit error detected," "2-bit error detected," or "error of 3 or more bits detected." Table 3-3 shows the relationship between syndrome values and decision results. The meanings of symbols used in the table are as follows.

- "+" : No error is detected.
- "C[n] (0≤n≤7)" : A 1-bit error is detected. The value of the check bit CB[n] is erroneous.
- "[m] (0≤m≤63)" : A 1-bit error is detected. The value of the data bit D[m] is erroneous.
- "T" : A 2-bit error is detected. It cannot be corrected.
- "M" : An error of 3 or more bits is detected. It cannot be corrected.

The 1-bit errors can be detected in all cases, and they can be corrected. The 2-bit errors can be detected in all cases, but they cannot be corrected. The errors of 3 or more bits may be detected in some cases, as shown in Table 3-3. However, they cannot be detected in all cases. Note that errors of 3 or more bits also cannot be corrected.

Table 3-3 Meanings of Syndrome Values

		Syndrome[7:4]															
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
Syndrome[3:0]	**0**	+	C4	C5	T	C6	T	T	62	C7	T	T	46	T	M	M	T
	1	C0	T	T	14	T	M	M	T	T	M	M	T	M	T	T	30
	2	C1	T	T	M	T	34	56	T	T	50	40	T	M	T	T	M
	3	T	18	8	T	M	T	T	M	M	T	T	M	T	2	24	T
	4	C2	T	T	15	T	35	57	T	T	51	41	T	M	T	T	31
	5	T	19	9	T	M	T	T	63	M	T	T	47	T	3	25	T
	6	T	20	10	T	M	T	T	M	M	T	T	M	T	4	26	T
	7	M	T	T	M	T	36	58	T	T	52	42	T	M	T	T	M
	8	C3	T	T	M	T	37	59	T	T	53	43	T	M	T	T	M
	9	T	21	11	T	M	T	T	M	M	T	T	M	T	5	27	T
	A	T	22	12	T	33	T	T	M	49	T	T	M	T	6	28	T
	B	17	T	T	M	T	38	60	T	T	54	44	T	1	T	T	M
	C	T	23	13	T	M	T	T	M	M	T	T	M	T	7	29	T
	D	M	T	T	M	T	39	61	T	T	55	45	T	M	T	T	M
	E	16	T	T	M	T	M	M	T	T	M	M	T	0	T	T	M
	F	T	M	M	T	32	T	T	M	48	T	T	M	T	M	M	T

3.6. Interrupts

WorkFLASH can generate interrupt requests in cases such as the following.

- When one write or erase operation is completed and the next one can begin
  An interrupt request is generated in WorkFLASH when the RDYINT bit in the WFCFG_SR register is "1" and the RDYIE bit in the WFCFG_ICR register is "1".
- When the hang up state is detected
  The hang up state resulting from a timeout detected while the command sequencer is operating causes the HANGINT bit in the WFCFG_SR register to change to "1". If the HANGIE bit in the WFCFG_ICR register is "1" at this time, an error interrupt request is generated in WorkFLASH.
- When the ECC logic detects a 1-bit error (correctable)
  An interrupt request is generated in WorkFLASH when the SECINT bit in the WFCFG_SECIR register is "1" and the SECIE bit in the WFCFG_SECIR register is "1".
- When write enable can be obtained
  The WERSTS bit in the FCFG_WARBR register is set to "1" when write enable is released and can be obtained from TCFLASH, WorkFLASH or SHE. A write enable release interrupt request is generated at this time.

3.7. Bus Error Response

WorkFLASH responds returns bus error responses to invalid bus accesses. The cause of bus errors can be checked with the bus error response status register (WFCFG_BERR). The causes of bus errors are the following.

Error Factor	WFCFG_BERR	
	bit	Field
- Setting the sector number that does not exist, and performing the sector erase.	11	ERSIGN
- Writing to a read-only register.	10	RORW
- Writing to the mirror area 4 (refer next table for detail condition when NWTM flag is set).	9	NWTM
- Command sequencer writes to WorkFLASH or perform a command from WFCFG_SEQCM register (after this action, write operations/commands are ignored) unless the command sequencer is idle. - Performing the sector erase suspend command or read/reset command while Flash is in the sector erase suspend state. - Performing write or read operation to the sector to be erased while Flash is in the sector erase suspend state. - Performing the sector erase suspend command while Flash is in the normal state. - Reading WorkFLASH while the command sequencer is operating.	8	ACCIGN
- Performing write operation to WFCFG_ECR more than once. - Attempting to write to a protected register by performing an invalid register write unlock sequence. The invalid sequences are described specifically below. - Writing to WFCFG_CPR two times in a row - Writing a wrong key value to WFCFG_CPR - Writing to following protected registers in the locked state. - WFCFG_CR - WFCFG_ECR - WFCFG_DBEIR - WFCFG_EEIR	7	ECRWL
- Writing to the registers other than below with unprivileged mode. - WFCFG_WCR - WFCFG_SEQCM - WFCFG_EMENR - WFCFG_ARBCLR - WFCFG_BERRCLR - FCFG_WARBR	6	UNACC
- Accessing the reserved area of WorkFLASH (reserved area in Flash and registers, refer next table for detail condition when RESA flag is set in access to WorkFLASH memory map)	5	RESA
- Setting the sector number that is protected to ERS bit of WFCFG_SEQCM. - Setting the number of last two sectors dedicated to SHE to WFCFG_SEQCM.	4	RWE
- Accessing Flash or registers with the prohibited data width.	2	SIZE
- Performing write or erase operation to the mirror area 1 or mirror area 3 while WE bit of WFCFG_CR is “0”. - Performing read/reset command while WE bit of WFCFG_CR is “0”. - Performing software reset while WE bit of WFCFG_CR is “0”.	1	CRWE
- ECC detects an uncorrectable bit error (bit error of more than one bit).	0	DED

Following table shows occurrence of bus error response in access to each mirror area addresses, with relevant field state in WFCFG_BERR (RESA bit and NWTM bit).

Area		Address	Write access			Read access		
			Bus error response	RESA bit	NWTM bit	Bus error response	RESA bit	NWTM bit
Mirror area 4	Reserved	0x0E3F_FFFF ~ 0x0E34_0000	Yes	Not set	Not set	Yes	Not set	Not set
		0x0E33_FFFF ~ 0x0E32_0000	Yes	Set	Not set	Yes	Set	Not set
		0x0E31_FFFF ~ 0x0E31_C000	Yes	Not set	Set	Yes	Set	Not set
	SA0-13	0x0E31_BFFF ~ 0x0E30_0000	Yes	Not set	Set	No	Not set	Not set
Mirror area 3	Reserved	0x0E2F_FFFF ~ 0x0E24_0000	Yes	Not set	Not set	Yes	Not set	Not set
		0x0E23_FFFF ~ 0x0E22_0000	Yes	Set	Not set	Yes	Set	Not set
		0x0E21_FFFF ~ 0x0E21_C000	Yes	Set	Not set	Yes	Set	Not set
	SA0-13	0x0E21_BFFF ~ 0x0E20_0000	No	Not set	Not set	No	Not set	Not set
Mirror area 2	Reserved	0x0E1F_FFFF ~ 0x0E14_0000	Yes	Not set	Not set	Yes	Not set	Not set
		0x0E13_FFFF ~ 0x0E12_0000	Yes	Set	Not set	Yes	Set	Not set
		0x0E11_FFFF ~ 0x0E11_C000	Yes	Set	Not set	Yes	Set	Not set
	SA0-13	0x0E11_BFFF ~ 0x0E10_0000	Yes	Set	Not set	Yes	Set	Not set
Mirror area 1	Reserved	0x0E0F_FFFF ~ 0x0E04_0000	Yes	Not set	Not set	Yes	Not set	Not set
		0x0E03_FFFF ~ 0x0E02_0000	Yes	Set	Not set	Yes	Set	Not set
		0x0E01_FFFF ~ 0x0E01_C000	Yes	Set	Not set	Yes	Set	Not set
	SA0-13	0x0E01_BFFF ~ 0x0E00_0000	No	Not set	Not set	No	Not set	Not set

3.8. WorkFLASH Access Restrictions

To prevent the data and programs in WorkFLASH from being read by third parties, operations on WorkFLASH are restricted. (For details, see the security specifications.)

4. Setting Procedure

This section explains the WorkFLASH setting procedures.

4.1. Setting the Number of Wait Cycles

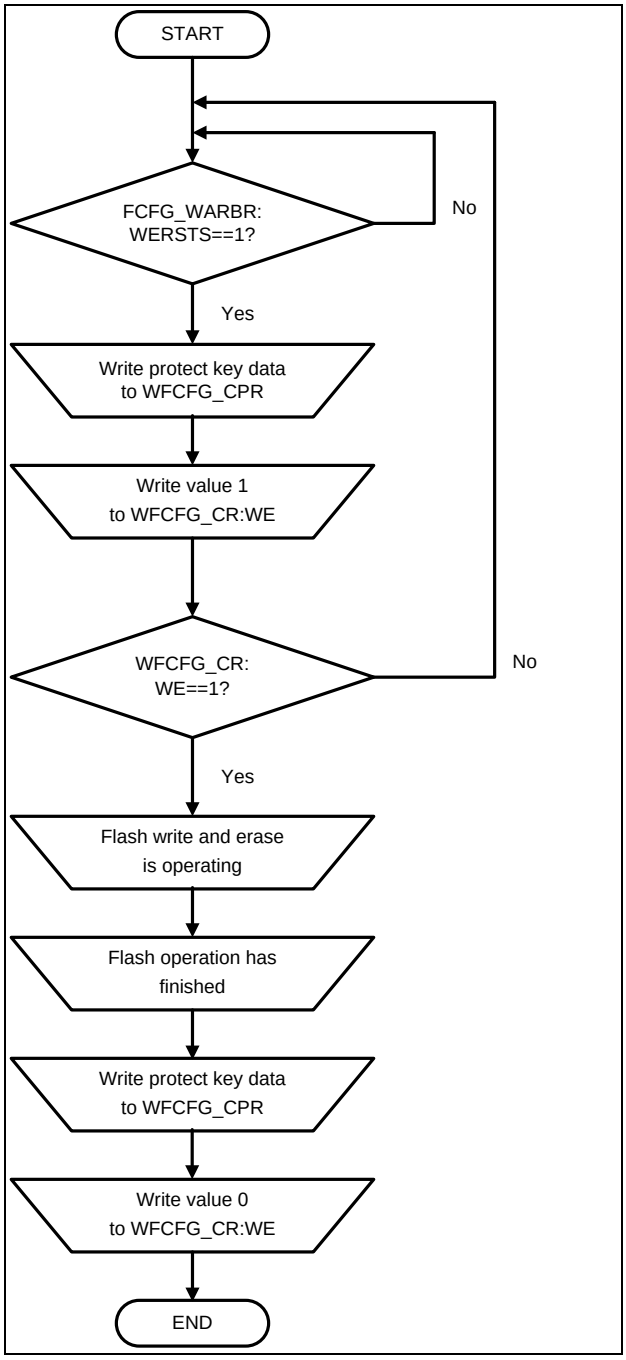
Since WorkFLASH and TCFLASH use the same Dual Port Flash memory, the FAWC bit setting in the TCFCFG_FCFGR register can insert a wait cycle at the Flash memory access time. For details on how to set it, see "4.1 Setting Wait Cycle Count" in the TCFLASH specifications.

4.2. Enabling Writing

To write data to WorkFLASH or erase its data, the write enable bit of WorkFLASH has to be set. Also, TCFLASH and WorkFLASH as well as SHE in the products cannot be simultaneously written and erased. Therefore, an arbitration function is included for the write enable of Flash interfaces (TCFLASH, WorkFLASH, SHE).

To write from WorkFLASH, write enable must be set through the following procedure.

Figure 4-1 Procedure for Setting Flash Write Enable



You can set write enable by using a write enable release (FCFG_WARBR:WERINT) interrupt instead of monitoring the write enable release status (FCFG_WARBR:WERSTS) bit.

To confirm the completion of Flash program/erase, check that the Programming/erasing ready bit (WFCFG _SR:RDY) has become “1”.

If writing and erasing are not required, set the write enable (WFCFG_CR:WE) bit to "0" to allow writing to Flash and erasing of Flash through the TCFLASH interface or SHE.

If write enable is obtained simultaneously, writing is enabled in accordance with the priority below:

1. SHE
2. WorkFLASH
3. TCFLASH

4.3. Writing Procedure

This section provides a procedure for data writing with an ECC attached. To execute writing with an ECC attached in the procedure shown in the figure below, the ECCOFF bit in the WFCFG_ECR register must be "0" (default value). In any attempt to write with a width of less than 64 bits, WorkFLASH reads the corresponding 64-bit data first and then replaces the bit portion to be written. After that, it creates 64-bit data, write the data to be written, and writes the ECC generated from the 64 bits.

To write only data without attaching an ECC, write the data via mirror area 3 instead of mirror area 1. In the writing via mirror area 3, the data is written but ECC data is not written.

Normally, for writing with an ECC and with a size of 32 bits or less, first write the 64-bit part without the ECC by using mirror area 3. After that, write the last bit portion together with the ECC by writing in mirror area 1. The following shows access for writing with 8/16/32/64 bits when ECC is enabled. Furthermore, the WFCFG_SR register is polled to wait for the completion of writing, as shown in Figure 4-2. Another method of detecting the completion is to use an RDY interrupt.

The Flash security function does not allow writing of protected sectors. For details, see the security specifications.

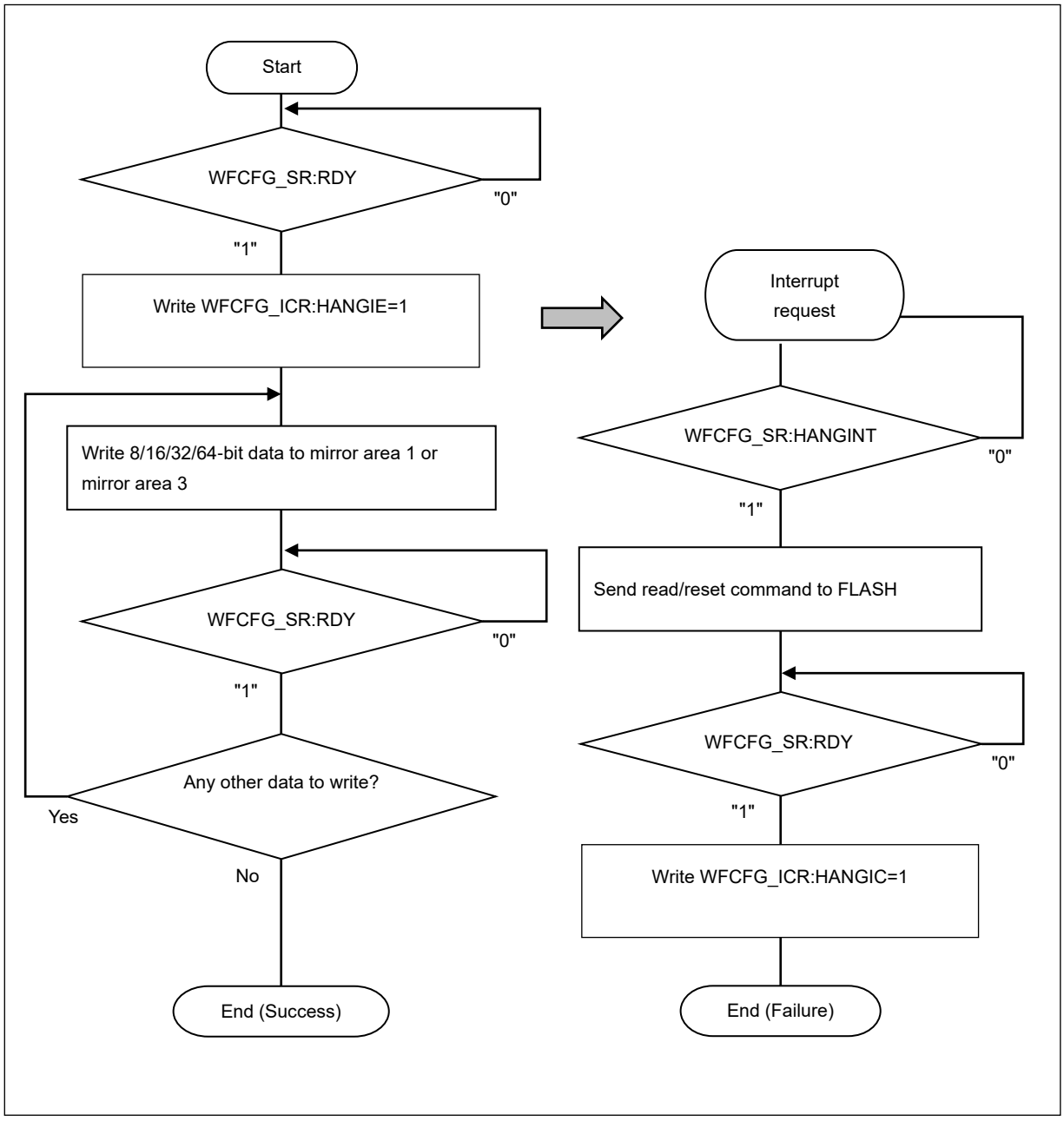
1. Writing with 64 bits when ECC is enabled
   - 1st time: Address=PA[mirror area 1], Data=PD64[63:0]

2. Writing with 32 bits when ECC is enabled
   - 1st time: Address=PA[mirror area 3], Data=PD32_0[31:0]
   - 2nd time: Address=PA[mirror area 1], Data=PD32_1[63:32]

3. Writing with 16 bits when ECC is enabled
   - 1st time: Address=PA[mirror area 3], Data=PD16_0[15:0]
   - 2nd time: Address=PA[mirror area 3], Data=PD16_1[31:16]
   - 3rd time: Address=PA[mirror area 3], Data=PD16_2[47:32]
   - 4th time: Address=PA[mirror area 1], Data=PD16_3[63:48]

4. Writing with 8 bits when ECC is enabled
   - 1st time: Address=PA[mirror area 3], Data=PD8_0[7:0]
   - 2nd time: Address=PA[mirror area 3], Data=PD8_1[15:8]
   - 3rd time: Address=PA[mirror area 3], Data=PD8_2[23:16]
   - 4th time: Address=PA[mirror area 3], Data=PD8_3[31:24]
   - 5th time: Address=PA[mirror area 3], Data=PD8_4[39:32]
   - 6th time: Address=PA[mirror area 3], Data=PD8_5[47:40]
   - 7th time: Address=PA[mirror area 3], Data=PD8_6[55:48]
   - 8th time: Address=PA[mirror area 1], Data=PD8_7[63:56]

Figure 4-2 Writing Procedure



The time it takes for HANGINT to become “1” is not equal to the Flash max writing time defined in datasheet. For example, when max writing time is defined as 400us in datasheet, it does not indicate HANGINT is set 400us after writing the data. HANGINT is designed so that the time for it to set is relatively longer than the max writing time defined in the datasheet. Hence, do not design the program that expects HANGINT=1 after the max writing time defined in datasheet (the same thing applies to sector erase).

If a write operation fails with HANGINT=1, perform re-writing after erasing the corresponding sector.

4.4. Sector Erasing Procedure

Writing the necessary information in the WFCFG_SEQCM register makes it possible to erase in units of sectors.

The command sequencer starts the sector erase operation on the specified sector as a result of the following. The sector number of the sector to erase has been set in the ERS[7:0] bits, and data with the OPC[2:0] bits set to 0b010 (code representing erase) has been written to the WFCFG_SEQCM register.

The sector erase can be suspended. During the sector erase, the command sequencer starts the sector erase suspend operation on the sector being erased when data with the OPC[2:0] bits set to 0b011 (code representing erase suspend) is written to the WFCFG_SEQCM register.

You can verify the suspension of the sector erase by checking whether the WFCFG_SR:ESPS bit is "1" and the WFCFG_SR:RDY bit is "1". Therefore, before suspending sector erase to write data to WorkFLASH or read its data, verification of the values of these bits is necessary.

To resume sector erase, the command sequencer starts the sector erase resume operation on the sector of the suspended erase when data that includes 0b010 (code representing erase) is written to the WFCFG_SEQCM register.

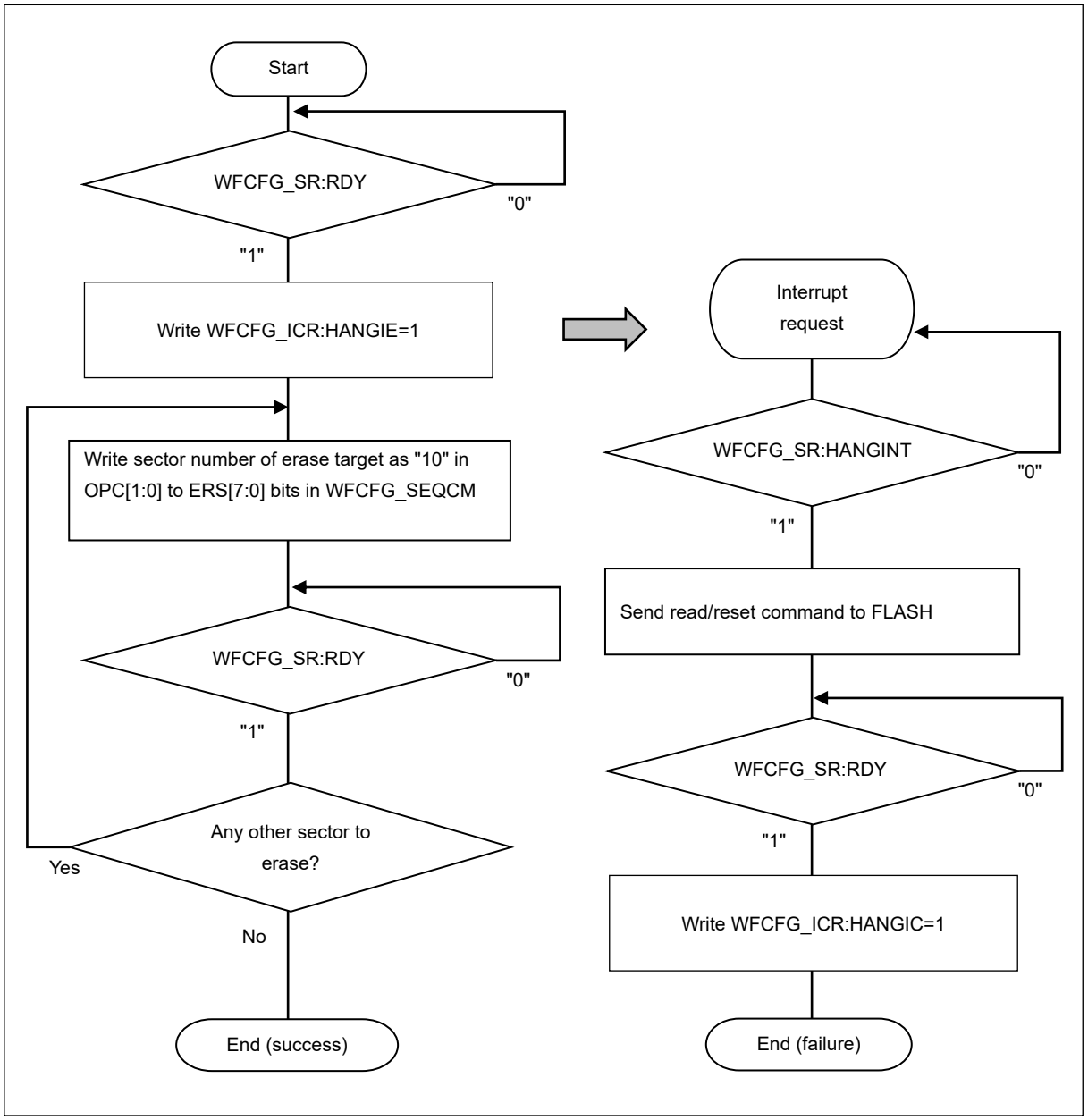
You can verify the completion of the sector erase operation by checking whether the value of the RDY bit in WFCFG_SR is "1". You can check the sector erase execution status with the SERS bit and ESPS bit in WFCFG_SR.

You can verify a sector erase timeout with the value of the HANGINT bit in WFCFG_SR.

When sector erase is suspended, sectors other than the sector to be erased can be read and written. The sector to be erased can be neither read nor written.

The Flash security function does not allow erasing of protected sectors. For details, see CHAPTER: Security.

Figure 4-3 Flow of Erasing a Sector



The time it takes for HANGINT to become “1” is not equal to the Flash max sector erase time defined in datasheet. For example, when max sector erase time is defined as 900ms in datasheet, it does not indicate HANGINT is set 900ms after the initiation of sector erase operation. HANGINT is designed so that the time for it to set is relatively longer than the max sector erase time defined in the datasheet. Hence, do not design the program that expects HANGINT=1 after the max sector erase time defined in datasheet.

5. Registers

This section explains the registers in WorkFLASH.

Register List

The register areas of WorkFLASH are in the peripheral function area, and they have a size of 1 KB.

Table 5-1 shows the placement of registers in the register areas.

Table 5-1 Register Map

Abbreviated Register Name	Description	Reference
WFCFG_CPR	WorkFLASH configuration protection key register	5.1
WFCFG_CR	WorkFLASH configuration register	5.2
WFCFG_ECR	WorkFLASH ECC control register	5.3
WFCFG_WCR	WorkFLASH write command sequencer configuration register	5.4
WFCFG_WSR	WorkFLASH write command sequencer status register	5.5
WFCFG_DBEIR	WorkFLASH data bit error injection register	5.6
WFCFG_EEIR	WorkFLASH ECC bit error injection register	5.7
WFCFG_ICR	WorkFLASH interrupt control register	5.8
WFCFG_SR	WorkFLASH status register	5.9
WFCFG_SECIR	WorkFLASH SEC interrupt register	5.10
WFCFG_EEAR	WorkFLASH ECC error address register	5.11
WFCFG_EMENR	WorkFLASH extra mode enable register	5.12
WFCFG_SEQCM	WorkFLASH sequencer command register	5.13
WFCFG_ARBERR	WorkFLASH arbitration error register	5.14
WFCFG_ARBCLR	WorkFLASH arbitration error clear register	5.15
WFCFG_BERR	WorkFLASH bus error response status register	5.16
WFCFG_BERRCLR	WorkFLASH bus error response status clear register	5.17
WFCFG_UCESR	WorkFLASH uncorrectable error status register	5.18
WFCFG_UCEAR	WorkFLASH uncorrectable error address register	5.19
FCFG_WARBR	FLASH write arbitration register	5.20

Table 5-2 Register Memory Map

Offset	Abbreviated Register Name	Description
0x000	WFCFG_CPR 00000000_00000000_00000000_00000000	WorkFLASH configuration protection key register
0x004	-	Reserved area
0x008	WFCFG_CR 00000000_00000000_00000000_00001001	WorkFLASH configuration register
0x00C	WFCFG_ECR 00000000_00000000_00000000_00000000	WorkFLASH ECC control register
0x010	WFCFG_WCR 00000000_00000000_00000001_00000000	WorkFLASH command sequencer configuration register
0x014	WFCFG_WSR 00000000_00000000_00000000_00000000	WorkFLASH command sequencer status register
0x018	-	Reserved area (moved for 64-bit migration)
0x01C	WFCFG_EEIR 00000000_00000000_00000000_00000000	WorkFLASH ECC bit error injection register
0x020	-	Reserved area
0x024	WFCFG_ICR 00000000_00000000_00000000_00000000	WorkFLASH interrupt control register
0x028	WFCFG_SR 00000000_00000000_00000000_00000001	WorkFLASH status register
0x02C	WFCFG_SECIR 00000000_00000000_00000000_00000000	WorkFLASH SEC interrupt register
0x030	WFCFG_EEAR 00000000_00000000_00000000_00000000	WorkFLASH ECC error address register
0x034	-	Reserved area
0x038	WFCFG_EMENR 00000000_00000000_00000000_00000000	WorkFLASH extra mode enable register
0x03C to 0x04C	-	Reserved area
0x050	-	Reserved area
0x054	WFCFG_SEQCM 00000000_11111111_00000000_00000000	WorkFLASH sequencer command mode register
0x058	WFCFG_ARBERR 00000000_00000000_00000000_00000000	WorkFLASH arbitration error register
0x05c	WFCFG_ARBCLR 00000000_00000000_00000000_00000000	WorkFLASH arbitration error clear register
0x060	WFCFG_BERR 00000000_00000000_00000000_00000000	WorkFLASH bus error response status register
0x064	WFCFG_BERRCLR 00000000_00000000_00000000_00000000	WorkFLASH bus error response status clear register
0x068	FCFG_WARBR 00000001_00000000_00000000_00000000	FLASH write arbitration register
0x06C	WFCFG_UCESR 00000000_00000000_00000000_00000000	WorkFLASH uncorrectable error status register

Offset	Abbreviated Register Name	Description
0x070	WFCFG_UCEAR 00000000_00000000_00000000_00000000	WorkFLASH uncorrectable error address register
0x074	-	Reserved area
0x078	WFCFG_DBEIR (lower32) 00000000_00000000_00000000_00000000	WorkFLASH data bit error injection register, lower 32 bits
0x07c	WFCFG_DBEIR (upper32) 00000000_00000000_00000000_00000000	WorkFLASH data bit error injection register, upper 32 bits
0x080 to 0x3FC	-	Reserved area

5.1. WorkFLASH Configuration Protection Key Register: WFCFG_CPR

The WorkFLASH configuration protection key register (WFCFG_CPR) is used to protect the following registers from unintended writing:

- WorkFLASH configuration register (WFCFG_CR)

- WorkFLASH ECC control register (WFCFG_ECR)

-WorkFLASH data bit error injection register (WFCFG_DBEIR)

- WorkFLASH ECC bit error injection register (WFCFG_EEIR)

Before writing to these registers, write the correct configuration protection key value (0xCF6DF1A5) to the WorkFLASH configuration protection key register of the same unit to set the unlock state. When unlocked, writing (setting change) is enabled.

After being unlocked, writing to the above registers is done. Then, it is locked again.

For any attempted writing to the above registers using an invalid procedure, WorkFLASH makes a bus error response. For example, suppose the configuration protection key value is incorrect, or an attempt is made to write to the above registers without unlocking writing. For access with a CPU, a data abort exception is generated because of the bus error response.

Be sure to write 32 bits to the WorkFLASH configuration protection key register.

bit	31	0
Field	CPR[31:0]	
R/W Attribute	R, W	
Protection Attribute	WP	
Initial Value	00000000_00000000_00000000_00000000	

[bit31:0] CPR[31:0]: Configuration Protection Key

Writing to the following protected registers is unlocked when the correct configuration protection key value (0xCF6DF1A5) is written to this register so that new values can be set:

- WorkFLASH configuration register (WFCFG_CR)
- WorkFLASH ECC control register (WFCFG_ECR)
- WorkFLASH data bit error injection register (WFCFG_DBEIR)
- WorkFLASH ECC bit error injection register (WFCFG_EEIR)

When writing to these registers is unlocked, 0xFFFFFFFF is read from this register. When writing to these registers is locked, 0x00000000 is read from this register.

5.2. WorkFLASH Configuration Register: WFCFG_CR

The WorkFLASH configuration register has the following functions for controlling the Flash memory in WorkFLASH:

- Enabling writing to the Flash memory

- Resetting the Flash memory

The setting of this register can be changed only when the correct configuration protection key value is written to the WorkFLASH configuration protection key register and writing to this register is unlocked.

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0, WX							
Protection Attribute	WPS							
Initial Value	00000000							

bit	23	22	21	20	19	18	17	16
Field	Reserved							SWFRST
R/W Attribute	R0, WX							R0, W
Protection Attribute	WPS							
Initial Value	0000000							0

bit	15	14	13	12	11	10	9	8
Field	Reserved							WE
R/W Attribute	R0, WX							R,W
Protection Attribute	WPS							
Initial Value	0000000							0

bit	7	6	5	4	3	2	1	0
Field	Reserved						Reserved	Reserved
R/W Attribute	R0, WX						R1, WX	R0, WX
Protection Attribute	WPS							
Initial Value	000000						1	0

[bit31:17] Reserved: Reserved Bits

[bit16] SWFRST: Software Reset

SWFRST	Description
0	There is no effect on operation.
1	Reset the Flash memory and the command sequencer.

When the TCFLASH interface or SHE has obtained write enable, even writing "1" to this bit cannot reset the Flash memory. When write enable has not been obtained, writing "1" to this bit causes a bus error.

If "1" is written to this bit at the same time as write enable is obtained, no reset is issued to the Flash memory. It causes a bus error response. To issue a reset to the Flash memory, write enable must be obtained before the writing of "1" in this bit. For the flow of obtaining write enable, see "4.2 Enabling Writing."

[bit15:9] Reserved: Reserved Bits

[bit8] WE: Write Enable

This bit enables/disables Flash memory writing/erasing. The detection of any writing or erasing of the Flash memory when this bit is "0" causes a bus error response by WorkFLASH.

If another Flash interface has obtained write enable, write enable cannot be set here. In this case, even the writing of "1" would not be reflected, and the read value would be "0". For details on how to set write enable, see 4.2 Enabling Writing.

WE	Description
0	Disable writing and erasing.
1	Enable writing and erasing.

[bit7:0] Reserved: Reserved Bits

5.3. WorkFLASH ECC Control Register: WFCFG_ECR

This register is used to control the operation of the ECC logic. The setting of this register can be changed only when the correct configuration protection key value is written to the WorkFLASH configuration protection key register and writing to this register is unlocked. You can write this register only once. For any second or subsequent writing to this register, WorkFLASH makes a bus error response (ECRWL).

bit	31	8
Field	Reserved	
R/W Attribute	R0, WX	
Protection Attribute	WPS	
Initial Value	00000000_00000000_00000000	

bit	7	6	5	4	3	2	1	0
Field	Reserved							ECCOFF
R/W Attribute	R0, WX							R/W
Protection Attribute	WPS							
Initial Value	0000000							0

[bit31:1] Reserved: Reserved Bits

[bit0] ECCOFF: ECC Off

This bit enables or disables ECC generation and check by access via mirror area 1 or 4. If ECC generation and check are disabled, no ECC is generated and no ECC check is performed even during access via mirror area 1 or 4.

ECCOFF	Description
0	Generate an ECC and perform an ECC check by access via mirror area 1 or 4 .
1	Neither generate an ECC nor perform an ECC check by access via mirror area 1 or 4.

5.4. WorkFLASH Write Command Sequencer Configuration Register: WFCFG_WCR

This register can set whether to operate the write command sequencer and write using DMA transfer.

bit	31	16
Field	Reserved	
R/W Attribute	R0, WX	
Protection Attribute	-	
Initial Value	00000000_00000000	

bit	15	14	13	12	11	10	9	8
Field	Reserved							Reserved
R/W Attribute	R0, WX							R/W1
Protection Attribute	-							
Initial Value	0000000							1

bit	7	6	5	4	3	2	1	0
Field	Reserved							DMAEN
R/W Attribute	R0, WX							R/W
Protection Attribute	-							
Initial Value	0000000							0

[bit31:1] Reserved: Reserved Bits

[bit0] DMAEN: DMA Enable

When this bit is "1" and the ST[1:0] bits in the WorkFLASH write command sequencer status register is "00", WorkFLASH generates a DMA transfer request and makes a request to the DMAC for the data to be written next.

DMAEN	Description
0	Do not generate a DMA transfer request.
1	Generate a DMA transfer request.

5.5. WorkFLASH Write Command Sequencer Status Register: WFCFG_WSR

Reading this register can provide information on the status of the write command sequencer.

This register is read-only. Writing to this register returns a bus error response (RORW).

bit	31	8
Field	Reserved	
R/W Attribute	R0, WX	
Protection Attribute	-	
Initial Value	00000000_00000000_00000000	

bit	7	6	5	4	3	2	1	0
Field	Reserved						ST[1:0]	
R/W Attribute	R0, WX						R, WX	
Protection Attribute	-							
Initial Value	000000						00	

[bit31:2] Reserved: Reserved Bits

[bit1:0] ST[1:0]: Write Command sequencer Status

These bits represent the status of the write command sequencer. Do not access Flash memory when the write command sequencer is in a state other than the idle state.

ST[1:0]	Description
0b00	Indicates that the write command sequencer is in the idle state.
0b01	Indicates that the write command sequencer is sending a command sequence to Flash memory.
0b10	Indicates that the write command sequencer is waiting for the completion of a Flash memory operation.
0b11	Reserved. This state never occurs.

5.6. WorkFLASH Data Bit Error Injection Register: WFCFG_DBEIR

This register is used to perform an ECC logic operation test by injecting errors into the data bits read from Flash memory.

The setting of this register can be changed only when the correct configuration protection key value is written to the WorkFLASH configuration protection key register and writing to this register is unlocked.

bit	63	0
Field	DBEIR[63:0]	
R/W Attribute	R/W	
Protection Attribute	WPS	
Initial Value	00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000	

[bit63:0] DBEIR[63:0]: Data Bit Error Injection Location

These bits send a calculated value to the ECC check logic. The value is the exclusive OR of the value in this register and the value of the data bits read from Flash memory. (This is a 64-bit register.)

DBEIR[i] (63≥i≥0)	Description
0	Send the bits at the same location as the data read from Flash memory, to the ECC check logic as they are.
1	Invert the bits at the same location as the data read from Flash memory, before sending them to the ECC check logic.

5.7. WorkFLASH ECC Bit Error Injection Register: WFCFG_EEIR

This register is used to perform an ECC logic operation test (dummy normal test) by injecting errors into the ECC bits read from Flash memory.

The setting of this register can be changed only when the correct configuration protection key value is written to the WorkFLASH configuration protection key register and writing to this register is unlocked.

bit	31							8
Field	Reserved							
R/W Attribute	R0, WX							
Protection Attribute	WPS							
Initial Value	00000000_00000000_00000000							
bit	7	6	5	4	3	2	1	0
Field	EEIR[7:0]							
R/W Attribute	R/W							
Protection Attribute	WPS							
Initial Value	00000000							

[bit31:8] Reserved: Reserved Bits

[bit7:0] EEIR[7:0]: ECC Bit Error Injection Location

These bits send a calculated value to the ECC check logic. The value is the exclusive OR of the value in this register and the value of the ECC data read from Flash memory.

EEIR[i] (7≥i≥0)	Description
0	Send the EEIR bits at the same location as the ECC data read from Flash memory, to the ECC check logic as they are.
1	Invert the EEIR bits at the same location as the ECC data read from Flash memory, before sending them to the ECC check logic.

5.8. WorkFLASH Interrupt Control Register: WFCFG_ICR

This register can enable interrupt request generation for each factor and clear an interrupt factor.

bit	31							16
Field	Reserved							
R/W Attribute	R0, WX							
Protection Attribute	WP							
Initial Value	00000000_00000000							

bit	15	14	13	12	11	10	9	8
Field	Reserved						HANGIC	RDYIC
R/W Attribute	R0, WX						R0, W	R0, W
Protection Attribute	WP							
Initial Value	000000						0	0

bit	7	6	5	4	3	2	1	0
Field	Reserved						HANGIE	RDYIE
R/W Attribute	R0, WX						R/W	R/W
Protection Attribute	WP							
Initial Value	000000						0	0

[bit31:10] Reserved: Reserved Bits

[bit9] HANGIC: Hang Interrupt Clear

HANGIC	Description
0	Writing "0" to this bit has no meaning.
1	Writing "1" to this bit clears the HANGINT bit in the WFCFG_SR register.

[bit8] RDYIC: Ready Interrupt Clear

RDYIC	Description
0	Writing "0" to this bit has no meaning.
1	Writing "1" to this bit clears the RDYINT bit in the WFCFG_SR register.

[bit7:2] Reserved: Reserved Bits

[bit1] HANGIE: Hang Interrupt Enable

HANGIE	Description
0	Disable generation of hang interrupt requests.
1	Enable generation of hang interrupt requests.

[bit0] RDYIE: Ready Interrupt Enable

RDYIE	Description
0	Disable generation of ready interrupt requests.
1	Enable generation of ready interrupt requests.

5.9. WorkFLASH Status Register: WFCFG_SR

This register is a read-only register that retains the state of Flash memory and individual interrupt factors. Writing to this register returns a bus error response (RORW).

bit	31	16
Field	Reserved	
R/W Attribute	R0, WX	
Protection Attribute	-	
Initial Value	00000000_00000000	

bit	15	14	13	12	11	10	9	8
Field	Reserved			WERINT	Reserved	Reserved	HANGINT	RDYINT
R/W Attribute	R0, WX			R, WX	R0, WX	R0, WX	R, WX	R, WX
Protection Attribute	-							
Initial Value	000			0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	Reserved		ESPS	Reserved	SERS	Reserved	Reserved	RDY
R/W Attribute	R0, WX		R, WX	R0, WX	R, WX	R0, WX	R, WX	R, WX
Protection Attribute	-							
Initial Value	00		0	0	0	0	0	1

[bit31:13] Reserved: Reserved Bits

[bit12] WERINT: Write Enable Release Interrupt

This register is a mirror of WERINT in "5.20 FLASH Write Arbitration Register: FCFG_WARBR". For details, see the description of the WERINT bit in "5.20 FLASH Write Arbitration Register: FCFG_WARBR"

WERINT	Description
0	No new write enable can be obtained.
1	This indicates that write enable can be obtained.

[bit11:10] Reserved: Reserved Bits

[bit9] HANGINT: Hang Up Interrupt

This bit indicates that a hang up interrupt request was generated because a transition of Flash memory to the hang up 1 state was detected. This bit is cleared when "1" is written to the HANGIC bit in the WFCFG_ICR register.

HANGINT	Description
0	No hang up interrupt request has been generated.
1	A hang up interrupt request has been generated.

[bit8] RDYINT: Ready Interrupt

This bit indicates that a ready interrupt request was generated because a transition of Flash memory to the ready state was detected. This bit is cleared when "1" is written to the RDYIC bit in the WFCFG_ICR register.

RDYINT	Description
0	No ready interrupt request has been generated.
1	A ready interrupt request has been generated.

[bit7:6] Reserved: Reserved Bits

[bit5] ESPS: Erase SusPend Status

This bit represents the WorkFLASH sector erase suspend status.

ESPS	Description
0	WorkFLASH has not suspended sector erase.
1	WorkFLASH received the sector erase suspend command.

This bit indicates the receipt of the sector erase suspend command. Therefore, you can verify the suspension of the sector erase by checking whether the WFCFG_SR:ESPS bit is "1" and the WFCFG_SR:RDY bit is "1". Therefore, before suspending sector erase to write data to WorkFLASH or read its data, verification of the values of these bits is necessary.

[bit4] Reserved: Reserved Bit

[bit3] SERS: Sector ERase Status

This bit represents the WorkFLASH sector erase status. When the write enable bit (WFCFG_CR:WE) is “0” (disabled), “0” is read from this bit.

SERS	Description
0	WorkFLASH is not erasing a sector.
1	WorkFLASH is erasing a sector.

[bit2:1] Reserved: Reserved Bits

[bit0] RDY: Ready

This bit indicates whether Flash memory is in the ready state. In the ready state, Flash memory can start the execution of a new operation.

RDY	Description
0	Flash memory is operating. In this state, it can receive only the read/reset command.
1	Flash memory can start the execution of the next command.

If Flash memory writing or erasing has failed (when WFCFG_SR:HANGINT="1"), RDY remains "0". Therefore, to perform new read, write, and erase operations after such a writing or erasing failure, it is necessary to send the read/reset command to Flash memory and verify that the RDY bit is set to "1".

5.10. WorkFLASH SEC Interrupt Register: WFCFG_SECIR

This register contains the status flags, enable bits, and clear bits related to 1-bit error correction interrupts.

bit	31	30	29	28	27	26	25	24
Field	SYN							
R/W Attribute	R, WX							
Protection Attribute	WP							
Initial Value	00000000							

bit	23	22	21	20	19	18	17	16
Field	Reserved							SECINT
R/W Attribute	R0, WX							R, WX
Protection Attribute	WP							
Initial Value	0000000							0

bit	15	14	13	12	11	10	9	8
Field	Reserved							SECIC
R/W Attribute	R0, WX							R0, W
Protection Attribute	WP							
Initial Value	0000000							0

bit	7	6	5	4	3	2	1	0
Field	Reserved							SECIE
R/W Attribute	R0, WX							R/W
Protection Attribute	WP							
Initial Value	0000000							0

[bit31:24] SYN: Syndrome

These bits store the syndrome from the detection of a 1-bit error.

[bit23:17]: Reserved Bits

[bit16] SECINT: 1-Bit Error Correction Interrupt

This bit indicates whether there is a 1-bit error correction interrupt request. If the ECC check during reading has detected and corrected a 1-bit error, a 1-bit error correction interrupt request is generated. Flash ECC is generated and detected for every 64bits. Note that if a read operation with the size of is 32-bit wide or less is performed, and an ECC error exists in other byte lanes than the read address, this bit is still set. This bit is read-only. Writing to this bit has no meaning. This bit is cleared when "1" is written to the SECIC bit in the same register.

SECINT	Description
0	A 1-bit error correction interrupt request has not been generated.
1	A 1-bit error correction interrupt request has been generated.

[bit15:9] Reserved: Reserved Bits

[bit8] SECIC: 1-Bit Error Correction Interrupt Clear

SECIC	Description
0	Writing "0" to this bit has no meaning.
1	Writing "1" to this bit clears the SECINT bit in the same register.

[bit7:1] Reserved: Reserved Bits

[bit0] SECIE: 1-Bit Error Correction Interrupt Enable

SECIE	Description
0	Disable generation of 1-bit error correction interrupt requests.
1	Enable generation of 1-bit error correction interrupt requests

5.11. WorkFLASH ECC Error Address Register: WFCFG_EEAR

This register retains the address at which a 1-bit error was detected during reading. If the 1-bit error was detected multiple times, the register retains the address at which the error was last detected.

This register is read-only. Writing to this register returns a bus error response (RORW).

bit	31	0
Field	EEAR[31:0]	
R/W Attribute	R, WX	
Protection Attribute	-	
Initial Value	00000000_00000000_00000000_00000000	

[bit31:0] EEAR[31:0]: Error Address

These bits have the address at which a 1-bit error was detected by the ECC check during reading. If the error was detected multiple times, they retain the address at which the error was last detected. Flash ECC is generated and detected for every 64bits. Note that if a read operation with the size of is 32-bit wide or less is performed, and an ECC error exists in other byte lanes than the read address, the address of the read operation is retained in this register.

5.12. WorkFLASH Extra Mode Enable Register: WFCFG_EMENR

This register can set an extra mode.

bit	31	16
Field	Reserved	
R/W Attribute	R0, WX	
Protection Attribute	-	
Initial Value	00000000_00000000	

bit	15	14	13	12	11	10	9	8
Field	Reserved							AEE
R/W Attribute	R0, WX							R/W
Protection Attribute	-							
Initial Value	0000000							0

bit	7	6	5	4	3	2	1	0
Field	Reserved							Reserved
R/W Attribute	R0, WX							R/W0
Protection Attribute	-							
Initial Value	0000000							0

[bit31:9] Reserved: Reserved Bits

[bit8] AEE: Read Arbitration Error Enable

AEE	Description
0	Disable read arbitration errors. In this mode, the execution of reading has to wait until the WFIF is selected for arbitration. If the higher-priority SHEIF side occupies the bus for a long time, there is a long wait for execution. In the worst case, this causes a watchdog timer reset.
1	In this mode, reading is completed without being blocked by arbitration, but it may be completed with an arbitration error. This bit can be disabled only if all buses are stopped and idle.

[bit7:0] Reserved: Reserved Bits

5.13. WorkFLASH Sequencer Command Register: WFCFG_SEQCM

This register is used to specify the execution of the read/reset command or the sector erase command for the command sequencer. If the execution of the command is completed with "1" set in the RDY bit in the WFCFG_SR register, this register is cleared.

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0, WX							
Protection Attribute	-							
Initial Value	00000000							

bit	23	22	21	20	19	18	17	16
Field	ERS[7:0]							
R/W Attribute	R, W							
Protection Attribute	-							
Initial Value	11111111							

bit	15	14	13	12	11	10	9	8
Field	Reserved							
R/W Attribute	R0, WX							
Protection Attribute	-							
Initial Value	00000000							

bit	7	6	5	4	3	2	1	0
Field	Reserved						OPC[1:0]	
R/W Attribute	R0, WX						R, W	
Protection Attribute	-							
Initial Value	000000						00	

[bit31:24] Reserved: Reserved Bits

[bit23:16] ERS[7:0]: Sector Erase Instruction

ERS[7:0]	Description
Initial Value (All"1")	This status has no selected sector. Disable sector erase.
Sector number exceeding number of implemented sectors	The number can be set, but there is a bus error response (ERSIGN) if the OPC bit specifies erase execution.
Sector number of implemented sector	Erase the specified sector. If SHE is supported, the last two sectors cannot be erased. Any attempt to set the last two sectors causes a bus error response (RWE). The sectors targeted for sector protection cannot be erased. Any attempt to set them causes a bus error response (RWE).

ERS[7:0] is reset to the initial value when the command specified by OPC[2:0] is complete. However, this reset does not occur if no operation is selected.

These bits are cleared with the completion of the sector erase suspend command, but the value of ERS[7:0] is ignored when resuming the sector erase with the sector erase command. Therefore, ERS[7:0] do not have to be reconfigured for resuming the sector erase.

The Flash security function does not allow erasing of protected sectors. For details, see the security specifications.

[bit15:2] Reserved: Reserved Bits

[bit1:0] OPC[1:0]: Command

OPC[1:0]	Description
2b00	No operation
2b01	Execute the read/reset command.
2b10	Execute the sector erase command. ERS[7:0] specifies the sector erase target. If sector erase has been suspended, the sector erase resume command is executed. When the sector erase resume command is executed, ERS[7:0] is ignored.
2b11	Execute the sector erase suspend command. When the sector erase suspend command is executed, ERS[7:0] is ignored.

OPC[1:0] is reset to the initial value when the command specified by OPC[1:0] is complete.

While WE bit of WFCFG_CR is "0", writing OPC other than "00" causes bus error.

5.14. WorkFLASH Arbitration Error Register: WFCFG_ARBERR

This register shows that a read request has been received while the SHE side was receiving an arbitration selection. This register is read-only. Writing to this register returns a bus error response (RORW).

bit	31	16
Field	Reserved	
R/W Attribute	R0, WX	
Protection Attribute	-	
Initial Value	00000000_00000000	

bit	15	14	13	12	11	10	9	8
Field	Reserved							
R/W Attribute	R0, WX							
Protection Attribute	-							
Initial Value	00000000							

bit	7	6	5	4	3	2	1	0
Field	Reserved							ARBERR
R/W Attribute	R0, WX							R,WX
Protection Attribute	-							
Initial Value	0000000							0

[bit31:1] Reserved: Reserved bits

[bit0] ARBERR: Read Arbitration Error

ARBERR	Description
0	No read arbitration error has occurred.
1	A read arbitration error has occurred. The read data that is returned when a read error occurs is all "1". This bit can be cleared by the writing of "1" to WFCFG_ARBCLR:ARBCLR.

5.15. WorkFLASH Arbitration Error Clear Register: WFCFG_ARBCLR

This register gives instructions to clear the ARBERR field in the arbitration error register (WFCFG_ARBERR).

bit	31	16
Field	Reserved	
R/W Attribute	R0, WX	
Protection Attribute	-	
Initial Value	00000000_00000000	

bit	15	14	13	12	11	10	9	8
Field	Reserved							
R/W Attribute	R0, WX							
Protection Attribute	-							
Initial Value	00000000							

bit	7	6	5	4	3	2	1	0
Field	Reserved							ARBCLR
R/W Attribute	R0, WX							R0,W
Protection Attribute	-							
Initial Value	0000000							0

[bit0] ARBCLR: Read Arbitration Error Clear

ARBCLR	Description
0	No effect
1	Clear the ARBERR flag in WFCFG_ARBERR.

5.16. WorkFLASH Bus Error Response Status Register: WFCFG_BERR

This register is a read-only register that retains the flags representing the factors of the bus error responses by WorkFLASH. Each flag can be cleared by the writing of "1" to the corresponding bit in the WFCFG_BERRCLR register.

This register is read-only. Writing to this register returns a bus error response (RORW).

bit	31	16
Field	Reserved	
R/W Attribute	R0, WX	
Protection Attribute	-	
Initial Value	00000000_00000000	

bit	15	14	13	12	11	10	9	8
Field	Reserved				ERSIGN	RORW	NWTM	ACCIGN
R/W Attribute	R0, WX				R, WX	R, WX	R, WX	R, WX
Protection Attribute	-							
Initial Value	0000				0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	ECRWL	UNACC	RESA	RWE	Reserved	SIZE	CRWE	DED
R/W Attribute	R, WX	R, WX	R, WX	R,WX	R0, WX	R, WX	R, WX	R, WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

[bit31:12] Reserved: Reserved Bits

[bit11] ERSIGN: ERS Writing Violation

ERSIGN	Description
0	No ERS writing violation has been detected.
1	This indicates that an ERS writing violation has been detected, followed by a bus error response, in the following case: - When writing to ERS[7:0] bits of WFCFG_SEQCM the number of a sector that is not equipped in the device, and attempting to perform the sector erase.

[bit10] RORW: Write Access to a Read-Only Register

RORW	Description
0	Write access to a read-only register has not been detected.
1	This indicates that a bus error response was made because write access to a read-only register was detected.

[bit9] NWTM: Writing to Mirror Area 4

NWTM	Description
0	Writing to mirror area 4 has not been detected.
1	This indicates that a bus error response was made because writing to mirror area 4 was detected. However, in case the reserved area is written, RESA bit is set instead of this bit.

[bit8] ACCIGN: Command Overrun

ACCIGN	Description
0	No command overrun has been detected.
1	This indicates that a new command was written to Flash memory during execution of the preceding command, and the subsequent command was ignored. At this time, WorkFLASH made a bus error response to the bus master that wrote the subsequent command. This value is also set by attempts to execute the following commands when sector erase is suspended: ・Execution of a command other than the sector erase resume command when Flash is in the sector erase suspend state ・Execution of a write/read command for a sector where sector erase is suspended

[bit7] ECRWL: Protection Sequence Violation

ECRWL	Description
0	Writing to the protected register in violation of the protection sequence has not been detected.
1	WorkFLASH has detected a violation of the protection sequence and made a bus error response, in the following cases: ・Writing twice in a row to the protection key register WFCFG_CPR ・Incorrect key value written to the protection key register ・Attempt to write to a protected register in the locked state ・Attempt to write to the ECCOFF bit in the WFCFG_ECR register a second or subsequent time

[bit6] UNACC: Unprivileged Writing

UNACC	Description
0	No unprivileged writing has been detected.
1	This indicates that a bus error response was made because writing to a register area in the unprivileged state was detected.

[bit5] RESA: Reserved Area Access

RESA	Description
0	Access to a reserved area has not been detected.
1	This indicates that a bus error response was made because access to a reserved area in Flash memory or the register area was detected.

[bit4] RWE: Write protected sector access error

RWE	Description
0	- Access to a protected sector has not been detected. - No ERS writing violation has been detected.
1	- This indicates that a bus error response was made because access to a protected sector was detected. - A sector assigned to SHE has been written to the ERS[7.0] bits in WFCFG_SEQCM.

[bit3] Reserved Bit

[bit2] SIZE: Access Size Violation

SIZE	Description
0	Access with an unsupported size has not been detected.
1	This indicates that access with an invalid size to Flash memory or a register has been detected, followed by a bus error response.

[bit1] CRWE: Writing Prohibition Violation

CRWE	Description
0	No writing prohibition violation has been detected.
1	This indicates one of the following. - A bus error response was made because of an attempt to write to Flash memory (mirror area 1 or mirror area 3) when the WE bit in the WFCFG_CR register was “0”. However, in case the reserved area is written, RESA bit is set instead of this bit. - A bus error response was made because of an attempt to execute software reset (WFCFG_CR:SWFRST) when the WE bit in the WFCFG_CR register was “0”.

[bit0] DED: Uncorrectable Error Detection

DED	Description
0	No uncorrectable error has been detected.
1	This indicates that an uncorrectable error was detected by the ECC check during reading, followed by a bus error response. Flash ECC is generated and detected for every 64bits. Note that if a read operation with the size of is 32-bit wide or less is performed, and an ECC error exists in other byte lanes than the read address, this bit is still set.

5.17. WorkFLASH Bus Error Response Status Clear Register: WFCFG_BERRCLR

This register can clear each flag representing the factor of the bus error response by WorkFLASH. To clear the flag, write "1" to the bit corresponding to the flag bit in the WFCFG_BERR register.

bit	31	16
Field	Reserved	
R/W Attribute	R0, WX	
Protection Attribute	-	
Initial Value	00000000_00000000	

bit	15	14	13	12	11	10	9	8
Field	Reserved				ERSIGN CLR	RORW CLR	NWTM CLR	ACCIGN CLR
R/W Attribute	R0, WX				R0, W	R0, W	R0, W	R0, W
Protection Attribute	-							
Initial Value	0000				0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	ECRWL CLR	UNAC CLR	RESA CLR	RWECLR	Reserved	SIZE CLR	CRWE CLR	DED CLR
R/W Attribute	R0, W	R0, W	R0, W	R0, W	R0, WX	R0, W	R0, W	R0, W
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

[bit31:12] Reserved: Reserved Bits

[bit11] ERSIGNCLR: ERSIGN Clear

ERSIGNCLR	Description
0	Writing "0" to this bit has no meaning.
1	Writing "1" to this bit clears the ERSIGN bit in the WFCFG_BERR register.

[bit10] RORWCLR: RORW Clear

RORWCLR	Description
0	Writing "0" to this bit has no meaning.
1	Writing "1" to this bit clears the RORW bit in the WFCFG_BERR register.

[bit9] NWTMCLR: NWTM Clear

NWTMCLR	Description
0	Writing "0" to this bit has no meaning.
1	Writing "1" to this bit clears the NWTM bit in the WFCFG_BERR register.

[bit8] ACCIGNCLR: ACCIGN Clear

ACCIGNCLR	Description
0	Writing "0" to this bit has no meaning.
1	Writing "1" to this bit clears the ACCIGN bit in the WFCFG_BERR register.

[bit7] ECRWLCLR: ECRWL Clear

ECRWLCLR	Description
0	Writing "0" to this bit has no meaning.
1	Writing "1" to this bit clears the ECRWL bit in the WFCFG_BERR register.

[bit6] UNACCLR: UNACC Clear

UNACCLR	Description
0	Writing "0" to this bit has no meaning.
1	Writing "1" to this bit clears the UNACC bit in the WFCFG_BERR register.

[bit5] RESACLR: RESA Clear

RESACLR	Description
0	Writing "0" to this bit has no meaning.
1	Writing "1" to this bit clears the RESA bit in the WFCFG_BERR register.

[bit4] RWECLR: RWE Clear

RWECLR	Description
0	Writing "0" to this bit has no meaning.
1	Writing "1" to this bit clears the RWE bit in the WFCFG_BERR register.

[bit3] Reserved Bit

[bit2] SIZECLR: SIZE Clear

SIZECLR	Description
0	Writing "0" to this bit has no meaning.
1	Writing "1" to this bit clears the SIZE bit in the WFCFG_BERR register.

[bit1] CRWECLR: CRWE Clear

CRWECLR	Description
0	Writing "0" to this bit has no meaning.
1	Writing "1" to this bit clears the CRWE bit in the WFCFG_BERR register.

[bit0] DEDCLR: DED Clear

DEDCLR	Description
0	Writing "0" to this bit has no meaning.
1	Writing "1" to this bit clears the DED bit in the WFCFG_BERR register.

5.18. WorkFLASH Uncorrectable Error Status Register: WFCFG_UCESR

This register contains fields representing information about uncorrectable error detection interrupts.

This register is read-only. Writing to this register returns a bus error response (RORW).

bit	31	30	29	28	27	26	25	24
Field	SYN							
R/W Attribute	R, WX							
Protection Attribute	-							
Initial Value	00000000							

bit	23	0
Field	Reserved	
R/W Attribute	R0, WX	
Protection Attribute	-	
Initial Value	00000000_00000000_00000000	

[bit31:24] SYN: Syndrome

These bits store the syndrome from the detection of a 2-bit error.

[bit23:0] Reserved: Reserved Bits

5.19. WorkFLASH Uncorrectable Error Address Register: WFCFG_UCEAR

This register retains the address at which an uncorrectable error was detected during reading. If the uncorrectable error was detected multiple times, the register retains the address at which the error was last detected.

This register is read-only. Writing to this register returns a bus error response (RORW).

bit	31	0
Field	UCEA[31:0]	
R/W Attribute	R, WX	
Protection Attribute	-	
Initial Value	00000000_00000000_00000000_00000000	

[bit31:0] UCEA[31:0]: Uncorrectable Error Address

These bits have the address at which an uncorrectable error was detected by the ECC check during reading. If the error was detected multiple times, they retain the address at which the error was last detected.

5.20. FLASH Write Arbitration Register: FCFG_WARBR

This register performs arbitration of write enable in order to write and erase Flash memory.

bit	31	25	24
Field	Reserved		WERSTS
Initial Value	R0, WX		R, WX
Protection Attribute	-		-
Initial Value	0000000		1

bit	23	17	16
Field	Reserved		WERINT
Initial Value	R0, WX		R, WX
Protection Attribute	-		-
Initial Value	0000000		0

bit	15	9	8
Field	Reserved		WERINTCLR
Initial Value	R0, WX		R0, W
Protection Attribute	-		-
Initial Value	0000000		0

bit	7	1	0
Field	Reserved		WERINTE
Initial Value	R0, WX		R/W
Protection Attribute	-		-
Initial Value	0000000		0

[bit31:25] Reserved: Reserved Bits

[bit24] WERSTS: Write Enable Release Status

WERSTS	Description
0	Since one of the Flash units has obtained write enable, a new write enable cannot be set. At this time, write enable is not set for TCFLASH or WorkFLASH, so even the writing of "1" to the write enable bit of WorkFLASH or SHE would not be reflected.
1	Write enable for Flash can be set.

[bit23:17] Reserved: Reserved Bits

[bit16] WERINT: Write Enable Release Interrupt

WERINT	Description
0	No new write enable can be obtained.
1	This indicates that write enable can be obtained.

This bit can be set only when WERINTE="1".

This register has the mirror described in "5.9. WorkFLASH Status Register: WFCFG_SR." Both have the same values.

[bit15:9] Reserved: Reserved Bits

[bit8] WERINTCLR: Write Enable Release Interrupt Clear

WERINTCLR	Description
0	Writing "0" to this bit has no meaning.
1	Writing "1" to this bit clears the WERINT bit in this register.

[bit7:1] Reserved: Reserved Bits

[bit0] WERINTE: Write Enable Release Interrupt

WERINTE	Description
0	Disable generation of write enable release interrupt requests.
1	Enable generation of write enable release interrupt requests.

6. Other

This section provides notes on using WorkFLASH.

Reset during Execution of Writing or Erasing

If the products is reset during execution of writing, the contents of the address targeted for writing are shown as undefined. Alternatively, if it is reset partway through a sector erase, the contents of the sector targeted for erasing are shown as undefined. In such cases, after the completion of the reset, retry suspended writing or erasing.

Reading from Flash Memory

Data cannot be read from WorkFLASH while writing or erasing of WorkFLASH is being executed. For this reason, copy the required data and programs from WorkFLASH to RAM before starting the write or erase operation, so that they do not need to be read partway through writing or erasing.

Waiting for the Completion of a Reset from Software

Flash memory is reset by the writing of "1" to the SWFRST bit in the WFCFG_CR register. After performing a reset on a unit, monitor the RDY bit in the WFCFG_SR register of the unit, and wait for the completion of the reset before accessing it.

To generate a software reset, the write enable (WFCFG_CR:WE) bit must be "1". A software reset cannot be generated in a state without writing enabled. A bus error response is made when a software reset could not be generated.

WorkFLASH Status Register Bits

Since some status information of flash memory is common to Code side (TCFLASH) and Work side (WorkFLASH) in Dual Port FLASH macro,
WorkFLASH status register bits listed below are affected by TCFLASH operation in addition to WorkFLASH operation:
- WFCFG_SR.ESPS

Use this bit only when program/erase is performed in WorkFLASH.

MPU Attribute/Write Access Type

The MPU attribute is supported the below table when each master writes to the Work Flash.

	CPU		DMA	DAP
	Normal	**Device**		
Dword	Support	Non-support	Support	Non-support
Word	Non-support	Support	Support	Support
Hword	Non-support	Support	Support	Support
Byte	Non-support	Support	Support	Support

7. Secure Hardware Extension (SHE) Support

This section describes the functions and restrictions relating to SHE support by WorkFLASH.

Overview

S6J3200 series containing 48 KB/112 KB/240 KB WorkFLASH have in total 6/14/30 sectors allocated for general use cases and two additional sectors which are not part of memory map but internally used only by SHE IP.The following two buses provided by WorkFLASH include SHE IP support:

- WorkFLASH interface (WFIF)
- SHE interface (SHEIF)

Hardware barrier

A hardware barrier is provided to prevent access from the WFIF to SHEIF sectors. A similar hardware barrier is provided to also prevent access from the SHEIF to WFIF sectors.

All attempts from the WFIF to access the sectors protected by the hardware barrier cause a bus error response and set the RESA bit in the WFCFG_BERR register.

Separation of Register sets

The WFIF and SHEIF settings and status registers have been implemented independently. The configuration registers for the WFIF can be set only in access from the WFIF side. Only the WFIF status registers are updated in access from the WFIF. SHEIF cannot be accessed from the user side.

Arbitration of FLASH Macro Access

An arbitration circuit is implemented so that the WFIF and SHEIF can access and share FLASH macros. The arbitration circuit uses a fixed preferred scheme for each transfer. The SHEIF has higher priority than the WFIF.

> Example: Even for a FLASH macro performing an AHB read with a burst length of 16, arbitration is performed on each and every transfer in the burst. This means that the read burst access can be interrupted by SHEIF access.

All FLASH macro accesses including command sequencer operations (write, sector erase, and read reset) behave similarly and independently of the arbitration state, excluding the exception of command execution operation time.

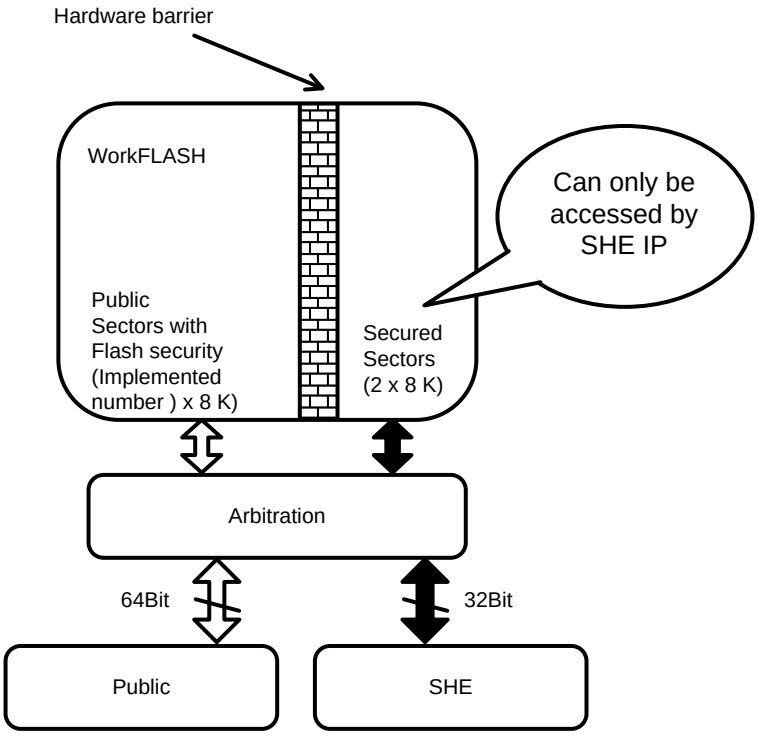
> Example: If the WFIF has valid write access, but the SHEIF has the current access right, the WFIF access is accepted, but its execution has to wait until the end of the SHEIF access.

Once the command sequencer starts execution of a command, arbitration is maintained until the end of the WFIF command. The behavior of read access by the WFIF varies depending on the arbitration setting and status.

- When WFCFG_EMENR:AEE=0
  - Read access stops until an arbitration selection is made and read data is returned from the FLASH macro.
- When WFCFG_EMENR:AEE=1
  - If the WFIF is selected for arbitration, read data is returned from the FLASH macro.
  - If the SHEIF is selected for arbitration, the WFCFG_ARBERR:ARBERR flag is set to 1, the arbitration error is indicated, and the returned read data is all "1". Such non-blocking read access helps prevent (prolonged) stoppages of the bus while the SHEIF operation of write or sector erase is under execution. After the ARBERR flag in WFCFG_ARBERR:ARBERR is set to "1" or after

multiple read accesses, an application must check whether normal data has been read. This means, when an access from SHEIF is made it is impossible to execute instruction codes on WorkFLASH.

Figure 7-1 Hardware Barrier Separating the WFIF and SHEIF Data Paths



8. Dual Port FLASH Macro Support

This section describes restrictions when using a Dual Port FLASH macro as a FLASH macro.

Differences from the Case of Using a Single Port FLASH Macro

- WorkFLASH uses the port on the Work side of Dual Port FLASH.
- Since the writing and erasing circuit is shared with the Code side, partial operation on the Code side affects the Work side.

Software control that takes the foregoing into consideration is necessary.

Availability of concurrent Dual Port Operation on the Code and Work Sides

	Code:Read/NOP Work:Read/NOP	Code:Read/NOP Work:PGM/ERS	Code:PGM/ERS Work:Read/NOP	Code:PGM/ERS Work:PGM/ERS
Operation available	OK	OK	OK	NG

Access to Dual Port on the Code and Work Sides

The Flash areas accessed by each interface are described below.

Interface	TCFLASH	WorkFLASH	SHE
Accessed Area	Code	Work	Work

CHAPTER 19: BootROM Hardware Interface

This chapter explains the BootROM hardware interface.

1. **Overview**
2. **Configuration**
3. **Explanation of Operation**
4. **Setting Procedure Example**
5. **Registers**
6. **Others**

BOOTROMHW-TXXPT03P01R01L05-E1-XX

1. Overview

This section provides an overview of the BootROM hardware interface.

The BootROM hardware interface is the interface between CPU and BootROM. The BootROM hardware interface sets the setting registers for the user-defined exception handler. The interface has 2 exception vector register sets. When one of the sets is in use, the setting of the other one can be changed.

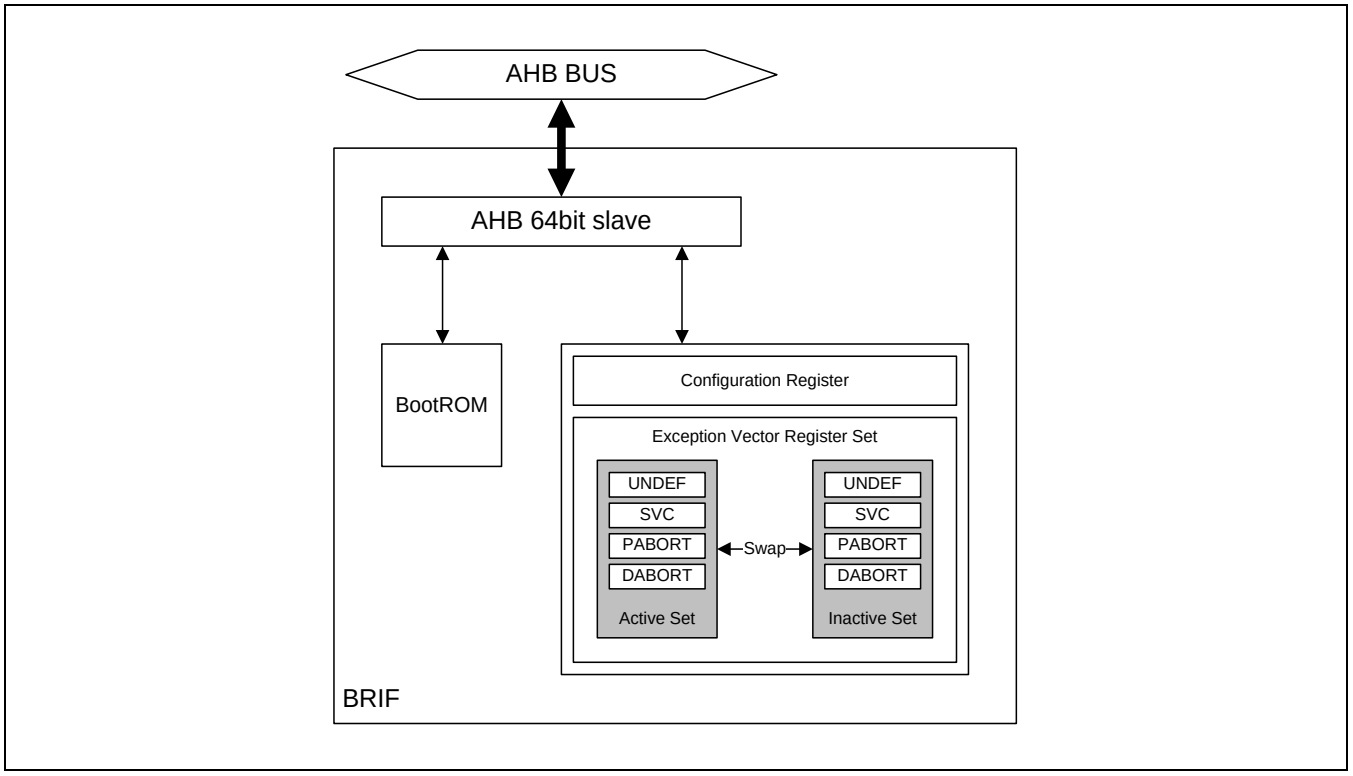
Notes:

- *In this chapter, the BootROM hardware interface is sometimes referred to by its abbreviation BRIF (BootRom hardware InterFace).*
- *Among the abbreviated register names of the BootROM hardware interface, "EXCFG" in "EXCFG_***" indicates exception vector configuration (EXception vector ConFiGuration).*

2. Configuration

This section explains a block diagram of the BootROM hardware interface.

Figure 2-1 BootROM Hardware Interface Block Diagram



3. Explanation of Operation

This section explains the operation of the BootROM hardware interface.

Sequence Protection by the Lock Release Register

The BootROM hardware interface provides the sequence protection function made available by the lock release register (EXCFG_UNLOCK). To update the exception vector configuration, the lock release value must be written in EXCFG_UNLOCK in advance to release locking. Write the lock value after update of the exception vector configuration to lock it again.

ARM Cortex-R5 Exception Vector Processing

ARM Cortex-R5 processes exceptions in the following steps:

1. Exception occurs.
2. The control jumps to the execution code of the corresponding entry of the fixed exception vector table of BootROM.
3. The load instruction at the exception entry loads the value of each exception vector register of PC.
4. The control jumps to the starting address of the exception handler, and the exception handler is executed.

The exception entry for the BootROM vector table is fixed. However, in order to make the jump destination changeable, the starting address of the ARM Cortex-R5 exception handler (inline literal) is stored in registers. These registers are mapped to addresses lower than BootROM, and are referred to by the BootROM exception entry when an exception occurs. This arrangement allows the exception vectors to be dynamically defined.

In order to allow all exception vector registers to be redefined by a single access, there are two register sets: active and inactive sets. These two sets are mapped to addresses lower than BootROM. The address to which the active set is mapped is at an active position that the BootROM exception entry refers to at the time of exception occurrence. The address to which the inactive set is mapped is at an inactive position, in which the values can be changed. Writing "1" to the exception vector register set swap bit of the setting register (EXCFG_CNFG:SWAP) swaps the contents of the active and inactive sets. This action changes the settings of the active set.

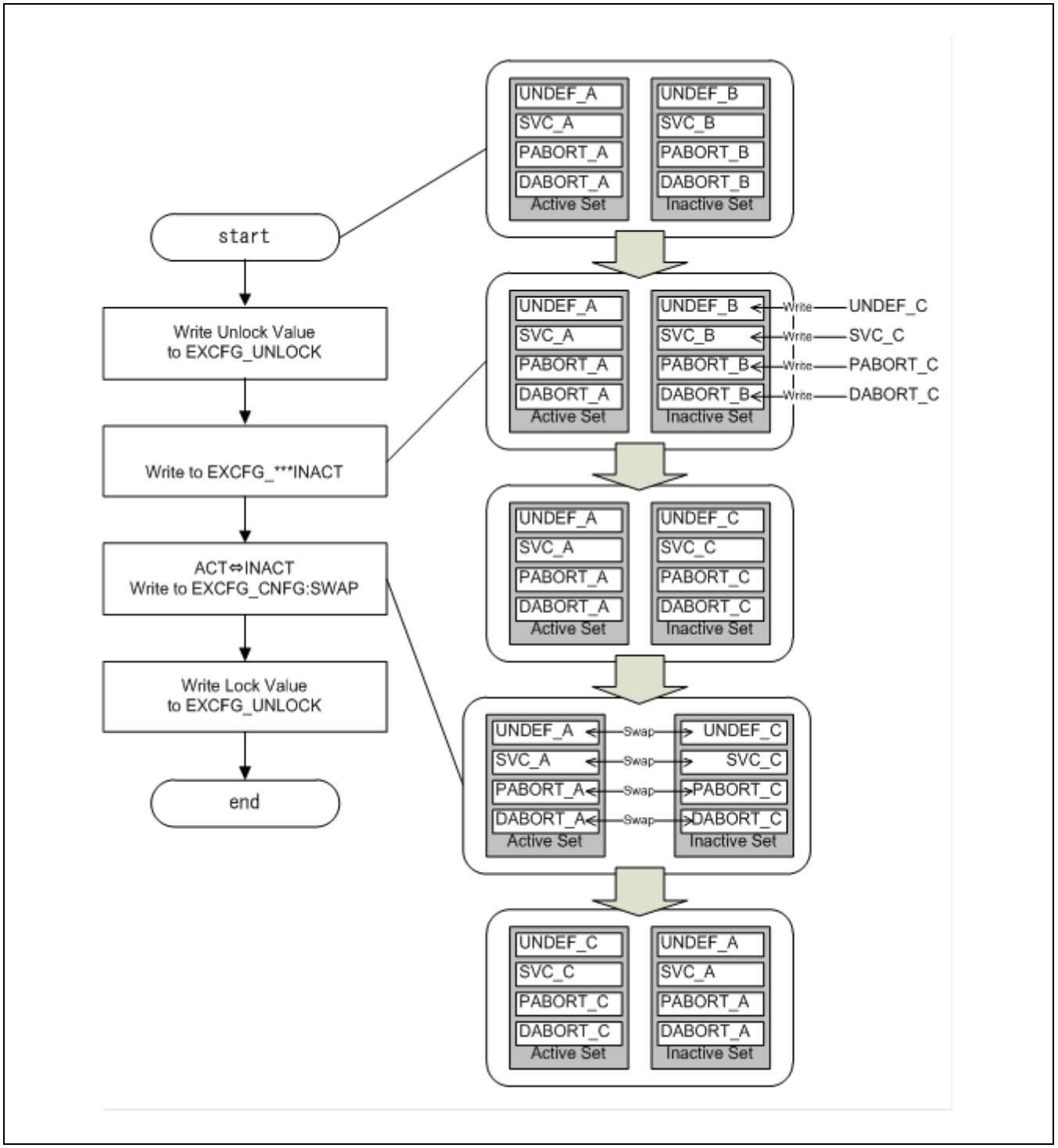
Note:

- *It takes about 30 to 40 clocks of the clock supplied to this block (CLK_HPM) after writing "1" to EXCFG_CNFG:SWAP until the exception handler is enabled with the completion of swap. It is unpredictable which handler will be executed during the swapping period.*

4. Setting Procedure Example

This section explains a setting procedure example of BootROM hardware interface.

Figure 4-1 Setting Flow of BootROM Hardware Interface



5. Registers

This section explains the registers used by the BootROM hardware interface.

The base address of the registers BootROM hardware interface is 0xFFFE_FC00. The base address of ROM is 0xFFFF_0000.

Table 5-1 List of BootROM Hardware Interface Registers

Abbreviated Register Name	Register Name	Reference
EXCFG_UNLOCK	EXCFG lock release register	5.1
EXCFG_CNFG	EXCFG setting register	5.2
EXCFG_UNDEFINACT	EXCFG inactive set - undefined instruction vector register	5.3
EXCFG_SVCINACT	EXCFG inactive set - supervisor call vector register	5.4
EXCFG_PABORTINACT	EXCFG inactive set - prefetch abort vector register	5.5
EXCFG_DABORTINACT	EXCFG inactive set - data abort vector register	5.6
EXCFG_UNDEFACT	EXCFG active set - undefined instruction vector register	5.7
EXCFG_SVCACT	EXCFG active set - supervisor call vector register	5.8
EXCFG_PABORTACT	EXCFG active set - prefetch abort vector register	5.9
EXCFG_DABORTACT	EXCFG active set - data abort vector register	5.10

Table 5-2 Register Memory Layout of BootROM Hardware Interface

Offset	+7	+6	+5	+4	+3	+2	+1	+0
0x00000000 | 0x00000350	Reserved				Reserved			
0x00000358	Reserved				EXCFG_UNLOCK 00000000_00000000_00000000_00000000			
0x00000360	Reserved				EXCFG_CNFG 00000000_00000000_00000000_00000001			
0x00000368 0x00000378	Reserved				Reserved			
0x00000380	EXCFG_UNDEFINACT 11111111_11111111_00000000_00100100				Reserved			
0x00000388	EXCFG_PABORTINACT 11111111_11111111_00000000_00101100				EXCFG_SVCINACT 11111111_11111111_00000000_00101000			
0x00000390	Reserved				EXCFG_DABORTINACT 11111111_11111111_00000000_00110000			
0x00000398	Reserved				Reserved			
0x000003A0 | 0x000003B8	Reserved				Reserved			
0x000003C0	EXCFG_UNDEFACT 11111111_11111111_00000000_00100100				Reserved			
0x000003C8	EXCFG_PABORTACT 11111111_11111111_00000000_00101100				EXCFG_SVCACT 11111111_11111111_00000000_00101000			
0x000003D0	Reserved				EXCFG_DABORTACT 11111111_11111111_00000000_00110000			
0x000003D8	Reserved				Reserved			
0x000003E0 | 0x000003F8	Reserved				Reserved			

* BootROM is mapped to an area starting from 0xFFFF0000.

5.1. EXCFG Lock Release Register (EXCFG_UNLOCK)

This register controls the write lock of the registers of the BootROM hardware interface.

Bit	31	0
Field	UNLOCK	
R/W Attribute	R0,W	
Protection Attribute	WP	
Initial Value	00000000_00000000_00000000_00000000	

[bit31:0] UNLOCK: Lock Release Bits of BootROM Hardware Interface

These bits control the write lock of the setting registers of the BootROM hardware interface. All registers except EXCFG_UNLOCK are subject to sequence protection by this register.

UNLOCK	Description
0xACC5B007	Unlock value (which enables writing)
0xB007ECF6	Lock value (which disables writing)
Other than above	Setting prohibited (bus error returned)

Note:

- *The value of all 32 bits of this register is needed to determine the lock status. Therefore, writing to this register must be made with 32-bit access. 8-bit, 16-bit, or 64-bit access is not allowed.*

5.2. EXCFG Setting Register (EXCFG_CNFG)

This register indicates the lock status of the BootROM hardware interface. This register also swaps the contents of the active and inactive sets of the exception vector registers.

Bit	31	16
Field	Reserved	
R/W Attribute	R0,WX	
Protection Attribute	WPS	
Initial Value	00000000_00000000	

Bit	15	14	13	12	11	10	9	8
Field	Reserved							SWAP
R/W Attribute	R0,WX							R0,W
Protection Attribute	WPS							
Initial Value	0000000							0

Bit	7	6	5	4	3	2	1	0
Field	Reserved							LST
R/W Attribute	R0,WX							R,WX
Protection Attribute	WPS							
Initial Value	0000000							1

[bit31:9] Reserved: Reserved Bits

[bit8] SWAP: Exception Vector Register Swap Bit

Writing "1" to this bit swaps the contents of the active and inactive sets of the exception vector registers.

Bit	Description
0	Invalid (no effect on operation)
1	Swap execution

[bit7:1] Reserved: Reserved Bits

[bit0] LST: BootROM Hardware Interface Lock Status Bit

This bit indicates the lock status of the BootROM hardware interface.

Bit	Description
0	Unlocked state
1	Locked state

Note:

- *It takes about 30 to 40 clocks of the clock supplied to this block (CLK_HPM) after writing "1" to EXCFG_CNFG:SWAP until the exception handler is enabled with the completion of swap. It is unpredictable which handler will be executed during the swapping period.*

5.3. EXCFG Inactive Set - Undefined Instruction Vector Register (EXCFG_UNDEFINACT)

This register is included in the inactive set of the exception vector registers, and stores the starting address of the exception handler for the undefined instruction exception.

Bit	31	0
Field	UNDEFVEC	
R/W Attribute	R,W	
Protection Attribute	WPS	
Initial Value	11111111_11111111_00000000_00100100	

[bit31:0] UNDEFVEC: Undefined Instruction Vector Bits

These bits set the starting address of the exception handler for the undefined instruction exception.

Note:

- *Writing "1" to EXCFG_CNFG:SWAP swaps the value of this register and the value of EXCFG_UNDEFACT.*

5.4. EXCFG Inactive Set - Supervisor Call Vector Register (EXCFG_SVCINACT)

This register is included in the inactive set of the exception vector registers, and stores the starting address of the exception handler for the supervisor call exception.

Bit	31	0
Field	SVCVEC	
R/W Attribute	R,W	
Protection Attribute	WPS	
Initial Value	11111111_11111111_00000000_00101000	

[bit31:0] SVCVEC: Supervisor Call Vector Bits

These bits set the starting address of the exception handler for the supervisor call exception.

Note:

- *Writing "1" to EXCFG_CNFG:SWAP swaps the value of this register and the value of EXCFG_SVCACT.*

5.5. EXCFG Inactive Set - Prefetch Abort Vector Register (EXCFG_PABORTINACT)

This register is included in the inactive set of the exception vector registers, and stores the starting address of the exception handler for the prefetch abort exception.

Bit	31	0
Field	PABORTVEC	
R/W Attribute	R,W	
Protection Attribute	WPS	
Initial Value	11111111_11111111_00000000_00101100	

[bit31:0] PABORTVEC: Prefetch Abort Vector Bits

These bits set the starting address of the exception handler for the prefetch abort exception.

Note:

- *Writing "1" to EXCFG_CNFG:SWAP swaps the value of this register and the value of EXCFG_PABORTACT.*

5.6. EXCFG Inactive Set - Data Abort Vector Register (EXCFG_DABORTINACT)

This register is included in the inactive set of the exception vector registers, and stores the starting address of the exception handler for the data abort exception.

Bit	31	0
Field	DABORTVEC	
R/W Attribute	R,W	
Protection Attribute	WPS	
Initial Value	11111111_11111111_00000000_00110000	

[bit31:0] DABORTVEC: Data Abort Vector Bits

These bits set the starting address of the exception handler for the data abort exception.

Note:

- *Writing "1" to EXCFG_CNFG:SWAP swaps the value of this register and the value of EXCFG_DABORTACT.*

5.7. EXCFG Active Set - Undefined Instruction Vector Register (EXCFG_UNDEFACT)

This register is included in the active set of the exception vector registers, and stores the starting address of the exception handler for the undefined instruction exception.

Bit	31	0
Field	UNDEFVEC	
R/W Attribute	R,W	
Protection Attribute	WPS	
Initial Value	11111111_11111111_00000000_00100100	

[bit31:0] UNDEFVEC: Undefined Instruction Vector Bits

These bits store the starting address of the exception handler for the undefined instruction exception.

Notes:

- *Writing to this register is prohibited. Normal operation is not guaranteed when doing so. To set this register, set the value in the inactive set, and then execute swapping. For details on the setting procedure, see 3 Explanation of Operation and 4 Setting Procedure Example.*
- *Writing "1" to EXCFG_CNFG:SWAP swaps the value of this register and the value of EXCFG_UNDEFINACT.*

5.8. EXCFG Active Set - Supervisor Call Vector Register (EXCFG_SVCACT)

This register is included in the active set of the exception vector registers, and stores the starting address of the exception handler for the supervisor call exception.

Bit	31	0
Field	SVCVEC	
R/W Attribute	R,W	
Protection Attribute	WPS	
Initial Value	11111111_11111111_00000000_00101000	

[bit31:0] SVCVEC: Supervisor Call Vector Bits

These bits store the starting address of the exception handler for the supervisor call exception.

Notes:

- *Writing to this register is prohibited. Normal operation is not guaranteed when doing so. To set this register, set the value in the inactive set, and then execute swapping. For details on the setting procedure, see 3 Explanation of Operation and 4 Setting Procedure Example.*
- *Writing "1" to EXCFG_CNFG:SWAP swaps the value of this register and the value of EXCFG_SVCINACT.*

5.9. EXCFG Active Set - Prefetch Abort Vector Register (EXCFG_PABORTACT)

This register is included in the active set of the exception vector registers, and stores the starting address of the exception handler for the prefetch abort exception.

Bit	31	0
Field	PABORTVEC	
R/W Attribute	R,W	
Protection Attribute	WPS	
Initial Value	11111111_11111111_00000000_00101100	

[bit31:0] PABORTVEC: Prefetch Abort Vector Bits

These bits store the starting address of the exception handler for the prefetch abort exception.

Notes:

- *Writing to this register is prohibited. Normal operation is not guaranteed when doing so. To set this register, set the value in the inactive set, and then execute swapping. For details on the setting procedure, see 3 Explanation of Operation and 4 Setting Procedure Example.*
- *Writing "1" to EXCFG_CNFG:SWAP swaps the value of this register and the value of EXCFG_PABORTINACT.*

5.10. EXCFG Active Set - Data Abort Vector Register (EXCFG_DABORTACT)

This register is included in the active set of the exception vector registers, and stores the starting address of the exception handler for the data abort exception.

Bit	31	0
Field	DABORTVEC	
R/W Attribute	R,W	
Protection Attribute	WPS	
Initial Value	11111111_11111111_00000000_00110000	

[bit31:0] DABORTVEC: Data Abort Vector Bits

These bits store the starting address of the exception handler for the data abort exception.

Notes:

- *Writing to this register is prohibited. Normal operation is not guaranteed when doing so. To set this register, set the value in the inactive set, and then execute swapping. For details on the setting procedure, see 3 Explanation of Operation and 4 Setting Procedure Example.*
- *Writing "1" to EXCFG_CNFG:SWAP swaps the value of this register and the value of EXCFG_DABORTINACT.*

6. Others

This section explains precautions on the use of the BootROM hardware interface.

Register access or ROM access of the BootROM hardware interface returns a bus error in the following cases:

- Sequence protection violation by EXCFG_UNLOCK
- Attempt at writing in EXCFG_UNLOCK a value other than lock release or locking
- Privilege protection violation
- Read or write attempt in which access is made only to a register undefined area
- Write attempt in which access is made only to bits of WX attribute (including register undefined area)
- Attempt at writing in ROM
- Attempt at reading from a read protected area of ROM

CHAPTER 20: BootROM Software Interface

This chapter explains the BootROM Software Interface.

1. **Overview**
2. **BootROM Markers**
3. **BootROM Operation**
4. **Notes**

BOOTROMSW-TXXPT03P01R01L05-E1-XX

1. Overview

The BootROM software is built-in firmware that is executed after reset and before execution of user applications.

The BootROM software chiefly performs mode judgment, security setting, and hardware watchdog timer setting.

It also reads a Secure Boot configuration from the marker and issues a Secure Boot processing request to the SHE.

This section explains the functions of the BootROM software and a configuration based on the marker in the flash memory.

BootROM Functions

The following lists the functions of the BootROM software.

1. Makes a setting for enabling the VIC port and floating-point operations.
2. Judges the operating mode based on the MODE register and MD pin, and branches to a processing handler according to the mode.
3. Makes a setting for security based on the TCFCFG0_SECSTAT.SECEN bit and the Debug Authentication Record (DDR) value.
4. Configures the Secure Boot processing according to the Boot Description Record (BDR) value, and issues a Secure Boot processing request to the SHE.
5. Waits for debugger connection based on the value of BDR_DWEM of the Boot Description Record (BDR).
6. Makes a setting for the hardware watchdog timer according to the Watchdog Description Record (WDR) value.
7. Judges the start address of a user application according to the values of BDR_ABVM and BDR_ABVEM of the Boot Description Record (BDR).
8. The exception vector table of the core is fixed at 0xFFFF0000. Therefore, the BootROM software references the interrupt controller, which is user-configurable, and the BootROM hardware interface to branch to an exception handler.
9. Issues a hard reset if an exception occurs before the user exception handler is set.

2. BootROM Markers

This section explains the BootROM markers.

2.1 Overview of BootROM Markers

2.2 Register Lists

2.3 Security Record (SR)

2.4 Debugger Key Enable Marker (DDR_DSM)

2.5 Debugger Security Key Marker 0 (DDR_DSKM0)

2.6 Debugger Security Key Marker 1 (DDR_DSKM1)

2.7 Debugger Security Key Marker 2 (DDR_DSKM2)

2.8 Debugger Security Key Marker 3 (DDR_DSKM3)

2.9 SHE Secure Boot Mode Marker (BDR_SBMM)

2.10 SHE Secure Boot Size Marker (BDR_SBSM)

2.11 Debugger Connection Wait Enable Marker (BDR_DWEM)

2.12 Alternative Boot Vector Marker (BDR_ABVM)

2.13 Alternative Boot Vector Enable Marker (BDR_ABVEM)

2.14 Hardware Watchdog Interrupt Configuration Marker (WDR_INTM)

2.15 Hardware Watchdog Trigger 0 Configuration Marker (WDR_TRG0CFGM)

2.16 Hardware Watchdog Trigger 1 Configuration Marker (WDR_TRG1CFGM)

2.17 Hardware Watchdog Lower Limit RUN Setting Marker (WDR_RUNLLM)

2.18 Hardware Watchdog Upper Limit RUN Setting Marker (WDR_RUNULM)

2.19 Hardware Watchdog Lower Limit PSS Setting Marker (WDR_PSSLLM)

2.20 Hardware Watchdog Upper Limit PSS Setting Marker (WDR_PSSULM)

2.21 Hardware Watchdog Reset Delay Counter Marker (WDR_RSTDLYM)

2.22 Hardware Watchdog Configuration Marker (WDR_CFGM)

2.23 Hardware Watchdog Configuration Enable Marker (WDR_CEM)

2.1. Overview of BootROM Markers

This section provides an overview of the BootROM markers.

Overview

BootROM markers are registers in the TCFLASH. Unlike ordinary registers, the values of BootROM markers do not directly control hardware functions. The BootROM software reads BootROM markers and controls hardware functions.

Types of BootROM Markers

There are the following 4 types of BootROM markers.

1. Security Record (SR)
   Markers used for configuring security setting
2. Debugger Authentication Record (DDR)
   Markers used for making debugger connection setting
3. Boot Description Record (BDR)
   Markers used for making startup setting
4. Watchdog Description Record (WDR)
   Markers used for making hardware watchdog timer setting

2.2. Register Lists

This section explains the lists of the BootROM marker registers.

The BootROM marker registers comprise the following 4 register groups.

- Security Record (SR) register group
- Debugger Authentication Record (DDR) register group
- Boot Description Record (BDR) register group
- Watchdog Description Record (WDR) register group

List of the BootROM Marker Registers (SR Registers)

Table 2-1 List of the BootROM Marker Registers (SR Registers)

Abbreviated Register Name	Register Name	Reference
MK_SER	Security Enable Marker	2.3
MK_SSR	Security Scope Marker	2.3
MK_CEER	Chip Erase Enable Marker	2.3
MK_SOER	Security Overwrite Enable Marker	2.3
MK_SWPOER	Sector Write Permission Overwrite Enable Marker	2.3
MK_WSWPR	Work Flash Sector Write Permissions Marker	2.3
MK_C0SWPR	Code Flash 0 Sector Write Permissions of the small sectors Marker	2.3
MK_C1SWPR	Code Flash 0 Sector Write Permissions of the large sectors Marker	2.3
MK_C2SWPR	Code Flash 1 Sector Write Permissions of the large sectors Marker	2.3
MK_C3SWPR	Code Flash 2 Sector Write Permissions of the large sectors Marker	2.3

List of the BootROM Marker Registers (DDR Registers)

Table 2-2 List of the BootROM Marker Registers (DDR Registers)

Abbreviated Register Name	Register Name	Reference
DDR_DSM	Debugger Key Enable Marker	2.4
DDR_DSKM0	Debugger Security Key Marker 0	2.5
DDR_DSKM1	Debugger Security Key Marker 1	2.6
DDR_DSKM2	Debugger Security Key Marker 2	2.7
DDR_DSKM3	Debugger Security Key Marker 3	2.8

List of the BootROM Marker Registers (BDR Registers)

Table 2-3 List of the BootROM Marker Registers (BDR Registers)

Abbreviated Register Name	Register Name	Reference
BDR_SBMM	SHE Secure Boot Mode Marker	2.9
BDR_SBSM	SHE Secure Boot Size Marker	2.10
BDR_DWEM	Debugger Connection Wait Enable Marker	2.11
BDR_ABVM	Alternative Boot Vector Marker	2.12
BDR_ABVEM	Alternative Boot Vector Enable Marker	2.13

List of the BootROM Marker Registers (WDR Registers)

Table 2-4 List of the BootROM Marker Registers (WDR Registers)

Abbreviated Register Name	Register Name	Reference
WDR_INTM	Hardware Watchdog Interrupt Configuration Marker	2.14
WDR_TRG0CFGM	Hardware Watchdog Trigger 0 Configuration Marker	2.15
WDR_TRG1CFGM	Hardware Watchdog Trigger 1 Configuration Marker	2.16
WDR_RUNLLM	Hardware Watchdog Lower Limit RUN Setting Marker	2.17
WDR_RUNULM	Hardware Watchdog Upper Limit RUN Setting Marker	2.18
WDR_PSSLLM	Hardware Watchdog Lower Limit PSS Setting Marker	2.19
WDR_PSSULM	Hardware Watchdog Upper Limit PSS Setting Marker	2.20
WDR_RSTDLYM	Hardware Watchdog Reset Delay Counter Marker	2.21
WDR_CFGM	Hardware Watchdog Configuration Marker	2.22
WDR_CEM	Hardware Watchdog Configuration Enable Marker	2.23

Memory Layout of the BootROM Markers

The BootROM markers are located in the beginning area of sector 0 (SA0) of the TCFLASH flash memory.

Table 2-5 BootROM Marker Area

BootROM Markers	TCM Area	AXI Area
SR	+ 0x009F_0000	+ 0x019F_0000
DDR	+ 0x009F_0080	+ 0x019F_0080
BDR	+ 0x009F_00C0	+ 0x019F_00C0
WDR	+ 0x009F_0100	+ 0x019F_0100

Table 2-6 Memory Layout (SR)

Address	Register Name							
	SA0							
	+7	+6	+5	+4	+3	+2	+1	+0
TCM: 0x009F_0000 AXI: 0x019F_0000	Reserved				MK_SER 11111111_11111111_11111111_11111111			
TCM: 0x009F_0008 AXI: 0x019F_0008	Reserved				MK_SSR 11111111_11111111_11111111_11111111			
TCM: 0x009F_0010 AXI: 0x019F_0010	Reserved				MK_CEER 11111111_11111111_11111111_11111111			
TCM: 0x009F_0018 AXI: 0x019F_0018	Reserved				MK_SOER 11111111_11111111_11111111_11111111			
TCM: 0x009F_0020 AXI: 0x019F_0020	Reserved				MK_SWPOER 11111111_11111111_11111111_11111111			
TCM: 0x009F_0028 AXI: 0x019F_0028	Reserved				MK_WSWPR 11111111_11111111_11111111_11111111			
TCM: 0x009F_0030 AXI: 0x019F_0030	Reserved				MK_C0SWPR 11111111_11111111_11111111_11111111			
TCM: 0x009F_0038 AXI: 0x019F_0038	Reserved				MK_C1SWPR 11111111_11111111_11111111_11111111			
TCM: 0x009F_0040 AXI: 0x019F_0040	Reserved				MK_C2SWPR 11111111_11111111_11111111_11111111			
TCM: 0x009F_0048 AXI: 0x019F_0048	Reserved				MK_C3SWPR 11111111_11111111_11111111_11111111			
TCM: 0x009F_0050 AXI: 0x019F_0050	Reserved				Reserved			
TCM: 0x009F_0058 AXI: 0x019F_0058	Reserved				Reserved			
TCM: 0x009F_0060 AXI: 0x019F_0060	Reserved				Reserved			
TCM: 0x009F_0068 AXI: 0x019F_0068	Reserved				Reserved			
TCM: 0x009F_0070 AXI: 0x019F_0070	Reserved				Reserved			
TCM: 0x009F_0078 AXI: 0x019F_0078	Reserved				Reserved			

Table 2-7 Memory Layout (DDR)

Offset	Register Name							
	SA0							
	+7	+6	+5	+4	+3	+2	+1	+0
TCM: 0x009F_0080 AXI: 0x019F_0080	Reserved				Reserved			
TCM: 0x009F_0088 AXI: 0x019F_0088	Reserved				DDR_DSM 11111111_11111111_11111111_11111111			
TCM: 0x009F_0090 AXI: 0x019F_0090	Reserved				DDR_DSKM0 11111111_11111111_11111111_11111111			
TCM: 0x009F_0098 AXI: 0x019F_0098	Reserved				DDR_DSKM1 11111111_11111111_11111111_11111111			
TCM: 0x009F_00A0 AXI: 0x019F_00A0	Reserved				DDR_DSKM2 11111111_11111111_11111111_11111111			
TCM: 0x009F_00A8 AXI: 0x019F_00A8	Reserved				DDR_DSKM3 11111111_11111111_11111111_11111111			
TCM: 0x009F_00B0 AXI: 0x019F_00B0	Reserved				Reserved			
TCM: 0x009F_00B8 AXI: 0x019F_00B8	Reserved				Reserved			

Table 2-8 Memory Layout (BDR)

Offset	Register Name							
	SA0							
	+7	+6	+5	+4	+3	+2	+1	+0
TCM: 0x009F_00C0 AXI: 0x019F_00C0	Reserved				BDR_SBMM 11111111_11111111_11111111_11111111			
TCM: 0x009F_00C8 AXI: 0x019F_00C8	Reserved				BDR_SBSM 11111111_11111111_11111111_11111111			
TCM: 0x009F_00D0 AXI: 0x019F_00D0	Reserved				Reserved			
TCM: 0x009F_00D8 AXI: 0x019F_00D8	Reserved				Reserved			
TCM: 0x009F_00E0 AXI: 0x019F_00E0	Reserved				BDR_DWEM 11111111_11111111_11111111_11111111			
TCM: 0x009F_00E8 AXI: 0x019F_00E8	Reserved				BDR_ABVM 11111111_11111111_11111111_11111111			
TCM: 0x009F_00F0 AXI: 0x019F_00F0	Reserved				BDR_ABVEM 11111111_11111111_11111111_11111111			
TCM: 0x009F_00F8 AXI: 0x019F_00F8	Reserved				Reserved			

Table 2-9 Memory Layout (WDR)

Offset	Register Name							
	SA0							
	+7	+6	+5	+4	+3	+2	+1	+0
TCM: 0x009F_0100 AXI: 0x019F_0100	Reserved				WDR_INTM 11111111_11111111_11111111_11111111			
TCM: 0x009F_0108 AXI: 0x019F_0108	Reserved				WDR_TRG0CFGM 11111111_11111111_11111111_11111111			
TCM: 0x009F_0110 AXI: 0x019F_0110	Reserved				WDR_TRG1CFGM 11111111_11111111_11111111_11111111			
TCM: 0x009F_0118 AXI: 0x019F_0118	Reserved				WDR_RUNLLM 11111111_11111111_11111111_11111111			
TCM: 0x009F_0120 AXI: 0x019F_0120	Reserved				WDR_RUNULM 11111111_11111111_11111111_11111111			
TCM: 0x009F_0128 AXI: 0x019F_0128	Reserved				WDR_PSSLLM 11111111_11111111_11111111_11111111			
TCM: 0x009F_0130 AXI: 0x019F_0130	Reserved				WDR_PSSULM 11111111_11111111_11111111_11111111			
TCM: 0x009F_0138 AXI: 0x019F_0138	Reserved				WDR_RSTDLYM 11111111_11111111_11111111_11111111			
TCM: 0x009F_0140 AXI: 0x019F_0140	Reserved				WDR_CFGM 11111111_11111111_11111111_11111111			
TCM: 0x009F_0148 AXI: 0x019F_0148	Reserved				WDR_CEM 11111111_11111111_11111111_11111111			

2.3. Security Record (SR)

This section explains the Security Record, which is used for configuring the security setting.

Security Record (SR)

For details of the Security Record, see "3.3. Security Marker Definition" in the chapter on Security.

2.4. Debugger Key Enable Marker (DDR_DSM)

This marker is used to enable the debugger connection key.

Bit	31	0
Field	DSEM [31:0]	
R/W Attribute	R/W	
Protection Attribute	-	
Initial Value	11111111_11111111_11111111_11111111	

[bit31:0] DSEM: Debugger Key Enable Marker

These bits are used to enable the debugger security key. The following are the setting values.

bit31:0	Description
0x59F71234	Debugger key enabled
Other than above	Debugger connection disabled (initial value)

If debugger key is enabled with these bits, a 128-bit authentication key (DDR_DSKM0 to 3 registers) can be used to open a debugger connection in case the device is in the security enabled state.

If security is disabled with the MK_SER register, an SR register, a 128-bit authentication key is not required when a debugger is connected.

MK_SER	DDR_DSEM	Description
Security disabled	Debugger key disabled	The debugger can be connected without a key.
Security disabled	Debugger key enabled	The debugger can be connected without a key.
Security enabled	Debugger key disabled	Either an application based authentication scheme has to be used or a debugger cannot be connected.
Security enabled	Debugger key enabled	A 128-bit authentication key can be used to open the debugger connection.

2.5. Debugger Security Key Marker 0 (DDR_DSKM0)

This marker is used to set an authentication key for debugger connection.

Bit	31	0
Field	DSKM [127:96]	
R/W Attribute	R/W	
Protection Attribute	-	
Initial Value	11111111_11111111_11111111_11111111	

[bit31:0] DSKM (bit127:96): Debugger Security Key Marker (bit127:96)
These bits are a 128-bit authentication key for debugger connection when debugger connection is enabled by Debugger Key Enable Marker (DSEM [31:0] bits of the DDR_DSM register). These bits are bit127:96 of the key.

A trivial key DSKM[127:0] equal to all-0 or all-1 cannot be used to authenticate the debugger connection.

2.6. Debugger Security Key Marker 1 (DDR_DSKM1)

This marker is used to set an authentication key for debugger connection.

Bit	31	0
Field	DSKM [95:64]	
R/W Attribute	R/W	
Protection Attribute	-	
Initial Value	11111111_11111111_11111111_11111111	

[bit31:0] DSKM (bit95:64): Debugger Security Key Marker (bit95:64)
These bits are a 128-bit authentication key for debugger connection when debugger connection is enabled by Debugger Key Enable Marker (DSEM [31:0] bits of the DDR_DSM register). These bits are bit95:64 of the key.

A trivial key DSKM[127:0] equal to all-0 or all-1 cannot be used to authenticate the debugger connection.

2.7. Debugger Security Key Marker 2 (DDR_DSKM2)

This marker is used to set an authentication key for debugger connection.

Bit	31	0
Field	DSKM [63:32]	
R/W Attribute	R/W	
Protection Attribute	-	
Initial Value	11111111_11111111_11111111_11111111	

[bit31:0] DSKM (bit63:32): Debugger Security Key Marker (bit63:32)
These bits are a 128-bit authentication key for debugger connection when debugger connection is enabled by Debugger Key Enable Marker (DSEM [31:0] bits of the DDR_DSM register). These bits are bit63:32 of the key.

A trivial key DSKM[127:0] equal to all-0 or all-1 cannot be used to authenticate the debugger connection.

2.8. Debugger Security Key Marker 3 (DDR_DSKM3)

This marker is used to set an authentication key for debugger connection.

Bit	31	0
Field	DSKM [31:0]	
R/W Attribute	R/W	
Protection Attribute	-	
Initial Value	11111111_11111111_11111111_11111111	

[bit31:0] DSKM (bit31:0): Debugger Security Key Marker (bit31:0)
These bits are a 128-bit authentication key for debugger connection when debugger connection is enabled by Debugger Key Enable Marker (DSEM [31:0] bits of the DDR_DSM register). These bits are bit31:0 of the key.

A trivial key DSKM[127:0] equal to all-0 or all-1 cannot be used to authenticate the debugger connection.

2.9. SHE Secure Boot Mode Marker (BDR_SBMM)

This marker is used to set a mode of Secure Boot.

Bit	31	0
Field	SBMM [31:0]	
R/W Attribute	R/W	
Protection Attribute	-	
Initial Value	11111111_11111111_11111111_11111111	

[bit31:0] SBMM: SHE Secure Boot Mode Marker

These bits select a mode of Secure Boot.

bit31:0	Description
0x00000000	"Measuring during application" mode is selected. There is a possibility that the BootROM starts a user application before the SecureBoot processing completes.
Other than above	"Measuring before application" mode is selected. BootROM waits for the SecureBoot processing to complete, and then starts a user application.

2.10. SHE Secure Boot Size Marker (BDR_SBSM)

This marker is used to make the SHE channel master setting.

Bit	31	0
Field	SBSM [31:0]	
R/W Attribute	R/W	
Protection Attribute	-	
Initial Value	11111111_11111111_11111111_11111111	

[bit31:0] SBSM: SHE Secure Boot Size Marker

These bits are used to set the size of the area for the program to be tested with the SHE in bytes.

The value of the size in bytes specified in these bits is used also as data for calculation of the CMAC algorithm.

Specify the size with an alignment of 128 bits.

The start address of the area for the program to be tested with SHE is determined based on the values of the BDR_ABVM register and BDR_ABVEM register.

If the BDR_ABVEM register is set to 0x292D3A7B, the value of the BDR_ABVM register is the start address.

If the BDR_ABVEM register is set to a value other than 0x292D3A7B, 0x00A00000 is the start address.

2.11. Debugger Connection Wait Enable Marker (BDR_DWEM)

This marker is used to set waiting for debugger connection after clearing a hard reset.

Bit	31	0
Field	DWEM [31:0]	
R/W Attribute	R/W	
Protection Attribute	-	
Initial Value	11111111_11111111_11111111_11111111	

[bit31:0] DWEM: Debugger Connection Wait Enable Marker

These bits are used to specify whether to enable wait for debugger connection after clearing a hard reset.

bit31:0	Description
0x292D3A7B	Wait for debugger connection after clearing a hard reset is not enabled. – Debugger connection is not awaited and control is immediately handed over to a user program.
Other than above	Wait for debugger connection after clearing a hard reset is enabled (initial value).

If wait for debugger connection is enabled, control is not handed over from the BootROM software to a user program until the processing of "Debugger connection wait" in "3. BootROM Operation" is complete.

2.12. Alternative Boot Vector Marker (BDR_ABVM)

This marker is used to set the start address of a user program.

Bit	31	0
Field	ABVM [31:0]	
R/W Attribute	R/W	
Protection Attribute	-	
Initial Value	11111111_11111111_11111111_11111111	

[bit31:0] ABVM: Alternative Boot Vector Marker

These bits are used to set the start address of a user program.

If the Alternative Boot Vector Enable Marker (BDR_ABVEM) is used to enable the setting of the start address of a user program, the start address set in these bits is valid. If the Alternative Boot Vector Enable Marker (BDR_ABVEM) is not used and the setting of the start address of a user program is not enabled, the start address of the user program is 0x00A00000 (fixed address).

2.13. Alternative Boot Vector Enable Marker (BDR_ABVEM)

This marker is used to enable the setting of the start address of a user program.

Bit	31	0
Field	ABVEM [31:0]	
R/W Attribute	R/W	
Protection Attribute	-	
Initial Value	11111111_11111111_11111111_11111111	

[bit31:0] ABVEM: Alternative Boot Vector Enable Marker

These bits are used to enable the setting of the start address of a user program.

bit31:0	Description
0x292D3A7B	The setting of the start address of a user program is enabled. − The start address of a user program is the value set in the Alternative Boot Vector Marker (BDR_ABVM).
Other than above	The setting of the start address of a user program is not enabled (initial value). − The start address of a user program is 0x00A00000 (fixed address).

2.14. Hardware Watchdog Interrupt Configuration Marker (WDR_INTM)

This marker is used to set the hardware watchdog interrupt configuration register (HWDG_INT).

Bit	31	24
Field	Reserved	
R/W Attribute	R1/W1	
Protection Attribute	-	
Initial Value	11111111	

Bit	23	22	21	20	19	18	17	16
Field	Reserved						RSTENM	IRQENM
R/W Attribute	R1/W1						R/W	R/W
Protection Attribute	-							
Initial Value	111111						1	1

Bit	15	0
Field	Reserved	
R/W Attribute	R1/W1	
Protection Attribute	-	
Initial Value	11111111_11111111	

[bit31:18] Reserved: Reserved Bits

[bit17] RSTENM: Reset Enable Marker

This bit is a marker that controls output when a watchdog error occurs.

Bit	Description
0	Generate an NMI when a watchdog error occurs.
1	Generate a reset when a watchdog error occurs (initial value).

[bit16] IRQENM: Prior Warning Interrupt Enable Marker

This bit is a marker that enables a prior warning interrupt.

Bit	Description
0	Do not enable a prior warning interrupt.
1	Enable a prior warning interrupt (initial value).

For the prior warning interrupt, see "CHAPTER HARDWARE WATCHDOG TIMER."

[bit15:0] Reserved: Reserved Bits

2.15. Hardware Watchdog Trigger 0 Configuration Marker (WDR_TRG0CFGM)

This marker is used to set the hardware watchdog trigger 0 configuration register (HWDG_TRG0CFG).

Bit	31	8
Field	Reserved	
R/W Attribute	R1/W1	
Protection Attribute	-	
Initial Value	11111111_11111111_11111111	

Bit	7	0
Field	WDGTRG0CFGM [7:0]	
R/W Attribute	R/W	
Protection Attribute	-	
Initial Value	11111111	

[bit31:8] Reserved: Reserved Bits

[bit7:0] WDGTRG0CFGM: Watchdog Trigger 0 Configuration Marker

These bits are a marker that is used to set a value to be written to the hardware watchdog trigger 0 register (HWDG_TRG0) for execution of the watchdog counter clear protection trigger sequence.

For the watchdog counter clear protection trigger sequence, see "CHAPTER HARDWARE WATCHDOG TIMER."

2.16. Hardware Watchdog Trigger 1 Configuration Marker (WDR_TRG1CFGM)

This marker is used to set the hardware watchdog trigger 1 configuration register (HWDG_TRG1CFG).

Bit	31	8
Field	Reserved	
R/W Attribute	R1/W1	
Protection Attribute	-	
Initial Value	11111111_11111111_11111111	

bit	7	0
Field	WDGTRG1CFGM [7:0]	
R/W Attribute	R/W	
Protection Attribute	-	
Initial Value	11111111	

[bit31:8] Reserved: Reserved Bits

[bit7:0] WDGTRG1CFGM: Watchdog Trigger 1 Configuration Marker

These bits are a marker that is used to set a value to be written to the hardware watchdog trigger 1 register (HWDG_TRG1) for execution of the watchdog counter clear protection trigger sequence.

For the watchdog counter clear protection trigger sequence, see "CHAPTER HARDWARE WATCHDOG TIMER."

2.17. Hardware Watchdog Lower Limit RUN Setting Marker (WDR_RUNLLM)

This marker is used to set the hardware watchdog lower limit RUN setting register (HWDG_RUNLLS).

Bit	31	0
Field	WDGRUNLLM [31:0]	
R/W Attribute	R/W	
Protection Attribute	-	
Initial Value	11111111_11111111_11111111_11111111	

[bit31:0] WDGRUNLLM: Window Lower Limit for RUN Setting Marker

These bits are a marker that is used to set the window lower limit value for RUN.

For the window of RUN, see "CHAPTER HARDWARE WATCHDOG TIMER."

2.18. Hardware Watchdog Upper Limit RUN Setting Marker (WDR_RUNULM)

This marker is used to set the hardware watchdog upper limit RUN setting register (HWDG_RUNULS).

Bit	31	0
Field	WDGRUNULM [31:0]	
R/W Attribute	R/W	
Protection Attribute	-	
Initial Value	11111111_11111111_11111111_11111111	

[bit31:0] WDGRUNULM: Window Upper Limit for RUN Setting Marker

These bits are a marker that is used to set the window upper limit value for RUN.

For the window of RUN, see "CHAPTER HARDWARE WATCHDOG TIMER."

2.19. Hardware Watchdog Lower Limit PSS Setting Marker (WDR_PSSLLM)

This marker is used to set the hardware watchdog lower limit PSS setting register (HWDG_PSSLLS).

Bit	31	0
Field	WDGPSSLLM [31:0]	
R/W Attribute	R/W	
Protection Attribute	-	
Initial Value	11111111_11111111_11111111_11111111	

[bit31:0] WDGPSSLLM: Window Lower Limit for PSS Setting Marker

These bits are a marker that is used to set the window lower limit value for PSS.

For the window of PSS, see "CHAPTER HARDWARE WATCHDOG TIMER."

2.20. Hardware Watchdog Upper Limit PSS Setting Marker (WDR_PSSULM)

This marker is used to set the hardware watchdog upper PSS setting register (HWDG_PSSULS).

Bit	31	0
Field	WDGPSSULM [31:0]	
R/W Attribute	R/W	
Protection Attribute	-	
Initial Value	11111111_11111111_11111111_11111111	

[bit31:0] WDGPSSULM: Window Upper Limit for PSS Setting Marker

These bits are a marker that is used to set the window upper limit value for PSS.

For the window of PSS, see "CHAPTER HARDWARE WATCHDOG TIMER."

2.21. Hardware Watchdog Reset Delay Counter Marker (WDR_RSTDLYM)

This marker is used to set the hardware watchdog reset delay counter register (HWDG_RSTDLY).

Bit	31	16
Field	Reserved	
R/W Attribute	R1/W1	
Protection Attribute	-	
Initial Value	11111111_11111111	

Bit	15	0
Field	WDGRSTDLYM [15:0]	
R/W Attribute	R/W	
Protection Attribute	-	
Initial Value	11111111_11111111	

[bit31:16] Reserved: Reserved Bits

[bit15:0] WDGRSTDLYM: Reset/NMI Delay Counter Marker

These bits are used to set the number of cycles for the delay time that is to be inserted before generation of the watchdog reset request or watchdog interrupt request (NMI).

For the number of cycles for the delay time, see "CHAPTER HARDWARE WATCHDOG TIMER."

2.22. Hardware Watchdog Configuration Marker (WDR_CFGM)

This marker is used to set the hardware watchdog configuration register (HWDG_CFG).

Bit	31	24
Field	Reserved	
R/W Attribute	R1/W1	
Protection Attribute	-	
Initial Value	11111111	

Bit	23	22	21	20	19	18	17	16
Field	Reserved			OBSSELM [4:0]				
R/W Attribute	R1/W1			R/W				
Protection Attribute	-							
Initial Value	111			11111				

Bit	15	14	13	12	11	10	9	8
Field	Reserved						CLKSELM	
R/W Attribute	R1/W1						R/W	
Protection Attribute	-							
Initial Value	111111						11	

Bit	7	6	5	4	3	2	1	0
Field	Reserved							
R/W Attribute	R1/W1							
Protection Attribute	-							
Initial Value	11111111							

[bit31:21] Reserved: Reserved Bits

[bit20:16] OBSSELM: Watchdog Counter Monitor Bit Output Selection Marker

These bits are used to select any 1 bit in the watchdog counter (32 bits) as the output of the watchdog counter monitor bit.

bit20:16	Description
00000	Bit 0 is selected as the output of the watchdog counter monitor bit.
00001	Bit 1 is selected as the output of the watchdog counter monitor bit.
00010	Bit 2 is selected as the output of the watchdog counter monitor bit.
...	...
11111	Bit 31 is selected as the output of the watchdog counter monitor bit (initial value).

[bit15:10] Reserved: Reserved Bits

[bit9:8] CLKSELM: Clock Selection Marker

These bits are used to select a source clock of the watchdog counter.

bit9:8	Description
*0	High-speed CR clock selected
*1	Low-speed CR clock selected (initial value)

* means "don't care" (the value is ignored).

[bit7:0] Reserved: Reserved Bits

2.23. Hardware Watchdog Configuration Enable Marker (WDR_CEM)

This marker is used to enable various marker settings of the hardware watchdog (WDR).

Bit	31	0
Field	CEM [31:0]	
R/W Attribute	R/W	
Protection Attribute	-	
Initial Value	11111111_11111111_11111111_11111111	

[bit31:0] CEM: Configuration Enable Marker

These bits are used to enable various markers of the hardware watchdog (WDR).

bit31:0	Description
0x292D3A7B	The hardware watchdog is started based on the settings defined with the WDR.
Other than above	The hardware watchdog is started not based on the settings defined with the WDR (initial value). – The hardware watchdog operates based on the default settings.

3. BootROM Operation

This section explains the operation of the BootROM.

Starting the Device

After any type of reset, the BootROM software is performed first.

If the BootROM software selects user mode with an external pin, a user application is started.

The following shows a process flow of the BootROM software in user mode.

Figure 3-1 User Mode Process Flow (Hard Reset)

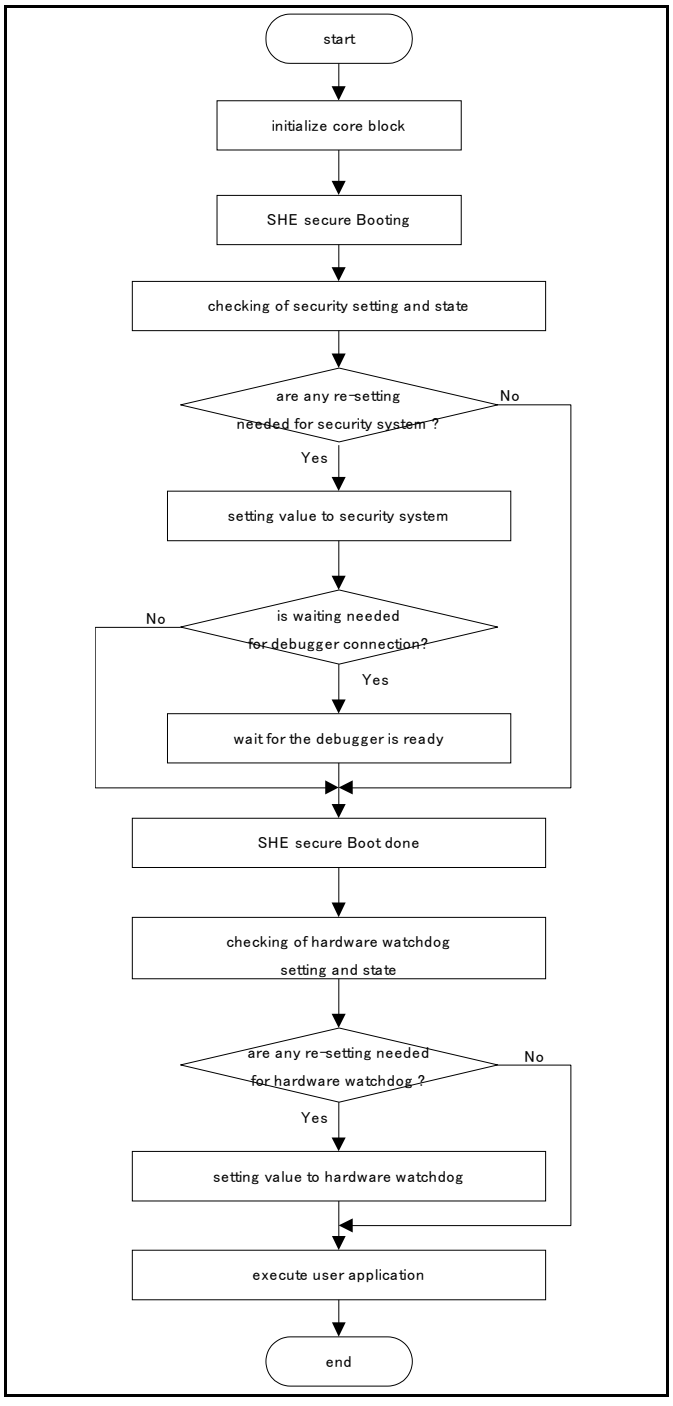
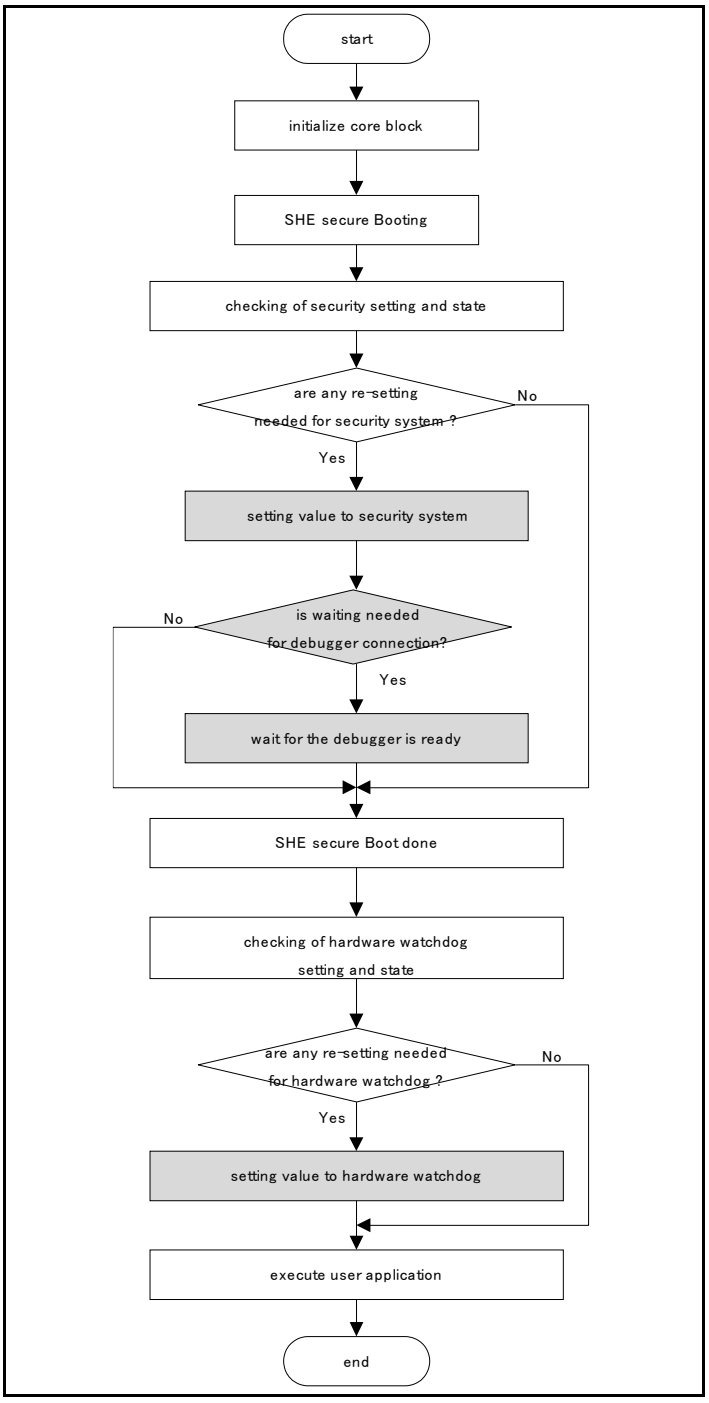
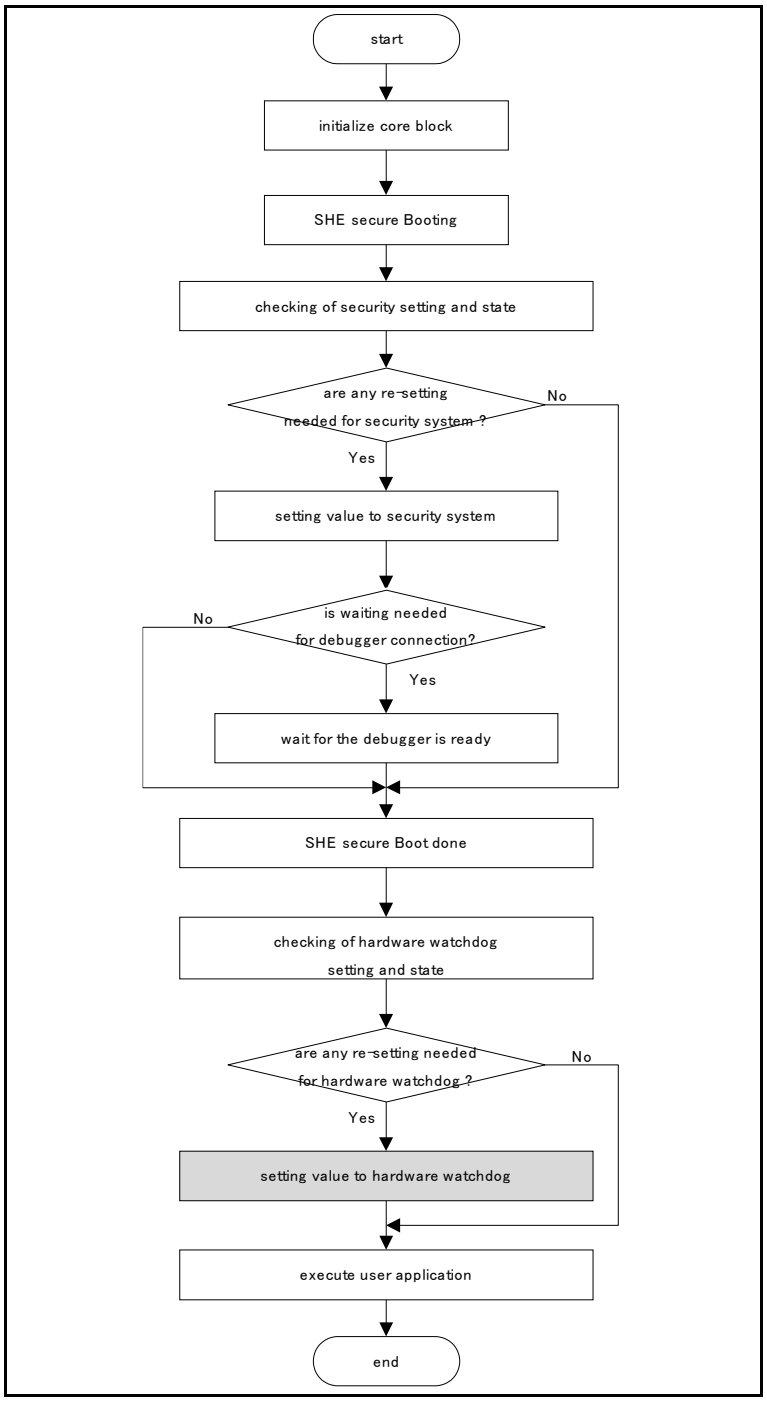


Figure 3-2 User Mode Process Flow (Soft Reset)



For a soft reset, security setting, debugger connection wait, and hardware watchdog setting are not performed.

Figure 3-3 User Mode Process Flow (Standby Reset PD2)



For standby reset PD2, hardware watchdog setting is not performed.

Core Initial Setting

During the core initial setting, the following settings are made.

- Initialization of the general-purpose registers
- VIC port enable setting
- FPU enable setting
- The settings of the stack pointers for the following modes, which are used by the BootROM software
  1. System mode
  2. ABORT
  3. Undefined instruction exception

Table 3-1 BootROM Software CPU Initial Settings

Function	BootROM Software Processing
System control	VIC port enable setting Access right setting of the coprocessor for the FPU
MPU control	Not set
Cache control	Not set
TCM control	Not set
System performance	Cycle Counter is used for debugger connection wait (it is initialized after being used).
FPU control	FPU enable setting
Debugging reset	Not set

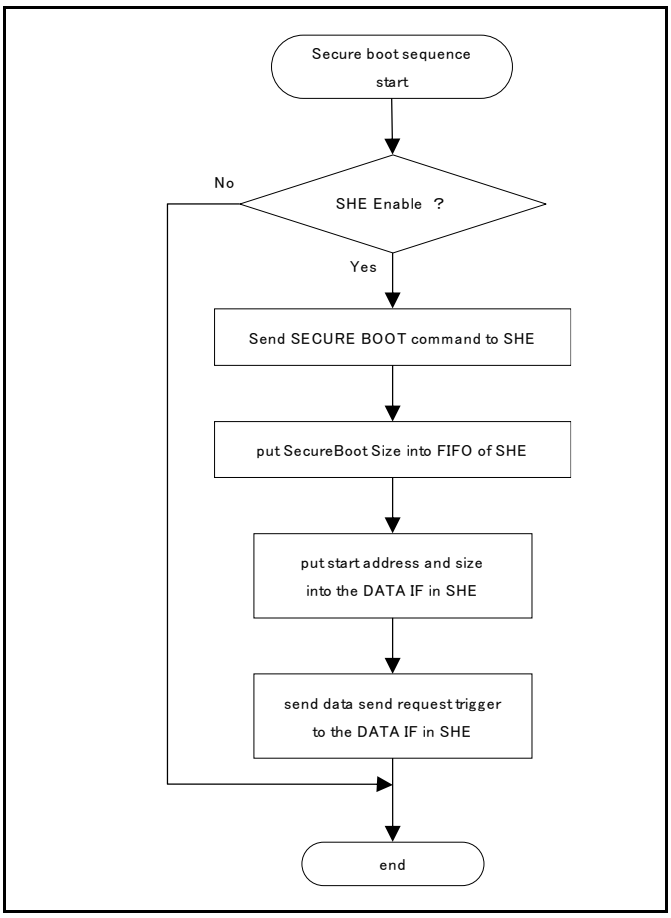
Notes:

- *For Cortex-R5 setting that is not made by the BootROM software, a user application shall make the setting as required.*
- *For Cortex-R5 registers, see the material provided by ARM, "Cortex™-R5 Revision: r1p2 Technical Reference Manual (ARM DDI 0460D)."*

SHE Secure Boot Processing

Processing for SHE Secure Boot configuration is performed according to the values of BDR_SBSM, BDR_ABVM, and BDR_ABVEM of the Boot Description Record (BDR). The BootROM program issues a Secure Boot processing request. If the SHE is disabled, the Secure Boot processing is not performed.

Figure 3-4 SHE Secure Boot Process Flow



The BootROM program issues the Secure Boot command to request the SHE to perform Secure Boot processing. Also, it sets the Secure Boot size value read from BDR_SBSM in the input FIFO of the SHE for CMAC algorithm operation.

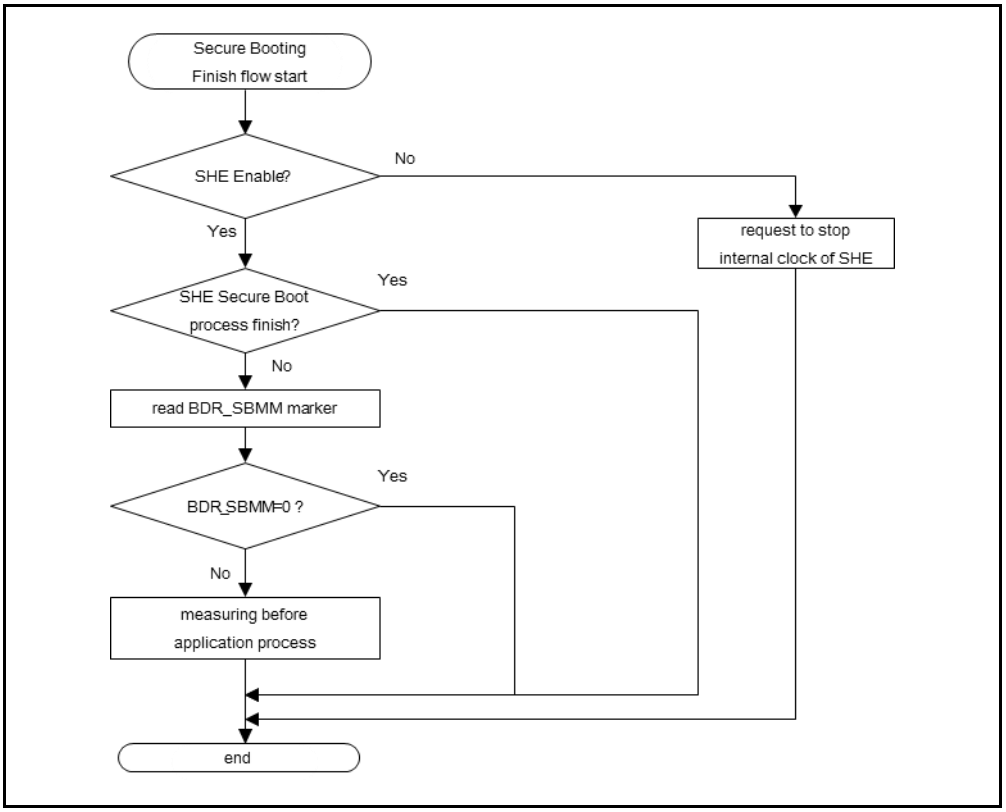
In the DATA IF register of the SHE, the BootROM program sets the Secure Boot Start address determined based on the values of BDR_ABVM and BDR_ABVEM and the Secure Boot size value read from BDR_SBSM. Then, it issues a data transfer request trigger, thus starting the testing of the program area specified by the Secure Boot Start address and Secure Boot size.

When BDR_ABVEM enables the setting of an alternative boot vector, the Secure Boot Start address is the value of BDR_ABVM. Otherwise, the Secure Boot Start address is 0x00A00000.

SHE Secure Boot Finish Processing

Whether the Secure Boot Processing started by the SHE Secure Boot processing has been completed is checked.

Figure 3-5 SHE Secure Boot Finish Process Flow



Read the Status register to check if the processing has been completed. If it is complete, then the SHE Secure Boot Finish processing is completed. If it has not been completed, the BootROM program performs the following processing according to the value of BDR_SBMM.

- When BDR_SBMM is 0x00000000 ("Measuring during application" mode is selected)
  The BootROM program completes the SHE Secure Boot Finish processing without waiting for the completion of the SHE Secure Boot processing.
- BDR_SBMM is not 0x00000000 ("Measuring before application" is selected.)
  The BootROM program completes the SHE Secure Boot Finish processing after waiting until the SHE Secure Boot processing is completed.

The BootROM program reads the Status register of the SHE. If the read value of the Status register indicates that the SHE is disabled, the BootROM program sets a request for stopping the SHE internal CLK, and then completes the SHE Secure Boot Finish processing.

Security Setting

As to the security setting, access limit setting of the flash memory, the enabling of the use of the debugger, and the security key setting are performed.

The security setting is made according to the value of the TCFCFG0_SECSTAT.SECEN bit.

The following is the condition for enabling the use of the debugger.

The value of the Debugger Key Enable Marker (DDR_DSM) is 0x59F71234.

When the reset factor is a soft reset, since the security setting is not initialized, the security setting is not made.

For access restriction by security setting, see "CHAPTER SECURITY."

Debugger Connection Wait

When debugger connection wait is enabled by the Debugger Connection Wait Enable Marker (BDR_DWEM) setting, the debugger connection wait processing is performed.

This processing enables debugging from the first instruction of an application.

The following shows details of the processing.

1. *Debugger Connection Wait Enable Marker (BDR_DWEM) is checked.*
2. *If the debugger connection wait is not enabled, the next BootROM software processing is performed without debugger connection wait being performed.*
3. *Whether the debugger is connected is checked.*
4. *If the debugger is not connected for a period of 19,200 high-speed CR clock cycles, the connection wait is canceled and the next BootROM software processing is performed.*
5. *If the debugger is connected, whether the debugger setting is completed is checked.*
6. *If there is no notification of setting completion from the debugger, the wait state is canceled after 8,388,608 (2 to 23rd power) high-speed CR clock cycles.*

Hardware Watchdog Setting

The setting of the hardware watchdog is made according to the value of the Watchdog Description Record (WDR).

If the reset factor is a soft reset or standby reset (PD2), the setting of the hardware watchdog is not made.

The lock bit of the hardware watchdog is set with the initial values being maintained and the hardware watchdog is set in the operating state if the following conditions are met: The reset factor is neither soft reset nor standby reset (PD2) and the value of the Hardware Watchdog Configuration Enable Marker (WDR_CEM) is not 0x292D3A7B.

The value of Watchdog Description Record (WDR) and the lock bit are set in the hardware watchdog and the hardware watchdog is set in the operating state if the following conditions are met: The reset factor is neither soft reset nor standby reset (PD2) and the value of the Hardware Watchdog Configuration Enable Marker (WDR_CEM) is 0x292D3A7B.

Executing a User Application

The BootROM software jumps to the user application in its last processing.

If the Alternative Boot Vector Enable Marker (BDR_ABVEM) is enabled, it jumps to the value of the Alternative Boot Vector Marker (BDR_ABVM).

If the Alternative Boot Vector Enable Marker is not enabled, the software jumps to the fixed address of 0x00A00000.

Before jumping to the user application, the BootROM software clears the stack area and general-purpose registers R0 to R12 it has used.

Exception Vector Table

In the BootROM, there is a core exception vector table.

In the exception vector table, processing for branching to exception handlers is stored.

An exception handler address is specified by a register of the interrupt controller or BootROM hardware interface, and the BootROM software jumps to the value of the register.

The following table includes the registers that are referenced by the BootROM software when exceptions occur.

Table 3-2 Reference Registers When Exceptions Occur

Exception	Reference Register
Undefined instruction Exception	EXCFG_UNDEFACT
SVC	EXCFG_SVCACT
Instruction abort	EXCFG_PABORTACT
Data abort	EXCFG_DABORTACT
NMI(FIQ)	IRC_NMIVASBR

4. Notes

This section provides notes regarding the BootROM.

RAM Area Used by the BootROM Software

For the execution of the BootROM software, the TCRAM area (address range of 0x00000000 to 0x000000CF) is used.

Since the area used by the BootROM software is initialized to 0 at the end of the BootROM software processing, its contents are changed from those before the reset.

Interruption Setting

The BootROM software does not set the I bit and F bit of the core internal register CPSR.

(The default value is 1, which means "Disable.")

Before using IRQ or FIQ, a user application shall set the I and F bits to 0.

ARM Core Setting

1. R0-R12 are cleared to 0.
2. Bit24(VE) of System Control Register (SCTLR) is set to 1.
3. Bit23:22(CP11) and bit21:20(CP10) of Coprocessor Access Control Register (CPACR) are set to 0b11 (Privileged and User mode access).
4. Bit30(EN) of Floating-Point Exception Register (FPEXC) is set to 1.

CHAPTER 21: Exclusive Access Memory (EAM)



This chapter provides an overview of the exclusive access memory (EAM) and describes its configuration, operation, and memory area.

1. **Overview**
2. **Configuration**
3. **Explanation of Operation**
4. **Operation Examples**
5. **Memory Area**

EAM-TXXPT03P01R01L05-E1-XX

1. Overview

This section describes the features of the exclusive access memory (EAM) block.

If a conflicting access occurs, Cortex™-R5F can exclusively access the memory area of this block by using the AXI exclusive command for the memory area.

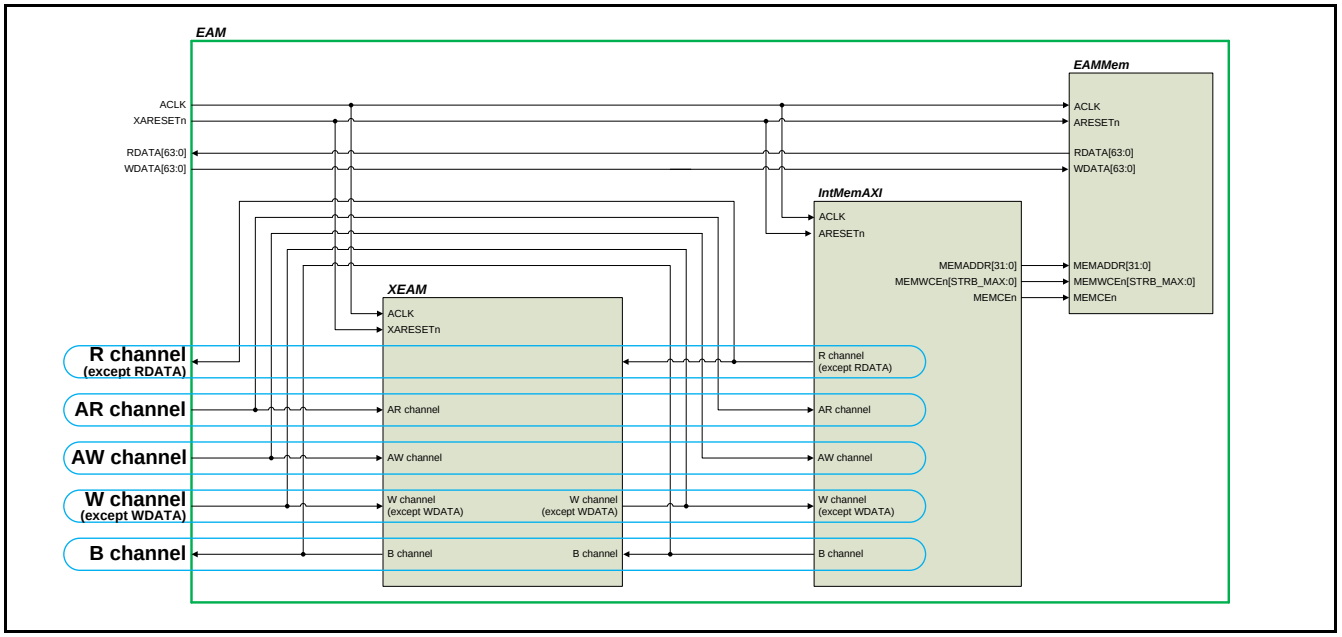
Features

- This block is located as a slave on the HPM, and this memory area can be accessed exclusively from Cortex-R5F.
- This block has 48 bytes of memory area that can be accessed exclusively.
- Exclusion processing by this block is positioned as the global monitor. (The local monitor is located on Cortex-R5F.) For details on the local monitor and the global monitor, see the ARM® Architecture Reference Manual ARM®v7-A and ARM®v7-R edition(ARM DDI 0406B).
- Up to 4 ID and address range sets can be monitored for exclusive access.
- The write data width is 64 bits.
- The read data width is 64 bits.
- The ID width is 8 bits.

2. Configuration

This section shows the configuration of the exclusive access memory (EAM).

Figure 2-1 Block Diagram of Exclusive Access Memory (EAM)



- XEAM

  Exclusive access processing is performed.

  Up to 4 ID and address range sets can be monitored simultaneously.

- IntMemAXI

  Conversion from the AXI interface to the memory interface is performed.

- EAMMem

  This block has a capacity of 48 bytes as a memory area, and reads and writes data using the memory interface.

3. Explanation of Operation

This section describes the operation for each instruction.

The letter "x" in LDRx, STRx, LDREXx, STREXx, and so on in the commands described below represents the size of the command: "D" = Doubleword, "" = Word, "H" = Halfword, and "B" = Byte. (There is not suffix of "x" in the Word case.)

3.1. General Aspects of Transfer

This section outlines the operation for normal instructions (such as STRx and LDRx) and exclusive instructions (such as STREXx and LDREXx).

- This block is located as a slave on the HPM, and receives and processes an exclusive access (in addition to normal access) from a master.
- The memory area in this block has a capacity of 48 bytes. This memory area supports exclusive access. Normal read (LDRx), normal write (STRx), exclusive load (LDREXx), and exclusive store (STREXx) can be performed.
- IntMemAXI in this block has no address buffer mounted. Therefore, it has no WID and processes both normal write (STRx) and exclusive store instruction (STREXx) without interleaving. It processes address and data pairs only one by one. So, consider a transfer method for a master.
- This block does not support AxCACHE and AxPROT.

3.2. Normal Access

This section describes the operation for normal instructions (such as STRx and LDRx).

Read Instruction (LDRx)

- This instruction reads the relevant address in EAMMem according to a read instruction (LDRx) from a master.

Write Instruction (STRx)

- This instruction writes to the relevant address in EAMMem according to a write instruction (STRx) from a master.

3.3. Exclusive Access

This section outlines the operation for exclusive instructions (such as STREXx and LDREXx).

- Up to 4 ID (except for WID) and address sets are monitored simultaneously for the exclusive load instruction (LDREXx) and exclusive store instruction (STREXx) from a master.

- Whether access is exclusive or normal is determined by ARLOCK[1:0] (Read) or AWLOCK[1:0] (Write). The access is handled as exclusive load when ARLOCK[1:0] is "0b01" and as exclusive store when AWLOCK[1:0] is "0b01". In cases other than these, it is handled as normal read (LDRx) or write (STRx).

3.3.1. Exclusive Load Instruction (LDREXx)

This section describes the operation for the exclusive load instruction (LDREXx).

This section uses the following rules in transfer example tables:

The transfer example tables are described in ascending order of time.

However, transfers that have "+" in the Simultaneous Reception column are received simultaneously.

ADD (Address) A, B, C, and D are addresses whose address ranges do not overlap in any part. ADD (Address) A and ADD (Address) @A and the like are addresses whose address ranges overlap in some part.

- When an exclusive load instruction (LDREXx) is received, 1 address is monitored for 1 ID.
- Exclusive load instruction (LDREXx) monitoring is performed on 4 sets of different IDs and addresses. (A register that holds an ID and address as a set is hereafter referred to as a monitoring register.)
- If an exclusive load instruction (LDREXx) is determined as exclusive access NG with the following conditions, the response (RRESP[1:0]) to the master is set to OKAY ("0b00"). Otherwise, it is determined as exclusive access OK and EXOKAY ("0b01") is sent.

1. If normal write (STRx) and an exclusive load instruction (LDREXx) are received simultaneously and they have the same address or their address ranges overlap in some part, the exclusive load instruction (LDREXx) is determined as exclusive access NG. (At this time, normal write (STRx) is written normally and the information of the exclusive load instruction (LDREXx) that is determined as exclusive access NG is not held by a monitoring register.)
2. If an exclusive store instruction (STREXx) and an exclusive load instruction (LDREXx) are received simultaneously and they have the same address or their address ranges overlap in some part, the exclusive load instruction (LDREXx) is determined as exclusive access NG. (At this time, the exclusive store instruction (STREXx) is processed normally and the information of the exclusive load instruction (LDREXx) that is determined as exclusive access NG is not held by a monitoring register.)
3. If an exclusive load instruction (LDREXx) is sent from a master by a method other than single burst (ARLENS[3:0]="0b0000" and ARBURSTS[1:0]="0b01"), it is determined as exclusive access NG.
4. If monitoring registers already hold 4 exclusive load instructions (LDREXx) when an exclusive load instruction (LDREXx) is received, the ID of the received exclusive load instruction (LDREXx) does not match the values in the monitoring registers, and address ranges do not overlap in any part, the instruction is determined as exclusive access NG. (At this time, the information of the exclusive load instruction (LDREXx) that is determined as exclusive access NG is not held by a monitoring register.)

The following table provides an example of operation when normal write (STRx) and an exclusive load instruction (LDREXx) are received simultaneously and they have the same address, as described in 1. above.

Table 3-1 When Normal Write (STRx) and Exclusive Load Instruction (LDREXx) Are Received Simultaneously and They Have the Same Address

Simultaneous Reception	Transfer Type	Ax ID	ADD	Response	Monitoring Register 1		Monitoring Register 2		Monitoring Register 3		Monitoring Register 4	
					ID	ADD	ID	ADD	ID	ADD	ID	ADD
	LDREXx	0	A	EXOKAY	0	A						
+	STRx	1	A	OKAY								
+	LDREXx	0	A	OKAY								
	LDREXx	3	B	EXOKAY	3	B						
+	STRx	0	C	OKAY	3	B						
+	LDREXx	2	C	OKAY	3	B						

The following tables provide examples of operation when an exclusive store instruction (STREXx) and an exclusive load instruction (LDREXx) are received simultaneously, as described in 2. above.

Table 3-2 When Exclusive Store Instruction (STREXx) and Exclusive Load Instruction (LDREXx) Are Received Simultaneously and They Have the Same Address

Simultaneous Reception	Transfer Type	Ax ID	ADD	Response	Monitoring Register 1		Monitoring Register 2		Monitoring Register 3		Monitoring Register 4	
					ID	ADD	ID	ADD	ID	ADD	ID	ADD
	LDREXx	0	A	EXOKAY	0	A						
+	STREXx	0	B	OKAY	0	A						
+	LDREXx	1	B	OKAY	0	A						
	STREXx	0	A	EXOKAY								

Table 3-3 When Exclusive Store Instruction (STREXx) and Exclusive Load Instruction (LDREXx) Are Received Simultaneously and They Have Different Addresses

Simultaneous Reception	Transfer Type	Ax ID	ADD	Response	Monitoring Register 1		Monitoring Register 2		Monitoring Register 3		Monitoring Register 4	
					ID	ADD	ID	ADD	ID	ADD	ID	ADD
	LDREXx	0	A	EXOKAY	0	A						
+	STREXx	0	B	OKAY	0	A						
+	LDREXx	1	A	EXOKAY	1	A						
	STREXx	1	A	EXOKAY								

Table 3-4 When Exclusive Store Instruction (STREXx) and Exclusive Load Instruction (LDREXx) Are Received Simultaneously and They Have Different Addresses

Simultaneous Reception	Transfer Type	Ax ID	ADD	Response	Monitoring Register 1		Monitoring Register 2		Monitoring Register 3		Monitoring Register 4	
					ID	ADD	ID	ADD	ID	ADD	ID	ADD
	LDREXx	0	A	EXOKAY	0	A						
+	STREXx	0	A	EXOKAY								
+	LDREXx	1	B	EXOKAY	1	B						
	STREXx	1	B	EXOKAY								

The following tables provide examples of operations when address ranges do not overlap in any part and when address ranges overlap in some part, as described in 4. above.

Table 3-5 When Exclusive Load Instruction (LDREXx) Whose Address Range (of ID Different from IDs Held by Monitoring Registers) Does Not Overlap in Any Part Is Received

Simultaneous Reception	Transfer Type	Ax ID	ADD	Response	Monitoring Register 1		Monitoring Register 2		Monitoring Register 3		Monitoring Register 4	
					ID	ADD	ID	ADD	ID	ADD	ID	ADD
	LDREXx	0	A	EXOKAY	0	A						
	LDREXx	1	B	EXOKAY	0	A	1	B				
	LDREXx	2	C	EXOKAY	0	A	1	B	2	C		
	LDREXx	3	D	EXOKAY	0	A	1	B	2	C	3	D
	LDREXx	4	E	OKAY	0	A	1	B	2	C	3	D

Table 3-6 When Exclusive Load Instruction (LDREXx) Whose Address Range (of the Same ID as ID Held by Monitoring Register) Overlaps in Some Part Is Received

Simultaneous Reception	Transfer Type	Ax ID	ADD	Response	Monitoring Register 1		Monitoring Register 2		Monitoring Register 3		Monitoring Register 4	
					ID	ADD	ID	ADD	ID	ADD	ID	ADD
	LDREXx	0	A	EXOKAY	0	A						
	LDREXx	1	B	EXOKAY	0	A	1	B				
	LDREXx	2	C	EXOKAY	0	A	1	B	2	C		
	LDREXx	3	D	EXOKAY	0	A	1	B	2	C	3	D
	LDREXx	3	@A	EXOKAY			1	B	2	C	3	@A

3.3.2. Exclusive Store Instruction (STREXx)

This section describes the operation for the exclusive store instruction (STREXx).

This section uses the following rules in transfer example tables:

The transfer example tables are described in ascending order of time.

However, transfers that have "+" in the Simultaneous Reception column are received simultaneously.

ADD (Address) A, B, C, and D are addresses whose address ranges do not overlap in any part. ADD (Address) A and ADD (Address) @A and the like are addresses whose address ranges overlap in some part.

- If an exclusive store instruction (STREXx) is determined as exclusive access OK, a write command is sent to IntMemAXI without particularly processing the write command.

- If an exclusive store instruction (STREXx) is under the following conditions, it is determined as exclusive access NG, write strobe is dropped to IntMemAXI (0x00), and a write command that does not update data is sent. The response (BRESP[1:0]) to the master is set to OKAY ("0b00"). If the instruction is not under the following conditions, it is determined as exclusive access OK and EXOKAY ("0b01") is sent.

1. If an exclusive store instruction (STREXx) that is different from exclusive load instructions (LDREXx) held by monitoring registers is received, it is determined as exclusive access NG. (This includes a case where normal write (STRx) cleared an internally held exclusive load instruction (LDREXx).)

2. If an exclusive store instruction (STREXx) is sent from a master by means other than single burst (AWLENS[3:0]="0b0000" and AWBURSTS[1:0]="0b01"), it is determined as exclusive access NG.

The following tables provide examples of operation that is described in 1. above.

Table 3-7 When Exclusive Store Instruction (STREXx) Different from Contents of Monitoring Registers Is Received

Simultaneous Reception	Transfer Type	Ax ID	ADD	Response	Monitoring Register 1		Monitoring Register 2		Monitoring Register 3		Monitoring Register 4	
					ID	ADD	ID	ADD	ID	ADD	ID	ADD
	LDREXx	0	A	EXOKAY	0	A						
	LDREXx	1	B	EXOKAY	0	A	1	B				
	LDREXx	2	C	EXOKAY	0	A	1	B	2	C		
	LDREXx	3	D	EXOKAY	0	A	1	B	2	C	3	D
	STREXx	0	E	OKAY	0	A	1	B	2	C	3	D

Table 3-8 When Normal Write (STRx) Cleared Monitoring Register

Simultaneous Reception	Transfer Type	Ax ID	ADD	Response	Monitoring Register 1		Monitoring Register 2		Monitoring Register 3		Monitoring Register 4	
					ID	ADD	ID	ADD	ID	ADD	ID	ADD
	LDREXx	0	A	EXOKAY	0	A						
	LDREXx	1	B	EXOKAY	0	A	1	B				
	STRx	0	A	OKAY			1	B				
	STREXx	0	A	OKAY			1	B				

4. Operation Examples

This section describes the operation of this block for exclusive access from a master.

This section uses the following rules in transfer example tables:

The transfer example tables are described in ascending order of time.

However, transfers that have "+" in the Simultaneous Reception column are received simultaneously.

ADD (Address) A, B, C, and D are addresses whose address ranges do not overlap in any part. ADD (Address) A and ADD (Address) @A and the like are addresses whose address ranges overlap in some part.

The following tables provide examples of the operation of this block when exclusive access is received from a master.

Table 4-1 When Exclusive Load Instructions (LDREXx) Have Different IDs and Address Ranges

Simultaneous Reception	Transfer Type	Ax ID	ADD	Response	Monitoring Register 1		Monitoring Register 2		Monitoring Register 3		Monitoring Register 4	
					ID	ADD	ID	ADD	ID	ADD	ID	ADD
	LDREXx	0	A	EXOKAY	0	A						
	LDREXx	1	B	EXOKAY	0	A	1	B				
	LDREXx	2	C	EXOKAY	0	A	1	B	2	C		
	LDREXx	3	D	EXOKAY	0	A	1	B	2	C	3	D
	STREXx	0	A	EXOKAY			1	B	2	C	3	D
	STREXx	1	B	EXOKAY					2	C	3	D
	STREXx	2	C	EXOKAY							3	D
	STREXx	3	D	EXOKAY								

Table 4-2 When Exclusive Load Instructions (LDREXx) Have Different IDs and the Same Address Range

Simultaneous Reception	Transfer Type	Ax ID	ADD	Response	Monitoring Register 1		Monitoring Register 2		Monitoring Register 3		Monitoring Register 4	
					ID	ADD	ID	ADD	ID	ADD	ID	ADD
	LDREXx	0	A	EXOKAY	0	A						
	LDREXx	1	A	EXOKAY	1	A						
	LDREXx	2	A	EXOKAY	2	A						
	LDREXx	3	A	EXOKAY	3	A						
	STREXx	0	A	OKAY	3	A						
	STREXx	1	A	OKAY	3	A						
	STREXx	2	A	OKAY	3	A						
	STREXx	3	A	EXOKAY								

Table 4-3 When Exclusive Store Instructions (STREXx) That Have Different IDs for the Same Address Range Arrive

Simultaneous Reception	Transfer Type	Ax ID	ADD	Response	Monitoring Register 1		Monitoring Register 2		Monitoring Register 3		Monitoring Register 4	
					ID	ADD	ID	ADD	ID	ADD	ID	ADD
	LDREXx	0	A	EXOKAY	0	A						
	STREXx	1	A	OKAY	0	A						
	LDREXx	1	A	EXOKAY	1	A						
	STREXx	2	A	OKAY	1	A						
	LDREXx	2	A	EXOKAY	2	A						
	STREXx	3	A	OKAY	2	A						
	LDREXx	3	A	EXOKAY	3	A						
	STREXx	0	A	OKAY	3	A						

Row 2: If STREXx is determined as exclusive access NG, the monitoring register holds values as they are.

Row 3: If LDREXx has the same address range, the current ID is overwritten with the ID of LDREXx.

Table 4-4 When ID and Address Range of Exclusive Load Instruction (LDREXx) Do Not Match Those of Exclusive Store Instruction (STREXx)

Simultaneous Reception	Transfer Type	Ax ID	ADD	Response	Monitoring Register 1		Monitoring Register 2		Monitoring Register 3		Monitoring Register 4	
					ID	ADD	ID	ADD	ID	ADD	ID	ADD
	LDREXx	0	A	EXOKAY	0	A						
	LDREXx	1	B	EXOKAY	0	A	1	B				
	LDREXx	2	C	EXOKAY	0	A	1	B	2	C		
	LDREXx	3	D	EXOKAY	0	A	1	B	2	C	3	D
	STREXx	0	D	OKAY			1	B	2	C	3	D
	STREXx	1	C	OKAY					2	C	3	D
	STREXx	2	B	OKAY							3	D
	STREXx	3	A	OKAY								

Table 4-5 When Order of Exclusive Load Instructions (LDREXx) Is Different from That of Exclusive Store Instructions (STREXx)

Simultaneous Reception	Transfer Type	Ax ID	ADD	Response	Monitoring Register 1		Monitoring Register 2		Monitoring Register 3		Monitoring Register 4	
					ID	ADD	ID	ADD	ID	ADD	ID	ADD
	LDREXx	0	A	EXOKAY	0	A						
	LDREXx	1	B	EXOKAY	0	A	1	B				
	LDREXx	2	C	EXOKAY	0	A	1	B	2	C		
	LDREXx	3	D	EXOKAY	0	A	1	B	2	C	3	D
	STREXx	3	D	EXOKAY	0	A	1	B	2	C		
	STREXx	2	C	EXOKAY	0	A	1	B				
	STREXx	1	B	EXOKAY	0	A						
	STREXx	0	A	EXOKAY								

Table 4-6 When 5 Exclusive Load Instructions (LDREXx) Are Received

Simultaneous Reception	Transfer Type	Ax ID	ADD	Response	Monitoring Register 1		Monitoring Register 2		Monitoring Register 3		Monitoring Register 4	
					ID	ADD	ID	ADD	ID	ADD	ID	ADD
	LDREXx	0	A	EXOKAY	0	A						
	LDREXx	1	B	EXOKAY	0	A	1	B				
	LDREXx	2	C	EXOKAY	0	A	1	B	2	C		
	LDREXx	3	D	EXOKAY	0	A	1	B	2	C	3	D
	LDREXx	0	E	EXOKAY	0	E	1	B	2	C	3	D
	STREXx	0	A	OKAY			1	B	2	C	3	D
	STREXx	1	B	EXOKAY					2	C	3	D
	STREXx	2	C	EXOKAY							3	D
	STREXx	3	D	EXOKAY								
	STREXx	0	E	OKAY								

Row 5: The values are overwritten by LDREXx that has the same ID and a different address range.
Row 6: The response is OKAY because a monitoring register is overwritten (in row 5).

Table 4-7 When 5 Exclusive Load Instructions (LDREXx) Are Received

Simultaneous Reception	Transfer Type	Ax ID	ADD	Response	Monitoring Register 1		Monitoring Register 2		Monitoring Register 3		Monitoring Register 4	
					ID	ADD	ID	ADD	ID	ADD	ID	ADD
	LDREXx	0	A	EXOKAY	0	A						
	LDREXx	1	B	EXOKAY	0	A	1	B				
	LDREXx	2	C	EXOKAY	0	A	1	B	2	C		
	LDREXx	3	D	EXOKAY	0	A	1	B	2	C	3	D
	LDREXx	4	E	OKAY	0	A	1	B	2	C	3	D
	STREXx	0	A	EXOKAY			1	B	2	C	3	D
	STREXx	1	B	EXOKAY					2	C	3	D
	STREXx	2	C	EXOKAY							3	D
	STREXx	3	D	EXOKAY								
	STREXx	4	E	OKAY								

Row 5: The monitoring registers are not overwritten because the ID is different from the 4 IDs that are held. LDREXx=OKAY

Table 4-8 When Exclusive Store Instruction (STREXx) Arrives before Exclusive Load Instruction (LDREXx)

Simultaneous Reception	Transfer Type	Ax ID	ADD	Response	Monitoring Register 1		Monitoring Register 2		Monitoring Register 3		Monitoring Register 4	
					ID	ADD	ID	ADD	ID	ADD	ID	ADD
	STREXx	0	A	OKAY								
	LDREXx	0	A	EXOKAY	0	A						

Row 1: The response is OKAY because STREXx is received when there are no monitoring targets in the monitoring registers.

Table 4-9 When There Is Normal Write (STRx) That Has the Same Address between Exclusive Load Instruction (LDREXx) and Exclusive Store Instruction (STREXx) (When Normal Write Is Received, Monitoring Registers Are Cleared)

Simultaneous Reception	Transfer Type	Ax ID	ADD	Response	Monitoring Register 1		Monitoring Register 2		Monitoring Register 3		Monitoring Register 4	
					ID	ADD	ID	ADD	ID	ADD	ID	ADD
	LDREXx	0	A	EXOKAY	0	A						
	STRx	0	A	OKAY								
	STREXx	0	A	OKAY								
	LDREXx	0	B	EXOKAY	0	B						
	STRx	1	B	OKAY								
	STREXx	0	B	OKAY								

Row 2: Normal write (STRx) to the same ID and address as those of the exclusive load instruction (LDREXx)

Row 6: Normal write (STRx) to a different ID but from the same address as that of the exclusive load instruction (LDREXx)

Table 4-10 When Exclusive Load Instruction (LDREXx) and Exclusive Store Instruction (STREXx) Arrive Simultaneously (If It Arrives Simultaneously with STREXx, STREXx Takes Precedence)

Simultaneous Reception	Transfer Type	Ax ID	ADD	Response	Monitoring Register 1		Monitoring Register 2		Monitoring Register 3		Monitoring Register 4	
					ID	ADD	ID	ADD	ID	ADD	ID	ADD
	LDREXx	1	A	EXOKAY	1	A						
+	LDREXx	2	A	OKAY	1	A						
+	STREXx	0	A	OKAY	1	A						

Row 3: The response is STREXx=OKAY and monitoring registers are not cleared because the ID does not match the ID in the monitoring register.

Row 2: The response is LDREXx=OKAY and monitoring registers are not updated because the address matches the address of STREXx received simultaneously.

Table 4-11 When Exclusive Load Instruction (LDREXx) and Exclusive Store Instruction (STREXx) Arrive Simultaneously (If It Arrives Simultaneously with STREXx, STREXx Takes Precedence)

Simultaneous Reception	Transfer Type	Ax ID	ADD	Response	Monitoring Register 1		Monitoring Register 2		Monitoring Register 3		Monitoring Register 4	
					ID	ADD	ID	ADD	ID	ADD	ID	ADD
	LDREXx	1	A	EXOKAY	1	A						
+	LDREXx	2	B	EXOKAY	2	B						
+	STREXx	1	A	EXOKAY								

Row 3: The response is STREXx=EXOKAY and monitoring registers are cleared because both the ID and address match those in the monitoring register.

Row 2: The address does not match the address of STREXx received simultaneously. Therefore, it is held as a new monitoring target in the monitoring register.

Table 4-12 When Exclusive Load Instruction (LDREXx) and Exclusive Store Instruction (STREXx) Arrive Simultaneously (If It Arrives Simultaneously with STREXx, STREXx Takes Precedence)

Simultaneous Reception	Transfer Type	Ax ID	ADD	Response	Monitoring Register 1		Monitoring Register 2		Monitoring Register 3		Monitoring Register 4	
					ID	ADD	ID	ADD	ID	ADD	ID	ADD
	LDREXx	1	A	EXOKAY	1	A						
+	LDREXx	1	B	EXOKAY	1	B						
+	STREXx	0	A	OKAY	1	A						

Row 3: The response is STREXx=OKAY and monitoring registers are not cleared because the ID does not match the ID in the monitoring register.

Row 2: The monitoring register is updated because the address does not match the address of STREXx received simultaneously and the ID is the same as the ID in the monitoring register.

Table 4-13 When Exclusive Load Instruction (LDREXx) and Exclusive Store Instruction (STREXx) Arrive Simultaneously (If It Arrives Simultaneously with STREXx, STREXx Takes Precedence)

Simultaneous Reception	Transfer Type	Ax ID	ADD	Response	Monitoring Register 1		Monitoring Register 2		Monitoring Register 3		Monitoring Register 4	
					ID	ADD	ID	ADD	ID	ADD	ID	ADD
+	LDREXx	1	A	EXOKAY	1	A						
+	STREXx	0	B	OKAY								

Row 2: The response is STREXx=OKAY because there is no monitoring target in monitoring registers (no LDREXx is received before STREXx).

Row 1: The address does not match the address of STREXx received simultaneously. Therefore, it is held as a new monitoring target in the monitoring register.

Table 4-14 When Exclusive Load Instruction (LDREXx) Is Received Simultaneously with Normal Write (STRx) and They Have the Same Address (If It Is Received Simultaneously with Normal Write to the Same Address Range, OKAY Is Returned to LDREXx)

Simultaneous Reception	Transfer Type	Ax ID	ADD	Response	Monitoring Register 1		Monitoring Register 2		Monitoring Register 3		Monitoring Register 4	
					ID	ADD	ID	ADD	ID	ADD	ID	ADD
+	LDREXx	0	A	OKAY								
+	STRx	1	A	OKAY								

Row 2: The response is OKAY because the transfer type is normal write (STRx).

Row 1: The response is LDREXx=OKAY because the address is the same as that of normal Write (STRx).

Table 4-15 When Exclusive Load Instruction (LDREXx) Is Received Simultaneously with Normal Write (STRx) and They Have the Same Address (If It Is Received Simultaneously with Normal Write to the Same Address Range, OKAY Is Returned to LDREXx)

Simultaneous Reception	Transfer Type	Ax ID	ADD	Response	Monitoring Register 1		Monitoring Register 2		Monitoring Register 3		Monitoring Register 4	
					ID	ADD	ID	ADD	ID	ADD	ID	ADD
	LDREXx	1	B	EXOKAY	1	B						
+	LDREXx	0	B	OKAY								
+	STRx	2	B	OKAY								
	STREXx	1	B	OKAY								

Row 3: If the address is the same as in the monitoring register, the monitoring register is cleared.
Row 2: The response is LDREXx=OKAY because the address is same as that of normal Write (STRx).
Row 4: The response is STREXx=OKAY because the monitoring register is cleared by normal write (STRx).

Table 4-16 When Exclusive Load Instruction (LDREXx) Is Received Simultaneously with Normal Write (STRx) and Their Address Ranges Do Not Match

Simultaneous Reception	Transfer Type	Ax ID	ADD	Response	Monitoring Register 1		Monitoring Register 2		Monitoring Register 3		Monitoring Register 4	
					ID	ADD	ID	ADD	ID	ADD	ID	ADD
+	LDREXx	0	A	EXOKAY	0	A						
+	STRx	1	B	OKAY								

Table 4-17 When Exclusive Store Instruction (STREXx) Does Not Arrive after Exclusive Load Instruction (LDREXx) (Monitoring Continues Until STREXx Arrives)

Simultaneous Reception	Transfer Type	Ax ID	ADD	Response	Monitoring Register 1		Monitoring Register 2		Monitoring Register 3		Monitoring Register 4	
					ID	ADD	ID	ADD	ID	ADD	ID	ADD
	LDREXx	0	A	EXOKAY	0	A						
		…	…		0	A						
		…	…		0	A						
		…	…		0	A						

Table 4-18 When Address Ranges Overlap in Some Part and Exclusive Store Instruction (STREXx) with Previous Monitoring Target Address Is Received after Overwriting (When Address Ranges A and @A Overlap in Some Part, Monitoring Target Is Overwritten)

Simultaneous Reception	Transfer Type	Ax ID	ADD	Response	Monitoring Register 1		Monitoring Register 2		Monitoring Register 3		Monitoring Register 4	
					ID	ADD	ID	ADD	ID	ADD	ID	ADD
	LDREX	0	A	EXOKAY	0	A						
	LDREXD	0	@A	EXOKAY	0	@A						
	STREX	0	A	OKAY	0	@A						
	STREXD	0	@A	EXOKAY								

Row 2: The address range is overwritten with the later address range because the address ranges overlap in some part.

Row 3: OKAY is returned to STREX for the previous address range (that does not exactly match the monitoring target address). When the response is OKAY, WSTRB is dropped to prevent the relevant area being overwritten, and therefore the monitoring target continues to be used.

Table 4-19 When Address Ranges Overlap in Some Part and Exclusive Store Instruction (STREXx) with Previous Monitoring Target Address Is Received after Overwriting (When Address Ranges A and @A Overlap in Some Part, Monitoring Target Is Overwritten)

Simultaneous Reception	Transfer Type	Ax ID	ADD	Response	Monitoring Register 1		Monitoring Register 2		Monitoring Register 3		Monitoring Register 4	
					ID	ADD	ID	ADD	ID	ADD	ID	ADD
	LDREXD	0	@A	EXOKAY	0	@A						
	LDREX	0	A	EXOKAY	0	A						
	STREXD	0	@A	OKAY	0	A						
	STREX	0	A	EXOKAY								

Row 2: The address range is overwritten with the later address range because the address ranges overlap in some part.

Row 3: OKAY is returned to STREXD for the previous address range (that does not exactly match the monitoring target address). When the response is OKAY, WSTRB is dropped to prevent the relevant area being overwritten, and therefore the monitoring target continues to be used.

5. Memory Area

This section describes the memory area of the exclusive access memory (EAM).

Table 5-1 EAM Memory List

Abbreviated Memory Name	Memory Name	Reference
EAM	Exclusive access memory	5.1

Table 5-2 EAM Memory Layout

Offset	Register Name			
	+3	+2	+1	+0
0x0000_0000 | 0x0000_002C	Exclusive Access Memory 00000000_00000000_00000000_00000000			

5.1. Exclusive Access Memory

All 48-byte memories (exclusive access memories) mounted on the exclusive access memory (EAM) allow exclusive access. There are no special access restrictions for this memory area.

Base(0x0280_0000)+0x0000_0000

Bit	31 … 0
Field	Exclusive Access Memory
R/W Attribute	R/W
Protection Attribute	-
Initial Value	00000000_00000000_00000000_00000000

:

Base(0x0280_0000)+0x0000_002C

Bit	31 … 0
Field	Exclusive Access Memory
R/W Attribute	R/W
Protection Attribute	-
Initial Value	00000000_00000000_00000000_00000000

(Base+0x0000_0000 to Base+0x0000_002C)

[bit31:0] Exclusive access memory

These bits allow writing and reading of exclusive access data.

CHAPTER 22: Interrupt Controller

This chapter explains the interrupt controller.

1. **Overview**
2. **Configuration**
3. **Explanation of Operation**
4. **Setting Procedure Examples**
5. **Registers**
6. **Others**

1. Overview

This section provides an overview of the interrupt controller.

The interrupt controller supports 512 channels for normal interrupts (IRQs) and 32 channels for non-maskable interrupts (NMIs). All IRQs are processed so that they issue a single interrupt request (nIRQ) to the CPU. All NMIs are processed so that they issue a single fast interrupt request (nFIQ) to the CPU.

The interrupt controller has the following functions.

- Controls the issue of interrupts (nIRQ and nFIQ) to the CPU.
- Supports 512 IRQ channels for nIRQ generation.
- Supports 32 NMI channels for nFIQ generation.
- Default interrupt priority levels. (Those channels with the lower channel numbers take precedence.)
- Internally synchronizes asynchronous interrupts.
- IRQ priority level control which allows 32-level settings.
- Controls IRQ priority level masking.
- NMI priority level control which allows 16-level settings. (Note that only NMI0 is fixed to level 0.)
- Supports vectored interrupt for both IRQ and NMI.
- A 32-bit vector address per interrupt channel.
- Generates a software interrupt for both IRQ and NMI.
- Supports privileged mode for access restriction.

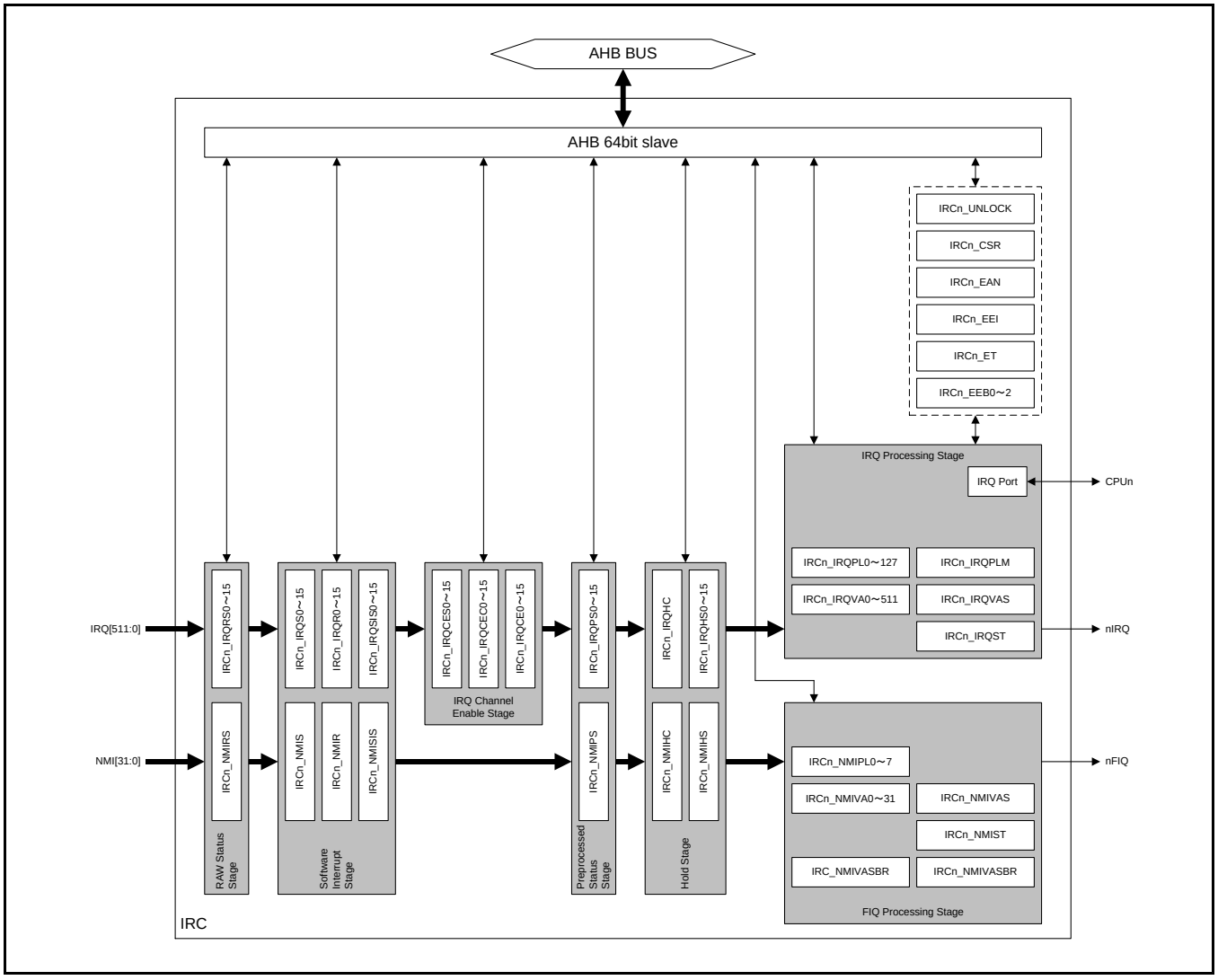
Notes:

- *In this chapter, you may see the abbreviated notation, IRC, which stands for InterRupt Controller. You may also see the notations ISR and VIC, which stand for Interrupt Service Routine and Vectored Interrupt Controller, respectively.*
- *Regarding the abbreviations used for the registers of the interrupt controller, the "n" in "IRCn_***" corresponds to the CPU number, "m" in "IRCn_**m" to the NMI channel number (0 to 31), and "r" in "IRCn_***r" to the IRQ channel number (0 to 511).*
- *For details on the interrupt factor assignment to each NMI/IRQ channel, see product specification.*

2. Configuration

This section explains the block diagram of the interrupt controller.

Figure 2-1 Block Diagram of Interrupt Controller



3. Explanation of Operation

This section explains the operation of the interrupt controller.

Interrupt Controller Stages

RAW Status Stage

For each IRQ/NMI channel, the interrupt controller posts notification of the RAW status. The RAW status is the interrupt status existing immediately after an input to the interrupt controller. In this status, the interrupt controller does not execute any processing. For both IRQ and NMI, notification of the status is obtained by reading the RAW status register (IRCn_IRQRS0 to IRCn_IRQRS15, IRCn_NMIRS).

Software Interrupt Stage

The interrupt controller can generate a software interrupt for each IRQ/NMI channel. By writing "1" to a bit in the IRQ software interrupt set register (IRCn_IRQS0 to IRCn_IRQS15), a software interrupt is set for the corresponding IRQ channel. By writing "1" to a bit of the IRQ software interrupt reset register (IRCn_IRQR0 to IRCn_IRQR15), the software interrupt for the corresponding IRQ channel is reset. The IRQ software interrupt status is obtained by reading the IRQ software interrupt status register (IRCn_IRQSIS0 to IRCn_IRQSIS15).

Similarly, for NMI, the interrupt controller has the NMI software interrupt set register (IRCn_NMIS), NMI software interrupt reset register (IRCn_NMIR), and NMI software interrupt status register (IRCn_NMISIS). The operation is same as for IRQ.

Although the interrupt controller supports 512 channels for IRQ and 32 channels for NMI, not every channel is assigned an interrupt factor. Therefore some channels are unused. These unused channels can be used for software interrupts. You can use software interrupts to test the ISRs for hardware interrupts.

IRQ Channel Enable Stage

For each IRQ channel, the interrupt controller can set either enable or disable. It cannot disable NMI channels. By writing "1" to a bit in the IRQ channel enable set register (IRCn_IRQCES0 to IRCn_IRQCES15), the corresponding IRQ channel is enabled. By writing "1" to a bit of the IRQ channel enable clear register (IRCn_IRQCEC0 to IRCn_IRQCEC15), the enabled status of the corresponding IRQ channel is cleared. The IRQ channel enable status is obtained by reading the IRQ channel enable setting register (IRCn_IRQCE0 to IRCn_IRQCE15). The interrupt controller can also be set either enable or disable by writing directly to the registers from IRCn_IRQCE0 to IRCn_IRQCE15.

Preprocessed Status Stage

The interrupt controller posts notification of the preprocessed interrupt status. In this case, preprocessing means the processing which is executed in the software interrupt stage and the IRQ channel enable stage. For both IRQ and NMI, the status is obtained by reading the preprocessed interrupt status register (IRCn_IRQPS0 to IRCn_IRQPS15, IRCn_NMIPS).

Hold Stage

For each IRQ/NMI channel, the interrupt controller holds the interrupt hold status. When the IRQ is applied to the CPU, the corresponding bit in the IRQ hold status register (IRCn_IRQHS0 to IRCn_IRQHS15) becomes "1". When the NMI is applied to the CPU and the CPU which has accepted the nFIQ reads IRC_NMIVASBR, the corresponding bit in the NMI hold status register (IRCn_NMIHS) becomes "1". After ISR ends, the corresponding hold status bit must be cleared. By writing, to the hold

clear register (IRCn_IRQHC, IRCn_NMIHC), the number of the channel for which the hold status is to be cleared, the corresponding hold status bit is cleared.

IRQ Processing Stage

IRQs are processed according to their software priority which is set in the IRQ priority level register (IRCn_IRQPL0 to IRCn_IRQPL127) and their hardware priority, and the channel having the highest priority level is applied. There are 32 software priority levels that can be set. If 2 requests have the same software priority level, that with the higher hardware priority level takes precedence. Hardware priority is fixed and that channel having the lower channel number has the higher priority level.

For IRQ, the interrupt controller has an interrupt mask function provided through the IRQ priority level mask register (IRCn_IRQPLM). If an IRQ channel has a software priority level (which is set in registers IRCn_IRQPL0 to IRCn_IRQPL127) that is higher than the value set in IRCn_IRQPLM, the channel is masked.

FIQ Processing Stage

NMIs are processed according to their software priority which is set in the NMI priority level register (IRCn_NMIPL0 to IRCn_NMIPL7) and their hardware priority, and the channel having the highest priority level is applied. There are 16 software priority levels that can be set. If 2 requests have the same software priority level, that with the higher hardware priority level takes precedence. Hardware priority is fixed and that channel having the lower channel number has the higher priority level. The software priority level of NMI0 is fixed to "0". Since NMI0 has the highest hardware priority level, it always has the highest priority.

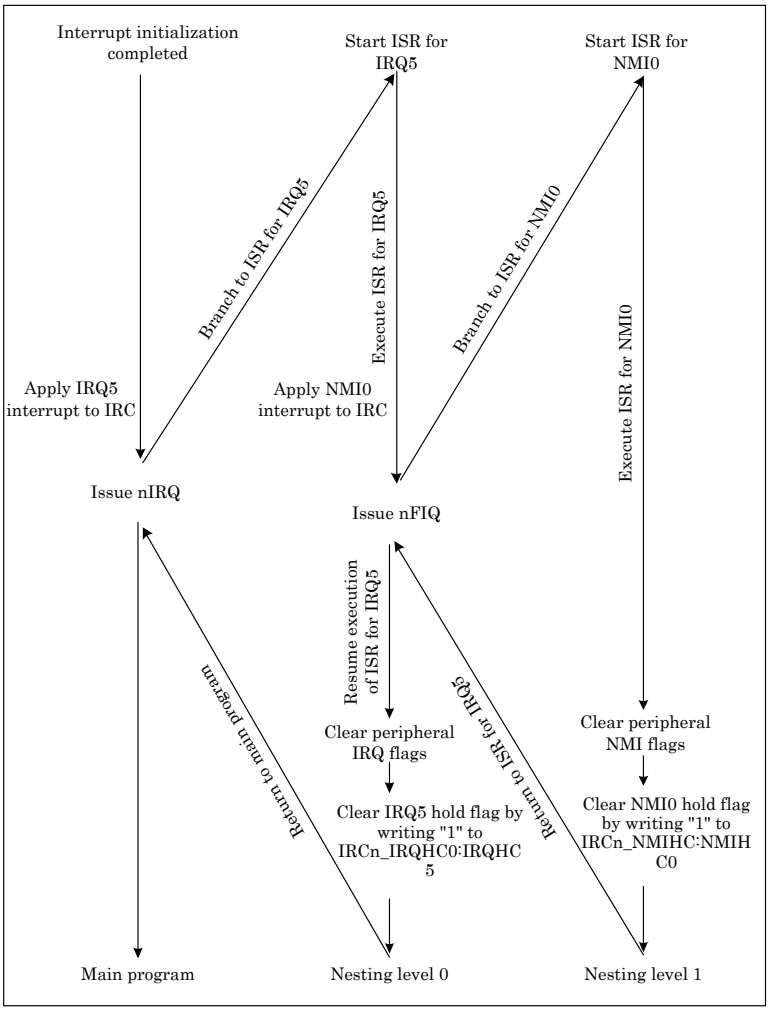
Nesting IRQ and NMI

The ARM® Cortex™-R5 register set does not automatically process nested interrupts. This processing should be provided within the ISR by applications. If this is not provided in the ISR, the first active IRQ is executed, but next IRQ will not be accepted and ISR will not be executed until the active ISR ends. This is also true for NMIs. However, since the interrupt controller uses nIRQ for IRQs and nFIQ for NMIs, and since NMIs have a higher hardware priority than that of IRQs, NMIs can interrupt IRQs.

For details on the save/restore of interrupts, see also the Cortex™-R5 Revision:r1p2 Technical Reference Manual (ARM DDI 0460D).

Interrupt Operation When Particular Software is Not Used for Nested Processing

Figure 3-1 ISR Processing Flow When NMI Generation Causes Nested IRQ

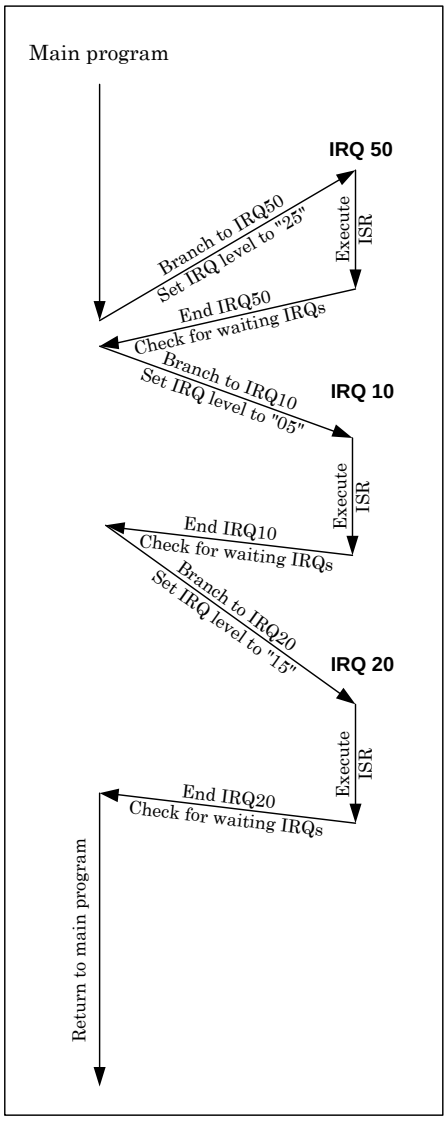


The following paragraphs explain the example shown in Figure 3-1.

When IRQ5 occurred, the interrupt controller issues nIRQ to the CPU. The interrupt controller sets the hold bit corresponding to IRQ5. The CPU holds the current program execution and saves CPU's current execution status. It then reads the vector address of IRQ5 and starts executing IRQ5's ISR. If NMI0 occurs during the previous process, the controller issues nFIQ to the CPU. Since NMIs have a higher priority than IRQs, the CPU stops executing the ISR for IRQ5 and saves CPU's current execution status. It then reads the vector address of NMI0. At this point, interrupt controller sets the hold flag corresponding to NMI0. Then starts executing NMI0's ISR. Before the ISR for NMI0 ends, the ISR clears the peripheral interrupt cause of NMI0. Then also clears NMI0's hold bit. After the ISR for NMI0 ends, the CPU resumes the ISR for IRQ5. Before the ISR for IRQ5 ends, the ISR clears the peripheral interrupt cause of IRQ5. Then also clears IRQ5's hold bit. After the ISR for IRQ5 ends, the CPU returns to the main program.

If the interrupt that is applied here is a software interrupt caused by a write operation to a register from IRCn_IRQS0 to IRCn_IRQS15 or IRCn_NMIS, the ISR needs to reset the software interrupt. This must be done by writing to a register between IRCn_IRQR0 and IRCn_IRQR15 or IRCn_NMIR, rather than by clearing the peripheral interrupt.

Figure 3-2 ISR Processing Flow When There Is No Nested IRQ



The following paragraph explains the example shown in Figure 3-2.

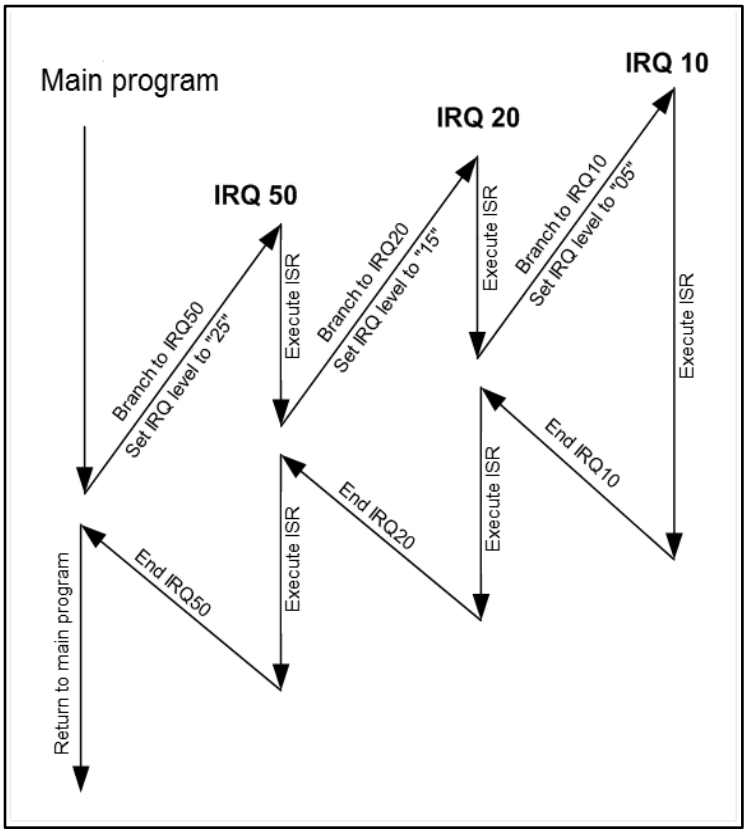
The priority levels of IRQ10, 20, and 50 used in this example are assumed to be "5", "15", and "25", respectively.

First, IRQ50 occurs. Since there are no waiting IRQs, IRQ50 is executed. At this point, "25" can be read from IRCn_IRQST:IRQPS. During the execution of the ISR for IRQ50, IRQ20 occurs. IRQ20 will wait

because ISR for IRQ50 has not ended. Then similarly, IRQ10 occurs. As IRQ20, IRQ10 waits until the ISR for IRQ50 ends. When the ISR for IRQ50 ends, IRQ10 and 20 enter the wait state. Since the priority level of IRQ10 is higher than that of IRQ20, which occurred earlier than IRQ10, the ISR for IRQ10 is executed next. At this point, "5" can be read from IRCn_IRQST:IRQPS. After the ISR for IRQ10 ends, the ISR for IRQ20 is executed. At this point, "15" can be read from IRCn_IRQST:IRQPS. After the ISR for IRQ20 ends, there are no other waiting IRQs. Therefore, the CPU returns to the main program.

Interrupt Operation When Particular Software Is Used for Nested Processing

Figure 3-3 ISR Processing Flow When There Are Nested IRQs



The following paragraphs explain the example shown in Figure 3-3.

As in the case of Figure 3-2, the priority levels of IRQ10, 20, and 50, used in this example, are assumed to be "5", "15", and "25", respectively.

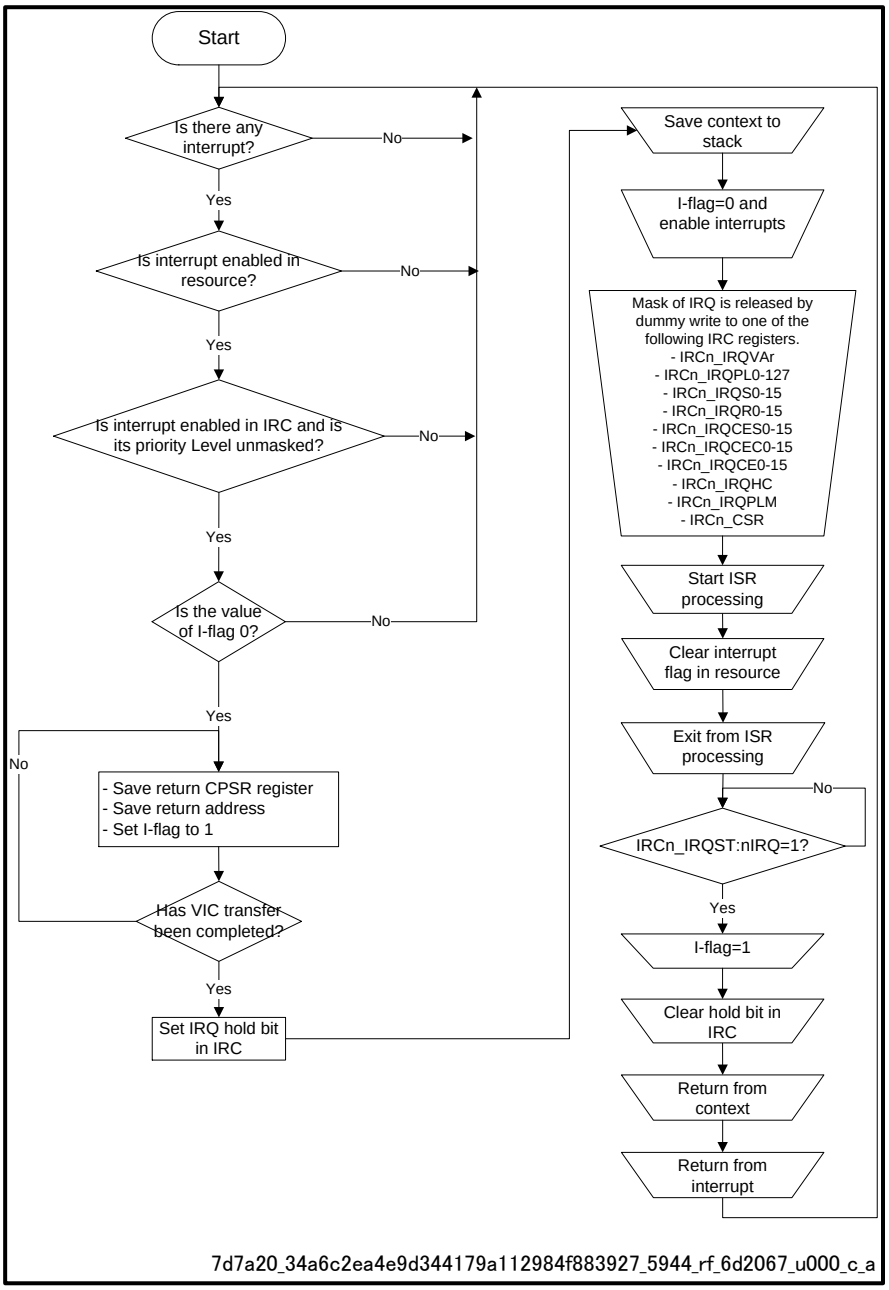
First, IRQ50 occurs. Since there are no waiting IRQs, IRQ50 is executed. At this point, "25" can be read from IRCn_IRQST:IRQPS. During the execution of the ISR for IRQ50, IRQ20 occurs. The ISR for IRQ50 is stopped and the ISR for IRQ20 is executed. At this point, "15" can be read from IRCn_IRQST:IRQPS. Then, similarly, IRQ10 occurs and the ISR for IRQ20 is stopped. At this point, "5" can be read from IRCn_IRQST:IRQPS. The ISR for IRQ10 is executed. After the ISR for IRQ10 ends, the ISR for IRQ20 will resume. After the ISR for IRQ20 ends, the ISR for IRQ50 will resume. After the ISR for IRQ50 ends, there are no other waiting IRQs. Therefore the CPU returns to the main program.

Example of Software Used to Process Nested IRQ/NMI

IRQ Processing

Since the ARM architecture does not support nested nIRQ interrupts, applications should be used to handle any nested processing. The ISR should internally save the values of 2 registers (LR_irq and SPSR_irq) and registers which values will be changed by ISR process in the system stack and then allow interrupts again. Then, if a new IRQ occurs and it has a higher priority level than that of the ISR that is being executed, it can interrupt the ISR. When the ISR ends, the saved values of the 2 registers (LR_irq and SPSR_irq) and registers which values will be changed by ISR process should be restored from the system stack.

Figure 3-4 Nest Supported IRQ Processing Flow



Below is an example of nest supported ISR in assembler code.

```
IRQxx_Handler:
SUB         lr, lr, #4
SRSFD       SP!,#0x1f           ; Save LR_irq and SPS_irq to the system stack.
CPS         #0x1f               ; Switch to system mode.
PUSH        {r0-r3, r12}        ; Save the remaining registers to the system stack.
AND         r1, sp, #4          ; Confirm that the stack is aligned to 8 bytes.
SUB         sp, sp, r1
PUSH        {r1, lr}
CPSIE       i                   ; Enable the interrupt.

...         ; ISR code that may be interrupted by other interrupts having a higher priority.
; End of ISR
CPSID       i                   ; Disable interrupt.
POP         {r1, lr}            ; Restore LR_sys and return the system stack to its previous
                                state.
ADD         sp, sp, r1
POP         {r0-r3, r12}        ; Restore the saved registers.
RFEFD       SP!                 ; Return from system mode.
```

Notes:

- *If you use nested IRQs or NMIs (or both), you should calculate the additional system stack size needed according to the depth of the nesting level being used.*
- *All registers which values will be changed by ISR process must be saved to system stack.*

NMI Processing

Fast interrupts are set as NMIs. The NMI flag is placed in the interrupt source module. This differs from IRQs in that there is no enable bit in the resource or interrupt controller.

The NMI entry is processed as follows:

1. The NMI interrupt flag is set by hardware or software and issued to the interrupt controller.
2. The interrupt controller evaluates the priority level. In a case when more than one NMI occurs, the NMI with the highest priority level is selected. Then the NMI is issued to the CPU.
3. The CPSR is copied to SPSR_fiq, and the return address is set in r14_fiq. Then, in the CPSR, the mode is set in fiq, and I-flag and F-flag are set (this disables the acceptance of new NMIs). This processing is all executed by the hardware.
4. Processor reads the register in interrupt controller which stores the vector address of selected NMI. This read access sets the hold bit for the corresponding NMI. This read operation is executed by the BootROM code (instruction in the FIQ entry).
5. Transit to the vector address that was read from the register in the interrupt controller. Then corresponding ISR is executed.
6. If nested NMIs support is required, the F-flag must be cleared by software.
   Process 7 must be done before this operation.
7. Before the F-flag is cleared, the context should be saved to the stack by software.
8. After the F-flag is cleared, another NMI having a higher priority level can interrupt the current NMI by nest.

If nested NMI support is required, operation 6 to 8 must be executed to save the context to stack. This could be expressed by below example. This is only an example. You may be able to use other assembler instructions.

- Execute SRS (save the return state). This instruction saves r14_fiq and SPSR_fiq to the stack (the address is selected through r13_fiq). r13_fiq is also updated as appropriate.
- Execute STM (save multiple registers). This instruction saves the defined number of registers (r0 to r14) to the stack that is defined in r13_fiq.
- Update r13_fiq as appropriate (cannot be used with STM).
- Clear the F-flag (enable nested NMIs).

The F-flag is set at every NMI accepted by CPU. Therefore, above save operation will not be stopped by other NMIs unless F-flag is cleared.

The steps for ending the NMI interrupt are as follows:

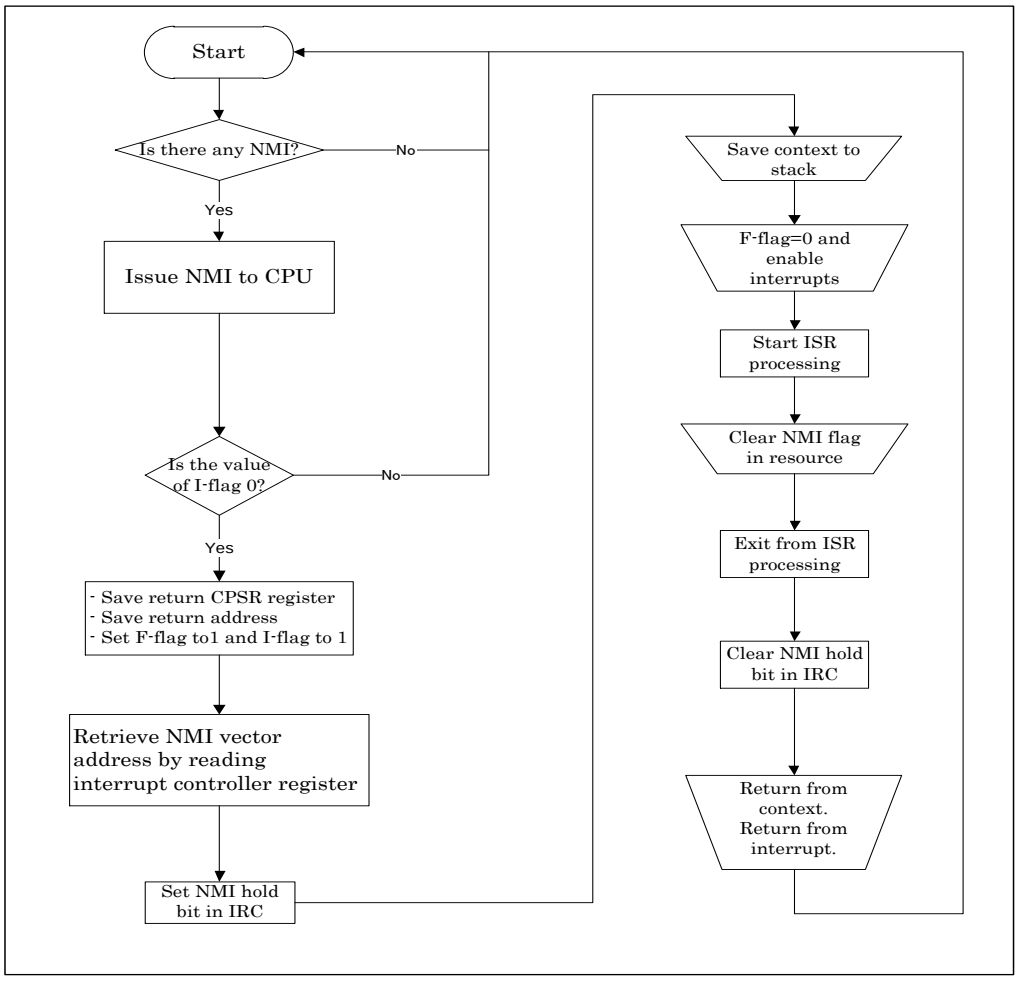
1. The NMI flag which caused the NMI interrupt must be cleared by software during the execution of the ISR. This should be completed as early as possible in the ISR to prevent the interrupt controller from accepting the same NMI flag.
2. The hold bit in the interrupt controller should be cleared during the execution of the ISR by writing the NMI number to an appropriate register (IRCn_NMIHC) in the interrupt controller. This should also be executed by the software.
3. After the above processing ends, the saved context including PC and CPSR in stack is restored in order to return from the interrupt. This operation must be done by software. When the CPSR is restored, the state of the register immediately before the generation of NMI is restored. Restoration is also applied to an enable status of an interrupt.

If nested NMI support is required, the software must execute the following processing to restore the saved context and to return from the interrupt. Several instructions must not be used to restore and returning from the interrupt. If executed, transition to a new NMI process may happen while restore and returning from the interrupt.

Execute LDM (end the exception handling). This instruction restores the saved user register as well as the CPSR and PC from the stack. r13_fiq is updated as appropriate. Required context to be restored must be restored by this one instruction only.

NMI processing flow is shown in Figure 3-5.

Figure 3-5 Nest Supported NMI Processing Flow



ECC Support and Test

The interrupt controller is equipped with SRAM which has an ECC protection function. Assume that a 1-bit error (correctable) occurs in the ECC. The IRCn_EEI:EEIS bit is set and the IRQ is generated for both IRCn_IRQVAr register reading and reading from the IRQ processing stage. If an error of 2 or more bits (uncorrectable) occurs, the IRCn_EEI:EENS bit is set and the NMI is generated at IRCn_IRQVAr register reading. Upon reading from the IRQ processing stage, transition is made to the ISR indicated by IRCn_IRQEEVA. Therefore, make sure that a valid ISR address is set for IRCn_IRQEEVA.

ECC Test Function

The interrupt controller has a function for generating pseudo ECC errors by corrupting the data read from SRAM. Using this function, you can check the operation of the ECC error handler while developing applications. By setting the ECC error bit registers (IRCn_EEB0 to IRCn_EEB2), you can invert the data read from SRAM by using the bitwise operation. This test function is enabled/disabled using the ECC test register (IRCn_ET).

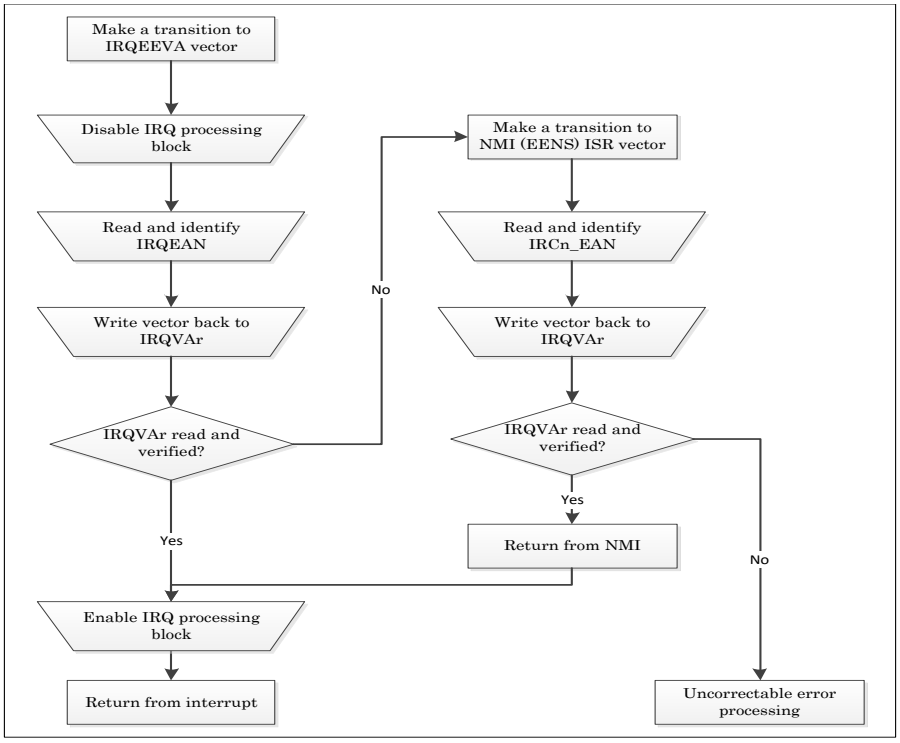
About IRCn_IRQEEVA

In the IRQ processing stage, if an SRAM 2-bit ECC error is detected, a transition is made to the ISR configured in IRCn_IRQEEVA. This behavior always occurs upon the occurrence of an error. So, IRCn_IRQEEVA must be properly set.

When a transition is made to this ISR, the IRQ hold bit for the erroneous IRCn_IRQVAr channel number is not generated. Moreover, this IRCn_IRQEEVA interrupt has no hold bit. Therefore, the interrupt factor need not be cleared within this ISR.

The ISR of IRCn_IRQEEVA needs to perform processing for correcting SRAM errors, using the following procedure.

Figure 3-6 SRAM 2BitECC Error Correction Flow by IRQEEVA



In this ISR, a software reset can be issued without any error correction, by rewriting to IRCn_IRQVAr.

How to Control Multiple Interrupts

The IRC controls the interrupt priority at a timing different from the CPU timing. So, if interrupt enable or disable is performed or if an interrupt level is changed, the software and hardware need to be synchronized according to a given procedure.

Procedure for Initializing and Setting the Interrupt Function

When initializing all interrupts for the first time, initialize and set the interrupts in the following order.

1. Setting peripheral function interrupts
2. Clear CPU I-FLAG
3. Setting the IRCn_CSR:IRQEN bit

Temporarily Enabling/Disabling Interrupts Using IRC_CSR_IRQEN with Interrupts Enabled

Before rewriting IRCn_CSR:IRQEN, verify that the IRCn_IRQST:nIRQ bit is 1. The following shows an example.

```
while(IRCn_IRQST:nIRQ==0) ; // if 0, repeat reading
IRCn_CSR:IRQEN=0; // safe for changing enable bit
```

Changing an Interrupt Level Using IRC_IRQPLM and IRC_IRQPLx with Interrupts Enabled

Before rewriting IRCn_IRQPLM and IRC_IRQPLx, verify that the IRCn_IRQST:nIRQ bit is 1. The following shows an example.

```
while(IRCn_IRQST:nIRQ==0) ; // if 0, repeat reading
IRCn_IRQPLM=0x2; // safe for changing interrupt level
```

Hold Clear Procedure Using IRC_IRQHC with Interrupts Enabled

Assume that writing to IRCn_IRQHC is to be performed after the I-FLAG clear instruction is executed within the interrupt handler. Before rewriting IRCn_IRQHC, verify that the IRCn_IRQST:nIRQ bit is 1. The following shows an example.

```
while(IRCn_IRQST:nIRQ==0) ; // if 0, repeat reading
IRCn_IRQHC=0x80; // safe for changing interrupt level
```

Processing at the Head of the IRQ Interrupt Handler

Until interrupt acceptance is enabled with the I-FLAG clear instruction in the interrupt processing handler, directly write the desired values to each of the IRCn_CSR:IRQEN, IRCn_IRQPLM, and IRCn_IRQHC registers, rather than using the procedure described above. Moreover, even if these register values do not need to be changed, write dummy values in one of the IRC registers listed below.

- IRCn_IRQVAr
- IRCn_IRQPL0-127
- IRCn_IRQS0-15
- IRCn_IRQR0-15

- IRCn_IRQCES0-15
- IRCn_IRQCEC0-15
- IRCn_IRQCE0-15
- IRCn_IRQHC
- IRCn_IRQPLM
- IRCn_CSR

Notes:

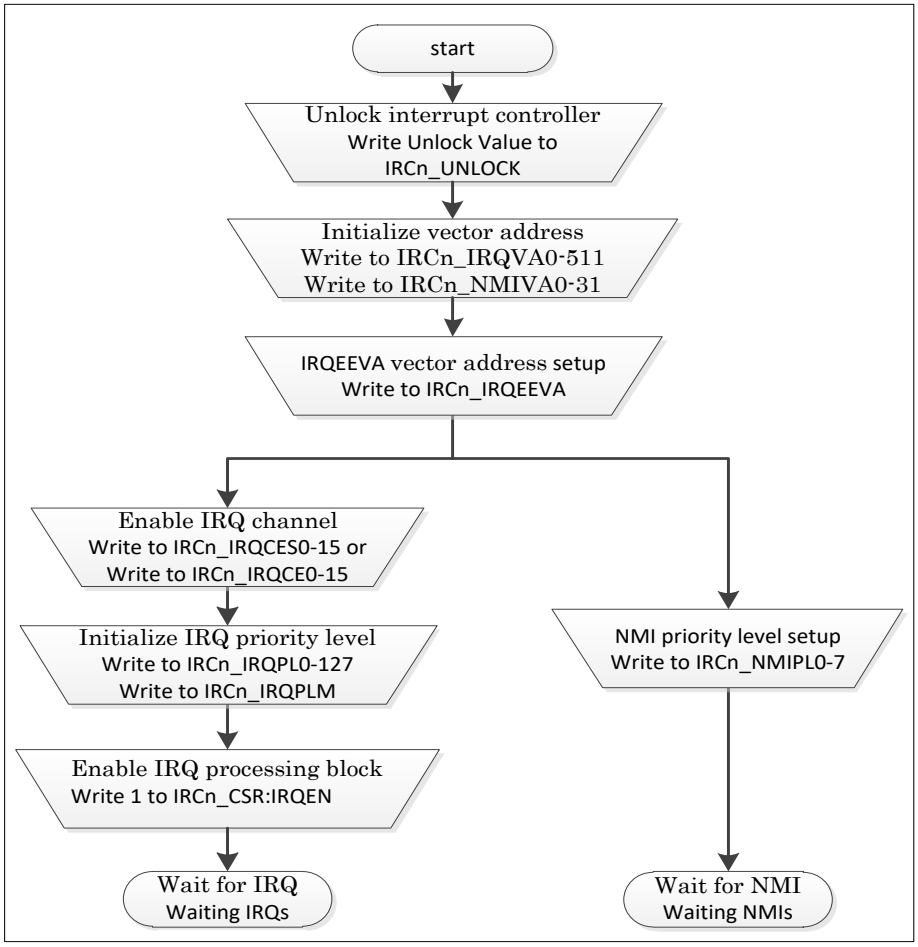
- *If dummy values are not written to any of the registers within the IRC, the generation of a new IRQ interrupt is suppressed until writing is complete. After writing to any of the registers within the IRC, the acceptance of a new IRQ interrupt is resumed.*
- *If the condition for returning a bus error is written during writing, interrupt acceptance is not released.*

4. Setting Procedure Examples

This section provides examples of the setting procedure for the interrupt controller.

Initial Setting

Figure 4-1 Interrupt Controller Initial Setting Flow



5. Registers

This section explains the registers that are used by the interrupt controller.

Table 5-1 Interrupt Controller Registers

Abbreviated Register Name	Register Name	Reference
IRCn_NMIVAS	IRC NMI vector address status register	5.1
IRCn_NMIST	IRC NMI status register	5.2
IRCn_IRQVAS	IRC IRQ vector address status register	5.3
IRCn_IRQST	IRC IRQ status register	5.4
IRCn_NMIVAm	IRC NMI vector address register	5.5
IRCn_IRQVAr	IRC IRQ vector address register	5.6
IRCn_NMIPL0	IRC NMI priority level register	5.7
IRCn_NMIPL1 to IRCn_NMIPL7	IRC NMI priority level register	5.8
IRCn_IRQPL0 to IRCn_IRQPL127	IRC IRQ priority level register	5.9
IRCn_NMIS	IRC NMI software interrupt set register	5.10
IRCn_NMIR	IRC NMI software interrupt reset register	5.11
IRCn_NMISIS	IRC NMI software interrupt status register	5.12
IRCn_IRQS0 to IRCn_IRQS15	IRC IRQ software interrupt set register	5.13
IRCn_IRQR0 to IRCn_IRQR15	IRC IRQ software interrupt reset register	5.14
IRCn_IRQSIS0 to IRCn_IRQSIS15	IRC IRQ software interrupt status register	5.15
IRCn_IRQCES0 to IRCn_IRQCES15	IRC IRQ channel enable set register	5.16
IRCn_IRQCEC0 to IRCn_IRQCEC15	IRC IRQ channel enable clear register	5.17
IRCn_IRQCE0 to IRCn_IRQCE15	IRC IRQ channel enable setting register	5.18
IRCn_NMIHC	IRC NMI hold clear register	5.19
IRCn_NMIHS	IRC NMI hold status register	5.20
IRCn_IRQHC	IRC IRQ hold clear register	5.21
IRCn_IRQHS0 to IRCn_IRQHS15	IRC IRQ hold status register	5.22
IRCn_IRQPLM	IRC IRQ priority level mask register	5.23
IRCn_CSR	IRC control/status register	5.24
IRCn_NMIRS	IRC NMI RAW status register	5.25
IRCn_NMIPS	IRC NMI preprocessed status register	5.26
IRCn_IRQRS0 to IRCn_IRQRS15	IRC IRQ RAW status register	5.27
IRCn_IRQPS0 to IRCn_IRQPS15	IRC IRQ preprocessed status register	5.28
IRCn_UNLOCK	IRC unlock register	5.29
IRCn_EEI	IRC ECC error interrupt register	5.30
IRCn_EAN	IRC ECC address number register	5.31
IRCn_ET	IRC ECC test register	5.32
IRCn_EEB0 to IRCn_EEB1	IRC ECC error bit register	5.33
IRCn_EEB2	IRC ECC error bit register	5.34
IRCn_NMIVASBR	IRC NMI vector address status register	5.35
IRC_NMIVASBR	IRC NMI vector address status mirror register	5.36
IRCn_IRQEEVA	IRC ECC error vector address register	5.37

Table 5-2 Register Memory Layout of Interrupt Controller (HSEL1)

Offset	+7	+6	+5	+4	+3	+2	+1	+0
0x00000000	IRCn_NMIST 00000000_00000000_00001111_00100000				IRCn_NMIVAS 00000000_00000000_00000000_00000000			
0x00000008	IRCn_IRQST 00000001_00011111_00000010_00000000				IRCn_IRQVAS 00000000_00000000_00000000_00000000			
0x00000010	IRCn_NMIVA1 XXXXXXXX_XXXXXXXX_XXXXXXXX_XXXXXX00				IRCn_NMIVA0 XXXXXXXX_XXXXXXXX_XXXXXXXX_XXXXXX00			
|	IRCn_NMIVA2 to IRCn_NMIVA29 XXXXXXXX_XXXXXXXX_XXXXXXXX_XXXXXX00							
0x00000088	IRCn_NMIVA31 XXXXXXXX_XXXXXXXX_XXXXXXXX_XXXXXX00				IRCn_NMIVA30 XXXXXXXX_XXXXXXXX_XXXXXXXX_XXXXXX00			
0x00000090	IRCn_IRQVA1 XXXXXXXX_XXXXXXXX_XXXXXXXX_XXXXXX00				IRCn_IRQVA0 XXXXXXXX_XXXXXXXX_XXXXXXXX_XXXXXX00			
|	IRCn_IRQVA2 to IRCn_IRQVA509 XXXXXXXX_XXXXXXXX_XXXXXXXX_XXXXXX00							
0x00000888	IRCn_IRQVA511 XXXXXXXX_XXXXXXXX_XXXXXXXX_XXXXXX00				IRCn_IRQVA510 XXXXXXXX_XXXXXXXX_XXXXXXXX_XXXXXX00			
0x00000890	IRCn_NMIPL1 00001111_00001111_00001111_00001111				IRCn_NMIPL0 00001111_00001111_00001111_00000000			
|	IRCn_NMIPL2 to IRCn_NMIPL5 00001111_00001111_00001111_00001111							
0x000008A8	IRCn_NMIPL7 00001111_00001111_00001111_00001111				IRCn_NMIPL6 00001111_00001111_00001111_00001111			
0x000008B0	IRCn_IRQPL1 00011111_00011111_00011111_00011111				IRCn_IRQPL0 00011111_00011111_00011111_00011111			
|	IRCn_IRQPL2 to IRCn_IRQPL125 00011111_00011111_00011111_00011111							
0x00000AA8	IRCn_IRQPL127 00011111_00011111_00011111_00011111				IRCn_IRQPL126 00011111_00011111_00011111_00011111			
0x00000AB0	IRCn_NMIR 00000000_00000000_00000000_00000000				IRCn_NMIS 00000000_00000000_00000000_00000000			
0x00000AB8	Reserved				IRCn_NMISIS 00000000_00000000_00000000_00000000			
0x00000AC0	IRCn_IRQS1 00000000_00000000_00000000_00000000				IRCn_IRQS0 00000000_00000000_00000000_00000000			
|	IRCn_IRQS2 to IRCn_IRQS13 00000000_00000000_00000000_00000000							
0x00000AF8	IRCn_IRQS15 00000000_00000000_00000000_00000000				IRCn_IRQS14 00000000_00000000_00000000_00000000			
0x00000B00	IRCn_IRQR1 00000000_00000000_00000000_00000000				IRCn_IRQR0 00000000_00000000_00000000_00000000			
|	IRCn_IRQR2 to IRCn_IRQR13 00000000_00000000_00000000_00000000							
0x00000B38	IRCn_IRQR15 00000000_00000000_00000000_00000000				IRCn_IRQR14 00000000_00000000_00000000_00000000			

Offset	+7	+6	+5	+4	+3	+2	+1	+0
0x00000B40	IRCn_IRQSIS1 00000000_00000000_00000000_00000000				IRCn_IRQSIS0 00000000_00000000_00000000_00000000			
|	IRCn_IRQSIS2 to IRCn_IRQSIS13 00000000_00000000_00000000_00000000							
0x00000B78	IRCn_IRQSIS15 00000000_00000000_00000000_00000000				IRCn_IRQSIS14 00000000_00000000_00000000_00000000			
0x00000B80	IRCn_IRQCES1 00000000_00000000_00000000_00000000				IRCn_IRQCES0 00000000_00000000_00000000_00000000			
|	IRCn_IRQCES2 to IRCn_IRQCES13 00000000_00000000_00000000_00000000							
0x00000BB8	IRCn_IRQCES15 00000000_00000000_00000000_00000000				IRCn_IRQCES14 00000000_00000000_00000000_00000000			
0x00000BC0	IRCn_IRQCEC1 00000000_00000000_00000000_00000000				IRCn_IRQCEC0 00000000_00000000_00000000_00000000			
|	IRCn_IRQCEC2 to IRCn_IRQCEC13 00000000_00000000_00000000_00000000							
0x00000BF8	IRCn_IRQCEC15 00000000_00000000_00000000_00000000				IRCn_IRQCEC14 00000000_00000000_00000000_00000000			
0x00000C00	IRCn_IRQCE1 00000000_00000000_00000000_00000000				IRCn_IRQCE0 00000000_00000000_00000000_00000000			
|	IRCn_IRQCE2 to IRCn_IRQCE13 00000000_00000000_00000000_00000000							
0x00000C38	IRCn_IRQCE15 00000000_00000000_00000000_00000000				IRCn_IRQCE14 00000000_00000000_00000000_00000000			
0x00000C40	IRCn_NMIHS 00000000_00000000_00000000_00000000				IRCn_NMIHC 00000000_00000000_00000000_00000000			
0x00000C48	Reserved				IRCn_IRQHC 00000000_00000000_00000000_00000000			
0x00000C50	IRCn_IRQHS1 00000000_00000000_00000000_00000000				IRCn_IRQHS0 00000000_00000000_00000000_00000000			
|	IRCn_IRQHS2 to IRCn_IRQHS13 00000000_00000000_00000000_00000000							
0x00000C88	IRCn_IRQHS15 00000000_00000000_00000000_00000000				IRCn_IRQHS14 00000000_00000000_00000000_00000000			
0x00000C90	Reserved				IRCn_IRQPLM 00000000_00000000_00000000_00100000			
0x00000C98	Reserved				IRCn_CSR 00000000_00000001_00000000_00000000			
0x00000CA0	Reserved				Reserved			
0x00000CA8	IRCn_NMIPS 00000000_00000000_00000000_00000000				IRCn_NMIRS 00000000_00000000_00000000_00000000			
0x00000CB0	IRCn_IRQRS1 00000000_00000000_00000000_00000000				IRCn_IRQRS0 00000000_00000000_00000000_00000000			
|	IRCn_IRQRS2 to IRCn_IRQRS13 00000000_00000000_00000000_00000000							

Offset	+7	+6	+5	+4	+3	+2	+1	+0
0x00000CE8	IRCn_IRQRS15 00000000_00000000_00000000_00000000				IRCn_IRQRS14 00000000_00000000_00000000_00000000			
0x00000CF0	IRCn_IRQPS1 00000000_00000000_00000000_00000000				IRCn_IRQPS0 00000000_00000000_00000000_00000000			
|	IRCn_IRQPS2 to IRCn_IRQPS13 00000000_00000000_00000000_00000000							
0x00000D28	IRCn_IRQPS15 00000000_00000000_00000000_00000000				IRCn_IRQPS14 00000000_00000000_00000000_00000000			
0x00000D30	Reserved				IRCn_UNLOCK 00000000_00000000_00000000_00000000			
0x00000D38	IRCn_IRQEEVA 00000000_00000000_00000000_00000000				Reserved			
0x00000D40	IRCn_EAN 00000000_00000000_00000000_00000000				IRCn_EEI 00000000_00000000_00000000_00000000			
0x00000D48	IRCn_EEB0 00000000_00000000_00000000_00000000				IRCn_ET 00000000_00000000_00000000_00000000			
0x00000D50	IRCn_EEB2 00000000_00000000_00000000_00000000				IRCn_EEB1 00000000_00000000_00000000_00000000			
0x00000D58	Reserved				Reserved			
0x00000D60	Reserved				Reserved			
0x00000D68	Reserved				Reserved			
0x00000D70	Reserved				Reserved			
0x00000D78 | 0x00000FF8	Reserved				Reserved			

Register Memory Layout of Interrupt Controller (HSEL2)

Offset	+7	+6	+5	+4	+3	+2	+1	+0
0x00000000 | 0x000003F0	Reserved				Reserved			
0x000003F8	IRCn_NMIVASBR 00000000_00000000_00000000_00000000				Reserved			

* In the ERRCFG area, 1 KB should be allocated separately for each CPU.

Register Memory Layout of Interrupt Controller (HSEL3)

Offset	+7	+6	+5	+4	+3	+2	+1	+0
0x00000000 | 0x000003F0	Reserved				Reserved			
0x000003F8	IRC_NMIVASBR 00000000_00000000_00000000_00000000				Reserved			

* In the ERRCFG area, 1 KB should be allocated and shared by all CPUs.

5.1. IRC NMI Vector Address Status Register (IRCn_NMIVAS)

This register indicates the vector address status of the NMI that the CPU accepted most recently.

Bit	31	0
Field	NMIVAS	
R/W Attribute	R,WX	
Protection Attribute	RP	
Initial Value	00000000_00000000_00000000_00000000	

[bit31:0] NMIVAS: NMI Vector Address Status Bits

These bits indicate the vector address status of the NMI that the CPU accepted most recently.

Note:

- *By reading the IRC_NMIVASBR by the CPU which accepted the nFIQ, the interrupt controller will determine that the interrupt has been accepted and then update the register.*

5.2. IRC NMI Status Register (IRCn_NMIST)

This register indicates the priority level and channel number status of the NMI that the CPU accepted most recently.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	Reserved															
R/W Attribute	R0,WX															
Protection Attribute	-															
Initial Value	00000000_00000000															

Bit	15	14	13	12	11	10	9	8
Field	Reserved				NMIPS			
R/W Attribute	R0,WX				R,WX			
Protection Attribute	-				-			
Initial Value	0000				1111			

Bit	7	6	5	4	3	2	1	0
Field	Reserved		NMISN					
R/W Attribute	R0,WX		R,WX					
Protection Attribute	-		-					
Initial Value	00		100000					

[bit31:12] Reserved: Reserved Bits

[bit11:8] NMIPS: NMI Priority Status Bits

These bits indicate the priority status (0 to 15) of the NMI that the CPU accepted most recently.

[bit7:6] Reserved: Reserved Bits

[bit5:0] NMISN: NMI Channel Number Bits

These bits indicate the channel number status (0 to 31) of the NMI that the CPU accepted most recently. The initial value indicates that no interrupt has been accepted; bit5 will never be "1" unless it is in the initial state.

Note:

- *By reading the IRC_NMIVASBR by the CPU to which the nFIQ is applied, the interrupt controller will determine that the interrupt has been accepted and then update the register.*

5.3. IRC IRQ Vector Address Status Register (IRCn_IRQVAS)

This register indicates the vector address status of the IRQ that the CPU accepted most recently.

Bit	31	0
Field	IRQVAS	
R/W Attribute	R,WX	
Protection Attribute	-	
Initial Value	00000000_00000000_00000000_00000000	

[bit31:0] IRQVAS: IRQ Vector Address Status Bits

These bits indicate the vector address status of the IRQ that the CPU accepted most recently.

Notes:

- *By receiving acknowledgment of the VIC port from the CPU which accepted the nIRQ, the interrupt controller will determine that the interrupt has been accepted and then update the register.*
- *If a 2-bit ECC error is detected in the IRQ processing block and a branch is made to the IRQEEVA handler, the value of the erroneous IRQ is read as an IRQVAS value.*

5.4. IRC IRQ Status Register (IRCn_IRQST)

This register indicates the priority level and channel number status of the IRQ that the CPU accepted most recently. This register also indicates the suppress status of newly accepted interrupts.

Bit	31	30	29	28	27	26	25	24
Field	Reserved							nIRQ
R/W Attribute	R0,WX							R,WX
Protection Attribute	-							-
Initial Value	0000000							1

Bit	23	22	21	20	19	18	17	16
Field	Reserved			IRQPS				
R/W Attribute	R0,WX			R,WX				
Protection Attribute	-			-				
Initial Value	000			11111				

Bit	15	14	13	12	11	10	9	8
Field	Reserved						IRQSN[9:8]	
R/W Attribute	R0,WX						R,WX	
Protection Attribute	-						-	
Initial Value	000000						10	

Bit	7	6	5	4	3	2	1	0
Field	IRQSN[7:0]							
R/W Attribute	R,WX							
Protection Attribute	-							
Initial Value	00000000							

[bit31:25] Reserved: Reserved Bits

[bit24] nIRQ: IRQ Interrupt Status Bit

If 0 is read from this bit, the IRC is either generating the IRQ for the CPU or IRQ generation to CPU is stopped by IRC. Once this bit is read, the IRC will not accept any subsequent interrupts. Moreover, if data is written to any one of the IRC registers listed below, the stopped interrupt acceptance is released.

- IRCn_IRQVAr
- IRCn_IRQPL0-127
- IRCn_IRQS0-15
- IRCn_IRQR0-15
- IRCn_IRQCES0-15
- IRCn_IRQCEC0-15
- IRCn_IRQCE0-15
- IRCn_IRQHC
- IRCn_IRQPLM
- IRCn_CSR

Note:

- *If the condition for returning a bus error is written during writing, no interrupt acceptance is released.*

[bit23:21] Reserved: Reserved Bits

[bit20:16] IRQPS: IRQ Priority Status Bits

These bits indicate the priority status (0 to 31) of the IRQ that the CPU accepted most recently.

[bit15:10] Reserved: Reserved Bits

[bit9:0] IRQSN: IRQ Channel Number Bits

These bits indicate the channel number status (0 to 511) of the IRQ that the CPU accepted most recently. The initial value means that no interrupt has been accepted; bit9 will never be "1" unless it is in the initial state.

Notes:

- *By receiving the acknowledgment of VIC port from the CPU to which the nIRQ is applied, the interrupt controller will determine that the interrupt has been accepted and will then update the register.*
- *If a 2-bit ECC error is detected in the IRQ processing block and a branch is made to the IRQEEVA handler, the value of the erroneous IRQ is read as the IRQSN and IRQPS register values.*

5.5. IRC NMI Vector Address Register (IRCn_NMIVAm)

This register indicates the 32-bit vector address of individual NMI channels. Before the corresponding NMI peripheral is enabled, the vector address should be initialized by software. The same type of register is installed for each NMI channel. The "m" in the abbreviated register name corresponds to the NMI channel number, m (0 to 31).

Bit	31	8
Field	NMIVA[31:8]	
R/W Attribute	R/W	
Protection Attribute	WPS	
Initial Value	XXXXXXXX_XXXXXXXX_XXXXXXXX	

Bit	7	6	5	4	3	2	1	0
Field	NMIVA[7:2]						NMIVA[1:0]	
R/W Attribute	R/W						R0,WX	
Protection Attribute	WPS						-	
Initial Value	XXXXXX						00	

[bit31:0] NMIVA: NMI Vector Address Bits

These bits set the vector address of NMIm.

Notes:

- *Since all vector addresses are 32 bits long, the lowest 2 bits are fixed to 0b00.*
- *The bit configurations of registers IRCn_NMIVA0 to IRCn_NMIVA31 are all the same.*

5.6. IRC IRQ Vector Address Register (IRCn_IRQVAr)

This register indicates the 32-bit vector address of individual IRQ channels. Before the corresponding IRQ channel is enabled, the vector address should be initialized by software. The same type of register is installed for each IRQ channel. The "r" in the abbreviated register name corresponds to the IRQ channel number, r (0 to 511). This register is on SRAM of the interrupt controller.

Bit	31	8
Field	IRQVA[31:8]	
R/W Attribute	R/W	
Protection Attribute	WPS	
Initial Value	XXXXXXXX_XXXXXXXX_XXXXXXXX	

Bit	7	6	5	4	3	2	1	0
Field	IRQVA[7:2]						IRQVA[1:0]	
R/W Attribute	R/W						R0,WX	
Protection Attribute	WPS						-	
Initial Value	XXXXXX						00	

[bit31:0] IRQVA: IRQ Vector Address Bits

These bits set the vector address of IRQr.

Notes:

- *Before this register is read, the vector address should be initialized. For initialization, use either 32-bit or 64-bit access.*
- *Since all vector addresses are 32 bits long, the lowest 2 bits are fixed to 0b00.*
- *The bit configurations of registers IRCn_IRQVA0 to IRCn_IRQVA511 are all the same.*
- *This register consists of SRAM having the ECC protection function. If an ECC Error occurs, overwrite the register with the correct values.*

5.7. IRC NMI Priority Level Register (IRCn_NMIPL0)

This register indicates the priority level of NMI0. It also sets the priority level (0 to 15) of each of channels NMI1 to NMI3. For each channel, a different value can be set. The smaller the setting value, the higher the priority. The initial value for channels other than channel 0 is "15", which indicates the lowest level. For channel 0 only, the value is fixed to "0", which is the highest level.

Bit	31	30	29	28	27	26	25	24
Field	Reserved				NMIPL3			
R/W Attribute	R0,WX				R/W			
Protection Attribute	-				WPS			
Initial Value	0000				1111			

Bit	23	22	21	20	19	18	17	16
Field	Reserved				NMIPL2			
R/W Attribute	R0,WX				R/W			
Protection Attribute	-				WPS			
Initial Value	0000				1111			

Bit	15	14	13	12	11	10	9	8
Field	Reserved				NMIPL1			
R/W Attribute	R0,WX				R/W			
Protection Attribute	-				WPS			
Initial Value	0000				1111			

Bit	7	6	5	4	3	2	1	0
Field	Reserved				NMIPL0			
R/W Attribute	R0,WX				R0,WX			
Protection Attribute	-				-			
Initial Value	0000				0000			

[bit31:28] Reserved: Reserved Bits

[bit27:24] NMIPL3: NMI3 Priority Level Bits

These bits set the priority level of NMI3.

[bit23:20] Reserved: Reserved Bits

[bit19:16] NMIPL2: NMI2 Priority Level Bits
These bits set the priority level of NMI2.

[bit15:12] Reserved: Reserved Bits

[bit11:8] NMIPL1: NMI1 Priority Level Bits
These bits set the priority level of NMI1.

[bit7:4] Reserved: Reserved Bits

[bit3:0] NMIPL0: NMI0 Priority Level Bits
These bits indicate the priority level of NMI0. The priority level of NMI0 is fixed to "0".

5.8. IRC NMI Priority Level Register (IRCn_NMIPL1 to IRCn_NMIPL7)

These registers set the priority level (0 to 15) of each NMI channel. For each channel, a different value can be set. The smaller the setting value, the higher the priority. The initial value is "15", which is the lowest level. The 7 registers are of the same type, and each is identified by the number at the end of the abbreviated register name. For each register, 4 channels are assigned, so that 32 channels are supported by the 8 registers including IRCn_NMIPL0.

Bit	31	30	29	28	27	26	25	24
Field	Reserved				NMIPL7			
R/W Attribute	R0,WX				R/W			
Protection Attribute	-				WPS			
Initial Value	0000				1111			

Bit	23	22	21	20	19	18	17	16
Field	Reserved				NMIPL6			
R/W Attribute	R0,WX				R/W			
Protection Attribute	-				WPS			
Initial Value	0000				1111			

Bit	15	14	13	12	11	10	9	8
Field	Reserved				NMIPL5			
R/W Attribute	R0,WX				R/W			
Protection Attribute	-				WPS			
Initial Value	0000				1111			

Bit	7	6	5	4	3	2	1	0
Field	Reserved				NMIPL4			
R/W Attribute	R0,WX				R/W			
Protection Attribute	-				WPS			
Initial Value	0000				1111			

[bit31:28] Reserved: Reserved Bits

[bit27:24] NMIPL7: NMI7 Priority Level Bits

These bits set the priority level of NMI7.

[bit23:20] Reserved: Reserved Bits

[bit19:16] NMIPL6: NMI6 Priority Level Bits
These bits set the priority level of NMI6.

[bit15:12] Reserved: Reserved Bits

[bit11:8] NMIPL5: NMI5 Priority Level Bits
These bits set the priority level of NMI5.

[bit7:4] Reserved: Reserved Bits

[bit3:0] NMIPL4: NMI4 Priority Level Bits
These bits set the priority level of NMI4.

Notes:

- *The register bit configuration described in this section is for IRCn_NMIPL1, which is representative of those registers that are all the same. The bit configurations of registers IRCn_NMIPL1 to IRCn_NMIPL7 are all the same. The description of IRCn_NMIPL0 is separated from others because NMI0 is handled differently from other channels.*
- *NMI4 to NMI7 are assigned to IRCn_NMIPL1, and the subsequent channels are assigned in the same way until NMI28 to NMI31 are assigned to IRCn_NMIPL7.*
- *The "m" in bit field name NMIPLm corresponds to the NMI channel number, m (0 to 31).*

5.9. IRC IRQ Priority Level Register (IRCn_IRQPL0 to IRCn_IRQPL127)

These registers set the priority level (0 to 31) of each IRQ channel. For each channel, a different value can be set. The smaller the setting value, the higher the priority. The initial value is "31", which is the lowest level. The 128 registers are all of the same type, and each is identified by the number at the end of the abbreviated register name. For each register, 4 channels are assigned, so that 512 channels are supported by the 128 registers.

Bit	31	30	29	28	27	26	25	24
Field	Reserved			IRQPL3				
R/W Attribute	R0,WX			R/W				
Protection Attribute	-			WPS				
Initial Value	000			11111				

Bit	23	22	21	20	19	18	17	16
Field	Reserved			IRQPL2				
R/W Attribute	R0,WX			R/W				
Protection Attribute	-			WPS				
Initial Value	000			11111				

Bit	15	14	13	12	11	10	9	8
Field	Reserved			IRQPL1				
R/W Attribute	R0,WX			R/W				
Protection Attribute	-			WPS				
Initial Value	000			11111				

Bit	7	6	5	4	3	2	1	0
Field	Reserved			IRQPL0				
R/W Attribute	R0,WX			R/W				
Protection Attribute	-			WPS				
Initial Value	000			11111				

[bit31:29] Reserved: Reserved Bits

[bit28:24] IRQPL3: IRQ3 Priority Level Bits

These bits set the priority level of IRQ3.

[bit23:21] Reserved: Reserved Bits

[bit20:16] IRQPL2: IRQ2 Priority Level Bits
These bits set the priority level of IRQ2.

[bit15:13] Reserved: Reserved Bits

[bit12:8] IRQPL1: IRQ1 Priority Level Bits
These bits set the priority level of IRQ1.

[bit7:5] Reserved: Reserved Bits

[bit4:0] IRQPL0: IRQ0 Priority Level Bits
These bits set the priority level of IRQ0.

Notes:

- *The register bit configuration described in this section is for IRCn_IRQPL0, which is representative of those registers that are all the same The bit configurations of registers IRCn_IRQPL0 to IRCn_IRQPL127 are all the same.*
- *IRQ0 to IRQ3 are assigned to IRCn_IRQPL0, and the subsequent channels are assigned in the same way until IRQ508 to IRQ511 are assigned to IRCn_IRQPL127.*
- *In case of writing to this register, please perform it in any of the following states (= IRC is stopping).*
  - Until writing the target register from reading that nIRQ bit of IRCn_IRQST register is "1". Please refer to the description of nIRQ bit for the target register.
  - The state that IRQEN bit of IRCn_CSR register is "0" (immediately after reset, etc).

5.10. IRC NMI Software Interrupt Set Register (IRCn_NMIS)

This register sets a software interrupt for individual NMI channels. For each channel, 1 bit is assigned, and the bit location corresponds to the NMI channel number, m (0 to 31).

Bit	31	30	29	28	27	26	25	24
Field	NMIS31	NMIS30	NMIS29	NMIS28	NMIS27	NMIS26	NMIS25	NMIS24
R/W Attribute	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
Protection Attribute	WPS	WPS	WPS	WPS	WPS	WPS	WPS	WPS
Initial Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	NMIS23	NMIS22	NMIS21	NMIS20	NMIS19	NMIS18	NMIS17	NMIS16
R/W Attribute	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
Protection Attribute	WPS	WPS	WPS	WPS	WPS	WPS	WPS	WPS
Initial Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	NMIS15	NMIS14	NMIS13	NMIS12	NMIS11	NMIS10	NMIS9	NMIS8
R/W Attribute	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
Protection Attribute	WPS	WPS	WPS	WPS	WPS	WPS	WPS	WPS
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	NMIS7	NMIS6	NMIS5	NMIS4	NMIS3	NMIS2	NMIS1	NMIS0
R/W Attribute	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
Protection Attribute	WPS	WPS	WPS	WPS	WPS	WPS	WPS	WPS
Initial Value	0	0	0	0	0	0	0	0

[bit31] to [bit0] NMISm: NMIm Software Interrupt Set Bits

These bits set a software interrupt for the corresponding NMIm.

Bit	Description
0	Invalid (no effect on operation)
1	Set a software interrupt for the corresponding NMIm.

5.11. IRC NMI Software Interrupt Reset Register (IRCn_NMIR)

This register resets the software interrupts for individual NMI channels. For each channel, 1 bit is assigned, and the bit location corresponds to the NMI channel number, m (0 to 31).

Bit	31	30	29	28	27	26	25	24
Field	NMIR31	NMIR30	NMIR29	NMIR28	NMIR27	NMIR26	NMIR25	NMIR24
R/W Attribute	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
Protection Attribute	WP	WP	WP	WP	WP	WP	WP	WP
Initial Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	NMIR23	NMIR22	NMIR21	NMIR20	NMIR19	NMIR18	NMIR17	NMIR16
R/W Attribute	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
Protection Attribute	WP	WP	WP	WP	WP	WP	WP	WP
Initial Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	NMIR15	NMIR14	NMIR13	NMIR12	NMIR11	NMIR10	NMIR9	NMIR8
R/W Attribute	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
Protection Attribute	WP	WP	WP	WP	WP	WP	WP	WP
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	NMIR7	NMIR6	NMIR5	NMIR4	NMIR3	NMIR2	NMIR1	NMIR0
R/W Attribute	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
Protection Attribute	WP	WP	WP	WP	WP	WP	WP	WP
Initial Value	0	0	0	0	0	0	0	0

[bit31] to [bit0] NMIRm: NMIm Software Interrupt Reset Bits

These bits reset a software interrupt for the corresponding NMIm.

Bit	Description
0	Invalid (no effect on operation)
1	Reset a software interrupt for the corresponding NMIm.

5.12. IRC NMI Software Interrupt Status Register (IRCn_NMISIS)

This register indicates the software interrupt status of individual NMI channels. For each channel, 1 bit is assigned, and the bit location corresponds to the NMI channel number, m (0 to 31).

Bit	31	30	29	28	27	26	25	24
Field	NMISIS31	NMISIS30	NMISIS29	NMISIS28	NMISIS27	NMISIS26	NMISIS25	NMISIS24
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	NMISIS23	NMISIS22	NMISIS21	NMISIS20	NMISIS19	NMISIS18	NMISIS17	NMISIS16
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	NMISIS15	NMISIS14	NMISIS13	NMISIS12	NMISIS11	NMISIS10	NMISIS9	NMISIS8
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	NMISIS7	NMISIS6	NMISIS5	NMISIS4	NMISIS3	NMISIS2	NMISIS1	NMISIS0
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

[bit31] to [bit0] NMISISm: NMIm Software Interrupt Status Bits

These bits indicate the software interrupt status of the corresponding NMIm.

Bit	Description
0	Software interrupt is not set.
1	Software interrupt is set.

5.13. IRC IRQ Software Interrupt Set Register (IRCn_IRQS0 to IRCn_IRQS15)

These registers set a software interrupt for individual IRQ channels. For each channel, 1 bit is assigned. The 16 registers are all of the same type, and each is identified by the number at the end of the abbreviated register name. For each register, 32 channels are assigned, so that 512 channels are supported by the 16 registers.

Bit	31	30	29	28	27	26	25	24
Field	IRQS31	IRQS30	IRQS29	IRQS28	IRQS27	IRQS26	IRQS25	IRQS24
R/W Attribute	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
Protection Attribute	WPS	WPS	WPS	WPS	WPS	WPS	WPS	WPS
Initial Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	IRQS23	IRQS22	IRQS21	IRQS20	IRQS19	IRQS18	IRQS17	IRQS16
R/W Attribute	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
Protection Attribute	WPS	WPS	WPS	WPS	WPS	WPS	WPS	WPS
Initial Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	IRQS15	IRQS14	IRQS13	IRQS12	IRQS11	IRQS10	IRQS9	IRQS8
R/W Attribute	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
Protection Attribute	WPS	WPS	WPS	WPS	WPS	WPS	WPS	WPS
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	IRQS7	IRQS6	IRQS5	IRQS4	IRQS3	IRQS2	IRQS1	IRQS0
R/W Attribute	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
Protection Attribute	WPS	WPS	WPS	WPS	WPS	WPS	WPS	WPS
Initial Value	0	0	0	0	0	0	0	0

[bit31] to [bit0] IRQSr: IRQr Software Interrupt Set Bits

These bits set a software interrupt for the corresponding IRQr.

Bit	Description
0	Invalid (no effect on operation)
1	Set a software interrupt for the corresponding IRQr.

Notes:

- *The register bit configuration described in this section is for IRCn_IRQS0, which is representative of those registers that are all the same. The bit configurations of registers IRCn_IRQS0 to IRCn_IRQS15 are all the same.*
- *IRQ0 to IRQ31 are assigned to IRCn_IRQS0, and the subsequent channels are assigned in the same way until IRQ480 to IRQ511 are assigned to IRCn_IRQS15.*
- *The "r" in bit field name IRQSr corresponds to the IRQ channel number, r (0 to 511).*

5.14. IRC IRQ Software Interrupt Reset Register (IRCn_IRQR0 to IRCn_IRQR15)

These registers reset a software interrupt for individual IRQ channels. For each channel, 1 bit is assigned. The 16 registers are all of the same type, and each is identified by the number at the end of the abbreviated register name. For each register, 32 channels are assigned, so that 512 channels are supported by the 16 registers.

Bit	31	30	29	28	27	26	25	24
Field	IRQR31	IRQR30	IRQR29	IRQR28	IRQR27	IRQR26	IRQR25	IRQR24
R/W Attribute	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
Protection Attribute	WP	WP	WP	WP	WP	WP	WP	WP
Initial Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	IRQR23	IRQR22	IRQR21	IRQR20	IRQR19	IRQR18	IRQR17	IRQR16
R/W Attribute	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
Protection Attribute	WP	WP	WP	WP	WP	WP	WP	WP
Initial Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	IRQR15	IRQR14	IRQR13	IRQR12	IRQR11	IRQR10	IRQR9	IRQR8
R/W Attribute	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
Protection Attribute	WP	WP	WP	WP	WP	WP	WP	WP
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	IRQR7	IRQR6	IRQR5	IRQR4	IRQR3	IRQR2	IRQR1	IRQR0
R/W Attribute	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
Protection Attribute	WP	WP	WP	WP	WP	WP	WP	WP
Initial Value	0	0	0	0	0	0	0	0

[bit31] to [bit0] IRQRr: IRQr Software Interrupt Reset Bits

These bits reset a software interrupt for the corresponding IRQr.

Bit	Description
0	Invalid (no effect on operation)
1	Reset a software interrupt for the corresponding IRQr.

Notes:

- *The register bit configuration described in this section is for IRCn_IRQR0, which is representative of those registers that are all the same. The bit configurations of registers IRCn_IRQR0 to IRCn_IRQR15 are all the same.*
- *IRQ0 to IRQ31 are assigned to IRCn_IRQR0, and the subsequent channels are assigned in the same way until IRQ480 to IRQ511 are assigned to IRCn_IRQR15.*
- *The "r" in bit field name IRQRr corresponds to the IRQ channel number, r (0 to 511).*

5.15. IRC IRQ Software Interrupt Status Register (IRCn_IRQSIS0 to IRCn_IRQSIS15)

These registers indicate the software interrupt status of the individual IRQ channels. For each channel, 1 bit is assigned. The 16 registers are all of the same type, and each is identified by the number at the end of the abbreviated register name. For each register, 32 channels are assigned, so that 512 channels are supported by the 16 registers.

Bit	31	30	29	28	27	26	25	24
Field	IRQSIS31	IRQSIS30	IRQSIS29	IRQSIS28	IRQSIS27	IRQSIS26	IRQSIS25	IRQSIS24
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	IRQSIS23	IRQSIS22	IRQSIS21	IRQSIS20	IRQSIS19	IRQSIS18	IRQSIS17	IRQSIS16
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	IRQSIS15	IRQSIS14	IRQSIS13	IRQSIS12	IRQSIS11	IRQSIS10	IRQSIS9	IRQSIS8
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	IRQSIS7	IRQSIS6	IRQSIS5	IRQSIS4	IRQSIS3	IRQSIS2	IRQSIS1	IRQSIS0
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

[bit31] to [bit0] IRQSISr: IRQr Software Interrupt Status Bits

These bits indicate the software interrupt status of the corresponding IRQr.

Bit	Description
0	Software interrupt is not set.
1	Software interrupt is set.

Notes:

- *The register bit configuration described in this section is for IRCn_IRQSIS0, which is representative of those registers that are all the same. The bit configurations of registers IRCn_IRQSIS0 to IRCn_IRQSIS15 are all the same.*
- *IRQ0 to IRQ31 are assigned to IRCn_IRQSIS0, and the subsequent channels are assigned in the same way until IRQ480 to IRQ511 are assigned to IRCn_IRQSIS15.*
- *The "r" in bit field name IRQSISr corresponds to the IRQ channel number, r (0 to 511).*

5.16. IRC IRQ Channel Enable Set Register (IRCn_IRQCES0 to IRCn_IRQCES15)

These registers set enable for individual IRQ channels. For each channel, 1 bit is assigned. The 16 registers are all of the same type, and each is identified by the number at the end of the abbreviated register name. For each register, 32 channels are assigned, so that 512 channels are supported by the 16 registers.

Bit	31	30	29	28	27	26	25	24
Field	IRQCES31	IRQCES30	IRQCES29	IRQCES28	IRQCES27	IRQCES26	IRQCES25	IRQCES24
R/W Attribute	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
Protection Attribute	WPS	WPS	WPS	WPS	WPS	WPS	WPS	WPS
Initial Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	IRQCES23	IRQCES22	IRQCES21	IRQCES20	IRQCES19	IRQCES18	IRQCES17	IRQCES16
R/W Attribute	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
Protection Attribute	WPS	WPS	WPS	WPS	WPS	WPS	WPS	WPS
Initial Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	IRQCES15	IRQCES14	IRQCES13	IRQCES12	IRQCES11	IRQCES10	IRQCES9	IRQCES8
R/W Attribute	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
Protection Attribute	WPS	WPS	WPS	WPS	WPS	WPS	WPS	WPS
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	IRQCES7	IRQCES6	IRQCES5	IRQCES4	IRQCES3	IRQCES2	IRQCES1	IRQCES0
R/W Attribute	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
Protection Attribute	WPS	WPS	WPS	WPS	WPS	WPS	WPS	WPS
Initial Value	0	0	0	0	0	0	0	0

[bit31] to [bit0] IRQCESr: IRQr Channel Enable Set Bits

These bits set channel enable for the corresponding IRQr.

Bit	Description
0	Invalid (no effect on operation)
1	Set channel enable for IRQr.

Notes:

- *The register bit configuration described in this section is for IRCn_IRQCES0, which is representative of those registers that are all the same. The bit configurations of registers IRCn_IRQCES0 to IRCn_IRQCES15 are all the same.*
- *IRQ0 to IRQ31 are assigned to IRCn_IRQCES0, and the subsequent channels are assigned in the same way until IRQ480 to IRQ511 are assigned to IRCn_IRQCES15.*
- *The "r" in bit field name IRQCESr corresponds to the IRQ channel number, r (0 to 511).*

5.17. IRC IRQ Channel Enable Clear Register (IRCn_IRQCEC0 to IRCn_IRQCEC15)

These registers clear enable for individual IRQ channels. For each channel, 1 bit is assigned. The 16 registers are all of the same type, and each is identified by the number at the end of the abbreviated register name. For each register, 32 channels are assigned, so that 512 channels are supported by the 16 registers.

Bit	31	30	29	28	27	26	25	24
Field	IRQCEC31	IRQCEC30	IRQCEC29	IRQCEC28	IRQCEC27	IRQCEC26	IRQCEC25	IRQCEC24
R/W Attribute	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
Protection Attribute	WPS	WPS	WPS	WPS	WPS	WPS	WPS	WPS
Initial Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	IRQCEC23	IRQCEC22	IRQCEC21	IRQCEC20	IRQCEC19	IRQCEC18	IRQCEC17	IRQCEC16
R/W Attribute	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
Protection Attribute	WPS	WPS	WPS	WPS	WPS	WPS	WPS	WPS
Initial Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	IRQCEC15	IRQCEC14	IRQCEC13	IRQCEC12	IRQCEC11	IRQCEC10	IRQCEC9	IRQCEC8
R/W Attribute	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
Protection Attribute	WPS	WPS	WPS	WPS	WPS	WPS	WPS	WPS
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	IRQCEC7	IRQCEC6	IRQCEC5	IRQCEC4	IRQCEC3	IRQCEC2	IRQCEC1	IRQCEC0
R/W Attribute	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
Protection Attribute	WPS	WPS	WPS	WPS	WPS	WPS	WPS	WPS
Initial Value	0	0	0	0	0	0	0	0

[bit31] to [bit0] IRQCECr: IRQr Channel Enable Clear Bits

These bits clear channel enable for the corresponding IRQr.

Bit	Description
0	Invalid (no effect on operation)
1	Clear channel enable for IRQr.

Notes:

- *The register bit configuration described in this section is for IRCn_IRQCEC0, which is representative of those registers that are all the same. The bit configurations of registers IRCn_IRQCEC0 to IRCn_IRQCEC15 are all the same.*
- *IRQ0 to IRQ31 are assigned to IRCn_IRQCEC0, and the subsequent channels are assigned in the same way until IRQ480 to IRQ511 are assigned to IRCn_IRQCEC15.*
- *The "r" in bit field name IRQCECr corresponds to the IRQ channel number, r (0 to 511).*

5.18. IRC IRQ Channel Enable Setting Register (IRCn_IRQCE0 to IRCn_IRQCE15)

These registers set enable of individual IRQ channels. For each channel, 1 bit is assigned. The 16 registers are all of the same type, and each is identified by the number at the end of the abbreviated register name. For each register, 32 channels are assigned, so that 512 channels are supported by the 16 registers.

Bit	31	30	29	28	27	26	25	24
Field	IRQCE31	IRQCE30	IRQCE29	IRQCE28	IRQCE27	IRQCE26	IRQCE25	IRQCE24
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	WPS	WPS	WPS	WPS	WPS	WPS	WPS	WPS
Initial Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	IRQCE23	IRQCE22	IRQCE21	IRQCE20	IRQCE19	IRQCE18	IRQCE17	IRQCE16
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	WPS	WPS	WPS	WPS	WPS	WPS	WPS	WPS
Initial Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	IRQCE15	IRQCE14	IRQCE13	IRQCE12	IRQCE11	IRQCE10	IRQCE9	IRQCE8
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	WPS	WPS	WPS	WPS	WPS	WPS	WPS	WPS
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	IRQCE7	IRQCE6	IRQCE5	IRQCE4	IRQCE3	IRQCE2	IRQCE1	IRQCE0
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	WPS	WPS	WPS	WPS	WPS	WPS	WPS	WPS
Initial Value	0	0	0	0	0	0	0	0

[bit31] to [bit0] IRQCEr: IRQr Channel Enable Setting Bits

These bits set channel enable for the corresponding IRQr.

Bit	Description
0	Clear channel enable for IRQr.
1	Set channel enable for IRQr.

Notes:

- *The register bit configuration described in this section is for IRCn_IRQCE0, which is representative of those registers that are all the same. The bit configurations of registers IRCn_IRQCE0 to IRCn_IRQCE15 are all the same.*
- *IRQ0 to IRQ31 are assigned to IRCn_IRQCE0, and the subsequent channels are assigned in the same way until IRQ480 to IRQ511 are assigned to IRCn_IRQCE15.*
- *The "r" in bit field name IRQCEr corresponds to the IRQ channel number, r (0 to 511).*

5.19. IRC NMI Hold Clear Register (IRCn_NMIHC)

This register clears the hold bit for the ISR for the current NMI. The NMI channel number for which the hold bit is to be cleared is written.

Bit	31	8
Field	Reserved	
R/W Attribute	R0,WX	
Protection Attribute	-	
Initial Value	00000000_00000000_00000000	

Bit	7	6	5	4	3	2	1	0
Field	Reserved			NMIHCN				
R/W Attribute	R0,WX			R0,W				
Protection Attribute	-			WP				
Initial Value	000			00000				

[bit31:5] Reserved: Reserved Bits

[bit4:0] NMIHCN: Hold Clear NMI Channel Number Bits

These bits set the channel number corresponding to the ISR for the NMI.

5.20. IRC NMI Hold Status Register (IRCn_NMIHS)

This register indicates the hold status of the individual NMI channels. For each channel, 1 bit is assigned, and the bit location corresponds to the NMI channel number, m (0 to 31).

Bit	31	30	29	28	27	26	25	24
Field	NMIHS31	NMIHS30	NMIHS29	NMIHS28	NMIHS27	NMIHS26	NMIHS25	NMIHS24
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	NMIHS23	NMIHS22	NMIHS21	NMIHS20	NMIHS19	NMIHS18	NMIHS17	NMIHS16
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	NMIHS15	NMIHS14	NMIHS13	NMIHS12	NMIHS11	NMIHS10	NMIHS9	NMIHS8
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	NMIHS7	NMIHS6	NMIHS5	NMIHS4	NMIHS3	NMIHS2	NMIHS1	NMIHS0
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

[bit31] to [bit0] NMIHSm: NMIm Hold Status Bits

These bits indicate the hold status for NMIm.

Bit	Description
0	The interrupt for NMIm is not applied to the CPU.
1	The interrupt for NMIm is being applied to the CPU.

5.21. IRC IRQ Hold Clear Register (IRCn_IRQHC)

This register clears the hold bit for the ISR for the current IRQ. The IRQ channel number for which the hold bit is to be cleared is written.

Bit	31	16
Field	Reserved	
R/W Attribute	R0,WX	
Protection Attribute	-	
Initial Value	00000000_00000000	

Bit	15	14	13	12	11	10	9	8
Field	Reserved							IRQHCN[8]
R/W Attribute	R0,WX							R0,W
Protection Attribute	-							WP
Initial Value	0000000							0

Bit	7	6	5	4	3	2	1	0
Field	IRQHCN[7:0]							
R/W Attribute	R0,W							
Protection Attribute	WP							
Initial Value	00000000							

[bit31:9] Reserved: Reserved Bits

[bit8:0] IRQHCN: Bits for IRQ Channel Number for which Hold is to be Cleared.

These bits set the channel number corresponding to the ISR for the IRQ.

Notes:

- *This register is forbidden to be cleared by byte access write. Access size must be larger than half size.*
  Bus error will occur if write operation in byte size is done.
- *In case of writing to this register, please perform it in any of the following states (= IRC is stopping).*
  - Until writing the target register from reading that nIRQ bit of IRCn_IRQST register is "1". Please refer to the description of nIRQ bit for the target register.
  - The state that IRQEN bit of IRCn_CSR register is "0" (immediately after reset, etc).

5.22. IRC IRQ Hold Status Register (IRCn_IRQHS0 to IRCn_IRQHS15)

These registers indicate the hold status of individual IRQ channels. For each channel, 1 bit is assigned. The 16 registers are all of the same type, and each is identified by the number at the end of the abbreviated register name. For each register, 32 channels are assigned, so that 512 channels can be supported by the 16 registers.

Bit	31	30	29	28	27	26	25	24
Field	IRQHS31	IRQHS30	IRQHS29	IRQHS28	IRQHS27	IRQHS26	IRQHS25	IRQHS24
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	IRQHS23	IRQHS22	IRQHS21	IRQHS20	IRQHS19	IRQHS18	IRQHS17	IRQHS16
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	IRQHS15	IRQHS14	IRQHS13	IRQHS12	IRQHS11	IRQHS10	IRQHS9	IRQHS8
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	IRQHS7	IRQHS6	IRQHS5	IRQHS4	IRQHS3	IRQHS2	IRQHS1	IRQHS0
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

[bit31] to [bit0] IRQHSr: IRQr Hold Status Bits

These bits indicate the hold status of IRQr.

Bit	Description
0	The interrupt for IRQr is not applied to the CPU.
1	The interrupt for IRQr is being applied to the CPU.

Notes:

- *The register bit configuration described in this section is for IRCn_IRQHS0, which is representative of those registers that are all the same. The bit configurations of registers IRCn_IRQHS0 to IRCn_IRQHS15 are all the same.*
- *IRQ0 to IRQ31 are assigned to IRCn_IRQHS0, and the subsequent channels are assigned in the same way until IRQ480 to IRQ511 are assigned to IRCn_IRQHS15.*
- *The "r" in bit field name IRQHSr corresponds to the IRQ channel number, r (0 to 511).*

5.23. IRC IRQ Priority Level Mask Register (IRCn_IRQPLM)

This register sets the interrupt mask based on the priority level. The IRQ channel whose priority level is equal to or higher than the value set in this register is masked. The initial value for this register is 32. That is, no priority levels are masked in the initial state.

Bit	31 ... 8
Field	Reserved
R/W Attribute	R0,WX
Protection Attribute	-
Initial Value	00000000_00000000_00000000

Bit	7	6	5	4	3	2	1	0
Field	Reserved		IRQPLM					
R/W Attribute	R0,WX		R/W					
Protection Attribute	-		WPS					
Initial Value	00		100000					

[bit31:6] Reserved: Reserved Bits

[bit5:0] IRQPLM: IRQ Priority Level Mask Bits

These bits set the priority level mask value.

Note:

- *In case of writing to this register, please perform it in any of the following states (= IRC is stopping).*
  - Until writing the target register from reading that nIRQ bit of IRCn_IRQST register is "1". Please refer to the description of nIRQ bit for the target register.
  - The state that IRQEN bit of IRCn_CSR register is "0" (immediately after reset, etc).

5.24. IRC Control/Status Register (IRCn_CSR)

This register sets enable/disable of the IRQ processing block of the interrupt controller. It also indicates the lock status of the interrupt controller.

Bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	-							
Initial Value	00000000							

Bit	23	22	21	20	19	18	17	16
Field	Reserved							LST
R/W Attribute	R0,WX							R,WX
Protection Attribute	-							-
Initial Value	0000000							1

Bit	15	14	13	12	11	10	9	8
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	-							
Initial Value	00000000							

Bit	7	6	5	4	3	2	1	0
Field	Reserved							IRQEN
R/W Attribute	R0,WX							R/W
Protection Attribute	-							WPS
Initial Value	0000000							0

[bit31:17] Reserved: Reserved Bits

[bit16] LST: Interrupt Controller Lock Status

This bit indicates the lock status of the interrupt controller.

Bit	Description
0	Unlocked state
1	Locked state

[bit15:1] Reserved: Reserved Bits

[bit0] IRQEN: IRQ Processing Block Enable/Disable Setting Bit

This bit sets enable/disable of the IRQ processing block of the interrupt controller. If the IRQ processing block is disabled, the interrupt controller does not accept a new IRQ. The accepted interrupt is provided to the CPU.

Bit	Description
0	IRQ processing block is disabled.
1	IRQ processing block is enabled.

Note:

- *In case of writing to this register, please perform it in any of the following states (= IRC is stopping).*
  - Until writing the target register from reading that nIRQ bit of IRCn_IRQST register is "1". Please refer to the description of nIRQ bit for the target register.
  - The state that IRQEN bit of IRCn_CSR register is "0" (immediately after reset, etc).

5.25. IRC NMI RAW Status Register (IRCn_NMIRS)

This register indicates the RAW status of individual NMI channels. For each channel, 1 bit is assigned, and the bit location corresponds to the NMI channel number, m (0 to 31).

Bit	31	30	29	28	27	26	25	24
Field	NMIRS31	NMIRS30	NMIRS29	NMIRS28	NMIRS27	NMIRS26	NMIRS25	NMIRS24
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	NMIRS23	NMIRS22	NMIRS21	NMIRS20	NMIRS19	NMIRS18	NMIRS17	NMIRS16
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	NMIRS15	NMIRS14	NMIRS13	NMIRS12	NMIRS11	NMIRS10	NMIRS9	NMIRS8
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	NMIRS7	NMIRS6	NMIRS5	NMIRS4	NMIRS3	NMIRS2	NMIRS1	NMIRS0
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

[bit31] to [bit0] NMIRSm: RAW Status Bits for NMIm

These bits indicate the RAW status of NMIm.

Bit	Description
0	Interrupt is not set.
1	Interrupt is set.

5.26. IRC NMI Preprocessed Status Register (IRCn_NMIPS)

This register indicates the preprocessed status of individual NMI channels. For each channel, 1 bit is assigned, and the bit location corresponds to the NMI channel number, m (0 to 31).

Bit	31	30	29	28	27	26	25	24
Field	NMIPS31	NMIPS30	NMIPS29	NMIPS28	NMIPS27	NMIPS26	NMIPS25	NMIPS24
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	NMIPS23	NMIPS22	NMIPS21	NMIPS20	NMIPS19	NMIPS18	NMIPS17	NMIPS16
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	NMIPS15	NMIPS14	NMIPS13	NMIPS12	NMIPS11	NMIPS10	NMIPS9	NMIPS8
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	NMIPS7	NMIPS6	NMIPS5	NMIPS4	NMIPS3	NMIPS2	NMIPS1	NMIPS0
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

[bit31] to [bit0] NMIPSm: Preprocessed Status Bits for NMIm

These bits indicate the preprocessed status of NMIm.

Bit	Description
0	Interrupt is not set.
1	Interrupt is set.

5.27. IRC IRQ RAW Status Register (IRCn_IRQRS0 to IRCn_IRQRS15)

These registers indicate the RAW status of individual IRQ channels. For each channel, 1 bit is assigned. The 16 registers are all of the same type, and each is identified by the number at the end of the abbreviated register name. For each register, 32 channels are assigned, so that 512 channels are supported by the 16 registers.

Bit	31	30	29	28	27	26	25	24
Field	IRQRS31	IRQRS30	IRQRS29	IRQRS28	IRQRS27	IRQRS26	IRQRS25	IRQRS24
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	IRQRS23	IRQRS22	IRQRS21	IRQRS20	IRQRS19	IRQRS18	IRQRS17	IRQRS16
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	IRQRS15	IRQRS14	IRQRS13	IRQRS12	IRQRS11	IRQRS10	IRQRS9	IRQRS8
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	IRQRS7	IRQRS6	IRQRS5	IRQRS4	IRQRS3	IRQRS2	IRQRS1	IRQRS0
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

[bit31] to [bit0] IRQRSr: IRQr RAW Status Bits

These bits indicate the RAW status of IRQr.

Bit	Description
0	Interrupt is not set.
1	Interrupt is set.

Notes:

- *The register bit configuration described in this section is for IRCn_IRQRS0, which is representative of those registers that are all the same. The bit configurations of registers IRCn_IRQRS0 to IRCn_IRQRS15 are all the same.*
- *IRQ0 to IRQ31 are assigned to IRCn_IRQRS0, and the subsequent channels are assigned in the same way until IRQ480 to IRQ511 are assigned to IRCn_IRQRS15.*
- *The "r" in bit field name IRQRSr corresponds to the IRQ channel number, r (0 to 511).*

5.28. IRC IRQ Preprocessed Status Register (IRCn_IRQPS0 to IRCn_IRQPS15)

These registers indicate the preprocessed status of individual IRQ channels. For each channel, 1 bit is assigned. The 16 registers are all of the same type, and each is identified by the number at the end of the abbreviated register name. For each register, 32 channels are assigned, so that 512 channels are supported by the 16 registers.

Bit	31	30	29	28	27	26	25	24
Field	IRQPS31	IRQPS30	IRQPS29	IRQPS28	IRQPS27	IRQPS26	IRQPS25	IRQPS24
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	IRQPS23	IRQPS22	IRQPS21	IRQPS20	IRQPS19	IRQPS18	IRQPS17	IRQPS16
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	IRQPS15	IRQPS14	IRQPS13	IRQPS12	IRQPS11	IRQPS10	IRQPS9	IRQPS8
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	IRQPS7	IRQPS6	IRQPS5	IRQPS4	IRQPS3	IRQPS2	IRQPS1	IRQPS0
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

[bit31] to [bit0] IRQPSr: IRQr Preprocessed Status Bits

These bits indicate the preprocessed status of IRQr.

Bit	Description
0	Interrupt is not set.
1	Interrupt is set.

Notes:

- *The register bit configuration described in this section is for IRCn_IRQPS0, which is representative of those registers that are all the same. The bit configurations of registers IRCn_IRQPS0 to IRCn_IRQPS15 are all the same.*
- *IRQ0 to IRQ31 are assigned to IRCn_IRQPS0, and the subsequent channels are assigned in the same way until IRQ480 to IRQ511 are assigned to IRCn_IRQPS15.*
- *The "r" in bit field name IRQPSr corresponds to the IRQ channel number, r (0 to 511).*

5.29. IRC Unlock Register (IRCn_UNLOCK)

This register controls write lock for the interrupt controller to each register.

Bit	31	0
Field	UNLOCK	
R/W Attribute	R0,W	
Protection Attribute	WP	
Initial Value	00000000_00000000_00000000_00000000	

[bit31:0] UNLOCK: Interrupt Controller Unlock Bits

These bits control write lock for the interrupt controller to the setting registers.
The target registers for sequence protection by this register are those other than IRCn_NMIHC, IRCn_IRQHC, IRCn_NMIR, and IRCn_IRQR0 to IRCn_IRQR15.

UNLOCK	Description
0x17ACC911	Unlock value (which enables writing)
0x17B10C11	Lock value (which disables writing)
Other than above	Setting prohibited (bus error returned)

Note:

- The values of all 32 bits of this register are needed to determine the lock status. Therefore, writing to this register must be done with 32-bit access. Neither 8-bit nor 16-bit access is allowed.

5.30. IRC ECC Error Interrupt Register (IRCn_EEI)

This register indicates the ECC error interrupt status. It also clears the interrupt. There are 2 types of ECC error interrupt: NMI and IRQ.

Bit	31	30	29	28	27	26	25	24
Field	Reserved							EEIS
R/W Attribute	R0,WX							R,WX
Protection Attribute	-							-
Initial Value	0000000							0

Bit	23	22	21	20	19	18	17	16
Field	Reserved							EEIC
R/W Attribute	R0,WX							R0,W
Protection Attribute	-							WPS
Initial Value	0000000							0

Bit	15	14	13	12	11	10	9	8
Field	Reserved							EENS
R/W Attribute	R0,WX							R,WX
Protection Attribute	-							-
Initial Value	0000000							0

Bit	7	6	5	4	3	2	1	0
Field	Reserved							EENC
R/W Attribute	R0,WX							R0,W
Protection Attribute	-							WPS
Initial Value	0000000							0

[bit31:25] Reserved: Reserved Bits

[bit24] EEIS: ECC Error IRQ Status Bit

This bit indicates the status of the ECC error IRQ. When data is read from SRAM installed in the interrupt controller, this bit becomes "1" if a single-bit ECC error occurs.

Bit	Description
0	Single bit ECC error has not occurred.
1	Single bit ECC error has occurred.

[bit23:17] Reserved: Reserved Bits

[bit16] EEIC: ECC Error IRQ Clear Bit

This bit clears the ECC error IRQ.

Bit	Description
0	Invalid (no effect on operation)
1	Clear the ECC error IRQ (EEIS).

[bit15:9] Reserved: Reserved Bits

[bit8] EENS: ECC Error NMI Status Bit

This bit indicates the status of the ECC error NMI. When data is read from SRAM installed in the interrupt controller through the IRCn_IRQVAr register, this bit becomes "1" if a double-bit ECC error occurs.

Bit	Description
0	Double bit ECC error has not occurred.
1	Double bit ECC error has occurred.

Note:

- *This flag is not set in accessing the IRQ register in the IRQ processing stage.*

[bit7:1] Reserved: Reserved Bits

[bit0] EENC: ECC Error NMI Clear Bit

This bit clears the ECC error NMI.

Bit	Description
0	Invalid (no effect on operation)
1	Clear ECC error NMI (EENS).

Note:

- *EENS must be cleared only after the IRQ vector address is reinitialized.*

5.31. IRC ECC Address Number Register (IRCn_EAN)

When a single-bit or double-bit ECC error occurs, this register indicates the SRAM address of the error.

Bit	31	8
Field	Reserved	
R/W Attribute	R0,WX	
Protection Attribute	-	
Initial Value	00000000_00000000_00000000	

Bit	7	6	5	4	3	2	1	0
Field	EAN							
R/W Attribute	R,WX							
Protection Attribute	-							
Initial Value	00000000							

[bit31:8] Reserved: Reserved Bits

[bit7:0] EAN: ECC Error Occurrence Address Bits

These bits indicate the SRAM address where the ECC error is generated.

Notes:

- *If a single-bit error occurs after a double-bit error, SRAM address will not be updated while IRCn_EEI:EENS=1.*
- *If a single-bit or double-bit error occurs continuously, this register indicates the address of the last error.*

5.32. IRC ECC Test Register (IRCn_ET)

This register sets enable/disable of the test mode for the ECC protection function of SRAM installed in the interrupt controller.

Bit	31	8
Field	Reserved	
R/W Attribute	R0,WX	
Protection Attribute	-	
Initial Value	00000000_00000000_00000000	

Bit	7	6	5	4	3	2	1	0
Field	Reserved							ET
R/W Attribute	R0,WX							R/W
Protection Attribute	-							WPS
Initial Value	0000000							0

[bit31:1] Reserved: Reserved Bits

[bit0] ET: ECC Test Enable/Disable Setting Bit

This bit sets enable/disable of the ECC test mode.

Bit	Description
0	ECC test mode is disabled.
1	ECC test mode is enabled.

5.33. IRC ECC Error Bit Register (IRCn_EEB0 to IRCn_EEB1)

These registers are used in ECC test mode. In the data read from SRAM, you can spuriously corrupt any of the bits in the data area.

Bit	31 ... 8
Field	EEB[29:6]
R/W Attribute	R/W
Protection Attribute	WPS
Initial Value	00000000_00000000_00000000

Bit	7	6	5	4	3	2	1	0
Field	EEB[5:0]						Reserved	
R/W Attribute	R/W						R0,WX	
Protection Attribute	WPS						-	
Initial Value	000000						00	

[bit31:2] EEB: ECC Error Occurrence Bits

These bits are used to invert the data read from SRAM using the bitwise operation. If IRCn_ET:ET is "1", [29:0] of the data read from SRAM is XORed with EEB, and the result is handled as the read data. This means that the bits corresponding to those to which "1" is written within EEB are inverted. As a result, the SRAM data is spuriously corrupted and used for the test for the ECC protection function.

[bit1:0] Reserved: Reserved Bits

Note:

- *The register bit configuration described in this section is for IRCn_EEB0, which is representative of those registers that are all the same. The bit configuration of register IRCn_EEB1 is the same. IRCn_EEB1:EEB corresponds to the bits [66:37] of the data read from SRAM.*

5.34. IRC ECC Error Bit Register (IRCn_EEB2)

This register is used in ECC test mode. In the data read from SRAM, you can spuriously corrupt any of the bits in the ECC parity area.

Bit	31	16
Field	Reserved	
R/W Attribute	R0,WX	
Protection Attribute	-	
Initial Value	00000000_00000000	

Bit	15	14	13	12	11	10	9	8
Field	Reserved	EEBO						
R/W Attribute	R0,WX	R/W						
Protection Attribute	-	WPS						
Initial Value	0	0000000						

Bit	7	6	5	4	3	2	1	0
Field	Reserved	EEBE						
R/W Attribute	R0,WX	R/W						
Protection Attribute	-	WPS						
Initial Value	0	0000000						

[bit31:15] Reserved: Reserved Bits

[bit14:8] EEBO: ECC Error Occurrence Bits

These bits are used to invert the data read from SRAM using the bitwise operation. If IRCn_ET:ET is "1", [73:67] of the data read from SRAM is XORed with EEBO, and the result is handled as the read data. This means the bits corresponding to those to which "1" is written within EEBO are inverted. As a result, the SRAM data is spuriously corrupted and used for the test for the ECC protection function.

[bit7] Reserved: Reserved Bit

[bit6:0] EEBE: ECC Error Occurrence Bits

These bits are used to invert the data read from SRAM using the bitwise operation. If IRCn_ET:ET is "1", [36:30] of the data read from SRAM is XORed with EEBE, and the result is handled as the read data. This means that the bits corresponding to those to which "1" is written within EEBE are inverted. As a result, the SRAM data is spuriously corrupted and used for the test for the ECC protection function.

5.35. IRC NMI Vector Address Status Register (IRCn_NMIVASBR)

This register indicates the vector address status of the NMI that was applied to the CPU most recently.

Bit	31	0
Field	NMIVAS	
R/W Attribute	R,WX	
Protection Attribute	RP	
Initial Value	00000000_00000000_00000000_00000000	

[bit31:0] NMIVAS: NMI Vector Address Status Bits

These bits indicate the vector address status of the NMI that was applied to the CPU most recently.

5.36. IRC NMI Vector Address Status Mirror Register (IRC_NMIVASBR)

This register is a mirror register of IRCn_NMIVASBR, and is dedicated to BootROM. The CPU can read this register and branch to the ISR for the NMI.

Bit	31	0
Field	NMIVAS	
R/W Attribute	R,WX	
Protection Attribute	RP	
Initial Value	00000000_00000000_00000000_00000000	

[bit31:0] NMIVAS: NMI Vector Address Status Bits

These bits indicate the vector address status of the NMI that was applied to the CPU most recently.

Notes:

- *This register is dedicated to BootROM and is used as follows:*

1. An NMI occurs.
2. The execution code jumps to the corresponding entry in the fixed exception vector table of BootROM.
3. The load instruction at the exception entry point loads the value stored in IRC_NMIVASBR.
4. The code jumps to the vector address of the accepted NMI and executes the ISR for the NMI.

- *Each CPU reads the same address from the register where the stored address is mapped to that which is common to all CPUs.*
- *When a read operation is executed by CPUn, the register for IRCn is read. AHB masters other than CPUs cannot read this register.*

5.37. IRC ECC Error Vector Address Register (IRCn_IRQEEVA)

This register is used to perform correct error handling when SRAM holding the IRQ vector is read by the IRQ processing block and a 2-bit ECC error is detected.

Bit	31	8
Field	IRQVA[31:8]	
R/W Attribute	R/W	
Protection Attribute	WPS	
Initial Value	00000000_00000000_00000000	

Bit	7	6	5	4	3	2	1	0
Field	IRQVA[7:2]						IRQVA[1:0]	
R/W Attribute	R/W						R0,WX	
Protection Attribute	WPS						-	
Initial Value	000000						00	

[bit31:0] IRQVA: IRQ Vector Address Bits

These bits set the address of the error handler when SRAM holding the IRQ vector is read by the IRQ processing block and a 2-bit ECC error is detected.

Notes:

- *Since all vector addresses are 32 bits long, the lowest 2 bits are fixed to "00".*
- *If a 2-bit ECC error occurs in vector SRAM during IRQ processing, transition is made to this vector address, rather than to the interrupt vector of the IRQ that has occurred. A transition to this vector cannot be changed. Therefore, the vector address must be set to this register to enable correct error handling.*
- *Set a value in this register before enabling the IRQ processing block.*

6. Others

This section lists precautions to be observed when using the interrupt controller.

- Software interrupts must be cleared by ISR.
- If 2 interrupts of the same priority level are asserted at the same time, that with the smaller channel number takes precedence. If an interrupt occurs, followed by another of the same priority level, the latter interrupt will be masked until the service routine for the former is completed.
- All vector addresses should be set during the initialization process.
- For the interrupt controller to work properly, the peripheral interrupt must be cleared before the corresponding hold bit in the interrupt controller is cleared.
- In the case of a software interrupt, the software interrupt setting bit in the interrupt controller must be cleared before the corresponding hold bit in the interrupt controller is cleared.
- If a double-bit ECC error in SRAM is detected by reading IRCn_IRQVAr by CPU, a correct vector address must be written again in the relevant IRCn_IRQVAr register with the corresponding NMI handler. Then, the error bit must be cleared by writing to IRCn_EEI:EENC.
- If a double-bit ECC error in SRAM is detected by reading IRCn_IRQVAr by IRQ processing unit, correct vector address must be rewritten to IRCn_IRQVAr register corresponded to IRQEEVA handler.

The conditions under which a bus error is returned for register access in the interrupt controller are shown below:

- Sequence protection violation by IRCn_UNLOCK
- Writing values other than the lock value and unlock value to IRCn_UNLOCK
- Privilege protection violation
- Read or write attempt in which access is made only to an undefined register area
- Read or write attempt in which access is made only to reserved bits
- Write attempt in which access is made only to bits having the WX write attribute (including the undefined register area)
- When write by byte access is performed to IRCn_IRQHC register

CHAPTER 23: Time Protection

This chapter explains time protection.

1. **Overview**
2. **Configuration**
3. **Explanation of Operation**
4. **Setting Procedure Examples**
5. **Registers**
6. **Others**

TPU-T00PT03P01R01L04-E1-XX

1. Overview

This section provides an overview of time protection.

Modern operating systems (especially those with time-based triggers) need to manage various time-based operations of tasks. For example, tasks involve the following types of time:

- Overall maximum execution time
- Time limit from start to end
- Maximum tolerated period during which certain-level interrupts are prohibited
- Maximum tolerated period during which all interrupts are prohibited
- Minimum rate of restart

To satisfy the requirements, each CPU is equipped with a time protection unit to provide hardware-based support.

The time protection unit provides the following features:

- 8 timers of the same type are provided, each of which is denoted by the timer number "m". (m = 0 to 7)
  These timers are used for the execution time, the lock period, the arrival time interval, and the protection of the time limit.
- 24-bit up-counter, for which normal mode or overflow mode can be selected
- Generation of NMI interrupts
- Setting of end counts and preload values
- Preload function that loads the preload value in the overflow mode
- Free-run function that restarts the timer automatically
- Clock division enabled by a global prescaler (division ratio: 1/1 to 1/64)
  This function divides the system clock into the frequency required as the input to each timer-specific prescaler.
- Clock division by each timer-specific prescaler (division ratio: 1/1, 1/2, 1/4, or 1/16)
  Dividing the output clock from the global prescaler into the clock used by the timer
- Reading the current count value
- Control of start, stop, and restart of each timer
- Notification of each timer status (stopped or operating)
- Equipped with a slave interface of AHB 64 bits for register access
- Support of the debug mode for stalling timers

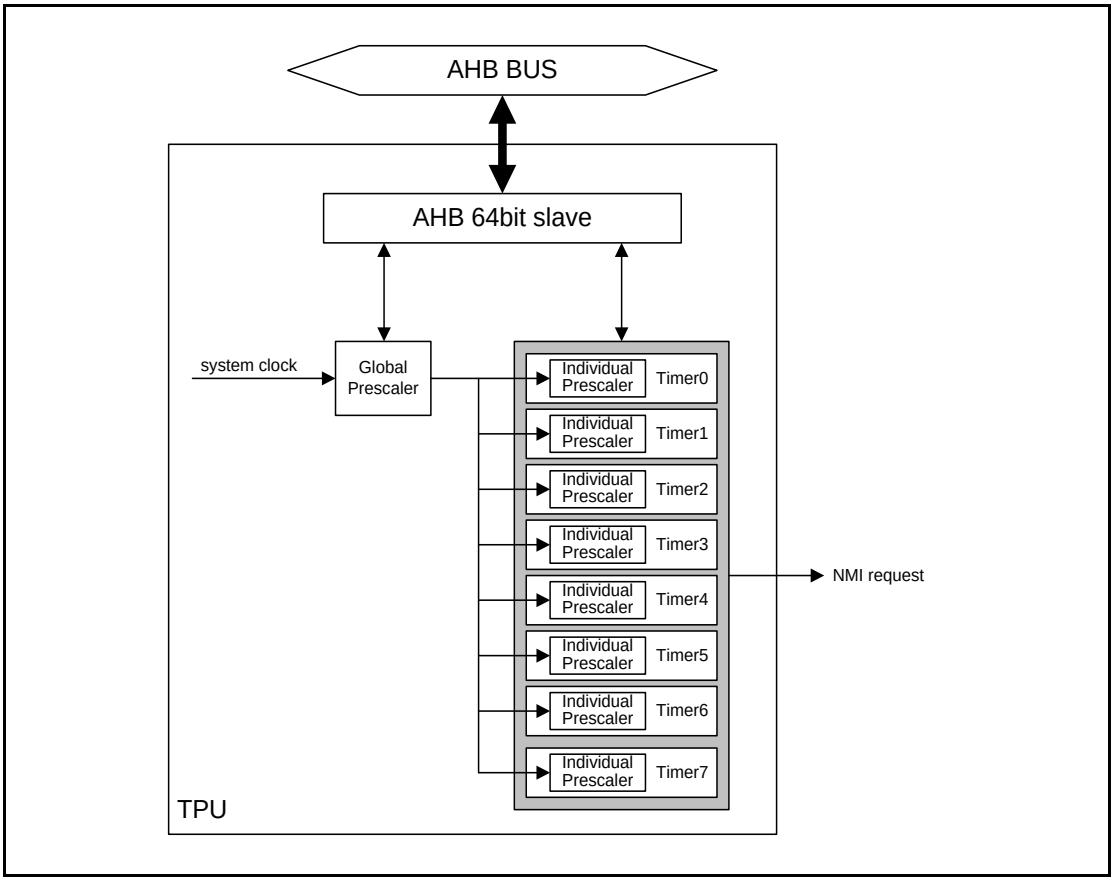
Notes:

- *This chapter sometimes refers to the time protection unit by its abbreviation TPU (Time Protection Unit).*
- *"m" in the abbreviated register name of "TPU0_**m" or "TPU0_***:**m" indicates the timer number.*
- *In this chapter, interrupt indicates the NMI interrupt.*

2. Configuration

This section explains a block diagram of the time protection unit.

Figure 2-1 Block Diagram of Time Protection Unit



3. Explanation of Operation

This section explains the operation of the time protection unit.

Sequence Protection by the Lock Release Register

The time protection unit provides the sequence protection feature using a lock release register (TPU0_UNLOCK). The lock must be released by writing the lock release value in TPU0_UNLOCK before the timer setting can be updated. Writing the locking value after making a timer setting sets the locked state again.

Timer Operation Mode

Normal Mode

The condition for generating an interrupt request in the normal mode is "the current timer count value >= end count value". The current timer count value being counted up is compared with the end-count/preload-value setting bits (TPU0_TCN0m:ECPL) in timer control register 0. If the current count value is equal to or larger than the end-count value, an interrupt request occurs and the corresponding bit (TPU0_TIR:IRm) in the timer interrupt request register becomes "1". If an interrupt request occurs and the corresponding bit (TPU0_TIE:IEm) in the timer interrupt enable register is enabled, an interrupt for CPU0 is generated.

To start the timer operation, write "1" in the operation start bit (TPU0_TCN0m:START) in timer control register 0. The timer is reset and the counting starts from 0x000000. During timer operation, the corresponding bit (TPU0_TST:STm) in the timer status register keeps the value "1".

The timer automatically stops when an interrupt request occurs.
To stop the timer operation arbitrarily, write "1" in the operation stop bit (TPU0_TCN0m:STOP) in timer control register 0. TPU0_TST:STm becomes "0" when the timer stops. The current timer count value of the stopped timer is stalled.

To restart the timer, write "1" in the operation restart bit (TPU0_TCN0m:CONT) in timer control register 0. The current timer count value that has been stalled at the time of timer stop is used as the starting point of the counting. TPU0_TST:STm keeps the value "1" during timer operation.

Overflow Mode

The condition for generating an interrupt request in the overflow mode is timer overflow. If a count up occurs when the current timer count value is 0xFFFFFF, an interrupt request occurs and TPU0_TIR:IRm becomes "1". If an interrupt request occurs and TPU0_TIE:IEm is enabled, an interrupt to CPU0 is generated.

Start, stop, and restart of the operation are the same as those in normal mode.

Debug Mode

All timers stop when "1" is written in the debug mode enable/disable bit (TPU0_CFG:DBGE) in the configuration register and the CPU enters the debug state. The current timer count values of the stopped timers are stalled. The timers start operation as soon as the above condition is no longer true.

For the definition of the debug state, see 12.8 of Cortex™-R5 Revision:r1p2 Technical Reference Manual (ARM DDI 0460D).

Preload Function

This function sets the counting start value in the overflow mode to any specified value, instead of setting it to the normal "0". This function is enabled by writing "1" in the preload function enable/disable setting bit (TPU0_TCN1m:PL) in timer control register 1. Any counting start value being set is called a preload value. The preload value is set by TPU0_TCN0m:ECPL.

This function cannot be used in the normal mode.

Free-Run Function

This function automatically restarts the timer, instead of stopping it, when an interrupt request occurs. This function is enabled by writing "1" in the free-run function enable/disable setting bit (TPU0_TCN1m:FRT) in timer control register 1.

This function can be used in both normal and overflow modes.

The count starting value at restart is the same as in normal operation: 0x000000 for the normal mode, and 0x000000 or the preload value in the overflow mode.

Because the timer restarts as soon as an interrupt request occurs, the condition TPU0_TST:STm = 1 continues. When the condition for interrupt request generation is satisfied, TPU0_TIR:IRm becomes "1". The value stays at "1" if the previous interrupt request is not cleared.

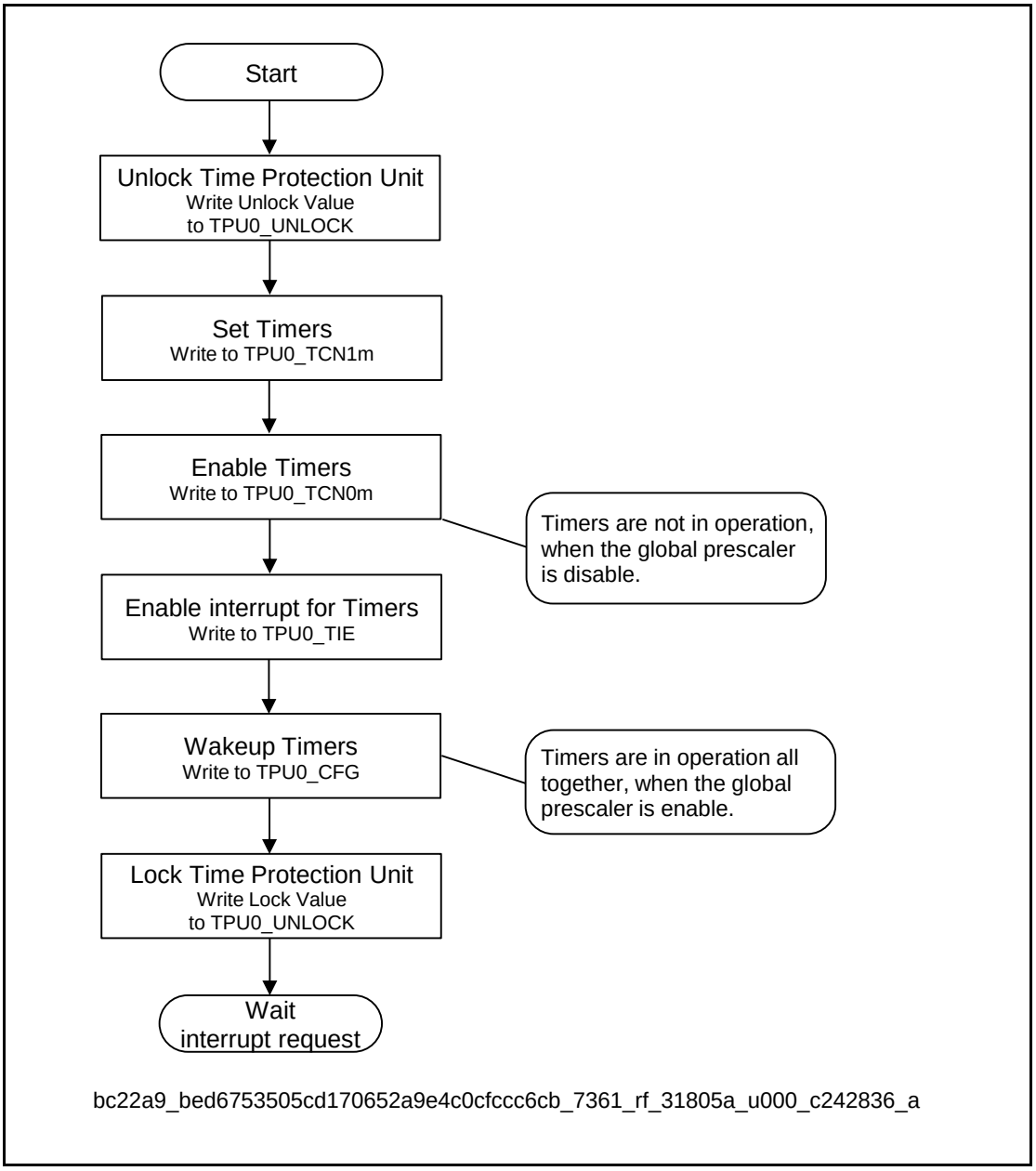
4. Setting Procedure Examples

This section explains provides setting procedure examples of the time protection unit.

When Multiple Timers are Simultaneously Started from the Initial State

When multiple timers are simultaneously started from the initial state, the enable setting of the global prescaler is used as the trigger. Configure TPU0_TCN1m and TPU0_TCN0m of the target timers to the operational status before enabling the global prescaler. At this time the timers are not still operational because the counters are not supplied with the clock. Enabling the global prescaler simultaneously starts the timers that have previously been configured with the operation start setting.

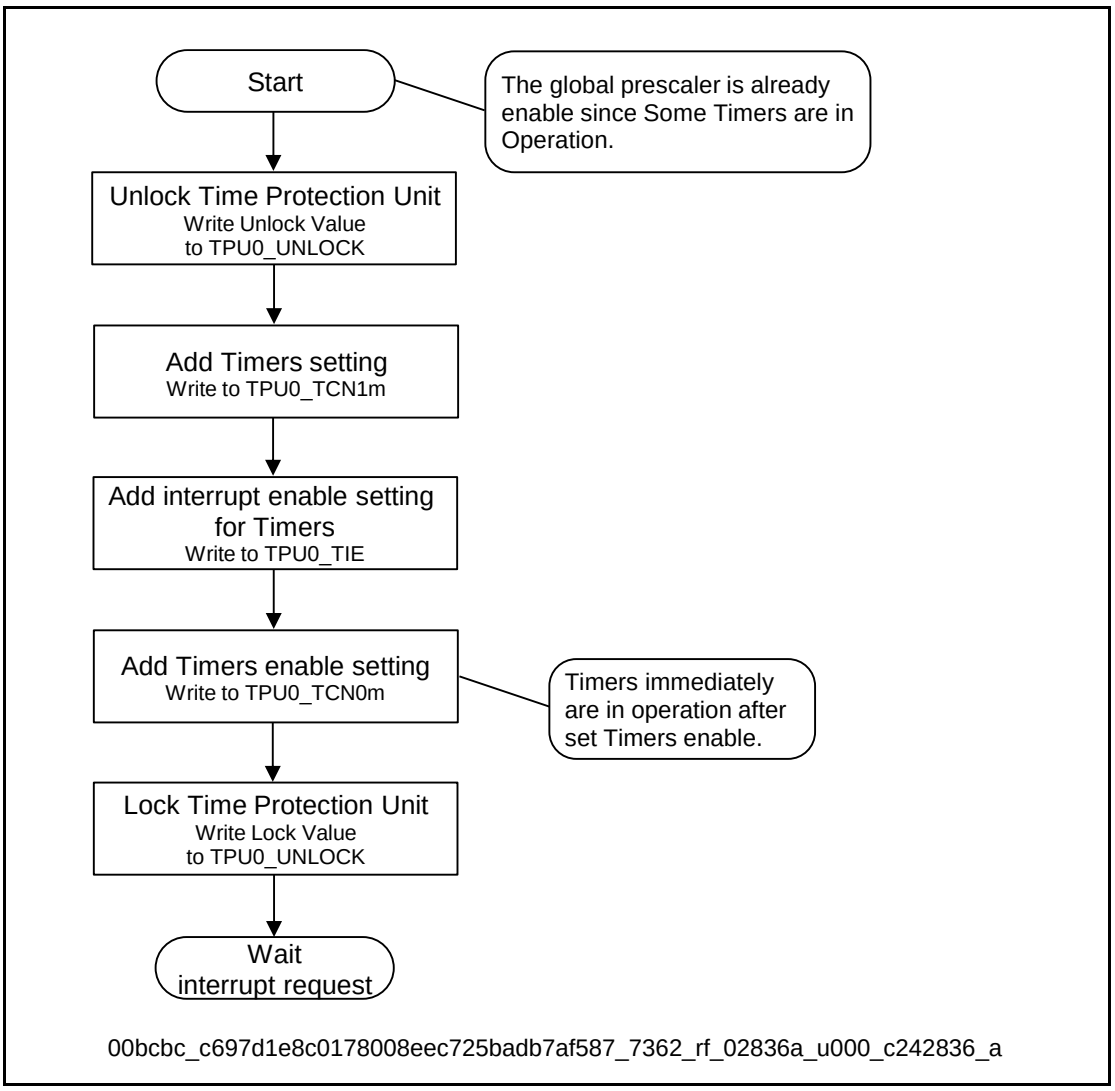
Figure 4-1 Time Protection Unit Setting Flow (When Multiple Timers are Simultaneously Started From the Initial State)



Starting an Additional Timer while Some Timers are in Operation

When starting an additional timer while some timers are in operation, the setting action for starting the additional timer directly triggers the start of the timer because the global prescaler is already in operation providing timer clocks to timers.

Figure 4-2 Time Protection Unit Setting Flow (Starting an Additional Timer While Some Timers are in Operation)



Note:

- *Because the timer is started when the global prescaler is already in operation, the state of the global prescaler varies depending on the timing of the setting action for the timer start. For this reason, the actual time required before generation of an interrupt may have a margin of error. For example, suppose that the global prescaler has been started with a division ratio of 1/12. Also, suppose that it takes a period of 2 system clock cycles between the setting action for the timer start and the detection of the first clock edge of the global clock. In this case, there will be a time difference of 10 system clock cycles. The maximum time difference depends on the division ratio of the global prescaler.*

5. Registers

This section explains the registers used by the time protection unit.

Table 5-1 Register List of Time Protection Unit

Abbreviated Register Name	Register Name	Reference
TPU0_UNLOCK	TPU lock release register	5.1
TPU0_LST	TPU lock status register	5.2
TPU0_CFG	TPU configuration register	5.3
TPU0_TIR	TPU timer interrupt request register	5.4
TPU0_TST	TPU timer status register	5.5
TPU0_TIE	TPU timer interrupt enable register	5.6
TPU0_TCN0m	TPU timer m control register 0	5.7
TPU0_TCN1m	TPU timer m control register 1	5.8
TPU0_TCCm	TPU timer m current count register	5.9

Table 5-2 Register Memory Layout of Time Protection Unit

Offset	+7	+6	+5	+4	+3	+2	+1	+0
0x00000000	TPU0_LST 00000000_00000000_00000000_00000001				TPU0_UNLOCK 00000000_00000000_00000000_00000000			
0x00000008	TPU0_TIR 00000000_00000000_00000000_00000000				TPU0_CFG 00000000_00000000_00000000_00000000			
0x00000010	TPU0_TIE 00000000_00000000_00000000_00000000				TPU0_TST 00000000_00000000_00000000_00000000			
0x00000018	Reserved				Reserved			
0x00000020 | 0x00000028	Reserved				Reserved			
0x00000030	TPU0_TCN01 00000000_00000000_00000000_00000000				TPU0_TCN00 00000000_00000000_00000000_00000000			
0x00000038	TPU0_TCN03 00000000_00000000_00000000_00000000				TPU0_TCN02 00000000_00000000_00000000_00000000			
0x00000040	TPU0_TCN05 00000000_00000000_00000000_00000000				TPU0_TCN04 00000000_00000000_00000000_00000000			
0x00000048	TPU0_TCN07 00000000_00000000_00000000_00000000				TPU0_TCN06 00000000_00000000_00000000_00000000			
0x00000050	TPU0_TCN11 00000000_00000000_00000000_00000000				TPU0_TCN10 00000000_00000000_00000000_00000000			
0x00000058	TPU0_TCN13 00000000_00000000_00000000_00000000				TPU0_TCN12 00000000_00000000_00000000_00000000			
0x00000060	TPU0_TCN15 00000000_00000000_00000000_00000000				TPU0_TCN14 00000000_00000000_00000000_00000000			
0x00000068	TPU0_TCN17 00000000_00000000_00000000_00000000				TPU0_TCN16 00000000_00000000_00000000_00000000			
0x00000070	TPU0_TCC1 00000000_00000000_00000000_00000000				TPU0_TCC0 00000000_00000000_00000000_00000000			
0x00000078	TPU0_TCC3 00000000_00000000_00000000_00000000				TPU0_TCC2 00000000_00000000_00000000_00000000			
0x00000080	TPU0_TCC5 00000000_00000000_00000000_00000000				TPU0_TCC4 00000000_00000000_00000000_00000000			
0x00000088	TPU0_TCC7 00000000_00000000_00000000_00000000				TPU0_TCC6 00000000_00000000_00000000_00000000			
0x00000090 | 0x000003F8	Reserved				Reserved			

5.1. TPU Lock Release Register (TPU0_UNLOCK)

This register controls the write lock of the registers of the time protection unit.

Bit	31	0
Field	UNLOCK	
R/W Attribute	R0,W	
Protection Attribute	WP	
Initial Value	00000000_00000000_00000000_00000000	

[bit31:0] UNLOCK: Lock Release Bits of the Time Protection Unit

These bits control the write lock of the configuration registers of the time protection unit.
TPU0_CFG and TPU0_TCN10 to TPU0_TCN17 are subject to sequence protection by this register.

UNLOCK	Description
0xACC5A110	Unlock value (which enables writing)
0xB10CACC5	Lock value (which disables writing)
Other than above	Setting prohibited (bus error returned)

Note:

- *The value of all 32 bits of this register is needed to determine the lock status. Therefore, writing to this register must be made with 32-bit or 64-bit access. 8-bit or 16-bit access is not allowed.*

5.2. TPU Lock Status Register (TPU0_LST)

This register indicates the lock status of the time protection unit.

Bit	31	8
Field	Reserved	
R/W Attribute	R0,WX	
Protection Attribute	-	
Initial Value	00000000_00000000_00000000	

Bit	7	6	5	4	3	2	1	0
Field	Reserved							LST
R/W Attribute	R0,WX							R,WX
Protection Attribute	-							
Initial Value	0000000							1

[bit31:1] Reserved: Reserved Bits

[bit0] LST: Lock Status of the Time Protection Unit

This bit indicates the lock status of the time protection unit.

Bit	Description
0	Unlocked state
1	Locked state

5.3. TPU Configuration Register (TPU0_CFG)

This register makes the setting of the global prescaler and enables/disables interrupts for the time protection unit. This register also makes the setting of enable/disable of the debug mode.

Bit	31	30	29	28	27	26	25	24
Field	Reserved							DBGE
R/W Attribute	R0,WX							R/W
Protection Attribute	WPS							
Initial Value	0000000							0

Bit	23	22	21	20	19	18	17	16
Field	GLBPSE	Reserved	GLBPS					
R/W Attribute	R/W	R0,WX	R/W					
Protection Attribute	WPS							
Initial Value	0	0	000000					

Bit	15	14	13	12	11	10	9	8
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	WPS							
Initial Value	00000000							

Bit	7	6	5	4	3	2	1	0
Field	Reserved							INTE
R/W Attribute	R0,WX							R/W
Protection Attribute	WPS							
Initial Value	0000000							0

[bit31:25] Reserved: Reserved Bits

[bit24] DBGE: Debug Mode Enable/Disable Setting Bit

This bit sets enable/disable of the debug mode.

Bit	Description
0	Disable debug mode.
1	Enable debug mode.

[bit23] GLBPSE: Global Prescaler Enable/Disable Setting Bit

This bit sets enable/disable of the global prescaler. If this bit is set to disable, the global prescaler counter stops after the next increment pulse. All timers also stop as a result.

Bit	Description
0	Disable global prescaler.
1	Enable global prescaler.

[bit22] Reserved: Reserved Bit

[bit21:16] GLBPS: Global Prescaler Division Setting Bit

This bit sets the division ratio of the global prescaler. The input system clock is divided according to the configured division ratio, generating a clock of a lower frequency. A divided clock is provided to each timer.

Bit						Description
0	0	0	0	0	0	1/1
0	0	0	0	0	1	1/2
0	0	0	0	1	0	1/3
-	-	-	-	-	-	- - -
1	1	1	1	1	1	1/64

[bit15:1] Reserved: Reserved Bits

[bit0] INTE: Time Protection Unit Interrupt Enable/Disable Setting Bit

This bit makes the enable/disable setting of interrupts of the time protection unit. The enable setting allows generation of interrupts to CPU0. The disable setting prohibits generation of interrupts to CPU0.

Bit	Description
0	Disable interrupts of the time protection unit.
1	Enable interrupts of the time protection unit.

Notes:

- *For details of the debug mode, see "3.Explanation of Operation."*
- *Interrupts to CPU0 is generated when TPU0_CFG:INTE= 1 and TPU0_TIR:IRm= 1 and TPU0_TIE:IEm= 1.*
- *It is prohibition that carries out timer starting by TPU0_CFG:GLBPS= 0b000000 and TPU0_TCN1m:PS= 0b00.*

5.4. TPU Timer Interrupt Request Register (TPU0_TIR)

This register indicates the interrupt request status of each timer. Each timer is assigned with 1 bit, and the bit position indicates the timer number "m".

Bit	31	8
Field	Reserved	
R/W Attribute	R0,WX	
Protection Attribute	-	
Initial Value	00000000_00000000_00000000	

Bit	7	6	5	4	3	2	1	0
Field	IR7	IR6	IR5	IR4	IR3	IR2	IR1	IR0
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

[bit31:8] Reserved: Reserved Bits

[bit7] IR7: Timer 7 Interrupt Request Bit
[bit6] IR6: Timer 6 Interrupt Request Bit
[bit5] IR5: Timer 5 Interrupt Request Bit
[bit4] IR4: Timer 4 Interrupt Request Bit
[bit3] IR3: Timer 3 Interrupt Request Bit
[bit2] IR2: Timer 2 Interrupt Request Bit
[bit1] IR1: Timer 1 Interrupt Request Bit
[bit0] IR0: Timer 0 Interrupt Request Bit

These bits indicate the interrupt request status of individual timers. Indication is made for each timer about whether the interrupt request flag is on or there is a retained interrupt request.

Bit	Description
0	An interrupt request does not exist for timer m.
1	An interrupt request exists for timer m.

Note:

- *An interrupt to CPU0 is generated when TPU0_CFG:INTE= 1 and TPU0_TIR:IRm= 1 and TPU0_TIE:IEm= 1.*

5.5. TPU Timer Status Register (TPU0_TST)

This register indicates the operation status of each timer. Each timer is assigned 1 bit, and the bit position indicates the timer number "m".

Bit	31	8
Field	Reserved	
R/W Attribute	R0,WX	
Protection Attribute	-	
Initial Value	00000000_00000000_00000000	

Bit	7	6	5	4	3	2	1	0
Field	ST7	ST6	ST5	ST4	ST3	ST2	ST1	ST0
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

[bit31:8] Reserved: Reserved Bits

[bit7] ST7: Timer 7 Status Bit
[bit6] ST6: Timer 6 Status Bit
[bit5] ST5: Timer 5 Status Bit
[bit4] ST4: Timer 4 Status Bit
[bit3] ST3: Timer 3 Status Bit
[bit2] ST2: Timer 2 Status Bit
[bit1] ST1: Timer 1 Status Bit
[bit0] ST0: Timer 0 Status Bit

These bits indicate the operation status of individual timers. Their indications show whether individual timers are operating or stopped.

Bit	Description
0	Timer m is stopped.
1	Timer m is operating.

5.6. TPU Timer Interrupt Enable Register (TPU0_TIE)

This register sets enable/disable of interrupt of individual timers. Each timer is assigned 1 bit, and the bit position indicates the timer number "m".

Bit	31	8
Field	Reserved	
R/W Attribute	R0,WX	
Protection Attribute	WP	
Initial Value	00000000_00000000_00000000	

Bit	7	6	5	4	3	2	1	0
Field	IE7	IE6	IE5	IE4	IE3	IE2	IE1	IE0
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	WP							
Initial Value	0	0	0	0	0	0	0	0

[bit31:8] Reserved: Reserved Bits

[bit7] IE7: Timer 7 Interrupt Enable/Disable Setting Bit
[bit6] IE6: Timer 6 Interrupt Enable/Disable Setting Bit
[bit5] IE5: Timer 5 Interrupt Enable/Disable Setting Bit
[bit4] IE4: Timer 4 Interrupt Enable/Disable Setting Bit
[bit3] IE3: Timer 3 Interrupt Enable/Disable Setting Bit
[bit2] IE2: Timer 2 Interrupt Enable/Disable Setting Bit
[bit1] IE1: Timer 1 Interrupt Enable/Disable Setting Bit
[bit0] IE0: Timer 0 Interrupt Enable/Disable Setting Bit

These bits set enable/disable of interrupts of individual timers. If these bits are enabled, interrupt requests of corresponding timers generate interrupts to CPU0. If these bits are disabled, interrupt requests of corresponding timers do not generate interrupts to CPU0.

Bit	Description
0	Disable interrupt of timer m.
1	Enable interrupt of timer m.

Note:

- *An interrupt to CPU0 is generated when TPU0_CFG:INTE= 1 and TPU0_TIR:IRm= 1 and TPU0_TIE:IEm= 1.*

5.7. TPU Timer m Control Register 0 (TPU0_TCN0m)

This register controls operation of individual timers. This register controls start, stop, and restart of timers, and controls timer interrupts. This register also sets the end counts and the preload values of timers. Registers of the same type are provided for individual timers, and "m" in the abbreviated register name indicates the timer number m (0 to 7).

Bit	31	30	29	28	27	26	25	24
Field	START	STOP	CONT	IES	IEC	IRC	Reserved	
R/W Attribute	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,WX	
Protection Attribute	WP							
Initial Value	0	0	0	0	0	0	00	

Bit	23	0
Field	ECPL	
R/W Attribute	R/W	
Protection Attribute	WP	
Initial Value	00000000_00000000_00000000	

[bit31] START: Timer Operation Start Bit

Writing "1" in this bit resets the timer and starts its operation.

Bit	Description
0	Invalid (no effect on operation)
1	Start timer operation.

[bit30] STOP: Timer Operation Stop Bit

Writing "1" in this bit stops the timer operation. It does not reset the timer.

Bit	Description
0	Invalid (no effect on operation)
1	Stop timer operation.

[bit29] CONT: Timer Operation Restart Bit

Writing "1" in this bit restarts the timer from the previously stalled state.

Bit	Description
0	Invalid (no effect on operation)
1	Restart timer operation.

[bit28] IES: Timer Interrupt Enable Set Bit

Writing "1" in this bit sets the corresponding bit in the timer interrupt enable register. "1" is written in TPU0_TIE:IEm.

Bit	Description
0	Invalid (no effect on operation)
1	Enable timer interrupt.

[bit27] IEC: Timer Interrupt Enable Clear Bit

Writing "1" in this bit clears the corresponding bit in the timer interrupt enable register. "0" is written in TPU0_TIE:IEm.

Bit	Description
0	Invalid (no effect on operation)
1	Disable timer interrupt.

[bit26] IRC: Timer Interrupt Request Clear Bit

Writing "1" in this bit clears the corresponding bit in the timer interrupt request enable register. TPU0_TIR:IRm is cleared to "0".

Bit	Description
0	Invalid (no effect on operation)
1	Clear timer interrupt flag.

[bit25:24] Reserved: Reserved Bits

[bit23:0] ECPL: End Count / Preload Value Setting Bit

The value of these bits is used as the end count in the normal operation mode, and as the preload value when the operation mode is overflow mode and the preload function is enabled.

Notes:

- *When multiple bits among START, STOP, and CONT are set to "1" simultaneously, the priority order is START > CONT > STOP.*
- *For details on timer operation modes, end counts, and preload values, see "3.Explanation of Operation."*
- *TPU0_TCN00 to TPU0_TCN07 share the same register bit configuration.*

5.8. TPU Timer m Control Register 1 (TPU0_TCN1m)

This register controls operation of individual timers. This register configures the timer operation mode and individual prescalers. Registers of the same type are provided for individual timers, and "m" in the abbreviated register name indicates the timer number m (0 to 7).

Bit	31	8
Field	Reserved	
R/W Attribute	R0,WX	
Protection Attribute	WPS	
Initial Value	00000000_00000000_00000000	

Bit	7	6	5	4	3	2	1	0
Field	Reserved			PL	FRT	TMOD	PS	
R/W Attribute	R0,WX			R/W	R/W	R/W	R/W	
Protection Attribute	WPS							
Initial Value	000			0	0	0	00	

[bit31:5] Reserved: Reserved Bits

[bit4] PL: Preload Function Enable/Disable Setting Bit

This bit sets enable/disable of the preload function. The preload function is effective only when the timer operation mode is the overflow mode.

Bit	Description
0	Disable preload function.
1	Enable preload function.

[bit3] FRT: Free-Run Function Enable/Disable Setting Bit

This bit sets enable/disable of the free-run function.

Bit	Description
0	Disable free-run function.
1	Enable free-run function.

[bit2] TMOD: Timer Operation Mode Setting Bit

This bit sets the operation mode of the timer.

Bit	Description
0	Normal mode
1	Overflow mode

[bit1:0] PS: Individual Prescaler Division Setting Bit

This bit sets the division ratio of the individual prescaler. An individual prescaler takes the output clock of the global prescaler, and divides it according to the configured division ratio to generate a clock of a timer-specific frequency.

Bits		Description
0	0	Divided by 1
0	1	Divided by 2
1	0	Divided by 4
1	1	Divided by 16

Notes:

- *For details on timer operation modes, the preload function, and the free-run function, see "3.Explanation of Operation."*
- *TPU0_TCN10 to TPU0_TCN17 share the same register bit configuration.*
- *It is prohibition that carries out timer starting by TPU0_CFG:GLBPS= 0b000000 and TPU0_TCN1m:PS= 0b00.*

5.9. TPU Timer m Current Count Value Register (TPU0_TCCm)

This register indicates the current count value of each timer. Registers of the same type are provided for individual timers, and "m" in the abbreviated register name indicates the timer number m (0 to 7).

Bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	-							
Initial Value	00000000							

Bit	23	0
Field	TCC	
R/W Attribute	R,WX	
Protection Attribute	-	
Initial Value	00000000_00000000_00000000	

[bit31:24] Reserved: Reserved Bits

[bit23:0] TCC: Timer Current Count Value Bits

These bits indicate the current timer count value. They can be read at any time. However, because the timer does not stop and continues counting when the value is read, the value being read may become different from the current timer count value by the time CPU0 gets the value.

Note:

- *TPU0_TCC0 to TPU0_TCC7 share the same register bit configuration.*

6. Others

This section explains the notes when using the time protection unit.

An access to the register of the time protection unit returns a bus error in the following conditions:

- Sequence protection violation against TPU0_UNLOCK
- Writing in TPU0_UNLOCK a value other than that of lock release or locking.
- Privilege protection error
- Read or write attempt in which access is made only to a register undefined area
- Write attempt in which access is made only to bits of WX attribute (including register undefined area)

CHAPTER 24: Security

This chapter explains the functions and operations of the security feature in this product.

1. **Overview**
2. **Security Scope and Access Restriction by the Security**
3. **Using and Configuring the Security**
4. **Registers**

In this chapter, there is a place with two specifications, "SWP_NORMAL_type" and "SWP_LITE_type". Basically please refer to "SWP_NORMAL_type" specification, but when SWP_LITE_type is used, please refer to "SWP_LITE_type" specification and see the Product's HWM.

1. Overview

This section shows brief overview of security features in this product.

This product has security features which forbid unauthorized access to Flash contents and unauthorized use of debugger.

This chapter describes security features supported in this product and procedure to configure it.

2. Security Scope and Access Restriction by the Security

This section explains restriction by the security.

2.1. Security Scope

This section explains scopes of the security.

Security Scope

This product has following "security scope" to protect the device from unauthorized access.

Flash Protection

Protect contents of Flash from external access.

Protect the device from unauthorized debugger access in USERMODE.

Device Protection

Protect contents of Flash from external access.

Protect the device from unauthorized debugger access in USERMODE.

Protect the device from unauthorized code execution in non-USERMODE.

Besides Flash protection it also:

- Prevents read-out of register contents
- Prevents read-out of RAM contents.

Notes:

- *As shown in 2.2 Access Restriction by the Security, It is not restricted to execute a full chip erase (Macro erase) in the parts without SHE*
- *Even with SHE activated parts macro erase of WorkFlash (except SHE sectors) and macro erase of Code Flash can be done (using Serial Loader or JTAG equipment). The two sectors reserved for SHE operation are not erased.*

Refer to datasheet of product.

If being written with "MK_CEER = Selectable", below is spec.

> The ability to execute "chip erase" function depends on the permission set by the marker MK_CEER.
> The possibility of doing chip erase can be removed, when the user intents to forbid the usage of other parties software on his system. However this removes any possibility for a recovery of a device when not implementing an unlock mechanism or forgetting the key for a device unlock.
> To protect the software IP only from being disassembled, debugged or copied – the normal security is enough and chip erase functionality should stay enabled.

Other spec is below.

> The ability to execute "chip erase" always exists. The marker MK_CEER does not have any functionality.

Debugger Security

Debugger connection has to be authenticated to protect the device from unauthorized access. When the device boots in USERMODE, the debugger connection is not initialized by default. Debugger connection to secured device has to be configured by user software.

Two options exist for the authentication process:

1. Flash Marker and 128 bit Flash Key based method
2. Application based authentication (by access to security state controller registers)

2.2. Access Restriction by the Security

This section tabulates access restriction in each operation mode and security scope.

Access Restriction in USERMODE

Operation	USERMODE Security=Disabled	USERMODE Security= Flash Protection	USERMODE Security= Device Protection
TCFlash Macro Erase	Enabled *1	Enabled *1	Enabled *1
TCFlash Sector Erase	Enabled *1	Enabled *1	Enabled *1
WorkFlash Sector Erase	Enabled *1	Enabled *1	Enabled *1
Programming	Enabled *1	Enabled *1	Enabled *1
Reading	Enabled	Enabled	Enabled
Debugger connection	Enabled	Enabled *2	Enabled *2

*1: TCFCFG0_xxxx registers has to be configured by user software.

*2: Debugger connection has to be authorized and initialized by 128-bit passcode or by the user software.

Access Restriction in Non-USERMODE

Operation	Non-USERMODE Security=Disabled	Non-USERMODE Security= Flash Protection Chip Erase Enabled = Enabled	Non-USERMODE Security= Flash Protection Chip Erase Enabled = Disabled	Non-USERMODE Security= Device Protection Chip Erase Enabled = Enabled	Non-USERMODE Security= Device Protection Chip Erase Enabled = Disabled
TCFlash Macro Erase	Enabled	Restricted *1	Disabled	Restricted *2	Disabled
TCFlash Sector Erase	Enabled	Disabled			
WorkFlash Sector Erase	Enabled	Restricted *1			
Programming	Enabled	Restricted *1			
Reading	Enabled	Disabled			
Use of Serial Programmer	Enabled	Restricted *1			
Use of Parallel Programmer	Enabled	Restricted *1			
Debugger connection	Enabled	Enabled *3	Enabled*4		

*1: It has to follow the steps shown in Erase Order Restriction in Non-USERMODE to Devices with Flash Protection or Device Protection.

*2: Only a certain sequence of erase operations (full-chip erase) can be triggered by the writer equipment (Serial or JTAG).

*3: Access to Flash is restricted (only full-chip erase is possible).

*4: Access to Flash is restricted (Nothing can be done to Flash).

3. Using and Configuring the Security

This section explains how to configure the security.

The security settings of the device are stored in a region named Security Marker. After any RSTX_PD2H, the device automatically loads the security settings written in Security Marker and applies the settings to the Security State Registers. To apply the security settings, the Security Marker area has to be programmed according to the definition written in this section.

Each of the security control marker in the Flash has its 1:1 copy in its according security register.

During the runtime of the application, the Security State Registers could be accessed. It could modify the registers state until the overwrite feature gets disabled. That gives the application the chance to temporary change the security setting without reprogramming the Flash. That feature can be used to unlock security after a successful authentication. It can be used for gaining debugger access or for other maintenance tasks.

3.1. Set Security Setting

This section describes how to set security settings.

To enable the security setting, you need to program Security Marker as shown in this section.

Device without Protection

Program Security Marker to following:

MK_SER : Security = Disabled

MK_SSR : Scope = (don’t care)

Device with Flash Protection

Program Security Marker to following:

MK_SER : Security = Enabled

MK_SSR : Scope = Flash Protection

Device with Device Protection

Program Security Marker to following:

MK_SER : Security = Enabled

MK_SSR : Scope = Device Protection

Examples

Address	Security Off (erased)	Flash Protection	Device Protection
0x019F0000 (MK_SER)	0xFFFFFFFF	0x00000001	0x00000001
0x019F0004	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF
0x019F0008 (MK_SSR)	0xFFFFFFFF	0xFFFFFFFF	0x00000001
0x019F000C	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF

3.2. Erasing Security Setting

This section describes how to erase security settings and make possible to reprogram the device.

To erase the security setting and make possible to reprogram the device, you need to reprogram Security Marker as shown in this section.

Device with Flash Protection

In USERMODE

Program following configuration to the security marker

MK_SER : Security = Disabled

By Using Write Equipment

Erase and programming can be possible by the steps shown in Erase Order Restriction in Non-USERMODE to Devices with Flash Protection or Device Protection.

To disable security, program following configuration to the security marker

1. Execute chip erase.
2. MK_SER : Security = Disabled

Device with Device Protection

In USERMODE

Program following configuration to the security marker

MK_SER : Security = Disabled

By Using Write Equipment

Execute "full chip erase" from writer equipments. Then flash contents are erased by the steps shown in Erase Order Restriction in Non-USERMODE to Devices with Flash Protection or Device Protection.

Following configuration will be automatically written to the security marker by the full chip erase operation.

MK_SER : Security = Disabled

Erase Order Restriction in Non-USERMODE to Devices with Flash Protection or Device Protection

For the devices with Flash Protection or Device Protection, erase and programming is possible only by following steps.

[W]→[T2]→[T1]→[T0]→[P]

or [W]→[T1]→[T2]→[T0]→[P]

or [T2]→[W]→[T1]→[T0]→[P]

or [T2]→[T1]→[W]→[T0]→[P]

or [T1]→[W]→[T2]→[T0]→[P]

or [T1]→[T2]→[W]→[T0]→[P]

[W] WorkFlash sector erase to all sectors : from the top sector to the bottom sector. (i.e.: … → sector #5 → sector #4 → sector #3 → sector #2 → sector #1 → sector #0)

[T2] TCFlash#2 macro erase

[T1] TCFlash#1 macro erase

[T0] TCFlash#0 macro erase

[P] Program any data to TCFlash and WorkFlash.

Notes:

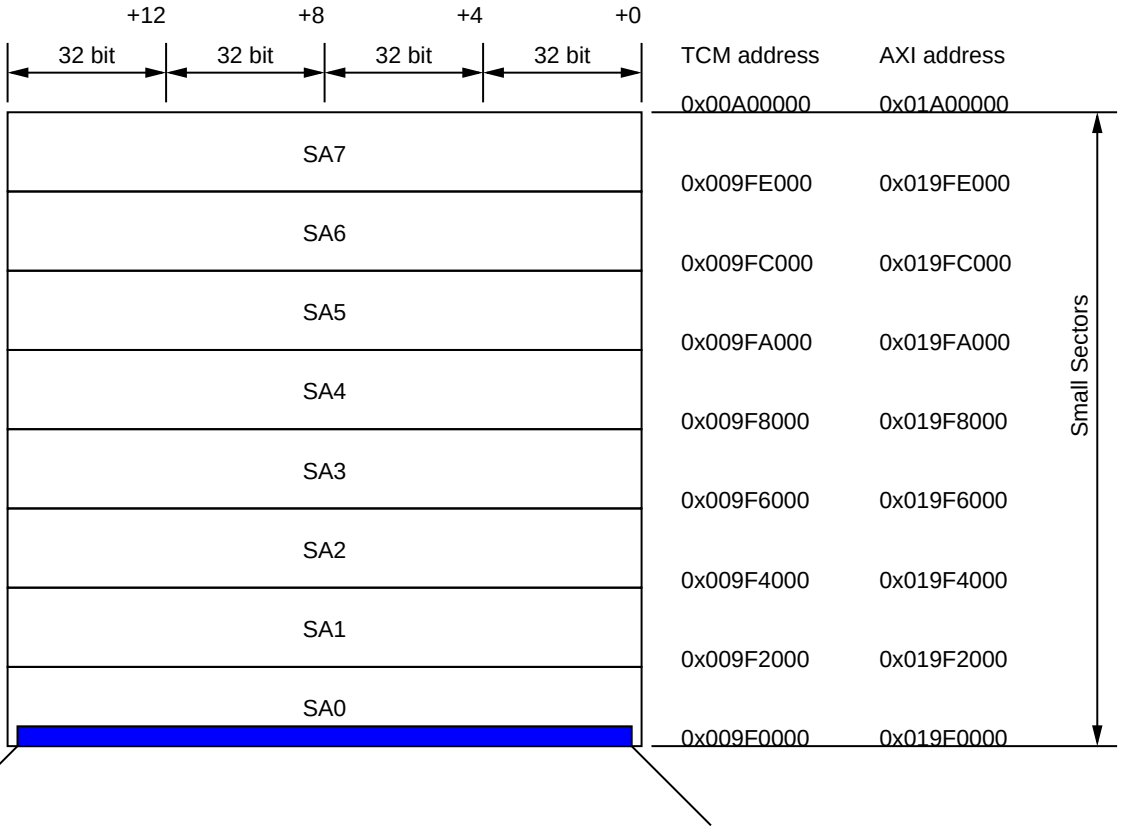
- *If there is a reset between the steps, you have to restart the steps from the first step.*
- *To disable security, program following configuration to the security marker in the last step. {MK_SER : Security = Disabled}*

3.3. Security Marker Definition

This section describes details of the Flash Marker which is used to configure the security setting.

The security configuration is stored in the Flash Marker which is located to head of the SA0 area in small sector area of TCFlash#0.

This section describes details of each marker definition.



+12	+8	+4	+0	Offset for AXI read/write access
Reserved	MK_C3SWPR	Reserved	MK_CS2SWPR	0x019F0040
Reserved	MK_C1SWPR	Reserved	MK_CS0SWPR	0x019F0030
Reserved	MK_WSWPR	Reserved	MK_SWPOER	0x019F0020
Reserved	MK_SOER	Reserved	MK_CEER	0x019F0010
Reserved	MK_SSR	Reserved	MK_SER	0x019F0000

The security markers are transferred after each RSTX_PD2H into the appropriate registers containing the security state of the device. This process is called "Security Fetch".

No errors will succeed in loading the marker values from Flash to registers.

Single bit errors get corrected and will succeed in loading the marker values from Flash to registers.

Fatal errors will not succeed in loading the marker values from Flash to registers. The according security register will keep its previous initial value.

To check detail of the integrity of ECC of the above marker section, it is recommended, to read the markers by the application. The normal ECC error detection by the TCFLASH interface can be utilized for that.

3.3.1. Security Enable Marker (MK_SER)

This section describes the function of the Security Enable Marker.

Offset	0x0000
Size	4 bytes
Name	SER[31:0]

[bit31:0] SER: Security Enable

The value is used to enable the security feature.

Value	Description
0x00000001	The security is enabled. This value should be used as lock code.
0xFFFFFFFF	The security is disabled (unlock value).
Other	Any other value enables the security, but should not be used.

The marker value is automatically copied to TCFCFG0_SER after each RSTX_PD2H.

Note:

- *Single ECC errors are corrected. When double ECC errors are detected, the register is not updated from Flash and security will be enabled.*

3.3.2. Security Scope Marker (MK_SSR)

This section describes the function of the Security Scope Marker.

Offset	0x0008
Size	4 bytes
Name	SSR[31:0]

[bit31:0] SSR: Security Scope

The value is used to select security scope of the device.

Value	Description
0xFFFFFFFF	The security scope is set to Flash protection. Flash access is only possible in USERMODE by the application. The Debug interface is disabled in USERMODE to protect from external access to the Flash (it could be enabled by unlocking after authentication). Debug interface can be used in non-USERMODE, while Flash access is disabled.
Other	The security scope is set to Device protection. It includes no accessibility to Flash, RAM and registers. The Debug interface is disabled all modes (it could be enabled by unlocking after authentication).

The marker value is automatically copied to TCFCFG0_SSR after each RSTX_PD2H.

Note:

- *Single ECC errors are corrected. When double ECC errors are detected, the scope will be set Device protection.*

3.3.3. Chip Erase Enable (MK_CEER)

The functionality of Chip Erase Enable is described in this section.

Offset	0x0010
Size	4 bytes
Name	CEER[31:0]

Refer to datasheet of product.

If being written with "MK_CEER = Selectable", below is spec.

[bit31:0] CEER: Chip Erase Enable

The value controls the ability to execute the "chip erase" function.

Value	Description
0xFFFFFFFF	The MCU will support the "chip erase" command. As an effect a secured device can be unlocked by erasing all its internal secret content, except SHE.
Other	The MCU has disabled the "chip erase" command. Without knowing authentication scheme, the device cannot be erased anymore (independent from its operation mode).

The marker value is automatically copied to TCFCFG0_CEER after each RSTX_PD2H.

Other spec is below.

The marker value does not have any effect to both USERMODE / non-USERMODE operations.

Chip erase operation can be done regardless of this value. A secured device can be unlocked by erasing all its internal secret content, except SHE.

3.3.4. Security Overwrite Enable Marker (MK_SOER)

This section describes the function of the Security Overwrite Enable Marker.

Offset	0x0018
Size	4 bytes
Name	SOER[31:0]

The marker value is automatically copied to TCFCFG0_SOER after each RSTX_PD2H.

[bit31:0] SOER: Security Overwrite Enable

Value	Description
0xFFFFFFFF	The security settings in the registers (SER, SSR, CEER and SOER) can be overwritten.
Other	Any other value disables the write access to the security settings registers including TCFCFG0_SER, TCFCFG0_SSR, TCFCFG0_CEER and TCFCFG0_SOER.

Notes:

- *Single ECC errors are corrected. When double ECC errors are detected, the register is not updated from Flash and the security settings cannot be overwritten.*
- *The marker value does not have any effect to non-USERMODE operations.*

3.3.5. Sector Write Permission Overwrite Enable Marker (MK_SWPOER)

This section describes the function of the Sector Write Permission Overwrite Enable Marker.

Offset	0x0020
Size	4 bytes
Name	SOER[31:0]

The marker value is automatically copied to TCFCFG0_SWPOER after each RSTX_PD2H.

[bit31:0] SWPOER: Sector Write Permission Overwrite Enable

Value	Description
0xFFFFFFFF	The sector write permissions can be overwritten.
Other	Any other value disables the write access to the sector write permission registers including TCFCFG0_SWPOER, TCFCFG0_WSWP, TCFCFG0_C0SWP, TCFCFG0_C1SWP, TCFCFG0_C2SWP and TCFCFG0_C3SWP.

Notes:

- *The marker value does not have any effect to non-USERMODE operations.*
- *In non-USERMODE, any program or sector erase operation cannot be done regardless this marker value when the security is enabled. Only chance is to execute a "full chip erase" command.*
- *Single ECC errors are corrected. When double ECC errors are detected the sector write permissions cannot be overwritten.*

3.3.6. Work Flash Sector Write Permissions (MK_WSWPR)

This section describes the function of the WSWP Default Value Setting from Flash.

Offset	0x0028
Size	4 bytes
Name	WSWP[31:0]

The marker value is automatically copied to TCFCFG0_WSWP after each RSTX_PD2H.

For details please refer to the description of the TCFCFG0_WSWP register.

Notes:

- *The marker value does not have any effect to non-USERMODE operations.*
- *In non-USERMODE, any program or sector erase operation cannot be done regardless this marker value when the security is enabled. Only chance is to execute a “full chip erase” command.*
- *Single ECC errors are corrected. When double ECC errors are detected, the sector write permissions are not set.*

3.3.7. Code Flash Sector Write Permissions, Small Sectors (MK_C0SWPR)

This section describes the function of the C0SWP Default Value Setting from Flash.

Offset	0x0030
Size	4 bytes
Name	C0SWP[31:0]

The marker value is automatically copied to TCFCFG0_C0SWP after each RSTX_PD2H.

For details please refer to the description of the TCFCFG0_C0SWP register.

Notes:

- *The marker value does not have any effect to non-USERMODE operations.*
- *In non-USERMODE, any program or sector erase operation cannot be done regardless this marker value when the security is enabled. Only chance is to execute a "full chip erase" command.*
- *Single ECC errors are corrected. When double ECC errors are detected, the sector write permissions are not set.*

3.3.8. Code Flash Sector Write Permissions, Large Sectors (MK_C1SWPR)

This section describes the function of the C1SWP Default Value Setting from Flash.

Offset	0x0038
Size	4 bytes
Name	C1SWP[31:0]

The marker value is automatically copied to TCFCFG0_C1SWP after each RSTX_PD2H.

For details please refer to the description of the TCFCFG0_C1SWP register.

Notes:

- *The marker value does not have any effect to non-USERMODE operations.*
- *In non-USERMODE, any program or sector erase operation cannot be done regardless this marker value when the security is enabled.*
- *Single ECC errors are corrected. When double ECC errors are detected the sector write permissions are not set.*

3.3.9. Code Flash Sector Write Permissions, Large Sectors (MK_C2SWPR)

This section describes the function of the C2SWP Default Value Setting from Flash.

Offset	0x0040
Size	4 bytes
Name	C2SWP[31:0]

The marker value is automatically copied to TCFCFG0_C2SWP after each RSTX_PD2H.

For details please refer to the description of the TCFCFG0_C2SWP register.

Notes:

- *The marker value does not have any effect to non-USERMODE operations.*
- *In non-USERMODE, any program or sector erase operation cannot be done regardless this marker value when the security is enabled. Only chance is to execute a "full chip erase" command.*
- *Single ECC errors are corrected. When double ECC errors are detected the sector write permissions are not set.*

3.3.10. Code Flash Sector Write Permissions, Large Sectors (MK_C3SWPR)

This section describes the function of the C3SWP Default Value Setting from Flash.

Offset	0x0048
Size	4 bytes
Name	C3SWP[31:0]

The marker value is automatically copied to TCFCFG0_C3SWP after each RSTX_PD2H.

For details please refer to the description of the TCFCFG0_C3SWP register.

Notes:

- *The marker value does not have any effect to non-USERMODE operations.*
- *In non-USERMODE, any program or sector erase operation cannot be done regardless this marker value when the security is enabled. Only chance is to execute a "full chip erase" command.*
- *Single ECC errors are corrected. When double ECC errors are detected the sector write permissions are not set.*

3.4. Gain access in USERMODE

This section describes how to gain access of Flash and debugger in USERMODE.

3.4.1. Gaining program / erase permission to Flash Sectors

After each RSTX_PD2H the default permissions are loaded for each sector from its according Flash values. Sector permissions are grouped for its Flash regions as follows:

- TCFCFG0_WSWP: Work Flash 0
- TCFCFG0_C0SWP: Code Flash 0, small sectors
- TCFCFG0_C1SWP: Code Flash 0, large sectors (devices up to 2 MB Flash)
- TCFCFG0_C2SWP: Code Flash 1, large sectors (only on devices with 4 MB+ Flash)
- TCFCFG0_C3SWP: Code Flash 2, large sectors (only on devices with 6 MB+ Flash)

Each of the bits in the above registers control write permissions (erase and program) of single Flash sectors. Lower bits reflect the permission of the lower sector number. Higher bits reflect the setting of the higher sector number. Depending on the Flash size equipped with the device only such sector write permission registers or bits have an effect, for which the according Flash sector exists.

The sector permission registers are defining accessibility of Flash during USERMODE.

- "1" – Enable erase and program permission
- "0" – Disable erase and program permission

The settings do not have any effect in non-USERMODE. Writing sectors in non-USERMODE is permitted regardless of the configuration of the sector permissions.

In non-USERMODE Flash access can only be forbidden by the activation of security.

When sector write permissions should be frozen right from the beginning, set MK_SWPOER to DISABLED. Note that this is only valid for USERMODE. In non-USERMODE by using Flash programming equipment the Flash could be still erased (in case of non-secured device).

If the control of the sector write permissions should be possible by the application software, MK_SWPOER must be set to ENABLED.

After changing the sector write permissions by the application, any further change can be disabled by writing TCFCFG0_SWPOER = DISABLED. Sector write permissions and TCFCFG0_SWPOER itself is locked until the next RSTX_PD2H. This is useful to restrict updating sectors only during boot loader operation for functional safety reasons.

When keeping TCFCFG0_SWPOER = ENABLED, continuous change of sector write permissions is possible.

To enable macro erase in USERMODE, all sector write permissions need to be set.

Consequently, if any sector write permission of the TCFLASH is disabled, the macro erase command for the particular TCFLASH macro cannot be executed. That means the ability of macro erase is controlled on a per Flash macro basis. If other physically separated Flash macro does not have any own restricted sectors, its macro erase could be executed.

Logically the macro erase permissions would evaluate from sector erase permissions as follows:

```
TCFLASH0_macro_erase_permission =   (TCFCFG0_C0SWP[7:0] == 0xFF) &&
                                    (TCFCFG0_C1SWP == 0xFFFF_FFFF);
TCFLASH1_macro_erase_permission =   (TCFCFG0_C2SWP == 0xFFFF_FFFF);
TCFLASH2_macro_erase_permission =   (TCFCFG0_C3SWP == 0xFFFF_FFFF);
```

The macro erase command is not supported by the WORKFLASH.

- **SWP_LITE_type**

  Following procedure is required to ensure the operation of Sector Write Permissions.

  When programing to TCFLASH, all of following Sector Write Permissions register bits must be set to “1” to permit the TCFLASH programing.

  - Code Flash 0 Sector Write Permissions of the Small Sectors (TCFCFG0_C0SWP)
  - Code Flash 0 Sector Write Permissions of the Large Sectors (TCFCFG0_C1SWP)
  - Code Flash 1 Sector Write Permissions of the Large Sectors (TCFCFG0_C2SWP)
  - Code Flash 2 Sector Write Permissions of the Large Sectors (TCFCFG0_C3SWP)

  After programing, set the original values of Sector Write Permissions again to protect the TCFLASH data.

- **SWP_NORMAL_type**

  No additional procedure is required.

- **end_of_type_listing**

3.4.2. Gaining Debugger Access

Device with Protection (Flash Protection or Device Protection)

To gain the debugger access, do one of the followings:

- Set TCFCFG0_SER to "security disabled" value from the software. The authentication method can be freely chosen by the application.
- Enter 128-bit password from the debugger. Refer "CHAPTER : BOOTROM SOFTWARE INTERFACE" for more details.

Device without Protection

Debugger access is given by default.

4. Registers

After each RSTX_PD2H, the security information fetched from Code Flash #0 is stored in registers. The security state registers can be read and written by the application, to change the security state. That is needed to implement several unlocking mechanisms by the application software, i.e. to temporary allow a debugger access after sending a user defined unlock key.

Table 4-1 Security Register List

Abbreviated Register Name	Register Name	Reference
TCFCFG0_SECSTAT	Security Status Register	4.1
TCFCFG0_SER	Security Enable Register	4.2
TCFCFG0_SSR	Security Scope Register	4.3
TCFCFG0_CEER	Chip Erase Enable Register	4.4
TCFCFG0_SOER	Security Overwrite Enable Register	4.5

Table 4-2 Sector Write Permissions Register List

Abbreviated Register Name	Register Name	Reference
TCFCFG0_SWPOER	Sector Write Permission Overwrite Enable Register	4.6
TCFCFG0_WSWP	Work Flash Sector Write Permissions	4.7
TCFCFG0_C0SWP	Code Flash 0 Sector Write Permissions (small sectors)	4.8
TCFCFG0_C1SWP	Code Flash 0 Sector Write Permissions (large sectors)	4.9
TCFCFG0_C2SWP	Code Flash 1 Sector Write Permissions (large sectors)	4.10
TCFCFG0_C3SWP	Code Flash 2 Sector Write Permissions (large sectors)	4.11

Note:

- *64-bit access is not supported to the registers.*

Table4-3 shows the register address map.

Table 4-3 Operation Security Register Map

Offset	Register Name/Initial Value
0x0000_0000	TCFCFG0_SECSTAT 00000000_00000000_0000000*_XX1**1**
0x0000_0004	TCFCFG0_SER ********_********_********_********
0x0000_0008	TCFCFG0_SSR ********_********_********_********
0x0000_000C	TCFCFG0_CEER ********_********_********_********
0x0000_0010	TCFCFG0_SOER ********_********_********_********
0x0000_0014	TCFCFG0_SWPOER ********_********_********_********
0x0000_0018	TCFCFG0_WSWP ********_********_********_********
0x0000_001C	TCFCFG0_C0SWP ********_********_********_********
0x0000_0020	TCFCFG0_C1SWP ********_********_********_********
0x0000_0024	TCFCFG0_C2SWP ********_********_********_********
0x0000_0028	TCFCFG0_C3SWP ********_********_********_********

"X": Initial value undefined (0 or 1)

"*": Initial value "0" or "1" according to the setting (designed by higher layer)

4.1. Security Status Register (TCFCFG0_SECSTAT)

This section describes the function of the Security Status Register.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved							
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved							
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	*	*	*	*	*	*	*	*
BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved							UMV
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	1

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	Reserved	SFDONE	SWPOE	SECOE	CEEN	SECSC	SECEN
ACCESS_TYPE	RX,WX	RX,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE	-							
INITIAL_VALUE	X	X	1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1

[bit31:16] Reserved: Reserved bits

These bits are not used. Writing has no effect.

[bit15:9] Reserved: Reserved bits

These bits are not used. Writing has no effect.

[bit8] UMV: Specification of the unlock marker value

Bit	Description
0	The device supports the non-trivial unlock marker value 0x692E3A7B.
1	The device supports the trivial unlock marker value 0xFFFFFFFF.

The register bit UMV indicates what value has to be written to the markers for the purpose of unlocking or to configure the less secure option. It has effect on the interpretation of MK_SER, MK_SSR, MK_CEER, MK_SOER and MK_SWPOER.

"1" is always read.

UMV is a read-only bit. Any write attempt will cause a bus error.

[bit7:6] Reserved: Reserved bits

These bits are not used. Any write attempt will cause a bus error.

[bit5] SFDONE: Security fetch done register

Bit	Description
0	The security fetch is ongoing.
1	The security fetch is finished.

The register bit SFDONE indicates the progress of the evaluation of the security fetch of the security state controller.

SFDONE is a read-only bit. Any write attempt will cause a bus error.

[bit4] SWPOE: Sector write permission overwrite enable register

Bit	Description
0	Sector write permissions are fixed.
1	Sector write permissions can be overwritten by the application.

The register bit SWPOE indicates the ability to overwrite the sector write permissions.

The sector write permissions are initialized from Flash memory after RSTX_PD2H (security fetch).

SWPOE is a read-only bit. Any write attempt will cause a bus error.

[bit3] SECOE: Security overwrite enable register

Bit	Description
0	The security state information (SER, SSR, CEER, SOER) is fixed.
1	The security state information (SER, SSR, CEER, SOER) can be overwritten by the application.

The register bit SECOE indicates the ability to overwrite the security state by the application in user mode.

Other modes that user mode do not allow the overwriting of the security state. The reason is, that unlocking can be done only by the application (using some authentication scheme).

The security state is initialized from Flash markers and read after RSTX_PD2H (security fetch).

SECOE is a read-only bit. Any write attempt will cause a bus error.

[bit2] CEEN: Chip erase enable register

Bit	Description
0	A full erasure of the chip (TCFLASH and WORKFLASH) is not permitted.
1	A full erasure of the chip (TCFLASH and WORKFLASH) is always permitted, regardless of the security enable or the state of the sector write permissions.

The register bit CEEN indicates the permission to erase the full device.

Dedicated memory of the SHE module cannot be erased, regardless of that setting.

CEEN is a read-only bit. Any write attempt will cause a bus error.

Refer to datasheet of product.

If being written with "MK_CEER != Selectable", below is spec.

In this product CEEN is always ‘1’ (fixed, because MK_CEER marker is not supported).

[bit1] SECSC: Security scope register

Bit	Description
0	The security scope is Flash protection. - During non-user mode Flash memory cannot be accessed. - At user mode the debug interface cannot be used.
1	The security scope is device protection. - During non-user mode Flash, RAM and registers cannot be accessed. - The debug interface cannot be used at any mode.

The register bit SECSC indicates the scope of the security. The value of that register is only relevant when security is enabled (SECEN=1).

SECSC is a read-only bit. Any write attempt will cause a bus error.

[bit0] SECEN: Security enable register

Bit	Description
0	The security is disabled.
1	The security is enabled.

The register bit SECEN indicates the general security enabled state.

SECEN is a read-only bit. Any write attempt will cause a bus error.

4.2. Security Enable Register (TCFCFG0_SER)

This section describes the function of the Security Enable Register.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	SER[31:24]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	WP							
INITIAL_VALUE	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	SER[23:16]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	WP							
INITIAL_VALUE	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	SER[15:8]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	WP							
INITIAL_VALUE	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	SER[7:0]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	WP							
INITIAL_VALUE	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1

[bit31:0] SER: Security enable register

SER	Description
0x00000001	The security is enabled. This value should be used as lock code.
0xFFFFFFFF	The security is disabled (unlock value).
Other	Any other value enables the security, but should not be used.

After RSTX_PD2H the Security Enable Marker (MK_SER) is transferred from Flash into the Security Enable Register (TCFCFG0_SER).

TCFCFG0_SER can be written by the application, when TCFCFG0_SOER indicates that overwrite is enabled. Otherwise the write access will cause a bus error. TCFCFG0_SER is typically used, to gain debug access of a secured device after a successful authentication. The authentication method should be implemented in the application before setting the MK_SER to "secured".

The register is written to 0xFFFFFFFF to gain debugger access in USERMODE in application. As alternative, authentication by hardware using 128 bit key can be used (see DDR_DSM, BOOTROM

SOFTWARE INTERFACE). The key is stored in Flash and does not require software for the authentication.

Note:

- *Single ECC errors are corrected. When double ECC errors are detected, the register is not updated from Flash and security will be enabled.*

4.3. Security Scope Register (TCFCFG0_SSR)

This section describes the function of the Security Scope Register.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	SSR[31:24]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	WP							
INITIAL_VALUE	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	SSR[23:16]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	WP							
INITIAL_VALUE	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	SSR[15:8]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	WP							
INITIAL_VALUE	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	SSR[7:0]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	WP							
INITIAL_VALUE	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1

[bit31:0] SSR: Security scope register

SSR	Description
0xFFFFFFFF	The security scope is Flash protection. Debugger connection can be used in non-USERMODE without the possibility of having Flash access. At USERMODE the debugger interface is disabled.
Other	The security scope is Device protection. It includes no accessibility to Flash, RAM and registers from outside the device at non-USERMODE.

After RSTX_PD2H the Security Scope Marker (MK_SSR) is transferred from Flash into the Security Scope Register (TCFCFG0_SSR).

TCFCFG0_SSR can be written by the application, when TCFCFG0_SOER indicates that overwrite is enabled. Otherwise the write access will cause a bus error. Writing this register has no effect on the execution of the application.

Note:

- *Single ECC errors are corrected. When double ECC errors are detected, the register is not updated from Flash and the scope will be set to the more safe state of Device protection.*

4.4. Chip Erase Enable Register (TCFCFG0_CEER)

This section describes the function of the Chip Erase Enable Register.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	CEER[31:24]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	WP							
INITIAL_VALUE	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	CEER[23:16]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	WP							
INITIAL_VALUE	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	CEER[15:8]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	WP							
INITIAL_VALUE	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	CEER[7:0]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	WP							
INITIAL_VALUE	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1

[bit31:0] CEER: Chip erase enable register

Refer to datasheet of product.

If being written with "MK_CEER = Selectable", below is spec.

Full-chip erase can be forbidden using this marker.

CEER	Description
0xFFFFFFFF	The chip erase of a secured device is enabled.
Other	Any other value disables the chip erase in non-USERMODE.

Without enabling the security by TCFCFG0_SER, the Chip Erase Enable Register has no meaning.

Other spec is below.

Disabling chip erase is not possible by this product. Writing to that register has no effect.

Bit	Description
Any	No effect

After each RSTX_PD2H the Chip Erase Enable Marker (MK_CEER) is transferred from Flash into the Chip Erase Enable Register (TCFCFG0_CEER).

TCFCFG0_CEER can be written by the application, when TCFCFG0_SOER indicates that overwrite is enabled. Otherwise the write access will cause a bus error.

Refer to datasheet of product.

If being written with "MK_CEER = Selectable", below is spec.

The chip erase can be triggered by JTAG equipment, writing the chip erase sequence to the general purpose registers of the security checker or transferring the chip erase sequence through serial interface when using Serial Loader equipment. It erases all sectors of the WORKFLASH and TCFLASH. Sectors used for SHE are not erased.

Notes:

- *A secured device without an unlock mechanism implemented by the application and without the ability to erase the full device cannot be updated or tested anymore. Not programming the unlock marker value to MK_CEER should be used with absolute care!*
- *Single ECC errors are corrected. When double ECC errors are detected, the register is not updated from Flash and the chip erase will be disabled.*

4.5. Security Overwrite Enable Register (TCFCFG0_SOER)

This section describes the function of the Security Overwrite Enable Register.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	SOER[31:24]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	WP							
INITIAL_VALUE	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	SOER[23:16]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	WP							
INITIAL_VALUE	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	SOER[15:8]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	WP							
INITIAL_VALUE	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	SOER[7:0]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	WP							
INITIAL_VALUE	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1

[bit31:0] SOER: Security overwrite enable register

SOER	Description
0xFFFFFFFF	The security settings can be overwritten.
Other	Any other value disables the write access to the security settings registers including TCFCFG0_SER, TCFCFG0_SSR, TCFCFG0_CEER and TCFCFG0_SOER.

After each RSTX_PD2H the Security Overwrite Enable Marker (MK_SOER) is transferred from Flash into the Security Overwrite Enable Register (TCFCFG0_SOER).

TCFCFG0_SOER can be written by the application, when TCFCFG0_SOER indicates that overwrite is enabled. Otherwise the write access will cause a bus error. TCFCFG0_SOER is used to implement a self-lock mechanism by intention. The application could disable the write access to the security registers in the main branch of the application for functional safety reasons, whenever it seems a

protected sequence is not sufficient. It cannot be enabled again, except after execution of a RSTX_PD2H.

Byte and half word access to TCFCFG0_SOER cannot be allowed, because a partial modification would lock immediately the write access. The attempt will be rejected with a bus error response.

Note:

- *Single ECC errors are corrected. When double ECC errors are detected, the register is not updated from Flash and the security settings cannot be overwritten.*

4.6. Sector Write Permission Overwrite Enable Register (TCFCFG0_SWPOER)

This section describes the function of the Sector Write Permission Overwrite Enable Register.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	SWPOER[31:24]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	WP							
INITIAL_VALUE	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	SWPOER[23:16]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	WP							
INITIAL_VALUE	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	SWPOER[15:8]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	WP							
INITIAL_VALUE	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	SWPOER[7:0]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	WP							
INITIAL_VALUE	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1

[bit31:0] SWPOER: Sector write permission overwrite enable register

SWPOER	Description
0xFFFFFFFF	Write to *SWP* registers is enabled. The sector write permissions and this register can be overwritten.
Other	Write to *SWP* registers is disabled. Any other value disables the write access to the sector write permission registers including TCFCFG0_SWPOER, TCFCFG0_WSWP, TCFCFG0_C0SWP, TCFCFG0_C1SWP, TCFCFG0_C2SWP and TCFCFG0_C3SWP.

After each RSTX_PD2H the Sector Write Permission Overwrite Enable Marker (MK_SWPOER) is transferred from Flash into the Sector Write Permission Overwrite Enable Register (TCFCFG0_SWPOER).

TCFCFG0_SWPOER can be written by the application, when TCFCFG0_SWPOER indicates that overwrite is enabled. Otherwise the write access will cause a bus error. TCFCFG0_SWPOER is used to implement a self-lock mechanism by intention. The application could disable the write access to the

sector write permission registers in the main branch of the application for functional safety reasons, whenever it seems a protected sequence is not sufficient. Once TCFCFG0_SWPOER was set to "disabled", it cannot be set to "enabled" again, except after execution of a RSTX_PD2H.

Byte and half word access to TCFCFG0_SWPOER cannot be allowed, because a partial modification would lock immediately the write access. The attempt will be rejected with a bus error response.

Note:

- *Single ECC errors are corrected. When double ECC errors are detected, the register is not updated from Flash and the sector write permissions cannot be overwritten.*

4.7. Work Flash Sector Write Permissions Register (TCFCFG0_WSWP)

This section describes the function of the Work Flash Sector Write Permission Register.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	WP							
INITIAL_VALUE	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	WP							
INITIAL_VALUE	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	Reserved	WSWP13	WSWP12	WSWP11	WSWP10	WSWP9	WSWP8
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	WP							
INITIAL_VALUE	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	WSWP7	WSWP6	WSWP5	WSWP4	WSWP3	WSWP2	WSWP1	WSWP0
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	WP							
INITIAL_VALUE	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1

After each RSTX_PD2H the Work Flash Sector Write Permissions (WSWP) are transferred from Flash into the Work Flash Sector Write Permissions Register (TCFCFG0_WSWP).

TCFCFG0_WSWP can be written by the application, when TCFCFG0_SWPOER indicates that overwrite is enabled. Otherwise the write access will cause a bus error and the permission set is fixed.

Notes:

- *Single ECC errors are corrected. When double ECC errors are detected, the register is not updated from Flash and the sector write permissions are not set.*
- *Sector write permissions do not protect Flash sectors in non-USERMODE.*
- *If any of sector permission is set to disabled, macro erase is disabled.*

Bit [0] WSWP0: Work Flash Sector Write Permission SA0

Value	Description
0	Work Flash SA0 cannot be programmed or erased.
1	Work Flash SA0 program or erase is permitted.

Bit [1] WSWP1: Work Flash Sector Write Permission SA1

Value	Description
0	Work Flash SA1 cannot be programmed or erased.
1	Work Flash SA1 program or erase is permitted.

Bit [2] WSWP2: Work Flash Sector Write Permission SA2

Value	Description
0	Work Flash SA2 cannot be programmed or erased.
1	Work Flash SA2 program or erase is permitted.

Bit [3] WSWP3: Work Flash Sector Write Permission SA3

Value	Description
0	Work Flash SA3 cannot be programmed or erased.
1	Work Flash SA3 program or erase is permitted.

Bit [4] WSWP4: Work Flash Sector Write Permission SA4

Value	Description
0	Work Flash SA4 cannot be programmed or erased.
1	Work Flash SA4 program or erase is permitted.

Bit [5] WSWP5: Work Flash Sector Write Permission SA5

Value	Description
0	Work Flash SA5 cannot be programmed or erased.
1	Work Flash SA5 program or erase is permitted.

Bit [6] WSWP6: Work Flash Sector Write Permission SA6

Value	Description
0	Work Flash SA6 cannot be programmed or erased.
1	Work Flash SA6 program or erase is permitted.

Bit [7] WSWP7: Work Flash Sector Write Permission SA7

Value	Description
0	Work Flash SA7 cannot be programmed or erased.
1	Work Flash SA7 program or erase is permitted.

Bit [8] WSWP8: Work Flash Sector Write Permission SA8

Value	Description
0	Work Flash SA8 cannot be programmed or erased.
1	Work Flash SA8 program or erase is permitted.

Bit [9] WSWP9: Work Flash Sector Write Permission SA9

Value	Description
0	Work Flash SA9 cannot be programmed or erased.
1	Work Flash SA9 program or erase is permitted.

Bit [10] WSWP10: Work Flash Sector Write Permission SA10

Value	Description
0	Work Flash SA10 cannot be programmed or erased.
1	Work Flash SA10 program or erase is permitted.

Bit [11] WSWP11: Work Flash Sector Write Permission SA11

Value	Description
0	Work Flash SA11 cannot be programmed or erased.
1	Work Flash SA11 program or erase is permitted.

Bit [12] WSWP12: Work Flash Sector Write Permission SA12

Value	Description
0	Work Flash SA12 cannot be programmed or erased.
1	Work Flash SA12 program or erase is permitted.

Bit [13] WSWP13: Work Flash Sector Write Permission SA13

Value	Description
0	Work Flash SA13 cannot be programmed or erased.
1	Work Flash SA13 program or erase is permitted.

Bit [31:14] Reserved

4.8. Code Flash 0 Sector Write Permissions of the Small Sectors Register (TCFCFG0_C0SWP)

This section describes the function of the Code Flash Sector Write Permission Register of the small sectors.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	WP							
INITIAL_VALUE	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	WP							
INITIAL_VALUE	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	WP							
INITIAL_VALUE	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	C0SWP7	C0SWP6	C0SWP5	C0SWP4	C0SWP3	C0SWP2	C0SWP1	C0SWP0
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	WP							
INITIAL_VALUE	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1

After each RSTX_PD2H the Code Flash Sector Write Permissions of the small sectors (C0SWP) are transferred from Flash into the Code Flash Sector Write Permissions Register of the small sectors (TCFCFG0_C0SWP).

TCFCFG0_C0SWP can be written by the application, when TCFCFG0_SWPOER indicates that overwrite is enabled. Otherwise the write access will cause a bus error and the permission set is fixed.

Notes:

- *Single ECC errors are corrected. When double ECC errors are detected, the register is not updated from Flash and the sector write permissions are not set.*
- *Sector write permissions are ignored in non-USERMODE.*
- *If any of sector permission is set to disabled, macro erase is disabled.*

Bit [0] C0SWP0: Code Flash Sector Write Permission SA0

Value	Description
0	Code Flash SA0 cannot be programmed or erased.
1	Code Flash SA0 program or erase is permitted.

Bit [1] C0SWP1: Code Flash Sector Write Permission SA1

Value	Description
0	Code Flash SA1 cannot be programmed or erased.
1	Code Flash SA1 program or erase is permitted.

Bit [2] C0SWP2: Code Flash Sector Write Permission SA2

Value	Description
0	Code Flash SA2 cannot be programmed or erased.
1	Code Flash SA2 program or erase is permitted.

Bit [3] C0SWP3: Code Flash Sector Write Permission SA3

Value	Description
0	Code Flash SA3 cannot be programmed or erased.
1	Code Flash SA3 program or erase is permitted.

Bit [4] C0SWP4: Code Flash Sector Write Permission SA4

Value	Description
0	Code Flash SA4 cannot be programmed or erased.
1	Code Flash SA4 program or erase is permitted.

Bit [5] C0SWP5: Code Flash Sector Write Permission SA5

Value	Description
0	Code Flash SA5 cannot be programmed or erased.
1	Code Flash SA5 program or erase is permitted.

Bit [6] C0SWP6: Code Flash Sector Write Permission SA6

Value	Description
0	Code Flash SA6 cannot be programmed or erased.
1	Code Flash SA6 program or erase is permitted.

Bit [7] C0SWP7: Code Flash Sector Write Permission SA7

Value	Description
0	Code Flash SA7 cannot be programmed or erased.
1	Code Flash SA7 program or erase is permitted.

4.9. Code Flash 0 Sector Write Permissions of the Large Sectors Register (TCFCFG0_C1SWP)

This section describes the function of the Code Flash 0 Sector Write Permission Register of the large sectors.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	C1SWP39	C1SWP38	C1SWP37	C1SWP36	C1SWP35	C1SWP34	C1SWP33	C1SWP32
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	WP							
INITIAL_VALUE	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	C1SWP31	C1SWP30	C1SWP29	C1SWP28	C1SWP27	C1SWP26	C1SWP25	C1SWP24
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	WP							
INITIAL_VALUE	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	C1SWP23	C1SWP22	C1SWP21	C1SWP20	C1SWP19	C1SWP18	C1SWP17	C1SWP16
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	WP							
INITIAL_VALUE	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	C1SWP15	C1SWP14	C1SWP13	C1SWP12	C1SWP11	C1SWP10	C1SWP9	C1SWP8
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	WP							
INITIAL_VALUE	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1

After each RSTX_PD2H the Code Flash Sector Write Permissions of the large sectors (C1SWP) are transferred from Flash into the Code Flash Sector Write Permissions Register of the large sectors (TCFCFG0_C1SWP).

TCFCFG0_C1SWP can be written by the application, when TCFCFG0_SWPOER indicates that overwrite is enabled. Otherwise the write access will cause a bus error and the permission set is fixed.

Notes:

- *Single ECC errors are corrected. When double ECC errors are detected, the register is not updated from Flash and the sector write permissions are not set.*
- *Sector write permissions are ignored in non-USERMODE.*
- *If any of sector permission is set to disabled, macro erase is disabled.*

Bit [0] C1SWP8: Code Flash 0 Sector Write Permission SA8

Value	Description
0	Code Flash SA8 cannot be programmed or erased.
1	Code Flash SA8 program or erase is permitted.

Bit [1] C1SWP9: Code Flash 0 Sector Write Permission SA9

Value	Description
0	Code Flash SA9 cannot be programmed or erased.
1	Code Flash SA9 program or erase is permitted.

Bit [2] C1SWP10: Code Flash 0 Sector Write Permission SA10

Value	Description
0	Code Flash SA10 cannot be programmed or erased.
1	Code Flash SA10 program or erase is permitted.

Bit [3] C1SWP11: Code Flash 0 Sector Write Permission SA11

Value	Description
0	Code Flash SA11 cannot be programmed or erased.
1	Code Flash SA11 program or erase is permitted.

Bit [4] C1SWP12: Code Flash 0 Sector Write Permission SA12

Value	Description
0	Code Flash SA12 cannot be programmed or erased.
1	Code Flash SA12 program or erase is permitted.

Bit [5] C1SWP13: Code Flash 0 Sector Write Permission SA13

Value	Description
0	Code Flash SA13 cannot be programmed or erased.
1	Code Flash SA13 program or erase is permitted.

Bit [6] C1SWP14: Code Flash 0 Sector Write Permission SA14

Value	Description
0	Code Flash SA14 cannot be programmed or erased.
1	Code Flash SA14 program or erase is permitted.

Bit [7] C1SWP15: Code Flash 0 Sector Write Permission SA15

Value	Description
0	Code Flash SA15 cannot be programmed or erased.
1	Code Flash SA15 program or erase is permitted.

Bit [8] C1SWP16: Code Flash 0 Sector Write Permission SA16

Value	Description
0	Code Flash SA16 cannot be programmed or erased.
1	Code Flash SA16 program or erase is permitted.

Bit [9] C1SWP17: Code Flash 0 Sector Write Permission SA17

Value	Description
0	Code Flash SA17 cannot be programmed or erased.
1	Code Flash SA17 program or erase is permitted.

Bit [10] C1SWP18: Code Flash 0 Sector Write Permission SA18

Value	Description
0	Code Flash SA18 cannot be programmed or erased.
1	Code Flash SA18 program or erase is permitted.

Bit [11] C1SWP19: Code Flash 0 Sector Write Permission SA19

Value	Description
0	Code Flash SA19 cannot be programmed or erased.
1	Code Flash SA19 program or erase is permitted.

Bit [12] C1SWP20: Code Flash 0 Sector Write Permission SA20

Value	Description
0	Code Flash SA20 cannot be programmed or erased.
1	Code Flash SA20 program or erase is permitted.

Bit [13] C1SWP21: Code Flash 0 Sector Write Permission SA21

Value	Description
0	Code Flash SA21 cannot be programmed or erased.
1	Code Flash SA21 program or erase is permitted.

Bit [14] C1SWP22: Code Flash 0 Sector Write Permission SA22

Value	Description
0	Code Flash SA22 cannot be programmed or erased.
1	Code Flash SA22 program or erase is permitted.

Bit [15] C1SWP23: Code Flash 0 Sector Write Permission SA23

Value	Description
0	Code Flash SA23 cannot be programmed or erased.
1	Code Flash SA23 program or erase is permitted.

Bit [16] C1SWP24: Code Flash 0 Sector Write Permission SA24

Value	Description
0	Code Flash SA24 cannot be programmed or erased.
1	Code Flash SA24 program or erase is permitted.

Bit [17] C1SWP25: Code Flash 0 Sector Write Permission SA25

Value	Description
0	Code Flash SA25 cannot be programmed or erased.
1	Code Flash SA25 program or erase is permitted.

Bit [18] C1SWP26: Code Flash 0 Sector Write Permission SA26

Value	Description
0	Code Flash SA26 cannot be programmed or erased.
1	Code Flash SA26 program or erase is permitted.

Bit [19] C1SWP27: Code Flash 0 Sector Write Permission SA27

Value	Description
0	Code Flash SA27 cannot be programmed or erased.
1	Code Flash SA27 program or erase is permitted.

Bit [20] C1SWP28: Code Flash 0 Sector Write Permission SA28

Value	Description
0	Code Flash SA28 cannot be programmed or erased.
1	Code Flash SA28 program or erase is permitted.

Bit [21] C1SWP29: Code Flash 0 Sector Write Permission SA29

Value	Description
0	Code Flash SA29 cannot be programmed or erased.
1	Code Flash SA29 program or erase is permitted.

Bit [22] C1SWP30: Code Flash 0 Sector Write Permission SA30

Value	Description
0	Code Flash SA30 cannot be programmed or erased.
1	Code Flash SA30 program or erase is permitted.

Bit [23] C1SWP31: Code Flash 0 Sector Write Permission SA31

Value	Description
0	Code Flash SA31 cannot be programmed or erased.
1	Code Flash SA31 program or erase is permitted.

Bit [24] C1SWP32: Code Flash 0 Sector Write Permission SA32

Value	Description
0	Code Flash SA32 cannot be programmed or erased.
1	Code Flash SA32 program or erase is permitted.

Bit [25] C1SWP33: Code Flash 0 Sector Write Permission SA33

Value	Description
0	Code Flash SA33 cannot be programmed or erased.
1	Code Flash SA33 program or erase is permitted.

Bit [26] C1SWP34: Code Flash 0 Sector Write Permission SA34

Value	Description
0	Code Flash SA34 cannot be programmed or erased.
1	Code Flash SA34 program or erase is permitted.

Bit [27] C1SWP35: Code Flash 0 Sector Write Permission SA35

Value	Description
0	Code Flash SA35 cannot be programmed or erased.
1	Code Flash SA35 program or erase is permitted.

Bit [28] C1SWP36: Code Flash 0 Sector Write Permission SA36

Value	Description
0	Code Flash SA36 cannot be programmed or erased.
1	Code Flash SA36 program or erase is permitted.

Bit [29] C1SWP37: Code Flash 0 Sector Write Permission SA37

Value	Description
0	Code Flash SA37 cannot be programmed or erased.
1	Code Flash SA37 program or erase is permitted.

Bit [30] C1SWP38: Code Flash 0 Sector Write Permission SA38

Value	Description
0	Code Flash SA38 cannot be programmed or erased.
1	Code Flash SA38 program or erase is permitted.

Bit [31] C1SWP39: Code Flash 0 Sector Write Permission SA39

Value	Description
0	Code Flash SA39 cannot be programmed or erased.
1	Code Flash SA39 program or erase is permitted.

4.10. Code Flash 1 Sector Write Permissions of the Large Sectors Register (TCFCFG0_C2SWP)

This section describes the function of the Code Flash 1 Sector Write Permission Register of the large sectors.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	C2SWP39	C2SWP38	C2SWP37	C2SWP36	C2SWP35	C2SWP34	C2SWP33	C2SWP32
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	WP							
INITIAL_VALUE	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	C2SWP31	C2SWP30	C2SWP29	C2SWP28	C2SWP27	C2SWP26	C2SWP25	C2SWP24
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	WP							
INITIAL_VALUE	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	C2SWP23	C2SWP22	C2SWP21	C2SWP20	C2SWP19	C2SWP18	C2SWP17	C2SWP16
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	WP							
INITIAL_VALUE	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	C2SWP15	C2SWP14	C2SWP13	C2SWP12	C2SWP11	C2SWP10	C2SWP9	C2SWP8
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	WP							
INITIAL_VALUE	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1

After each RSTX_PD2H the Code Flash 1 Sector Write Permissions of the large sectors (C2SWP) are transferred from Flash into the Code Flash 1 Sector Write Permissions Register of the large sectors (TCFCFG0_C2SWP).

TCFCFG0_C2SWP can be written by the application, when TCFCFG0_SWPOER indicates that overwrite is enabled. Otherwise the write access will cause a bus error and the permission set is fixed.

Notes:

- *Single ECC errors are corrected. When double ECC errors are detected, the register is not updated from Flash and the sector write permissions are not set.*
- *Sector write permissions are ignored in non-USERMODE.*
- *If any of sector permission is set to disabled, macro erase is disabled.*

Bit [0] C2SWP8: Code Flash 1 Sector Write Permission SA8

Value	Description
0	Code Flash SA8 cannot be programmed or erased.
1	Code Flash SA8 program or erase is permitted.

Bit [1] C2SWP9: Code Flash 1 Sector Write Permission SA9

Value	Description
0	Code Flash SA9 cannot be programmed or erased.
1	Code Flash SA9 program or erase is permitted.

Bit [2] C2SWP10: Code Flash 1 Sector Write Permission SA10

Value	Description
0	Code Flash SA10 cannot be programmed or erased.
1	Code Flash SA10 program or erase is permitted.

Bit [3] C2SWP11: Code Flash 1 Sector Write Permission SA11

Value	Description
0	Code Flash SA11 cannot be programmed or erased.
1	Code Flash SA11 program or erase is permitted.

Bit [4] C2SWP12: Code Flash 1 Sector Write Permission SA12

Value	Description
0	Code Flash SA12 cannot be programmed or erased.
1	Code Flash SA12 program or erase is permitted.

Bit [5] C2SWP13: Code Flash 1 Sector Write Permission SA13

Value	Description
0	Code Flash SA13 cannot be programmed or erased.
1	Code Flash SA13 program or erase is permitted.

Bit [6] C2SWP14: Code Flash 1 Sector Write Permission SA14

Value	Description
0	Code Flash SA14 cannot be programmed or erased.
1	Code Flash SA14 program or erase is permitted.

Bit [7] C2SWP15: Code Flash 1 Sector Write Permission SA15

Value	Description
0	Code Flash SA15 cannot be programmed or erased.
1	Code Flash SA15 program or erase is permitted.

Bit [8] C2SWP16: Code Flash 1 Sector Write Permission SA16

Value	Description
0	Code Flash SA16 cannot be programmed or erased.
1	Code Flash SA16 program or erase is permitted.

Bit [9] C2SWP17: Code Flash 1 Sector Write Permission SA17

Value	Description
0	Code Flash SA17 cannot be programmed or erased.
1	Code Flash SA17 program or erase is permitted.

Bit [10] C2SWP18: Code Flash 1 Sector Write Permission SA18

Value	Description
0	Code Flash SA18 cannot be programmed or erased.
1	Code Flash SA18 program or erase is permitted.

Bit [11] C2SWP19: Code Flash 1 Sector Write Permission SA19

Value	Description
0	Code Flash SA19 cannot be programmed or erased.
1	Code Flash SA19 program or erase is permitted.

Bit [12] C2SWP20: Code Flash 1 Sector Write Permission SA20

Value	Description
0	Code Flash SA20 cannot be programmed or erased.
1	Code Flash SA20 program or erase is permitted.

Bit [13] C2SWP21: Code Flash 1 Sector Write Permission SA21

Value	Description
0	Code Flash SA21 cannot be programmed or erased.
1	Code Flash SA21 program or erase is permitted.

Bit [14] C2SWP22: Code Flash 1 Sector Write Permission SA22

Value	Description
0	Code Flash SA22 cannot be programmed or erased.
1	Code Flash SA22 program or erase is permitted.

Bit [15] C2SWP23: Code Flash 1 Sector Write Permission SA23

Value	Description
0	Code Flash SA23 cannot be programmed or erased.
1	Code Flash SA23 program or erase is permitted.

Bit [16] C2SWP24: Code Flash 1 Sector Write Permission SA24

Value	Description
0	Code Flash SA24 cannot be programmed or erased.
1	Code Flash SA24 program or erase is permitted.

Bit [17] C2SWP25: Code Flash 1 Sector Write Permission SA25

Value	Description
0	Code Flash SA25 cannot be programmed or erased.
1	Code Flash SA25 program or erase is permitted.

Bit [18] C2SWP26: Code Flash 1 Sector Write Permission SA26

Value	Description
0	Code Flash SA26 cannot be programmed or erased.
1	Code Flash SA26 program or erase is permitted.

Bit [19] C2SWP27: Code Flash 1 Sector Write Permission SA27

Value	Description
0	Code Flash SA27 cannot be programmed or erased.
1	Code Flash SA27 program or erase is permitted.

Bit [20] C2SWP28: Code Flash 1 Sector Write Permission SA28

Value	Description
0	Code Flash SA28 cannot be programmed or erased.
1	Code Flash SA28 program or erase is permitted.

Bit [21] C2SWP29: Code Flash 1 Sector Write Permission SA29

Value	Description
0	Code Flash SA29 cannot be programmed or erased.
1	Code Flash SA29 program or erase is permitted.

Bit [22] C2SWP30: Code Flash 1 Sector Write Permission SA30

Value	Description
0	Code Flash SA30 cannot be programmed or erased.
1	Code Flash SA30 program or erase is permitted.

Bit [23] C2SWP31: Code Flash 1 Sector Write Permission SA31

Value	Description
0	Code Flash SA31 cannot be programmed or erased.
1	Code Flash SA31 program or erase is permitted.

Bit [24] C2SWP32: Code Flash 1 Sector Write Permission SA32

Value	Description
0	Code Flash SA32 cannot be programmed or erased.
1	Code Flash SA32 program or erase is permitted.

Bit [25] C2SWP33: Code Flash 1 Sector Write Permission SA33

Value	Description
0	Code Flash SA33 cannot be programmed or erased.
1	Code Flash SA33 program or erase is permitted.

Bit [26] C2SWP34: Code Flash 1 Sector Write Permission SA34

Value	Description
0	Code Flash SA34 cannot be programmed or erased.
1	Code Flash SA34 program or erase is permitted.

Bit [27] C2SWP35: Code Flash 1 Sector Write Permission SA35

Value	Description
0	Code Flash SA35 cannot be programmed or erased.
1	Code Flash SA35 program or erase is permitted.

Bit [28] C2SWP36: Code Flash 1 Sector Write Permission SA36

Value	Description
0	Code Flash SA36 cannot be programmed or erased.
1	Code Flash SA36 program or erase is permitted.

Bit [29] C2SWP37: Code Flash 1 Sector Write Permission SA37

Value	Description
0	Code Flash SA37 cannot be programmed or erased.
1	Code Flash SA37 program or erase is permitted.

Bit [30] C2SWP38: Code Flash 1 Sector Write Permission SA38

Value	Description
0	Code Flash SA38 cannot be programmed or erased.
1	Code Flash SA38 program or erase is permitted.

Bit [31] C2SWP39: Code Flash 1 Sector Write Permission SA39

Value	Description
0	Code Flash SA39 cannot be programmed or erased.
1	Code Flash SA39 program or erase is permitted.

4.11. Code Flash 2 Sector Write Permissions of the Large Sectors Register (TCFCFG0_C3SWP)

This section describes the function of the Code Flash 2 Sector Write Permission Register of the large sectors.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	C3SWP39	C3SWP38	C3SWP37	C3SWP36	C3SWP35	C3SWP34	C3SWP33	C3SWP32
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	WP							
INITIAL_VALUE	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	C3SWP31	C3SWP30	C3SWP29	C3SWP28	C3SWP27	C3SWP26	C3SWP25	C3SWP24
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	WP							
INITIAL_VALUE	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	C3SWP23	C3SWP22	C3SWP21	C3SWP20	C3SWP19	C3SWP18	C3SWP17	C3SWP16
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	WP							
INITIAL_VALUE	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	C3SWP15	C3SWP14	C3SWP13	C3SWP12	C3SWP11	C3SWP10	C3SWP9	C3SWP8
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	WP							
INITIAL_VALUE	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1

After each RSTX_PD2H the Code Flash 2 Sector Write Permissions of the large sectors (C3SWP) are transferred from Flash into the Code Flash 2 Sector Write Permissions Register of the large sectors (TCFCFG0_C3SWP).

TCFCFG0_C3SWP can be written by the application, when TCFCFG0_SWPOER indicates that overwrite is enabled. Otherwise the write access will cause a bus error and the permission set is fixed.

Notes:

- *Single ECC errors are corrected. When double ECC errors are detected, the register is not updated from Flash and the sector write permissions are not set.*
- *Sector write permissions are ignored in non-USERMODE.*
- *If any of sector permission is set to disabled, macro erase is disabled.*

Bit [0] C3SWP8: Code Flash 2 Sector Write Permission SA8

Value	Description
0	Code Flash SA8 cannot be programmed or erased.
1	Code Flash SA8 program or erase is permitted.

Bit [1] C3SWP9: Code Flash 2 Sector Write Permission SA9

Value	Description
0	Code Flash SA9 cannot be programmed or erased.
1	Code Flash SA9 program or erase is permitted.

Bit [2] C3SWP10: Code Flash 2 Sector Write Permission SA10

Value	Description
0	Code Flash SA10 cannot be programmed or erased.
1	Code Flash SA10 program or erase is permitted.

Bit [3] C3SWP11: Code Flash 2 Sector Write Permission SA11

Value	Description
0	Code Flash SA11 cannot be programmed or erased.
1	Code Flash SA11 program or erase is permitted.

Bit [4] C3SWP12: Code Flash 2 Sector Write Permission SA12

Value	Description
0	Code Flash SA12 cannot be programmed or erased.
1	Code Flash SA12 program or erase is permitted.

Bit [5] C3SWP13: Code Flash 2 Sector Write Permission SA13

Value	Description
0	Code Flash SA13 cannot be programmed or erased.
1	Code Flash SA13 program or erase is permitted.

Bit [6] C3SWP14: Code Flash 2 Sector Write Permission SA14

Value	Description
0	Code Flash SA14 cannot be programmed or erased.
1	Code Flash SA14 program or erase is permitted.

Bit [7] C3SWP15: Code Flash 2 Sector Write Permission SA15

Value	Description
0	Code Flash SA15 cannot be programmed or erased.
1	Code Flash SA15 program or erase is permitted.

Bit [8] C3SWP16: Code Flash 2 Sector Write Permission SA16

Value	Description
0	Code Flash SA16 cannot be programmed or erased.
1	Code Flash SA16 program or erase is permitted.

Bit [9] C3SWP17: Code Flash 2 Sector Write Permission SA17

Value	Description
0	Code Flash SA17 cannot be programmed or erased.
1	Code Flash SA17 program or erase is permitted.

Bit [10] C3SWP18: Code Flash 2 Sector Write Permission SA18

Value	Description
0	Code Flash SA18 cannot be programmed or erased.
1	Code Flash SA18 program or erase is permitted.

Bit [11] C3SWP19: Code Flash 2 Sector Write Permission SA19

Value	Description
0	Code Flash SA19 cannot be programmed or erased.
1	Code Flash SA19 program or erase is permitted.

Bit [12] C3SWP20: Code Flash 2 Sector Write Permission SA20

Value	Description
0	Code Flash SA20 cannot be programmed or erased.
1	Code Flash SA20 program or erase is permitted.

Bit [13] C3SWP21: Code Flash 2 Sector Write Permission SA21

Value	Description
0	Code Flash SA21 cannot be programmed or erased.
1	Code Flash SA21 program or erase is permitted.

Bit [14] C3SWP22: Code Flash 2 Sector Write Permission SA22

Value	Description
0	Code Flash SA22 cannot be programmed or erased.
1	Code Flash SA22 program or erase is permitted.

Bit [15] C3SWP23: Code Flash 2 Sector Write Permission SA23

Value	Description
0	Code Flash SA23 cannot be programmed or erased.
1	Code Flash SA23 program or erase is permitted.

Bit [16] C3SWP24: Code Flash 2 Sector Write Permission SA24

Value	Description
0	Code Flash SA24 cannot be programmed or erased.
1	Code Flash SA24 program or erase is permitted.

Bit [17] C3SWP25: Code Flash 2 Sector Write Permission SA25

Value	Description
0	Code Flash SA25 cannot be programmed or erased.
1	Code Flash SA25 program or erase is permitted.

Bit [18] C3SWP26: Code Flash 2 Sector Write Permission SA26

Value	Description
0	Code Flash SA26 cannot be programmed or erased.
1	Code Flash SA26 program or erase is permitted.

Bit [19] C3SWP27: Code Flash 2 Sector Write Permission SA27

Value	Description
0	Code Flash SA27 cannot be programmed or erased.
1	Code Flash SA27 program or erase is permitted.

Bit [20] C3SWP28: Code Flash 2 Sector Write Permission SA28

Value	Description
0	Code Flash SA28 cannot be programmed or erased.
1	Code Flash SA28 program or erase is permitted.

Bit [21] C3SWP29: Code Flash 2 Sector Write Permission SA29

Value	Description
0	Code Flash SA29 cannot be programmed or erased.
1	Code Flash SA29 program or erase is permitted.

Bit [22] C3SWP30: Code Flash 2 Sector Write Permission SA30

Value	Description
0	Code Flash SA30 cannot be programmed or erased.
1	Code Flash SA30 program or erase is permitted.

Bit [23] C3SWP31: Code Flash 2 Sector Write Permission SA31

Value	Description
0	Code Flash SA31 cannot be programmed or erased.
1	Code Flash SA31 program or erase is permitted.

Bit [24] C3SWP32: Code Flash 2 Sector Write Permission SA32

Value	Description
0	Code Flash SA32 cannot be programmed or erased.
1	Code Flash SA32 program or erase is permitted.

Bit [25] C3SWP33: Code Flash 2 Sector Write Permission SA33

Value	Description
0	Code Flash SA33 cannot be programmed or erased.
1	Code Flash SA33 program or erase is permitted.

Bit [26] C3SWP34: Code Flash 2 Sector Write Permission SA34

Value	Description
0	Code Flash SA34 cannot be programmed or erased.
1	Code Flash SA34 program or erase is permitted.

Bit [27] C3SWP35: Code Flash 2 Sector Write Permission SA35

Value	Description
0	Code Flash SA35 cannot be programmed or erased.
1	Code Flash SA35 program or erase is permitted.

Bit [28] C3SWP36: Code Flash 2 Sector Write Permission SA36

Value	Description
0	Code Flash SA36 cannot be programmed or erased.
1	Code Flash SA36 program or erase is permitted.

Bit [29] C3SWP37: Code Flash 2 Sector Write Permission SA37

Value	Description
0	Code Flash SA37 cannot be programmed or erased.
1	Code Flash SA37 program or erase is permitted.

Bit [30] C3SWP38: Code Flash 2 Sector Write Permission SA38

Value	Description
0	Code Flash SA38 cannot be programmed or erased.
1	Code Flash SA38 program or erase is permitted.

Bit [31] C3SWP39: Code Flash 2 Sector Write Permission SA39

Value	Description
0	Code Flash SA39 cannot be programmed or erased.
1	Code Flash SA39 program or erase is permitted.

CHAPTER 25: Memory Protection Unit for AXI of SHE

This chapter explains the functions and operations of the Memory Protection Unit for AMBA AXI protocol bus (MPU AXI).

1. Overview
2. Configuration and Block Diagram
3. Operation of the MPU AXI
4. Registers
5. Notes on Using MPU AXI

MPUAXI-TXXPT03P01R01L06-E1-XX

1. Overview

This section describes the features of the MPU AXI.

Features of the MPU AXI

The MPU AXI module monitors the accesses from AXI masters and checks each access against an authorized set of access permissions. Access permissions ('permission attributes' here onwards) are defined by 'access permissions' bits. These bits are explained in Section 'MPU access permissions'. MPU AXI provides eight regions and one background region. Each region has corresponding access permission bits that defines the permission attributes for that particular region. Any unauthorized access to memory space is flagged using Non-Maskable Interrupt. MPU AXI also collects information about the unauthorized bus access and stores it in its internal registers.

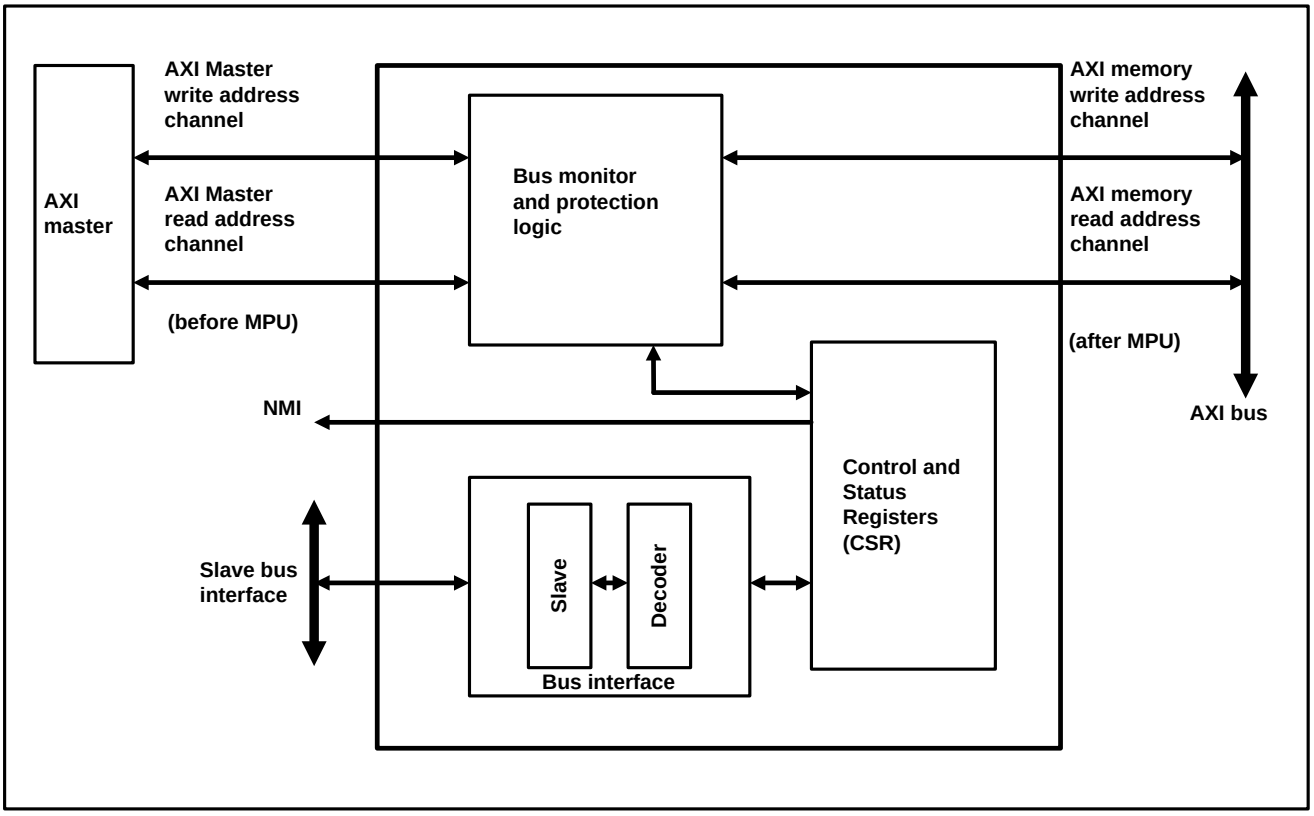
- Each of the eight regions in MPU AXI is specified using corresponding start address and end address
- Background region covers entire 4 GB address space
- On unauthorized access MPU AXI generates an NMI to the CPU
- MPU AXI collects information about the AXI master bus access that caused unauthorized access
- Supports 8-bit, 16-bit, and 32-bit bus accesses for configuration of registers in MPU AXI
- Supports lock, unlock feature for protection of registers from illegal write accesses
- Modification of registers in MPU AXI is only allowed in privileged mode
- Optionally supports privileged mode overwrite feature that allows overwrite of privilege attribute of memory side AXI interface

2. Configuration and Block Diagram

This section shows a block diagram of MPU AXI.

■ Block Diagram of MPU AXI

Figure 2-2-1 Block Diagram of MPU AXI



■ Bus Interface

The bus masters can access the MPU AXI module through its slave bus interface.

■ Bus Monitor and Protection Logic

This logic monitors the transaction on AXI master interfaces. It finds out the region/s where the current transfer belongs to and then applies permissions based on the region match. It signals any permission violation to CSR logic that in turn generates NMI interrupt. All transactions on AXI master interfaces (including the transfer that caused permission violation) are blocked until NMI is cleared.

■ Control and Status Registers

The operation of MPU AXI can be controlled and monitored through its Control and Status Registers (CSR)

3. Operation of the MPU AXI

This section describes the operation of MPU AXI.

MPU AXI provides start address and end address for each of the eight regions. MPU AXI regions are defined with granularity of 128 bytes.

Start address specifies the first address of the region and is specified by registers MPUXSHE_SADDR1 to MPUXSHE_SADDR8 for region 1 to region 8 respectively. Since the region granularity is 128 bytes least significant 7 bits of start addresses will read 0.

End addresses are specified by registers MPUXSHE_EADDR1 to MPUXSHE_EADDR8 for region 1 to region 8 respectively. Least significant 7 bits of End Address Registers are read-only bits and will always read "1". This ensures the granularity of 128 bytes.

■ AXI Burst Monitoring

Bus monitor and protection logic monitors AXI master 'write address channel' signals and 'read address channel' signals ('AXI master interfaces' here onwards). The transactions on AXI master interfaces are manipulated (if required) before they are passed on to AXI memory 'write address channel' signals and 'read address channel' signals ('AXI memory interfaces' here onwards).

AXI protocol supports following features

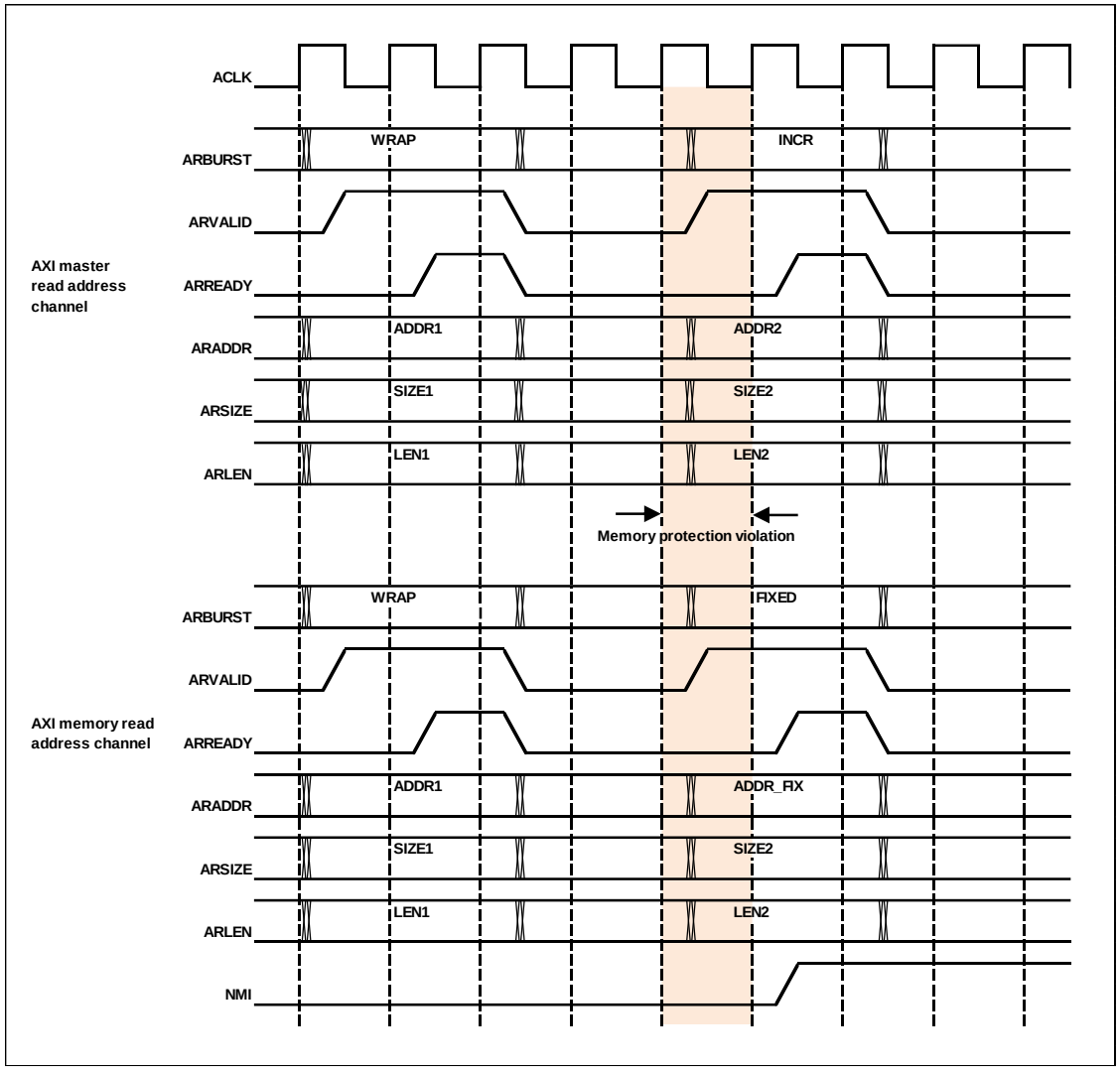
- Separate address or control and data phases
- Burst based transaction where only start address is issued
- Separate write and read address/control channels
- Separate write and read data channels

AXI master begins each burst by driving transaction control information and address of the first byte in the transaction. As the burst transaction progresses, AXI slave calculates the addresses of subsequent transfers in the burst. The AWLEN (write address channel signal), ARLEN (read address channel signal) specifies the number of data transfers for a burst transaction. Each burst can be of 1 to 16 transfers long. The AWSIZE, ARSIZE signals specifies the maximum number of data transfers in terms of bytes in each data transfer within a burst. The AXI protocol defines FIXED, incrementing and wrapping burst types.

As the addresses are defined in separate channels than data channels MPU AXI monitors and controls only address channels of AXI. If memory protection violation is detected entire burst transaction is manipulated to FIXED address burst with address of predefined value.

Figure 3-3-1 shows timings for read address channel signals. First burst is of WRAP type for which no memory protection violation was detected. Second burst is of INCR type for which memory protection violation was detected and hence MPU AXI manipulates the ARBURST to FIXED type and ARADDR to predefined value (shown as ADDR_FIX).

Figure 3-3-1 MPU AXI Example Timings



MPU AXI uses burst start address (AWADDR, ARADDR signals), burst type (AWBURST, ARBURST signals), burst length (AWLEN, ARLEN signals), and burst size (AWSIZE, ARSIZE signals) from AXI burst transaction to calculate the lowest address and the highest address of the burst and then both the addresses are used to find the region match.

For FIXED type of burst:

Lowest address = Highest address = Burst start address

For INCR type of burst:

Lowest address = Burst start address

Highest address = Burst start address + (num_bytes x num_transfers) – 1

where, num_bytes is number of bytes in each transfer of burst and

num_transfers is number of transfers in the burst

For WRAP type of burst:

Lowest address = Wrap boundary

Highest address = Wrap boundary + (num_bytes x num_transfers) – 1

MPU Access Permissions

Region control registers in MPU AXI, MPUXSHE_CTRL1 to MPUXSHE_CTRL8 are used to control the access permission for region 1 to region 8 respectively. Also MPUXSHE_CTRL0 is used to control the access permission for background region.

Table 3-1 Access Permissions

AP Bits	Access in Privileged Mode	Access in Non- Privileged Mode	Comment
0b000 (default)	No access	No access	All bus accesses are blocked and hence would generate memory protection violation.
0b001	read, write	No access	Reads and writes are permitted in privileged mode only. Any access in non- privileged mode would generate memory protection violation.
0b010	read, write	read only	Reads and writes are permitted in privileged mode. Only reads are permitted in non-privileged mode. Writes in non- privileged mode would generate memory protection violation.
0b011	read, write	read, write	All bus transfers are permitted. No memory protection violation is generated in this mode.
0b100	No access	No access	All bus accesses are blocked and hence generate memory protection violation.
0b101	read only	No access	Reads are permitted in privileged mode only. Writes in privileged mode and any access in non-privileged mode would generate memory protection violation.
0b110	read only	read only	Reads are permitted in privileged as well as non-privileged mode. Any write access would generate memory protection violation.
0b111	read, write	read, write	All bus transfers are permitted. No memory protection violation is generated in this mode.

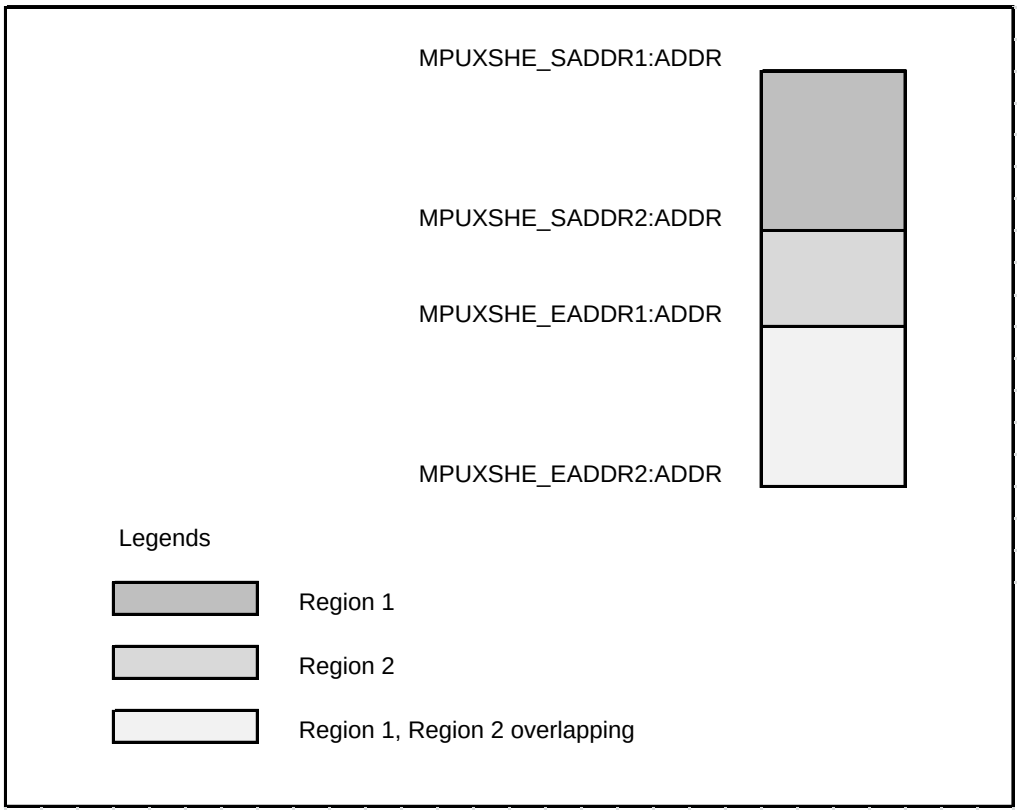
MPU Priority Decision

MPU AXI provides start address and end address for each region. Start address specifies the first address of the region and end address specifies the last address of the region. It is allowed in MPU AXI that the region may overlap each other. Hence it is also possible that an AXI transaction address may match with multiple regions.

For AXI protocol transaction following scenarios are possible.

1. Lowest address and highest address might fall in the same region.
2. Lowest address might match in one region, but highest address might match in some other region.
3. Scenario mentioned in point 1 here, happens for multiple regions due to region overlapping.
4. Scenario mentioned in point 2 here, happens for multiple regions due to region overlapping.

Figure 3-3-2 Example Region Overlapping



MPU AXI finds region match signals based on lowest address of AXI burst transaction and also separate region match signals based on highest address. Due to overlapping of regions it is possible lowest address and/or highest address may match with multiple regions.

MPU AXI supports upto eight regions. Each region is identified as region 1, region 2, region 3, and so on upto region 8. Region 8 is given highest priority, region 7 has second highest priority, and so on with region 1 as the second lowest priority region. Background region is the lowest priority region.

Whenever the AXI transaction lowest address matches with multiple regions the permissions corresponding to the highest priority (among all matching region) region is applied. Similarly, if AXI transaction highest address matches with multiple regions the permissions corresponding to highest priority (among all matching region) region is applied.

It is possible that lowest address and highest address may match with different highest priority regions. In this case the more restrictive permission is applied. Table 3-2 shows access permissions with corresponding restriction index for privileged mode access. Restriction index 1 has the most restrictive access and access permissions with restriction index 3 has the least restrictive access. This table is applied when privileged mode attribute of current transaction is of type privileged mode.

Similarly, Table 3-3 shows access permissions with corresponding restriction index for non- privileged mode access. This table is applied when privileged mode attribute of current transaction is of type non-privileged mode.

Table 3-2 Restrictive Access Permission Matrix for Privileged Mode Access

AP Bits	Access in Privileged Mode	Restriction Index
0b000, 0b100	No access	1
0b101, 0b110	read only	2
0b001, 0b010, 0b011, 0b111	read, write	3

Table 3-3 Restrictive Access Permission Matrix for Non-Privileged Mode Access

AP Bits	Access in Non-Privileged Mode	Restriction Index
0b000, 0b001, 0b100, 0b101	No access	1
0b010, 0b110	read only	2
0b011, 0b111	read, write	3

Bus Monitor and Protection Logic

All transactions on the AXI master interfaces are monitored and checked for permitted access.

- Bus monitor and protection logic within MPU AXI compares the lowest address and highest address of current transaction with start addresses and end addresses of each region to find region match where the current transaction matches to among the eight defined regions
- As explained in Section MPU priority decision the AXI transaction address may match with multiple regions. The permission attributes of highest priority region are checked against the attributes of currently applied transaction from AXI master
- If the attributes of currently applied transaction are within permitted attributes, current transaction is passed on to the AXI memory interfaces
- If the attributes are not within permitted attributes current transaction is blocked. The Non Maskable Interrupt (MPUXSHE_CTRL0:NMI) flag is set. If the memory protection violation is detected on write address channel, the address and control information is stored in MPUXSHE_WERRA and MPUXSHE_WERRC registers respectively. If the memory protection violation is detected on read address channel, the address and control information is stored in MPUXSHE_RERRA and MPUXSHE_RERRC registers respectively. It is possible that memory protection violation is detected simultaneously on both the channels
- All further transaction are blocked until the MPUXSHE_CTRL0:NMI flag is cleared by software. Further monitoring of AXI master interfaces is also stalled until the MPUXSHE_CTRL0:NMI flag is cleared

When a transfer is blocked, MPU AXI does the following actions:

- Current transaction on AXI master is manipulated to FIXED address burst
- The transaction address is changed to predefined address. Check device specific datasheet for details of predefined address

Privileged Mode Overwrite Feature

Optionally MPU AXI supports privileged mode overwrite feature.

When this mode is enabled privileged mode attribute on the AXI memory interfaces are set by register bit setting MPUXSHE_CTRL0:PROT.

Note:

- *Bus monitor and protection logic for detecting memory protection violation uses privilege mode attribute on AXI master interfaces and not MPUXSHE_CTRL0:PROT bit setting even when privileged mode overwrite feature is enabled.*

4. Registers

This section describes the registers of MPU AXI.

The MPU AXI module contains various registers to configure its operation, to monitor its status and to read the information it has collected from the AXI master interfaces at the time of the memory protection violation.

The MPU AXI module is allocated 1 KB of MCU address space for mapping the Configuration and Status Registers (i.e. CSRs). The address area allocated to MPU AXI and the Control and Status Registers in MPU AXI are explained in this section.

The suffix 'n' in the register name indicates that the register is an instance 'n' of the module.

■ Registers of MPU AXI

The following registers are available for MPU AXI:

- MPU AXI Control Register (MPUXSHE_CTRL0)
- MPU AXI NMI Enable Register (MPUXSHE_NMIEN)
- MPU AXI Write Error Control Register (MPUXSHE_WERRC)
- MPU AXI Write Error Address Register (MPUXSHE_WERRA)
- MPU AXI Read Error Control Register (MPUXSHE_RERRC)
- MPU AXI Read Error Address Register (MPUXSHE_RERRA)
- MPU AXI Region Control Registers (MPUXSHE_CTRL1~8)
- MPU AXI Start Address Registers (MPUXSHE_SADDR1~8)
- MPU AXI End Address Registers (MPUXSHE_EADDR1~8)
- MPU AXI Unlock Register (MPUXSHE_UNLOCK)
- MPU AXI Module ID Register (MPUXSHE_MID)

■ Memory Layout of MPU AXI Registers

Offset	+3	+2	+1	+0
0x00000000	MPUXSHE_CTRL0 00000000 00000000 00000001 00000000			
0x00000004	MPUXSHE_NMIEN 00000000 00000000 00000000 00000001			
0x00000008	MPUXSHE_WERRC 00000000 00000000 00000XXX XXXXXXX0			
0x0000000C	MPUXSHE_WERRA XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x00000010	MPUXSHE_RERRC 00000000 00000000 00000XXX XXXXXXX0			
0x00000014	MPUXSHE_RERRA XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x00000018	MPUXSHE_CTRL1 00000000 00000000 00000000 00000000			

Offset	+3	+2	+1	+0
0x0000001C	MPUXSHE_SADDR1 00000000 00000000 00000000 00000000			
0x00000020	MPUXSHE_EADDR1 00000000 00000000 00000000 01111111			
0x00000024	MPUXSHE_CTRL2 00000000 00000000 00000000 00000000			
0x00000028	MPUXSHE_SADDR2 00000000 00000000 00000000 00000000			
0x0000002C	MPUXSHE_EADDR2 00000000 00000000 00000000 01111111			
0x00000030	MPUXSHE_CTRL3 00000000 00000000 00000000 00000000			
0x00000034	MPUXSHE_SADDR3 00000000 00000000 00000000 00000000			
0x00000038	MPUXSHE_EADDR3 00000000 00000000 00000000 01111111			
0x0000003C	MPUXSHE_CTRL4 00000000 00000000 00000000 00000000			
0x00000040	MPUXSHE_SADDR4 00000000 00000000 00000000 00000000			
0x00000044	MPUXSHE_EADDR4 00000000 00000000 00000000 01111111			
0x00000048	MPUXSHE_CTRL5 00000000 00000000 00000000 00000000			
0x0000004C	MPUXSHE_SADDR5 00000000 00000000 00000000 00000000			
0x00000050	MPUXSHE_EADDR5 00000000 00000000 00000000 01111111			
0x00000054	MPUXSHE_CTRL6 00000000 00000000 00000000 00000000			
0x00000058	MPUXSHE_SADDR6 00000000 00000000 00000000 00000000			
0x0000005C	MPUXSHE_EADDR6 00000000 00000000 00000000 01111111			
0x00000060	MPUXSHE_CTRL7 00000000 00000000 00000000 00000000			
0x00000064	MPUXSHE_SADDR7 00000000 00000000 00000000 00000000			
0x00000068	MPUXSHE_EADDR7 00000000 00000000 00000000 01111111			
0x0000006C	MPUXSHE_CTRL8 00000000 00000000 00000000 00000000			

Offset	+3	+2	+1	+0
0x00000070	MPUXSHE_SADDR8 00000000 00000000 00000000 00000000			
0x00000074	MPUXSHE_EADDR8 00000000 00000000 00000000 01111111			
0x00000078	MPUXSHE_UNLOCK 00000000 00000000 00000000 00000000			
0x0000007C	MPUXSHE_MID 00000000 00001101 00000010 00000000			

4.1. MPU AXI Control Register (MPUXSHE_CTRL0)

MPU AXI Control Register can be used by the software to configure the MPU AXI. This register provides a bit to enable the MPU AXI monitoring and protection function. It also provides permission attributes for background region. It also provides controls for enabling or disabling of privileged mode overwrite feature. Lastly it provides MPU AXI lock status and controls the status for Non-Maskable Interrupt.

■ MPUXSHE_CTRL0

bit	31	30	29	28	27	26	25	24
Field	Reserved	Reserved	Reserved	Reserved	Reserved	AP[2]	AP[1]	AP[0]
R/W Attribute	R0	R0	R0	R0	R0	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	WPS	WPS	WPS
Initial Value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	MPUENC	MPUEN
R/W Attribute	R0	R0	R0	R0	R0	R0	R/W	R
Protection Attribute	-	-	-	-	-	-	WPS	-
Initial Value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved	Reserved	PROT	POEN	Reserved	Reserved	LST
R/W Attribute	R0	R0	R0	R/W	R/W	R0	R0	R
Protection Attribute	-	-	-	WPS	WPS	-	-	-
Initial Value	0	0	0	0	0	0	0	1

bit	7	6	5	4	3	2	1	0
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	NMICL	NMI
R/W Attribute	R0	R0	R0	R0	R0	R0	R0,W	R
Protection Attribute	-	-	-	-	-	-	WPS	-
Initial Value	0	0	0	0	0	0	0	0

Bit Position	Bit Field Name	Bit Description
[31:27]	Reserved	-
[26:24]	AP	Access Permissions for Background Region These bits are used to control access permissions for background region. For detailed description of these bits refer to Table 'Access Permissions'.
[23:18]	Reserved	-
[17]	MPUENC	MPU AXI Enable Control "0": MPU AXI monitoring and protection function is disabled "1": MPU AXI monitoring and protection function is enabled Read returns status of MPUENC bit. The region enable status to be effective takes some delay with reference to write on MPUENC bit. The actual enable status can be read through MPUXSHE_CTRL0:MPUEN bit.
[16]	MPUEN	MPU AXI Enable Status "0": MPU AXI monitoring and protection function is disabled. All accesses on AXI master write or read address channel interfaces are passed on to AXI memory write or read address channel interfaces without any protection "1": MPU AXI monitoring and protection function is enabled
[15:13]	Reserved	-
[12]	PROT	Privilege Attribute When privileged mode overwrite feature is available and also MPUXSHE_CTRL0:POEN = "1", privilege mode attribute on AXI memory interfaces are controlled by this bit. "0": Non-privilege mode "1": Privilege mode
[11]	POEN	Privileged Mode Overwrite Feature Enable "0": Privileged mode overwrite feature is disabled "1": Privileged mode overwrite feature is enabled Note: For availability of privileged mode overwrite feature, refer to device specific datasheet.
[10:9]	Reserved	-
[8]	LST	MPU Lock Status "0": MPU AXI is unlocked, registers in MPU AXI can be written "1": MPU AXI is locked, no registers (other than MPUXSHE_UNLOCK register) in MPU AXI can be written
[7:2]	Reserved	-
[1]	NMICL	NMI Interrupt Clear "0": No effect "1": Clears the NMI interrupt flag Read returns "0".

Bit Position	Bit Field Name	Bit Description
[0]	NMI	NMI Interrupt Flag This bit indicates that the memory protection violation is detected for an AXI transaction.

Note:

- *The register bits MPUXSHE_CTRL0:AP, MPUXSHE_CTRL0:MPUSTOPEN, MPUXSHE_CTRL0:POEN and MPUXSHE_CTRL0:PROT in MPUXSHE_CTRL0 register can only be written when MPU is disabled (MPUXSHE_CTRL0:MPUEN = "0"). This also implies that when MPUXSHE_CTRL0:MPUEN = "1":*
  - *32-bit or 16-bit access to this register is not allowed*
  - *8-bit access is required to disable the MPU*
  - *8-bit access is required to clear the NMI interrupt flag*

4.2. MPU AXI NMI Enable Register (MPUXSHE_NMIEN)

MPU AXI NMI Enable Register can be used by software to reset NMI enable bit. Default value of NMI enable bit is "1". This bit can be reset by software only once after reset operation.

■ **MPUXSHE_NMIEN**

bit	31	30	29	28	27	26	25	24
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W Attribute	R0	R0	R0	R0	R0	R0	R0	R0
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W Attribute	R0	R0	R0	R0	R0	R0	R0	R0
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W Attribute	R0	R0	R0	R0	R0	R0	R0	R0
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	NMIEN
R/W Attribute	R0	R0	R0	R0	R0	R0	R0	R/W
Protection Attribute	-	-	-	-	-	-	-	WPS
Initial Value	0	0	0	0	0	0	0	1

Bit Position	Bit Field Name	Bit Description
[31:8]	Reserved	-
[7:1]	Reserved	-
[0]	NMIEN	NMI Interrupt Enable This bit decides whether the NMI interrupt flag is routed to NMI interrupt signal or not. "0": NMI interrupt flag does not trigger NMI interrupt signal "1": NMI interrupt flag triggers NMI interrupt signal The value of this bit can be changed only once after reset.

4.3. MPU AXI Write Error Control Register (MPUXSHE_WERRC)

This is a read-only register that provides the software the control information of AXI transaction on AXI master write address channel interface for which memory protection violation was detected. Software can read this register to get privileged mode, burst length, burst size, and burst type information of AXI transaction.

■ MPUXSHE_WERRC

bit	31	30	29	28	27	26	25	24
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W Attribute	R0	R0	R0	R0	R0	R0	R0	R0
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W Attribute	R0	R0	R0	R0	R0	R0	R0	R0
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved	Reserved	Reserved	Reserved	AWSIZE[2]	AWSIZE[1]	AWSIZE[0]
R/W Attribute	R0	R0	R0	R0	R0	R	R	R
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	X	X	X

bit	7	6	5	4	3	2	1	0
Field	AWBURST[1]	AWBURST[0]	AWLEN[3]	AWLEN[2]	AWLEN[1]	AWLEN[0]	AWPROTPRIV	AWMPV
R/W Attribute	R	R	R	R	R	R	R	R
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	X	X	X	X	X	X	X	0

Bit Position	Bit Field Name	Bit Description
[31:16]	Reserved	-
[15:11]	Reserved	-
[10:8]	AWSIZE	AXI Transaction Burst Size This bit provides the status of AWSIZE[2:0] signals of AXI transaction for which memory protection violation is detected.
[7:6]	AWBURST	AXI Transaction Burst Type This bit provides the status of AWBURST[1:0] signals of AXI transaction for which memory protection violation is detected.
[5:2]	AWLEN	AXI Transaction Burst Length This bit provides the status of AWLEN[3:0] signals of AXI transaction for which memory protection violation is detected.
[1]	AWPROTPRIV	AXI Transaction Privileged Mode This bit provides the status of AWPROT[0] signal of AXI transaction for which memory protection violation is detected.

Bit Position	Bit Field Name	Bit Description
[0]	AWMPV	AXI Write Memory Protection Violation This bit indicates that memory protection violation is detected on AXI write address channel. Writing "1" to MPUXSHE_CTRL0:NMICL bit clears AWMPV bit.

4.4. MPU AXI Write Error Address Register (MPUXSHE_WERRA)

This is a read-only register that provides the software the write address of AXI transaction on AXI master write address channel interface for which memory protection violation was detected.

■ **MPUXSHE_WERRA**

bit	31	30	29	28	27	26	25	24
Field	AWADDR [31]	AWADDR [30]	AWADDR [29]	AWADDR [28]	AWADDR [27]	AWADDR [26]	AWADDR [25]	AWADDR [24]
R/W Attribute	R	R	R	R	R	R	R	R
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	X	X	X	X	X	X	X	X

bit	23	22	21	20	19	18	17	16
Field	AWADDR [23]	AWADDR [22]	AWADDR [21]	AWADDR [20]	AWADDR [19]	AWADDR [18]	AWADDR [17]	AWADDR [16]
R/W Attribute	R	R	R	R	R	R	R	R
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	X	X	X	X	X	X	X	X

bit	15	14	13	12	11	10	9	8
Field	AWADDR [15]	AWADDR [14]	AWADDR [13]	AWADDR [12]	AWADDR [11]	AWADDR [10]	AWADDR[9]	AWADDR[8]
R/W Attribute	R	R	R	R	R0	R	R	R
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	X	X	X	X	X	X	X	X

bit	7	6	5	4	3	2	1	0
Field	AWADDR[7]	AWADDR[6]	AWADDR[5]	AWADDR[4]	AWADDR[3]	AWADDR[2]	AWADDR[1]	AWADDR[0]
R/W Attribute	R	R	R	R	R	R	R	R
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	X	X	X	X	X	X	X	X

Bit Position	Bit Field Name	Bit Description
[31:0]	AWADDR	AXI Write Address Write address of AXI transaction for which memory protection violation is detected.

4.5. MPU AXI Read Error Control Register (MPUXSHE_RERRC)

This is a read-only register that provides the software the control information of AXI transaction on AXI master read address channel interface for which memory protection violation was detected. Software can read this register to get privileged mode, burst length, burst size, and burst type information of AXI transaction.

■ MPUXSHE_RERRC

bit	31	30	29	28	27	26	25	24
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W Attribute	R0	R0	R0	R0	R0	R0	R0	R0
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W Attribute	R0	R0	R0	R0	R0	R0	R0	R0
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved	Reserved	Reserved	Reserved	ARSIZE[2]	ARSIZE[1]	ARSIZE[0]
R/W Attribute	R0	R0	R0	R0	R0	R	R	R
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	X	X	X

bit	7	6	5	4	3	2	1	0
Field	ARBURST[1]	ARBURST[0]	ARLEN[3]	ARLEN[2]	ARLEN[1]	ARLEN[0]	ARPROTPRIV	ARMPV
R/W Attribute	R	R	R	R	R	R	R	R
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	X	X	X	X	X	X	X	0

Bit Position	Bit Field Name	Bit Description
[31:16]	Reserved	-
[15:11]	Reserved	-
[10:8]	ARSIZE	AXI Transaction Burst Size This bit provides the status of ARSIZE[2:0] signals of AXI transaction for which memory protection violation is detected.
[7:6]	ARBURST	AXI Transaction Burst Type This bit provides the status of ARBURST[1:0] signals of AXI transaction for which memory protection violation is detected.
[5:2]	ARLEN	AXI Transaction Burst Length This bit provides the status of ARLEN[3:0] signals of AXI transaction for which memory protection violation is detected.
[1]	ARPROTPRIV	AXI Transaction Privileged Mode This bit provides the status of ARPROT[0] signal of AXI transaction for which memory protection violation is detected.

Bit Position	Bit Field Name	Bit Description
[0]	ARMPV	AXI Read Memory Protection Violation This bit indicates that memory protection violation is detected on AXI read address channel. Writing "1" to MPUXSHE_CTRL0:NMICL bit clears ARMPV bit.

4.6. MPU AXI Read Error Address Register (MPUXSHE_RERRA)

This is a read-only register that provides the software the read address of AXI transaction on AXI master read address channel interface for which memory protection violation was detected.

■ **MPUXSHE_RERRA**

bit	31	30	29	28	27	26	25	24
Field	ARADDR [31]	ARADDR [30]	ARADDR [29]	ARADDR [28]	ARADDR [27]	ARADDR [26]	ARADDR [25]	ARADDR [24]
R/W Attribute	R	R	R	R	R	R	R	R
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	X	X	X	X	X	X	X	X

bit	23	22	21	20	19	18	17	16
Field	ARADDR [23]	ARADDR [22]	ARADDR [21]	ARADDR [20]	ARADDR [19]	ARADDR [18]	ARADDR [17]	ARADDR [16]
R/W Attribute	R	R	R	R	R	R	R	R
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	X	X	X	X	X	X	X	X

bit	15	14	13	12	11	10	9	8
Field	ARADDR [15]	ARADDR [14]	ARADDR [13]	ARADDR [12]	ARADDR [11]	ARADDR [10]	ARADDR[9]	ARADDR[8]
R/W Attribute	R	R	R	R	R0	R	R	R
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	X	X	X	X	X	X	X	X

bit	7	6	5	4	3	2	1	0
Field	ARADDR[7]	ARADDR[6]	ARADDR[5]	ARADDR[4]	ARADDR[3]	ARADDR[2]	ARADDR[1]	ARADDR[0]
R/W Attribute	R	R	R	R	R	R	R	R
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	X	X	X	X	X	X	X	X

Bit Position	Bit Field Name	Bit Description
[31:0]	ARADDR	AXI Read Address Read address of AXI transaction for which memory protection violation is detected.

4.7. MPU AXI Region Control Registers (MPUXSHE_CTRL1~8)

MPU AXI Region Control Register is used to specify access permission for a particular region. Software can also use this register for enabling or disabling the particular region. MPUXSHE_CTRL1 Control Register for Region 1 is explained below.

■ MPUXSHE_CTRL1

bit	31	30	29	28	27	26	25	24
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W Attribute	R0	R0	R0	R0	R0	R0	R0	R0
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W Attribute	R0	R0	R0	R0	R0	R0	R0	R0
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved	Reserved	Reserved	Reserved	AP[2]	AP[1]	AP[0]
R/W Attribute	R0	R0	R0	R0	R0	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	WP	WP	WP
Initial Value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	MPUENC	MPUEN
R/W Attribute	R0	R0	R0	R0	R0	R0	R/W	R
Protection Attribute	-	-	-	-	-	-	WPS	-
Initial Value	0	0	0	0	0	0	0	0

Bit Position	Bit Field Name	Bit Description
[31:16]	Reserved	-
[15:11]	Reserved	-
[10:8]	AP	Access Permissions These bits are used to control access permissions for region 1.
[7:2]	Reserved	-

Bit Position	Bit Field Name	Bit Description
[1]	MPUENC	Enable Control bit "0": Memory protection for region 1 is disabled "1": Memory protection for region 1 is enabled Read returns status of MPUENC bit. The region enable status to be effective takes some delay with reference to write on MPUENC bit. The actual enable status can be read through MPUXSHE_CTRL1:MPUEN bit.
[0]	MPUEN	Enable Status "0": Memory protection for region 1 is disabled "1": Memory protection for region 1 is enabled This is a read-only bit, writing to this bit does not have any effect.

Notes:

- *Access permission bits (i.e. MPUXSHE_CTRL1~8:AP) can only be written when corresponding region is disabled (MPUXSHE_CTRL1~8:MPUEN = "0") or MPU is disabled (MPUXSHE_CTRL0:MPUEN = "0").*
- *This also implies that when MPUXSHE_CTRL0:MPUEN = "1" and MPUXSHE_CTRL1~8:MPUEN = "1":*
  - *32-bit or 16-bit access to this register is not allowed*
  - *8-bit access is required to disable the MPU region*

4.8. MPU AXI Start Address Registers (MPUXSHE_SADDR1~8)

Each region in MPU AXI can be set by specifying start address and end address for that particular region. MPUXSHE_SADDR1~8 registers are used to specify start address for eight regions. Start address indicates the first address for that region. MPUXSHE_SADDR1 Start Address Register for Region 1 is explained below.

■ MPUXSHE_SADDR1

bit	31	30	29	28	27	26	25	24
Field	SADDR[31]	SADDR[30]	SADDR[29]	SADDR[28]	SADDR[27]	SADDR[26]	SADDR[25]	SADDR[24]
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	WPS	WPS	WPS	WPS	WPS	WPS	WPS	WPS
Initial Value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	SADDR[23]	SADDR[22]	SADDR[21]	SADDR[20]	SADDR[19]	SADDR[18]	SADDR[17]	SADDR[16]
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	WPS	WPS	WPS	WPS	WPS	WPS	WPS	WPS
Initial Value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	SADDR[15]	SADDR[14]	SADDR[13]	SADDR[12]	SADDR[11]	SADDR[10]	SADDR[9]	SADDR[8]
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	WPS	WPS	WPS	WPS	WPS	WPS	WPS	WPS
Initial Value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	SADDR[7]	SADDR[6]	SADDR[5]	SADDR[4]	SADDR[3]	SADDR[2]	SADDR[1]	SADDR[0]
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	WPS	WPS	WPS	WPS	WPS	WPS	WPS	WPS
Initial Value	0	0	0	0	0	0	0	0

Bit Position	Bit Field Name	Bit Description
[31:0]	SADDR	Start Address Start address for region 1

Note:

- *MPUXSHE_SADDR1~8 registers can only be written when corresponding region is disabled (MPUXSHE_CTRL1~8:MPUEN = "0") or MPU is disabled (MPUXSHE_CTRL0:MPUEN = "0").*

4.9. MPU AXI End Address Registers (MPUXSHE_EADDR1~8)

Each region in MPU AXI can be set by specifying start address and end address for that particular region. MPUXSHE_EADDR1~8 registers are used to specify end addresses for eight regions. End Address indicates the last address for that region. MPUXSHE_EADDR1 End Address Register for Region 1 is explained below.

■ MPUXSHE_EADDR1

bit	31	30	29	28	27	26	25	24
Field	EADDR[31]	EADDR[30]	EADDR[29]	EADDR[28]	EADDR[27]	EADDR[26]	EADDR[25]	EADDR[24]
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	WPS	WPS	WPS	WPS	WPS	WPS	WPS	WPS
Initial Value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	EADDR[23]	EADDR[22]	EADDR[21]	EADDR[20]	EADDR[19]	EADDR[18]	EADDR[17]	EADDR[16]
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	WPS	WPS	WPS	WPS	WPS	WPS	WPS	WPS
Initial Value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	EADDR[15]	EADDR[14]	EADDR[13]	EADDR[12]	EADDR[11]	EADDR[10]	EADDR[9]	EADDR[8]
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	WPS	WPS	WPS	WPS	WPS	WPS	WPS	WPS
Initial Value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	EADDR[7]	EADDR[6]	EADDR[5]	EADDR[4]	EADDR[3]	EADDR[2]	EADDR[1]	EADDR[0]
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	WPS	WPS	WPS	WPS	WPS	WPS	WPS	WPS
Initial Value	0	1	1	1	1	1	1	1

Bit Position	Bit Field Name	Bit Description
[31:0]	EADDR	End Address End address for region 1

Note:

- *MPUXSHE_EADDR1~8 registers can only be written when corresponding region is disabled (MPUXSHE_CTRL1~8:MPUEN = "0") or MPU is disabled (MPUXSHE_CTRL0:MPUEN = "0").*

4.10. MPU AXI Unlock Register (MPUXSHE_UNLOCK)

The software can use this register to lock or unlock the MPU AXI registers for write access.

■ MPUXSHE_UNLOCK

bit	31	30	29	28	27	26	25	24
Field	UNLOCK[31]	UNLOCK[30]	UNLOCK[29]	UNLOCK[28]	UNLOCK[27]	UNLOCK[26]	UNLOCK[25]	UNLOCK[24]
R/W Attribute	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
Protection Attribute	WP	WP	WP	WP	WP	WP	WP	WP
Initial Value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	UNLOCK[23]	UNLOCK[22]	UNLOCK[21]	UNLOCK[20]	UNLOCK[19]	UNLOCK[18]	UNLOCK[17]	UNLOCK[16]
R/W Attribute	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
Protection Attribute	WP	WP	WP	WP	WP	WP	WP	WP
Initial Value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	UNLOCK[15]	UNLOCK[14]	UNLOCK[13]	UNLOCK[12]	UNLOCK[11]	UNLOCK[10]	UNLOCK[9]	UNLOCK[8]
R/W Attribute	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
Protection Attribute	WP	WP	WP	WP	WP	WP	WP	WP
Initial Value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	UNLOCK[7]	UNLOCK[6]	UNLOCK[5]	UNLOCK[4]	UNLOCK[3]	UNLOCK[2]	UNLOCK[1]	UNLOCK[0]
R/W Attribute	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
Protection Attribute	WP	WP	WP	WP	WP	WP	WP	WP
Initial Value	0	0	0	0	0	0	0	0

Bit Position	Bit Field Name	Bit Description
[31:0]	UNLOCK	MPU AXI Unlock The MPU AXI Unlock Register protects the MPU AXI module from being modified accidently by software. The MPU AXI registers cannot be written until this register has been written with a specific unlock value. The correct value for unlocking can be written only in privileged mode. Reading this register always returns a zero. To lock the MPU AXI again software must write another value specific to lock. Write access to MPU AXI registers without unlocking or writing a value other than the lock or unlock values to this register causes a protection error. Notes:

4.11. MPU AXI Module ID Register (MPUXSHE_MID)

This is a read-only register with a unique module identification number which identifies the version of the MPU AXI module used in the MCU.

■ MPUXSHE_MID

bit	31	30	29	28	27	26	25	24
Field	MID[31]	MID[30]	MID[29]	MID[28]	MID[27]	MID[26]	MID[25]	MID[24]
R/W Attribute	R	R	R	R	R	R	R	R
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	MID[23]	MID[22]	MID[21]	MID[20]	MID[19]	MID[18]	MID[17]	MID[16]
R/W Attribute	R	R	R	R	R	R	R	R
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	1	1	0	1

bit	15	14	13	12	11	10	9	8
Field	MID[15]	MID[14]	MID[13]	MID[12]	MID[11]	MID[10]	MID[9]	MID[8]
R/W Attribute	R	R	R	R	R	R	R	R
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	1	0

bit	7	6	5	4	3	2	1	0
Field	MID[7]	MID[6]	MID[5]	MID[4]	MID[3]	MID[2]	MID[1]	MID[0]
R/W Attribute	R	R	R	R	R	R	R	R
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

Bit Position	Bit Field Name	Bit Description
[31:0]	MID	Module ID Always return a meaningless constant.

5. Notes on Using MPU AXI

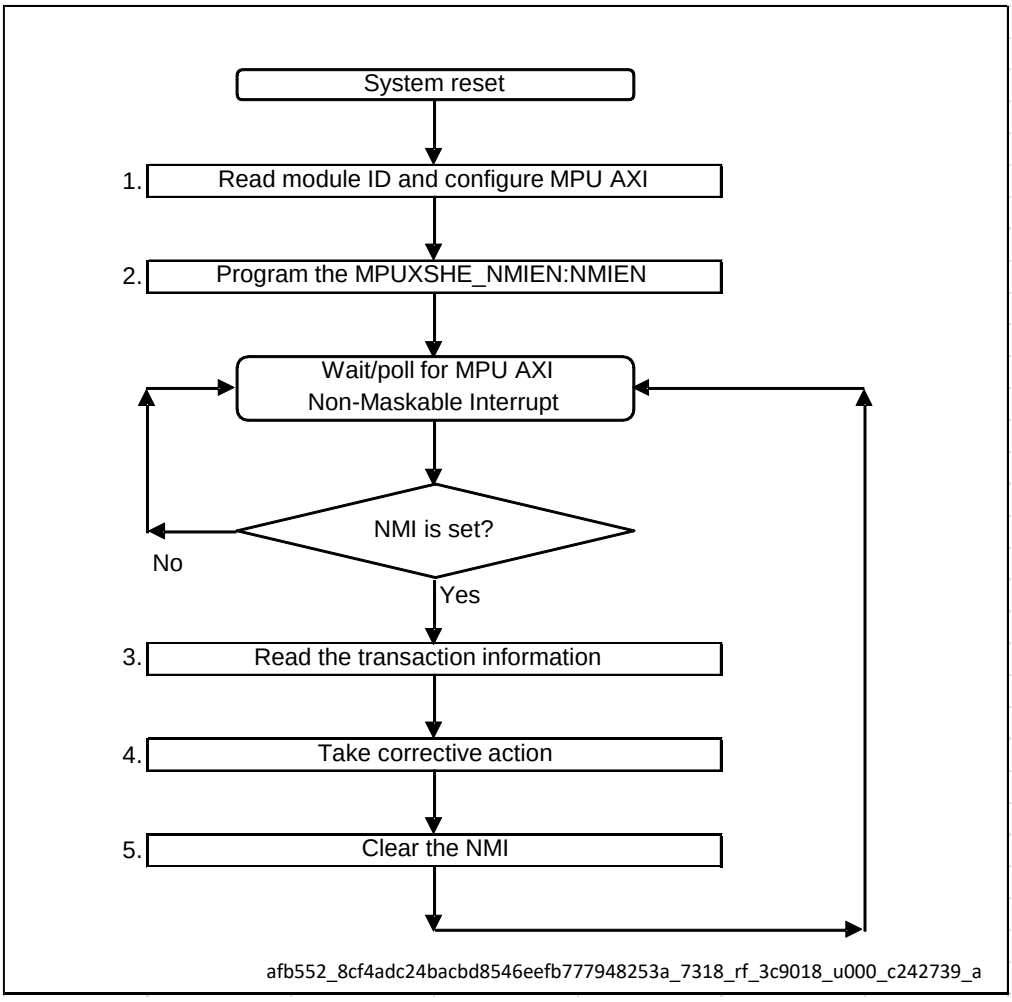
This section is the 'programmer's guide', which lists the usage notes for programming the MPU AXI module. It is recommended to read these guidelines before programming the MPU AXI module.

■ General Usage Notes

- Reserved bits return undefined values. The software programs shall be independent of the values read from the reserved register bits
- The MPU AXI supports storage of information of first memory protection violation on the bus. Therefore, once an NMI interrupt flag is set, the further monitoring of AXI master interface is stalled until MPUXSHE_CTRL0:NMI bit is cleared. This implies that any protection violation occurring on the bus while the MPUXSHE_CTRL0:NMI is set are simply ignored by MPU AXI. Software developers must therefore try to keep the ISR size small, so that the NMI interrupt of MPU AXI does not remain unattended for long

■ Steps in Programming the MPU AXI Module

Figure 5-1 Programmer's Flowchart



1. After the system reset, the software detects the module ID number of MPU AXI, by reading the MPUXSHE_MID register. This helps it in identifying the attributes and capabilities supported by the MPU AXI module. Then the software configures MPU AXI by setting appropriate registers.

2. By default, MPU AXI propagates the MPUXSHE_CTRL0:NMI flag to the CPU through the Interrupt Controller. If polling mode is desired, the software can reset the MPUXSHE_NMIEN:NMIEN bit to "0".

Note:

- *The MPUXSHE_NMIEN:NMIEN can be written only once after reset. Subsequent write accesses to this bit have no visible impact on the state of this bit.*

3. When the NMI is triggered or is in polling mode, if the software detects during its polling cycle that the MPUXSHE_CTRL0:NMI status flag is set, the CPU is invoked. This would read the status information collected and stored by MPU AXI in its CSR.

4. The software diagnoses the information about the transaction for which memory protection violation was detected and initiates a corrective action (if any).

5. Once the software has processed the information from the status registers, it shall clear the MPUXSHE_CTRL0:NMI flag by writing a "1" to the MPUXSHE_CTRL0:NMICL bit. Clearing MPUXSHE_CTRL0:NMI flag ensures that the MPU AXI starts monitoring the AXI master interfaces again for checking memory protection violation.

Note:

- *Software may clear the NMI flag before taking corrective action, hence steps 4 and 5 can be interchanged.*

CHAPTER 26: Secure Hardware Extension (SHE)

This chapter explains the function and operation of the SHE module.

1. Overview
2. Block Diagram of the SHE Module
3. Secure Hardware Extension RegistersOperation of the Secure Hardware Extension
4. Operation of the Secure Hardware Extension
5. Notes on Using SHE
6. Enhanced Secure Hardware Extension
7. References

1. Overview

This section describes the features and the block diagram of the Secure Hardware Extension (SHE) module.

Features of SHE

The Secure Hardware Extension (SHE) module is an on-chip peripheral of the microcontroller that provides cryptographic functions and secure key storage. This chapter describes the hardware part of the SPANSION SHE implementation which, together with the software driver, fulfills the 'SHE Functional Specification v1.1, rev 439' [1] as well as Errata and amendments to this specification.

Features of the SHE Module Are:

- It implements all commands defined by the Functional Specification of SHE (Chapter 7), such as
  - It provides AES-128 encryption and decryption operations [2] in Electronic Cipher Book (ECB) and Cipher-Block Chaining (CBC) modes [3]
  - It supports the generation of the cipher-based message authentication code (CMAC) [4]
  - It implements the Miyaguchi-Preneel compression function (see [5] Algorithm 9.43).

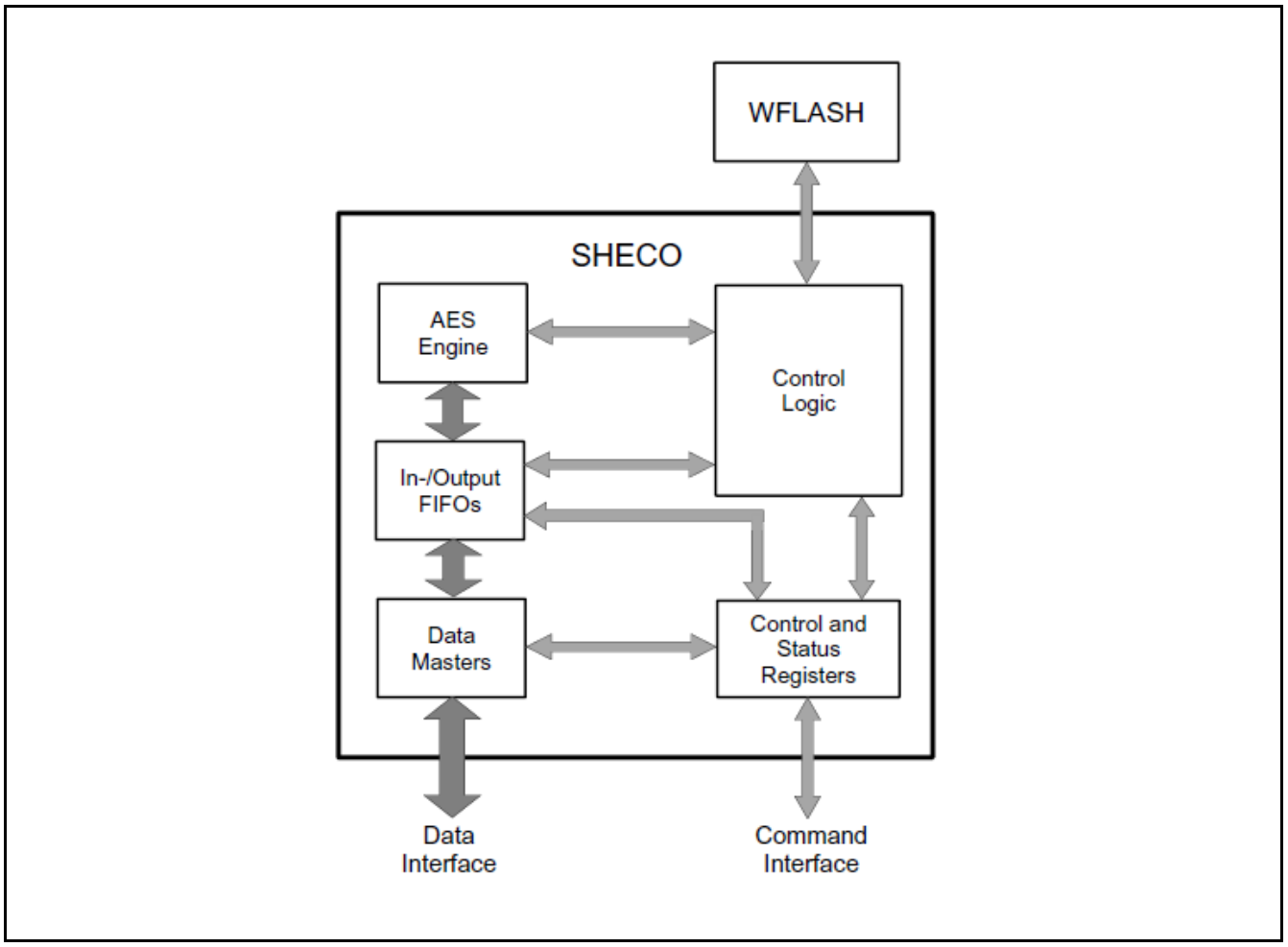
  Note:
  - *The Miyaguchi-Preneel compression function is not directly accessible by the user and will only be used internally if required during command processing*
  - It provides a random number generation function
- SHE supports secure booting (See SHE Functional Specification, Chapter 10) by ensuring:
  - Measurement before application start-up
  - Measurement during application start-up
  - Secure boot mode, start address and length of the boot loader are configurable by the user
- Secure key storage implemented in the WFLASH (Work Flash, refer to chapter WORKFLASH for more information). In this implementation:
  - Two small sectors are exclusively accessible from SHE and are not available for the user software (public side)
  - There is a round robin arbitration of accesses from SHE and the CPU
- It has a unique random and read-only SECRET_KEY inserted during production, which is only accessible inside the SHE module
- SHE has a unique read-only Identification item (UID), which is inserted during production
- It has a Command Interface for accessing configuration and status registers. This can also be used for data transfer to and from internal First-in/First-out (FIFOs)
- It has a data interface for direct access to microcontroller memory locations. This interface is:
  - Implemented as a master interface
  - Independent of read and write transfers
  - Configured by the software
- It implements FIFO-based data transfer for:
  - Command parameters
  - Data to be processed. Only 128-bit data blocks can be processed by SHE, hence all padding has to be done by the application
  - Results
- The internal SHE clock implements clock gating in order to reduce power consumption
- It features external debugger notification in order to recognize if debugging facilities of the MCU are activated

Note:

Bus-error occurs when register is accessed at SHE-OFF parts.

2. Block Diagram of the SHE Module

Figure 2-1 Block Diagram of the SHE Module



3. Secure Hardware Extension Registers

This section lists and describes all the SHE registers.

Registers of the SHE Module

The following registers are available for the SHE module:

Configuration Registers for Command Interface

- SHE Command Register (SHE_CMD)
- SHE Command Cancel Register (SHE_CMDCANCEL)
- SHE Clock Control Register (SHE_CLKCTRL)

Status Registers for Command Interface

- SHE Status Register (SHE_STATUS)
- SHE Error Code Register (SHE_ERC)
- SHE Clock Status Register (SHE_CLKSTAT)
- SHE Module ID Register (SHE_MID)

Interrupt Registers for Command Interface

- Interrupt Request Register (SHE_IRQ)
- Interrupt Request Enable Register (SHE_IRQEN)
- Interrupt Request Clear Register (SHE_IRQCLR)

Configuration Registers for Data Interface

- Input Channel Master Start Address Register (SHE_IMSTADDR)
- Output Channel Master Start Address Register (SHE_OMSTADDR)
- Input Channel Master Data Transfer Counter (SHE_IMSTCNT)
- Output Channel Master Data Transfer Counter (SHE_OMSTCNT)
- Input Channel Master Start Trigger (SHE_IMSTSTART)
- Output Channel Master Start Trigger (SHE_OMSTSTART)
- Input FIFO Configuration Register (SHE_IFIFOCFG)
- Output FIFO Configuration Register (SHE_OFIFOCFG)
- Input FIFO Compare Value Register (SHE_COMPARE0 to 1)

Status Registers for Data Interface

- Compare Register Access Status Register (SHE_COMPACC)
- Data Master Status Register (SHE_MSTSTATUS)
- Input Channel Master Error Response Address Register (SHE_IMSTERRADDR)
- Output Channel Master Error Response Address Register (SHE_OMSTERRADDR)
- FIFO Status Register (SHE_FIFOSTATUS)
- FIFO Load Register (SHE_FIFOLOAD)
- Input FIFO Data Counter Register (SHE_DATACNT0 to 1)

Data Transfer Registers

- Input FIFO Write Data Register (SHE_IFIFOWRDATA0 to 31)
- Output FIFO Read Data Register (SHE_OFIFORDDATA0 to 31)

Memory Layout of SHE Registers

Table 3-1 Memory Layout of the SHE Module Registers

Offset	+3	+2	+1	+0
0x00000000	SHE_CMD 00000000_00000000_00000000_00000000			
0x00000004	SHE_CMDCANCEL 00000000_00000000_00000000_00000000			
0x00000008	SHE_CLKCTRL 00000000_00000000_00000000_00000000			
0x0000000C	SHE_STATUS 01110000_00000000_00000000_00000000			
0x00000010	SHE_ERC 00000000_00010000_00000000_00000000			
0x00000014	SHE_CLKSTAT 00000000_00000000_00000000_00000000			
0x00000018	SHE_MID 00000000_00001111_00000010_00000000			
0x0000001C	SHE_IRQ 00000000_00000000_00000000_00000000			
0x00000020	SHE_IRQEN 00000000_00000000_00000000_00000000			
0x00000024	SHE_IRQCLR 00000000_00000000_00000000_00000000			
0x00000028	SHE_IMSTADDR 00000000_00000000_00000000_00000000			
0x0000002C	SHE_OMSTADDR 00000000_00000000_00000000_00000000			
0x00000030	SHE_IMSTCNT 00000000_00000000_00000000_00000000			
0x00000034	SHE_OMSTCNT 00000000_00000000_00000000_00000000			
0x00000038	SHE_IMSTSTART 00000000_00000000_00000000_00000000			
0x0000003C	SHE_OMSTSTART 00000000_00000000_00000000_00000000			
0x00000040	SHE_IFIFOCFG 00000000_00000001_00000000_00000000			
0x00000044	SHE_OFIFOCFG 00000000_00000001_00000000_00000000			

Offset	+3	+2	+1	+0
0x00000048	SHE_COMPARE0 11111111_11111111_11111111_11111111			
0x0000004C	SHE_COMPARE1 00000111_11111111_11111111_11111111			
0x00000050	SHE_COMPACC 00000000_00000000_00000000_00000000			
0x00000054	SHE_MSTSTATUS 00000000_00000001_00000000_00000001			
0x00000058	SHE_IMSTERRADDR 00000000_00000000_00000000_00000000			
0x0000005C	SHE_OMSTERRADDR 00000000_00000000_00000000_00000000			
0x00000060	SHE_FIFOSTATUS 00000000_00000000_00000000_00000000			
0x00000064	SHE_FIFOLOAD 00000000_00110000_00000000_00110000			
0x00000068	SHE_DATACNT0 00000000_00000000_00000000_00000000			
0x0000006C	SHE_DATACNT1 00000000_00000000_00000000_00000000			
0x00000070 -0x000000FC	Reserved 00000000_00000000_00000000_00000000			
0x00000100	SHE_IFIFOWRDATA0 0000000_ 00000000_00000000_00000000			
0x00000104	SHE_IFIFOWRDATA1 0000000_ 00000000_00000000_00000000			
0x00000108	SHE_IFIFOWRDATA2 0000000_ 00000000_00000000_00000000			
0x0000010C	SHE_IFIFOWRDATA3 0000000_ 00000000_00000000_00000000			
0x00000110	SHE_IFIFOWRDATA4 0000000_ 00000000_00000000_00000000			
0x00000114	SHE_IFIFOWRDATA5 0000000_ 00000000_00000000_00000000			
0x00000118	SHE_IFIFOWRDATA6 0000000_ 00000000_00000000_00000000			
0x0000011C	SHE_IFIFOWRDATA7 0000000_ 00000000_00000000_00000000			

Offset	+3	+2	+1	+0
0x00000120	SHE_IFIFOWRDATA8 0000000_ 00000000_00000000_00000000			
0x00000124	SHE_IFIFOWRDATA9 0000000_ 00000000_00000000_00000000			
0x00000128	SHE_IFIFOWRDATA10 0000000_ 00000000_00000000_00000000			
0x0000012C	SHE_IFIFOWRDATA11 0000000_ 00000000_00000000_00000000			
0x00000130	SHE_IFIFOWRDATA12 0000000_ 00000000_00000000_00000000			
0x00000134	SHE_IFIFOWRDATA13 0000000_ 00000000_00000000_00000000			
0x00000138	SHE_IFIFOWRDATA14 0000000_ 00000000_00000000_00000000			
0x0000013C	SHE_IFIFOWRDATA15 0000000_ 00000000_00000000_00000000			
0x00000140	SHE_IFIFOWRDATA16 0000000_ 00000000_00000000_00000000			
0x00000144	SHE_IFIFOWRDATA17 0000000_ 00000000_00000000_00000000			
0x00000148	SHE_IFIFOWRDATA18 0000000_ 00000000_00000000_00000000			
0x0000014C	SHE_IFIFOWRDATA19 0000000_ 00000000_00000000_00000000			
0x00000150	SHE_IFIFOWRDATA20 0000000_ 00000000_00000000_00000000			
0x00000154	SHE_IFIFOWRDATA21 0000000_ 00000000_00000000_00000000			
0x00000158	SHE_IFIFOWRDATA22 0000000_ 00000000_00000000_00000000			
0x0000015C	SHE_IFIFOWRDATA23 0000000_ 00000000_00000000_00000000			
0x00000160	SHE_IFIFOWRDATA24 0000000_ 00000000_00000000_00000000			
0x00000164	SHE_IFIFOWRDATA25 0000000_ 00000000_00000000_00000000			
0x00000168	SHE_IFIFOWRDATA26 0000000_ 00000000_00000000_00000000			

Offset	+3	+2	+1	+0
0x0000016C	SHE_IFIFOWRDATA27 0000000_ 00000000_00000000_00000000			
0x00000170	SHE_IFIFOWRDATA28 0000000_ 00000000_00000000_00000000			
0x00000174	SHE_IFIFOWRDATA29 0000000_ 00000000_00000000_00000000			
0x00000178	SHE_IFIFOWRDATA30 0000000_ 00000000_00000000_00000000			
0x0000017C	SHE_IFIFOWRDATA31 0000000_ 00000000_00000000_00000000			
0x00000180	SHE_OFIFORDDATA0 0000000_ 00000000_00000000_00000000			
0x00000184	SHE_OFIFORDDATA1 0000000_ 00000000_00000000_00000000			
0x00000188	SHE_OFIFORDDATA2 0000000_ 00000000_00000000_00000000			
0x0000018C	SHE_OFIFORDDATA3 0000000_ 00000000_00000000_00000000			
0x00000190	SHE_OFIFORDDATA4 0000000_ 00000000_00000000_00000000			
0x00000194	SHE_OFIFORDDATA5 0000000_ 00000000_00000000_00000000			
0x00000198	SHE_OFIFORDDATA6 0000000_ 00000000_00000000_00000000			
0x0000019C	SHE_OFIFORDDATA7 0000000_ 00000000_00000000_00000000			
0x000001A0	SHE_OFIFORDDATA8 0000000_ 00000000_00000000_00000000			
0x000001A4	SHE_OFIFORDDATA9 0000000_ 00000000_00000000_00000000			
0x000001A8	SHE_OFIFORDDATA10 0000000_ 00000000_00000000_00000000			
0x000001AC	SHE_OFIFORDDATA11 0000000_ 00000000_00000000_00000000			
0x000001B0	SHE_OFIFORDDATA12 0000000_ 00000000_00000000_00000000			
0x000001B4	SHE_OFIFORDDATA13 0000000_ 00000000_00000000_00000000			

Offset	+3	+2	+1	+0
0x000001B8	SHE_OFIFORDDATA14 0000000_ 00000000_00000000_00000000			
0x000001BC	SHE_OFIFORDDATA15 0000000_ 00000000_00000000_00000000			
0x000001C0	SHE_OFIFORDDATA16 0000000_ 00000000_00000000_00000000			
0x000001C4	SHE_OFIFORDDATA17 0000000_ 00000000_00000000_00000000			
0x000001C8	SHE_OFIFORDDATA18 0000000_ 00000000_00000000_00000000			
0x000001CC	SHE_OFIFORDDATA19 0000000_ 00000000_00000000_00000000			
0x000001D0	SHE_OFIFORDDATA20 0000000_ 00000000_00000000_00000000			
0x000001D4	SHE_OFIFORDDATA21 0000000_ 00000000_00000000_00000000			
0x000001D8	SHE_OFIFORDDATA22 0000000_ 00000000_00000000_00000000			
0x000001DC	SHE_OFIFORDDATA23 0000000_ 00000000_00000000_00000000			
0x000001E0	SHE_OFIFORDDATA24 0000000_ 00000000_00000000_00000000			
0x000001E4	SHE_OFIFORDDATA25 0000000_ 00000000_00000000_00000000			
0x000001E8	SHE_OFIFORDDATA26 0000000_ 00000000_00000000_00000000			
0x000001EC	SHE_OFIFORDDATA27 0000000_ 00000000_00000000_00000000			
0x000001F0	SHE_OFIFORDDATA28 0000000_ 00000000_00000000_00000000			
0x000001F4	SHE_OFIFORDDATA29 0000000_ 00000000_00000000_00000000			
0x000001F8	SHE_OFIFORDDATA30 0000000_ 00000000_00000000_00000000			
0x000001FC	SHE_OFIFORDDATA31 0000000_ 00000000_00000000_00000000			

Notes:

- *Initial value of SHE_CMD depends on the boot configuration.*

- *Initial value of SHE_STATUS depends on the boot configuration.*
- *Initial value of SHE_IRQ depends on the boot configuration.*

3.1. Configuration Registers for the Command Interface

3.1.1. SHE Command Register (SHE_CMD)

The SHE command register is used to initiate the execution of sequential commands, i.e. all commands defined in Chapter 7 of the SHE Functional Specification except for CMD_CANCEL and CMD_GET_STATUS.

Writing valid command opcode to this register starts command execution. Writing invalid command opcode generates an ERC_GENERAL_ERROR error code in the error register bit field SHE_ERC:CMDMAIN.

The busy flag (SHE_STATUS:BUSY) is set and the done flag (SHE_STATUS:DONE) is cleared by the hardware with a write access to the SHE_CMD register. Moreover, the error code register SHE_ERC:CMDMAIN is set to ERC_INVALID to indicate that error code is not yet available.

A new sequential command can only be started if the previous one has finished. Hence the register is locked for write accesses while command execution is ongoing, i.e. SHE_STATUS:BUSY is "1". Any write access will trigger a bus error response. Moreover, the command register SHE_CMD is also locked during the execution of the command CMD_CANCEL, i.e. while SHE_STATUS:BUSY is high. Refer to Section 2.2.1 for more information about using this register.

Bit	31	30	29	28	27	26	25	24
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	CMD[7]	CMD[6]	CMD[5]	CMD[4]	CMD[3]	CMD[2]	CMD[1]	CMD[0]
R/W Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Table 3-2 SHE Command Register (SHE_CMD) Bits

Bit Position	Bit Field Name	Bit Description
[31:8]	Reserved	-
[7:0]	CMD	Command Opcode register Listed below are possible opcode values with the related commands. Other values are reserved and will generate an ERC_GENERAL_ERROR error code in the SHE_ERC:CMDMAIN error register. 0x01: CMD_ENCRYPT_ECB 0x02: CMD_ENCRYPT_CBC 0x03: CMD_DECRYPT_ECB 0x04: CMD_DECRYPT_CBC 0x05: CMD_GENERATE_MAC 0x06: CMD_VERIFY_MAC 0x07: CMD_LOAD_KEY 0x08: CMD_LOAD_PLAIN_KEY 0x09: CMD_EXPORT_RAM_KEY 0x0A: CMD_INIT_RNG 0x0B: CMD_EXTEND_SEED 0x0C: CMD_RND 0x0D: CMD_SECURE_BOOT 0x0E: CMD_BOOT_FAILURE 0x0F: CMD_BOOT_OK 0x11: CMD_GET_ID 0x13: CMD_DEBUG 0x41: CMD_ENC_ECB_ESHE (ESHE only) 0x42: CMD_ENC_CBC_ESHE (ESHE only) 0x43: CMD_DEC_ECB_ESHE (ESHE only) 0x44: CMD_DEC_CBC_ESHE (ESHE only) 0x45: CMD_GENERATE_MAC_ESHE (ESHE only) 0x46: CMD_VERIFY_MAC_ESHE (ESHE only) 0x47: CMD_LOAD_KEY_ESHE (ESHE only) Notes: 1) The concurrent command CMD_CANCEL is started using its own register (SHE_CMDCANCEL). 2) The concurrent command CMD_GET_STATUS simply reads the SHE status register (SHE_STATUS).

Notes:

- *Initial value depends on the boot configuration.*
- *In the case of the other values except above ones, it is not in the normal status. Please take appropriate measures for the system characteristic.*

3.1.2. SHE Command Cancel Register (SHE_CMDCANCEL)

This is a dedicated register that initiates the CMD_CANCEL command concurrently with another ongoing command.

The busy flag (SHE_STATUS:BUSY) is set and done flag (SHE_STATUS:DONE) is cleared by the hardware upon write access to the SHE_CMDCANCEL register. Moreover, the error code for the CMD_CANCEL command (SHE_ERC:CANCELMAIN) is set to ERC_INVALID.

Note:

- *The SHE command register (SHE_CMD) is locked while the CMD_CANCEL command is being processed.*

Bit	31	30	29	28	27	26	25	24
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	CANCELREQ
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R,W
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Table 3-3 SHE Command Cancel Register (SHE_CMDCANCEL) Bits

Bit Position	Bit Field Name	Bit Description
[31:8]	Reserved	-
[7:1]	Reserved	-
[0]	CANCELREQ	Cancel Request bit Any ongoing command execution inside SHE can be canceled by writing this bit to "1". It remains "1" while the cancel request is being processed and will be cleared by the SHE control logic afterwards.

3.1.3. SHE Clock Control Register (SHE_CLKCTRL)

This register is used for enabling/disabling the clock. The software can disable the internal clock of the SHE module in order to reduce power consumption.

Note:

- *Only the SHE_CLKCTRL and SHE_CLKSTAT registers are accessible if the internal SHE clock is disabled. Accessing other SHE registers will generate a bus error response. Refer to Section "Clock gating for SHE" for more details.*

Bit	31	30	29	28	27	26	25	24
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	DISREQ
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R,W
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	ENREQ
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R,W
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Table 3-4 SHE Clock Control Register (SHE_CLKCTRL) Bits

Bit Position	Bit Field Name	Bit Description
[31:24]	Reserved	-
[23:17]	Reserved	-
[16]	DISREQ	Clock Disable Request bit Writing this bit to "1" triggers a clock disable request to the SHE control logic. DISREQ is automatically cleared when the SHE clock is disabled (i.e. SHE_CLKSTAT:CLKOFF = "1"). The internal SHE clock can only be disabled if SHE is in idle mode, i.e. if the clock disable request is written during ongoing command execution, then the clock won't be disabled until SHE becomes idle. This bit can only be written to "1". Writing "0" is ignored. Writing "1" while the SHE clock is disabled is ignored and the DISREQ bit remains "0".
[15:8]	Reserved	-
[7:1]	Reserved	-
[0]	ENREQ	Clock Enable Request bit Writing this bit to "1" triggers a clock enable request to the SHE control logic. ENREQ is automatically cleared when the SHE clock is enabled (i.e. SHE_CLKSTAT:CLKOFF = "0"). This bit can only be written to "1". Writing "0" is ignored. Writing "1" while the SHE clock is enabled is ignored and the ENREQ bit remains "0".

3.2. Status Registers for the Command Interface

3.2.1. SHE Status Register (SHE_STATUS)

A status register provides the state of SHE. It contains eight status bits, which are defined in the SHE Functional Specification (Chapter 6) along with additional status bits required for more detailed information.

Bit	31	30	29	28	27	26	25	24
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	FATALERR
R/W Attribute	R0,WX	R1,WX	R1,WX	R1,WX	R0,WX	R0,WX	R0,WX	R,WX
Protection Attribute	-							
Initial Value	0	1	1	1	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	Reserved	Reserved	Reserved	Reserved	FLASHDED	FLASHSEC	RAMDED	RAMSEC
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	INITDONE	DONE
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	INTDEBUGGER	EXTDEBUGGER	RNDINIT	BOOTOK	BOOTFINISHED	BOOTINIT	SECUREBOOT	BUSY
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Table 3-5 SHE Status Register (SHE_STATUS) Bits

Bit Position	Bit Field Name	Bit Description
[31]	Reserved	-
[30:28]	Reserved	-
[27:25]	Reserved	-
[24]	FATALERR	Fatal Error flag This flag is set in the event of an unrecoverable error inside the SHE module. It is cleared by a reset only. If an unrecoverable error is detected, SHE immediately stops its current operation, moves to a safe state and becomes unresponsive. A reset is required before SHE can be used again. Note: The update of the error code register SHE_ERC may take up to 100 clock cycles after the rise of the FATALERR flag, depending on the severity of the error.
[23:20]	Reserved	-
[19]	FLASHDED	Flash Double Bit ECC Error flag This flag indicates that a double bit Error Correction Code (ECC) error occurred in SHE's flash memory (dedicated part of WFLASH). The flag is set on the first occurrence of the error and can only be cleared by a reset.
[18]	FLASHSEC	Flash Single Bit ECC Warning flag This flag indicates that a single bit ECC error occurred in SHE's flash memory (i.e. a dedicated part of WFLASH) and has been corrected. The flag is set on the first occurrence of the error and can only be cleared by a reset.
[17]	RAMDED	RAM Double Bit ECC Error flag This flag indicates that a double bit ECC error occurred in SHE's internal RAM. The flag is set on the first occurrence of the error and can only be cleared by a reset.
[16]	RAMSEC	RAM Single Bit ECC Warning flag This flag indicates that a single bit ECC error in SHE's internal RAM occurred and has been corrected. The flag is set on the first occurrence of the error and can only be cleared by a reset.
[15:10]	Reserved	-
[9]	INITDONE	SHE Initialization status flag This flag indicates that the initialization phase of SHE is finished.
[8]	DONE	Done status flag This flag indicates that command processing inside SHE is finished and an appropriate error code is present in the error code register SHE_ERC. It is cleared immediately upon a 32-bit read access to SHE_ERC.
[7]	INTDEBUGGER	Internal Debugger status flag Defined in the SHE Functional Specification (Chapter 6). See the 'References' section of this chapter. This bit is set if the internal debugging mechanisms of SHE are activated. It is cleared by the reset only.

Bit Position	Bit Field Name	Bit Description
[6]	EXTDEBUGGER	External Debugger status flag Defined in the SHE Functional Specification (Chapter 6). See the 'References' section of this chapter. This bit is set if an external debugger is connected to the MCU, i.e. it reflects the input to activate the debugger. It is cleared by the reset only. Note: External debugger detection is active even if the SHE clock is switched off. More precisely, if an external debugger is attached while SHE is in power saving mode, then this will be detected and the EXTDEBUGGER bit set after the SHE clock has been enabled.
[5]	RNDINIT	Random Seed Initialization status flag Defined in the SHE Functional Specification (Chapter 6). See the 'References' section of this chapter. This bit is set if the random number generator has been initialized. It is cleared after successful authentication during the CMD_DEBUG command.
[4]	BOOTOK	Boot OK status flag Defined in the SHE Functional Specification (Chapter 6). See the 'References' section of this chapter. This bit is set if secure booting (CMD_SECURE_BOOT command) has succeeded. If the CMD_BOOT_FAILURE command is called, then the bit is erased. The Boot OK status flag is also cleared if some parts of the MCU enter power saving mode (refer to the Section titled 'Power saving functionality' for more information).
[3]	BOOTFINISHED	Boot Finished status flag Defined in the SHE Functional Specification (Chapter 6). See the 'References' section of this chapter. This bit is set when secure booting has been completed by calling either the CMD_BOOT_FAILURE or CMD_BOOT_OK commands (in the SHE_CMD register), or if the CMD_SECURE_BOOT command failed while verifying BOOT_MAC.
[2]	BOOTINIT	Boot Initialization status flag Defined in the SHE Functional Specification (Chapter 6). See the 'References' section of this chapter. This bit is set if secure booting has been personalized during the boot sequence.
[1]	SECUREBOOT	Secure Boot Activated status flag Defined in the SHE Functional Specification (Chapter 6). See the 'References' section of this chapter. This bit is set if secure booting is activated.

Bit Position	Bit Field Name	Bit Description
[0]	BUSY	Busy status flag Defined in the SHE Functional Specification (Chapter 6). See the 'References' section of this chapter. This bit is set whenever SHE is processing a command. The BUSY status flag is set immediately upon the start of a new command, i.e. write access to the command register (SHE_CMD) while BUSY is "0" or write access to the command cancel register SHE_CMDCANCEL. The BUSY status flag is cleared immediately upon a 32-bit read access to the error code register (SHE_ERC) while the SHE_STATUS:DONE flag is "1".

Notes:

- *Initial value depends on the boot configuration.*
- *In the case of the other values except above ones, it is not in the normal status. Please take appropriate measures for the system characteristic.*

3.2.2. SHE Error Code Register (SHE_ERC)

The SHE module outputs error code after the execution of a command in the SHE_ERC register. This register is divided into two parts: error codes for regular/sequential commands (all SHE commands except for CMD_CANCEL and CMD_GET_STATUS) and error codes for CMD_CANCEL.

The CMD_CANCEL command requires a dedicated error code register because it can be executed concurrently with other commands and also returns an error code. The currently executed command will return ERC_CANCELED to inform the software about its cancellation.

Table 3-6 lists only several often used codes, Refer to chapter 5.1.2 for the complete list of the error codes.

Bit	31	30	29	28	27	26	25	24
Field	CANCELEXTD[7]	CANCELEXTD[6]	CANCELEXTD[5]	CANCELEXTD[4]	CANCELEXTD[3]	CANCELEXTD[2]	CANCELEXTD[1]	CANCELEXTD[0]
R/W Attribute	R,WX	R,WX	R0,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	CANCELMAIN[7]	CANCELMAIN[6]	CANCELMAIN[5]	CANCELMAIN[4]	CANCELMAIN[3]	CANCELMAIN[2]	CANCELMAIN[1]	CANCELMAIN[0]
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	0	0	0	1	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	CMDEXTD[7]	CMDEXTD[6]	CMDEXTD[5]	CMDEXTD[4]	CMDEXTD[3]	CMDEXTD[2]	CMDEXTD[1]	CMDEXTD[0]
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	CMDMAIN[7]	CMDMAIN[6]	CMDMAIN[5]	CMDMAIN[4]	CMDMAIN[3]	CMDMAIN[2]	CMDMAIN[1]	CMDMAIN[0]
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Table 3-6 SHE Error Code Register (SHE_ERC) Bits

Bit Position	Bit Field Name	Bit Description
[31:24]	CANCELEXTD	Extended Command CMD_CANCEL Execution Error Code These bits contain the extended error code to the error code stored in the eight CANCELMAIN bits and is only valid if the value of the CANCELMAIN bits is equal to ERC_GENERAL_ERROR. It can be used to retrieve additional information about errors that have occurred during the processing of CMD_CANCEL. 0x00: ERC_INVALID CMD_CANCEL processing ongoing; error code not yet available
[23:16]	CANCELMAIN	Command CMD_CANCEL Execution Error Code These bits contain the dedicated error code for the CMD_CANCEL command. The default value is ERC_CANCEL_IDLE and is written after a reset and after the start of a new command (i.e. a write access to the SHE_CMD register while SHE is idle). The value changes to ERC_INVALID immediately after a cancel request (i.e. a write access to SHE_CMDCANCEL) and will be updated with the correct error code when CMD_CANCEL has finished processing. 0x00: ERC_INVALID CMD_CANCEL processing ongoing; error code not yet available 0x01: ERC_NO_ERROR CMD_CANCEL processing has completed without error 0x0D: ERC_GENERAL_ERROR A general error occurred during CMD_CANCEL processing 0x10: ERC_CANCEL_IDLE (Default value) No CMD_CANCEL processing is ongoing
[15:8]	CMDEXTD	Extended SHE Command Execution Error Code These bits contain the extended error code to the error code already stored in the eight CMDMAIN bits. The extended error code is only valid if the value of CMDMAIN is not equal to ERC_INVALID or ERC_NO_ERROR. It can be used to retrieve additional information about errors that have occurred. 0x00: ERC_INVALID The command is ongoing and the error code is not available, or the command has successfully finished (depending on value of CMDMAIN) 0x80: ERC_EXCESS_INPUT_DATA The input FIFO was not empty at the moment when the compare match event occurred

Bit Position	Bit Field Name	Bit Description
[7:0]	CMDMAIN	SHE Command Execution Error Code These bits contains the error codes which can be returned for the execution of all SHE commands with the exception of CMD_CANCEL and CMD_GET_STATUS. Two error code values are added and will be handled by the SHE software driver. Other values are reserved. The ERC_BUSY error code (not listed below) must be generated by the software driver if it receives an error response while writing to the locked command register (SHE_CMD). Upon the start of a new command, the error code is reset to ERC_INVALID until after the command has executed, when it is then changed to an appropriate value, e.g. it will be set to ERC_CANCELED if CMD_CANCEL was issued during command execution. Possible error code values, defined in the SHE Functional Specification (Chapter 8) are summarized below: 0x00: ERC_INVALID The current command is still ongoing and error code is not available (Default value) 0x01: ERC_NO_ERROR No error has occurred and the command will be executed 0x02: ERC_SEQUENCE_ERROR Sequence of commands or subcommands is out of order 0x03: ERC_KEY_NOT_AVAILABLE A key is locked due to either a failed boot measurement or an active debugger 0x04: ERC_KEY_INVALID A function is called to perform an operation with a key that is not permitted for the given operation 0x05: ERC_KEY_EMPTY The application attempts to use a key not yet initialized 0x06: ERC_NO_SECURE_BOOT Returned by the CMD_SECURE_BOOT command when the conditions for a secure boot process have not been met and the process has to be canceled. It is also returned if a function changing the boot status is called without secure booting or after the secure boot process has finished 0x07: ERC_KEY_WRITE_PROTECTED A key update is attempted on a memory slot that has been write protected or when an attempt to active the debugger is started when a key is write-protected 0x08: ERC_KEY_UPDATE_ERROR A key update did not succeed due to errors in verifying the messages 0x09: ERC_RNG_SEED Returned by the CMD_RND and CMD_DEBUG commands if the seed has not been initialized 0x0A: ERC_NO_DEBUGGING Internal debugging is not possible because authentication with the challenge response protocol has not succeeded 0x0C: ERC_MEMORY_FAILURE See Chapter 8 in the SHE Functional Specification 0x0D: ERC_GENERAL_ERROR See Chapter 8 in the SHE Functional Specification 0x0E: ERC_CANCELED Command execution has been canceled

3.2.3. SHE Clock Status Register (SHE_CLKSTAT)

This register provides information about whether the SHE module clock is enabled or disabled in order to reduce power consumption.

Bit	31	30	29	28	27	26	25	24
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	CLKOFF
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Table 3-7 SHE Clock Status Register (SHE_CLKSTAT) Bits

Bit Position	Bit Field Name	Bit Description
[31:8]	Reserved	-
[7:1]	Reserved	-
[0]	CLKOFF	Clock Disabled flag This bit is set to "1" if the SHE clock is disabled. Otherwise it is cleared

3.2.4. SHE Module ID Register (SHE_MID)

This is a read-only register with a unique module identification number (MID) which identifies the SHE module version used in the MCU.

Bit	31	30	29	28	27	26	25	24
Field	MID[31]	MID[30]	MID[29]	MID[28]	MID[27]	MID[26]	MID[25]	MID[24]
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	MID[23]	MID[22]	MID[21]	MID[20]	MID[19]	MID[18]	MID[17]	MID[16]
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R1,WX	R1,WX	R1,WX	R1,WX
Protection Attribute	-							
Initial Value	0	0	0	0	1	1	1	1

Bit	15	14	13	12	11	10	9	8
Field	MID[15]	MID[14]	MID[13]	MID[12]	MID[11]	MID[10]	MID[9]	MID[8]
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R1,WX	R0,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	1	0

Bit	7	6	5	4	3	2	1	0
Field	MID[7]	MID[6]	MID[5]	MID[4]	MID[3]	MID[2]	MID[1]	MID[0]
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Table 3-8 SHE Module ID Register (SHE_MID) Bits

Bit Position	Bit Field Name	Bit Description
[31:0]	MID	Module ID register This register identifies the particular version of the hardware used in the device. Note: Refer to the device specific datasheet for more details about the module identification number.

3.3. Interrupt Registers for Command Interface

3.3.1. Interrupt Request Register (SHE_IRQ)

This register contains active interrupt requests, each of which is associated with a specific event. The lower half-word represents regular interrupts, i.e. events which are expected during command execution. All these interrupts are combined and propagated to the interrupt controller on a single shared interrupt line. The upper half-word represents error interrupts. These interrupts are also combined, resulting in a shared error interrupt line.

All interrupts can be enabled and cleared independently of each other using the SHE_IRQEN and SHE_IRQCLR registers respectively. Refer to Section “Interrupt request generation” for more information on interrupt generation/processing.

Bit	31	30	29	28	27	26	25	24
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	FATALERR	IFIFOLOCKERR	OMSTIFSELERR	IMSTIFSELERR	OMSTERR	IMSTERR	OFIFORDERR	IFIFOWRERR
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	Reserved	Reserved	OFIFORDTH	IFIFOWRTH	OMSTIDLE	IMSTIDLE	DONE	COMPAREMATCH
R/W Attribute	R0,WX	R0,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Table 3-9 Interrupt Request Register (SHE_IRQ) Bits

Bit Position	Bit Field Name	Bit Description
[31:24]	Reserved	-
[23]	FATALERR	Fatal Error Interrupt flag This bit is set in the event of an unrecoverable error inside the SHE module (on the rising edge of SHE_STATUS:FATALERR). The Fatal Error Interrupt flag can only be cleared by software (refer to the SHE_IRQCLR:FATALERR bit).
[22]	IFIFOLOCKERR	Write To Locked Input FIFO Interrupt flag This bit is set if a write access to the locked input FIFO is performed either by the input channel master or through the command interface. The Write To Locked Input FIFO Interrupt flag can only be cleared by software (refer to the SHE_IRQCLR:IFIFOLOCKERR bit).
[21]	OMSTIFSELERR	Output Data Channel Interface Selection Error Interrupt flag This bit is set if a data transfer start for the output data channel master is requested (i.e. write access to the SHE_OMSTSTART register while in idle state) and the output FIFO interface selection is configured to 'Command Interface'. The Output Data Channel Interface Selection Error Interrupt flag can only be cleared by the software (refer to the SHE_IRQCLR:OMSTIFSELERR bit).
[20]	IMSTIFSELERR	Input Data Channel Interface Selection Error Interrupt flag This bit is set if a data transfer start for the input data channel master is requested (i.e. write access to the SHE_IMSTSTART register while in idle state) and the input FIFO interface selection is configured to 'Command Interface'. The Input Data Channel Interface Selection Error Interrupt flag can only be cleared by software (refer to the SHE_IRQCLR:IMSTIFSELERR bit).
[19]	OMSTERR	Output Channel Master Error Interrupt flag This bit is set if the output channel master receives an error response on the AXI bus (on the rising edge of SHE_MSTSTATUS:OMSTERR). The Output Channel Master Error Interrupt flag can only be cleared by the software (refer to the SHE_IRQCLR:OMSTERR bit).
[18]	IMSTERR	Input Channel Master Error Interrupt flag This bit is set if the input channel master receives an error response on the AXI bus (on the rising edge of SHE_MSTSTATUS:IMSTERR). The Input Channel Master Error Interrupt flag can only be cleared by the software (refer SHE_IRQCLR:IMSTERR bit).
[17]	OFIFORDERR	Read From Empty Output FIFO Error Interrupt flag This bit is set if a read access to the empty output First-in First-out (FIFO) is performed over the Command Interface. The Read From Empty Output FIFO Error Interrupt flag can only be cleared by software (refer to the SHE_IRQCLR:OFIFORDERR bit).
[16]	IFIFOWRERR	Write To Full Input FIFO Error Interrupt flag This bit is set if a write access to the full input FIFO is performed over the Command Interface. The Write To Full Input FIFO Error Interrupt flag can only be cleared by software (refer to the SHE_IRQCLR:IFIFOWRERR bit).

Bit Position	Bit Field Name	Bit Description
[15:8]	Reserved	-
[7:6]	Reserved	-
[5]	OFIFORDTH	Output FIFO Above Threshold Interrupt flag This bit is set if the amount of data words stored in the output FIFO becomes greater than the configured read threshold value (on the rising edge of SHE_FIFOSTATUS:OFIFORDTH). The Output FIFO Above Threshold Interrupt flag can only be cleared by software (refer to the SHE_IRQCLR:OFIFORDTH bit).
[4]	IFIFOWRTH	Input FIFO Below Threshold Interrupt flag This bit is set if the amount of data words stored in the input FIFO is less than the configured write threshold value (on the rising edge of SHE_FIFOSTATUS:IFIFOWRTH). The Input FIFO Below Threshold Interrupt flag can only be cleared by software (refer to the SHE_IRQCLR:IFIFOWRTH bit).
[3]	OMSTIDLE	Output Channel Master Idle Interrupt flag This bit is set if the output channel master moves to the idle state (on the rising edge of SHE_MSTSTATUS:OMSTIDLE). The Output Channel Master Idle Interrupt flag can only be cleared by software (refer to the SHE_IRQCLR:OMSTIDLE bit).
[2]	IMSTIDLE	Input Channel Master Idle Interrupt flag This bit is set if the input channel master moves to the idle state (on the rising edge of SHE_MSTSTATUS:IMSTIDLE). The Input Channel Master Idle Interrupt flag can only be cleared by software (refer to the SHE_IRQCLR:IMSTIDLE bit).
[1]	DONE	Command Execution Done Interrupt flag This bit is set if SHE internal processing has finished (on the rising edge of SHE_STATUS:DONE). The Command Execution Done Interrupt flag can only be cleared by software (refer to the SHE_IRQCLR:DONE bit).
[0]	COMPAREMATCH	Compare Match Interrupt flag This bit is set if the required amount of data has been transferred through the input FIFO (on the rising edge of SHE_FIFOSTATUS:COMPAREMATCH). The Compare Match Interrupt flag can only be cleared by software (refer to the SHE_IRQCLR:COMPAREMATCH bit).

Note:

- *Initial value depends on the boot configuration.*

3.3.2. Interrupt Request Enable Register (SHE_IRQEN)

This register enables the signaling of interrupts stored in SHE_IRQ to the host CPU. Refer to Section “Interrupt request generation” for more information on interrupt generation/processing.

Bit	31	30	29	28	27	26	25	24
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	FATALERR	IFIFOLOCKERR	OMSTIFSELERR	IMSTIFSELERR	OMSTERR	IMSTERR	OFIFORDERR	IFIFOWRERR
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	Reserved	Reserved	OFIFORDTH	IFIFOWRTH	OMSTIDLE	IMSTIDLE	DONE	COMPAREMATCH
R/W Attribute	R0,WX	R0,WX	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Table 3-10 Interrupt Request Enable Register (SHE_IRQEN) Bits

Bit Position	Bit Field Name	Bit Description
[31:24]	Reserved	-
[23]	FATALERR	Fatal Error Interrupt Enable bit If set to "1", the interrupt request stored in SHE_IRQ:FATALERR will be propagated to the interrupt controller.
[22]	IFIFOLOCKERR	Write To Locked Input FIFO Interrupt Enable bit If set to "1", the interrupt request stored in SHE_IRQ:IFIFOLOCKERR will be propagated to the interrupt controller.
[21]	OMSTIFSELERR	Output Data Channel Interface Selection Error Interrupt Enable bit If set to "1", the interrupt request stored in SHE_IRQ:OMSTIFSELERR will be propagated to the interrupt controller.
[20]	IMSTIFSELERR	Input Data Channel Interface Selection Error Interrupt Enable bit If set to "1", the interrupt request stored in SHE_IRQ:IMSTIFSELERR will be propagated to the interrupt controller.

Bit Position	Bit Field Name	Bit Description
[19]	OMSTERR	Output Data Channel Master Error Interrupt Enable bit If set to "1", the interrupt request stored in SHE_IRQ:OMSTERR will be propagated to the interrupt controller.
[18]	IMSTERR	Input Data Channel Master Error Interrupt Enable bit If set to "1", the interrupt request stored in SHE_IRQ:IMSTERR will be propagated to the interrupt controller.
[17]	OFIFORDERR	Read From Empty Output FIFO Error Interrupt Enable bit If set to "1", the interrupt request stored in SHE_IRQ:OFIFORDERR will be propagated to the interrupt controller.
[16]	IFIFOWRERR	Write To Full Input FIFO Error Interrupt Enable bit If set to "1", the interrupt request stored in SHE_IRQ:IFIFOWRERR will be propagated to the interrupt controller.
[15:8]	Reserved	-
[7:6]	Reserved	-
[5]	OFIFORDTH	Output FIFO Above Threshold Interrupt Enable bit If set to "1", the interrupt request stored in SHE_IRQ:OFIFORDTH will be propagated to the interrupt controller.
[4]	IFIFOWRTH	Input FIFO Below Threshold Interrupt Enable bit If set to "1", the interrupt request stored in SHE_IRQ:IFIFOWRTH will be propagated to the interrupt controller.
[3]	OMSTIDLE	Output Data Channel Master Idle Interrupt Enable bit If set to "1", the interrupt request stored in SHE_IRQ:OMSTIDLE will be propagated to the interrupt controller.
[2]	IMSTIDLE	Input Data Channel Master Idle Interrupt Enable bit If set to "1", the interrupt request stored in SHE_IRQ:IMSTIDLE will be propagated to the interrupt controller.
[1]	DONE	Command Execution Done Interrupt Enable bit If set to "1", the interrupt request stored in SHE_IRQ:DONE will be propagated to the interrupt controller.
[0]	COMPAREMATCH	Compare Match Interrupt Enable bit If set to "1", the interrupt request stored in SHE_IRQ:COMPAREMATCH will be propagated to the interrupt controller.

3.3.3. Interrupt Request Clear Register (SHE_IRQCLR)

This register clears interrupt requests stored in SHE_IRQ. Refer to Section “Interrupt request generation” for more information on interrupt generation/processing.

Bit	31	30	29	28	27	26	25	24
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	FATALERR	IFIFOLOCKERR	OMSTIFSELERR	IMSTIFSELERR	OMSTERR	IMSTERR	OFIFORDERR	IFIFOWRERR
R/W Attribute	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	Reserved	Reserved	OFIFORDTH	IFIFOWRTH	OMSTIDLE	IMSTIDLE	DONE	COMPAREMATCH
R/W Attribute	R0,WX	R0,WX	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Table 3-11 Interrupt Request Clear Register (SHE_IRQCLR) Bits

Bit Position	Bit Field Name	Bit Description
[31:24]	Reserved	-
[23]	FATALERR	Fatal Error Interrupt Clear bit Writing this bit to "1" clears the interrupt request stored in SHE_IRQ:FATALERR.
[22]	IFIFOLOCKERR	Write To Locked Input FIFO Interrupt Clear bit Writing this bit to "1" clears the interrupt request stored in SHE_IRQ:IFIFOLOCKERR.
[21]	OMSTIFSELERR	Output Data Channel Interface Selection Error Interrupt Clear bit Writing this bit to "1" clears the interrupt request stored in SHE_IRQ:OMSTIFSELERR.
[20]	IMSTIFSELERR	Input Data Channel Interface Selection Error Interrupt Clear bit Writing this bit to "1" clears the interrupt request stored in SHE_IRQ:IMSTIFSELERR.
[19]	OMSTERR	Output Data Channel Master Error Interrupt Clear bit Writing this bit to "1" clears the interrupt request stored in SHE_IRQ:OMSTERR.
[18]	IMSTERR	Input Data Channel Master Error Interrupt Clear bit Writing this bit to "1" clears the interrupt request stored in SHE_IRQ:IMSTERR.
[17]	OFIFORDERR	Read From Empty Output FIFO Error Interrupt Clear bit Writing this bit to "1" clears the interrupt request stored in SHE_IRQ:OFIFORDERR.
[16]	IFIFOWRERR	Write To Full Input FIFO Error Interrupt Clear bit Writing this bit to "1" clears the interrupt request stored in SHE_IRQ:IFIFOWRERR.
[15:8]	Reserved	-
[7:6]	Reserved	-
[5]	OFIFORDTH	Output FIFO Above Threshold Interrupt Clear bit Writing this bit to "1" clears the interrupt request stored in SHE_IRQ:OFIFORDTH.
[4]	IFIFOWRTH	Input FIFO Below Threshold Interrupt Clear bit Writing this bit to "1" clears the interrupt request stored in SHE_IRQ:IFIFOWRTH.
[3]	OMSTIDLE	Output Data Channel Master Idle Interrupt Clear bit Writing this bit to "1" clears the interrupt request stored in SHE_IRQ:OMSTIDLE.
[2]	IMSTIDLE	Input Data Channel Master Idle Interrupt Clear bit Writing this bit to "1" clears the interrupt request stored in SHE_IRQ:IMSTIDLE.
[1]	DONE	Command Execution Done Interrupt Clear bit Writing this bit to "1" clears the interrupt request stored in SHE_IRQ:DONE.
[0]	COMPAREMATCH	Compare Match Interrupt Clear bit Writing this bit to "1" clears the interrupt request stored in SHE_IRQ:COMPAREMATCH.

3.4. Configuration Registers for the Data Interface

3.4.1. Input Channel Master Start Address Register (SHE_IMSTADDR)

This register contains the memory start address for data transfer by the data interface input channel master. The address stored has to be aligned to the 64-bit boundary.

Moreover, the start address should be aligned to the 128-byte boundary if the transfer is supposed to cross the 4KB address boundary. Otherwise at least one short burst transfer (on reaching the 4KB boundary) will be performed. This limitation comes from the AXI protocol which cannot perform bursts over the 4KB address boundary.

This register must be configured by the user prior to the transfer. After the transfer has started the register is locked for writing. Write access to the locked register produces a bus error response. The lock is released if the required amount of data has been transferred (i.e. SHE_IMSTCNT is zero) or if the data transfer has been canceled.

Bit	31	30	29	28	27	26	25	24
Field	IMSTADDR[31]	IMSTADDR[30]	IMSTADDR[29]	IMSTADDR[28]	IMSTADDR[27]	IMSTADDR[26]	IMSTADDR[25]	IMSTADDR[24]
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	IMSTADDR[23]	IMSTADDR[22]	IMSTADDR[21]	IMSTADDR[20]	IMSTADDR[19]	IMSTADDR[18]	IMSTADDR[17]	IMSTADDR[16]
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	IMSTADDR[15]	IMSTADDR[14]	IMSTADDR[13]	IMSTADDR[12]	IMSTADDR[11]	IMSTADDR[10]	IMSTADDR[9]	IMSTADDR[8]
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	IMSTADDR[7]	IMSTADDR[6]	IMSTADDR[5]	IMSTADDR[4]	IMSTADDR[3]	IMSTADDR[2]	IMSTADDR[1]	IMSTADDR[0]
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R0,WX	R0,WX	R0,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Table 3-12 Input Channel Master Start Address Register (SHE_IMSTADDR) Bits

Bit Position	Bit Field Name	Bit Description
[31:0]	IMSTADDR	Input Channel Master Start Address

3.4.2. Output Channel Master Start Address Register (SHE_OMSTADDR)

This register contains the memory start address for data transfer by the data interface output channel master. The address stored has to be aligned to the 64-bit boundary.

Moreover, the start address should be aligned to the 128-byte boundary if the transfer is supposed to cross the 4KB address boundary. Otherwise at least one short burst transfer (on reaching the 4KB boundary) will be performed. This limitation comes from the AXI protocol, which cannot perform bursts over the 4KB address boundary.

This register has to be configured by the user prior to the transfer. After the transfer has started the register is locked for writing. Write access to the locked register produces a bus error response. The lock is released if the required amount of data has been transferred (i.e. SHE_OMSTCNT is zero) or if the data transfer has been canceled.

Bit	31	30	29	28	27	26	25	24
Field	OMSTADDR [31]	OMSTADDR [30]	OMSTADDR [29]	OMSTADDR [28]	OMSTADDR [27]	OMSTADDR [26]	OMSTADDR [25]	OMSTADDR [24]
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	OMSTADDR [23]	OMSTADDR [22]	OMSTADDR [21]	OMSTADDR [20]	OMSTADDR [19]	OMSTADDR [18]	OMSTADDR [17]	OMSTADDR [16]
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	OMSTADDR [15]	OMSTADDR [14]	OMSTADDR [13]	OMSTADDR [12]	OMSTADDR [11]	OMSTADDR [10]	OMSTADDR [9]	OMSTADDR [8]
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	OMSTADDR [7]	OMSTADDR [6]	OMSTADDR [5]	OMSTADDR [4]	OMSTADDR [3]	OMSTADDR [2]	OMSTADDR [1]	OMSTADDR [0]
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R0,WX	R0,WX	R0,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Table 3-13 Output Channel Master Start Address Register (SHE_OMSTADDR) Bits

Bit Position	Bit Field Name	Bit Description
[31:0]	OMSTADDR	Output Channel Master Start Address

3.4.3. Input Channel Master Data Transfer Counter (SHE_IMSTCNT)

This register defines the number of 64-bit double words to be transferred by the input channel master.

It has to be configured by the user prior to starting the transfer. After the transfer has started the register is locked for write accesses and will be updated by the hardware to represent the amount of remaining data transfers. Write access to the locked register produces a bus error response. The lock is released if the required amount of data has been transferred (i.e. SHE_IMSTCNT is zero) or if the data transfer has been canceled.

Bit	31	30	29	28	27	26	25	24
Field	Reserved	Reserved	Reserved	IMSTCNT[28]	IMSTCNT[27]	IMSTCNT[26]	IMSTCNT[25]	IMSTCNT[24]
R/W Attribute	R0,WX	R0,WX	R0,WX	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	IMSTCNT[23]	IMSTCNT[22]	IMSTCNT[21]	IMSTCNT[20]	IMSTCNT[19]	IMSTCNT[18]	IMSTCNT[17]	IMSTCNT[16]
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	IMSTCNT[15]	IMSTCNT[14]	IMSTCNT[13]	IMSTCNT[12]	IMSTCNT[11]	IMSTCNT[10]	IMSTCNT[9]	IMSTCNT[8]
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	IMSTCNT[7]	IMSTCNT[6]	IMSTCNT[5]	IMSTCNT[4]	IMSTCNT[3]	IMSTCNT[2]	IMSTCNT[1]	IMSTCNT[0]
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Table 3-14 Input Channel Master Data Transfer Counter (SHE_IMSTCNT) Bits

Bit Position	Bit Field Name	Bit Description
[31:29]	Reserved	-
[28:0]	IMSTCNT	Input Channel Master Data Transfer Counter

3.4.4. Output Channel Master Data Transfer Counter (SHE_OMSTCNT)

This register defines the number of 64-bit double words to be transferred by the output channel master.

This register has to be configured by the user prior to starting the transfer. After the transfer has started the register is locked for write accesses and will be updated by the hardware to represent the amount of remaining data transfers. Writing access to the locked register produces a bus error response. The lock is released if the required amount of data has been transferred (SHE_OMSTCNT is zero) or if the data transfer has been canceled.

Bit	31	30	29	28	27	26	25	24
Field	Reserved	Reserved	Reserved	OMSTCNT[28]	OMSTCNT[27]	OMSTCNT[26]	OMSTCNT[25]	OMSTCNT[24]
R/W Attribute	R0,WX	R0,WX	R0,WX	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	OMSTCNT[23]	OMSTCNT[22]	OMSTCNT[21]	OMSTCNT[20]	OMSTCNT[19]	OMSTCNT[18]	OMSTCNT[17]	OMSTCNT[16]
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	OMSTCNT[15]	OMSTCNT[14]	OMSTCNT[13]	OMSTCNT[12]	OMSTCNT[11]	OMSTCNT[10]	OMSTCNT[9]	OMSTCNT[8]
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	OMSTCNT[7]	OMSTCNT[6]	OMSTCNT[5]	OMSTCNT[4]	OMSTCNT[3]	OMSTCNT[2]	OMSTCNT[1]	OMSTCNT[0]
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Table 3-15 Output Channel Master Data Transfer Counter (SHE_OMSTCNT) Bits

Bit Position	Bit Field Name	Bit Description
[31:29]	Reserved	-
[28:0]	OMSTCNT	Output Channel Master Data Transfer Counter

3.4.5. Input Channel Master Start Trigger (SHE_IMSTSTART)

This register is used to start data transfer by the input data channel master.

Bit	31	30	29	28	27	26	25	24
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	IMSTSTART
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R,W
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Table 3-16 Input Channel Master Start Trigger (SHE_IMSTSTART) Bits

Bit Position	Bit Field Name	Bit Description
[31:8]	Reserved	-
[7:1]	Reserved	-
[0]	IMSTSTART	Input Channel Master Start Trigger Writing "1" to this bit starts the transfer of the input channel master. Writing "0" has no effect. The start trigger will be reset through the hardware when the required amount of data has been transferred to the input FIFO or if the data transfer was canceled. If the start of a new data transfer is requested and the input FIFO interface selection is configured to 'Command Interface' (which is in fact an error condition), the SHE_IMSTSTART bit remains zero but the SHE_IRQ:IMSTIFSELERR error interrupt flag is set instead.

3.4.6. Output Channel Master Start Trigger (SHE_OMSTSTART)

This register is used to start data transfer by the output data channel master.

Bit	31	30	29	28	27	26	25	24
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	OMSTSTART
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R,W
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Table 3-17 Output Channel Master Start Trigger (SHE_OMSTSTART) Bits

Bit Position	Bit Field Name	Bit Description
[31:8]	Reserved	-
[7:1]	Reserved	-
[0]	OMSTSTART	Output Channel Master Start Trigger Writing "1" to this bit starts the transfer of the output channel master. Writing "0" has no effect. The start trigger will be automatically reset through the hardware when the required amount of data has been transferred or if the data transfer was canceled. If the start of new data transfer is requested and the output FIFO interface selection is configured to 'Command Interface' (which is in fact an error condition), the SHE_OMSTSTART bit remains zero but the SHE_IRQ:OMSTIFSELERR error interrupt flag is set instead.

3.4.7. Input FIFO Configuration Register (SHE_IFIFOCFG)

This register controls the interface selection as well as the write threshold setting for the input FIFO.

Bit	31	30	29	28	27	26	25	24
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	IFSEL
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	Reserved	Reserved	WRTHRESHOLD[5]	WRTHRESHOLD[4]	WRTHRESHOLD[3]	WRTHRESHOLD[2]	WRTHRESHOLD[1]	WRTHRESHOLD[0]
R/W Attribute	R0,WX	R0,WX	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Table 3-18 Input FIFO Configuration Register (SHE_IFIFOCFG) Bits

Bit Position	Bit Field Name	Bit Description
[31:24]	Reserved	-
[23:17]	Reserved	-
[16]	IFSEL	Interface Selection bit 1: Input FIFO can be written through the Command Interface 0: Input FIFO can be written through the Data Interface If the data transfer has to be performed by the input data channel master, this bit must be configured (to "0") prior to starting the transfer. After the transfer starts, the bit is locked for writing and any attempt to write to it produces a bus error response. The lock is released if the required amount of data has been transferred (i.e. SHE_IMSTCNT is zero) or if the data transfer has been canceled
[15:8]	Reserved	-
[7:6]	Reserved	-
[5:0]	WRTHRESHOLD	Programmable Write Threshold Of Input FIFO The programmable write threshold of the input FIFO (in terms of 32-bit words) can be used for interrupt-based software design. If the amount of stored 32-bit data words (the amount is defined by the SHE_FIFOLOAD:IFIFOLOAD bits) is less than the value of WRTHRESHOLD, the IFIFOWRTH bit in the SHE_FIFOSTATUS register is set. Moreover, an interrupt request (i.e. SHE_IRQ:IFIFOWRTH) is set on the rising edge of SHE_FIFOSTATUS:IFIFOWRTH. The capacity of the input FIFO is 1,536 bits, i.e. 32 bits wide and 48 words deep. The threshold value is not verified by SHE for its feasibility. This means that if the programmed threshold value exceeds the capacity of the FIFO (i.e. WRTHRESHOLD > 48), the SHE_FIFOSTATUS:IFIFOW-RTH status bit will never be cleared. Note: This bit field should be written using 8- or 16-bit accesses in case of pending data transfer by the input data channel master (SHE_MSTSTATUS:IMSTLOCK == "1"). Otherwise there will be a bus error response because the IFSEL bit is locked for writing.

3.4.8. Output FIFO Configuration Register (SHE_OFIFOCFG)

This register controls the interface selection as well as the read threshold setting for the output FIFO.

Bit	31	30	29	28	27	26	25	24
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	IFSEL
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	Reserved	Reserved	RDTHRESHOLD[5]	RDTHRESHOLD[4]	RDTHRESHOLD[3]	RDTHRESHOLD[2]	RDTHRESHOLD[1]	RDTHRESHOLD[0]
R/W Attribute	R0,WX	R0,WX	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Table 3-19 Output FIFO Configuration Register (SHE_OFIFOCFG) Bits

Bit Position	Bit Field Name	Bit Description
[31:24]	Reserved	-
[23:17]	Reserved	-
[16]	IFSEL	Interface Selection bit 1: Output FIFO can be read through the Command Interface 0: Output FIFO can be read through the Data Interface If the data transfer has to be performed by the output data channel master, this bit must be configured (to "0") prior to starting the transfer. After the transfer starts, this bit is locked for writing and any attempt to write to this bit produces a bus error response. The lock is released if the required amount of data has been transferred (SHE_OMSTCNT is zero) or if the data transfer has been canceled
[15:8]	Reserved	-
[7:6]	Reserved	-
[5:0]	RDTHRESHOLD	Programmable Read Threshold Of Output FIFO The programmable read threshold of the output FIFO (in terms of 32-bit words) can be used for interrupt-based software design. If the amount of stored 32-bit data words (the amount is defined by the SHE_FIFOLOAD:OFIFOLOAD bits) is greater than the value of RDTHRESHOLD, the OFIFORDTH bit in the SHE_FIFOSTATUS register is set. Moreover, an interrupt request (SHE_IRQ:OFIFORDTH) is set on the rising edge of SHE_FIFOSTATUS:OFIFORDTH. The capacity of the output FIFO is equal to 1,536 bits, i.e. 32 bits wide and 48 words deep. The threshold value is not verified by SHE for its feasibility. This means that if the programmed threshold value exceeds the capacity of the FIFO (i.e. RDTHRESHOLD > 47), the SHE_FIFOSTATUS:OFIFORDTH status bit will never be set. Note: This bit field should be written using 8- or 16-bit accesses in case of pending data transfer by the input data channel master (SHE_MSTSTATUS:OMSTLOCK == "1"). Otherwise there will be a bus error response because the IFSEL bit is locked for writing.

3.4.9. Input FIFO Compare Value Register (SHE_COMPARE0 to 1)

This register contains the amount of data (all parameters and the actual data used for processing) required for a single SHE API command represented in terms of 32-bit data words.

SHE_COMPARE is used by the hardware to recognize that the required amount of data for a single SHE command has been transferred through the input FIFO. For this purpose the value of the compare register SHE_COMPARE is continuously compared with the value of the data counter register SHE_DATACNT. When the data counter reaches the value of the compare register, the status bit SHE_FIFOSTATUS:COMPAREMATCH and an interrupt request bit SHE_IRQ:COMPAREMATCH are immediately set. This transition of the COMPAREMATCH bit is referenced to as a compare match event.

SHE_COMPARE is reset by the hardware to the maximal value (2^{59} - 1) every time the opcode for a new SHE command is written to the SHE_CMD register.

In most cases the SHE_COMPARE register is configured by the control logic of SHE according to selected command and/or parameter values and cannot be changed during command execution. But for CBC commands, the software can adjust the value of SHE_COMPARE during the command execution, i. e. if the total amount of data is not known before the command start. Since the SHE_COMPARE register is 59 bits wide and accessed using a 32-bit interface, it cannot be written through an atomic operation and a faulty compare match event can occur. Hence the software has to use the following approach, when adjusting it: To increase the compare value, the software can write the most significant 32 bits first and then adjust the least significant 32 bits. To decrease the compare value, the software can write the least significant 32 bits first and then adjust the most significant 32 bits.

Note:

- *The software is only allowed to change the value of the SHE_COMPARE register if a CBC command is performed.* ***Before trying to modify the SHE_COMPARE register the software should verify that write access is enabled by checking the value of the SHE_COMPACC:CPUEN bit.*** *Otherwise a bus error response is generated.*

Input FIFO Compare Value Register 0 (SHE_COMPARE0)

Bit	31	30	29	28	27	26	25	24
Field	COMPARE [31]	COMPARE [30]	COMPARE [29]	COMPARE [28]	COMPARE [27]	COMPARE [26]	COMPARE [25]	COMPARE [24]
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-							
Initial Value	1	1	1	1	1	1	1	1

Bit	23	22	21	20	19	18	17	16
Field	COMPARE [23]	COMPARE [22]	COMPARE [21]	COMPARE [20]	COMPARE [19]	COMPARE [18]	COMPARE [17]	COMPARE [16]
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-							
Initial Value	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8
Field	COMPARE [15]	COMPARE [14]	COMPARE [13]	COMPARE [12]	COMPARE [11]	COMPARE [10]	COMPARE [9]	COMPARE [8]
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-							
Initial Value	1	1	1	1	1	1	1	1

Bit	7	6	5	4	3	2	1	0
Field	COMPARE [7]	COMPARE [6]	COMPARE [5]	COMPARE [4]	COMPARE [3]	COMPARE [2]	COMPARE [1]	COMPARE [0]
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-							
Initial Value	1	1	1	1	1	1	1	1

Table 3-20 Input FIFO Compare Value Register 0 (SHE_COMPARE0) Bits

Bit Position	Bit Field Name	Bit Description
[31:0]	COMPARE	Least significant bits of the compare register The amount of data required for a single SHE Application Programming Interface (API) command

Input FIFO Compare Value Register 1 (SHE_COMPARE1)

Bit	31	30	29	28	27	26	25	24
Field	Reserved	Reserved	Reserved	Reserved	Reserved	COMPARE[26]	COMPARE[25]	COMPARE[24]
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W	R/W	R/W
Protection Attribute	-							
Initial Value	0	0	0	0	0	1	1	1

Bit	23	22	21	20	19	18	17	16
Field	COMPARE[23]	COMPARE[22]	COMPARE[21]	COMPARE[20]	COMPARE[19]	COMPARE[18]	COMPARE[17]	COMPARE[16]
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-							
Initial Value	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8
Field	COMPARE[15]	COMPARE[14]	COMPARE[13]	COMPARE[12]	COMPARE[11]	COMPARE[10]	COMPARE[9]	COMPARE[8]
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-							
Initial Value	1	1	1	1	1	1	1	1

Bit	7	6	5	4	3	2	1	0
Field	COMPARE[7]	COMPARE[6]	COMPARE[5]	COMPARE[4]	COMPARE[3]	COMPARE[2]	COMPARE[1]	COMPARE[0]
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-							
Initial Value	1	1	1	1	1	1	1	1

Table 3-21 Input FIFO Compare Value Register 1 (SHE_COMPARE1) Bits

Bit Position	Bit Field Name	Bit Description
[31:27]	Reserved	-
[26:0]	COMPARE	Most significant bits of the compare register The amount of data required for a single SHE API command

3.5. Status Registers for Data Interface

3.5.1. Compare Register Access Status Register (SHE_COMPACC)

This register represents the current status of the write access permissions to the SHE_COMPARE register.

Bit	31	30	29	28	27	26	25	24
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	CPUEN
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Table 3-22 Compare Register Access Status Register (SHE_COMPACC) Bits

Bit Position	Bit Field Name	Bit Description
[31:8]	Reserved	-
[7:1]	Reserved	-
[0]	CPUEN	CPU Write Access Enabled status flag 1: The software is allowed to write to the SHE_COMPARE register 0: The software is not allowed to write to the SHE_COMPARE register

3.5.2. Data Master Status Register (SHE_MSTSTATUS)

This register represents the current status of the input and output channel data masters.

Bit	31	30	29	28	27	26	25	24
Field	Reserved	Reserved	Reserved	Reserved	Reserved	OMSTERRRESP[1]	OMSTERRRESP[0]	OMSTERR
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	OMSTLOCK	OMSTIDLE
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved	Reserved	Reserved	Reserved	IMSTERRRESP[1]	IMSTERRRESP[0]	IMSTERR
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	IMSTLOCK	IMSTIDLE
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	1

Table 3-23 Data Master Status Register (SHE_MSTSTATUS) Bits

Bit Position	Bit Field Name	Bit Description
[31:27]	Reserved	-
[26:25]	OMSTERRRESP	Output Data Channel Master Error Response Code These bits contain the error response code of the AXI protocol. '00': OKAY (Normal access okay) A normal access has been successful. This code can also indicate an exclusive access failure '01': EXOKAY (Exclusive access okay) Either the read or write portion of an exclusive access has been successful '10': SLVERR (Slave error) Access has reached the slave successfully but the slave returns an error condition to the originating master Typically, a decode error is generated by an interconnect component to indicate that there is no slave at the transaction address The register is updated in the event of an AXI bus error response on the output data channel.
[24]	OMSTERR	Output Data Channel Master Error Response flag This bit is set in the event of an AXI bus error response on the output data channel. It is cleared on the rising edge of the DONE flag (SHE_STATUS:DONE). The rising edge of the OMSTERR error flag sets the SHE_IRQ:OMSTERR interrupt flag.
[23:18]	Reserved	-
[17]	OMSTLOCK	Output Data Channel Master Lock Enabled flag If this bit is high, the output master channel configuration registers (SHE_OFIFOCFG:IFSEL, SHE_OMSTADDR and SHE_OMSTCNT) are locked for writing. The Output Data Channel Master Lock Enabled flag is set at the start of a data transfer. And it is cleared after the required amount of data has been transferred (i.e. SHE_OMSTCNT is zero), or if the data transfer was canceled and the current burst finished.
[16]	OMSTIDLE	Output Data Channel Master Idle flag This bit is high if no data transfer is ongoing. It is cleared at the start of a new data transfer. The Output Data Channel Master Idle flag is set when data transfer has completed, i.e. the Output Channel Master Data Transfer Counter (SHE_OMSTCNT) is zero and all the data has been transferred to the requested memory location, or if data transfer has been canceled and the current burst finished. The rising edge of the OMSTIDLE flag sets SHE_IRQ:OMSTIDLE interrupt flag.
[15:11]	Reserved	-

Bit Position	Bit Field Name	Bit Description
[10:9]	IMSTERRRESP	Input Data Channel Master Error Response Code These bits contain the error response code of the AXI protocol. '00': OKAY (Normal access okay) Normal access has been successful. This value can also indicate an exclusive access failure '01': EXOKAY (Exclusive access okay) Either the read or write portion of an exclusive access has been successful '10': SLVERR (Slave error) Slave error is used when the access has successfully reached the slave but the slave returns an error condition to the originating master Typically, a decode error is generated by an interconnect component to indicate that there is no slave at the transaction address The register is updated in the event of an AXI bus error response on the input data channel.
[8]	IMSTERR	Input Data Channel Master Error Response flag This bit is set in the event of an AXI bus error response on the input data channel. It is cleared on the rising edge of the DONE flag (SHE_STATUS:DONE). The rising edge of IMSTERR error flag sets the SHE_IRQ:IMSTERR interrupt flag.
[7:2]	Reserved	-
[1]	IMSTLOCK	Input Data Channel Master Lock Enabled flag If this bit is high, the input master channel configuration registers (SHE_IFIFOCFG:IFSEL, SHE_IMSTADDR and SHE_IMSTCNT) are locked for writing. It is set at the start of a data transfer. The Input Data Channel Master Lock Enabled flag is cleared after the required amount of data has been transferred (i.e. SHE_IMSTCNT is zero), or if the data transfer was canceled and current burst has been finished.
[0]	IMSTIDLE	Input Data Channel Master Idle flag This bit is high if no data transfer is ongoing. It is cleared at the start of a new data transfer. The Input Data Channel Master Idle flag is set when the data transfer to the input FIFO has completed, i.e. the Input Channel Master Data Transfer Counter (SHE_IMSTCNT) is zero, or if the data transfer has been canceled and the current burst finished. The rising edge of the IMSTIDLE flag sets the SHE_IRQ:IMSTIDLE interrupt flag.

3.5.3. Input Channel Master Error Response Address Register (SHE_IMSTERRADDR)

This register is updated in case of an AXI bus error response on the input data channel and contains the address of the slave which issued the error response. The register content is valid until the next error response occurs.

In a read transaction, the AXI slave can give different responses for different transfers within a burst. Hence the SHE_IMSTERRADDR register stores the precise address.

Bit	31	30	29	28	27	26	25	24
Field	IMSTERRADDR[31]	IMSTERRADDR[30]	IMSTERRADDR[29]	IMSTERRADDR[28]	IMSTERRADDR[27]	IMSTERRADDR[26]	IMSTERRADDR[25]	IMSTERRADDR[24]
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	IMSTERRADDR[23]	IMSTERRADDR[22]	IMSTERRADDR[21]	IMSTERRADDR[20]	IMSTERRADDR[19]	IMSTERRADDR[18]	IMSTERRADDR[17]	IMSTERRADDR[16]
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	IMSTERRADDR[15]	IMSTERRADDR[14]	IMSTERRADDR[13]	IMSTERRADDR[12]	IMSTERRADDR[11]	IMSTERRADDR[10]	IMSTERRADDR[9]	IMSTERRADDR[8]
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	IMSTERRADDR[7]	IMSTERRADDR[6]	IMSTERRADDR[5]	IMSTERRADDR[4]	IMSTERRADDR[3]	IMSTERRADDR[2]	IMSTERRADDR[1]	IMSTERRADDR[0]
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Table 3-24 Input Channel Master Error Response Address Register (SHE_IMSTERRADDR) Bits

Bit Position	Bit Field Name	Bit Description
[31:0]	IMSTERRADDR	Input Data Channel Error Response Address

3.5.4. Output Channel Master Error Response Address Register (SHE_OMSTERRADDR)

This register is updated in the event of the AXI bus error response on the output data channel and contains the address of the slave which issued the error response. The register content is valid until the next error response occurs.

For a write transaction, there is just one response given for the entire burst and not for each data transfer within the burst. Hence the SHE_OMSTERRADDR register stores the imprecise address (i.e. the start address of the burst).

Bit	31	30	29	28	27	26	25	24
Field	OMSTERR ADDR[31]	OMSTERR ADDR[30]	OMSTERR ADDR[29]	OMSTERR ADDR[28]	OMSTERR ADDR[27]	OMSTERR ADDR[26]	OMSTERR ADDR[25]	OMSTERR ADDR[24]
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	OMSTERR ADDR[23]	OMSTERR ADDR[22]	OMSTERR ADDR[21]	OMSTERR ADDR[20]	OMSTERR ADDR[19]	OMSTERR ADDR[18]	OMSTERR ADDR[17]	OMSTERR ADDR[16]
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	OMSTERR ADDR[15]	OMSTERR ADDR[14]	OMSTERR ADDR[13]	OMSTERR ADDR[12]	OMSTERR ADDR[11]	OMSTERR ADDR[10]	OMSTERR ADDR[9]	OMSTERR ADDR[8]
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	OMSTERR ADDR[7]	OMSTERR ADDR[6]	OMSTERR ADDR[5]	OMSTERR ADDR[4]	OMSTERR ADDR[3]	OMSTERR ADDR[2]	OMSTERR ADDR[1]	OMSTERR ADDR[0]
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Table 3-25 Output Channel Master Error Response Address Register (SHE_OMSTERRADDR) Bits

Bit Position	Bit Field Name	Bit Description
[31:0]	OMSTERRADDR	Output Data Channel Error Response Address

3.5.5. FIFO Status Register (SHE_FIFOSTATUS)

This register represents the current status of the input and output FIFOs.

Bit	31	30	29	28	27	26	25	24
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	COMPAREMATCH
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	OFIFORDTH
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	IFIFOWRTH
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Table 3-26 FIFO Status Register (SHE_FIFOSTATUS) Bits

Bit Position	Bit Field Name	Bit Description
[31:24]	Reserved	-
[23:17]	Reserved	-
[16]	COMPAREMATCH	Compare Match Event flag This bit signalizes that the required amount of data has been transferred through the input FIFO and its write port is locked. It is set automatically if the value of the SHE_DATACNT register is greater than or equal to the value of the SHE_COMPARE register and remains high until the start of the next command. Increasing the value of the SHE_COMPARE register above the value of the SHE_DATACNT register does not clear the asserted COMPAREMATCH flag.
[15:9]	Reserved	-
[8]	OFIFORDTH	Output FIFO Above Threshold flag This bit is set if the amount of data words stored in the output FIFO (SHE_FIFOLOAD:OFIFOLOAD) is greater than the value configured in the read threshold register (SHE_OFIFOCFG:RDTHRESHOLD). Otherwise this bit is cleared.

Bit Position	Bit Field Name	Bit Description
[7:1]	Reserved	-
[0]	IFIFOWRTH	Input FIFO Below Threshold flag This bit is set if the amount of data words stored in the input FIFO (SHE_FIFOLOAD:IFIFOLOAD) is less than the value configured in the write threshold register (SHE_IFIFOCFG:WRTHRESHOLD). Otherwise this bit is cleared.

3.5.6. FIFO Load Register (SHE_FIFOLOAD)

This register represents the current load status of the input and output FIFOs.

Bit	31	30	29	28	27	26	25	24
Field	Reserved	Reserved	OFIFOLOAD[5]	OFIFOLOAD[4]	OFIFOLOAD[3]	OFIFOLOAD[2]	OFIFOLOAD[1]	OFIFOLOAD[0]
R/W Attribute	R0,WX	R0,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	Reserved	Reserved	OFIFOFREE[5]	OFIFOFREE[4]	OFIFOFREE[3]	OFIFOFREE[2]	OFIFOFREE[1]	OFIFOFREE[0]
R/W Attribute	R0,WX	R0,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	0	0	1	1	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved	IFIFOLOAD[5]	IFIFOLOAD[4]	IFIFOLOAD[3]	IFIFOLOAD[2]	IFIFOLOAD[1]	IFIFOLOAD[0]
R/W Attribute	R0,WX	R0,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	Reserved	Reserved	IFIFOFREE[5]	IFIFOFREE[4]	IFIFOFREE[3]	IFIFOFREE[2]	IFIFOFREE[1]	IFIFOFREE[0]
R/W Attribute	R0,WX	R0,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	0	0	1	1	0	0	0	0

Table 3-27 FIFO Load Register (SHE_FIFOLOAD) Bits

Bit Position	Bit Field Name	Bit Description
[31:30]	Reserved	-
[29:24]	OFIFOLOAD	The amount of data stored in the output FIFO is defined in terms of 32- bit data words
[23:22]	Reserved	-
[21:16]	OFIFOFREE	The amount of data which can be written into the output FIFO is defined in terms of 32-bit data words
[15:14]	Reserved	-
[13:8]	IFIFOLOAD	The amount of data stored in the input FIFO is defined in terms of 32- bit data words
[7:6]	Reserved	-
[5:0]	IFIFOFREE	The amount of data which can be written into the input FIFO is defined in terms of 32-bit data words

3.5.7. Input FIFO Data Counter Register (SHE_DATACNT0 to 1)

This register represents the amount of 32-bit data words transferred through the input FIFO.

SHE_DATACNT is used by the hardware to recognize that the required amount of data for a single SHE command has been transferred through the input FIFO. For this purpose the value of the compare register SHE_COMPARE is continuously compared with the value of the data counter register SHE_DATACNT. When the data counter reaches the value of the compare register, the status bit SHE_FIFOSTATUS:COMPAREMATCH and an interrupt request bit SHE_IRQ:COMPAREMATCH are immediately set. This transition of the COMPAREMATCH bit is referenced to as a compare match event.

SHE_DATACNT is reset by the hardware to zero every time the opcode for the new SHE command is written to the SHE_CMD register.

Input FIFO Data Counter Register 0 (SHE_DATACNT0)

Bit	31	30	29	28	27	26	25	24
Field	DATACNT[31]	DATACNT[30]	DATACNT[29]	DATACNT[28]	DATACNT[27]	DATACNT[26]	DATACNT[25]	DATACNT[24]
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	DATACNT[23]	DATACNT[22]	DATACNT[21]	DATACNT[20]	DATACNT[19]	DATACNT[18]	DATACNT[17]	DATACNT[16]
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	DATACNT[15]	DATACNT[14]	DATACNT[13]	DATACNT[12]	DATACNT[11]	DATACNT[10]	DATACNT[9]	DATACNT[8]
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	DATACNT[7]	DATACNT[6]	DATACNT[5]	DATACNT[4]	DATACNT[3]	DATACNT[2]	DATACNT[1]	DATACNT[0]
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Table 3-28 Input FIFO Data Counter Register 0 (SHE_DATACNT0) Bits

Bit Position	Bit Field Name	Bit Description
[31:0]	DATACNT	Least significant bits of the data counter register The amount of data which has been transferred through the input FIFO is defined in terms of 32-bit data words.

Input FIFO Data Counter Register 1 (SHE_DATACNT1)

Bit	31	30	29	28	27	26	25	24
Field	Reserved	Reserved	Reserved	Reserved	Reserved	DATACNT[26]	DATACNT[25]	DATACNT[24]
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	DATACNT[23]	DATACNT[22]	DATACNT[21]	DATACNT[20]	DATACNT[19]	DATACNT[18]	DATACNT[17]	DATACNT[16]
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	DATACNT[15]	DATACNT[14]	DATACNT[13]	DATACNT[12]	DATACNT[11]	DATACNT[10]	DATACNT[9]	DATACNT[8]
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	DATACNT[7]	DATACNT[6]	DATACNT[5]	DATACNT[4]	DATACNT[3]	DATACNT[2]	DATACNT[1]	DATACNT[0]
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Table 3-29 Input FIFO Data Counter Register 1 (SHE_DATACNT1) Bits

Bit Position	Bit Field Name	Bit Description
[31:27]	Reserved	-
[26:0]	DATACNT	Most significant bits of the data counter register The amount of data which has been transferred through the input FIFO is defined in terms of 32-bit data words.

3.6. Data Transfer Registers

3.6.1. Input FIFO Write Data Register (SHE_IFIFOWRDATA0 to 31)

The input FIFO write data registers are used to write data to the input FIFO through the command interface. A write access to any SHE_IFIFOWRDATA0 to 31 register is directly mapped to the write port of the input FIFO.

For this reason effectively there is only one register which is replicated thirty-two times in the register map. This is done in order to increase performance of data transfers to the input FIFO by allowing burst transfers over the AHB bus (command interface of SHE). In other words, if several data words have to be transferred into the input FIFO of SHE through the command interface, this should be performed using burst transfer feature of the AHB bus, which is more efficient than several single transfers.

Only 32-bit accesses are allowed to these registers. 64-bit accesses are split into two 32- bit accesses. 8-bit or 16-bit accesses to these register trigger an error response.

The software should check the load status of the FIFO before writing to it. In case the FIFO is full, an error interrupt flag (SHE_IRQ:IFIFOWRERR) is set and the data is lost. Similarly, an error interrupt flag (SHE_IRQ:IFIFOLOCKERR) is set and the data is lost if a write access to the locked input FIFO is performed. An error response on the command interface (AHB bus) is generated in case of a write access to this register if the data interface is selected through the SHE_IFIFOCFG:IFSEL bit.

Bit	31	30	29	28	27	26	25	24
Field	IFIFOWRDATA[31]	IFIFOWRDATA[30]	IFIFOWRDATA[29]	IFIFOWRDATA[28]	IFIFOWRDATA[27]	IFIFOWRDATA[26]	IFIFOWRDATA[25]	IFIFOWRDATA[24]
R/W Attribute	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	IFIFOWRDATA[23]	IFIFOWRDATA[22]	IFIFOWRDATA[21]	IFIFOWRDATA[20]	IFIFOWRDATA[19]	IFIFOWRDATA[18]	IFIFOWRDATA[17]	IFIFOWRDATA[16]
R/W Attribute	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	IFIFOWRDATA[15]	IFIFOWRDATA[14]	IFIFOWRDATA[13]	IFIFOWRDATA[12]	IFIFOWRDATA[11]	IFIFOWRDATA[10]	IFIFOWRDATA[9]	IFIFOWRDATA[8]
R/W Attribute	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	IFIFOWRDATA[7]	IFIFOWRDATA[6]	IFIFOWRDATA[5]	IFIFOWRDATA[4]	IFIFOWRDATA[3]	IFIFOWRDATA[2]	IFIFOWRDATA[1]	IFIFOWRDATA[0]
R/W Attribute	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Table 3-30 Input FIFO Write Data Register 0 (SHE_IFIFOWRDATA0) Bits

Bit Position	Bit Field Name	Bit Description
[31:0]	IFIFOWRDATA	A 32-bit register for accessing the write port of the input FIFO over the Command Interface

3.6.2. Output FIFO Read Data Register (SHE_OFIFORDDATA0 to 31)

The output FIFO read data registers are used to read data from the output FIFO through the command interface. A read access to any SHE_OFIFORDDATA0 to 31 register is directly mapped to the read port of the output FIFO.

For this reason effectively there is only one register which is replicated thirty-two times in the register map. This is done in order to increase performance of data transfers from the output FIFO by allowing burst transfers over the AHB bus (command interface of SHE). In other words, if several data words have to be transferred from the output FIFO of SHE through the command interface, this should be performed using the burst transfer feature of the AHB bus, which is more efficient than several single transfers.

Only 32-bit accesses are allowed to these registers. 64-bit accesses are split in two 32-bit accesses. 8-bit or 16-bit accesses to these register trigger an error response.

The software should check the load state of the FIFO before reading from it. In case the FIFO is empty, an error interrupt flag (SHE_IRQ:OFIFORDERR) is set and default data is read. An error response is generated in case of read access to this register if data interface is selected through SHE_OFIFOCFG:IFSEL bit.

Bit	31	30	29	28	27	26	25	24
Field	OFIFORDDATA[31]	OFIFORDDATA[30]	OFIFORDDATA[29]	OFIFORDDATA[28]	OFIFORDDATA[27]	OFIFORDDATA[26]	OFIFORDDATA[25]	OFIFORDDATA[24]
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	OFIFORDDATA[23]	OFIFORDDATA[22]	OFIFORDDATA[21]	OFIFORDDATA[20]	OFIFORDDATA[19]	OFIFORDDATA[18]	OFIFORDDATA[17]	OFIFORDDATA[16]
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	OFIFORDDATA[15]	OFIFORDDATA[14]	OFIFORDDATA[13]	OFIFORDDATA[12]	OFIFORDDATA[11]	OFIFORDDATA[10]	OFIFORDDATA[9]	OFIFORDDATA[8]
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	OFIFORDDATA[7]	OFIFORDDATA[6]	OFIFORDDATA[5]	OFIFORDDATA[4]	OFIFORDDATA[3]	OFIFORDDATA[2]	OFIFORDDATA[1]	OFIFORDDATA[0]
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Table 3-31 Output FIFO Read Data Register 0 (SHE_OFIFORDDATA0) Bits

Bit Position	Bit Field Name	Bit Description
[31:0]	OFIFORDDATA	A 32-bit register for accessing the read port of the output FIFO over the Command Interface

4. Operation of the Secure Hardware Extension

This chapter describes the operation of the SHE module in detail.

4.1. Operation of the Command Interface

Command interface of SHE is mainly used for accessing configuration and status registers of SHE . Additionally it can also be used for the transfer of parameters and data that are used to process commands defined in the SHE Functional Specification.

Feature List

Features of the command interface are:

- Provides configuration and status registers for SHE
- 32-bit Advanced High-Performance Bus (AHB) slave interface
- Command register locked during command execution
- There is a dedicated register for starting the CMD_CANCEL
- FIFO based transfer of parameters, processing data and results of command execution
- Support for interrupt-based software design
- Access protection through Peripheral Protection Unit (PPU) and master ID filter

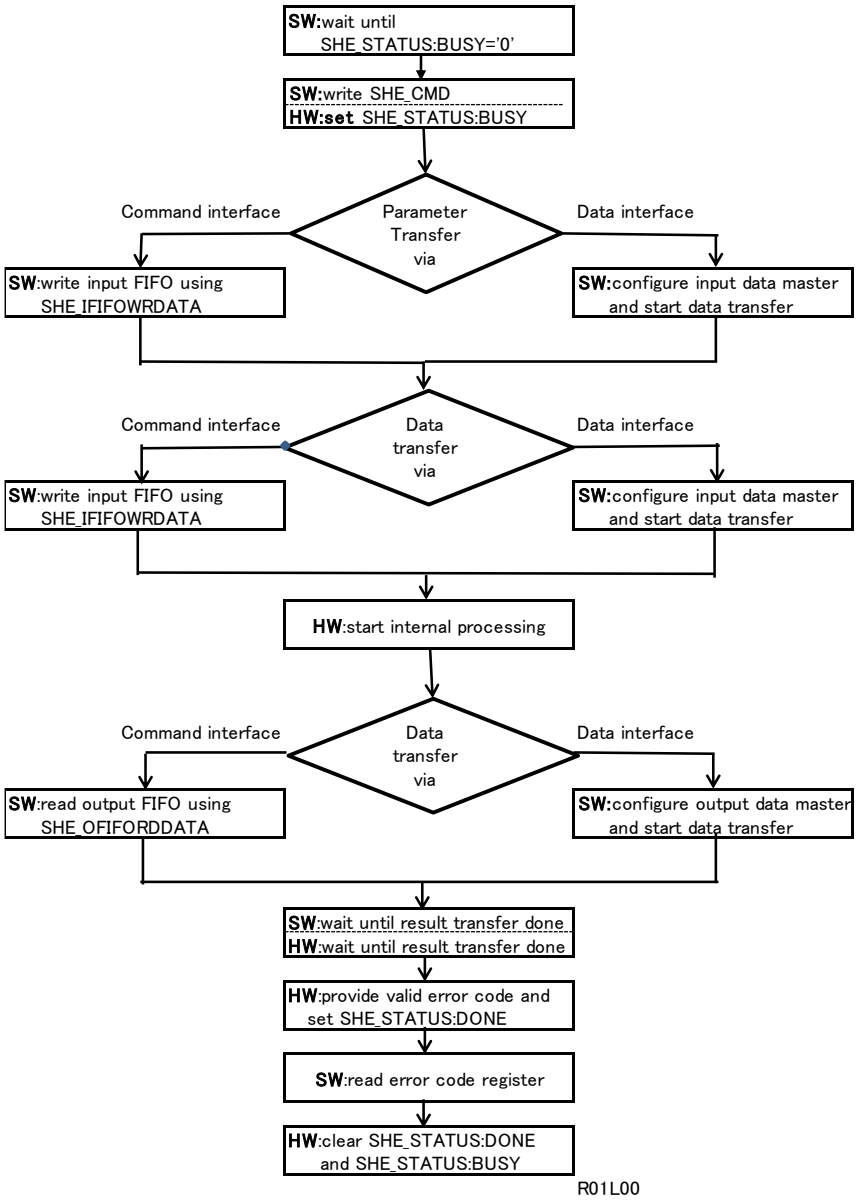
Command Execution

The following section describes the execution of SHE API commands inside the SHE module. All commands are defined in the SHE Functional Specification (Chapter 7) and can be separated into two groups: sequential and concurrent. As the name implies, sequential commands can only be executed sequentially, i.e. one command must finish before the next one starts. Concurrent commands can be executed in parallel to other commands, i.e. they can be started at an arbitrary point in time.

Sequential Command Execution

The processing of a sequential SHE command is presented in Figure 2-1 and is described in the following.

Figure 4-1 Flow Chart for the Execution of a Sequential SHE Command



A sequential SHE command can only be started if SHE is in idle mode, i.e. the BUSY bit in the SHE status register (SHE_STATUS) is low. In order to start the execution of the required SHE command, the user has to write the command opcode into the command register (see Section 3.1.1 SHE Command Register (SHE_CMD) description). The commands are defined in the SHE Functional Specification (Chapter 7) and listed in Table 4-1 below.

Note:

- *The concurrent commands CMD_CANCEL and CMD_GET_STATUS are not present in the overview table. CMD_CANCEL has its own command register (SHE_CMDCANCEL). CMD_GET_STATUS just performs a read access to the status register of SHE (SHE_STATUS) and therefore doesn't need to write an opcode to the command register.*

Table 4-1 Sequential SHE Commands Started Through the SHE_CMD Register

CMD_ENCRYPT_ECB
CMD_ENCRYPT_CBC
CMD_DECRYPT_ECB
CMD_DECRYPT_CBC
CMD_GENERATE_MAC
CMD_VERIFY_MAC
CMD_LOAD_KEY
CMD_LOAD_PLAIN_KEY
CMD_EXPORT_RAM_KEY
CMD_INIT_RNG
CMD_EXTEND_SEED
CMD_RND
CMD_SECURE_BOOT
CMD_BOOT_FAILURE
CMD_BOOT_OK
CMD_GET_ID
CMD_DEBUG

Refer to the description of the SHE Command Register (SHE_CMD) register for the opcode values.

As already mentioned, writing command opcode to the SHE_CMD register starts command execution. The busy flag (SHE_STATUS:BUSY) is set and the command register lock is enabled thereupon. A locked command register disables write access to it and protects it from being overwritten by accident or by malicious software. This lock is kept enabled until the complete processing has finished, i.e. as long as the busy flag (SHE_STATUS:BUSY) is high. While the command register is locked only read accesses are allowed. Any write access will trigger a bus error response.

After starting the command execution, the user has to provide the parameters and data which are required for the processing using the input FIFO of SHE. The input FIFO can be written either through the command interface (see Section 2.6.1 - SHE_IFIFOWRDATA register) or through the SHE data interface. Data transfer through the data interface is done using the input data master, which is capable of performing autonomous memory accesses.

The decision concerning which interface to use should be based on the amount of parameters and data which are required for processing. If only a few data transfers are required, it is more reasonable to use the SHE command interface because of the overhead due to the configuration of the input data master. If SHE is utilized for the processing of bigger data streams, e.g. CBC or Media Access Control (MAC) commands, the data interface input data master should be taken and configured accordingly. Refer to the description of the data interface for more information.

Under normal circumstances both FIFOs are empty after command execution. But there are some error scenarios that may lead to inconsistent FIFO states (e.g. more data than required was fed to the input FIFO). Hence, in order to avoid incorrect behavior from SHE or faulty computation results if some data is still present in the FIFOs, both FIFOs are cleared at the start a new command.

SHE expects the delivery of parameters and data for processing in a pre-defined order. This order as well as the order in which results are generated is shown in Table 3-2. The following commands are not presented in the table since they neither require any parameters or data for processing nor generate results:

- CMD_INIT_RNG
- CMD_BOOT_FAILURE
- CMD_BOOT_OK
- CMD_CANCEL
- CMD_GET_STATUS

In Table 3-2 "Sequential order for parameter and result transfer", please note the reversed parameter and result order for the CMD_DEBUG command: the CHALLENGE value is first provided by SHE and the AUTHORIZATION value is expected afterwards.

Each parameter has to be provided in one or several 64-bit large data blocks. Therefore, parameters with lengths less than 64 bits have to be padded with leading zeros and those wider than 64 bits have to be split into 64-bit large data blocks. The least significant part of the parameter must be transferred first. The same approach is also valid for SHE command results: small results are padded and large results are split.

Table 4-2 Sequential Order for Parameter and Result Transfer

Command	Parameter			Return Value		
	Seq. No.	Name	Size [Bits]	Seq. No.	Name	Size [Bits]
CMD_ENC_ECB	1 2	KEY_ID PLAINTEXT	64 128	1	CIPHERTEXT	128
CMD_ENC_CBC	1	KEY_ID	64			
	2	IV	128			
	3	PLAINTEXT	n x 128	1	CIPHERTEXT	n x 128
CMD_DEC_ECB	1	KEY_ID	64			
	2	CIPHERTEXT	128	1	PLAINTEXT	128
CMD_DEC_CBC	1	KEY_ID	64			
	2	IV	128			
	3	CIPHERTEXT	n x 128	1	PLAINTEXT	n x 128
CMD_GENERATE_MAC	1	KEY_ID	64			
	2	MESSAGE_LENGTH	64			
	3	MESSAGE	n^1 x 128	1	MAC	128
CMD_VERIFY_MAC	1	KEY_ID	64			
	2	MESSAGE_LENGTH	64			
	3	MAC_LENGTH	64			
	4	MAC	128			
	5	MESSAGE	n^1 x 128	1	VERIFICATION_STATUS	64
CMD_LOAD_KEY	1	M1	128			
	2	M2	256			
	3	M3	128	1	M4	256
				2	M5	128
CMD_LOAD_PLAIN_KEY	1	KEY	128		N/A	N/A
CMD_EXPORT_ RAM_KEY		N/A	N/A	1	M1	128
				2	M2	256
				3	M3	128
				4	M4	256
				5	M5	128
CMD_EXTEND_SEED	1	ENTROPY	128		N/A	N/A
CMD_RND		N/A	N/A	1	RND	128
CMD_SECURE_BOOT[2]	1 2	SIZE DATA	64 n^3 x 128		N/A	N/A
CMD_GET_ID	1	CHALLENGE	128	1	ID |[4] SREG	120 + 8
				2	MAC	128
CMD_DEBUG	1	AUTHORIZATION	128	1	CHALLENGE	128

[1] : n = ceil[5] (MESSAGE_LENGTH / 128)

[2] : CMD_SECURE_BOOT is always started by the Boot ROM of the host MCU and is not directly available for the user

[3] : n = ceil[5] (SIZE / 16)

[4] : Vertical bar ('|') symbol represents concatenation of two values. Left value is kept in most significant, right value in least significant bits.

[5] : ceil(x) = min{n∈Z_1 | n ≥ x}, Z_1 = {1,2,3,...}

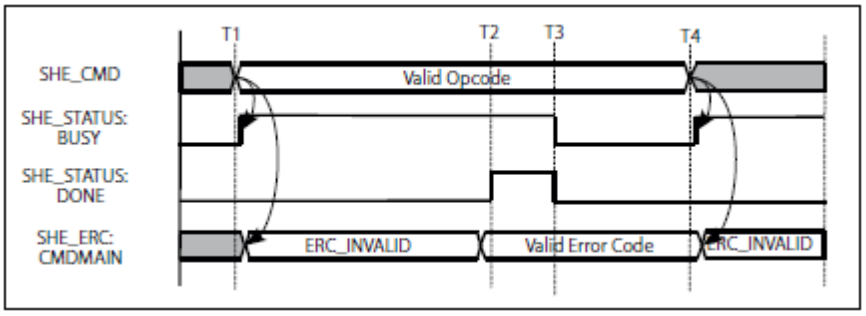
The processed results from SHE are written to the output FIFO in the order given in Table 4-2 above and can be read by the user either through the SHE command interface - see the Output FIFO Read Data Register (SHE_OFIFORDDATA0 to 31) - or data interface using the output data master. Similar to the input data master, the output data master should be used if many data transfers are required.

Once the complete result data has been read from the output FIFO, the SHE control logic writes valid error code to the error code register (SHE_ERC:CMDMAIN) and indicates the end of processing by setting the DONE status flag in the status register (SHE_STATUS). For the commands which require no parameters or do not generate results, as well as for the CMD_DEBUG command, the error code register (SHE_ERC:CMDMAIN) is updated and the SHE_STATUS:DONE flag is set once internal processing has completed.

Command execution is finished only when the software reads the error code register using a 32-bit data access. The SHE_STATUS:BUSY and SHE_STATUS:DONE flags are cleared thereupon.

Figure 4-2 illustrates the timing flow for the execution of a sequential SHE command. The command is started at moment T1 by a write access to the SHE_CMD register. Internal processing is finished at moment T2 and the error code register is read at moment T3. A new command can be started at any time after T3, for example at moment T4.

Figure 4-2 Timing Diagram for Execution of a Sequential SHE Command



Special Case Handling for CBC Operations

The majority of SHE commands have a fixed number of parameters, and the amount of data used for processing is also fixed. Hence SHE control logic knows the required amount of information which has to be transferred through the input FIFO. This value is stored in a special compare register - see Input

FIFO Compare Value Register (SHE_COMPARE0 to 1) for more details - which is configured through the SHE control logic upon the start of a command.

For two CBC commands, CMD_ENC_CBC and CMD_DEC_CBC, the amount of data used for processing is not always known before command execution (i.e. a priori). Hence command execution is started using the default value of SHE_COMPARE. Once the required amount of data is known, the software has to update the value of the SHE_COMPARE register. For example if SHE is used for a CBC command on a 1024-bit data stream, the total amount of data is 1,216 bits (64 bits for the key ID, 128 bits for initialization vector and 1,024 bits for the data itself). And since SHE_COMPARE represents data in terms of 32-bit data words, it has to be set by the software to 38 (1216 divided by 32).

Finally, the software has two possibilities of finishing an ongoing CBC command: adjusting the value of the SHE_COMPARE register; or issuing the CMD_CANCEL command. If CMD_CANCEL is used, the software has to make sure that the required amount of data has been processed in SHE and the results transferred from the SHE output FIFO.

Note:

- *The software is only allowed to change the value of SHE_COMPARE if a CBC command is performed. Otherwise a bus error response is generated.*

Concurrent Command Execution

CMD_GET_STATUS

The current status of SHE is provided in SHE_STATUS register which is always kept up to date by the SHE control logic. Since this register is directly accessible by the software, no special handling of the CMD_GET_STATUS command is required by SHE.

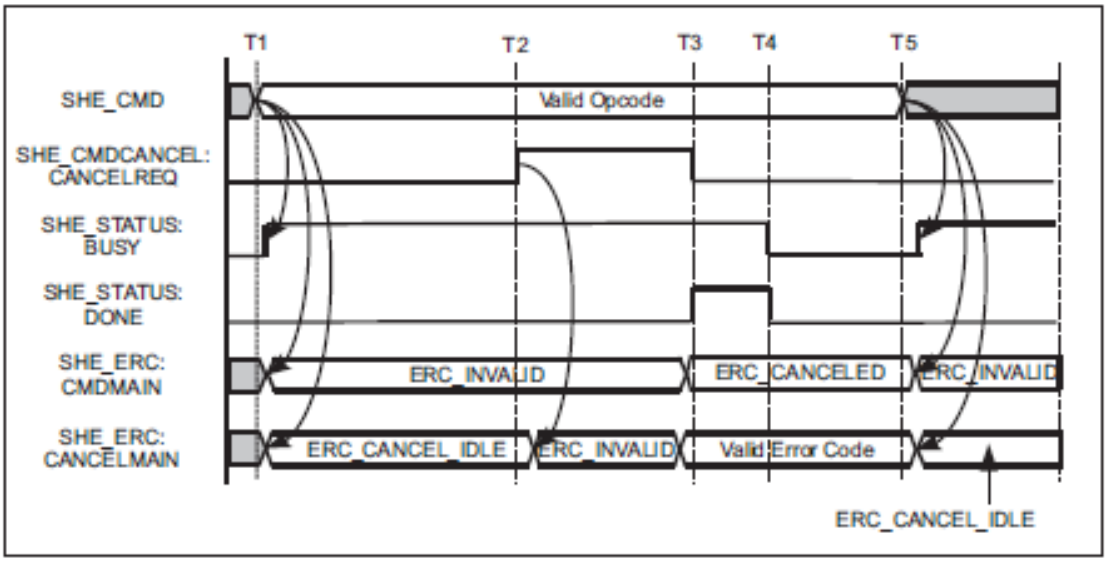
CMD_CANCEL

As already stated, execution of the CMD_CANCEL command is not started using the regular SHE command register but instead with a dedicated command cancel register (SHE_CMDCANCEL). To cancel an ongoing command the user has to set the SHE_CMDCANCEL:CANCELREQ bit to "1". Once this is detected by the SHE control logic, the internal processing of the CMD_CANCEL command starts. This is indicated by the change of error code in the error code register SHE_ERC for the sequential command (SHE_ERC:CMDMAIN) and command cancel (SHE_ERC:CANCELMAIN) to ERC_INVALID. Furthermore, similar to the processing of a sequential command, the busy flag (SHE_STATUS:BUSY) is kept high and the done flag (SHE_STATUS:DONE) is kept low during the processing of the CMD_CANCEL command. Additionally, the command register SHE_CMD is locked for writing while CMD_CANCEL is being processed, i.e. while SHE_STATUS:BUSY is high.

Once the internal processing is finished, the SHE control logic updates SHE_ERC and sets the SHE_STATUS:DONE flag. The error code for the canceled sequential command (SHE_ERC:CMDMAIN) is set to ERC_CANCELED and the error code for the command cancel (SHE_ERC:CANCELMAIN) is set to a valid value defined by the SHE Functional Specification. The user application has to read the error code register using 32-bit data access in order to complete command execution. The SHE_STATUS:BUSY and SHE_STATUS:DONE flags are then cleared.

Figure 4-3 Execution of CMD_CANCEL Command illustrates the timing flow for the execution of the CMD_CANCEL command.

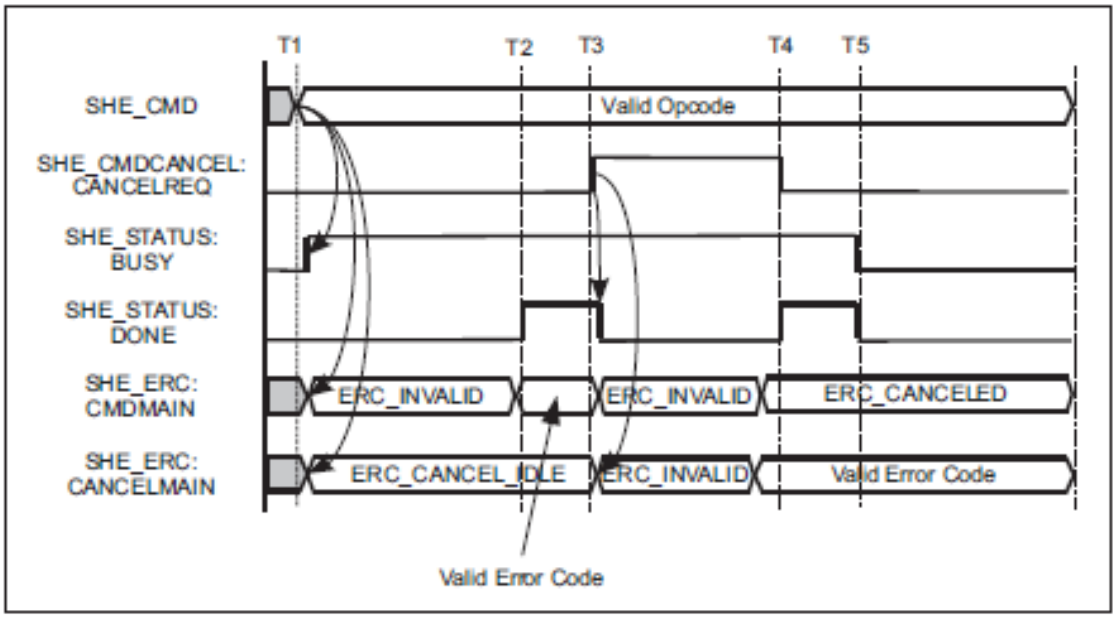
Figure 4-3 Execution of CMD_CANCEL Command



The cancel request for the command which was started at moment T1 is detected at moment T2. Internal processing of CMD_CANCEL is finished at moment T3 and the error code register is read at moment T4. A new command can be started at any time after T4, for example at moment T5.

Figure 4-4 illustrates the timing flow for the execution of a CMD_CANCEL command issued after the end of the internal data processing in SHE, i.e. when the valid error code (SHE_ERC:CMDMAIN) is available and the done flag (SHE_STATUS:DONE) is high.

Figure 4-4 Execution of CMD_CANCEL Command Issued after the Previous Command Has Completed



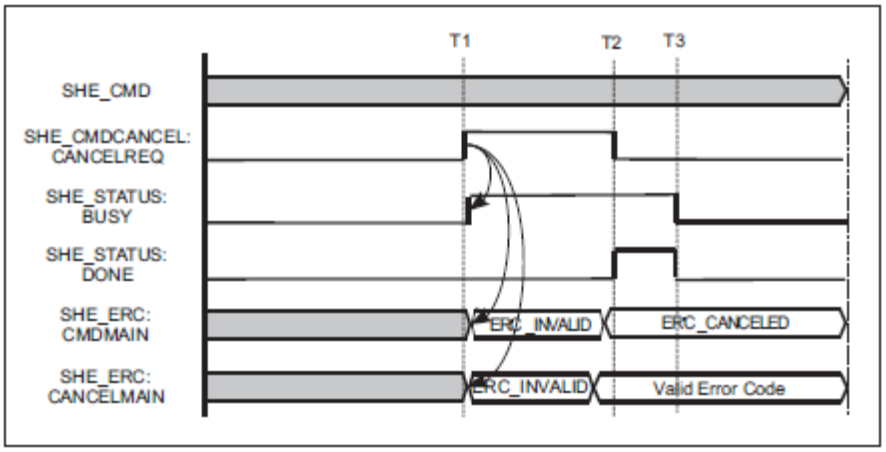
The internal processing of the sequential command, which was started at moment T1, is completed at moment T2. The cancel request is issued at moment T3. Both error codes (SHE_ERC:CMDMAIN and SHE_ERC:CANCELMAIN) are reset and the done flag (SHE_STATUS:DONE) is then cleared by the hardware. Internal processing of CMD_CANCEL is completed at moment T4 and the error code register is read at moment T5.

The CMD_CANCEL command can also be issued when no command is being executed. The appropriate timing diagram is shown in Figure 4-5. The cancel request is issued at moment T1. Both error codes, SHE_ERC:CMDMAIN and SHE_ERC:CANCELMAIN, are reset, the SHE_STATUS:BUSY flag is set and the SHE_STATUS:DONE flag is kept low by the hardware. Internal processing of CMD_CANCEL is completed at moment T2.

Note:

- *Only the error code for CMD_CANCEL (SHE_ERC:CANCELMAIN) is updated. The error code register for the sequential command (SHE_ERC:CMDMAIN) keeps its value (ERC_INVALID).*

Figure 4-5 Execution of CMD_CANCEL Command When No Command Execution Is Ongoing.



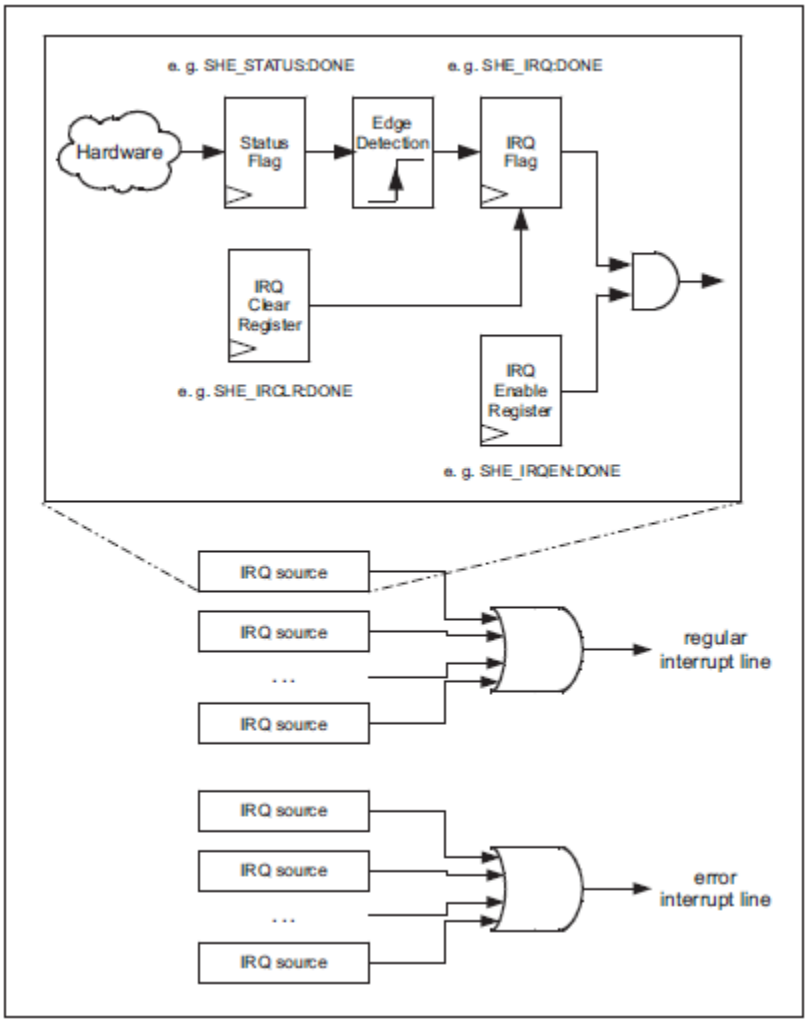
Interrupt Request Generation

In order to support interrupt-based application design, SHE provides two separate interrupt request lines to signal changes to its internal state. The first interrupt line is used for regular interrupts and is triggered by regular events that are expected during command processing. For example, the rising of the done status flag (SHE_STATUS:DONE) triggers the regular interrupt request. The second interrupt line is used to signal extraordinary events, i.e. errors.

Both interrupts are shared interrupts, i.e. each interrupt line has several interrupt sources. To distinguish between interrupt sources the user has to perform a read access to the SHE Interrupt Request register SHE_IRQ. This register represents all interrupts available in SHE and is divided into two parts: regular and error interrupts. All interrupts can be enabled and cleared independently of each other using the SHE_IRQEN and SHE_IRQCLR registers respectively. The architecture of the interrupt circuitry used in SHE is shown in Figure 4-6 In general it consists of the following parts:

- Status flags that represent the actual state and which change accordingly with it. For example, the DONE flag in the SHE_STATUS register.
- Edge detection logic. The rising edge of the status flag sets the interrupt request flag.
- Interrupt request flags representing pending interrupt request. Each flag is set by the rising edge of the corresponding status flag and remains high until it is cleared by the software, e.g. the DONE flag in SHE_IRQ register.
- IRQ enable bits. If written to "1", the corresponding interrupt request stored in the SHE_IRQ register is propagated to the appropriate interrupt line. For example, the SHE_IRQEN:DONE bit enables the generation of a SHE interrupt for the SHE_STATUS:DONE bit.
- The interrupt request flag should be cleared by the software during the execution of its interrupt service routine. This is achieved by writing "1" to the appropriate bit in the interrupt request clear register, i.e. SHE_IRQCLR:DONE.

Figure 4-6 Block Diagram of SHE Interrupt Generation Circuitry



Note:

- *All IRQ source circuits are identical to the one shown in detail.*

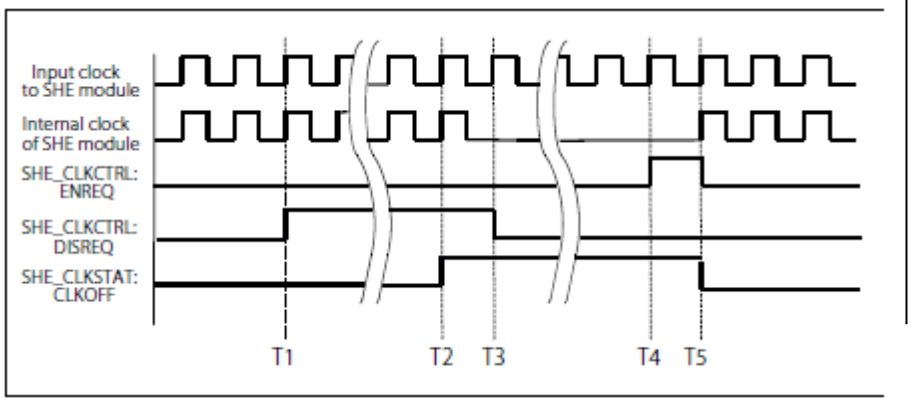
As already mentioned, several interrupt sources are combined and the resulting shared interrupt lines (one for regular and one for error interrupts) are then connected to the interrupt controller.

4.1.1. Power Saving Functionality

Clock Gating for SHE

The power consumption of SHE can be reduced by disabling its internal clock. Enabling/disabling the SHE clock can be performed using the SHE_CLKCTRL register. The current status of the clock is given by the SHE_CLKSTAT register. The procedure for clock gating is presented in Figure 4-7.

Figure 4-7 SHE Clock Disabling/Enabling



In order to disable the SHE clock the user has to set the SHE_CLKCTRL:DISREQ bit to "1" (T1). Please note that access to the SHE_CLKCTRL register can occur at an arbitrary point in time, but the SHE clock can only be disabled if SHE is in an idle state (i.e. SHE_STATUS:BUSY is "0"). Hence the pending clock disable request is kept high until SHE moves to the idle state and performs the internal operations required for clock gating. The CLKOFF bit in the SHE_CLKSTAT register is then set (T2). The pending disable request SHE_CLKCTRL:DISREQ is automatically cleared by the SHE hardware, and the SHE clock is disabled on the next cycle of the input clock (T3).

The clock is enabled by writing "1" to the SHE_CLKCTRL:ENREQ bit (T4). SHE_CLKCTRL:ENREQ and SHE_CLKSTAT:CLKOFF are cleared after which the internal clock is enabled (T5).

External debugger detection is active even if the SHE clock is switched off. More precisely, the external debugger will be detected if it is attached during SHE power-saving mode, and the SHE_STATUS:EXTDEBUGGER bit will be set after the SHE clock has been enabled.

Note:

- *Only the SHE_CLKCTRL and SHE_CLKSTAT registers are accessible if the internal SHE clock is disabled. Accessing other SHE registers will generate a bus error response.*

System Wide Clock/Power Gating

The status register bit SHE_STATUS:BOOTOK is automatically cleared whenever the CPU or a memory of the MCU enters power saving mode. This means for a TRAVEO™ T1G series device, SHE is notified if any clock is gated for at least one of the following modules (the appropriate clock name is given in parentheses):

- Cortex-R5 CPU (CLK_CPU)
- TCFLASH (CLK_FLCK, CLK_MEMC)

- WFLASH (CLK_FLCK, CLK_MEMC)
- System RAM (CLK_HPM)
- TCRAM (CLK_CPU, CLK_MEMC)

Note:

- *Only falling edges of the clock/power enable signals for the modules listed above are taken into account. Thus the SHE_STATUS:BOOTOK bit will be cleared only when the clock/power is disabled.*

4.1.2. Command Interface Access Issues

A master ID filter is also implemented. This means that only two specified masters, the CPU and debugger, are allowed to access the SHE command interface.

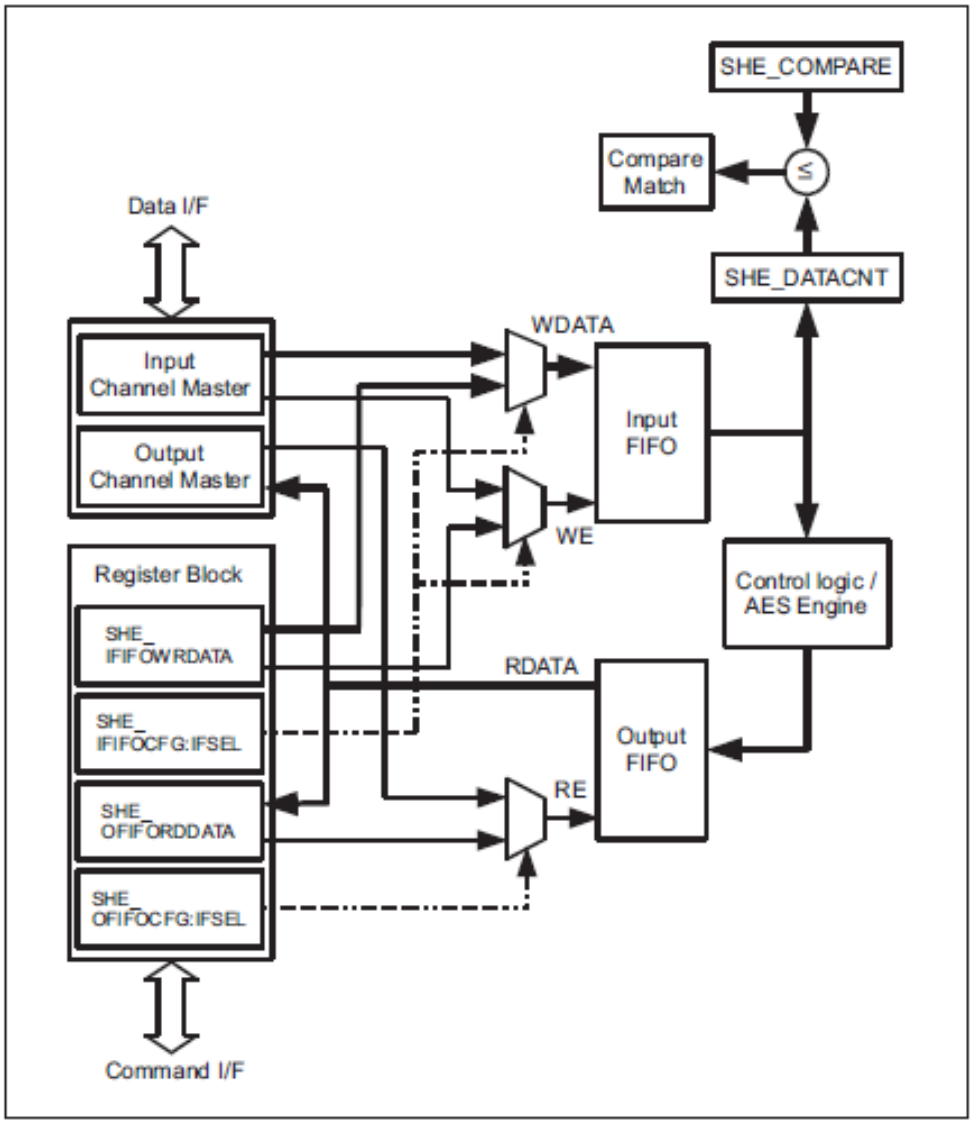
Since the command interface can also be utilized for the transfer of parameters and data used for processing, special registers exist for accessing the write port of the input FIFO (refer to the Input FIFO Write Data Register (SHE_IFIFOWRDATA0 to 31) description) and read port of the output FIFO (refer to the Output FIFO Read Data Register (SHE_OFIFORDDATA0 to 31) description). An error interrupt flag is set (refer to the Interrupt Request Register (SHE_IRQ) and bits IFIFOWRERR and OFIFORDERR) if write access to a full input FIFO or read access to an empty output FIFO is performed over the command interface. Moreover, an error response on the AHB bus is generated in the event of a write access to SHE_IFIFOWRDATA if the input FIFO is locked (see Section n "Input FIFO"). Hence the state of the FIFOs has to be checked before performing the accesses.

4.2. Operation of the Data Interface

All data required for the execution of a single SHE API command as well as results of commands can be transferred over the SHE data interface. This interface is controlled by two data master modules which transfer data to/from the attached FIFOs. The data flow diagram is shown in Figure 4-8.

By default the data transfer is done over the command interface, but alternatively the data can also be transferred over the data interface using data masters. Refer to SHE_IFIFOCFG:IFSEL, SHE_OFIFOCFG:IFSEL, SHE_IFIFOWRDATA and SHE_OFIFORDDATA register descriptions.

Figure 4-8 SHE Data Flow Diagram



Feature List

The features of the data interface are:

- A 64-bit AXI master interface
- Input and output channels for independent read and write data transfers
- Only incremental bursts to 64-bit aligned addresses are supported
- Direct memory access capability, which covers the complete address range of the MCU (32-bit)
- Configuration settings lock during transfer
- FIFOs for synchronization and data transfer between several parties with different data-path widths
- Support for interrupt-based software design

Operation of Data Master modules

The input and output channel masters are responsible for performing data transfers to and from SHE. They can perform independent read and write transfers between different memory locations of the MCU and SHE. In order to increase data transfer performance, both masters always try to perform burst transfers of the maximal possible length.

Both data masters have the same set of configuration registers:

- Start address in memory where the master has to read data from or store to. Refer to the Input Channel Master Start Address Register (SHE_IMSTADDR) and Output Channel Master Start Address Register (SHE_OMSTADDR) descriptions. The address has to be 64-bit aligned. The start address should also be aligned to the 128-byte boundary if the transfer is supposed to cross the 4KB address boundary. Otherwise at least one short burst transfer (on reaching the 4KB boundary) will be performed.
- The total number of transfers required for a single command. Refer to the Input Channel Master Data Transfer Counter (SHE_IMSTCNT) and the Output Channel Master Data Transfer Counter (SHE_OMSTCNT) register descriptions. This value, which has to be preloaded by software, is the number of 64-bit double words that have to be transferred. This value is decremented with every transfer and represents the amount of remaining data transfers.
- Data transfer start trigger. Refer to Input Channel Master Start Trigger (SHE_IMSTSTART) and Output Channel Master Start Trigger (SHE_OMSTSTART) registers. The transfer can only be started if the master idle flag (IMSTIDLE or OMSTIDLE in the SHE_MSTSTATUS register) is high. Trying to start the transfer while the master idle flag is low has no effect. Once started the transfer proceeds as long as the number of remaining transfers (SHE_*MSTCNT) is greater than zero.

The configuration settings can be written before or after the start of the command (write access to the SHE_CMD register). The data transfer itself (through the SHE_*MSTSTART register) should only be started after the start of the command. Otherwise the data which may have already been transferred into the input FIFO will be discarded because the FIFO is flushed upon the start of the command.

Note:

- *If the start of a new data transfer is requested (i.e. a write access to the SHE_*MSTSTART register) and the interface setting for the appropriate FIFO is 'Command Interface' - refer to either the Input FIFO Configuration Register (SHE_IFIFOCFG) or Output FIFO Configuration Register (SHE_OFIFOCFG) - which is in fact an error condition, the value of the SHE_*MSTSTART register remains zero but the SHE_IRQ:*MSTIFSELERR error interrupt flag is set instead and the transfer is not started.*

To prevent the manipulation of settings, the appropriate configuration registers are automatically locked after starting a data transfer. An enabled lock is signaled through the IMSTLOCK or OMSTLOCK bit in the SHE_MSTSTATUS register. Writing access to locked registers produces a bus error response. The lock is released either if the required amount of data has been transferred (SHE_*MSTCNT is zero) or if the data transfer was canceled in case of an error or CMD_CANCEL. The complete overview of lockable registers is given in Table 3-3 .

Table 4-3 Lockable Registers List

	Input Channel Master	Output Channel Master
Lock bit	SHE_MSTSTATUS:IMST-LOCK	SHE_MSTSTATUS:OMSTLOCK
Locked Registers	SHE_IFIFOCFG:IFSEL SHE_IMSTADDR SHE_IMSTCNT	SHE_OFIFOCFG:IFSEL SHE_OMSTADDR SHE_OMSTCNT

As already mentioned in Section "Sequential command execution", the decision concerning which interface should be selected for data transfer depends on the amount of data. The software may use only one, both or even no data masters. Also a combination of command interface and data interface is possible; the parameters can be transferred through the command interface and the subsequent data required for processing can be transferred by the input channel master. The results can also be read through the command interface or by the output channel master.

Behavior in the Event of a CMD_CANCEL Command

Both data masters feature special stop request input signals, which are controlled by the SHE control logic and can be set in order to cancel an ongoing data transfer. These signals will be asserted in the event of the pending CMD_CANCEL command.

If the stop request is detected by a data master, it completes any ongoing burst but cancels all subsequent ones. The ongoing burst is completed as follows: the input channel master reads the remaining data from the AXI bus but discards it and doesn’t write it into input FIFO. The output channel master stops reading the output FIFO and writes the default data to the bus (0x0...0), but AXI byte strobes are disabled.

Note:

- *The transfer counter (SHE_*MSTCNT) only counts regular transfers, i.e. the counter value is not changed if the data transfer has been canceled and "dummy" transfers are performed.*

After finishing the current burst, the master moves to an idle state and asserts the appropriate master idle flag (SHE_MSTSTATUS:*MSTIDLE). No new data bursts will be started until a new command is issued. The transfer counter (SHE_*MSTCNT) value is kept, allowing the software to determine the amount of data transferred.

Error Handling

Two kinds of errors can occur during the operation of the data masters:

1. Write access from the input channel master to the locked input FIFO.

   The input FIFO is locked when the required amount of data has been transferred (see Section 3.3 "Operation of the FIFO unit"). However, misconfiguration may cause the input channel master to execute write accesses to the input FIFO. In this case an error interrupt flag (SHE_IRQ:IFIFOLOCKERR) is set to notify the software of this error.

   Upon error detection, the input data master behaves in the same way as if it were processing a stop request (see ‘Behavior in the event of a CMD_CANCEL command’), i.e. an ongoing burst is completed without transferring the data to the input FIFO and all subsequent bursts are canceled. The output data master is not affected in this case and continues its operation.

2. Bus error response.

   An AXI slave can generate an error response in the event of an error during data transfer. Since SHE cannot resolve bus errors, the software should be informed about it. This will be done by setting the status flags (SHE_MSTSTATUS:IMSTERR, SHE_MSTSTATUS:OMS- TERR) and appropriate error interrupt flags - SHE_IRQ:IMSTERR for the input channel master and SHE_IRQ:OMSTERR for the output channel master. Besides the interrupt, the AXI response code (IMSTERRRESP and OMSTERRRESP in SHE_MSTSTATUS) and the error address (SHE_IMSTERRADDR and SHE_OMSTERRADDR) are stored.

According to the AXI protocol definition for a read transaction, the slave can give different responses for different transfers within a burst. Hence, on receiving an error response the input channel master behaves in the same way as if it were processing a stop request (see ‘Behavior in the event of a CMD_CANCEL command’), i.e. an ongoing burst is completed without transferring the data to the input FIFO, and all subsequent bursts are canceled.

According to the AXI protocol definition for a write transaction, there is just one response given for the entire burst and not for each data transfer within the burst. Hence, on receiving an error response the output channel master does not have to complete any ongoing burst and just cancels any outstanding transfers.

Error responses on both data channels are also signaled to the SHE control logic, which will cancel the current command execution and return an error code thereupon.

System integration issues

Memory Protection

For safety reasons both masters have a Memory Protection Unit (MPU) in order to protect the memory areas of the MCU. For details on MPU configuration, refer to Chapter: Memory Protection Unit for the AXI of SHE.

Instruction Flash Addressing

Instruction flash (TCFLASH) is separated in two parts (beginning at start address): small sectors and large sectors. Independent on the start position it is possible for the input channel master of SHE to address the complete memory area, e. g. for a validation of complete TCFLASH. By default the input channel master jumps from the last address of small sectors to the first address of large sectors. Furthermore an address wrap around occurs if input channel master reaches end of large sectors. In this case read accesses continue at the start address of small sectors. This feature together with the

previous one allow complete TCFLASH addressing independently from the starting location (128bit aligned).

Note:

- *The approach described above only applies if a valid address in the TCFLASH (not a reserved area) is given in the SHE_IMSTADDR register. This means if SHE_IMSTADDR points to a reserved area of the TCFLASH, no special action is performed and the usual Error Handling is done.*

Notes on the Operation of the Data masters

The data interface implements the AXI protocol using a 32-bit address and 64-bit data buses. Input and output data transfers can be carried out simultaneously by performing up to 16-beat long incremental bursts. This means up to 16 x 64 = 1,024 bits of data can be transferred in one burst. In order to simplify the design, no unaligned accesses are allowed, i.e. all addresses have to be aligned to the 64-bit boundary.

The number of transfers in a single burst is determined by the value of the total transfer counter (see Section 2.4.3 - SHE_IMSTCNT and Section 2.4.4 - SHE_OMSTCNT). While this value is greater than or equal to 16, the master issues bursts with a maximum length of 16. Once the number of remaining transfers becomes less than 16, this value will be taken to perform the last burst.

Due to the limitation of the AXI protocol, which cannot perform bursts over the 4KB address boundary, the master has to reduce the size of the burst if it is about to cross the 4KB boundary. All consecutive bursts will be performed using the approach described in the previous paragraph.

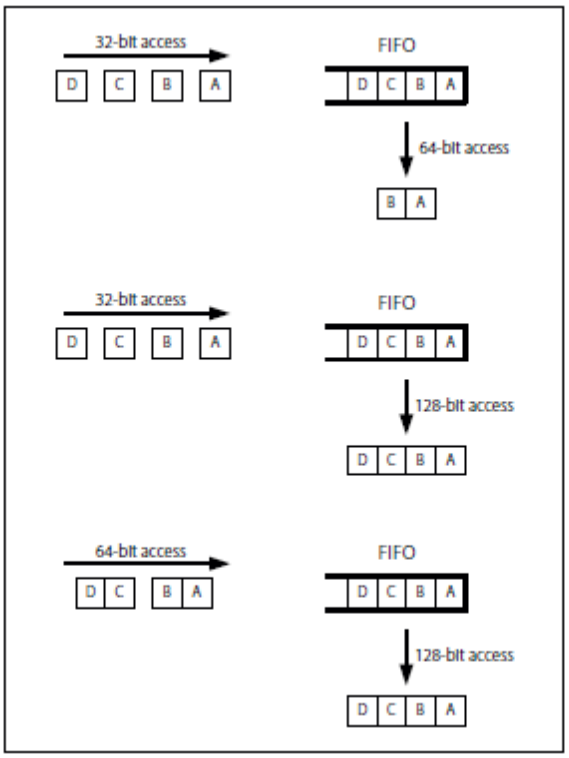
FIFO load information is also taken into account in order to maximize the efficiency of the data transfer. Incoming data transfers are only started if enough free space is available in the input FIFO. Outgoing data transfers are only started if the output FIFO contains enough data for a complete burst.

The data master performs no manipulation on the transferred data; it is ensured that the data which is delivered to the internal AES engine has the same content and order as it was provided by the user in the source memory location.

4.3. Operation of the FIFO Unit

All data transfers to and from the SHE module are done over two FIFO memories. Additionally they are used for synchronisation between several parties with different data-path widths. For example, the data can be written to the input FIFO via the command interface using 32-bit accesses or via the data interface using 64-bit accesses. The same is also valid for the output FIFO, which can be read either by a 32-bit or 64-bit access. Hence, to ensure the correct synchronisation and data order, the least significant part of the data has to be transferred first. More precisely, if the transfer of a data item has to be split into several transfers, the order shown in Figure 4-9 must be kept.

Figure 4-9 FIFO Data Ordering



With reference to Figure 4-9:

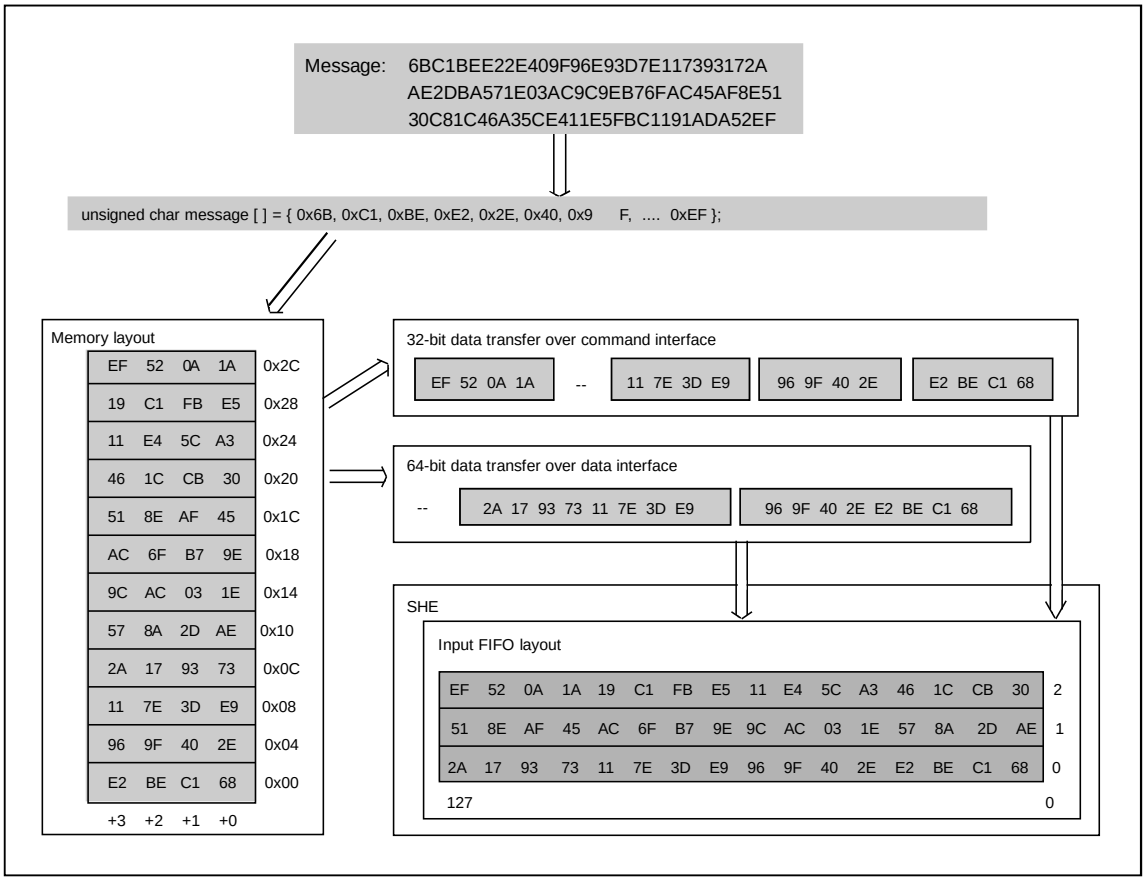
1. If a 64-bit double word has to be written/read using two 32-bit accesses, the least significant word, i.e. bits[31:0], has to be transferred first followed by the most significant word, i.e. bits [63:32].
2. If a 128-bit quad word has to be written/read using four 32-bit accesses, the least significant word, i.e. bits[31:0], has to be transferred first followed by bits[63:32] and bits[95:64]. The most significant word, i.e. bits[127:96], is transferred last.
3. If a 128-bit quad word has to be written/read using two 64-bit accesses, the least significant double word, i.e. bits[63:0], has to be transferred first followed by the most significant double word, i.e. bits[127:64].

Data ordering for the Command Parameters

Command parameters given in Table 3-2 must be transferred according to one of the following ways depending on the parameter width:

1. 64-bit wide parameters must be represented in the native little endian byte order of TRAVEO™ T1G host system and, as described in the Section 3.1, the least significant part of the parameter has to be transferred first. For example if the KEY_ID parameter is 4 (0x0000000000000004) and it will be transferred to SHE using two 32-bit accesses, the data for the first access will be 0x00000004 and for the second will be 0x00000000.
2. Parameters wider than 64-bits must be represented in the big endian ordering of bytes and the most significant byte (MSB) must be transferred first. Please refer to the Figure 4-10 for an example showing data transfer of a 384-bit large message. Since cryptographic messages are usually stored using byte arrays, the MSB of the message will be stored in the memory at the offset zero resulting in the memory layout given in the example. For the processing in SHE the message can be transferred either through the command interface (using SHE_IFIFOWRDATA registers) or through the data interface (using input data master). The resulting layout of the input FIFO of SHE represents the message with byte-wise swapped 128-bits data blocks. This swapping is then compensated by the SHE hardware during the internal processing. Note that the results of the SHE processing which are wider than 64 bits are also represented in the big endian ordering of bytes and are available in the output FIFO of SHE in the byte swapped way.

Figure 4-10 FIFO Data Ordering for Parameters Wider than 64 Bits



Input FIFO

The input FIFO of SHE is used for the transfer of parameters and data required for SHE command execution. It can be written automatically through the input data master or manually by the user through the command interface - refer to the Input FIFO Write Data Register (SHE_IFIFOWRDATA0 to 31) description. The input FIFO has a capacity of 1,536 bits, which is sufficient to store one and a half bursts from the input data master. The size of free space in the input FIFO can be determined by reading the SHE_FIFOLOAD:IFIFOFREE status register.

The input FIFO provides a programmable write threshold setting - refer to the Input FIFO Configuration Register (SHE_IFIFOCFG) description. If the amount of stored data words is less than the stored value, a status flag (SHE_FIFOSTATUS:IFIFOWRTH) and an interrupt flag (SHE_IRQ:IFIFOWRTH) are set.

SHE has to be able to recognize that the amount of data required for the execution of a SHE API function has been transferred through the input FIFO. For this purpose two specific registers are introduced: SHE_COMPARE and SHE_DATACNT. The former contains the required amount of data and the latter contains the amount of data actually transferred. According to the SHE specification, the largest possible data stream size for a function is 2^{64} bits. And since both registers represent data amounts in terms of 32-bit words, their size is specified to 59 bits.

SHE_DATACNT is an up-counter and is automatically reset to zero every time the opcode for a new SHE API function is written to the SHE_CMD register. Every read access to the input FIFO increases the value of the counter. When the counter reaches the value of the SHE_COMPARE register, the

status bit SHE_FIFOSTATUS:COMPAREMATCH and an interrupt request bit SHE_IRQ:COMPAREMATCH are immediately set. This transition of the COMPAREMATCH bit is referenced to as a compare match event.

SHE_COMPARE is reset by the hardware to the maximal value (2^{59} - 1) every time the opcode for a new SHE command is written to the SHE_CMD register. This approach avoids an immediate compare match event in case of low reset values (including zero value). In most cases the SHE_COMPARE register is configured through the SHE control logic according to selected command and/or parameter values and cannot be changed during command execution. The only exception to this rule is two SHE commands CMD_ENC_CBC and CMD_DEC_CBC. In this case the amount of processing data is not always known (i.e. a priori) and command execution is started using the default value of SHE_COMPARE. Once the required amount of data is known, the software updates the SHE_COMPARE register.

According to the functional specification of SHE, some commands require notification in case the amount of delivered data exceeds that required. This notification will be stored in the error code register and can be read by the software. But since the error code has to be read by software before a new command can be written, there is a time window from reading the error code to writing the new command where new data could enter the input FIFO without being recognized by the software (error code already read). In order to avoid this situation, the write port of the input FIFO has to be locked at the time when all input data have been transferred until the next command is written. This means the lock is enabled immediately after the compare match event and disabled upon writing new opcode to the SHE_CMD register. If any write access to the locked input FIFO is performed by the input data master or through the command interface, an error interrupt flag (SHE_IRQ:IFIFOLOCKERR) is set.

Output FIFO

The output FIFO of SHE is used to transfer the results of SHE commands. It can be read automatically by the output data master or manually by the user through the command interface - refer to the Output FIFO Read Data Register (SHE_OFIFORDDATA0 to 31) description. The capacity of the output FIFO is equal to 1,536 bits, which is sufficient to store one and a half bursts for the output data master. The load status of the output FIFO can be determined by reading the SHE_FIFOSTATUS:OFIFOLOAD status bit.

The output FIFO provides the programmable read threshold setting (see SHE_OFIFOCFG:RDTHRESHOLD bit description). If the amount of stored data words is greater than the stored value, a status flag (SHE_FIFOLOAD:OFIFORDTH) and an interrupt flag (SHE_IRQ:OFIFORDTH) are set.

To avoid an inconsistent state in case of excess data contained in FIFOs, they are both cleared automatically at the beginning of the command. More precisely, the FIFOs are cleared by a writing access to the command register SHE_CMD while SHE is not busy.

5. Notes on Using SHE

This section lists some issues which has to be taken care of when using the SHE module. Programmers should read these guidelines before programming the SHE module.

5.1. General Notes on Using SHE

SHE is connected to the same reset and the same clock signals as the CPU and the WFLASH. The only difference is that the clock of SHE cannot be switched off in the system controller and is always enabled. However SHE allows local clock gating (refer to 3.1.1 Power saving functionality).

Note:

- *Since secure key storage is implemented in WFLASH, the WFLASH clock must be enabled when SHE functionality is required.*

5.1.1. Notes on Using SHE Commands

CMD_SECURE_BOOT

Secure boot process is automatically started after every reset. Secure boot settings i.e. secure boot mode, start address and size if the protected memory are stored in the BDR area of the TCFLASH. The Boot ROM code reads these settings and afterwards evaluates them and configures the input data master of SHE and starts execution of the CMD_SECURE_BOOT.

Note that the user application is started only after the execution of CMD_SECURE_BOOT inside SHE has begun. Hence from the point of view of user application some SHE registers do not have their reset values as defined in this document. Depending on secure boot settings following registers can be effected:

- SHE_CMD
- SHE_STATUS
- SHE_ERC
- SHE_IMSTADDR
- SHE_IMSTCNT
- SHE_IMSTSTART
- SHE_IFIFOCFG
- SHE_COMPARE0 to 1
- SHE_MSTSTATUS
- SHE_FIFOSTATUS
- SHE_FIFOLOAD
- SHE_DATACNT0 to 1

Depending on secure boot mode, the secure boot process may not have finished when entering the application. The application software or SHE driver shall consider that SHE might still be busy and should either wait for its completion or cancel the process.

Autonomous Bootstrap Configuration of the Secure Boot Process

If the secure booting has been personalized during the same boot process, the following is valid for the BOOT_MAC memory slot:

- None of the flags are set
- The counter is equal to 1

Memory Used for Secure Boot

Though all of the memories of MCU can be used for the secure boot process, it is recommended to use only the instruction flash (TCFLASH), since write access to it is monitored by SHE. This means a write access to instruction flash during the secure boot leads to the secure boot failure.

Secure Boot Settings

Secure boot start address should be aligned to 64-bit boundary (lower 3 bits are "0"). Otherwise it will be automatically aligned by the Boot ROM code during the evaluation of the secure boot settings. Note that only the start address which is used for setting up the input data master of the SHE is modified, not the address in BDR which is the entry address to the application. The amount of data which is

transferred to SHE for the CMD_SECURE_BOOT is always a multiple of 128-bits. If the secure bit size parameter stored in BDR is not a multiple of 128-bits, following applies ‘data size of input data master configuration’ = ceil (‘size from BDR’ / 16) * 2. The amount of valid data is still given by the secure boot size parameter in the BDR.

If secure boot start address and/or secure boot size are modified by the Boot ROM, the secure boot process may try to access non-initialized memory regions before or after the authenticated area. If such access causes bus error, secure boot process will fail.

If secure boot start address and/or secure boot size are modified by Boot ROM during successful personalization (self-learning of BOOT_MAC) and the secure boot settings in the BDR are not changed by the user anymore, same adjustment applies to secure boot process during next reset cycles and successful secure boot is possible.

Furthermore this automated adjustment of the secure boot settings must be kept in mind if the BOOT_MAC is calculated offline and loaded into SHE using CMD_LOAD_KEY. Otherwise the BOOT_MAC calculated by SHE may not be equal to the reference BOOT_MAC and the secure boot will fail.

CMD_LOAD_KEY

The initial (empty) value of the non-volatile slots is: 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF. Special care has to be taken if the memory update command fails, i.e. the error code register

SHE_ERC:CMDMAIN contains an error code other than ERC_NO_ERROR. In such a case the user should check whether the old or the new version of the key is valid before repeating the update. If an error occurs before or during the physical write operation to the memory, the old value is still valid. If an error occurs after the physical write operation to the memory, the new value is valid.

SHE stores keys in a redundant way which is intended to cover aging and transient errors. Occurrence of such errors in the memory slot during the write by CMD_LOAD_KEY cause the command to fail despite redundancy and the NVM state is not changed. However in some rare cases such errors are accepted by CMD_LOAD_KEY and the memory slot is updated. In that case the command CMD_LOAD_KEY will succeed (SHE_ERC:CMDMAIN contains ERC_NO_ERROR), but the user will be notified by the warning code (other than ERC_INVALID) in the extended error code register (SHE_ERC:CMDEXTD) and can react by repeating the memory update procedure.

5.1.2. SHE Used Error Codes/Used Cancel Codes

Table 5-1 SHE Used Error Codes

Cancel Error	Command Error	Main Error	Description
0x0010	0x0001	ERC_NO_ERROR	Command was normally completed.
See Table 5-2	0x000E	ERC_CANCELED	The current command was aborted by SHE_CMDCANCEL:CANCELREQ.
0x0010	0x010D	ERC_GENERAL_ERROR	AES engine could not be stopped after a preceding error.
0x0010	0x020D	ERC_GENERAL_ERROR	Internal register access invoked a bus error.
0x0010	0x4D06	ERC_NO_SECURE_BOOT	TCFLASH was written during Secure Boot.
0x0010	0x4506	ERC_NO_SECURE_BOOT	Secure boot is disabled by the empty secure boot key.
0x0010	0x2f03	ERC_KEY_NOT_AVAILABLE	While a boot protected key was used, the BOOTOK flag was cleared, e.g. by clock switching.
0x0010	0x3003	ERC_KEY_NOT_AVAILABLE	While a debugger protected key was used, the EXTDEBUGGER flag was set due to the connecting debugger.
0x0010	0x390D	ERC_GENERAL_ERROR	The requested command is not available in the current state or mode.
0x0010	0x370D	ERC_GENERAL_ERROR	The AXI channel master stopped host-to- SHE transfer due to a bus error.
0x0010	0x360D	ERC_GENERAL_ERROR	The AXI channel master stopped SHE-to- host transfer due to a bus error.
0x0010	0x1904	ERC_KEY_INVALID	Invalid SHE slot number command argument.
0x0010	0x1004	ERC_KEY_INVALID	The RAM key to be exported was loaded as a plain key.
0x0010	0x1E09	ERC_RNG_SEED	The command requires initialized RNG but CMD_INIT_RNG was not executed before or was discarded by CMD_DEBUG.
0x0010	0x3A0D	ERC_GENERAL_ERROR	Illegal device test mode request in SHE operation mode.
0x0010	0x1C04	ERC_KEY_INVALID	Key update protocol violated by an invalid key ID or an authentication ID.
0x0010	0x2708	ERC_KEY_UPDATE_ERROR	Key update violated by unmatched UID.
0x0010	0x1B08	ERC_KEY_UPDATE_ERROR	Key update violated by unmatched M3.
0x0010	0x1308	ERC_KEY_UPDATE_ERROR	Key update protocol violated by a wildcard UID despite the wildcard being disabled for the slot.
0x0010	0x4E0D	ERC_GENERAL_ERROR	CMD_INIT_RNG failed due to missing TRNG calibration data.
0x0010	0x030D	ERC_GENERAL_ERROR	Command disabled or command code unknown.
0x0010	0x190A	ERC_NO_DEBUGGING	CMD_DEBUG failed due to invalid authentication.
0x0010	0x1107	ERC_KEY_WRITE_PROTECTED	CMD_DEBUG failed due to at least one write protected or invalid SHE slot.
0x0010	0x170D	ERC_GENERAL_ERROR	At any command end there are still unprocessed data in the input FIFO if an extra argument was falsely provided to SHE.

Cancel Error	Command Error	Main Error	Description
0x0010	0x3C0D	ERC_GENERAL_ERROR	This is a special asynchronous error code which may occur independently of the normal command end if the ECC double bit error in the SHE RAM does not allow further SHE operations. If it occurs, the lower byte of SHE_STATUS is not updated but SHE_STATUS:FATAL is set. SHE is halted.
0x0010	0x400D	ERC_GENERAL_ERROR	MCU device mode was changed on-the-fly during the current SHE operation mode. SHE is halted.
0x0010	0x420D	ERC_GENERAL_ERROR	Illegal combination of device mode signals and SHE mode signals. SHE is halted.
0x0010	0x3D0D	ERC_GENERAL_ERROR	Invalid NVM state found during SHE start- up. SHE is halted.
0x0010	0x3B0D	ERC_GENERAL_ERROR	Despite SHE being disabled due to the NVM state and device mode, the device mode changes again. SHE remains stopped.
0x0010	0x410D	ERC_GENERAL_ERROR	The device mode changes again despite SHE being in test mode. SHE remains stopped.
0x0010	0x430D	ERC_GENERAL_ERROR	NVM is blank. Fabrication is not complete. SHE is halted.
0x0010	0x3E0D	ERC_GENERAL_ERROR	The device mode changes despite SHE being in operation. SHE is halted.
0x0010	0x340C	ERC_MEMORY_FAILURE	Internal NVM address is out of range. NVM operation skipped.
0x0010	0x310C	ERC_MEMORY_FAILURE	Reading NVM showed there was insufficient redundancy to rely in data
0x0010	0x4A0D	ERC_GENERAL_ERROR	Slot ID is out of range. Internal error which should never occur.
0x0010	0x350D	ERC_GENERAL_ERROR	An active NVM slot revision was not completely loaded. This error should never occur.
0x0010	0x330C	ERC_MEMORY_FAILURE	Failed read verification of a word just written to the NVM.
0x0010	0x0B03	ERC_KEY_NOT_AVAILABLE	A boot protected key is used even though secure boot had failed (BOOTOK == 0).
0x0010	0x1A08	ERC_KEY_UPDATE_ERROR	Key update protocol violated by invalid key counter.
0x0010	0x0F03	ERC_KEY_NOT_AVAILABLE	A debugger protected key is used despite external debugger being connected.
0x0010	0x1204	ERC_KEY_INVALID	The key usage for encryption/decryption or MAC calculation/verification does not match the slot flag.
0x0010	0x1806	ERC_NO_SECURE_BOOT	Invalid sequence of secure boot commands CMD_SECURE_BOOT, CMD_BOOT_OK and CMD_BOOT_FAILURE.
0x0010	0x1F0C	ERC_MEMORY_FAILURE	A bus error occurred when accessing the NVM private bus.
0x0010	0x210D	ERC_GENERAL_ERROR	A double bit error occurred when reading the NVM.
0x0010	0x230C	ERC_MEMORY_FAILURE	The HANGINT error occurred when erasing a sector of the NVM.
0x0010	0x540C	ERC_MEMORY_FAILURE	SHE could not lock Flash by setting WE bit.
0x0010	0x5501	ERC_MEMORY_FAILURE	SHE may not have properly released Flash IF by clearing WE bit.

Cancel Error	Command Error	Main Error	Description
0x0010	0x080C	ERC_MEMORY_FAILURE	The NVM Master was either not idle before reading the NVM or does not get ready after sequencer operation (sector erase, blank check)
0x0010	0x200C	ERC_MEMORY_FAILURE	Flash interface reported that a word could not be written correctly.
0x0010	0x090C	ERC_MEMORY_FAILURE	The NVM Master was not idle before writing to the NVM.
0x0010	0x500C	ERC_MEMORY_FAILURE	The sector swap during the NVM update failed. The old values are still active but redundancy is reduced.
0x0010	0x0C0C	ERC_MEMORY_FAILURE	A required slot is not available due to an NVM error during start-up.
0x0010	0x0D0C	ERC_MEMORY_FAILURE	A RAM buffered NVM slot is corrupted by a double bit error.
0x0010	0x0E05	ERC_KEY_EMPTY	A slot needed for authentication, encryption, decryption MAC generation or verification is empty.
0x0010	0x1107	ERC_KEY_WRITE_PROTECTED	A slot to be updated is write protected.
0x0010	0x4F0C	ERC_MEMORY_FAILURE	The slot update failed and the slot still shows the old value but with reduced redundancy.
0x0010	0x510C	ERC_MEMORY_FAILURE	The slot update failed due to multiple defects and no longer contains any valid key.
0x0010	0x4F01	ERC_NO_ERROR	The slot update was incomplete. The slot contains the new value but the redundancy of the control marker has been reduced.
0x0010	0x140D	ERC_GENERAL_ERROR	TRNG returned an autocorrelation error.
0x0010	0x250D	ERC_GENERAL_ERROR	TRNG returned a CRNGT error.
0x0010	0x260D	ERC_GENERAL_ERROR	The TRNG returned a “Lost Sample” error.
0x0010	0x160D	ERC_GENERAL_ERROR	TRNG did not return RND on time.
0x0010	0x240D	ERC_GENERAL_ERROR	TRNG returned a “Von Neumann” error.

Table 5-2 SHE Used Cancel Codes

Cancel Error	Command Error	Main Error	Description
0x0010	See Table 4-1	See Table 4-1	Command was not cancelled.
0x0000	n.a.	n.a.	Cancellation requested but not yet processed.
0x0001	0x000E	ERC_CANCELED	Cancelled without any error.
0x010D	0x000E	ERC_CANCELED	AES engine could not be stopped during cancellation.
0x020D	0x000E	ERC_CANCELED	Internal register access invoked a bus error during cancellation.

6. Enhanced Secure Hardware Extension

This section describes Enhanced Secure Hardware Extension (ESHE) module which is an extension of the SHE. ESHE is fully compliant with Functional Specification of SHE[1] and provides some additional functionality which is described below.

6.1. Additional Generate Purpose Keys

ESHE has ten additional non-volatile memory slots KEY_11 through KEY_20 which means that ESHE provides in total twenty general-purpose keys KEY_<n> (where n is a number 1...20). The additional keys comply to KEY_<n> requirements from Functional Specification of SHE and can be used for bulk data processing through functions described in section [5.3].

Additional keys KEY_11 to KEY_20 are addressable by a four-bit address as shown in the Table 6-1.

Table 6-1 KEY_<n> (n = 11..20) Addressing

Key Name	Address (hexadecimal)
KEY_11	0x4
KEY_12	0x5
KEY_13	0x6
KEY_14	0x7
KEY_15	0x8
KEY_16	0x9
KEY_17	0xA
KEY_18	0xB
KEY_19	0xC
KEY_20	0xD

Note:

- *since keys KEY_11 to KEY_20 have the same addresses as the already existing keys KEY_1 to KEY_10 the differentiation which key is used it based on the command being executed. KEY_1 to KEY_10 can only be used together with commands specified by Functional Specification of SHE whereas KEY_11 to KEY_20 can only be used together with the new ESHE commands described in section [5.3].*

6.2. Additional Verify Only Flag

An additional key usage flag “VERIFY_ONLY” is implemented for KEY_<n> memory slots (where n is a number 1...20) and can be used for key usage restriction. If this flag is set to "1", the memory slot can only be used for the CMD_VERIFY_MAC command. Trying to use such key for en-/ decryption operations as well as for MAC generation will result in an error code ERC_KEY_INVALID. Otherwise, if the verify only flag is set to "0", the memory slot can be used either for en-/decryption operation or for MAC generation/verification operation depending on the value of the key usage flag as specified in the Functional Specification of SHE.

Following table summarizes the key usage according to values of VERIFY_ONLY and KEY_USAGE flags.

Table 6-2 The key usage according to values of VERIFY_ONLY and KEY_USAGE flags

VERIFY_ONLY Flag	KEY_USAGE	Key Usage
0	0	Encryption or decryption
0	1	MAC generation or verification
1	don't care	MAC verification

The memory update protocol for a KEY_<n> slot equipped with the VERIFY_ONLY flag is modified with respect to computing of the M2 message.

- The FID flag vector is extended by the VERIFY_ONLY flag
  F_{ID}' = WRITE_PROTECTION|BOOT_PROTECTION|DEBUGGER_PROTECTION|KEY_USAGE|WILDCARD|VERIFY_ONLY

- The size of the pattern to fill the first block with "0" bits padding data is reduced by one bit
  $M_2 = ENC_{CBC,K1,IV=0}(C_{ID}'|F_{ID}'|"0...0"_{94}|K_{ID}')$

6.3. Additional User-accessible Functions

ESHE provides seven additional user-accessible commands which can be used in combination with the general-purpose keys KEY_11 to KEY_20. These commands are derived from the commands applicable to KEY_<n> memory slots as defined by SHE Functional Specification and are marked with "_ESHE" suffix. The functionality of the ESHE commands is the same as of the appropriate SHE commands with the difference that ESHE commands are applicable only to KEY_11 to KEY_20. All new ESHE commands are listed in the Table 6-3 together with the opcode value which is required to start the execution through the SHE_CMD register.

Table 6-3 ESHE Commands

	SHE_CMD opcode
CMD_ENC_ECB_ESHE	– 0x41
CMD_ENC_CBC_ESHE	– 0x42
CMD_DEC_ECB_ESHE	– 0x43
CMD_DEC_CBC_ESHE	– 0x44
CMD_GENERATE_MAC_ESHE	– 0x45
CMD_VERIFY_MAC_ESHE	– 0x46
CMD_LOAD_KEY_ESHE	– 0x47

The order in which parameters and data for processing are expected as well as the order in which results are generated is shown in Table 6-4.

Table 6-4 Parameter Order for ESHE Commands

Command	Parameter			Return Value		
	Seq. No	Name	Size [Bits]	Seq. No	Name	Size [Bits]
CMD_ENC_ECB_ESHE	1	KEY ID[1]	64			
	2	PLAINTEXT	128			
				1	CIPHERTEXT	128
CMD_ENC_CBC_ESHE	1	KEY ID1	64			
	2	IV	128			
	3	PLAINTEXT	n x 128			
				1	CIPHERTEXT	n x 128
CMD_DEC_ECB_ESHE	1	KEY ID1	64			
	2	CIPHERTEXT	128			
				1	PLAINTEXT	128
CMD_DEC_CBC_ESHE	1	KEY ID1	64			
	2	IV	128			
	3	CIPHERTEXT	n x 128			
				1	PLAINTEXT	n x 128
CMD_GENERATE_MAC_ESHE	1	KEY ID1	64			
	2	MESSAGE LENGTH	64			
	3	MESSAGE	n2 x 128			
				1	MAC	128

Command	Parameter			Return Value		
	Seq. No	Name	Size [Bits]	Seq. No	Name	Size [Bits]
CMD_VERIFY_MAC_ESHE	1	KEY ID1	64			
	2	MESSAGE LENGTH	64			
	3	MAC LENGTH	64			
	4	MAC	128			
	5	MESSAGE	n2 x 128			
				1	VERIFICATION STATUS	64
CMD_LOAD_KEY_ESHE	1	M1	128			
	2	M2	256			
	3	M3	128			
				1	M4	256
				2	M5	128

[1] : valid values: 0x04, …, 0xD, which corresponds to KEY_11, …, KEY_20

[2] : n = ceil[3] (MESSAGE_LENGTH / 128)

[3] : ceil(x) = min{$n \in Z_1 \mid n \geq x$}, $Z_1 = \{1,2,3,...\}$

6.3.1. CMD_LOAD_KEY_ESHE

Secure key update for additional general-purpose keys KEY_11 to KEY_20 is done through the command CMD_LOAD_KEY_ESHE. The memory update protocol is derived from the one described in chapter 9 of Functional Specification of SHE[1]. The differences are described below.

Key update message M1 is generated according to chapter 9.1 of Functional Specification of SHE and is a concatenation of the UID, the ID of the memory slot to be updated and ID of the key used for the authentication. However the ID and AuthID are evaluated according to Table 6-5.

Table 6-5 Parameter Order for ESHE Commands

Key to update	Authentication	ID values to be used for CMD_LOAD_KEY_ESHE			
		KEY_{ID}	ID	KEY_{AuthID}	AuthID
KEY_<n>	KEY_<n>	KEY_<n>	0x4..0xD	same as KEYID	same as ID
KEY_<n>	MASTER_ECU_KE	KEY_<n>	0x4..0xD	MASTER_ECU_K	0x1
RAM_KEY	KEY_<n>	RAM_KEY	0xE	KEY_<n>	0x4..0xD

Note: n = 11..20

Key update and verification messages M2 - M5 are generated according to chapters 9.1 and 9.2 of Functional Specification of SHE. However the constants which are used for generation of intermediate keys K1 - k4 are modified by adding 0x00800000 00000000 00000000 00000000 which is done to prevent cross loading of keys between the memory slots reserved for KEY_1 through KEY_10 and the memory slots reserved for KEY_11 through KEY_20.

The complete message generation scheme for command CMD_LOAD_KEY_ESHE is shown below.

KEY_UPDATE_ENC_C* = KEY_UPDATE_ENC_C + 0x00800000 00000000 00000000 00000000

KEY_UPDATE_MAC_C* = KEY_UPDATE_MAC_C + 0x00800000 00000000 00000000 00000000

F_{ID}' =

WRITE_PROTECTION|BOOT_PROTECTION|DEBUGGER_PROTECTION|KEY_USAGE|WILDCARD|VERIFY_ONLY

$K_1 = KDF(K_{AuthID}, KEY_UPDATE_ENC_C^*)$

$K_2 = KDF(K_{AuthID}, KEY_UPDATE_MAC_C^*)$

$M_1 = UID|ID|AuthID$

$M_2 = ENC_{CBC,K1,IV=0}(C_{ID}'|F_{ID}'|"0...0"_{94}|K_{ID}')$

$M_3 = CMAC_{K2}(M_1|M_2)$

$K_3 = KDF(K_{ID}, KEY_UPDATE_ENC_C^*)$

$K_4 = KDF(K_{ID}, KEY_UPDATE_MAC_C^*)$

$M_4 = UID|ID|AuthID|ENC_{ECB,K3}(C_{ID})$

$M_5 = CMAC_{K4}(M_4)$

7. References

1. SHE Functional Specification v1.1 (rev 439) can be requested at: http://portal.automotive-his.de/index.php?option=com_content&task=view&id=31&Itemid=41&lang=english
2. Specification for the Advanced Encryption Standard (AES): http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf
3. SP 800-38 A -- Recommendation for Block Cipher Modes of Operation: Methods and Techniques: http://csrc.nist.gov/publications/nistpubs/800-38a/sp800-38a.pdf
4. SP 800-38 B -- Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication: http://csrc.nist.gov/publications/nistpubs/800-38B/SP_800-38B.pdf
5. Menezes, van Oorschot, Vanstone: Handbook of Applied Cryptography(HAC), August 2001, CRC Press, ISBN 0-8493-8523-7, http://www.cacr.math.uwaterloo.ca/hac/

CHAPTER 27: DMA Controller

This chapter explains DMA Controller.

1. **Overview**
2. **Configuration**
3. **Operational Description**
4. **Registers**
5. **Additional Information**

DMA-TXXPT03P01R01L08-E1-XX

1. Overview

The DMA Controller (DMAC) implements Direct Memory Access (DMA) with little CPU intervention. DMAC is performing complex data transfers through 'N' DMA channels. This section describes the features and the block diagram of DMA Controller.

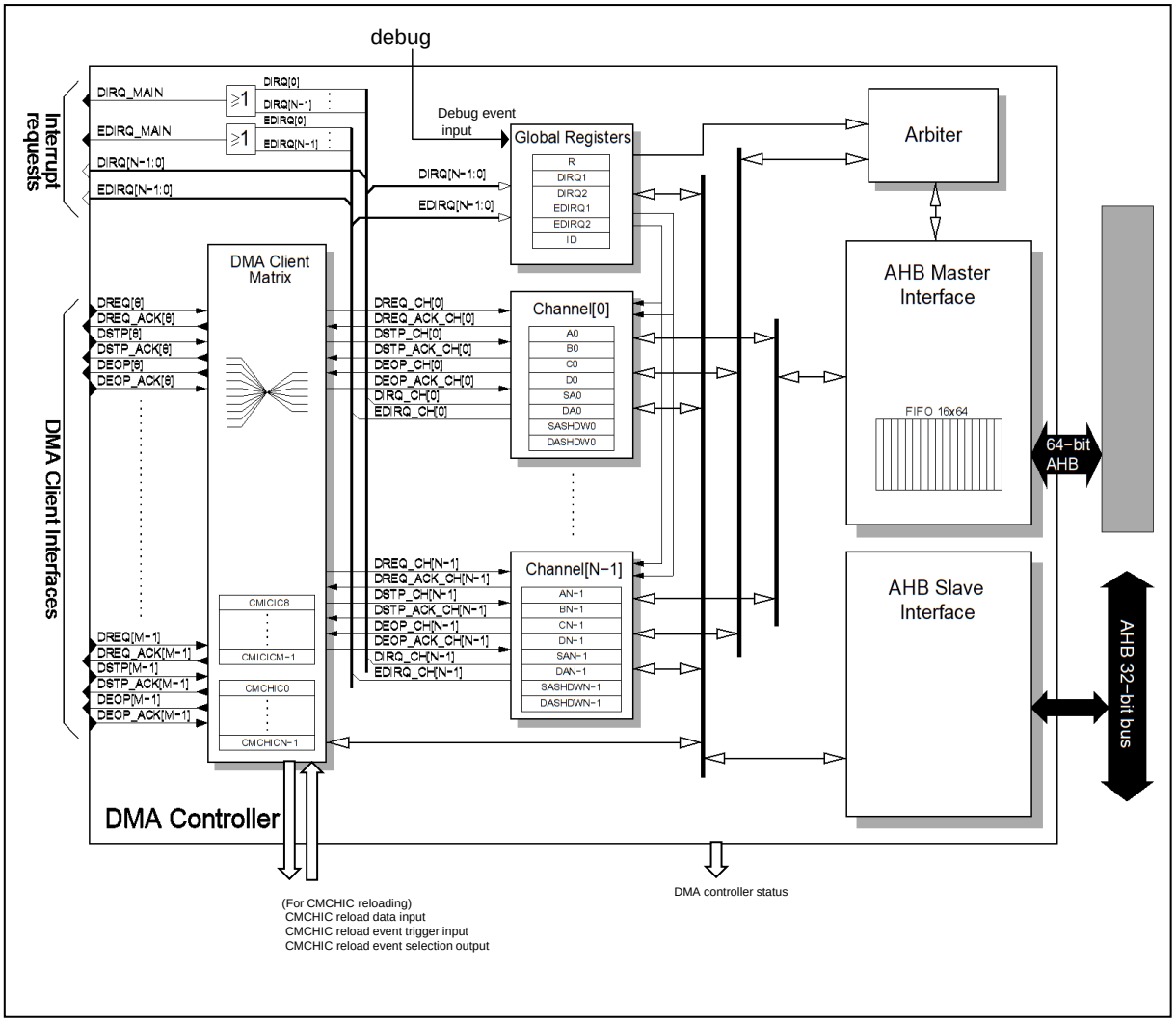
The following items are the features related to data transfer by the DMA Controller:

- Data can be transferred independently over multiple channels
- A high number of MCU internal and MCU DMA clients can be assigned to the available DMA channels
- Flexible priority between DMA channels (fixed, dynamic, or round-robin)
- DMA transfer request sources
  - Hardware request (internal clients)
  - Software request (register write)
- Transfer modes
  - Block transfer and burst transfer
  - Addressing: full 32-bit (incrementing, decrementing, or fixed)
  - Data types: 8-, 16-, 32-, or 64-bit wide data

2. Configuration

This section shows the block diagram.

Figure 2-1 Block Diagram



DREQ : DMA Request

DREQ_ACK : DMA Request Acknowledge

DSTP : DMA Stop Request

DSTP_ACK : DMA Stop Request Acknowledge

DEOP : DMA End of Processing

DEOP_ACK : DMA End of Processing Acknowledge

DIRQ : Completion Interrupt

EDIRQ : Error Interrupt

'i' represents instance number of DMAC. i = 0

'n' represents channel number. n = 0, 1, 2, …,15

'N' represents number of DMAC channels per DMAC instance. N = 16 in this product.

'm' represents client number. m = 8, 9, …, (product specification)

'M' represents number of client I/F. M = (product specification) in this product.

Note:

- *Please refer "CHAPTER:PLATFORM OVERVIEW" for the maximum numbers.*

3. Operational Description

This section describes the operation of the DMA controller.

Features of DMA Controller

- DMA Client Matrix, routing 'M' DMA clients to 'N' DMA channels
- DMA trigger
  - Hardware request (internal clients)
  - Software request
- Two transfer modes
  - Block transfer
  - Burst transfer
- 8-, 16-, 32-, or 64-bit wide data transfers
- Master transactions can be done in user or privileged mode
- Writing the configuration registers can only be done in privileged mode and 8-, 16-, or 32-bit wide. Illegal accesses result in an error response
- Reading the configuration registers can be done in user or privileged mode
- Incrementing, decrementing, or fixed addressing independent for source and destination
- Independent source and destination access protection
- Central interrupt flag register for completion and error interrupts
- Stop status per channel for analysis by ISRs or debugging
- Shadow registers for source and destination address
- Configurable debug event behavior (continue, halt, or stop)
- Three channel arbitration schemes
  - Fixed priority
  - Dynamic priority
  - Round-robin
- Client matrix reloading (CMCHIC reloading)

Global Functions of the DMA Controller

DMAC Enabling or Halting

After reset the DMA Controller is disabled. DMAC is enabled by setting DMA Enable (DMAi_R:DE) to "1". If DMA Enable is set to "1" the individual DMA Channel Enable (DMAi_An:EB) settings become effective. Setting DMA Enable to "0" disables the complete DMAC. Setting DMA Enable to "0" while there is a not completed transfer running on a DMA channel, the running transfer is stopped when its current block of data was transferred, and an error interrupt is issued. In case it is the last block of data of this DMA transfer a completion interrupt is issued. DMA channels which have a DMA transfer pending, but are not currently served are disabled and an error interrupt is issued. DMA channels which are enabled but have not started a DMA transfer (channel was enabled and there was not a transfer request yet) are simply disabled without issuing an error interrupt.

The DMA Controller can be halted completely (all DMA channels) by writing a single bit, DMA Halt (DMAi_R:DH). When this bit is set to "1", all DMA channels are requested to halt and not perform DMA transfers until this bit is cleared. After DMA Halt is cleared the halted DMA transfers continue at the point they were halted. If DMA Halt is set to "1" while a transfer is running, halting takes place once the current block of data of the running transfer is transferred. Depending on the clock ratio of the DMAC/source clock domain and DMAC/destination clock domain a considerable time can elapse between the time when the halt request was set and when the DMAC is actually halted.

Global disable and halt request condition of the DMAC is indicated by DMA Stop/Halt Request Flag (DMAi_R:DSHR). DMA Stop/Halt Request Flag is "0" if none of the below disable or halt request conditions is true:

- DMAi_R:DE bit is set to "0"
- DMAi_R:DH bit is set to "1"
- DMAi_R:DBE = 1, DMAi_R:DB[1:0] = 0b10 (stop on debug event), and a 'debug event' is pending
- DMAi_R:DBE = 1, DMAi_R:DB[1:0] = 0b01 (halt on debug event), and a 'debug event' is pending

If any of the above conditions is true, DMA Stop/Halt Request Flag is "1" indicating that DMA transfers of all channels are requested to halt or stop.

The condition that all channels are halted or have come to a stop after a global halt or stop request is indicated by DMA Stop/Halt Status Flag (DMAi_R:DSHS). DMA Stop/Halt Status Flag is "0" if one or more channels are not yet stopped or halted and "1" if all channels are stopped or halted.

DMAC Debug Behavior

The DMA Controller can be configured to react in a predefined way on a 'debug event' (e.g. debugger break point). The feature to react on a 'debug event' can be enabled with bit Debug Enable (DMAi_R:DBE). If enabled the DMACs behavior depend on the setting of Debug Behavior (DMAi_R:DB[1:0]). This feature is disabled after reset.

The behavior can be configured to stop all transfers (DMAi_R:DB[1:0] = 0b10), or to halt all transfers (DMAi_R:DB[1:0] = 0b01), or just to continue operation independent of debug events (DMAi_R:DB[1:0] = 0b00). Initial value is to continue operation independent of debug events.

3.1. DMA Channels

This section describes behavior of the DMA channel.

Modes of Operation

The DMA channels can operate in two modes:

- Block transfer mode
- Burst transfer mode

The mode must be set with bits Mode Select (DMAi_Bn:MS[1:0]). After reset, the channel is set to block transfer mode.

Block Transfer Mode

In block transfer mode the DMA client will request the transfers of a specified number of blocks of data. The number of blocks to be transferred is specified with Transfer Count (DMAi_An:TC[15:0]). Each block of data is transferred in one arbitration phase of the DMA arbiter. For each block to transfer a DMA transfer request, either a hardware request or software request is needed. The DMA client or the software continues to give requests until the DMAC has transferred the specified number of blocks and thus the DMA transfer is completed. The DMA transfer will be successfully completed if all blocks of data were transferred without error or unsuccessfully completed if an error condition occurred.

After each transferred block of data the DMA arbiter does another arbitration and proceed with the next requesting channel with the highest priority. The arbitration depends on the selected arbitration scheme.

DMA Transfer Requests

1. Hardware request

For a channel hardware request Input Select (DMAi_An:IS) need to be set to '01'. The DMA client which is routed through the DMA Client Matrix to the channel will give the trigger by asserting DREQ. Refer 3.2 for the function and configuration of the DMA Client Matrix.

2. Software request

For a software request Input Select need to be set to '00' and Software Trigger (DMAi_An:ST) must be set to "1". Software Trigger shall be set to "1" if the channel is ready to receive a software trigger, which is indicated with Software Trigger Ready (DMAi_Bn:SR) and no error condition is pending (DMAi_Bn:SS[2:0] is '000' or '101'). If Software Trigger is tried to set to "1" while Software Trigger Ready is "0", it is ignored. Software Trigger is automatically cleared by hardware once the trigger is accepted or an error condition occurs. The error conditions can be:

- DMA channel is disabled right after setting Software Trigger
- A debug event occurred and DMAC is configured to stop all transfers on a debug event
- The master interface receives an error response and the CPU sets the Software Trigger before receiving an error interrupt caused by the AHB error

Block of Data

A block of data is determined by the setting of Block Count (DMAi_An:BC[3:0]) and Transfer Width (DMAi_Bn:TW[1:0]). The DMA Controller will make DMAi_An:BC + 1 data transfers from source address range starting at Source Address (DMAi_SAn:SA) to destination address range starting at Destination Address (DMAi_DAn:DA). If DMAi_An:BC is set to "0" a single data transfer from Source Address (DMAi_SAn:SA) to Destination Address (DMAi_DAn:DA) will be done. The settings of Block Count (DMAi_An:BC[3:0]), Beat Limit (DMAi_An:BL[1:0]), Alternate (DMAi_An:AL), and Transfer Width (DMAi_Bn:TW[1:0]) define how the AHB master interface issues the DMAi_An:BC + 1 data transfers. The following table shows all possible combinations between DMAi_An:BC, DMAi_An:BL, and

DMAi_An:AL. Only these three influence the sequence of AHB transfers, whereas DMAi_Bn:TW only affects the data size which will be transported.

Table 3-1 Block Count/Beat Limit/Alternate Combinations

Block Count	Beat Limit	Alternate	Resulting Sequence of AHB Transfers
0	SINGLE	0	1x SINGLE RD + 1x SINGLE WR
1	SINGLE	0	2x SINGLE RD + 2x SINGLE WR
2	SINGLE	0	3x SINGLE RD + 3x SINGLE WR
3	SINGLE	0	4x SINGLE RD + 4x SINGLE WR
4	SINGLE	0	5x SINGLE RD + 5x SINGLE WR
5	SINGLE	0	6x SINGLE RD + 6x SINGLE WR
6	SINGLE	0	7x SINGLE RD + 7x SINGLE WR
7	SINGLE	0	8x SINGLE RD + 8x SINGLE WR
8	SINGLE	0	9x SINGLE RD + 9x SINGLE WR
9	SINGLE	0	10x SINGLE RD + 10x SINGLE WR
10	SINGLE	0	11x SINGLE RD + 11x SINGLE WR
11	SINGLE	0	12x SINGLE RD + 12x SINGLE WR
12	SINGLE	0	13x SINGLE RD + 13x SINGLE WR
13	SINGLE	0	14x SINGLE RD + 14x SINGLE WR
14	SINGLE	0	15x SINGLE RD + 15x SINGLE WR
15	SINGLE	0	16x SINGLE RD + 16x SINGLE WR
0	SINGLE	1	1x (1x SINGLE RD + 1x SINGLE WR)
1	SINGLE	1	2x (1x SINGLE RD + 1x SINGLE WR)
2	SINGLE	1	3x (1x SINGLE RD + 1x SINGLE WR)
3	SINGLE	1	4x (1x SINGLE RD + 1x SINGLE WR)
4	SINGLE	1	5x (1x SINGLE RD + 1x SINGLE WR)
5	SINGLE	1	6x (1x SINGLE RD + 1x SINGLE WR)
6	SINGLE	1	7x (1x SINGLE RD + 1x SINGLE WR)
7	SINGLE	1	8x (1x SINGLE RD + 1x SINGLE WR)
8	SINGLE	1	9x (1x SINGLE RD + 1x SINGLE WR)
9	SINGLE	1	10x (1x SINGLE RD + 1x SINGLE WR)
10	SINGLE	1	11x (1x SINGLE RD + 1x SINGLE WR)
11	SINGLE	1	12x (1x SINGLE RD + 1x SINGLE WR)
12	SINGLE	1	13x (1x SINGLE RD + 1x SINGLE WR)
13	SINGLE	1	14x (1x SINGLE RD + 1x SINGLE WR)
14	SINGLE	1	15x (1x SINGLE RD + 1x SINGLE WR)
15	SINGLE	1	16x (1x SINGLE RD + 1x SINGLE WR)
0	INCR4	0	1x SINGLE RD + 1x SINGLE WR
1	INCR4	0	2x SINGLE RD + 2x SINGLE WR
2	INCR4	0	3x SINGLE RD + 3x SINGLE WR
3	INCR4	0	1x 4_BEAT RD + 1x 4_BEAT WR
4	INCR4	0	1x 4_BEAT RD + 1x SINGLE RD + 1x 4_BEAT WR + 1x SINGLE WR
5	INCR4	0	1x 4_BEAT RD + 2x SINGLE RD + 1x 4_BEAT WR + 2x SINGLE WR
6	INCR4	0	1x 4_BEAT RD + 3x SINGLE RD + 1x 4_BEAT WR + 3x SINGLE WR
7	INCR4	0	2x 4_BEAT RD + 2x 4_BEAT WR
8	INCR4	0	2x 4_BEAT RD + 1x SINGLE RD + 2x 4_BEAT WR + 1x SINGLE WR

Block Count	Beat Limit	Alternate	Resulting Sequence of AHB Transfers
9	INCR4	0	2x 4_BEAT RD + 2x SINGLE RD + 2x 4_BEAT WR + 2x SINGLE WR
10	INCR4	0	2x 4_BEAT RD + 3x SINGLE RD + 2x 4_BEAT WR + 3x SINGLE WR
11	INCR4	0	3x 4_BEAT RD + 3x 4_BEAT WR
12	INCR4	0	3x 4_BEAT RD + 1x SINGLE RD + 3x 4_BEAT WR + 1x SINGLE WR
13	INCR4	0	3x 4_BEAT RD + 2x SINGLE RD + 3x 4_BEAT WR + 2x SINGLE WR
14	INCR4	0	3x 4_BEAT RD + 3x SINGLE RD + 3x 4_BEAT WR + 3x SINGLE WR
15	INCR4	0	4x 4_BEAT RD + 4x 4_BEAT WR
0	INCR4	1	1x SINGLE RD + 1x SINGLE WR
1	INCR4	1	2x (1x SINGLE RD + 1x SINGLE WR)
2	INCR4	1	3x (1x SINGLE RD + 1x SINGLE WR)
3	INCR4	1	1x (1x 4_BEAT RD + 1x 4_BEAT WR)
4	INCR4	1	1x (1x 4_BEAT RD + 1x 4_BEAT WR) + 1x SINGLE RD + 1x SINGLE WR
5	INCR4	1	1x (1x 4_BEAT RD + 1x 4_BEAT WR) + 2x (1x SINGLE RD + 1x SINGLE WR)
6	INCR4	1	1x (1x 4_BEAT RD + 1x 4_BEAT WR) + 3x (1x SINGLE RD + 1x SINGLE WR)
7	INCR4	1	2x (1x 4_BEAT RD + 1x 4_BEAT WR)
8	INCR4	1	2x (1x 4_BEAT RD + 1x 4_BEAT WR) + 1x SINGLE RD + 1x SINGLE WR
9	INCR4	1	2x (1x 4_BEAT RD + 1x 4_BEAT WR) + 2x (1x SINGLE RD + 1x SINGLE WR)
10	INCR4	1	2x (1x 4_BEAT RD + 1x 4_BEAT WR) + 3x (1x SINGLE RD + 1x SINGLE WR)
11	INCR4	1	3x (1x 4_BEAT RD + 1x 4_BEAT WR)
12	INCR4	1	3x (1x 4_BEAT RD + 1x 4_BEAT WR) + 1x SINGLE RD + 1x SINGLE WR
13	INCR4	1	3x (1x 4_BEAT RD + 1x 4_BEAT WR) + 2x (1x SINGLE RD + 1x SINGLE WR)
14	INCR4	1	3x (1x 4_BEAT RD + 1x 4_BEAT WR) + 3x (1x SINGLE RD + 1x SINGLE WR)
15	INCR4	1	4x (1x 4_BEAT RD + 1x 4_BEAT WR)
0	INCR8	0	1x SINGLE RD + 1x SINGLE WR
1	INCR8	0	2x SINGLE RD + 2x SINGLE WR
2	INCR8	0	3x SINGLE RD + 3x SINGLE WR
3	INCR8	0	1x 4_BEAT RD + 1x 4_BEAT WR
4	INCR8	0	1x 4_BEAT RD + 1x SINGLE RD + 1x 4_BEAT WR + 1x SINGLE WR
5	INCR8	0	1x 4_BEAT RD + 2x SINGLE RD + 1x 4_BEAT WR + 2x SINGLE WR
6	INCR8	0	1x 4_BEAT RD + 3x SINGLE RD + 1x 4_BEAT WR + 3x SINGLE WR
7	INCR8	0	1x 8_BEAT RD + 1x 8_BEAT WR
8	INCR8	0	1x 8_BEAT RD + 1x SINGLE RD + 1x 8_BEAT WR + 1x SINGLE WR
9	INCR8	0	1x 8_BEAT RD + 2x SINGLE RD + 1x 8_BEAT WR + 2x SINGLE WR
10	INCR8	0	1x 8_BEAT RD + 3x SINGLE RD + 1x 8_BEAT WR + 3x SINGLE WR
11	INCR8	0	1x 8_BEAT RD + 1x 4_BEAT RD + 1x 8_BEAT WR + 1x 4_BEAT WR
12	INCR8	0	1x 8_BEAT RD + 1x 4_BEAT RD + 1x SINGLE RD + 1x 8_BEAT WR + 1x 4_BEAT WR + 1x SINGLE WR
13	INCR8	0	1x 8_BEAT RD + 1x 4_BEAT RD + 2x SINGLE RD + 1x 8_BEAT WR + 1x 4_BEAT WR + 2x SINGLE WR
14	INCR8	0	1x 8_BEAT RD + 1x 4_BEAT RD + 3x SINGLE RD + 1x 8_BEAT WR + 1x 4_BEAT WR + 3x SINGLE WR
15	INCR8	0	2x 8_BEAT RD + 2x 8_BEAT WR
0	INCR8	1	1x SINGLE RD + 1x SINGLE WR
1	INCR8	1	2x (1x SINGLE RD + 1x SINGLE WR)

Block Count	Beat Limit	Alternate	Resulting Sequence of AHB Transfers
2	INCR8	1	3x (1x SINGLE RD + 1x SINGLE WR)
3	INCR8	1	1x (1x 4_BEAT RD + 1x 4_BEAT WR)
4	INCR8	1	1x (1x 4_BEAT RD + 1x 4_BEAT WR) + 1x SINGLE RD + 1x SINGLE WR
5	INCR8	1	1x (1x 4_BEAT RD + 1x 4_BEAT WR) + 2x (1x SINGLE RD + 1x SINGLE WR)
6	INCR8	1	1x (1x 4_BEAT RD + 1x 4_BEAT WR) + 3x (1x SINGLE RD + 1x SINGLE WR)
7	INCR8	1	1x (1x 8_BEAT RD + 1x 8_BEAT WR)
8	INCR8	1	1x (1x 8_BEAT RD + 1x 8_BEAT WR) + 1x SINGLE RD + 1x SINGLE WR
9	INCR8	1	1x (1x 8_BEAT RD + 1x 8_BEAT WR) + 2x (1x SINGLE RD + 1x SINGLE WR)
10	INCR8	1	1x (1x 8_BEAT RD + 1x 8_BEAT WR) + 3x (1x SINGLE RD + 1x SINGLE WR)
11	INCR8	1	1x (1x 8_BEAT RD + 1x 8_BEAT WR) + 1x (1x 4_BEAT RD + 1x 4_BEAT WR)
12	INCR8	1	1x (1x 8_BEAT RD + 1x 8_BEAT WR) + 1x (1x 4_BEAT RD + 1x 4_BEAT WR) + 1x SINGLE RD + 1x SINGLE WR
13	INCR8	1	1x (1x 8_BEAT RD + 1x 8_BEAT WR) + 1x (1x 4_BEAT RD + 1x 4_BEAT WR) + 2x (1x SINGLE RD + 1x SINGLE WR)
14	INCR8	1	1x (1x 8_BEAT RD + 1x 8_BEAT WR) + 1x (1x 4_BEAT RD + 1x 4_BEAT WR) + 3x (1x SINGLE RD + 1x SINGLE WR)
15	INCR8	1	2x (1x 8_BEAT RD + 1x 8_BEAT WR)
0	INCR16	0	1x SINGLE RD + 1x SINGLE WR
1	INCR16	0	2x SINGLE RD + 2x SINGLE WR
2	INCR16	0	3x SINGLE RD + 3x SINGLE WR
3	INCR16	0	1x 4_BEAT RD + 1x 4_BEAT WR
4	INCR16	0	1x 4_BEAT RD + 1x SINGLE RD + 1x 4_BEAT WR + 1x SINGLE WR
5	INCR16	0	1x 4_BEAT RD + 2x SINGLE RD + 1x 4_BEAT WR + 2x SINGLE WR
6	INCR16	0	1x 4_BEAT RD + 3x SINGLE RD + 1x 4_BEAT WR + 3x SINGLE WR
7	INCR16	0	1x 8_BEAT RD + 1x 8_BEAT WR
8	INCR16	0	1x 8_BEAT RD + 1x SINGLE RD + 1x 8_BEAT WR + 1x SINGLE WR
9	INCR16	0	1x 8_BEAT RD + 2x SINGLE RD + 1x 8_BEAT WR + 2x SINGLE WR
10	INCR16	0	1x 8_BEAT RD + 3x SINGLE RD + 1x 8_BEAT WR + 3x SINGLE WR
11	INCR16	0	1x 8_BEAT RD + 1x 4_BEAT RD + 1x 8_BEAT WR + 1x 4_BEAT WR
12	INCR16	0	1x 8_BEAT RD + 1x 4_BEAT RD + 1x SINGLE RD + 1x 8_BEAT WR + 1x 4_BEAT WR + 1x SINGLE WR
13	INCR16	0	1x 8_BEAT RD + 1x 4_BEAT RD + 2x SINGLE RD + 1x 8_BEAT WR + 1x 4_BEAT WR + 2x SINGLE WR
14	INCR16	0	1x 8_BEAT RD + 1x 4_BEAT RD + 3x SINGLE RD + 1x 8_BEAT WR + 1x 4_BEAT WR + 3x SINGLE WR
15	INCR16	0	1x 16_BEAT RD + 1x 16_BEAT WR
0	INCR16	1	1x SINGLE RD + 1x SINGLE WR
1	INCR16	1	2x (1x SINGLE RD + 1x SINGLE WR)
2	INCR16	1	3x (1x SINGLE RD + 1x SINGLE WR)
3	INCR16	1	1x (1x 4_BEAT RD + 1x 4_BEAT WR)
4	INCR16	1	1x (1x 4_BEAT RD + 1x 4_BEAT WR) + 1x SINGLE RD + 1x SINGLE WR
5	INCR16	1	1x (1x 4_BEAT RD + 1x 4_BEAT WR) + 2x (1x SINGLE RD + 1x SINGLE WR)
6	INCR16	1	1x (1x 4_BEAT RD + 1x 4_BEAT WR) + 3x (1x SINGLE RD + 1x SINGLE WR)
7	INCR16	1	1x (1x 8_BEAT RD + 1x 8_BEAT WR)

Block Count	Beat Limit	Alternate	Resulting Sequence of AHB Transfers
8	INCR16	1	1x (1x 8_BEAT RD + 1x 8_BEAT WR) + 1x SINGLE RD + 1x SINGLE WR
9	INCR16	1	1x (1x 8_BEAT RD + 1x 8_BEAT WR) + 2x (1x SINGLE RD + 1x SINGLE WR)
10	INCR16	1	1x (1x 8_BEAT RD + 1x 8_BEAT WR) + 3x (1x SINGLE RD + 1x SINGLE WR)
11	INCR16	1	1x (1x 8_BEAT RD + 1x 8_BEAT WR) + 1x (1x 4_BEAT RD + 1x 4_BEAT WR)
12	INCR16	1	1x (1x 8_BEAT RD + 1x 8_BEAT WR) + 1x (1x 4_BEAT RD + 1x 4_BEAT WR) + 1x SINGLE RD + 1x SINGLE WR
13	INCR16	1	1x (1x 8_BEAT RD + 1x 8_BEAT WR) + 1x (1x 4_BEAT RD + 1x 4_BEAT WR) + 2x (1x SINGLE RD + 1x SINGLE WR)
14	INCR16	1	1x (1x 8_BEAT RD + 1x 8_BEAT WR) + 1x (1x 4_BEAT RD + 1x 4_BEAT WR) + 3x (1x SINGLE RD + 1x SINGLE WR)
15	INCR16	1	1x 16_BEAT RD + 1x 16_BEAT WR

Note:

- *n_BEAT RD can be a 'n' beat incremental burst (INCRn) or it can be 'n' times a single (SINGLE) data transfer. n_BEAT RD will be 'n' times a SINGLE transfer if one or more of the following conditions is met:*
  - Fixed Source Address (DMAi_Dn:FS) is set to "1"
  - Decrement Source Address (DMAi_Dn:DES) is set to "1"
  - The 1 KB AHB address boundary will be crossed by the read block transfer
- *n_BEAT WR can be a 'n' beat incremental burst (INCRn) or it can be 'n' times a single (SINGLE) data transfer. n_BEAT WR will be 'n' times a SINGLE transfer if one or more of the following conditions is met:*
  - Fixed Destination Address (DMAi_Dn:FD) is set to "1"
  - Decrement Destination Address (DMAi_Dn:DED) is set to "1"
  - The 1 KB AHB address boundary will be crossed by the write block transfer.

After each successful read data transfer Source Address Shadow (DMAi_SASHDWn:SASHDW) will be either incremented, decremented, or remains unaltered. The behavior is determined by the settings of Fixed Source Address, Decrement Source Address, or Fixed Block Source Address. Destination Address Shadow (DMAi_DASHDWn:DASHDW) exhibits the same behavior with respect to the settings of Fixed Destination Address, Decrement Destination Address, or Fixed Block Destination Address and will be updated after each successful write data transfer. The tables below list the possible combinations and the resulting action.

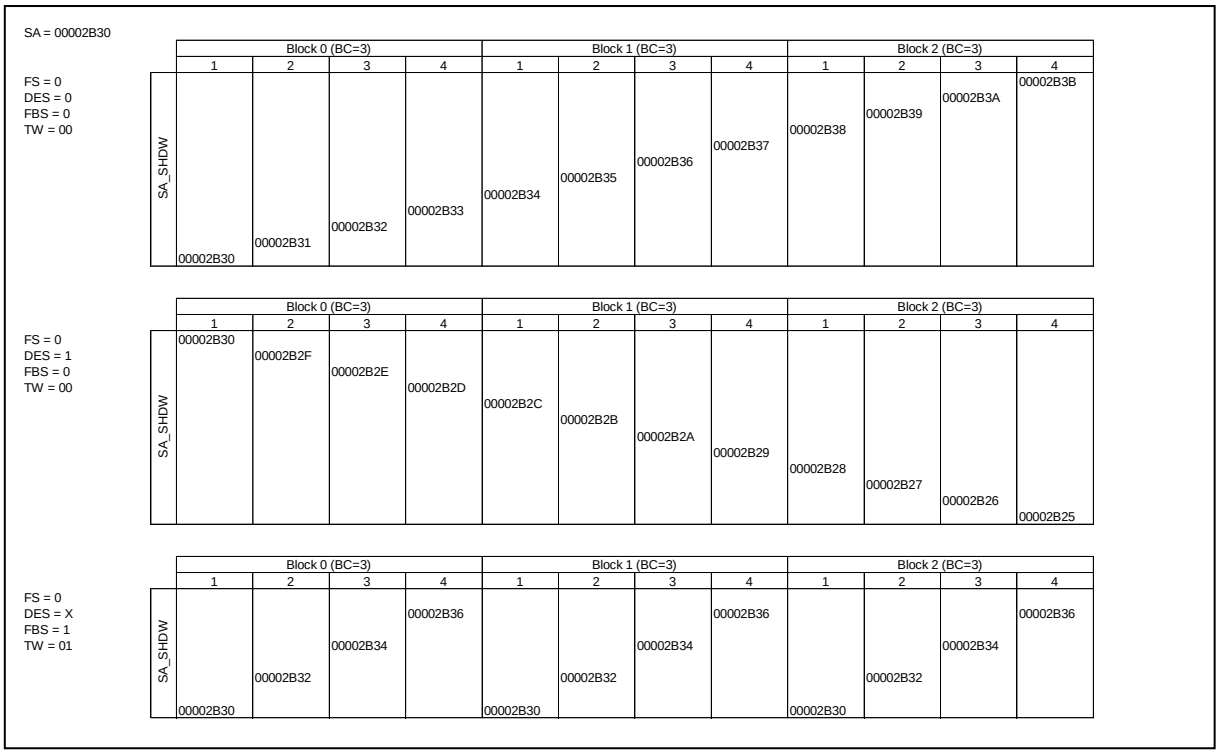
Table 3-2 Source Address Shadow Update Behavior

DMAi_Dn:FS	DMAi_Dn:DES	DMAi_Dn:FBS	Description of DMAi_SASHDWn:SASHDW Update Behavior
0	0	0	SASHDW is incremented at each successful read data transfer. Size of address increment depends on Transfer Width.
0	1	0	SASHDW is decremented at each successful read data transfer. Size of address decrement depends on Transfer Width.
0	X	1	SASHDW is incremented at each successful read data transfer. SASHDW is updated with the value stored in DMAi_SAn at the end of a block.
1	X	X	SASHDW remains constant.

Table 3-3 Destination Address Shadow Update Behavior

DMAi_Dn:FD	DMAi_Dn:DED	DMAi_Dn:FBD	Description of DMAi_DASHDWn:DASHDW Update Behavior
0	0	0	DASHDW is incremented at each successful write data transfer. Size of address increment depends on Transfer Width.
0	1	0	DASHDW is decremented at each successful write data transfer. Size of address decrement depends on Transfer Width.
0	X	1	DASHDW is incremented at each successful write data transfer. DASHDW is updated with the value stored in DMAi_DAn at the end of a block.
1	X	X	DASHDW remains constant.

Figure 3-1 Illustration of DMAi_SASHDWn:SASHDW Update



At the successful end of a DMA transfer Source Address or Destination Address can be updated with the value stored in Source Address Shadow or Destination Address Shadow respective. This can be configured with the bits Update Source Address (DMAi_Dn:US) or Update Destination Address (DMAi_Dn:UD).

DMA Transfer Size

The DMA transfer size is calculated by the following formula:

DMA transfer size [byte] = Number of data transfers * ($2^{\text{Transfer Width/8-1}}$)

= (DMAi_An:BC + 1) * (DMAi_An:TC + 1) * ($2^{\text{DMAi_Bn:TW}}$)

Transfer Count (DMAi_An:TC[15:0]) determines the number of blocks to be transferred in a DMA transfer. Block Count (DMAi_An:BC[3:0]) determines the number of data transfers in each block.

DMA Transfer Completion and Error Handling

Each DMA channel issues an interrupt at the end of a DMA transfer. This can be either a completion interrupt if the DMA transfer completed successfully or an error interrupt in case of an error condition or a stop request. A completion interrupt is signaled with flag of DIRQ (DMAi_Bn:DQ) and an error interrupt with flag of EDIRQ (DMAi_Bn:EQ). There is a DMA transfer end code associated with each interrupt which is encoded in Stop Status (DMAi_Bn:SS[2:0]).

If a DMA transfer is completed successfully and the completion interrupt raised, Stop Status will show 'normal end' (DMAi_Bn:SS = 0b101). If it is ended in error and the error interrupt raised, Stop Status will show one of the following possibilities:

- Stop request (DMAi_Bn:SS = 0b010)
- Source access error (DMAi_Bn:SS = 0b011)
- Destination access error (DMAi_Bn:SS = 0b100)

A 'stop request' during a running DMA transfer can be caused by assertion of the stop request signal (DSTP) of the DMA transfer requesting client, if the DMA channel is disabled (DMAi_An:EB), if the complete DMA Controller is disabled (DMAi_R:DE), or if a debug event occurs and the DMA Controller is configured to stop on a debug event.

Both interrupts, completion as well as error interrupt can be masked with bits Completion Interrupt (DMAi_Bn:CI) and Error Interrupt (DMAi_Bn:EI) respective. If these bits are set to "1" the interrupts are not masked. All unmasked completion interrupts are ORed and signaled to the Interrupt Controller. The same is done for the error interrupts.

All completion Interrupt Flags are in addition to the channel registers, available in two 32-bit registers (DMAi_DIRQ1 and DMAi_DIRQ2) for easier software handling. All Error Interrupts are handled in the same way and are available in register DMAi_EDIRQ1 and DMAi_EDIRQ2.

Completion interrupt DMAi_Bn:DQ must be cleared by setting Clear DIRQ (DMAi_Cn:CD). Error interrupt EQ must be cleared by setting Clear EDIRQ (DMAi_Cn:CE). Stop Status will be cleared to 'initial value' (DMAi_Bn:SS = 0b000) if DMAi_Bn:DQ or DMAi_Bn:EQ is set to "1".

Note:

– *For new transfer request accepted during on-going DMA transfer or after previous DMA transfer completion, this DMA controller starts the next transfer by clearing the interrupt flag. Therefore, in case it is unnecessary to start DMA transfer for next transfer request, it is required to disable DMA channel and clear transfer request of DMA client, and then clear the interrupt flag.*

Source and Destination Protection

Each DMA channel has the possibility to define source and destination protection information independently. This information will be used by the AHB master and driven on the AHB protection control signals (HPROTM[3:0]) for use by peripherals which have implemented access protection as defined by the AHB protocol. Protection information must be provided by Source Protection (DMAi_Bn:SP[3:0]) or Destination Protection (DMAi_Bn:DP[3:0]) if used. DMA Controller does only data transfers thus DMAi_Bn:SP[0] and DMAi_Bn:DP[0] are statically set to "1". Initial value indicates a 'Not cacheable/Not bufferable/Privileged access/Data access' source and destination protection.

Channel Disabling and Halting

After reset a DMA channel is disabled by default in order to properly configure it before a DMA request is serviced. DMA channel is enabled by setting Channel Enable (DMAi_An:EB) to "1". After setting DMAi_An:EB to "1" the channel waits for a DMA request.

Each DMA channel can be independently disabled. This is done by setting Channel Enable (DMAi_An:EB) to "0". Setting this bit to "0" can be done at any time but has different effects when it is done. If DMAi_An:EB is set to "0" during a running DMA transfer, the transfer is stopped at the next transfer gap, an error interrupt is raised and the channel is disabled. Transfer gap means the DMAC has transferred a block of data and the AHB master interface releases the bus request for a few cycles.

When DMAi_An:EB is set to "0" while an interrupt is pending (DMAi_Bn:DQ = 1 or DMAi_Bn:EQ = 1) or no DMA transfer is running there will be no other effect besides the channel is disabled.

Channel halting is done by setting Pause Bit (DMAi_An:PB) to "1". If this bit is set to "1" during a running DMA transfer, it will halt after completion of the current transferred block. If it is set to "1" before receiving a transfer request the halt state is entered immediately. Clearing this bit will put the channel into run state and it will wait for the next transfer request to continue the DMA transfer or if a transfer request is already pending, it will continue immediately.

Burst Transfer Mode

Burst transfer mode is almost identical to the block transfer mode. The only difference is the request of a DMA transfer. Whereas in block transfer mode one request for each block of data is needed, in burst transfer mode only one request is needed at the begin of a DMA transfer for the complete transfer. The requests needed for subsequent blocks of data is generated internally by the DMA Controller itself.

3.2. DMA Client Matrix

This section describes configuration and behavior of the DMA Client Matrix.

Overview

The DMA Client Matrix provides the possibility to route 'M' DMA clients to 'N' DMA channels. 'M' is greater than or equal to 'N'. The selection which DMA channel serves which DMA client will be set with Client Interface (DMAi_CMCHICn:CI). The configuration of the internal DMA clients will be done with the registers DMAi_CMICICm.

Modes of Operation

Each DMA Client Interface can work in one of the following modes:

1. Disabled mode

   Internal DMA clients:

   An internal DMA Client Interface is disabled if it is not selected by any of the DMA Client Matrix Channel Configuration Registers (DMAi_CMCHICn:CI). Reconfiguration of the internal DMA Client Interface shall only be done in disabled mode.

2. Normal mode

In this mode a DMA channel is routed directly to the specified (DMAi_CMCHICn:CI) DMA client. The operation of the DMA Client Matrix in this mode is fully transparent and behaves as if the DMA client would be connected directly to the DMA Channel Interface.

Functional Description

Purpose of the DMA Client Matrix is to provide flexibility in the use of available DMA channels. The configuration of the DMA Client Matrix is intended to be static and shall be done after the boot code execution when the software is setting up the system.

Structure of the DMA Client Matrix

The DMA Client Matrix will be a full matrix where each DMA Client Interface 'm' can be routed to every DMA Channel Interface 'n'.

DMA Client Matrix Configuration

The configuration of the DMA Client Matrix is done with the following registers:

- DMAC Client Matrix Internal Client Configuration Registers (DMAi_CMICICm)
- DMAC Client Matrix Channel Interface Configuration Registers (DMAi_CMCHICn)

The DMAC Client Matrix Internal Client Interface Configuration Register contains two signal behavior bits. For their function see the descriptions below:

The Config bit Behavior Request Acknowledge, DMAi_CMICICm:BEHREQACK, sets the behavior of the output signal DREQ_ACK[m] if the internal DMA Client Interface 'm' is not selected in any of the Channel Configuration Registers (DMAi_CMCHICn). The user can choose that DREQ_ACK[m] drives inactive level or that DREQ[m] is connected to DREQ_ACK[m] in that case. The later can be used to reset a, due to software misbehavior, falsely set DMA request signal without violating the two-way handshake protocol.

The Config bit Behavior Stop Acknowledge, DMAi_CMICICm:BEHSTPACK, sets the behavior of the output signal DSTP_ACK[m] if the internal DMA Client Interface 'm' is not selected in any of the channel Configuration Registers (DMAi_CMCHICn:CI). The user can choose that DSTP_ACK[m] drives inactive level or that DSTP[m] is connected to DSTP_ACK[m] in that case.

The DMAC Client Matrix Channel Interface Configuration Register contains up to nine selection bits. For their function see the description below:

The Selection bits Client Interface, DMAi_CMCHICn:CI, specify which DMA Client Interface 'm' is connected to the DMA Channel Interface 'n'. The configuration of these bits must take place before DMAi_R:DE and DMAi_An:EB is set to "1". The client interface number must be programmed as binary value to DMAi_CMCHICn:CI. Setting of CI makes the connection between DMA Client Interface defined by the value of CI and DMA Channel Interface 'n'. Selecting twice or more times the same DMA Client Interface in any of the DMA Client Matrix Channel Configuration Registers results in unpredictable behavior of the DMAC and must be avoided.

Availability of certain DMA clients depends on specific device. Refer to "CHAPTER: Overview" for the availability of DMA clients.

Automatic Reconfiguration of DMA Client Matrix

DMA Client Matrix can be reconfigured by a request from DMA Additional Control block. When DMA Additional Control Block launch a request to reconfigure the DMA Client Matrix (CMCHIC reload event trigger), DMAi_CMCHICn register value is automatically updated by the value in the specified bank.

Please refer "CHAPTER : DMA COMPLEX SUBSYSTEM" for more details.

Initialization and Application Information

Reset

The reset state of each DMA Client Matrix Configuration bit is shown in the register description of DMAi_CMICICm, and DMAi_CMCHICn. To summarize it, after hardware reset, all internal DMA Client Interfaces are disabled, all signals set to high-active level, and DMA Channel Interface 0 - 'N-1' is configured to route to DMA Client Interface 0 - 'N-1'.

Note:

- *Selecting the same DMA Client Interface in two or more DMA Channel Interfaces leads to unpredictable behavior of the DMAC. Therefore DMAi_CMCHICn:CI must be properly configured before enabling the DMAC and one or more of its channels.*

3.3. DMA Arbiter

This section describes configuration and behavior of the DMA arbiter which chooses a DMA channel based on the arbitration scheme.

Overview

The DMA arbiter is responsible for choosing a DMA channel based on the arbitration scheme selected in the Global Configuration Register (DMAi_R:PR). There are three arbitration schemes available:

- Fixed priority
- Dynamic priority
- Round-robin

The arbitration schemes are explained in detail in the following sections. The arbitration scheme can be changed any time, however it becomes effective only after the current running data transfer has been completed at the next transfer gap.

About the timing of the priority judgement, it is judged after the end of transferring 1 block, and before the start of transferring the next block.

Fixed Priority

In fixed priority arbitration scheme the DMA channels have a 'fixed' priority which can be set with Priority Number (DMAi_Bn:PN). Priority Number equal to "0" has the highest priority, whereas Priority Number equal to 127 has the lowest priority. DMA channels with equal Priority Number, the channel with the lowest channel number 'n' has the highest priority. The initial value of DMAi_Bn:PN is 127. Priority Number can be changed any time, however it becomes effective only after the current running data transfer has been completed at the next transfer gap. Fixed priority arbitration example shows an arbitration example for eight DMA channels to illustrate the behavior.

Table 3-4 Fixed Priority Arbitration Scheme

Arbitration Cycle	Requesting DMA Channel 'n'	PN of Requesting DMA Channel 'n'	Grant Given to DMA Channel 'n'
X+1	2	0	2
	4	2	
	7	6	
X+2	4	2	4
	7	6	
	8	5	
X+3	4	2	4
	7	6	
	8	5	
X+4	1	5	1
	7	6	
	8	5	
X+5	3	9	8
	7	6	
	8	5	

Dynamic Priority

The dynamic priority arbitration scheme is an extension of the fixed priority arbitration scheme. The priority of the DMA channels is dynamically adjusted based on the criterion whether a channel got a grant or not. If a channels request was granted its dynamic Priority Number is loaded with the Priority Number stored in DMAi_Bn:PN and if a channels request was not granted its dynamic Priority Number is decremented by "1". The arbiter is giving grant to the requesting DMA channel with the lowest dynamic Priority Number. If two or more requesting channels have equal dynamic Priority Numbers the DMA channel with the lowest channel number 'n' has the highest priority and will win the arbitration process. Priority Number can be changed any time, however it becomes effective after it has been re-loaded by the channel, i.e. after the channel has won the arbitration. Dynamic priority arbitration example shows an arbitration example for four DMA channels to illustrate the behavior.

Note:

- *In case a channel's re-programmed Priority Number should become effective immediately (not only after the re-programmed channel has won the arbitration), this behavior can be forced by changing the arbitration scheme to Fixed priority and then changing it back to dynamic priority while in halt state. However, it must be noted that the dynamic priority of all channels will be re-loaded with Priority Number (DMAi_Bn:PN) and not only the re-programmed channels.*

Table 3-5 Dynamic Priority Arbitration Scheme

Arbitration Cycle	Requesting DMA Channel		Dynamic PN of DMA Channel	PN of DMA Channel	Grant Given to DMA Channel
1	Ch. 0	Yes	1	1	0
	Ch. 1	Yes	2	2	
	Ch. 2	Yes	3	3	
	Ch. 3	No	3	3	
2	Ch. 0	No	1	1	1
	Ch. 1	Yes	1	2	
	Ch. 2	Yes	2	3	
	Ch. 3	No	3	3	
3	Ch. 0	No	1	1	2
	Ch. 1	No	2	2	
	Ch. 2	Yes	1	3	
	Ch. 3	Yes	3	3	
4	Ch. 0	No	1	1	1
	Ch. 1	Yes	2	2	
	Ch. 2	No	3	3	
	Ch. 3	Yes	2	3	
5	Ch. 0	No	1	1	3
	Ch. 1	No	2	2	
	Ch. 2	Yes	3	3	
	Ch. 3	Yes	1	3	

Round-robin

In round-robin arbitration scheme the turn is rotated in directional and cyclic order from DMA channel 0 to DMA channel 'n'. At most one DMA channel request can be granted at any time, this is defined as a turn being given. The turn is moved forward at each transfer gap. The turn's rotation is not strictly

round-robin in order not to waste an arbitration phase by giving the turn to a non-requesting DMA channel. Instead the turn is given to the next requesting DMA channel in the rotation direction. If no DMA channel was served last (only possible in initial state) the requesting DMA channel with the lowest channel number 'n' has the highest priority and will win the arbitration process. Round-robin arbitration example shows an arbitration example for eight DMA channels to illustrate the behavior.

Table 3-6 Round-Robin Priority Arbitration Scheme

Arbitration Cycle	Requesting DMA Channel	Grant Given to DMA Channel	Last Served DMA Channel
1	2	2	None
	4		
	7		
2	4	4	2
	7		
	8		
3	4	7	4
	7		
	8		
4	1	8	7
	4		
	8		
5	1	1	8
	4		
	7		

Application Information

Fixed Priority Arbitration

With this arbitration scheme the DMA channel request from the channel with the highest priority will be selected for service. If the DMAC is programmed that channel 0 is assigned the highest priority and this channel has a higher service request rate compared to the other channels, it is possible that this channel absorbs the complete bandwidth of the DMA Controller, means that the other channels will not be serviced.

Dynamic Priority Arbitration

With this arbitration scheme starving is tried to be avoided by assigning a requesting channel which got no grant the next higher priority level. However starving cannot be avoided if the DMAC is not programmed properly i.e. if channel 0 is assigned the highest priority the other channels cannot reach a higher priority than channel 0. If this channel has in addition a higher service request rate that the other channels, it will use the complete bandwidth of the DMAC.

Round-Robin Arbitration

With this arbitration scheme starving of requesting channels is not possible even if channel 0 has a service request rate, which is equal to or exceeds the arbitration rate.

3.4. DMA AHB Slave Interface

This section describes information about the slave interface of the DMAC.

This is the DMA Controller's system interface through which the DMACs registers are accessed.

Supported Data Transfers

The slave interface supports 8-, 16-, and 32-bit wide AHB data transfers. 16-bit and 32-bit accesses shall be 16-bit address respective 32-bit address aligned.

Single data and fixed incremental burst accesses are supported (SINGLE, INCR4, INCR8, and INCR16).

Note:

- *DMAi_Dn register must be accessed by 8-bit access.*
- *All writing to DMA Controller registers must be done in privileged mode.*

Data Transfer Response

The DMA AHB slave interface responds with the following possibilities to any kind of access:

- OKAY response
- ERROR response

The ERROR response will be given for accesses where a protection error or a register access error occurs.

Protection Error

A protection error is raised if any of following conditions is met.

- Writing registers in user mode.

Register Access Error

A register access error is raised if a read or write to a reserved address location is attempted. For the location of the reserved addresses see Memory layout of DMA Controller registers.

A register access error is raised if a write access is attempted to read only registers.

3.5. Additional Information

This section gives additional information for using DMAC.

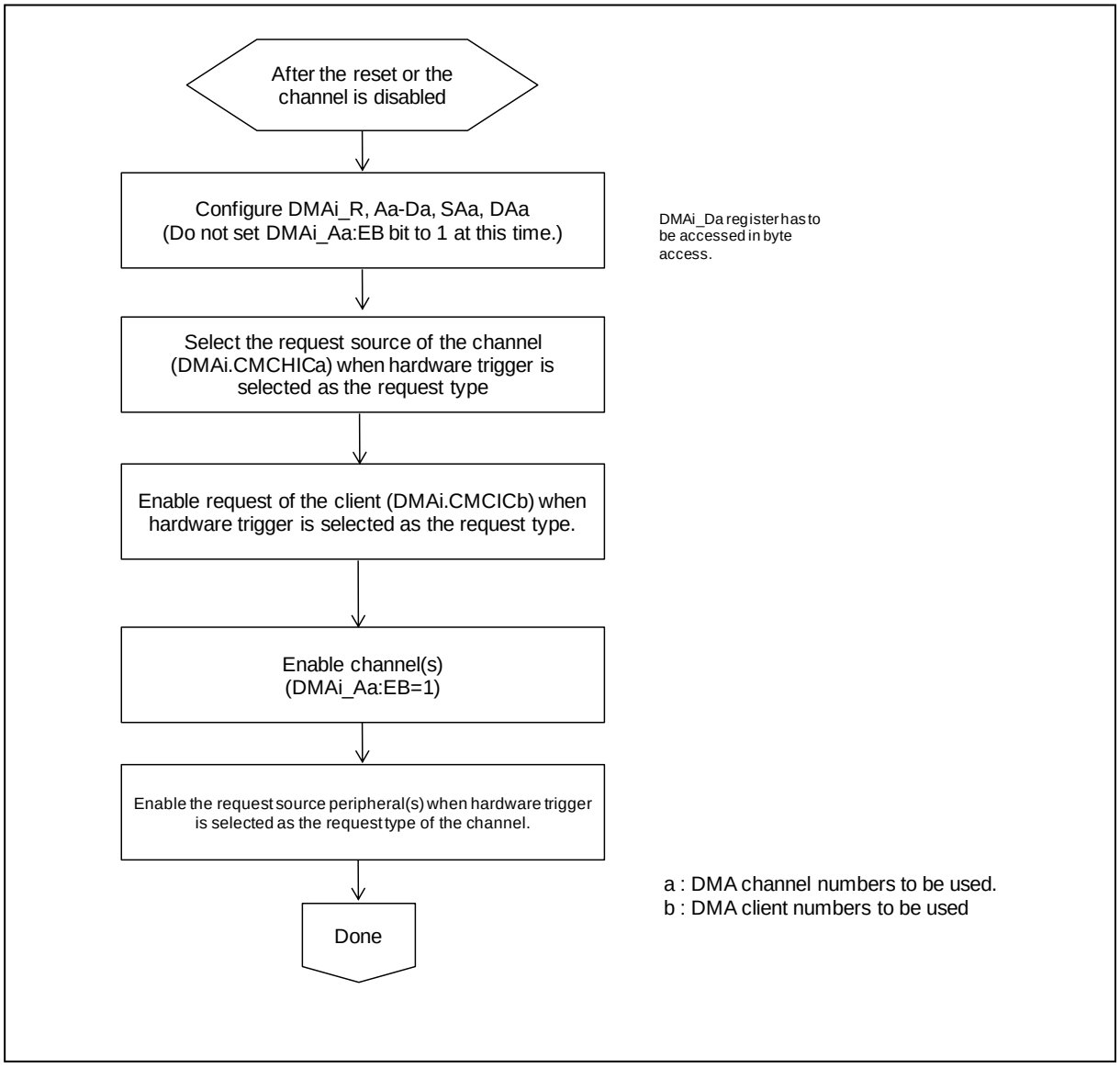
Stopping Clock for DMA Controller or entering PSS(Power Saving State) Mode

- Before stopping clock for DMA controller or entering PSS(Power Saving State) mode, please make sure followings :
  - All DMAC channels are disabled and their operations are completely stopped.
  - No pending or ongoing register access to DMA controller.
- Do not access DMA controller registers before the clock for DMA resumes.

Recommended Flow

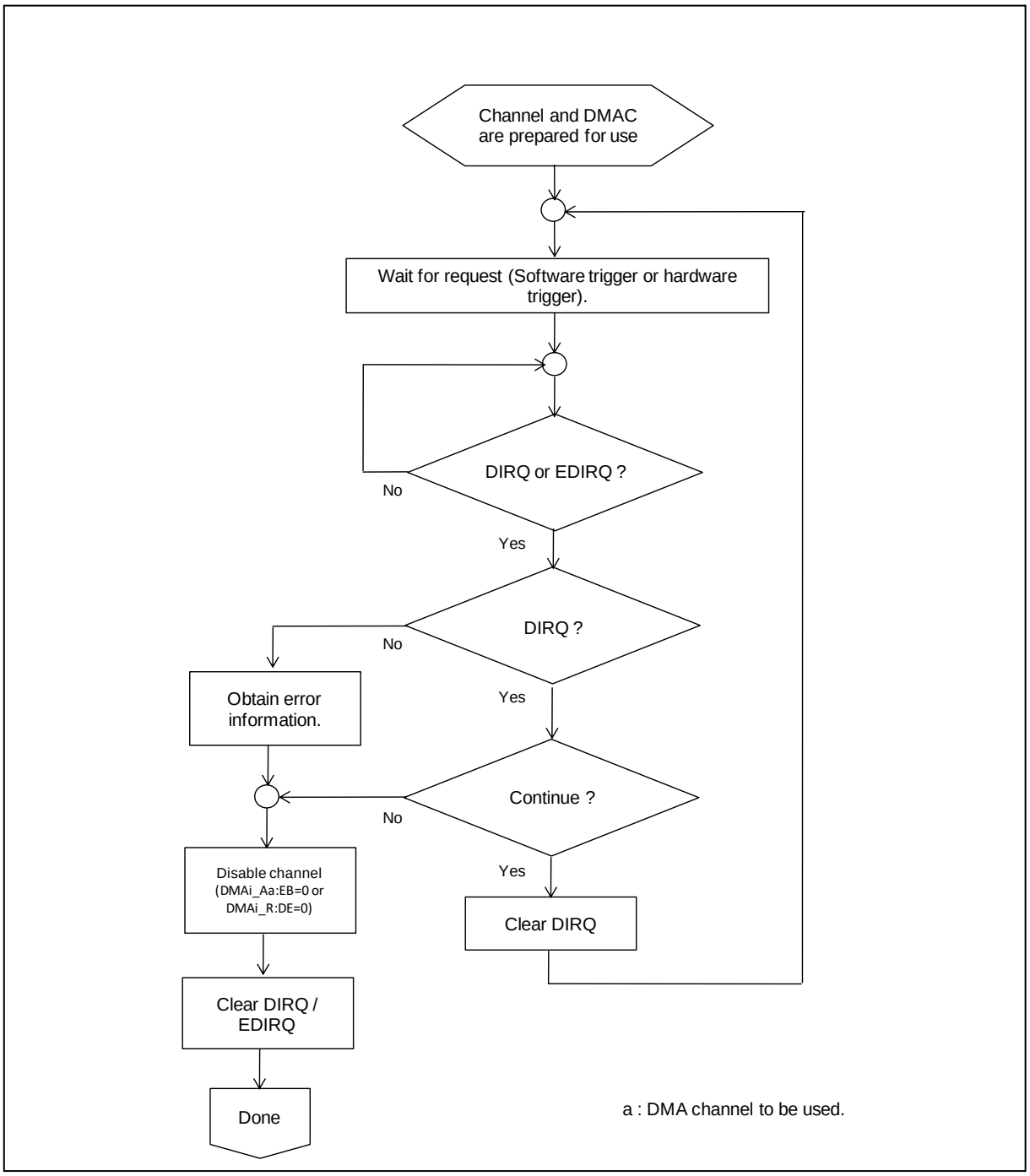
Configuring and Enabling a DMA Channel

Figure 3-2 Recommended Flow for Configuring and Enabling a DMA Channel(s)



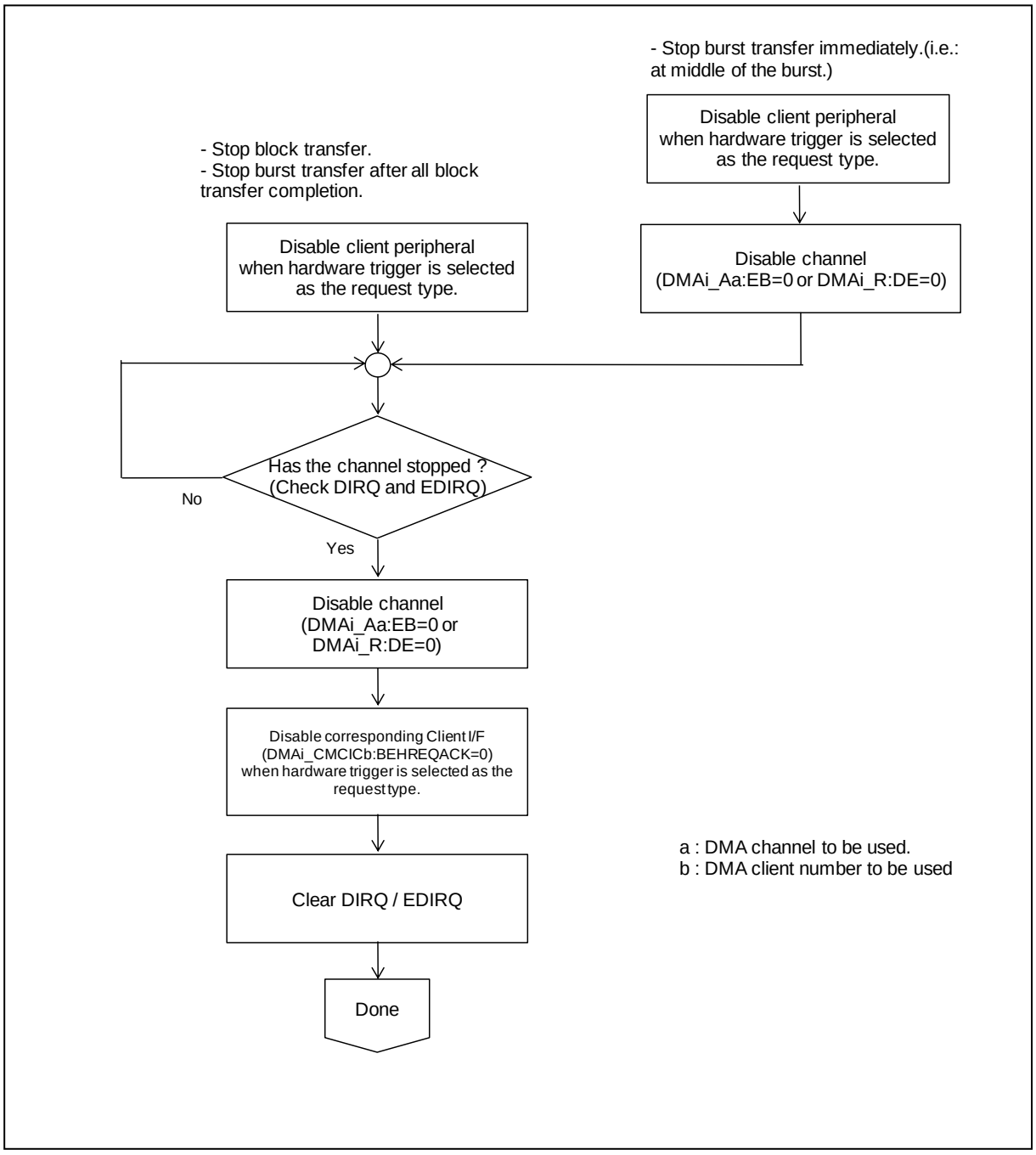
Handling DMAC Requests

Figure 3-3 Recommended Flow for Handling DMAC Requests



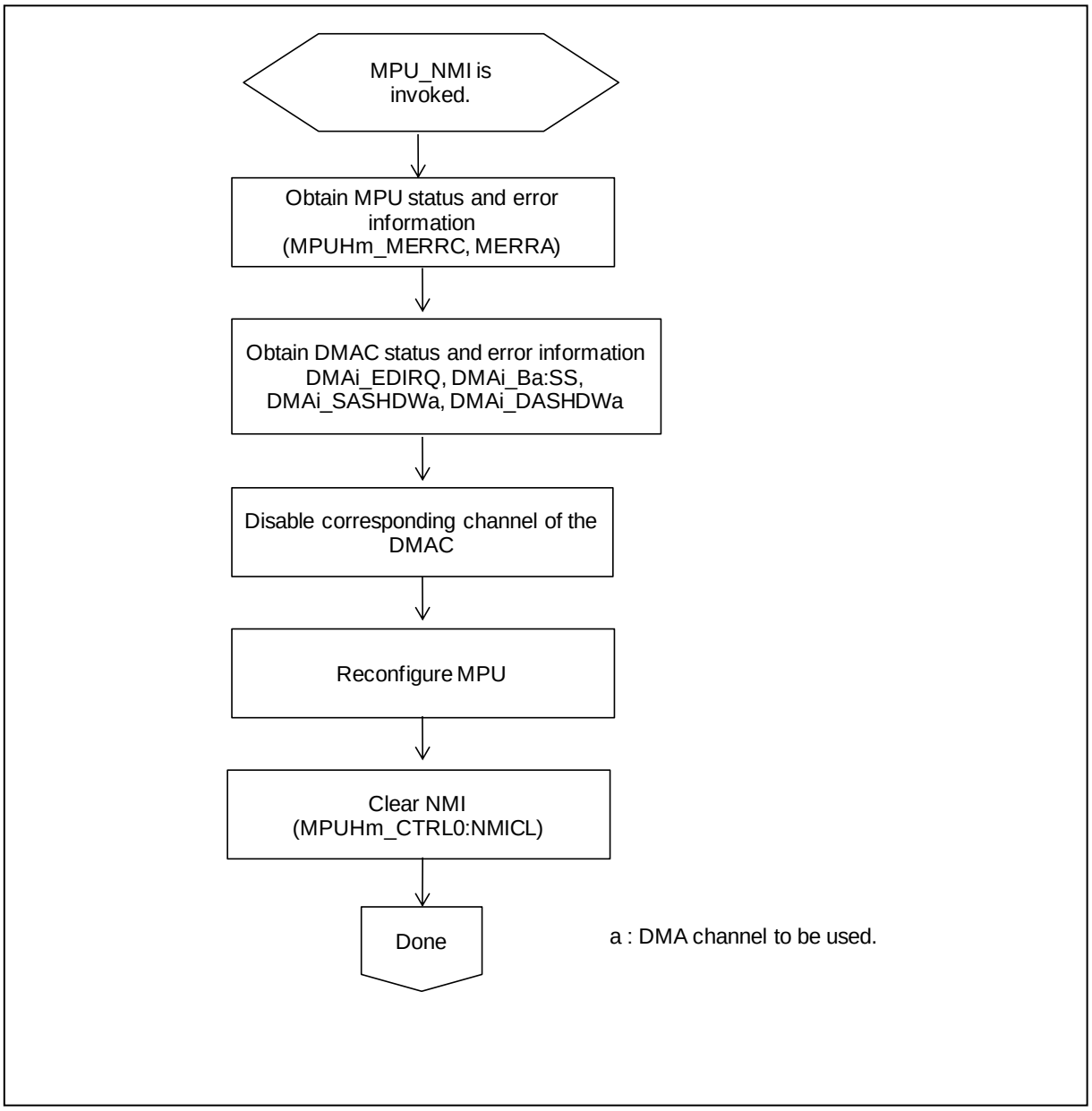
Stop Transfer

Figure 3-4 Recommended Flow for Stopping Transfer



Handling DMAC_MPU NMI

Figure 3-5 Recommended Flow for Handling DMAC_MPU NMI



4. Registers

This section gives the explanation of DMAC.

Table 4-1 List of Register Area

Offset	+3	+2	+1	+0
0x0000_0000(DMAC#0) 0x0000_4000(DMAC#1) + (n * 0x40)	DMAi_An 00000000_00001111_00000000_00000000			
0x0000_0004(DMAC#0) 0x0000_4004(DMAC#1) + (n * 0x40)	DMAi_Bn 00000000_00000000_00110011_01111111			
0x0000_0008(DMAC#0) 0x0000_4008(DMAC#1) + (n * 0x40)	DMAi_SAn 00000000_00000000_00000000_00000000			
0x0000_000C(DMAC#0) 0x0000_400C(DMAC#1) + (n * 0x40)	DMAi_DAn 00000000_00000000_00000000_00000000			
0x0000_0010(DMAC#0) 0x0000_4010(DMAC#1) + (n * 0x40)	DMAi_Cn 00000000_00000000_00000000_00000000			
0x0000_0014 (DMAC#0) 0x0000_4014(DMAC#1) + (n * 0x40)	DMAi_Dn 00000000_00000000_00000000_00000000			
0x0000_0018(DMAC#0) 0x0000_4018(DMAC#1) + (n * 0x40)	DMAi_SASHDWn 00000000_00000000_00000000_00000000			
0x0000_001C(DMAC#0) 0x0000_401C(DMAC#1) + (n * 0x40)	DMAi_DASHDWn 00000000_00000000_00000000_00000000			
0x0000_0020(DMAC#0) 0x0000_4020(DMAC#1) + (n * 0x40)	DMAi_En 00000000_00000000_00000000_00000000			
0x0000_0024(DMAC#0) 0x0000_4024(DMAC#1) + (n * 0x40) to 0x0000_003C(DMAC#0) 0x0000_403C(DMAC#1) + (n * 0x40)	Reserved			
0x0000_1000(DMAC#0) 0x0000_5000(DMAC#1)	DMAi_R 01000000_00000000_00000000_00000001			
0x0000_1004(DMAC#0) 0x0000_5004(DMAC#1)	DMAi_DIRQ1 00000000_00000000_00000000_00000000			
0x0000_1008(DMAC#0) 0x0000_5008(DMAC#1)	DMAi_DIRQ2 00000000_00000000_00000000_00000000			
0x0000_100C(DMAC#0) 0x0000_500C(DMAC#1)	DMAi_EDIRQ1 00000000_00000000_00000000_00000000			

Offset	+3	+2	+1	+0
0x0000_1010(DMAC#0) 0x0000_5010(DMAC#1)	DMAi_EDIRQ2 00000000_00000000_00000000_00000000			
0x0000_1014(DMAC#0) 0x0000_5014(DMAC#1)	DMAi_ID 00000000_00000001_00000101_00000000			
0x0000_1018(DMAC#0) 0x0000_5018(DMAC#1) to 0x0000_201F(DMAC#0) 0x0000_601F(DMAC#1)	Reserved			
0x0000_2020 (DMAC#0) 0x0000_6020(DMAC#1) The reserve area is changed by the following arithmetic expression. + ((m-8)*0x04)	DMAi_CMICICm 00000000_00000000_00000000_00000000			
0x0000_223C(DMAC#0) 0x0000_623C(DMAC#1) to 0x0000_27FF(DMAC#0) 0x0000_67FF(DMAC#1)	Reserved			
0x0000_2800(DMAC#0) 0x0000_6800(DMAC#1) + (n*0x04)	DMAi_CMCHICn 00000000_00000000_00000000_0000*1			
0x0000_2840(DMAC#0) 0x0000_6840(DMAC#1)	Reserved			

'i' represents instance number of DMAC. i = 0

'n' represents channel number. n = 0, 1, 2, …,15.

'N' represents number of DMAC channels per DMAC instance. N = 16 in this product.

'm' represents client number. m = 8, 9, …, (product specification).

'M' represents number of client I/F. M = (product specification) in this product.

Notes:

- *Please refer "CHAPTER: Overview" for the maximum numbers.*
- **1: Initial value of DMAi_CMCHICn:CI[7:0] is 'n'.*

4.1. DMA Controller Global Configuration Register (DMAi_R)

This register handles the enabling and halting of the entire DMA Controller, arbitration scheme of the DMA channel arbiter can be chosen, enabling of debug function, and the DMACs behavior in case of a 'debug event' can be selected.

bit	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24
Field	DE	DSHR	DBE	PR[1]	PR[0]	DH	DB[1]	DB[0]
Attribute	R/W	R,WX	R/W	R/W	R/W	R/W	R/W	R/W
Protection	WP							
Initial Value	0	1	0	0	0	0	0	0

bit	bit 23	bit 22	bit 21	bit 20	bit 19	bit 18	bit 17	bit 16
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
Protection	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
Protection	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	DSHS
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R,WX
Protection	WP							
Initial Value	0	0	0	0	0	0	0	1

[bit31] DE : DMA enable(DE)

If this bit is set to "0", the DMAC is disabled. This also means that all DMA channels are disabled independent of the settings of DMAi_An:EB.

If this bit is set to "1", the DMAC is enabled and the enabling of the channels depend on the setting of DMAi_An:EB.

When this bit is set to "0" during a DMA transfer, the channel which is in the middle of a transfer stops at the next transfer gap.

The transfer gap means that DMAC de-asserts bus request to the bus arbiter for a short time in the middle of a DMA transfer after a block of data has been transferred. This ensures that the bus is not completely blocked by a very long DMA transfer.

Bit	Description
0	DMAC globally disabled
1	DMAC globally enabled

[bit30] DSHR : DMA stop/halt request flag

This bit indicates that the DMA transfers of all channels have been requested to halt or disable.

This bit is set to "0" by hardware if none of the conditions below are true.

This bit is set to "1" by hardware if one or more of the conditions below are true.

Conditions for DMA Stop/Halt Request Flag:

- DMAi_R:DE is set to "0" (all channels are disabled)
- DMAi_R:DH is set to "1" (all channels are halted)
- DMAi_R:DBE is set to "1", DMAi_R:DB is set to '10' (stop on debug events), and a debug event is pending
- DMAi_R:DBE is set to "1", DMAi_R:DB is set to '01' (halt on debug events), and a debug event is pending

Bit	Description
0	Indicates that global halt/disable condition of DMAC is removed
1	Indicates that DMA transfers of all channels are requested to halt or disable

[bit29] DBE : Debug enable

This bit determines whether DMAC reacts on a debug event (i.e. debugger break point).

The behavior of the DMAC on the occurrence of a debug event depends on configuration bits, Debug Behavior (DMAi_R:DB).

Bit	Description
0	DMAC does not react on debug events
1	DMAC reacts on debug events. Reaction depends on setting of Debug Behavior (DMAi_R:DB)

[bit28:27] PR[1:0] : Priority type

These bits select the arbitration scheme of the DMAC arbiter. In case of dynamic priority, channel priority is updated at the transfer gap.

Bits	Description
00	Fixed priority
01	Dynamic priority
10	Round robin
11	Reserved

[bit26] DH : DMA halt

When this bit is set to a "1", all DMA channels are halted and do not perform DMA transfers until this bit is set back to "0". After it is cleared the halted DMA transfers continue at the point they were halted.

If this bit is set to "1" while a DMA transfer is ongoing, DMAC halts the transfer at the next transfer gap.

About the transfer gap, refer to the description of DMAi_R:DE bit.

[bit25:24] DB[1:0] : Debug behavior

Bits	Description
00	DMAC continues on debug event
01	DMAC halts all transfers on debug event
10	DMAC stops all transfers on debug event
11	Reserved

[bit0] DSHS : DMA stop/halt status flag

Bit	Description
0	Indicate that DMA transfer of at least one channel is running
1	Indicate that DMA transfers of all channels are halted or disabled

4.2. DMA Controller Global Completion Interrupt 1 Register (DMAi_DIRQ1)

The DMA Controller Global Completion Interrupt 1 Register combines the completion Interrupt Flags (DMAi_Bn:DQ) from DMA channels 0 to 31.

bit	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24
Field	DIRQ[31]	DIRQ[30]	DIRQ[29]	DIRQ[28]	DIRQ[27]	DIRQ[26]	DIRQ[25]	DIRQ[24]
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	bit 23	bit 22	bit 21	bit 20	bit 19	bit 18	bit 17	bit 16
Field	DIRQ[23]	DIRQ[22]	DIRQ[21]	DIRQ[20]	DIRQ[19]	DIRQ[18]	DIRQ[17]	DIRQ[16]
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
Field	DIRQ[15]	DIRQ[14]	DIRQ[13]	DIRQ[12]	DIRQ[11]	DIRQ[10]	DIRQ[9]	DIRQ[8]
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Field	DIRQ[7]	DIRQ[6]	DIRQ[5]	DIRQ[4]	DIRQ[3]	DIRQ[2]	DIRQ[1]	DIRQ[0]
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection	WP							
Initial Value	0	0	0	0	0	0	0	0

[bit31:0] DIRQ : Global completion interrupt 1

This is a read-only register which gives the status of DIRQ of channels 0 to 31. Channels which are not available on a particular device reads "0".

Note:

- *Channels 16 to 63 are not available in this product.*

4.3. DMA Controller Global Completion Interrupt 2 Register (DMAi_DIRQ2)

The DMA Controller Global Completion Interrupt 2 Register combines the completion Interrupt Flags (DMAi_Bn:DQ) from DMA channels 32 to 63.

bit	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24
Field	DIRQ[63]	DIRQ[62]	DIRQ[61]	DIRQ[60]	DIRQ[59]	DIRQ[58]	DIRQ[57]	DIRQ[56]
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	bit 23	bit 22	bit 21	bit 20	bit 19	bit 18	bit 17	bit 16
Field	DIRQ[55]	DIRQ[54]	DIRQ[53]	DIRQ[52]	DIRQ[51]	DIRQ[50]	DIRQ[49]	DIRQ[48]
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
Field	DIRQ[47]	DIRQ[46]	DIRQ[45]	DIRQ[44]	DIRQ[43]	DIRQ[42]	DIRQ[41]	DIRQ[40]
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Field	DIRQ[39]	DIRQ[38]	DIRQ[37]	DIRQ[36]	DIRQ[35]	DIRQ[34]	DIRQ[33]	DIRQ[32]
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection	WP							
Initial Value	0	0	0	0	0	0	0	0

[bit31:0] DIRQ : Global completion interrupt 2

This is a read-only register which gives the status of DIRQ of channels 32 to 63. Channels which are not available on a particular device reads "0".

Note:

- *Channels 16 to 63 are not available in this product.*

4.4. DMA Controller Global Error Interrupt 1 Register (DMAi_EDIRQ1)

The DMA Controller Global Error Interrupt 1 Register combines the Error Interrupt Flags (DMAi_Bn:EQ) from DMA channels 0 to 31.

bit	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24
Field	EDIRQ[31]	EDIRQ[30]	EDIRQ[29]	EDIRQ[28]	EDIRQ[27]	EDIRQ[26]	EDIRQ[25]	EDIRQ[24]
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	bit 23	bit 22	bit 21	bit 20	bit 19	bit 18	bit 17	bit 16
Field	EDIRQ[23]	EDIRQ[22]	EDIRQ[21]	EDIRQ[20]	EDIRQ[19]	EDIRQ[18]	EDIRQ[17]	EDIRQ[16]
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
Field	EDIRQ[15]	EDIRQ[14]	EDIRQ[13]	EDIRQ[12]	EDIRQ[11]	EDIRQ[10]	EDIRQ[9]	EDIRQ[8]
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Field	EDIRQ[7]	EDIRQ[6]	EDIRQ[5]	EDIRQ[4]	EDIRQ[3]	EDIRQ[2]	EDIRQ[1]	EDIRQ[0]
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection	WP							
Initial Value	0	0	0	0	0	0	0	0

[bit31:0] EDIRQ : Global error interrupt 1

This is a read-only register which gives the status of EDIRQ of channels 0 to 31. Channels which are not available on a particular device reads "0".

Note:

- *Channels 16 to 63 are not available in this product.*

4.5. DMA Controller Global Error Interrupt 2 Register (DMAi_EDIRQ2)

The DMA Controller Global Error Interrupt 2 Register combines the Error Interrupt Flags (DMAi_Bn:EQ) from DMA channels 32 to 63.

bit	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24
Field	EDIRQ[63]	EDIRQ[62]	EDIRQ[61]	EDIRQ[60]	EDIRQ[59]	EDIRQ[58]	EDIRQ[57]	EDIRQ[56]
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	bit 23	bit 22	bit 21	bit 20	bit 19	bit 18	bit 17	bit 16
Field	EDIRQ[55]	EDIRQ[54]	EDIRQ[53]	EDIRQ[52]	EDIRQ[51]	EDIRQ[50]	EDIRQ[49]	EDIRQ[48]
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
Field	EDIRQ[47]	EDIRQ[46]	EDIRQ[45]	EDIRQ[44]	EDIRQ[43]	EDIRQ[42]	EDIRQ[41]	EDIRQ[40]
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Field	EDIRQ[39]	EDIRQ[38]	EDIRQ[37]	EDIRQ[36]	EDIRQ[35]	EDIRQ[34]	EDIRQ[33]	EDIRQ[32]
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection	WP							
Initial Value	0	0	0	0	0	0	0	0

[bit31:0] EDIRQ : Global error interrupt 2

This is a read-only register which gives the status of EDIRQ of channels 32 to 63. Channels which are not available on a particular device reads "0".

Note:

- *Channels 16 to 63 are not available in this product.*

4.6. DMA Controller ID Register (DMAi_ID)

The DMA Controller ID Register always returns a constant. This register offers no function.

bit	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24
Field	MID[31]	MID[30]	MID[29]	MID[28]	MID[27]	MID[26]	MID[25]	MID[24]
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
Protection	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	bit 23	bit 22	bit 21	bit 20	bit 19	bit 18	bit 17	bit 16
Field	MID[23]	MID[22]	MID[21]	MID[20]	MID[19]	MID[18]	MID[17]	MID[16]
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R1,WX
Protection	WP							
Initial Value	0	0	0	0	0	0	0	1

bit	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
Field	MID[15]	MID[14]	MID[13]	MID[12]	MID[11]	MID[10]	MID[9]	MID[8]
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R1,WX	R0,WX	R1,WX
Protection	WP							
Initial Value	0	0	0	0	0	1	0	1

bit	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Field	MID[7]	MID[6]	MID[5]	MID[4]	MID[3]	MID[2]	MID[1]	MID[0]
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
Protection	WP							
Initial Value	0	0	0	0	0	0	0	0

[bit31:0] MID : Module number

Always return 0x00010500

4.7. DMA Controller Channel Configuration A Register Channel 'n' (DMAi_An)

The register specifies whether to enable or halt a DMA channel. It specifies the source of a DMA transfer request and DMA transfer parameters Block Count, Transfer Count, Beat Limit, Alternate, and Timeout.

bit	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24
Field	EB	PB	ST	IS[1]	IS[0]	AL	BL[1]	BL[0]
Attribute	R/W	R/W	R0,W	R/W	R/W	R/W	R/W	R/W
Protection	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	bit 23	bit 22	bit 21	bit 20	bit 19	bit 18	bit 17	bit 16
Field	BC[3]	BC[2]	BC[1]	BC[0]	TO[3]	TO[2]	TO[1]	TO[0]
Attribute	R/W	R/W	R/W	R/W	R/W1	R/W1	R/W1	R/W1
Protection	WP							
Initial Value	0	0	0	0	1	1	1	1

bit	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
Field	TC[15]	TC[14]	TC[13]	TC[12]	TC[11]	TC[10]	TC[9]	TC[8]
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Field	TC[7]	TC[6]	TC[5]	TC[4]	TC[3]	TC[2]	TC[1]	TC[0]
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection	WP							
Initial Value	0	0	0	0	0	0	0	0

[bit31] EB : Enable bit

This bit is used to enable/disable a DMA channel. If this bit is set to "1", the channel is enabled and waits for a request to start a DMA transfer (Before that, DMAi_R:DE bit needs to be set to "1" already). If this bit is set to "0", the channel is disabled and does not perform a DMA transfer. When this bit is set to "0" during a running DMA transfer which will not complete at the next transfer gap, the DMA transfer is terminated. This is regarded as a forced stop and an error interrupt is generated. When this bit is set to "0" while the last block of a DMA transfer is running, the DMA transfer completes at the next transfer gap and a completion interrupt is generated. About the transfer gap, refer to the description of DMAi_R:DE bit. This bit is useful to re-configure each configuration register of the channel after a DMA transfer.

Bit	Description
0	Channel is disabled
1	Channel is enabled

[bit30] PB : Pause bit

This bit is used to halt the transfer of the DMA channel. If this bit is set to "1", this channel halts the transfer and does not perform a DMA transfer until this bit is cleared.

When this bit is set to "1" while no transfer is ongoing DMAC enters the halt state immediately. When it was set to "1" during a running transfer, the halt state is entered at the next transfer gap. If the DMA transfer completes at the next transfer gap a completion interrupt is issued.

When this bit is set to "0" the halt condition is cleared and DMAC waits for the next request to continue the DMA transfer.

This bit is useful to halt a DMA transfer without re-configuration of each configuration register of the channel.

Bit	Description
0	Channel is not halted
1	Channel is halted

[bit29] ST : Software trigger

This bit is used to generate a software request. This bit can only be used when all of following conditions are satisfied.

- DMAi_Bn:SR = 1.
- The stop status (DMAi_Bn:SS) is either initial or normal.
- DIRQ is cleared.
- EDIRQ is cleared.
- In Block transfer mode, the previous transfer has been completed.

When this bit is set to "1", DMA transfer is requested because software request has been received. The DMAC sets this bit to "0" if the Software Trigger has been recognized, an internal channel request for service is set, and Software Trigger Ready (DMAi_Bn:SR) is set to "0". The software request is successful when DMAi_Bn:SR changes its status from "1" to "0". ST can only be read as "0".

Bit	Description
0	No software request
1	Software request

[bit28:27] IS[1:0] : Input select

These bits are used to select the trigger source of a DMA transfer request. When the trigger source of a DMA transfer is a software request, IS bits are set to '00'. When the trigger source of a DMA transfer is a hardware request, IS bits are set to '01'.

Bits	Description
00	Software request
01	Hardware request
10	Reserved
11	Reserved

[bit26] AL : Alternate

This bit decides whether the data transfers should alternate between read and write or should be contiguous reads followed by contiguous writes. The alternation takes place after each incremental read burst and after each single data read.

Bit	Description
0	Contiguous
1	Alternate

[bit25:24] BL[1:0] : Beat limit

Beat Limit controls the maximum burst length the AHB master can make on the AHB bus for this DMA channel.

Bits	Description
00	Single transfer (SINGLE)
01	4-beat incrementing burst (INCR4)
10	8-beat incrementing burst (INCR8)
11	16-beat incrementing burst (INCR16)

These bits are used in combination with Block Count to decide which type of burst has to be transmitted over the AHB interface.

[bit23:20] BC[3:0] : Block count

These bits specify the total length of a single block in block/burst transfer mode. The maximum block count is 16. For e.g. if BC = 4, the number of data transfers is 5 (BC + 1).

Alternate, Block Count, and Beat Limit together decide the bursts generated by the AHB master.

[bit19:16] TO[3:0] : Timeout

The bits are reserved in this product. Write 0b1111 to the bits.

[bit15:0] TC[15:0] : Transfer count

These bits are used to specify the transfer count for the block/burst transfer. The maximum transfer count is 65536. If TC is set to zero then one transfer is done and if TC is set to 65535 then 65536 transfers are done.

In burst and block transfer mode the TC represents the number of blocks of data transfers that the channel has to make before DMA 'End of Processing' (DEOP) is generated. For e.g. if TC = 9 and DMAi_An:BC = 9 then total number of transfers the DMAC does = (DMAi_An:BC + 1) x (TC + 1) = 10 x 10 = 100 transfers before DEOP is generated.

4.8. DMA Controller Channel Configuration B Register Channel 'n' (DMAi_Bn)

This register contains Error and Completion Interrupt Flags, and Interrupt Mask bits. Stop status of DMA operation is available in this register. Operation mode selection, transfer width, source, and destination protection information are located here. Software trigger ready flag shows readiness of DMA channel to receive a Software Trigger. Priority Number for DMA channel arbiter can be set here for fixed and dynamic priority arbitration scheme.

bit	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24
Field	DQ	EQ	MS[1]	MS[0]	TW[1]	TW[0]	SR	Reserved
Attribute	R,WX	R,WX	R0,W	R/W	R/W	R/W	R,WX	R0,WX
Protection	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	bit 23	bit 22	bit 21	bit 20	bit 19	bit 18	bit 17	bit 16
Field	Reserved	Reserved	Reserved	EI	CI	SS[2]	SS[1]	SS[0]
Attribute	R0,WX	R0,WX	R0,WX	R/W	R/W	R,WX	R,WX	R,WX
Protection	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
Field	SP[3]	SP[2]	SP[1]	SP[0]	DP[3]	DP[2]	DP[1]	DP[0]
Attribute	R/W	R/W	R/W	R1,WX	R/W	R/W	R/W	R1,WX
Protection	WP							
Initial Value	0	0	1	1	0	0	1	1

bit	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Field	Reserved	PN[6]	PN[5]	PN[4]	PN[3]	PN[2]	PN[1]	PN[0]
Attribute	R0,WX	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection	WP							
Initial Value	0	1	1	1	1	1	1	1

[bit31] DQ : Flag of DIRQ

DQ is set to "1" when a DMA transfer completed successfully. DQ is cleared by hardware if DMAi_Cn:CD (Clear DIRQ) bit is set to "1". Otherwise the value is retained.

[bit30] EQ : Flag of EDIRQ

EQ is set to "1" when a DMA transfer is finished with an error. EQ is cleared by hardware if DMAi_Cn:CE (Clear EDIRQ) bit is set to "1". Otherwise the value is retained.

[bit29:28] MS[1:0] : Mode select

MS sets the transfer mode of the channel.

Bits	Description
00	Block transfer mode
01	Burst transfer mode
10	Reserved
11	Reserved

[bit27:26] TW[1:0] : Transfer width

TW specifies the data width for every data transfer of the DMA transfer.

Bits	Description
00	Byte
01	Half word
10	Word
11	Double word

[bit25] SR : Software trigger ready

This bit is used to indicate a condition* that the DMA channel is ready to receive a software request. The following conditions can cause that the DMA channel is not ready to receive a software request. If one or more of the conditions below are true SR is set to "0" by hardware. If none of the conditions are true SR is set to "1" by hardware.

- DMAi_R:DE == 0; DMAC is disabled
- DMAi_R:DH == 1; All DMA channels are halted
- DMAi_An:EB == 0; DMA channel is disabled
- DMAi_An:PB == 1; DMA channel is halted
- DMAi_An:IS != 0b00; Input select is not set to software
- DEBUG == 1 and DMAi_R:DE == 1 and DMAi_R:DB == 0b01 or DMAi_R:DB == 0b10, debug event is pending while debug flag is set and Debug Behavior is set to HALT or STOP
- DMA transfer is active and a software request is pending

*: Refer the explanation of DMAi_An:ST bit in DMA Controller Channel Configuration A Register Channel 'n' (DMAi_An) for the rest of the conditions.

[bit20] EI : Error interrupt enable

This bit is used to control the issue of an error interrupt (EDIRQ). If this bit is set to "1", an error interrupt is issued due to any of the following transfer errors:

- Transfer stop request by signal DSTP, or disable the transfer with DMAi_An:EB or DMAi_R:DE, or debug event (if DMAi_R:DBE == 1 and DMAi_R:DB[1:0] == 0b10)
- Source access error
- Destination access error

Bit	Description
0	Error interrupt issuance is disabled
1	Error interrupt issuance is enabled

[bit19] CI : Completion interrupt enable

This bit is used to control the issue of a completion interrupt (DIRQ). If this bit is set to "1", a completion interrupt is issued after the DMA transfer completed normally.

Bit	Description
0	Completion interrupt issuance is disabled
1	Completion interrupt issuance is enabled

[bit18:16] SS[2:0] : Stop status

These bits are used to show the end code of DMA transfer. SS is set by hardware when an error or completion interrupt is raised and it is cleared by hardware when either DMAi_Cn:CE or DMAi_Cn:CD is set to "1".

Bits	Description
000	Initial value. Status : None
001	Reserved
010	Stop request by: – DSTP – Channel disable (DMAi_An:EB= 0) – DMA disable (DMAi_R:DE= 0) – Debug event (DMAi_R:DBE= 1 and DMAi_R:DB= 0b10) Status : Stop
011	Source access error Status : Error
100	Destination access error Status : Error
101	Normal end Status : End
110	Reserved
111	Reserved

When different events occur at the same time, the end code is displayed according to the following priority:

1. Reset
2. Cleared by clearing completion/error interrupt (DIRQ/EDIRQ)
3. Source access error
4. Destination access error
5. Stop request

Note:

– *If the interrupt bit is cleared then Stop Status is also cleared.*

[bit15:12] SP[3:0] : Source protection

These bits are used to control source access protection. During source accesses HPROTM is driven with SP during the AHB address phase.

- SP[3]: 0: Not cacheable
- SP[3]: 1: Cacheable
- SP[2]: 0: Not bufferable
- SP[2]: 1: Bufferable
- SP[1]: 0: User access
- SP[1]: 1: Privileged access
- SP[0]: 0: Instruction access (DMAC only makes data accesses thus SP[0] is fixed to "1" and writing it to "0" has no effect.)
- SP[0]: 1: Data access

SP is considered by sources which support protection control function.

Note:

- *Configuration of DP[3] has an effect.*
  - *DP[3]=0: Posted write buffer in HPM is used. Due to the posted write behavior, DMA cannot trap the error returned from the destination even if there is an error in the write access (e.g.: The destination register/memory requires Privilege level for writing, but DP[1] is "0").*
  - *DP[3]=0: Posted write buffer in HPM is unused. DMA always traps the error when there is an error in the write access. (DMAi_Bn:EQ)*
- *Configuration of DP[2] has no effect.*

[bit11:8] DP[3:0] : Destination protection

These bits are used to control destination access protection. During destination accesses HPROTM is driven with DP during the AHB address phase.

- DP[3]: 0: Not cacheable
- DP[3]: 1: Cacheable
- DP[2]: 0: Not bufferable
- DP[2]: 1: Bufferable
- DP[1]: 0: User access
- DP[1]: 1: Privileged access
- DP[0]: 0: Instruction access (DMAC only makes data accesses thus DP[0] is fixed to "1" and writing it to "0" has no effect.)
- DP[0]: 1: Data access

DP is only considered by destinations which support protection control function.

Note:

- *Configuration of DP[3] has an effect.*
  - *DP[3]=1: Posted write buffer in HPM is used. Due to the posted write behavior, DMA cannot trap the error returned from the destination even if there is an error in the write access (e.g.: The destination register/memory requires Privilege level for writing, but DP[1] is "0").*
  - *DP[3]=0: Posted write buffer in HPM is unused. DMA always traps the error when there is an error in the write access. (DMAi_Bn:EQ)*
- *Configuration of DP[2] has no effect.*

[bit6:0] PN[6:0] : Priority number

These bits are used to specify the Priority Number. The Priority Number is needed if fixed priority or dynamic priority has been selected as arbitration scheme. It has no significance if round robin arbitration scheme is selected. The channel with lower priority number value has a higher priority.

4.9. DMA Controller Channel Configuration Source Address Register Channel 'n' (DMAi_SAn)

This register holds the source address for the DMA transfer of channel 'n'.

bit	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24
Field	SA[31]	SA[30]	SA[29]	SA[28]	SA[27]	SA[26]	SA[25]	SA[24]
Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W
Protection	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	bit 23	bit 22	bit 21	bit 20	bit 19	bit 18	bit 17	bit 16
Field	SA[23]	SA[22]	SA[21]	SA[20]	SA[19]	SA[18]	SA[17]	SA[16]
Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W
Protection	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
Field	SA[15]	SA[14]	SA[13]	SA[12]	SA[11]	SA[10]	SA[9]	SA[8]
Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W
Protection	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Field	SA[7]	SA[6]	SA[5]	SA[4]	SA[3]	SA[2]	SA[1]	SA[0]
Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W
Protection	WP							
Initial Value	0	0	0	0	0	0	0	0

[bit31:0] SA : Source address

These bits are used to specify the source address to start a DMA transfer. Source Address is updated with the value of DMAi_SASHDWn at the end of a DMA transfer if Update Source Address (DMAi_Dn:US) is set to "1", Fixed Block Source Address (DMAi_Dn:FBS) is set to "0", and the transfer ends successfully. Otherwise it retains its value.

Note:

- *DMAi_SAn register must be loaded with aligned addresses with respect to the transfer width. If non-aligned addresses are loaded the DMAC converts it to an aligned address according to the setting of Transfer Width (DMAi_Bn:TW).*

Note:

- *Disable the channel (i.e. DMAi_R:DE=0 or DMAi_An:EB = 0) before configuring this register.*

4.10. DMA Controller Channel Configuration Destination Address Register Channel 'n' (DMAi_DAn)

This register holds the destination address for the DMA transfer of channel 'n'.

bit	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24
Field	DA[31]	DA[30]	DA[29]	DA[28]	DA[27]	DA[26]	DA[25]	DA[24]
Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W
Protection	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	bit 23	bit 22	bit 21	bit 20	bit 19	bit 18	bit 17	bit 16
Field	DA[23]	DA[22]	DA[21]	DA[20]	DA[19]	DA[18]	DA[17]	DA[16]
Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W
Protection	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
Field	DA[15]	DA[14]	DA[13]	DA[12]	DA[11]	DA[10]	DA[9]	DA[8]
Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W
Protection	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Field	DA[7]	DA[6]	DA[5]	DA[4]	DA[3]	DA[2]	DA[1]	DA[0]
Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W
Protection	WP							
Initial Value	0	0	0	0	0	0	0	0

[bit31:0] DA : Destination address

These bits are used to specify the destination address to start a DMA transfer. Destination Address is updated with the value of DMAi_DASHDWn at the end of a DMA transfer if Update Destination Address (DMAi_Dn:UD) is set to "1", Fixed Block Destination Address (DMAi_Dn:FBD) is set to "0", and the transfer ends successfully. Otherwise it retains its value.

Note:

- *DMAi_DAn register must be loaded with aligned addresses with respect to the transfer width. If non-aligned addresses are loaded the DMAC converts it to an aligned address according to the setting of Transfer Width (DMAi_Bn:TW).*

Note:

- *Disable the channel (i.e. DMAi_R:DE=0 or DMAi_An:EB = 0) before configuring this register.*

4.11. DMA Controller Channel Configuration C Register Channel 'n' (DMAi_Cn)

The DMA Controller Channel Configuration C Register is used to clear the Interrupt Flags set in the DMAi_Bn register. By writing a "1" to a bit in this register, the software can clear the corresponding flag in the DMAi_Bn register.

bit	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
Protection	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	bit 23	bit 22	bit 21	bit 20	bit 19	bit 18	bit 17	bit 16
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
Protection	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	CE
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,W
Protection	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	CD
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,W
Protection	WP							
Initial Value	0	0	0	0	0	0	0	0

[bit8] CE : Clear EDIRQ

Setting this bit clears the EDIRQ flag bit (DMAi_Bn:EQ). DMAC clears this bit automatically.

[bit0] CD : Clear DIRQ

Setting this bit clears the DIRQ flag bit (DMAi_Bn:DQ). DMAC clears this bit automatically.

4.12. DMA Controller Channel Configuration D Register Channel 'n' (DMAi_Dn)

This register controls the DMA channels addressing behavior for a DMA transfer. Addressing can be independently chosen for source and destination.

bit	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24
Field	FS	DES	US	FBS	Reserved	Reserved	Reserved	Reserved
Attribute	R/W	R/W	R/W	R/W	R0,WX	R0,WX	R0,WX	R0,WX
Protection	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	bit 23	bit 22	bit 21	bit 20	bit 19	bit 18	bit 17	bit 16
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
Protection	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
Field	FD	DED	UD	FBD	Reserved	Reserved	Reserved	Reserved
Attribute	R/W	R/W	R/W	R/W	R0,WX	R0,WX	R0,WX	R0,WX
Protection	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
Protection	WP							
Initial Value	0	0	0	0	0	0	0	0

Note:

- *This register must be accessed by 8-bit access to the effective bits.*

[bit31] FS : Fixed Source address

This bit is used to keep the source address at a fix value.

Bit	Description
0	Source address is incremented
1	Source address is kept fix. AHB master only makes single transfers to access the source

[bit30] DES : Decrement source address

If this bit is set the source address on the AHB interface is decremented on each AHB transfer. In this mode the AHB master makes only single transfers to access the source.

Bit	Description
0	Source address is incremented
1	Source address is decremented

Note:

- *Fixed Source Address (DMAi_Dn:FS) has higher priority than Decrement Source Address. This bit is only effective if DMAi_Dn:FS = 0 and DMAi_Dn:FBS = 0.*

[bit29] US : Update source address

Bit	Description
0	DMAi_SAn is not updated after DMA transfer is successfully completed. DMAi_SAn is retained
1	DMAi_SAn is updated with the next address after the DMA transfer is successfully completed. E.g. last source address is 0x0BF000FC

Note:

- *This bit is only effective if DMAi_Dn:FBS = 0*

[bit28] FBS : Fixed block source address

Bit	Description
0	Start address of first block of DMA transfer is set to the value stored in DMAi_SAn. Start address of consecutive blocks is the address following the previous block last address (according to Transfer Width (DMAi_Bn:TW))
1	Start address of each block is set to the value stored in DMAi_SAn

Note:

- *Setting of FBS is only effective if DMAi_Dn:FS = 0 and (DMAi_Bn:MS = 0b00 (block transfer mode) or DMAi_Bn:MS = 0b01 (burst transfer mode)).*

[bit15] FD : Fixed destination address

This bit is used to keep the destination address at a fix value.

Bit	Description
0	Destination address is incremented
1	Destination address is kept fixed. AHB master only makes single transfers to access the destination

[bit14] DED : Decrement destination address

If this bit is set the destination address on the AHB interface is decremented on each AHB transfer. In this mode the AHB master makes only single transfers to access the destination.

Bit	Description
0	Destination address is incremented
1	Destination address is decremented

Note:

- *Fixed Destination Address (DMAi_Dn:FD) has higher priority than Decrement Destination Address. This bit is only effective if DMAi_Dn:FD = 0 and DMAi_Dn:FBD = 0.*

[bit13] UD : Update destination address

Bit	Description
0	DMAi_DAn is not updated after DMA transfer is successfully completed. DMAi_DAn is retained
1	DMAi_DAn is updated with the next address after the DMA transfer is successfully completed. E.g. last destination address was 0x0BF40010, DMAi_Bn:TW = 10 (word) then DMAi_DAn is updated with address 0x0BF40014.

Note:

- *This bit is effective only if DMAi_Dn:FBD = 0*

[bit12] FBD : Fixed block destination address

Bit	Description
0	Start address of first block of DMA transfer is set to the value stored in DMAi_DAn. Start address of consecutive blocks is the address following the previous block last address (according to Transfer Width (DMAi_Bn:TW)
1	Start address of each block is set to the value stored in DMAi_DAn

Note:

- *Setting of FBD is only effective if FD = 0 and (DMAi_Bn:MS = 0b00 (block transfer mode) or DMAi_Bn:MS = 0b01 (burst transfer mode)).*

4.13. DMA Controller Channel Configuration E Register Channel 'n' (DMAi_En)

This register is prepared for reserved features. Do not use this register.

bit	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24
Field	EE	DC[14]	DC[13]	DC[12]	DC[11]	DC[10]	DC[9]	DC[8]
Attribute	R/W0	R/W0	R/W0	R/W0	R/W0	R/W0	R/W0	R/W0
Protection	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	bit 23	bit 22	bit 21	bit 20	bit 19	bit 18	bit 17	bit 16
Field	DC[7]	DC[6]	DC[5]	DC[4]	DC[3]	DC[2]	DC[1]	DC[0]
Attribute	R/W0	R/W0	R/W0	R/W0	R/W0	R/W0	R/W0	R/W0
Protection	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
Field	IC[15]	IC[14]	IC[13]	IC[12]	IC[11]	IC[10]	IC[9]	IC[8]
Attribute	R/W0	R/W0	R/W0	R/W0	R/W0	R/W0	R/W0	R/W0
Protection	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Field	IC[7]	IC[6]	IC[5]	IC[4]	IC[3]	IC[2]	IC[1]	IC[0]
Attribute	R/W0	R/W0	R/W0	R/W0	R/W0	R/W0	R/W0	R/W0
Protection	WP							
Initial Value	0	0	0	0	0	0	0	0

[bit31] EE : Reserved
Always write "0" to the bit when writing.

[bit30:16] DC[14:0] : Reserved
Always write "0" to the bits when writing.

[bit15:0] IC[15:0] : Reserved
Always write "0" to the bits when writing.

4.14. DMA Controller Channel Configuration Source Address Shadow Register Channel 'n' (DMAi_SASHDWn)

This register holds the address of the last source data transfer for the DMA transfer of channel 'n' in case of a source access error. If the DMA transfer is stopped it contains the next address following the last successful data transfer.

bit	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24
Field	SASHDW[31]	SASHDW[30]	SASHDW[29]	SASHDW[28]	SASHDW[27]	SASHDW[26]	SASHDW[25]	SASHDW[24]
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	bit 23	bit 22	bit 21	bit 20	bit 19	bit 18	bit 17	bit 16
Field	SASHDW[23]	SASHDW[22]	SASHDW[21]	SASHDW[20]	SASHDW[19]	SASHDW[18]	SASHDW[17]	SASHDW[16]
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
Field	SASHDW[15]	SASHDW[14]	SASHDW[13]	SASHDW[12]	SASHDW[11]	SASHDW[10]	SASHDW[9]	SASHDW[8]
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Field	SASHDW[7]	SASHDW[6]	SASHDW[5]	SASHDW[4]	SASHDW[3]	SASHDW[2]	SASHDW[1]	SASHDW[0]
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection	WP							
Initial Value	0	0	0	0	0	0	0	0

[bit31:0] SASHDW[31:0] : Source address shadow

At the start of a DMA transfer this register contains a copy of DMAi_SAn. During a DMA transfer it is either incremented, decremented, or stays constant according to the settings of DMAi_Dn:FS, DMAi_Dn:DES, DMAi_Dn:US, and DMAi_Dn:FBS. In case of a source access error SASHDW shows the address of the read access which causes the error. In case of a stop of the DMA transfer it shows the next address following the last read access done.

Note:

- *SASHDW does not show an aligned address if a non-aligned address is loaded into DMAi_SAn initially. However the DMAC will have made the read access to address SASHDW aligned according to the setting of Transfer Width (DMAi_Bn:TW).*

4.15. DMA Controller Channel Configuration Destination Address Shadow Register Channel 'n' (DMAi_DASHDWn)

This register holds the address of the last destination data transfer for the DMA transfer of channel 'n' in case of a destination access error. If the DMA transfer is stopped it contains the next address following the last successful data transfer.

bit	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24
Field	DASHDW[31]	DASHDW[30]	DASHDW[29]	DASHDW[28]	DASHDW[27]	DASHDW[26]	DASHDW[25]	DASHDW[24]
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	bit 23	bit 22	bit 21	bit 20	bit 19	bit 18	bit 17	bit 16
Field	DASHDW[23]	DASHDW[22]	DASHDW[21]	DASHDW[20]	DASHDW[19]	DASHDW[18]	DASHDW[17]	DASHDW[16]
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
Field	DASHDW[15]	DASHDW[14]	DASHDW[13]	DASHDW[12]	DASHDW[11]	DASHDW[10]	DASHDW[9]	DASHDW[8]
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Field	DASHDW[7]	DASHDW[6]	DASHDW[5]	DASHDW[4]	DASHDW[3]	DASHDW[2]	DASHDW[1]	DASHDW[0]
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection	WP							
Initial Value	0	0	0	0	0	0	0	0

[bit31:0] DASHDW[31:0] : Destination address shadow

At the start of a DMA transfer this register contains a copy of DMAi_DAn. During a DMA transfer it is either incremented, decremented, or stays constant according to the settings of DMAi_Dn:FD, DMAi_Dn:DED, DMAi_Dn:UD, and DMAi_Dn:FBD. In case of a destination access error DASHDW shows the address of the write access which causes the error. In case of a stop of the DMA transfer it shows the next address following the last write access done.

Note:

- *DASHDW does not show an aligned address if a non-aligned address is loaded into DMAi_DAn initially. However the DMAC will have made the write access to address DASHDW aligned according to the setting of Transfer Width (DMAi_Bn:TW).*

4.16. DMA Controller Client Matrix Internal Client Interface Configuration Register 'm' (DMAi_CMICICm)

This register controls internal client interface 'm'. Behavioral config bits determine the behavior of the acknowledge signals in case the DMA client is not selected. Also that this register controls enables DMA request for some peripherals.

bit	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24
Field	Reserved	Reserved	Reserved	Reserved	BEHSTPACK	Reserved	BEHREQACK	Reserved
Attribute	R0,WX	R0,WX	R0,WX	R/W0	R/W	R0,WX	R/W	R0,WX
Protection	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	bit 23	bit 22	bit 21	bit 20	bit 19	bit 18	bit 17	bit 16
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
Protection	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
Protection	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
Protection	WP							
Initial Value	0	0	0	0	0	0	0	0

Notes:

- *Attribute of the 27th bit (BEHSTPACK) of DMAi_CMICICm has "R/W" and "R0,WX" by difference of product specifications.*
- *DMAi_CMICICm registers are not implemeted depending on the product specification. In case of register access to those DMAi_CMICICm registers, response of bus error will occur*
- *About the implementation of these registers, please see the product's HWM.*

[bit28] Reserved

Always write "0" to this bit.

[bit27] BEHSTPACK : Behavior stop acknowledge

BEHSTPACK sets the behavior of the Internal DMA Client Interface 'm' output signal DSTP_ACK[m] if the client interface is not selected in any of the Channel Configuration Registers (DMAi_CMCHICn).

Bit	Description
0	DSTP_ACK[m] outputs inactive logic level
1	DSTP[m] connects directly to DSTP_ACK[m]

[bit25] BEHREQACK : Behavior request acknowledge

BEHREQACK sets the behavior of the Internal DMA Client Interface 'm' output signal DREQ_ACK[m] if the client interface is not selected in any of the Channel Configuration Registers (DMAi_CMCHICn).

Bit	Description (1)
0	DREQ_ACK[m] outputs inactive logic level.
1	DREQ[m] connects directly to DREQ_ACK[m].

This bit also works as DMA request enable signal to the corresponding peripheral which send DMA request by IRQ. The bit must be changed while the peripheral function is disabled to guarantee that its IRQ is kept to disabled.

Bit	Description (2)
0	For the corresponding peripheral whose DMA request is controlled by IRQ: DMA request by IRQ is disabled.
1	For the corresponding peripheral whose DMA request is controlled by IRQ: DMA request by IRQ is enabled.

4.17. DMA Controller Client Matrix Channel Interface Configuration Register 'n' (DMAi_CMCHICn)

This register controls internal channel interface 'n'. Behavioral config bits determine the behavior of the acknowledge signals in case the DMA client is not selected. Also the register contains bit fields which configure the behavior of CMCHIC reload event.

bit	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24
Field	Reserved	Reserved	RLESEL[5]	RLESEL[4]	RLESEL[3]	RLESEL[2]	RLESEL[1]	RLESEL[0]
Attribute	R0,WX	R0,WX	R,W	R,W	R,W	R,W	R,W	R,W
Protection	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	bit 23	bit 22	bit 21	bit 20	bit 19	bit 18	bit 17	bit 16
Field	Reserved	Reserved	Reserved	Reserved	Reserved	RLSLOT[2]	RLSLOT[1]	RLSLOT[0]
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R,W	R,W	R,W
Protection	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	CI[8]
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R,W
Protection	WP							
Initial Value	0	0	0	0	0	0	0	0

bit	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Field	CI[7]	CI[6]	CI[5]	CI[4]	CI[3]	CI[2]	CI[1]	CI[0]
Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W
Protection	WP							
Initial Value	0	0	0	0	*1	*1	*1	*1

*1: Initial value of DMAi_CMCHICn:CI[8:0] is 'n'.

(e.g.) Initial value of DMAi_CMCHIC0:CI[8:0]= 0b000000000
Initial value of DMAi_CMCHIC5:CI[8:0]= 0b000000101

[bit29:24] RLESEL[5:0] : Reload event selection replacement

The bits specified the CMCHIC reload event. Please refer "CHAPTER: DMA COMPLEX SUBSYSTEM" for more details.

Bits	Description
000000	CMCHIC Reload event of DMAi_CMCHICn is disabled.
000100	Negative edge of TOT of Reload Timer ch.0 in the corresponding DMA Additional Control
000101	Negative edge of TOT of Reload Timer ch.1 in the corresponding DMA Additional Control
000110	Negative edge of TOT of Reload Timer ch.2 in the corresponding DMA Additional Control
000111	Negative edge of TOT of Reload Timer ch.3 in the corresponding DMA Additional Control
001000 to 001111	CMCHIC Reload event of DMAi_CMCHICn is disabled.
010000	BSTART from ch.0
010001	BSTART from ch.1
.....	.....
011111	BSTART from ch.15
100000	BDONE from ch.0
100001	BDONE from ch.1
.....	.....
101111	BDONE from ch.15
Else	No signal is selected

[bit18:16] RLSLOT[2:0] : Reload slot replacement

The bit specifies which bank is copied DMAi_CMCHICn at CMCHIC reload event. Please refer "CHAPTER: DMA COMPLEX SUBSYSTEM" for more details.

Bits	Description
000	Value of DMAAn_CMCHICRDB0 register is copied to DMAi_CMCHICn at CMCHIC reload event.
001	Value of DMAAn_CMCHICRDB1 register is copied to DMAi_CMCHICn at CMCHIC reload event.
.....	.....
111	Value of DMAAn_CMCHICRDB8 register is copied to DMAi_CMCHICn at CMCHIC reload event.

[bit8:0] CI[8:0] : Client interface

Client Interface specifies that DMA channel 'n' is routed to the DMA Client given by the binary value of CI.

5. Additional Information

This section tabulates additional information of DMAC.

Following items are additional information of the DMAC in this product.

- "Debug event" is asserted by any of following condition :
  - CPU is in debug state

Notes:

When the channel is used with Multi-function Serial Interface to write the transmission data

or read the reception data.

- When corresponding TXBLKEN (at transmission) or RXBLKEN (at reception) is 0:

Configure DMAi_An:BC[3:0] to "0000".

- When corresponding TXBLKEN (at transmission) or RXBLKEN (at reception) is 1:

Configure DMAi_An:BC[3:0] to a suitable value regarding TBSIZE (at transmission) or FBYTE (at reception).

CHAPTER 28: DMA COMPLEX SUBSYSTEM

This chapter explains DMA COMPLEX SUBSYSTEM.

1. Overview
2. Configuration and Block Diagram
3. Operation of the DMA COMPLEX SUBSYSTEM
4. Registers
5. Overview in DMA COMPLEX SUBSYSTEM
6. Configuration and Block Diagram in DMA COMPLEX SUBSYSTEM
7. Operation of the 32-Bit Reload Timer in DMA COMPLEX SUBSYSTEM
8. Registers in DMA COMPLEX SUBSYSTEM

DMACOMP-TXXPT03P01R01L08-E1-XX

1. Overview

This section gives a brief overview of DMA COMPLEX SUBSYSTEM.

The DMA COMPLEX SUBSYSTEM consists of DMA Controller, DMA Additional Control block and MPU16 AHB which operate in combination.

The DMA COMPLEX SUBSYSTEM offers following features:

- DMA transfer by DMA Controller. Please refer to "CHAPTER: DMA Controller" for more detail.
- Protect for unauthorized bus access by DMA Controller. Please refer to "CHAPTER: MPU16 AHB" for more detail.
- More debug assist features by DMA Additional Control block.
- Complicated DMA transfer request using dedicated Reload Timer inside DMA Complex Subsystem.

2. Configuration and Block Diagram

This section shows a block diagram of DMA COMPLEX SUBSYSTEM and DMA Additional Control block.

Figure 2-1 Entire Block Diagram of DMA Complex Subsystem

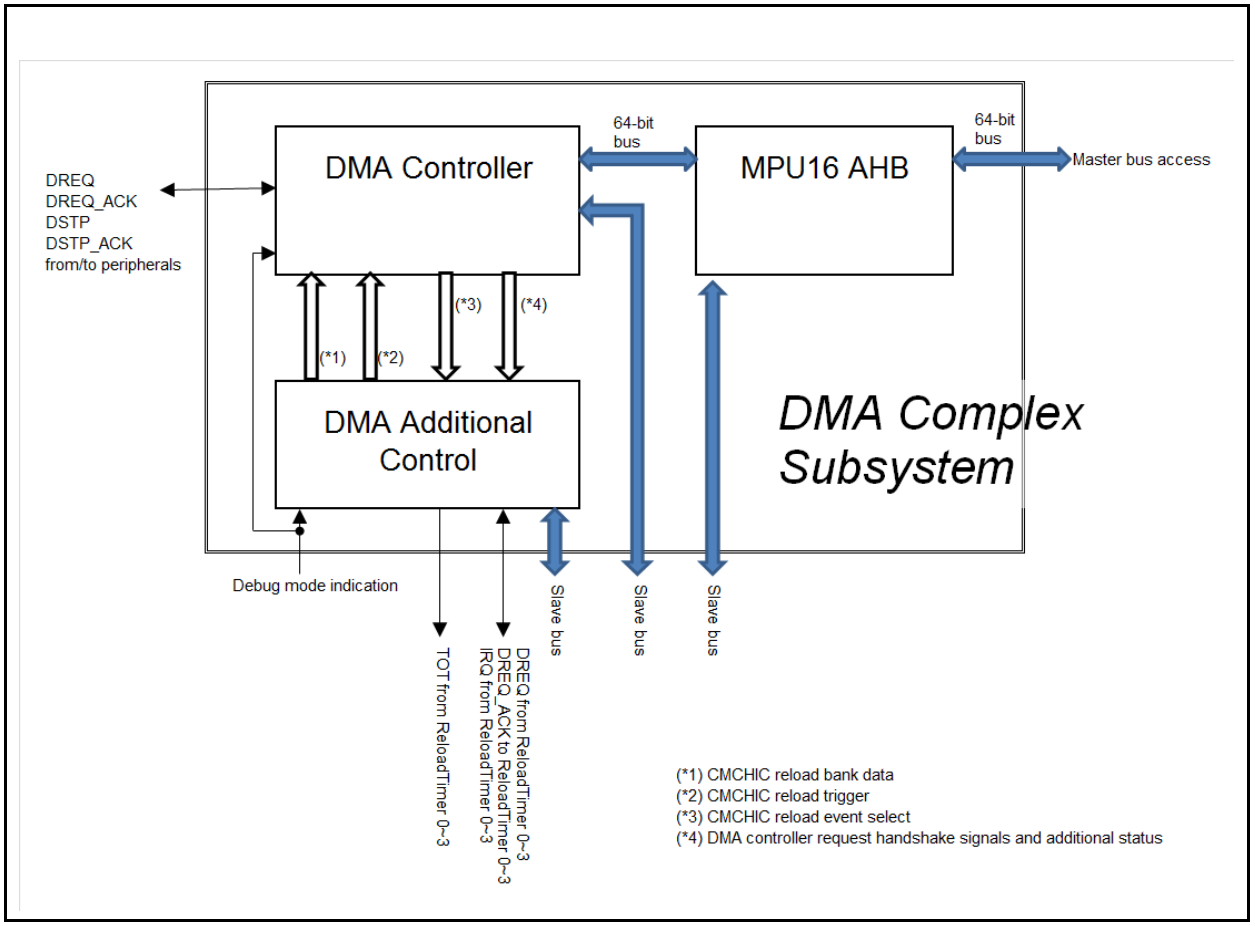
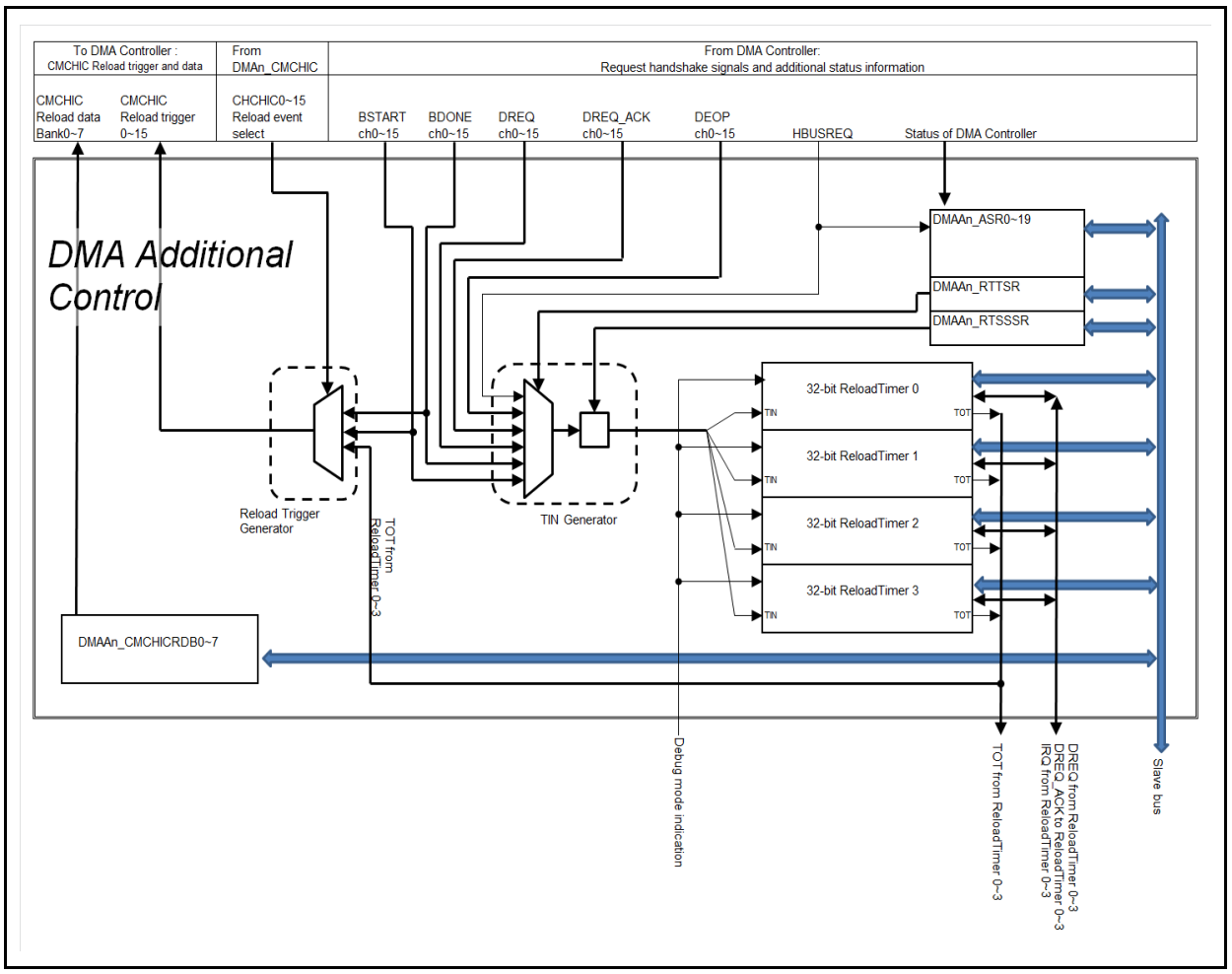


Figure 2-2 Entire Block Diagram of DMA Additional Control Block



3. Operation of the DMA COMPLEX SUBSYSTEM

This section describes the operation of DMA COMPLEX SUBSYSTEM.

The DMA COMPLEX SUBSYSTEM consists of following blocks:

- DMA Controller
- MPU16 AHB which operate in combination.
- DMA Additional Control block

The DMA Controller gives high-performance DMA transfer. Please refer to "CHAPTER: DMA Controller" for the details.

The MPU16 AHB is a memory protection unit which is inserted to a master bus and it offers protection features against unauthorized access by the DMA transfer. To cover all 16 channels for preventing unauthorized writing, the MPU16 AHB has 16 + 1 regions. Please refer to "CHAPTER: MPU16 AHB" for the details.

DMA Additional Control block offers more function for generating complicated DMA requests and obtaining DMA Controller status using its dedicated Reload Timer and additional status registers. DMA Additional Control block operation is explained in this section.

3.1. Additional Status Registers

This section is an introduction for additional status registers inside DMA Additional Control block.

DMA Additional Control block has several read-only status registers which offers several information of current DMA Controller status. Refer to "4 Registers" for more details of the registers.

3.2. CMCHIC Reloading

This section gives the explanation for CMCHIC reloading.

DMAn_CMCHICi registers have CMCHIC reload feature. When DMAn_CMCHICi register receives CMCHIC reload trigger, the contents of DMAn_CMCHICi registers value is replaced to the specified DMAAn_CMCHICRDBj register value. Where 'j' is specified by DMAn_CMCHICi:RLSLOT[2:0].

CMCHIC reload trigger is generated in Reload Trigger Generator block in DMA Additional Control block when specified CMCHIC reload event happens. Where, the CMCHIC reload event is specified by DMAn_CMCHICi:RLESEL[5:0] bits.

By CMCHIC reloading, client peripherals can be reconfigured as following figures.

Figure 3-1 Example of CMCHIC Reload Usage: Switching Client Peripheral

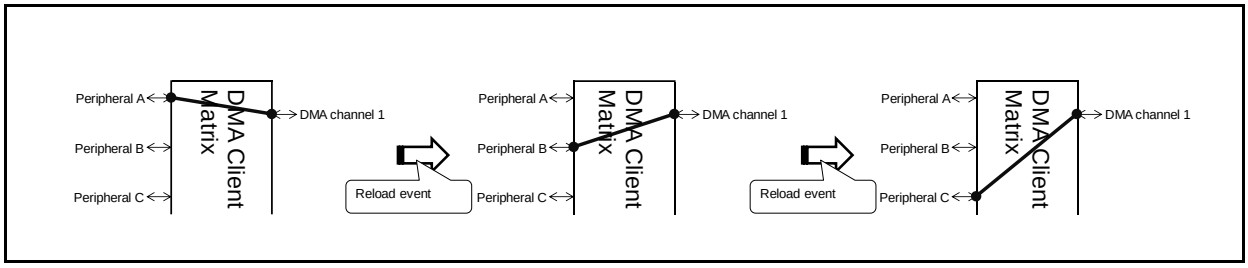
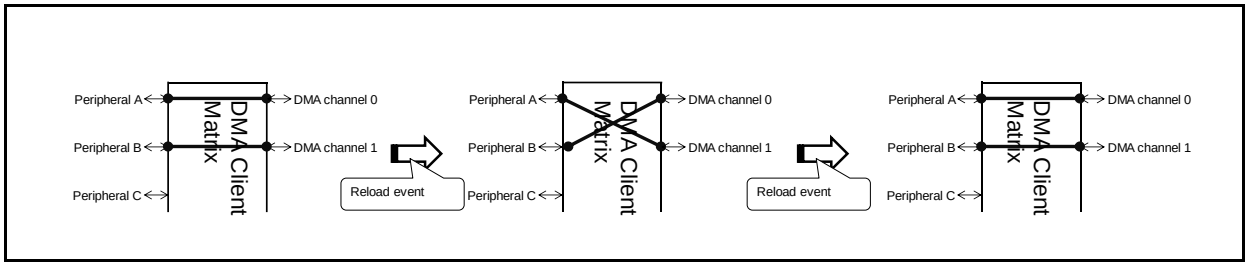


Figure 3-2 Example of CMCHIC Reload Usage: Swapping Client and Channel Connection



The CMCHIC reload event must not happen in middle of DREQ and DREQ_ACK handshaking. So the DMA trigger condition (e.g.: FIFO alarm value to trigger DMA request) must be carefully configured to make sure that any reload event does not happen in middle of DREQ and DREQ_ACK handshaking.

3.3. Reload Timer in DMA Additional Control

This section gives the explanation for Reload Timer in DMA Additional Control block.

The DMA Additional Control block has four 32-bit Reload Timer instances. The Reload Timer instances are used to following applications:

- Capturing events from DMA Controller
- Counting events from DMA Controller
- Generating another DMA request
- Generating IRQ
- Output pulse signal to TOT

Please refer to 3.3.1 for the connection of TIN input of Reload Timer instance is connected.

Please refer to 3.3.2 for the usage examples of Reload Timer.

Note:

- *TIN inputs of the Reload Timer in DMA Additional Controls are connected to DMA Complex Subsystem internal signals therefore they cannot be used to capture external pin inputs. Please refer to 3.3.1 Reload Timer Trigger Inputs and 4.7 DMA Additional Control Reload Timer Trigger Select Register (DMAAn_RTTSR) for TIN connection.*

3.3.1. Reload Timer Trigger Inputs

In this section, TIN sources of Reload Timer inside DMA Additional Control blocks are tabulated.

DMA Controller outputs some of DMA request handshakes and some of DMA status signals to be referred in DMA Additional Control block. In DMA Additional Control block, they are connected to the TIN input of Reload Timer via TIN generator block. The TIN source is selected by DMAAn_RTTSR register.

Table 3-1 Reload Timer TIN Input Selection

Source	Description
DREQ[i]	DMA request to DMA channel i from corresponding peripheral.
DREQ_ACK[i]	Acknowledge to the DMA request from DMA channel i.
DEOP[i]	DMA End of Operation. When the all blocks of block transfer or a burst transfer of DMA channel i is completed, the DEOP signal is asserted. It will be de-asserted soon.
BSTART[i]	Block transfer start. It will asserted at start of a transfer of a block by DMA channel i.
BDONE[i]	Block transfer done. It is asserted at completion of the current block of DMA channel i.
HBUSREQ	HBUSREQ is asserted to "1" at transfer starts. HBUSREQ is de-asserted to "0" at transfer completion of current block. By referring to HBUSREQ, "transfer gap" can be observed. Please refer to "CHAPTER : DMA Controller" for the "transfer gap".
Software trigger	When bit 'j' of DMAAn_RTSSSR is written, a "0"→"1"→"0" pulse will be given to TIN of ReloadTimer channel j. Using the DMAAn_RTSSSR, all of four Reload Timer can be triggered at the same time. Software trigger is always able to be used regardless DMAAn_RTTSR setting.

'i' indicates channel number of DMA.

'j' indicates channel number of Reload Timer in the DMA Additional Control block.

3.3.2. Reload Timer Usage Examples

In this section, usage examples of Reload Timer are given.

Table 3-2 Reload Timer Usage Examples

Usage	Description
Periodic timer interrupt generator	It can be used as periodic timer interrupt source.
Periodic timer DMA trigger	It can be uses as periodic timer DMA trigger.
Route BSTART / BDONE / DREQ / DREQ_ACK / DEOP / HBUSREQ to TOT output.	Assertion or de-assertion of these signals can be observed by TOT by triggering the timer and output TOT by the timer event.
Count the number of BSTART / BDONE / DREQ / DREQ_ACK / DEOP / HBUSREQ events.	By using these signals as counting clock of the Reload Timer, number of these signal events can be counted.
Obtain timestamp of BSTART / BDONE / DREQ / DREQ_ACK / DEOP / HBUSREQ event.	By using these signals as the start trigger of timer, the timestamp of these signal events can be calculated.
Trigger 2nd DMA channel by request / completion of 1st DMA channel transfer.	By following configuration, another DMA trigger can be generated by a DMA transfer request / completion. − TIN input : DREQ_ACK / BDONE / DEOP − ReloadTimer DMAAn_RLTm_DMACFG:ENDMAUF = 1
Trigger a block transfer to 2nd DMA channel per completion of 3 blocks of 1st DMA channel.	By following configuration, transfer requests can be generated at every four block transfer completion. − TIN input : BDONE − Use Reload timer in event counter mode (DMAAn_RLTm_TMCSR:CSL[2:0]= 0b110) − ReloadTimer DMAAn_RLTm_DMACFG:ENDMAUF= 1 − ReloadTimer DMAAn_RLTm_TMRLR= 0x00000002
Generating CMCHIC reload trigger.	TOT of Reload Timer can be used as CMCHIC reload event.

Figure 3-3 Chain Transfer Example

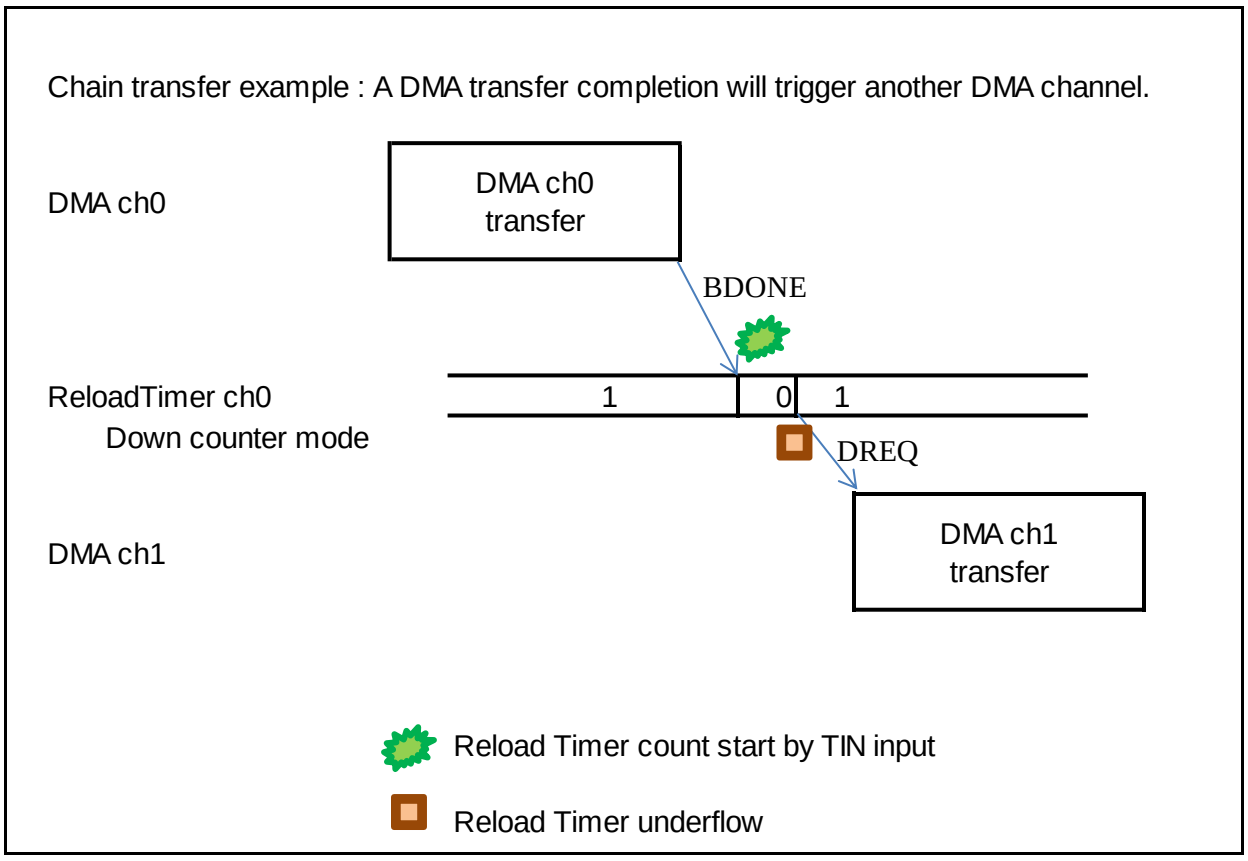


Figure 3-4 Chain Transfer Example

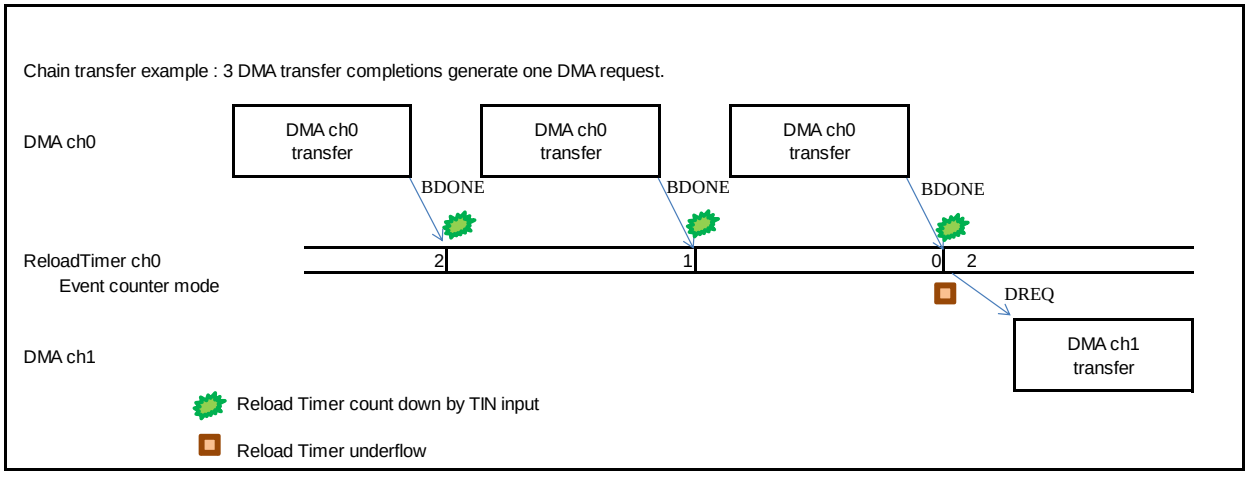
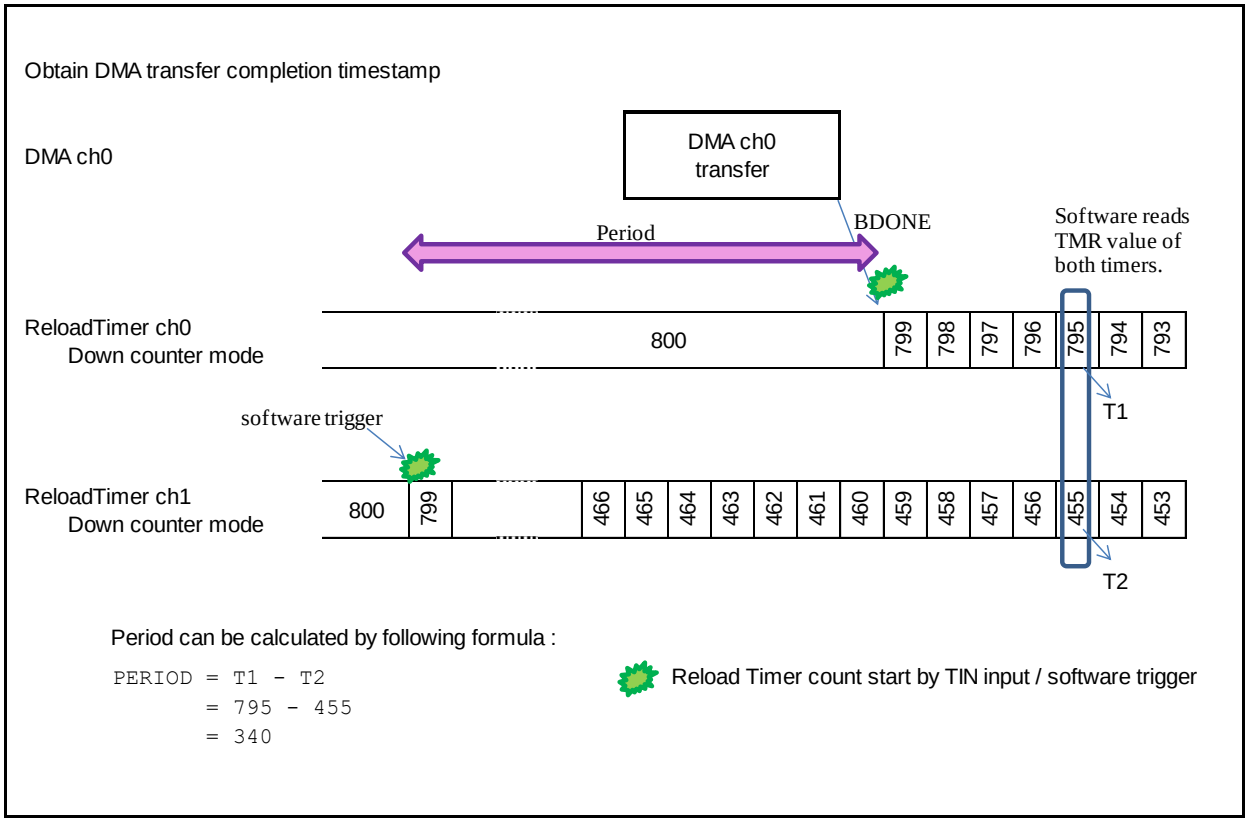


Figure 3-5 Example to Obtain Timestamp



4. Registers

This section describes the registers of DMA Additional Control block.

Please refer to "CHAPTER: DMA Controller" for the registers in DMA Controller.

Please refer to "CHAPTER: MPU16 AHB" for the registers in MPU16 AHB.

The DMA Additional Control block has four "32-bit Reload Timer" instances. Please refer to 8 Registers in DMA COMPLEX SUBSYSTEM.

Table 4-1 Memory Layout of Registers in DMA Additional Controls

Offset	Register Name/Initial Value			
	+3	+2	+1	+0
0x0000_0000	DMAAn_ASR0 00000000_00000000_00000000_00000000			
0x0000_0004	DMAAn_ASR1 00000000_00000000_00000000_00000000			
0x0000_0008	DMAAn_ASR2 00000000_00000000_00000000_00000000			
0x0000_000C	DMAAn_ASR3 00000000_00000000_00000000_00000000			
0x0000_0010	DMAAn_ASR4 00000000_00000000_00000000_00000000			
0x0000_0014	DMAAn_ASR5 00000000_00000000_00000000_00000000			
0x0000_0018	DMAAn_ASR6 00000000_00000000_00000000_00000000			
0x0000_001C	DMAAn_ASR7 00000000_00000000_00000000_00000000			
0x0000_0020	DMAAn_ASR8 00000000_00000000_00000000_00000000			
0x0000_0024	DMAAn_ASR9 00000000_00000000_00000000_00000000			
0x0000_0028	DMAAn_ASR10 00000000_00000000_00000000_00000000			
0x0000_002C	DMAAn_ASR11 00000000_00000000_00000000_00000000			
0x0000_0030	DMAAn_ASR12 00000000_00000000_00000000_00000000			
0x0000_0034	DMAAn_ASR13 00000000_00000000_00000000_00000000			
0x0000_0038	DMAAn_ASR14 00000000_00000000_00000000_00000000			
0x0000_003C	DMAAn_ASR15 00000000_00000000_00000000_00000000			
0x0000_0040	DMAAn_ASR16 00000000_00000000_00000000_00000000			

Offset	Register Name/Initial Value			
	+3	+2	+1	+0
0x0000_0044	DMAAn_ASR17 00000000_00000000_00000000_00000000			
0x0000_0048	DMAAn_ASR18 00000000_00000000_00000000_00000000			
0x0000_004C	DMAAn_ASR19 00000000_00000000_00000000_00000000			
0x0000_0050 | 0x0000_007C	-			
0x0000_0080	DMAAn_CMCHICRDB0 00000000_00000000_00000000_00000000			
0x0000_0084	DMAAn_CMCHICRDB1 00000000_00000000_00000000_00000000			
0x0000_0088	DMAAn_CMCHICRDB2 00000000_00000000_00000000_00000000			
0x0000_008C	DMAAn_CMCHICRDB3 00000000_00000000_00000000_00000000			
0x0000_0090	DMAAn_CMCHICRDB4 00000000_00000000_00000000_00000000			
0x0000_0094	DMAAn_CMCHICRDB5 00000000_00000000_00000000_00000000			
0x0000_0098	DMAAn_CMCHICRDB6 00000000_00000000_00000000_00000000			
0x0000_009C	DMAAn_CMCHICRDB7 00000000_00000000_00000000_00000000			
0x0000_00A0 | 0x0000_00FF	-			
0x0000_0100	DMAAn_RTTSR 00000000_00000000_00000000_00000000			
0x0000_0104 | 0x0000_011F	-			
0x0000_0120	DMAAn_RTSSSR 00000000_00000000_00000000_00000000			
0x0000_0124 | 0x0000_07FF	-			
0x0000_0800	32-bit ReloadTimer0 DMAAn_RLTm_DMACFG 00000000_00000000_00000000_00000000			
0x0000_0804	-			
0x0000_0808	32-bit ReloadTimer0 DMAAn_RLTm_TMCSR 00000000_00000000_00000000_00000000			
0x0000_080C	-			

Offset	Register Name/Initial Value			
	+3	+2	+1	+0
0x0000_0810	32-bit ReloadTimer0 DMAAn_RLTm_TMRLR XXXXXXXX_XXXXXXXX_XXXXXXXX_XXXXXXXX			
0x0000_0814	32-bit ReloadTimer0 DMAAn_RLTm_TMR XXXXXXXX_XXXXXXXX_XXXXXXXX_XXXXXXXX			
0x0000_0818 | 0x0000_081C	-			
0x0000_0820	32-bit ReloadTimer1 DMAAn_RLTm_DMACFG 00000000_00000000_00000000_00000000			
0x0000_0824	-			
0x0000_0828	32-bit ReloadTimer1 DMAAn_RLTm_TMCSR 00000000_00000000_00000000_00000000			
0x0000_082C	-			
0x0000_0830	32-bit ReloadTimer1 DMAAn_RLTm_TMRLR XXXXXXXX_XXXXXXXX_XXXXXXXX_XXXXXXXX			
0x0000_0834	32-bit ReloadTimer1 DMAAn_RLTm_TMR XXXXXXXX_XXXXXXXX_XXXXXXXX_XXXXXXXX			
0x0000_0838 | 0x0000_083C	-			
0x0000_0840	32-bit ReloadTimer2 DMAAn_RLTm_DMACFG 00000000_00000000_00000000_00000000			
0x0000_0844	-			
0x0000_0848	32-bit ReloadTimer2 DMAAn_RLTm_TMCSR 00000000_00000000_00000000_00000000			
0x0000_084C	-			
0x0000_0850	32-bit ReloadTimer2 DMAAn_RLTm_TMRLR XXXXXXXX_XXXXXXXX_XXXXXXXX_XXXXXXXX			
0x0000_0854	32-bit ReloadTimer2 DMAAn_RLTm_TMR XXXXXXXX_XXXXXXXX_XXXXXXXX_XXXXXXXX			
0x0000_0858 | 0x0000_085C	-			
0x0000_0860	32-bit ReloadTimer3 DMAAn_RLTm_DMACFG 00000000_00000000_00000000_00000000			
0x0000_0864	-			
0x0000_0868	32-bit ReloadTimer3 DMAAn_RLTm_TMCSR 00000000_00000000_00000000_00000000			
0x0000_086C	-			
0x0000_0870	32-bit ReloadTimer3 DMAAn_RLTm_TMRLR XXXXXXXX_XXXXXXXX_XXXXXXXX_XXXXXXXX			
0x0000_0874	32-bit ReloadTimer3 DMAAn_RLTm_TMR XXXXXXXX_XXXXXXXX_XXXXXXXX_XXXXXXXX			

Offset	Register Name/Initial Value			
	+3	+2	+1	+0
0x0000_0878 | 0x0000_087C	-			

Note:

- *The initial register value after reset indicates as follows:*
  - "1": Initial value "1"
  - "0": Initial value "0"
  - “X”: Initial value undefined
  - “-“: Reserved bit/Undefined bit
  - “*”: Initial value "0" or "1" according to the setting

4.1. DMA Additional Control Additional Status Register 0 (DMAAn_ASR0)

This register shows HBUSREQ status

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	HBUSREQ
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit0] HBUSREQ : HBUSREQ Status

HBUSREQ is asserted to "1" at transfer starts. HBUSREQ is de-asserted to "0" at transfer completion of current block.

4.2. DMA Additional Control Additional Status Register 1 (DMAAn_ASR1)

This register shows status of transfer of current block.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	BC_READ5	BC_READ4	BC_READ3	BC_READ2	BC_READ1	BC_READ0
ACCESS_TYPE	R0,WX	R0,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	Reserved	BC_WRITE5	BC_WRITE4	BC_WRITE3	BC_WRITE2	BC_WRITE1	BC_WRITE0
ACCESS_TYPE	R0,WX	R0,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit21:16] BC_READ[5:0] : Block Count for Reading

BC_READ shows number of remained block count in the current block of reading from source.

[bit5:0] BC_WRITE[5:0] : Block Count for Writing

BC_WRITE shows number of remained block count in the current block of writing to destination.

4.3. DMA Additional Control Additional Status Register 2 (DMAAn_ASR2)

This register shows selected channel by the DMA Arbiter

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	SELECTED 15	SELECTED 14	SELECTED 13	SELECTED 12	SELECTED 11	SELECTED 10	SELECTED 9	SELECTED 8
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	SELECTED 7	SELECTED 6	SELECTED 5	SELECTED 4	SELECTED 3	SELECTED 2	SELECTED 1	SELECTED 0
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit15:0] SELECTED[15:0]

The bits show current selected channel by DMA Arbiter.

Bits	Description
1000_0000_0000_0000	Channel 15 is selected by DMA arbiter
0100_0000_0000_0000	Channel 14 is selected by DMA arbiter
.....	.....
0000_0000_0000_0001	Channel 0 is selected by DMA arbiter
0000_0000_0000_0000	No channel is selected

4.4. DMA Additional Control Additional Status Register 3 (DMAAn_ASR3)

This register shows the requested status.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	REQ15	REQ14	REQ13	REQ12	REQ11	REQ10	REQ9	REQ8
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	REQ7	REQ6	REQ5	REQ4	REQ3	REQ2	REQ1	REQ0
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit15:0] REQ[15:0]

When the bit is "1", the channel is received a request but it has not executed.

Bit	Description
1	There is a pending request in channel 'x'.
0	No request

4.5. DMA Additional Control Additional Status Register i (DMAAn_ASRi) (i=4 to 19)

The register shows the number of resting of transfers in DMA channel 'i-4'

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	TC_SHDW 16
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	TC_SHDW 15	TC_SHDW 14	TC_SHDW 13	TC_SHDW 12	TC_SHDW 11	TC_SHDW 10	TC_SHDW 9	TC_SHDW 8
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	TC_SHDW 7	TC_SHDW 6	TC_SHDW 5	TC_SHDW 4	TC_SHDW 3	TC_SHDW 2	TC_SHDW 1	TC_SHDW 0
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit16:0] TC_SHDW[16:0]

The bits show the number of resting of transfers in DMA channel 'i-4'.

4.6. DMA Additional Control CMCHIC Reload Data Bank Register i (DMAAn_CMCHICRDBi) (i=0 to 7)

The register holds data to be copied to corresponding DMAn_CMCHICj register at CMCHIC reloading. For CMCHIC reloading, refer to 3.2 CMCHIC Reloading.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	RLESEL5	RLESEL4	RLESEL3	RLESEL2	RLESEL1	RLESEL0
ACCESS_TYPE	R/W0	R/W0	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	RLSLOT2	RLSLOT1	RLSLOT0
ACCESS_TYPE	R/W0	R/W0	R/W0	R/W0	R/W0	R/W	R/W	R/W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	CI8
ACCESS_TYPE	R/W0	R/W0	R/W0	R/W0	R/W0	R/W0	R/W0	R/W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	CI7	CI6	CI5	CI4	CI3	CI2	CI1	CI0
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit29:24] RLESEL[5:0]

The bits stores data which is copied to corresponding DMAn_CMCHICj:RLESEL[5:0] bits at CMCHIC reloading.

[bit18:16] RLSLOT[2:0]

The bits stores data which is copied to corresponding DMAn_CMCHICj:RLSLOT[2:0] bits at CMCHIC reloading.

[bit8:0] CI[8:0]

The bits stores data which is copied to corresponding DMAn_CMCHICj:CI[8:0] bits at CMCHIC reloading.

4.7. DMA Additional Control Reload Timer Trigger Select Register (DMAAn_RTTSR)

By this register, TIN inputs of Reload Timer instances are specified.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	RLT3TS6	RLT3TS5	RLT3TS4	RLT3TS3	RLT3TS2	RLT3TS1	RLT3TS0
ACCESS_TYPE	R0,WX	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	RLT2TS6	RLT2TS5	RLT2TS4	RLT2TS3	RLT2TS2	RLT2TS1	RLT2TS0
ACCESS_TYPE	R0,WX	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	RLT1TS6	RLT1TS5	RLT1TS4	RLT1TS3	RLT1TS2	RLT1TS1	RLT1TS0
ACCESS_TYPE	R0,WX	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	RLT0TS6	RLT0TS5	RLT0TS4	RLT0TS3	RLT0TS2	RLT0TS1	RLT0TS0
ACCESS_TYPE	R0,WX	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit30:24] RLT3TS[6:0]: Reload timer 3 trigger select

This bit selects the TIN source of Reload Timer 3 in DMA Additional Control block.

[bit22:16] RLT2TS[6:0]: Reload timer 2 trigger select

This bit selects the TIN source of Reload Timer 2 in DMA Additional Control block.

[bit14:8] RLT1TS[6:0]: Reload timer 1 trigger select

This bit selects the TIN source of Reload Timer 1 in DMA Additional Control block.

[bit6:0] RLT0TS[6:0]: Reload timer 0 trigger select

This bit selects the TIN source of Reload Timer 0 in DMA Additional Control block.

RLTkTS[6:0] (k = 3,2,1,0)	Description
0x00 to 0x0F	DREQ to DMA channel 0 to 15
0x10 to 0x1F	DREQ_ACK from DMA channel 0 to 15
0x20 to 0x2F	BSTART from DMA channel 0 to 15
0x30 to 0x3F	BDONE from DMA channel 0 to 15
0x40 to 0x4F	DEOP from DMA channel 0 to 15
0x50	HBUSREQ
Else	TIN is tied to "0".

Note:

- *Software trigger by DMAAn_RTSSSR register is always able to be used regardless RLTkTS[6:0] value.*

4.8. DMA Additional Control Reload Timer Synchronous Software Start Register (DMAAn_RTSSSR)

By using this register, a "0"-->"1"-->"0" pulse can be given to the TIN inputs of Reload Timer instances. The Reload Timer instances can be triggered simultaneously by using this register.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	Reserved	Reserved	Reserved	SSSR3	SSSR2	SSSR1	SSSR0
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,W	R0,W	R0,W	R0,W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit3:0] SSSR[3:0]: Synchronous software start register

By writing to "1" to bit k (k=3,2,1,0), a "0"→"1"→"0" pulse is given to TIN input of Reload Timer instance 'k'. Writing "0" has no effects.

SSSR[k] (k = 3,2,1,0)	Description
1	Give "0"→"1"→"0" pulse to TIN input of Reload Timer instance 'k'.
0	No effect.

5. Overview in DMA COMPLEX SUBSYSTEM

This section gives a brief overview of the 32-bit Reload Timer in DMA COMPLEX SUBSYSTEM.

Features of 32-Bit Reload Timer

The 32-bit Reload Timer consists of a 32-bit down counter, a 32-bit Reload register, one input(TIN), one output (TOT), and control registers. The 32-bit Reload Timer has the following features:

- External(TIN can capture DMA event (*1)) and internal clock/event source
- Trigger signal programmable as rising/falling edge or both
- Gated count function
- One-shot or reload counter mode
- Counter state can be made visible at external pin (*2)
- Prescaler with six different settings for the internal clock and two different settings for the external clock
- Support for MCU debug mode

Notes:

- *(*1) For the connection, please refer to 3.3.1 Reload Timer Trigger Inputs.*
- *(*2) Please refer to device specific datasheet whether the external output pin is supported or not.*

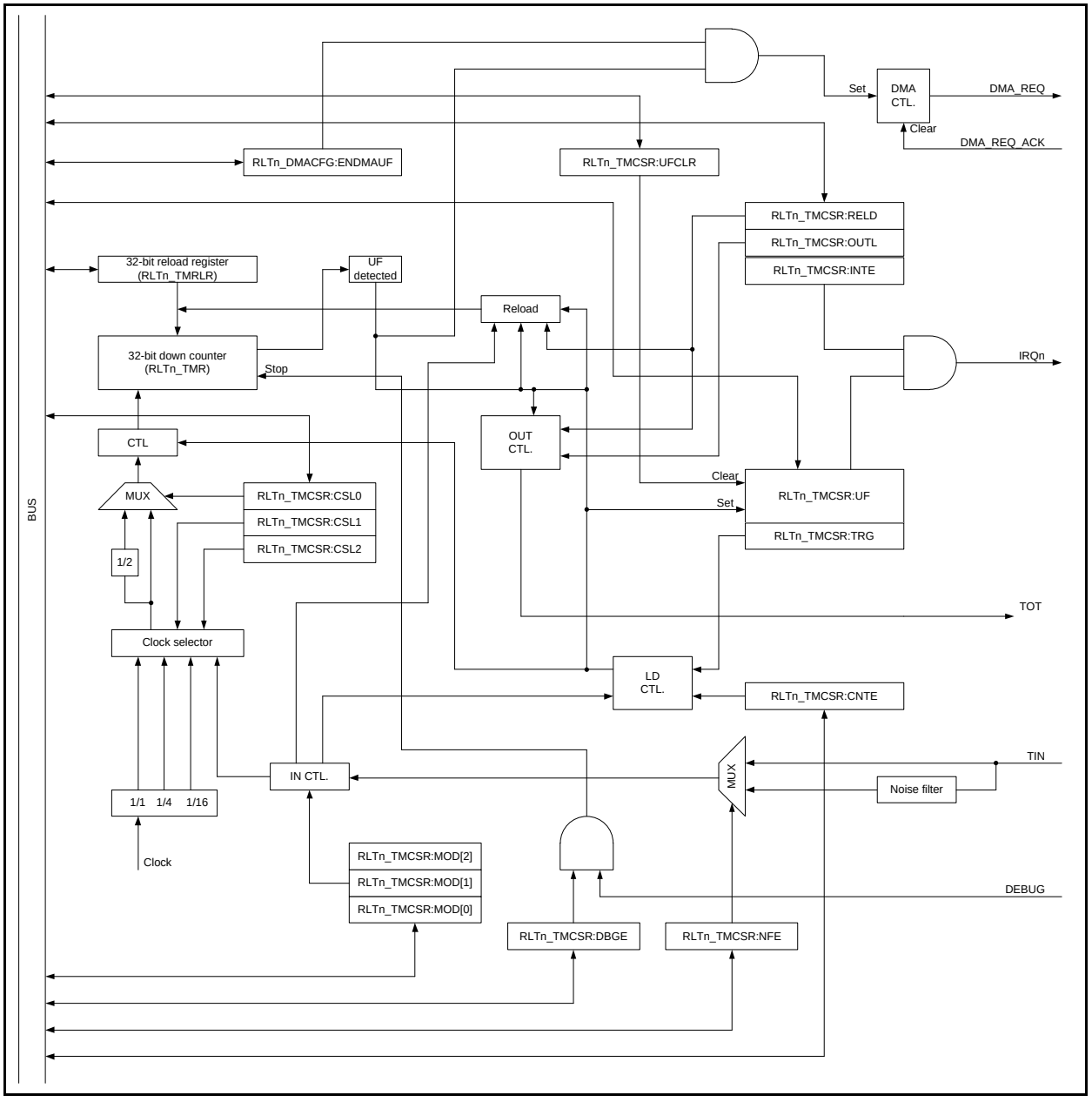
DMA and Interrupts

The 32-bit Reload Timer can generate DMA request which can be used to start DMA transfers.
The 32-bit Reload Timer can generate interrupt in case of underflow.

6. Configuration and Block Diagram in DMA COMPLEX SUBSYSTEM

This section shows a block diagram of the 32-bit Reload Timer in DMA COMPLEX SUBSYSTEM.

Figure 6-1 Block Diagram of 32-Bit Reload Timer in DMA COMPLEX SUBSYSTEM



Notes:

- *In the figure(s) in this section, RLTn_XXXXX means DMAAn_RLTm_XXXXX register.*
- *Registers of the same type are provided for individual RLTs, and "m" in the abbreviated register name indicates the RLT number m (0 to 3).*
- *For the connection of TIN, please refer to 3.3.1 Reload Timer Trigger Inputs.*

7. Operation of the 32-Bit Reload Timer in DMA COMPLEX SUBSYSTEM

This section describes the operation of the 32-bit Reload Timer in DMA COMPLEX SUBSYSTEM.

7.1 Internal Clock and External Event Counter Operations of 32-Bit Reload Timer
7.2 Underflow Operation of 32-Bit Reload Timer
7.3 Output Functions of 32-Bit Reload Timer
7.4 Counter Operation State
7.5 DMA Operation

7.1. Internal Clock and External Event Counter Operations of 32-Bit Reload Timer

In internal clock mode, the peripheral clock with different divider settings can be selected as the clock source for operating the Reload Timer. The external input TIN can be selected as either a trigger input, or as a gate input by a register setting.
In event counter mode, the TIN is used as an external event input. Each active edge on this input (rising, falling, or both) decrements the counter.
When DMAAn_RLTm_TMCSR:CSL0 = 1, then each second event is counted.

Internal Clock Operation of 32-Bit Reload Timer

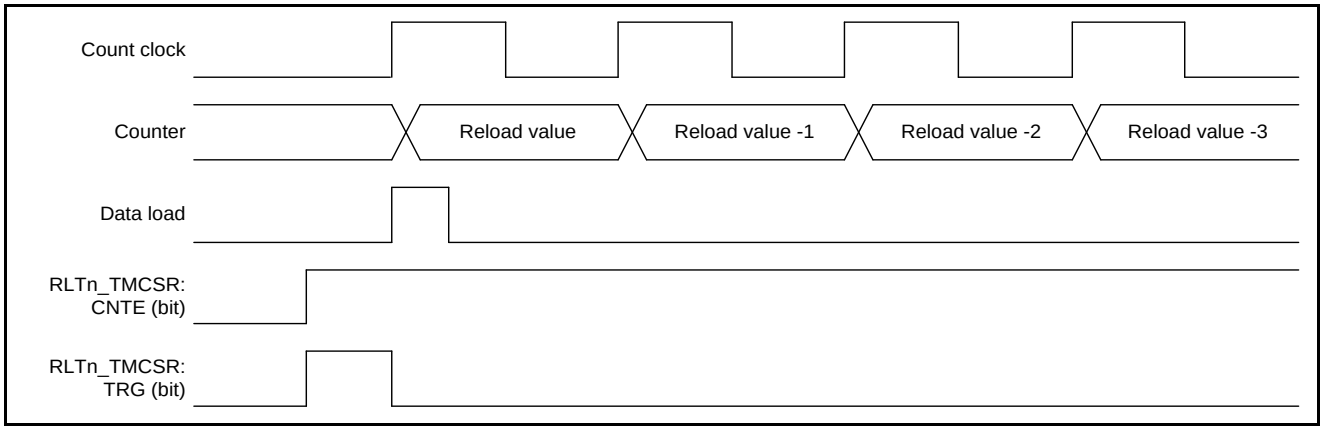
Writing "1" to both DMAAn_RLTm_TMCSR:CNTE and DMAAn_RLTm_TMCSR:TRG bits enables and starts counting at the same time. Using the DMAAn_RLTm_TMCSR:TRG bit as a trigger input is always available when the timer is enabled (DMAAn_RLTm_TMCSR:CNTE = 1), regardless of the operation mode.

Figure 7-1 shows counter activation and counter operation.

Notes:

- *In the figure(s) in this section, RLTn_XXXXX means DMAAn_RLTm_XXXXX register.*
- *Registers of the same type are provided for individual RLTs, and "m" in the abbreviated register name indicates the RLT number m (0 to 3).*

Figure 7-1 Activation and Operation of 32-Bit Reload Timer Counter



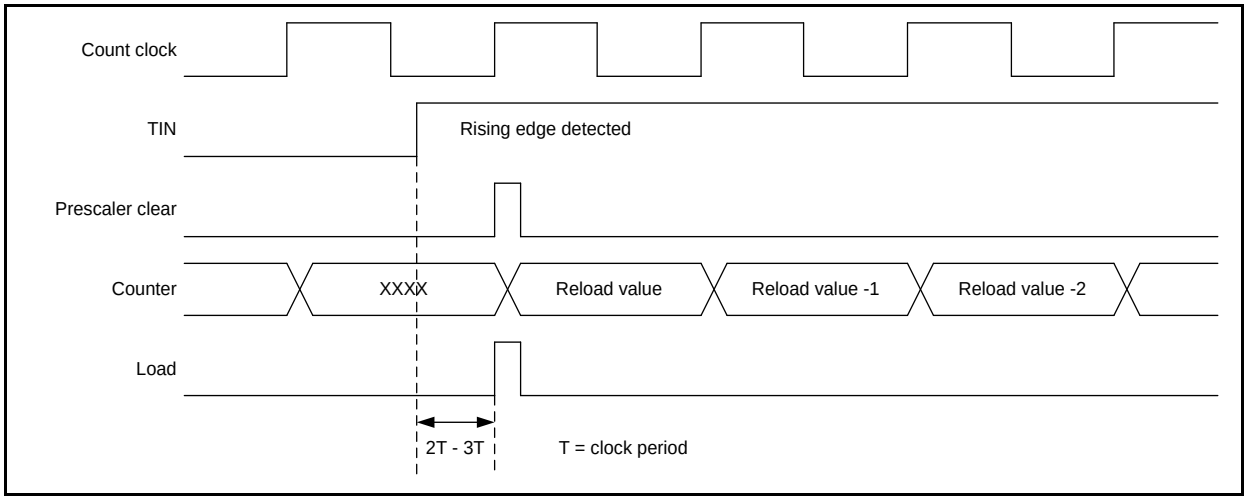
Input Functions of 32-Bit Reload Timer (In Internal Clock Mode)

The TIN input can be used as either a trigger input, or as a gate input, when an internal clock is selected as the clock source.

Trigger Input

When used as a trigger input, an active edge causes the timer to load the reload register contents and resets the internal prescaler. Then, count operation starts. For the minimum pulse width length of TIN refer to device specific datasheet.

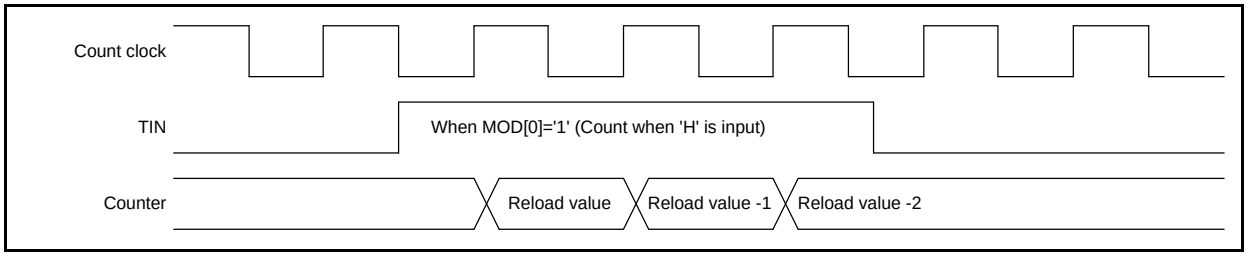
Figure 7-2 Trigger Input Operation of 32-Bit Reload Timer



Gate Input

When used as a gate input, the counter only counts while the active level specified by the DMAAn_RLTm_TMCSR:MOD[0] bit is input to the TIN. In this case, the count clock continues to operate unless stopped. The software trigger can be used in gate mode, regardless of the gate level. For the minimum pulse width length of TIN refer to device specific datasheet.

Figure 7-3 Gate Input Operation of 32-Bit Reload Timer



External Event Counter

When external event count mode is selected, the TIN is used as an external event input. The counter counts on the active edge specified in the DMAAn_RLTm_TMCSR. For the minimum pulse width length of TIN refer to device specific datasheet.

7.2. Underflow Operation of 32-Bit Reload Timer

An underflow is defined for this timer as the time when the counter value changes from 0x00000000 to 0xFFFFFFFF or reload occurs (DMAAn_RLTm_TMCSR:RELD= 1). Therefore, an underflow occurs after (reload register setting + 1) counts.

Underflow Operation of 32-Bit Reload Timer

If the DMAAn_RLTm_TMCSR:RELD bit is "1" and an underflow occurs, the contents of the reload register is loaded into the counter and counting continues.

If the DMAAn_RLTm_TMCSR:RELD bit is "0", counting stops when counter reaches 0xFFFFFFFF.

The DMAAn_RLTm_TMCSR:UF bit is set when the underflow occurs. If the DMAAn_RLTm_TMCSR:INTE bit is "1" at this time, an interrupt request is generated.

Figure 7-4 shows the operation when an underflow occurs with various values of DMAAn_RLTm_TMCSR:RELD. Figure 7-5 shows the clearing operation of underflow flag.

Notes:

- *In the figure(s) in this section, RLTn_XXXXX means DMAAn_RLTm_XXXXX register.*
- *Registers of the same type are provided for individual RLTs, and "m" in the abbreviated register name indicates the RLT number m (0 to 3).*

Figure 7-4 Underflow Operation of 32-Bit Reload Timer

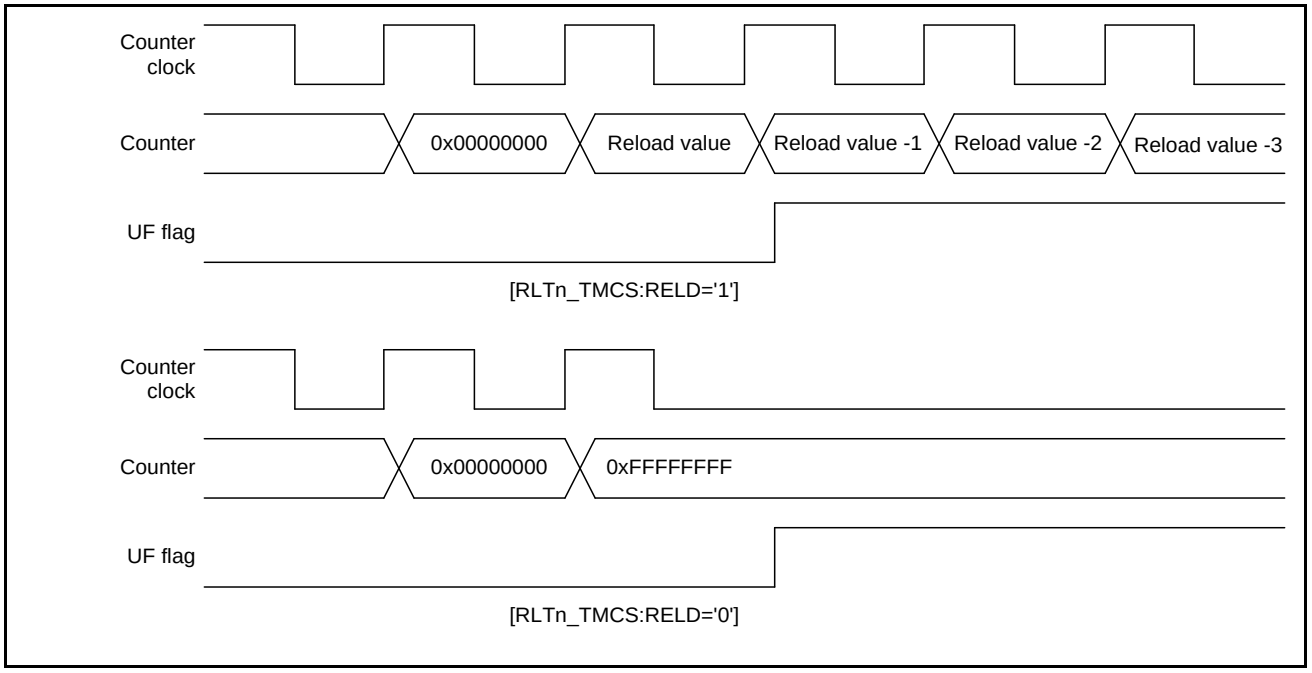
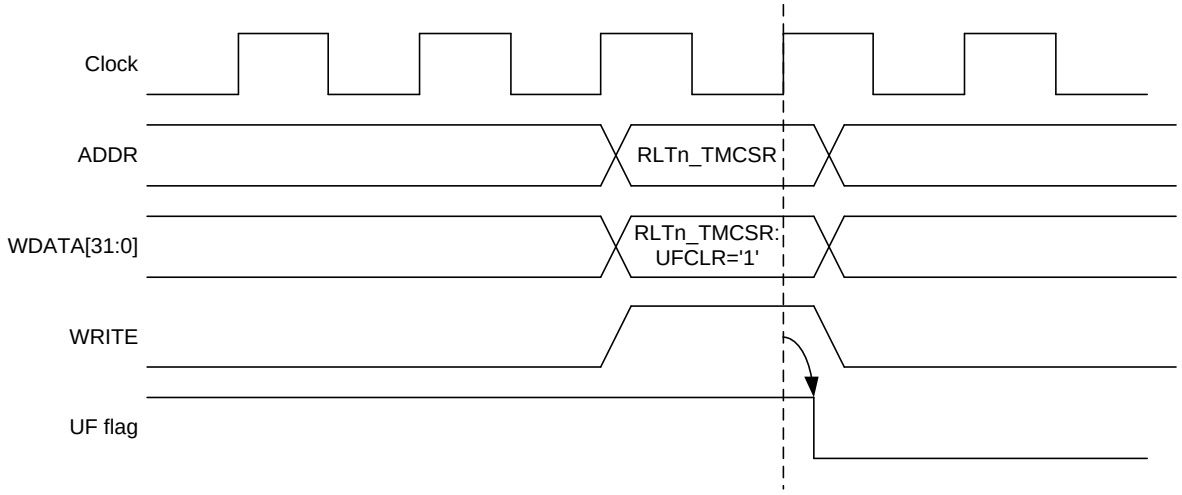


Figure 7-5 Clearing of Underflow Bit



7.3. Output Functions of 32-Bit Reload Timer

In reload mode, the TOT output performs toggle output (inverts at each underflow). In one-shot mode, the TOT output is used as a pulse output that shows the configured level while the counting is in progress.

Output Signal Functions of 32-Bit Reload Timer

The DMAAn_RLTm_TMCSR:OUTL bit sets the output polarity.
When DMAAn_RLTm_TMCSR:OUTL = 0, the initial value for toggle output is "L" and the one-shot pulse output is "H" while the count is in progress.
When DMAAn_RLTm_TMCSR:OUTL = 1, the output waveforms are opposite.
Figure 7-6 and Figure 7-7 show the output signal functions.

Notes:

- *In the figure(s) in this section, RLTn_XXXXX means DMAAn_RLTm_XXXXX register.*
- *Registers of the same type are provided for individual RLTs, and "m" in the abbreviated register name indicates the RLT number m (0 to 3).*

Figure 7-6 Output Signal Function of 32-Bit Reload Timer in Reload Mode

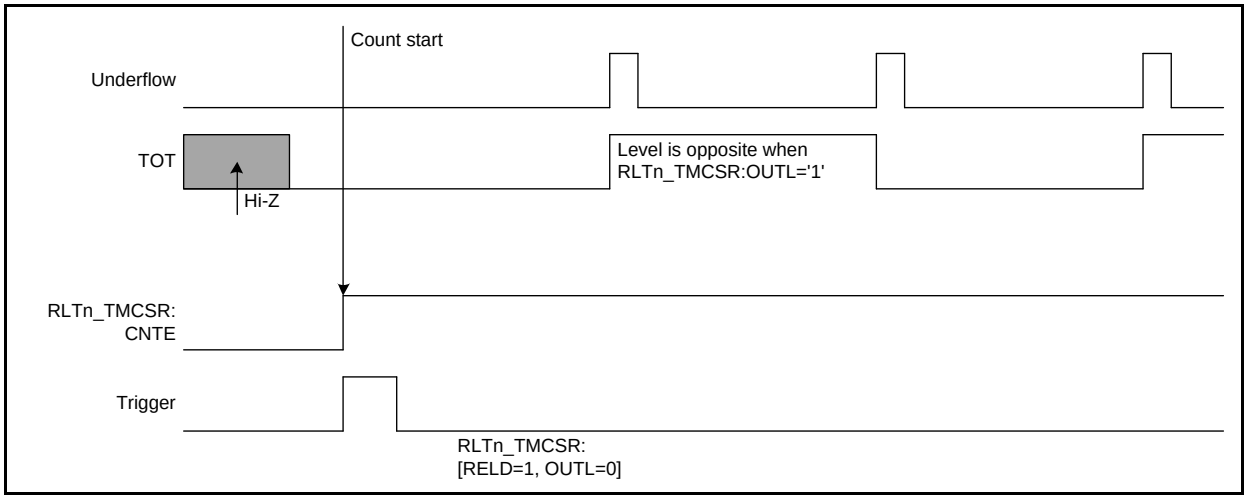
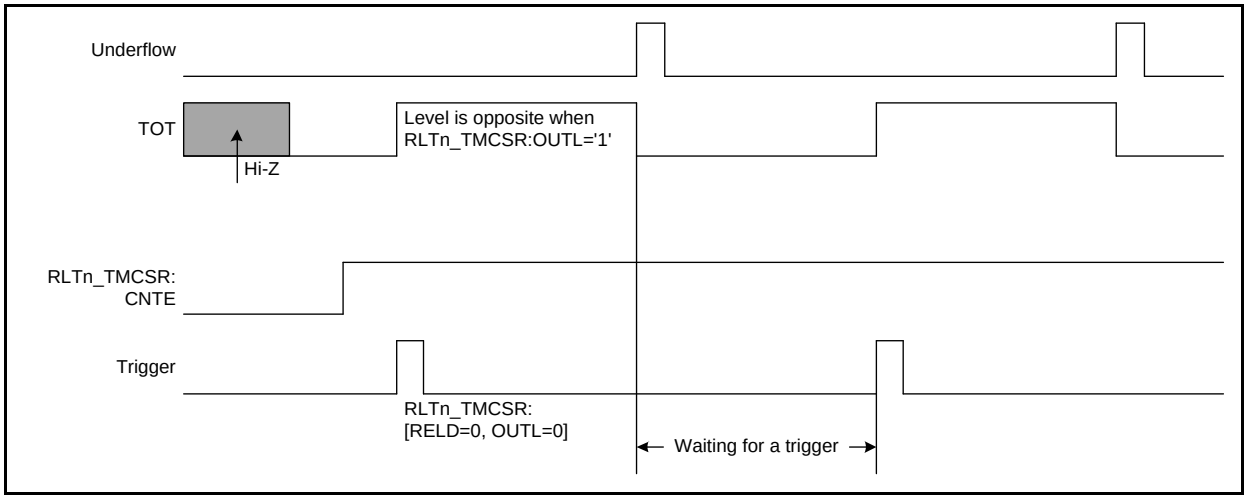


Figure 7-7 Output Signal Function of 32-Bit Reload Timer in One-Shot Mode



7.4. Counter Operation State

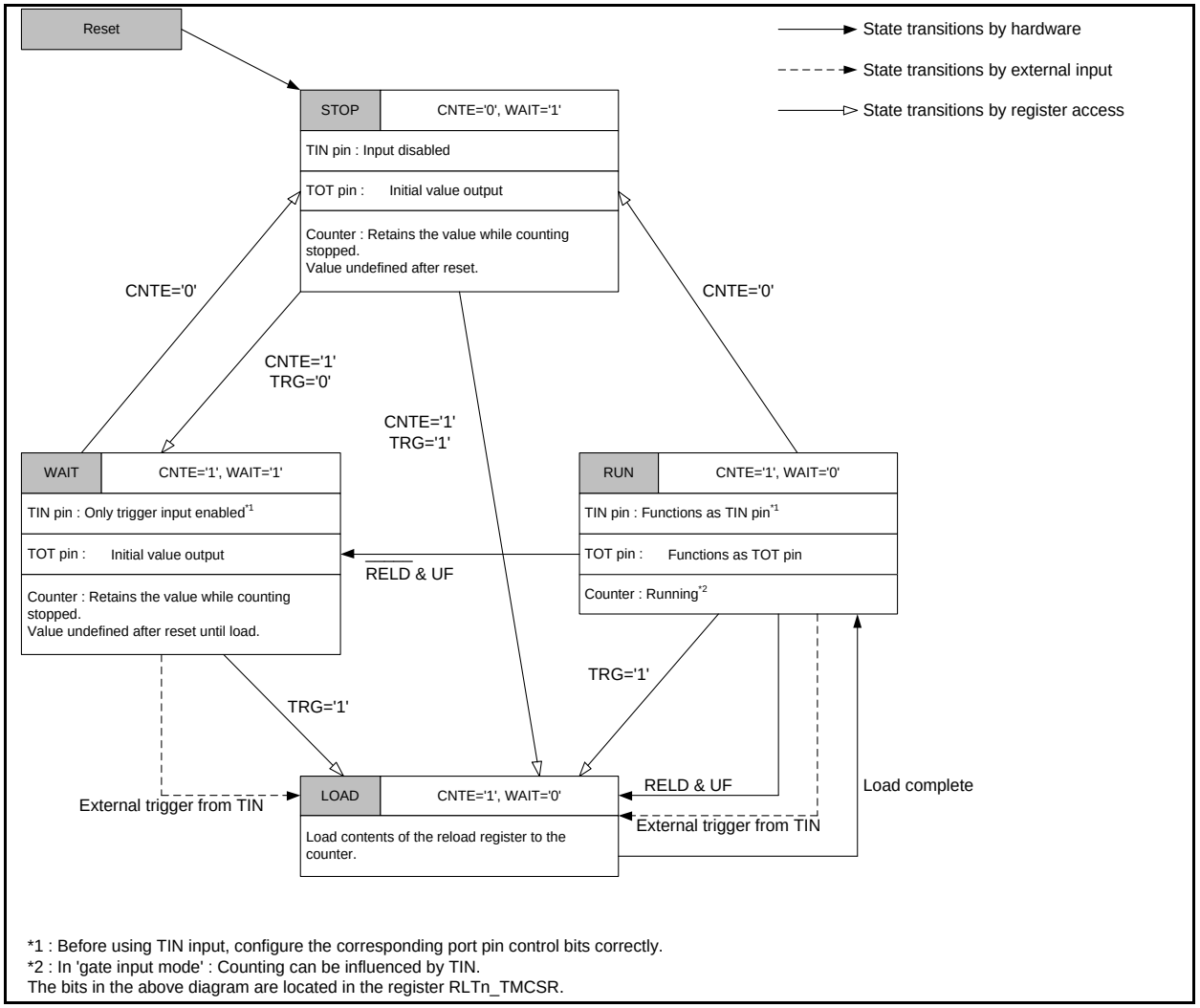
The counter state is determined by DMAAn_RLTm_TMCSR:CNTE bit in the Timer Control Status Register and the internal WAIT signal.

Available States for Reload Timer Are:

Stop state: DMAAn_RLTm_TMCSR:CNTE = 0 and WAIT = 1
Wait state: DMAAn_RLTm_TMCSR:CNTE = 1 and WAIT = 1
Run state: DMAAn_RLTm_TMCSR:CNTE = 1 and WAIT = 0

Counter Operation States

Figure 7-8 Counter Operation States



Notes:

- *In the figure(s) in this section, RLTn_XXXXX means DMAAn_RLTm_XXXXX register.*
- *Registers of the same type are provided for individual RLTs, and "m" in the abbreviated register name indicates the RLT number m (0 to 3).*
- *For the connection of TIN, please refer to 3.3.1 Reload Timer Trigger Inputs.*

7.5. DMA Operation

The DMA support is determined by the DMAAn_RLTm_DMACFG:ENDMAUF bit. Setting this bit enables DMA request generation for DMA. Assertion of DMA_REQ_ACK signal acknowledges request and hence DMA_REQ signal gets de-asserted.

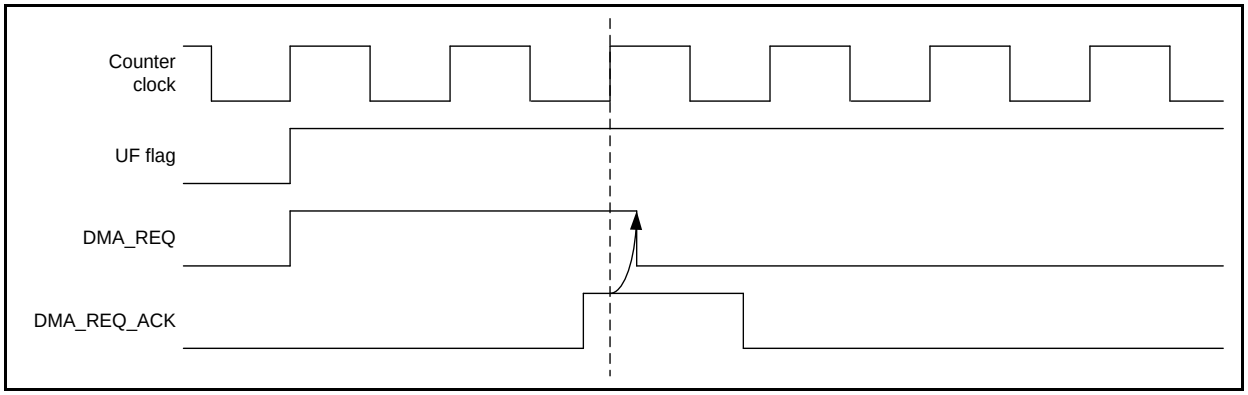
Enabling DMA Support

Writing "1" to DMAAn_RLTm_DMACFG:ENDMAUF enables DMA request generation for DMA when DMAAn_RLTm_TMCSR:UF bit sets. However, writing "0" to DMAAn_RLTm_DMACFG:ENDMAUF disables DMA request generation even if DMAAn_RLTm_TMCSR:UF bit sets.
When DMA_REQ_ACK is asserted the DMA_REQ signal gets de-asserted by acknowledging the DMA request.
Figure 7-9 shows behavior of DMA_REQ_ACK when asserted.

Note:

- *Registers of the same type are provided for individual RLTs, and "m" in the abbreviated register name indicates the RLT number m (0 to 3).*

Figure 7-9 De-Asserting DMA_REQ Signal



8. Registers in DMA COMPLEX SUBSYSTEM

This section describes the registers of 32-bit Reload Timer in DMA COMPLEX SUBSYSTEM.

Table 8-1 Memory Layout of 32-Bit Reload Timer Registers

Offset	+3	+2	+1	+0
0x00000000	DMAAn_RLTm_DMACFG 00000000_00000000_00000000_00000000			
0x00000004	Reserved			
0x00000008	DMAAn_RLTm_TMCSR 00000000_00000000_00000000_00000000			
0x0000000C	Reserved			
0x00000010	DMAAn_RLTm_TMRLR XXXXXXXX_XXXXXXXX_XXXXXXXX_XXXXXXXX			
0x00000014	DMAAn_RLTm_TMR XXXXXXXX_XXXXXXXX_XXXXXXXX_XXXXXXXX			

Note:

- *The initial register value after reset indicates as follows:*
  - "1": Initial value "1"
  - "0": Initial value "0"
  - “X”: Initial value undefined
  - “-“: Reserved bit/Undefined bit
  - “*”: Initial value "0" or "1" according to the setting

8.1. DMA Configuration Register（DMAAn_RLTm_DMACFG）

The DMA Configuration Register controls the DMA request generation for underflow condition. Registers of the same type are provided for individual RLTs, and "m" in the abbreviated register name indicates the RLT number m (0 to 3).

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	ENDMAUF
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:1] Reserved

[bit0] ENDMAUF : DMA enable for underflow (ENDMAUF)

Bit	Description
0	No DMA request is generated
1	A DMA request is generated when the counter, DMAAn_RLTm_TMR, underflows

8.2. Timer Control Status Register（DMAAn_RLTm_TMCSR）

The Timer Control Status Register controls the operation mode and interrupt of the 32-bit Reload Timer in DMA COMPLEX SUBSYSTEM. Registers of the same type are provided for individual RLTs, and "m" in the abbreviated register name indicates the RLT number m (0 to 3).

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	CNTE
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	TRG	UFCLR	UF
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,W	R0,W	R,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	MOD[2]	MOD[1]	MOD[0]	CSL2	CSL1	CSL0	Reserved	NFE
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R0,WX	R/W0
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	DBGE	Reserved	OUTL	RELD	INTE	Reserved	Reserved	Reserved
ACCESS_TYPE	R/W	R0,W0	R/W	R/W	R/W	R0,WX	R0,WX	R0,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:25] Reserved

[bit24] CNTE: Count enable

The Count Enable (CNTE) bit is a timer count enable bit.

Bit	Description
0	Stops count operation
1	Sets the timer to wait for a trigger

[bit23:19] Reserved

[bit18] TRG: Trigger

Software trigger bit.

Bit	Description
0	No effect
1	Applies a software trigger, causing the timer to load the reload register content to the counter and starts counting

Notes:

- *Applying a trigger using this register is only valid when DMAAn_RLTm_TMCSR:CNTE = 1. If DMAAn_RLTm_TMCSR:CNTE = 0, writing "1" to TRG has no effect.*
- *Set this bit in 'gate input mode' and in 'event count mode' to load the reload register content before counting starts.*

[bit17] UFCLR: Underflow interrupt clear

Bit	Description
0	No effect
1	Clears DMAAn_RLTm_TMCSR:UF bit

[bit16] UF: Underflow

The Underflow (UF) is timer interrupt request flag.

Bit	Description
0	No underflow occurred
1	When an underflow occurred

Note:

- *UF bit is cleared by writing "1" to DMAAn_RLTm_TMCSR:UFCLR bit.*

[bit15:13] MOD: Operation mode

The Operation Mode (MOD[2:0]) bits set the operation mode and input (TIN) functions.
Table 8-2 and Table 8-3 list the MOD[2:0] bit settings.

[bit12] CSL2: Clock select 2

The Clock Select 2 (CSL2) bit specifies the clock/event source and the clock division ratio.
Table 8-4 lists the selected clock sources for different CSL0/1/2 settings.

[bit11] CSL1: Clock select 1

The Clock Select 1 (CSL1) bit specifies the clock/event source and the clock division ratio.
Table 8-4 lists the selected clock sources for different CSL0/1/2 settings.

[bit10] CSL0: Clock select 0

The Clock Select 0 (CSL0) bit specifies the count clock division ratio.
Table 8-4 lists the selected clock sources for different CSL0/1/2 settings.

[bit9] Reserved

[bit8] NFE: Noise filter enable

Always write "0" to this bit.

[bit7] DBGE: Debug mode enable

This bit is used to enable/disable debug mode for RLT.

Bit	Description
0	Debug mode disabled
1	Debug mode enabled

Notes:

- *When DBGE is set to "1" and the processor is in debug state, the timer counter operation is paused, and writing to the DMAAn_RLTm_TMRLR register directly updates the timer counter (DMAAn_RLTm_TMR register). When the processor leaves debug state or DBGE is set to "0", the timer counter operation is resumed.*

[bit6] Reserved

[bit5] OUTL: Output level

This Output Level (OUTL) bit sets the output level for the TOT.
Refer to Table 8-5.

[bit4] RELD: Reload

This Reload (RELD) bit enables reload operations.
Refer to Table 8-5.

Bit	Description
0	The timer operates in one-shot mode. In this mode, the count operation stops when an underflow occurs due to the counter value changing from 0x00000000 to 0xFFFFFFFF
1	The timer operates in reload mode. In this mode, the timer loads the reload register contents into the counter and continues counting whenever an underflow occurs

[bit3] INTE: Interrupt enable

Timer interrupt request enable bit.

Bit	Description
0	No interrupt request is generated even when the DMAAn_RLTm_TMCSR:UF bit changes to "1"
1	An interrupt request is generated when the DMAAn_RLTm_TMCSR:UF bit changes to "1"

[bit2:0] Reserved

Table 8-2 DMAAn_RLTm_TMCSR:MOD[2:0] Bit Settings for Internal Clock Mode (DMAAn_RLTm_TMCSR:CSL1 / DMAAn_RLTm_TMCSR:CSL2 = 0b00, 0b01, or 0b10)

MOD[2]	MOD[1]	MOD[0]	Input Function	Active Edge or Level
0	0	0	Trigger disabled	-
0	0	1	Trigger input	Rising edge
0	1	0		Falling edge
0	1	1		Both edge
1	x	0	Gate input	"L" level
1	x	1		"H" level

Table 8-3 DMAAn_RLTm_TMCSR:MOD[2:0] Bit Settings for Event Counter Mode (DMAAn_RLTm_TMCSR:CSL1 / DMAAn_RLTm_TMCSR:CSL2 = 0b11)

MOD[2]	MOD[1]	MOD[0]	Input Function	Active Edge or Level
x	0	0	-	-
	0	1	Event input	Rising edge
	1	0		Falling edge
	1	1		Both edge

Table 8-4 Clock Sources for DMAAn_RLTm_TMCSR:CSL0 / DMAAn_RLTm_TMCSR:CSL1 / DMAAn_RLTm_TMCSR:CSL2 Bit Settings

CSL2	CSL1	CSL0	Count Clock (Time for Peripheral Clock)
0	0	0	1/1
0	0	1	1/2
0	1	0	1/4
0	1	1	1/8
1	0	0	1/16
1	0	1	1/32
1	1	0	External event count mode, each event on TIN
1	1	1	External event count mode, each second event on TIN

Table 8-5 DMAAn_RLTm_TMCSR: DMAAn_RLTm_TMCSR:OUTL, and DMAAn_RLTm_TMCSR:RELD Settings

OUTL	RELD	Output Waveform
0	0	Output an "H" level pulse during counting.
1	0	Output an "L" level pulse during counting.
0	1	Toggle output. Starts with "L" level output. Changes level on timer reload.
1	1	Toggle output. Starts with "H" level output. Changes level on timer reload.

Note:

- *Bits marked 'x' in the table can be set to any value.*

8.3. 32-Bit Reload Register (DMAAn_RLTm_TMRLR)

The Timer Reload Register holds the reload value of the 32-bit Reload Timer in DMA COMPLEX SUBSYSTEM. Registers of the same type are provided for individual RLTs, and "m" in the abbreviated register name indicates the RLT number m (0 to 3).

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	TMRLR[31:24]							
ACCESS_TYPE	R/W							
PROT_TYPE	-							
INITIAL_VALUE	XXXXXXXX							

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	TMRLR[23:16]							
ACCESS_TYPE	R/W							
PROT_TYPE	-							
INITIAL_VALUE	XXXXXXXX							

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	TMRLR[15:8]							
ACCESS_TYPE	R/W							
PROT_TYPE	-							
INITIAL_VALUE	XXXXXXXX							

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	TMRLR[7:0]							
ACCESS_TYPE	R/W							
PROT_TYPE	-							
INITIAL_VALUE	XXXXXXXX							

[bit31:0] TMRLR: Timer reload register

The Timer Reload Register (DMAAn_RLTm_TMRLR) is a 32-bit reload register holds the reload value. Initial value is undefined.

This register can be accessed only in 32-bit/64-bit mode.

When DMAAn_RLTm_TMCSR:DBGE is set to "1" and the processor is in debug state, writing to this register updates the timer counter immediately.

Note:

- *This register is not initialized by Hard Reset (i.e. “X”).*

8.4. 32-Bit Timer Register (DMAAn_RLTm_TMR)

Reading this register returns the count value of the 32-bit Reload Timer in DMA COMPLEX SUBSYSTEM. The initial value is undefined. Registers of the same type are provided for individual RLTs, and "m" in the abbreviated register name indicates the RLT number m (0 to 3).

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	TMR[31:24]							
ACCESS_TYPE	R,WX							
PROT_TYPE	-							
INITIAL_VALUE	XXXXXXXX							

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	TMR[23:16]							
ACCESS_TYPE	R,WX							
PROT_TYPE	-							
INITIAL_VALUE	XXXXXXXX							

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	TMR[15:8]							
ACCESS_TYPE	R,WX							
PROT_TYPE	-							
INITIAL_VALUE	XXXXXXXX							

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	TMR[7:0]							
ACCESS_TYPE	R,WX							
PROT_TYPE	-							
INITIAL_VALUE	XXXXXXXX							

[bit31:0] TMR: Timer register

Reading this Timer Register (DMAAn_RLTm_TMR) returns the count value of the 32-bit Reload Timer. Initial value is undefined.

This register can be accessed only in 32-bit/64-bit mode.

Note:

- *This register is not initialized by Hard Reset (i.e. "X").*

CHAPTER 29: MPU16 AHB

This chapter explains the function and operation of the Memory Protection Unit for the AMBA Advanced High Speed Bus (MPU16 AHB).

1. **Outline of the MPU16 AHB**
2. **MPU16 AHB Registers**
3. **Operation of the MPU16 AHB**

MPU16AHB-TXXPT03P01R01L06-E1-XX

1. Outline of the MPU16 AHB

This section describes the features and the block diagram of the MPU16 AHB.

Features of MPU16 AHB

The MPU16 AHB module monitors the accesses from AHB Masters and checks each access against an authorized set of Access Permissions. Access Permissions (known as “permission attributes” from here onwards) are defined by Access Permissions (AP) bits. These AP bits are explained in Section MPU access permissions.

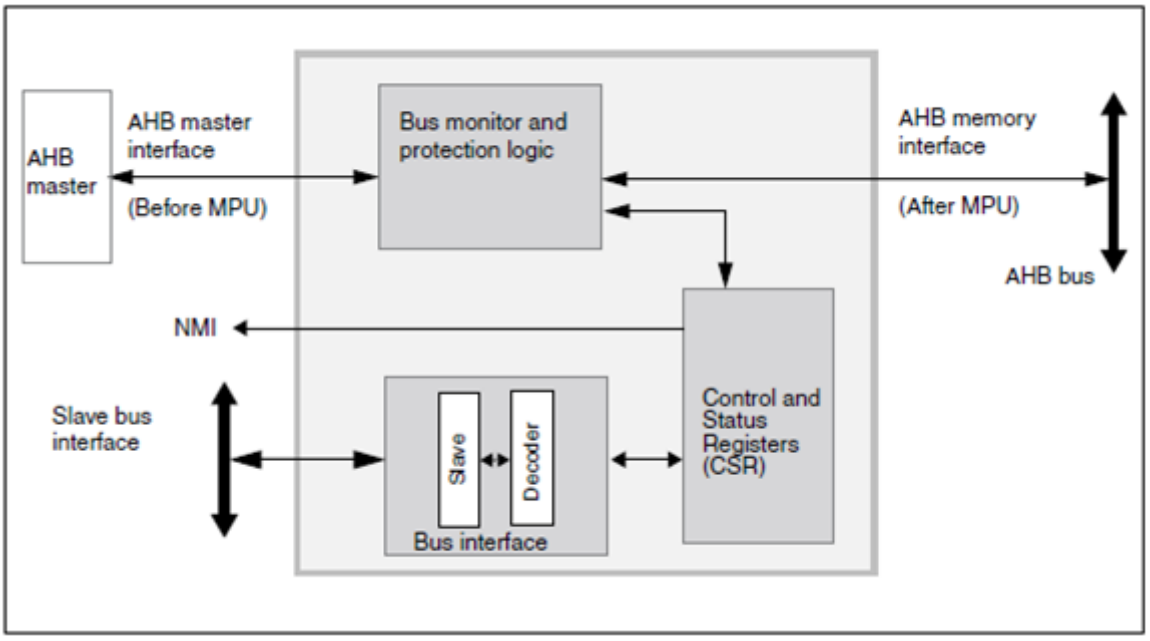
The MPU16 AHB provides 16 regions and one background region. Each region has corresponding Access Permission bits that define the permission attributes for that particular region. Any unauthorized access to that memory space is flagged using a non-maskable interrupt. MPU16 AHB also collects information about unauthorized bus access and stores it in its internal registers.

The features of the MPU16 AHB module are listed in this section:

- Each of the 16 regions in MPU16 AHB is specified using corresponding start address and end address
- The background region covers the entire 4GB address space
- With an unauthorized access, the MPU16 AHB generates an NMI to the CPU
- The MPU16 AHB collects information about the AHB Master bus access that caused the unauthorized access
- It supports 8-, 16- and 32-bit bus accesses for the configuration of the MPU16 AHB registers
- It supports lock and unlock feature to protect registers from illegal write accesses
- The modification of registers in the MPU16 AHB is only allowed in privileged mode
- MPU16 AHB registers can be written only after execution of the unlock sequence
- It supports a privileged mode overwrite feature that allows the privilege attribute of the memory side AHB interface to be overwritten

2. Configuration and Diagram

Figure 2-1 Block Diagram of MPU16 AHB



Bus Interface

The bus masters can access the MPU16 AHB module through its slave bus interface.

Bus Monitor and Protection Logic

This logic monitors the transfers on AHB master interface bus. It finds out the region/s where the current transfer belongs to and then applies permissions based on the region match. It signals any permission violation using an NMI interrupt. All transfers on the AHB Master Interface (including the one that caused permission violation) are blocked until the NMI is cleared.

Control and Status Registers

The operation of the MPU16 AHB can be controlled and monitored through its Control and Status Registers (CSR). For more details about the CSR refer to Section 2.

3. MPU16 AHB Registers

The MPU16 AHB module contains various registers to configure its operation, to monitor its status and to read the information it has collected from the AHB master interface at the time of a memory protection violation.

The MPU16 AHB module is allocated 4 KB of MCU address space for mapping the Configuration and Status Registers (CSRs). The address area allocated to the MPU16 AHB and the CSRs in the MPU16 AHB are explained in this section.

The suffix ‘n’ in the register name indicates that the register is an instance ‘n’ of the module.

Registers of MPU16 AHB

The following registers are available for the MPU16 AHB:

- MPU16 AHB Control Register (MPUHn_CTRL0)
- MPU16 AHB NMI Enable Register (MPUHn_NMIEN)
- MPU16 AHB Memory Error Control Register (MPUHn_MERRC)
- MPU16 AHB Memory Error Address Register (MPUHn_MERRA)
- MPU16 AHB Region Control Registers (MPUHn_CTRL1 to 16)
- MPU16 AHB Start Address Registers (MPUHn_SADDR1 to 16)
- MPU16 AHB End Address Registers (MPUHn_EADDR1 to 16)
- MPU16 AHB Unlock Register (MPUHn_UNLOCK)
- MPU16 AHB Module ID Register (MPUHn_MID)

Memory Layout of MPU16 AHB Registers

Table 3-1 Memory Layout of MPU16 AHB Registers

Offset	+3	+2	+1	+0
0x0000_0000	MPUHn_CTRL0 00000000_00000000_00000001_00000000			
0x0000_0004	MPUHn_NMIEN 00000000_00000000_00000000_00000001			
0x0000_0008	MPUHn_MERRC 00000000_00000000_00000000_00000000			
0x0000_000C	MPUHn_MERRA 00000000_00000000_00000000_00000000			
0x0000_0010	MPUHn_CTRL1 00000000_00000000_00000000_00000000			
0x0000_0014	MPUHn_SADDR1 00000000_00000000_00000000_00000000			
0x0000_0018	MPUHn_EADDR1 00000000_00000000_00000000_01111111			
0x0000_001C	MPUHn_CTRL2 00000000_00000000_00000000_00000000			
0x0000_0020	MPUHn_SADDR2 00000000_00000000_00000000_00000000			
0x0000_0024	MPUHn_EADDR2 00000000_00000000_00000000_01111111			
0x0000_0028	MPUHn_CTRL3 00000000_00000000_00000000_00000000			

Offset	+3	+2	+1	+0
0x0000_002C	MPUHn_SADDR3 00000000_00000000_00000000_00000000			
0x0000_0030	MPUHn_EADDR3 00000000_00000000_00000000_01111111			
0x0000_0034	MPUHn_CTRL4 00000000_00000000_00000000_00000000			
0x0000_0038	MPUHn_SADDR4 00000000_00000000_00000000_00000000			
0x0000_003C	MPUHn_EADDR4 00000000_00000000_00000000_01111111			
0x0000_0040	MPUHn_CTRL5 00000000_00000000_00000000_00000000			
0x0000_0044	MPUHn_SADDR5 00000000_00000000_00000000_00000000			
0x0000_0048	MPUHn_EADDR5 00000000_00000000_00000000_01111111			
0x0000_004C	MPUHn_CTRL6 00000000_00000000_00000000_00000000			
0x0000_0050	MPUHn_SADDR6 00000000_00000000_00000000_00000000			
0x0000_0054	MPUHn_EADDR6 00000000_00000000_00000000_01111111			
0x0000_0058	MPUHn_CTRL7 00000000_00000000_00000000_00000000			
0x0000_005C	MPUHn_SADDR7 00000000_00000000_00000000_00000000			
0x0000_0060	MPUHn_EADDR7 00000000_00000000_00000000_01111111			
0x0000_0064	MPUHn_CTRL8 00000000_00000000_00000000_00000000			
0x0000_0068	MPUHn_SADDR8 00000000_00000000_00000000_00000000			
0x0000_006C	MPUHn_EADDR8 00000000_00000000_00000000_01111111			
0x0000_0070	MPUHn_UNLOCK 00000000_00000000_00000000_00000000			
0x0000_0074	MPUHn_MID 00000000_00001101_00000000_00000000			
0x0000_0078 | 0x0000_010C	Reserved			
0x0000_0110	MPUHn_CTRL9 00000000_00000000_00000000_00000000			

Offset	+3	+2	+1	+0
0x0000_0114	MPUHn_SADDR9 00000000_00000000_00000000_00000000			
0x0000_0118	MPUHn_EADDR9 00000000_00000000_00000000_01111111			
0x0000_011C	MPUHn_CTRL10 00000000_00000000_00000000_00000000			
0x0000_0120	MPUHn_SADDR10 00000000_00000000_00000000_00000000			
0x0000_0124	MPUHn_EADDR10 00000000_00000000_00000000_01111111			
0x0000_0128	MPUHn_CTRL11 00000000_00000000_00000000_00000000			
0x0000_012C	MPUHn_SADDR11 00000000_00000000_00000000_00000000			
0x0000_0130	MPUHn_EADDR11 00000000_00000000_00000000_01111111			
0x0000_0134	MPUHn_CTRL12 00000000_00000000_00000000_00000000			
0x0000_0138	MPUHn_SADDR12 00000000_00000000_00000000_00000000			
0x0000_013C	MPUHn_EADDR12 00000000_00000000_00000000_01111111			
0x0000_0140	MPUHn_CTRL13 00000000_00000000_00000000_00000000			
0x0000_0144	MPUHn_SADDR13 00000000_00000000_00000000_00000000			
0x0000_0148	MPUHn_EADDR13 00000000_00000000_00000000_01111111			
0x0000_014C	MPUHn_CTRL14 00000000_00000000_00000000_00000000			
0x0000_0150	MPUHn_SADDR14 00000000_00000000_00000000_00000000			
0x0000_0154	MPUHn_EADDR14 00000000_00000000_00000000_01111111			
0x0000_0158	MPUHn_CTRL15 00000000_00000000_00000000_00000000			
0x0000_015C	MPUHn_SADDR15 00000000_00000000_00000000_00000000			
0x0000_0160	MPUHn_EADDR15 00000000_00000000_00000000_01111111			
0x0000_0164	MPUHn_CTRL16 00000000_00000000_00000000_00000000			
0x0000_0168	MPUHn_SADDR16 00000000_00000000_00000000_00000000			

Offset	+3	+2	+1	+0
0x0000_016C	MPUHn_EADDR16 00000000_00000000_00000000_01111111			

Figure 3-1 List of Register Mirror Area

Address of Register Mirror Area	Mirrored Peripheral	Mirror Area Behavior	PPU
0000_0400 to 0000_07FF 0000_0800 to 0000_0BFF 0000_0C00 to 0000_0FFF	MPU for DMAC#0	Accessing this region results to the access to MPU for DMAC#0 register area with following offset. Offset: (addr & 0000_03FF)	This area is covered by PPU #66 as well as the mirrored peripheral
0000_1400 to 0000_17FF 0000_1800 to 0000_1BFF 0000_1C00 to 0000_1FFF	MPU for DMAC#1	Accessing this region results to the access to MPU for DMAC#1 register area with following offset. Offset: (addr & 0000_03FF)	This area is covered by PPU #67 as well as the mirrored peripheral

3.1. MPU16 AHB Control Register (MPUHn_CTRL0)

MPU16 AHB Control Register can be used by the software to configure the MPU16 AHB. This register provides enable bit to enable the MPU16 AHB monitoring and protection function. It also provides permission attributes for background region. It also provides controls for enabling or disabling of privileged mode overwrite feature. Lastly it provides status of MPU16 AHB lock bit and Non-Maskable Interrupt flag.

Bit	31	30	29	28	27	26	25	24
Field	Reserved	Reserved	Reserved	Reserved	Reserved	AP[2]	AP[1]	AP[0]
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W	R/W	R/W
Protection Attribute	WPS							
Initial Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	MPUENC	MPUEN
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W	R,WX
Protection Attribute	WPS							
Initial Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved	Reserved	PROT	POEN	Reserved	Reserved	LST
R/W Attribute	R0,WX	R0,WX	R0,WX	R/W	R/W	R/W0	R0,WX	R,WX
Protection Attribute	WPS							
Initial Value	0	0	0	0	0	0	0	1

Bit	7	6	5	4	3	2	1	0
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	NMICL	NMI
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,W	R,WX
Protection Attribute	WPS							
Initial Value	0	0	0	0	0	0	0	0

Table 3-2 MPU16 AHB Control Register (MPUHn_CTRL0) Bits

Bit Position	Bit Field Name	Bit Description
[31:27]	Reserved	-
[26:24]	AP	Access Permissions for Background Region These bits are used to control access permissions for a background region. For more details about these bits, refer to Table 3-1. Note: AP[2:0] bits for background region can only be written when the MPU is disabled (MPUHn_CTRL0:MPUEN = 0).
[23:18]	Reserved	-
[17]	MPUENC	MPU AHB Enable Control 0: MPU AHB Monitoring and Protection Function is disabled 1: MPU AHB Monitoring and Protection Function is enabled

Bit Position	Bit Field Name	Bit Description
[16]	MPUEN	MPU AHB Enable Status 0: MPU AHB Monitoring and Protection Function is disabled. All accesses from the AHB Master Interface are passed on to the AHB memory interface without any protection 1: MPU AHB Monitoring and Protection Function is enabled
[15:13]	Reserved	-
[12]	PROT	Privileged Mode Attribute When the privileged mode overwrite feature is available and POEN = 1, the privilege attribute on AHB memory interface is controlled by this bit. 0: Non-privileged mode 1: Privileged mode
[11]	POEN	Privileged Mode Overwrite Feature Enable 0: Privileged mode overwrite feature is disabled 1: Privileged mode overwrite feature is enabled
[10]	Reserved	Reserved. Always write "0".
[9]	Reserved	-
[8]	LST	MPU Lock Status 0: MPU AHB is unlocked; registers in MPU AHB can be written 1: MPU AHB is locked; no registers (other than MPUHn_UNLOCK register) in MPU AHB can be written
[7:2]	Reserved	-
[1]	NMICL	NMI Interrupt Clear 0: No effect 1: Clears the NMI interrupt flag Read returns "0".
[0]	NMI	NMI Interrupt Flag This interrupt flag indicates that the memory protection violation was detected for an AHB transfer.

Note:

- *The register bits MPUHn_CTRL0:AP[2:0], MPUHn_CTRL0:POEN and MPUHn_CTRL0:PROT can only be written when the MPU is disabled (MPUHn_CTRL0:MPUEN = 0).*

3.2. MPU16 AHB NMI Enable Register (MPUHn_NMIEN)

MPU16 AHB NMI Enable Register can be used by software to reset the NMI enable bit. The default value of the NMI enable bit is 1. This bit can be reset by software only once after a reset operation.

Bit	31	30	29	28	27	26	25	24
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
Protection Attribute	WPS							
Initial Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
Protection Attribute	WPS							
Initial Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
Protection Attribute	WPS							
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	NMIEN
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W
Protection Attribute	WPS							
Initial Value	0	0	0	0	0	0	0	1

Table 3-3 MPU16 AHB NMI Enable Register (MPUHn_NMIEN) Bits

Bit Position	Bit Field Name	Bit Description
[31:8]	Reserved	-
[7:1]	Reserved	-
[0]	NMIEN	NMI Interrupt Enable This bit decides whether the NMI interrupt flag is routed to an NMI interrupt signal or not. 0: NMI interrupt flag does not trigger an NMI interrupt signal 1: NMI interrupt flag triggers an NMI interrupt signal The value of this bit can be changed only once after reset.

3.3. MPU16 AHB Memory Error Control Register (MPUHn_MERRC)

This is a read-only register that provides the control information of AHB transfer for which memory protection violation was detected. This register can be read to get the privileged mode and transfer mode (write/read) information of the AHB transfer.

Bit	31	30	29	28	27	26	25	24
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	HPROT	HWRITE
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Table 3-4 MPU16 AHB Memory Error Control Register (MPUHn_MERRC) Bits

Bit Position	Bit Field Name	Bit Description
[31:8]	Reserved	-
[7:2]	Reserved	-
[1]	HPROT	AHB Transfer Privileged Mode. This bit provides the status of the HPROT signal of the AHB transfer for which memory protection violation is detected.
[0]	HWRITE	AHB Transfer Mode. This bit provides the status of the HWRITE signal of the AHB transfer for which memory protection violation is detected.

Note:

- *This register is read-only. Write to this register will return a bus error.*

3.4. MPU AHB Memory Error Address Register (MPUHn_MERRA)

This is a read-only register that provides the address of AHB transfer for which memory protection violation was detected.

Bit	31	30	29	28	27	26	25	24
Field	HADDR[31]	HADDR[30]	HADDR[29]	HADDR[28]	HADDR[27]	HADDR[26]	HADDR[25]	HADDR[24]
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	HADDR[23]	HADDR[22]	HADDR[21]	HADDR[20]	HADDR[19]	HADDR[18]	HADDR[17]	HADDR[16]
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	HADDR[15]	HADDR[14]	HADDR[13]	HADDR[12]	HADDR[11]	HADDR[10]	HADDR[9]	HADDR[8]
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	HADDR[7]	HADDR[6]	HADDR[5]	HADDR[4]	HADDR[3]	HADDR[2]	HADDR[1]	HADDR[0]
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Table 3-5 MPU16 AHB Memory Error Address Register (MPUHn_MERRA) Bits

Bit Position	Bit Field Name	Bit Description
[31:0]	HADDR	AHB Address. The address of an AHB transfer for which memory protection violation was detected.

Note:

- *This register is read-only. Write to this register will return a bus error.*

3.5. MPU16 AHB Region Control Registers (MPUHn_CTRL1 to 16)

MPU16 AHB Region Control Register is used to specify access permission for a particular region. Software can also use this register for enabling or disabling the particular region.

MPUHn_CTRL1 Control Register for Region 1 is explained below.

Bit	31	30	29	28	27	26	25	24
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
Protection Attribute	WPS							
Initial Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
Protection Attribute	WPS							
Initial Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved	Reserved	Reserved	Reserved	AP[2]	AP[1]	AP[0]
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W	R/W	R/W
Protection Attribute	WPS							
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	MPUENC	MPUEN
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W	R,WX
Protection Attribute	WPS							
Initial Value	0	0	0	0	0	0	0	0

Table 3-6 MPU16 AHB Region Control Register for Region 1 (MPUHn_CTRL1) Bits

Bit Position	Bit Field Name	Bit Description
[31:16]	Reserved	-
[15:11]	Reserved	-
[10:8]	AP	Access Permissions These bits are used to control access permissions for region 1. For a detailed description of these bits refer to Table 3-1.
[7:2]	Reserved	-
[1]	MPUENC	Enable Control 0: Memory Protection for region 1 is disabled 1: Memory Protection for region 1 is enabled The enable status of the region can be read through the MPUHn_CTRL1:MPUEN bit.

Bit Position	Bit Field Name	Bit Description
[0]	MPUEN	Enable Status 0: Memory Protection for region 1 is disabled 1: Memory Protection for region 1 is enabled This is a read-only bit; writing to this bit has no effect.

Note:

- *For all 16 MPU16 AHB Region Control registers, the Access Permission bits (i.e. MPUHn_CTRL1 to 16:AP) can only be written when the corresponding region is disabled (i.e. MPUHn_CTRL1 to 16:MPUEN = 0) or the MPU is disabled (MPUHn_CTRL0:MPUEN = 0). This also implies that when MPUHn_CTRL0:MPUEN = 1 and MPUHn_CTRL1 to 16:MPUEN = 1:*
  - 32- or 16-bit access to this register is not allowed
  - 8-bit access is required to disable the MPU

3.6. MPU16 AHB Start Address Registers (MPUHn_SADDR1 to 16)

Each region in MPU16 AHB can be defined by specifying the start address and end address for that particular region. The MPUHn_SADDR1 to 16 registers are used to specify the start address for the 16 regions. The start address indicates the first address for that region. MPUHn_SADDR1 is the Start Address Register for Region 1 and is explained below.

Bit	31	30	29	28	27	26	25	24
Field	SADDR[31]	SADDR[30]	SADDR[29]	SADDR[28]	SADDR[27]	SADDR[26]	SADDR[25]	SADDR[24]
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	WPS							
Initial Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	SADDR[23]	SADDR[22]	SADDR[21]	SADDR[20]	SADDR[19]	SADDR[18]	SADDR[17]	SADDR[16]
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	WPS							
Initial Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	SADDR[15]	SADDR[14]	SADDR[13]	SADDR[12]	SADDR[11]	SADDR[10]	SADDR[9]	SADDR[8]
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	WPS							
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	SADDR[7]	SADDR[6]	SADDR[5]	SADDR[4]	SADDR[3]	SADDR[2]	SADDR[1]	SADDR[0]
R/W Attribute	R/W	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
Protection Attribute	WPS							
Initial Value	0	0	0	0	0	0	0	0

Table 3-7 MPU16 AHB Start Address Register for Region 1 (MPUHn_SADDR1) Bits

Bit Position	Bit Field Name	Bit Description
[31:0]	SADDR	Start Address Start address for region 1

Note:

- *the MPUHn_SADDR1 to 16 registers can only be written when the corresponding region is disabled (MPUHn_CTRL1 to 16:MPUEN = 0) or the MPU is disabled (MPUHn_CTRL0:MPUEN = 0).*

3.7. MPU16 AHB End Address Registers (MPUHn_EADDR1 to 16)

Each region in MPU16 AHB can be defined by specifying a start address and end address for that particular region. MPUHn_EADDR1 to 16 registers are used to specify the end addresses for the 16 regions. The end address indicates the last address for that region. The MPUHn_EADDR1 End Address Register for Region 1 is explained below.

Bit	31	30	29	28	27	26	25	24
Field	EADDR[31]	EADDR[30]	EADDR[29]	EADDR[28]	EADDR[27]	EADDR[26]	EADDR[25]	EADDR[24]
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	WPS							
Initial Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	EADDR[23]	EADDR[22]	EADDR[21]	EADDR[20]	EADDR[19]	EADDR[18]	EADDR[17]	EADDR[16]
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	WPS							
Initial Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	EADDR[15]	EADDR[14]	EADDR[13]	EADDR[12]	EADDR[11]	EADDR[10]	EADDR[9]	EADDR[8]
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	WPS							
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	EADDR[7]	EADDR[6]	EADDR[5]	EADDR[4]	EADDR[3]	EADDR[2]	EADDR[1]	EADDR[0]
R/W Attribute	R/W	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX
Protection Attribute	WPS							
Initial Value	0	1	1	1	1	1	1	1

Table 3-8 MPU16 AHB End Address Register for Region 1 (MPUHn_EADDR1) Bits

Bit Position	Bit Field Name	Bit Description
[31:0]	EADDR	End Address End address for region 1

Note:

- *MPUHn_EADDR1 to 16 registers can only be written when the corresponding region is disabled (MPUHn_CTRL1 to 16:MPUEN = 0) or the MPU is disabled (MPUHn_CTRL0:MPUEN = 0).*

3.8. MPU16 AHB Unlock Register (MPUHn_UNLOCK)

The software can use this register to lock or unlock the MPU16 AHB registers for write access.

Bit	31	30	29	28	27	26	25	24
Field	UNLOCK[31]	UNLOCK[30]	UNLOCK[29]	UNLOCK[28]	UNLOCK[27]	UNLOCK[26]	UNLOCK[25]	UNLOCK[24]
R/W Attribute	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
Protection Attribute	WP							
Initial Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	UNLOCK[23]	UNLOCK[22]	UNLOCK[21]	UNLOCK[20]	UNLOCK[19]	UNLOCK[18]	UNLOCK[17]	UNLOCK[16]
R/W Attribute	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
Protection Attribute	WP							
Initial Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	UNLOCK[15]	UNLOCK[14]	UNLOCK[13]	UNLOCK[12]	UNLOCK[11]	UNLOCK[10]	UNLOCK[9]	UNLOCK[8]
R/W Attribute	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
Protection Attribute	WP							
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	UNLOCK[7]	UNLOCK[6]	UNLOCK[5]	UNLOCK[4]	UNLOCK[3]	UNLOCK[2]	UNLOCK[1]	UNLOCK[0]
R/W Attribute	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
Protection Attribute	WP							
Initial Value	0	0	0	0	0	0	0	0

Table 3-9 MPU16 AHB Unlock Register (MPUHn_UNLOCK) Bits

Bit Position	Bit Field Name	Bit Description
[31:0]	UNLOCK	MPU16 AHB Unlock The MPU16 AHB Unlock Register protects the MPU16 AHB module from being modified accidently by software. The MPU16 AHB registers cannot be written until this register has been written with a specific unlock value. The correct value for unlocking can be written only in privileged mode. Reading this register always returns "0". To lock the MPU16 AHB again software must write another value specific to lock. A write access to the MPU16 AHB registers without unlocking or writing value other than the lock or unlock value to this register causes a protection error. Unlock value: 0xACCABB56 Lock value: 0x112ABB56 Notes: 1. This register cannot be written by an 8- or 16-bit write access; any such access causes a protection error.

3.9. MPU16 AHB Module ID Register (MPUHn_MID)

This is a read-only register which returns a meaningless constant.

Bit	31	30	29	28	27	26	25	24
Field	MID[31]	MID[30]	MID[29]	MID[28]	MID[27]	MID[26]	MID[25]	MID[24]
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	MID[23]	MID[22]	MID[21]	MID[20]	MID[19]	MID[18]	MID[17]	MID[16]
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R1,WX	R1,WX	R0,WX	R1,WX
Protection Attribute	-							
Initial Value	0	0	0	0	1	1	0	1

Bit	15	14	13	12	11	10	9	8
Field	MID[15]	MID[14]	MID[13]	MID[12]	MID[11]	MID[10]	MID[9]	MID[8]
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	MID[7]	MID[6]	MID[5]	MID[4]	MID[3]	MID[2]	MID[1]	MID[0]
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Table 3-10 MPU16 AHB Module ID Register (MPUHn_MID) Bits

Bit Position	Bit Field Name	Bit Description
[31:0]	MID	Return a meaningless constant.

Note:

– *This register is read-only. Write to this register will return a bus error.*

4. Operation of the MPU16 AHB

This section describes the operation of the MPU16 AHB.

MPU16 AHB Region Granularity and Priority Decision

The MPU16 AHB supports up to 16 regions and provides the start and end addresses for each of these 16 regions. MPU16 AHB regions are defined with a granularity of 128 bytes.

Each region is identified as Region 1, Region 2, Region 3 and so on up to Region 16, which is given highest priority. Region 15 has second highest priority and so on down to Region 1, which has the second lowest priority. The Background Region is designated the lowest priority region.

The Start Address specifies the first address of the region and is specified by registers MPUHn_SADDR1 to MPUHn_SADDR16 for region 1 to region 16 respectively. Since the region granularity is 128bytes, the least significant 7 bits of the start addresses will read 0.

End Addresses are specified by registers MPUHn_EADDR1 to MPUHn_EADDR16 for region 1 to region 16 respectively. The least significant 7 bits of the End Address registers are read only bits and will always read 1. This ensures the granularity of 128 bytes.

In the MPU16 AHB, regions may overlap each other, as shown in Figure 3-1. Hence it is also possible that an AHB address for any transfer may match with multiple regions. Whenever the AHB Transfer Address matches with multiple regions, the permissions corresponding to the highest priority region (among all matching regions) is applied.

AHB Burst Monitoring

The MPU16 AHB only supports Single Transfer Monitoring and not Burst Monitoring.

Before starting a burst transfer, MPU16_AHB calculates address range which is accessed by the burst transfer.

When the burst transfer violates the protection, the MPU16_AHB blocks entire of the burst transfer before MPU16_AHB starting the transfer.

Priority Decision of MPU

MPU16 AHB offers the starting address and the ending address of 16 each region.

The region is permitted to overlap mutually in MPU16 AHB.

Therefore, the AHB address of an arbitrary forwarding might be corresponding to two or more regions.

MPU supports 16 regions or less, and each region is identified as region 1, region 2, and region 3…. region 16.

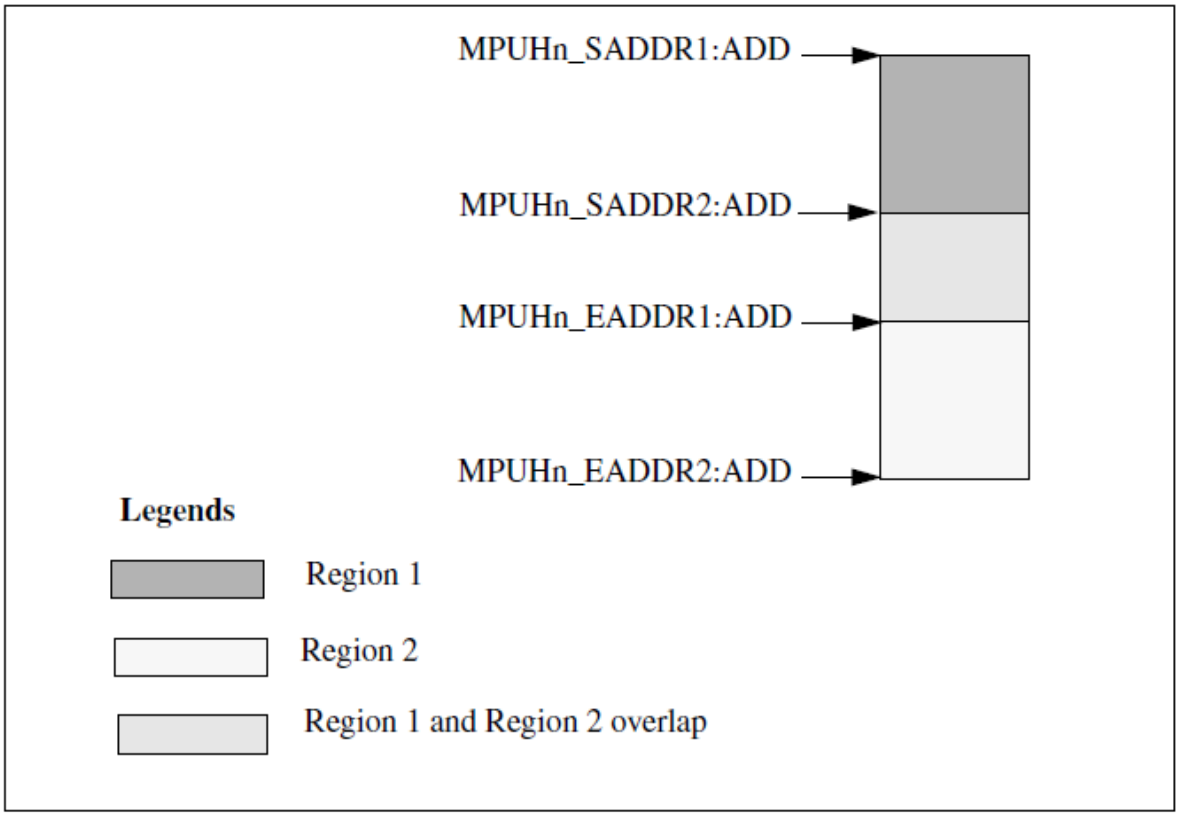
The priority of region 16 is the highest, and the priority of region 15 is high in the second.

The priority of the backing ground region is lowest most, and the priority of region 1 does secondarily by being low.

When the address of the AHB forwarding is corresponding to two or more regions, the permission attribute corresponding to the region with the highest priority in a corresponding region is applied.

The example of the overlap in region 1 and region 2 with Figure 3-1 is shown.

Figure 4-1 Example Region Overlapping



MPU Access Permissions

Region Control registers in the MPU16 AHB (i.e. MPUHn_CTRL1 to MPUHn_CTRL16) are used to control the access permission to Regions 1 to 16. Also MPUHn_CTRL0 is used to control the access permission for the Background Region.

Table 3-1 describes Access Permission bits (AP) in these registers and corresponding permission attributes.

Table 4-1 Access Permissions

AP Bits	Access in Privileged Mode	Access in Non-Privileged Mode	Comment
000 (default)	No access	No access	All bus accesses are blocked and hence would generate a memory protection violation.
001	Read, write	No access	Reads and writes are permitted in privileged mode only. Any access in non-privileged mode would generate a memory protection violation.
010	Read, write	Read only	Reads and writes are permitted in privileged mode. Only reads are permitted in non-privileged mode. Writes in non-privileged mode would generate a memory protection violation.
011	Read, write	Read, write	All bus transfers are permitted. No memory protection violation is generated in this mode.
100	No access	No access	All bus accesses are blocked and hence generate a memory protection violation.
101	Read only	No access	Reads are permitted in privileged mode only. Writes in privileged mode and any access in non-privileged mode would generate a memory protection violation.
110	Read only	Read only	Reads are permitted in privileged as well as non-privileged mode. Any write access would generate a memory protection violation.
111	Read, write	Read, write	All bus transfers are permitted. No memory protection violation is generated in this mode.

Bus Monitor and Protection Logic

All transfers on the AHB Master Interface are monitored and checked for permitted access.

- Bus monitor and protection logic within the MPU16 AHB compares the address of the current transfer with the start and end addresses of each region to find a region match, i.e. where the current transfer matches one of the 16 defined regions.

- As explained in Section “MPU16 AHB region granularity and priority decision”, the AHB transfer address may match multiple regions where the permission attributes of the region with highest priority are checked against the attributes of the currently applied transfer from the AHB master.
- If the attributes of the currently applied transfer are within the permitted attributes, the current transaction is passed on to the AHB memory interface.
- If the attributes are not within permitted attributes, the current transfer is blocked. The Non-Maskable Interrupt flag (MPUHn_CTRL0:NMI) is set. The address and control information of the current transfer is stored in MPUHn_MERRA and MPUHn_MERRC respectively.
- All further transfers are blocked until the MPUHn_CTRL0:NMI flag is cleared by software. Further monitoring of the AHB transfer addresses is also stalled until the MPUHn_CTRL0:NMI flag is cleared.

When a transfer is blocked, the MPU16 AHB performs the following actions:

- It drives idle transfers on the AHB memory interface
- It generates an error response on the AHB master interface

Privileged Mode Overwrite Feature

The MPU16 AHB supports privileged mode overwrite feature.

When this mode is enabled, the privileged mode attribute on the AHB memory interface is set by setting the MPUHn_CTRL0:PROT bit as explained in Section 2.1.

Note:

- *Bus monitor and protection logic for detecting memory protection violation uses the privileged mode attribute on the AHB master interface and not the MPUHn_CTRL0:PROT bit, even when the privileged mode overwrite feature is enabled.*

CHAPTER 30: CAN FD Controller(MCAN3.0.1)

This chapter explains the functions and operations of the CAN FD Controller.

1. Overview
2. Configuration
3. Explanation of Operations
4. Setup Procedure Examples
5. Registers
6. Message RAM

1. Overview

The CAN FD Controller performs communication according to ISO11898-1 (CAN specification Rev. 2.0 part A, B) and to the CAN FD specification V1.0.

All functions concerning the handling of messages are implemented by the Rx Handler and the Tx Handler. The Rx Handler manages message acceptance filtering, the transfer of received messages from the CAN Core to a Message RAM as well as providing receive message status information. The Tx Handler is responsible for the transfer of transmit messages from the Message RAM to the CAN Core as well as providing transmit status information.

Acceptance filtering is implemented by a combination of up to 192 filter elements where each one can be configured as a range, as a bit mask, or as a dedicated ID filter.

The CAN FD Controller is accessible by the CPU using a data width of 8/16/32-bits. The CAN FD Controller's clock domain concept allows the separation between the two input clocks, the CAN clock and the Bus clock.

Features of the CAN FD Controller

- Conforms with ISO11898-1 (CAN specification Rev. 2.0 part A, B) and the CAN FD specification V1.0
- CAN FD with up to 64 data bytes supported
- CAN Error Logging
- Acceptance filtering
- Two configurable Receive FIFOs
- Separate signaling on reception of High Priority Messages
- Up to 64 dedicated Receive Buffers
- Up to 32 dedicated Transmit Buffers
- Configurable Transmit FIFO
- Configurable Transmit Queue
- Configurable Transmit Event FIFO
- Direct Message RAM access for CPU
- Programmable loop-back test mode
- Maskable interrupts
- Two clock domains (CAN clock and Bus clock)
- Power-down support
- Debug on CAN support

Note:

- *Debug on CAN feature is not supported for TRAVEO™ T1G Platform.*

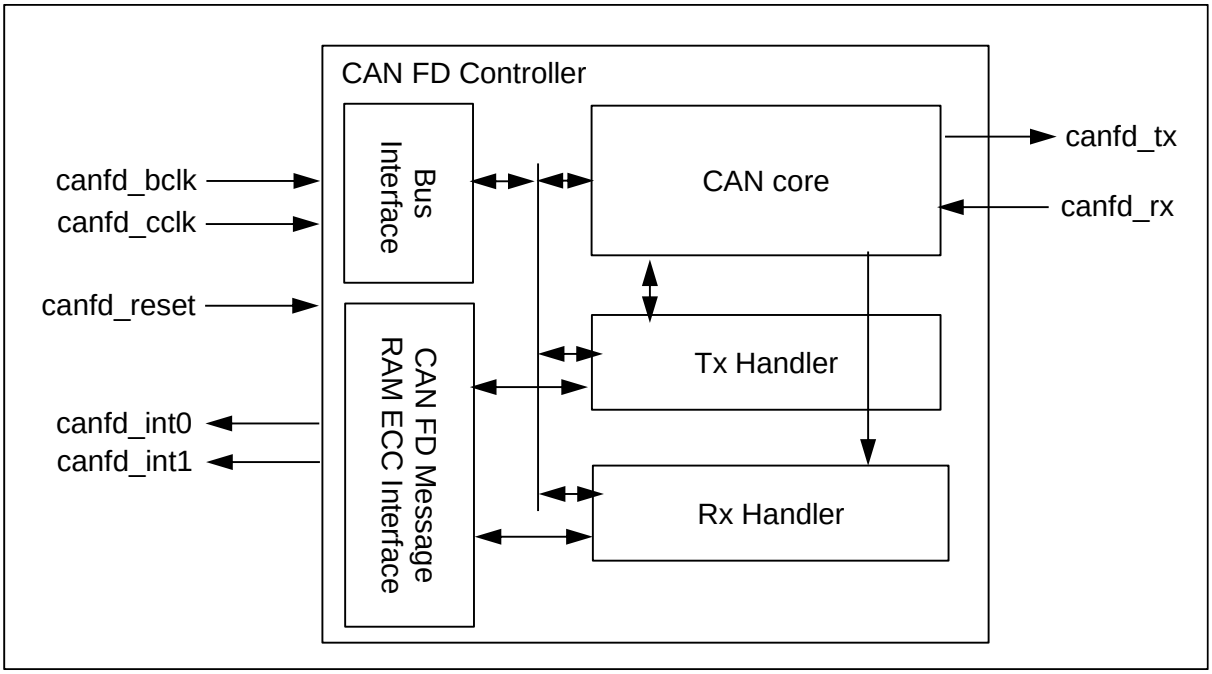
2. Configuration

This section provides configuration related information about the CAN FD controller.

2.1. Block Diagram
2.2. Dual Clock Sources
2.3. Dual Interrupt Lines

2.1. Block Diagram

Figure 2-1 CAN FD Controller Block Diagram



CAN Core

CAN Protocol Controller. Handles all ISO 11898-1 protocol functions. Supports 11-bit and 29-bit identifiers.

CAN FD Message RAM ECC Interface

Interface with the CAN FD Message RAM ECC module. This module contains

- a Message RAM that stores messages to be transmitted, received messages, acceptance filter elements, and Tx Events, and
- ECC (Error Correction Code) logic for the Message RAM.

Bus Interface
CPU interface bus with a data width of 8/16/32-bits.

Tx Handler
Controls the message transfer from the Message RAM to the CAN Core. A maximum of 32 Tx Buffers can be configured for transmission. Tx buffers can be used as dedicated Tx Buffers, as Tx FIFO, part of a Tx Queue, or as a combination of them. A Tx Event FIFO stores Tx timestamps together with the corresponding Message ID. Transmit cancellation is also supported.

Rx Handler
Controls the transfer of received messages from the CAN Core to the Message RAM. The Rx Handler supports two Receive FIFOs, each of configurable size, and up to 64 dedicated Rx Buffers for storage of all messages that have passed acceptance filtering. A dedicated Rx Buffer, in contrast to a Receive FIFO, is used to store only messages with a specific identifier. An Rx timestamp is stored together with each message. Up to 128 filters can be defined for 11-bit IDs and up to 64 filters for 29-bit IDs.

2.2. Dual Clock Sources

The two clocks of the CAN FD Controller must be configured to meet the following requirement in order to guarantee proper operation:

Bus clock (canfd_bclk) frequency > CAN clock (canfd_cclk) frequency

2.3. Dual Interrupt Lines

The CAN FD Controller provides two interrupt lines. Interrupts can be routed either to canfd_int0 or to canfd_int1. By default all interrupts are routed to interrupt line canfd_int0. By programming bits Enable Interrupt Line 0 (ILE.EINT0) and Enable Interrupt Line 1 (ILE.EINT1) the interrupt lines can be enabled or disabled separately.

3. Explanation of Operations

This section explains the operations of the CAN FD Controller.

3.1.Operating Modes
3.2.Timestamp Generation
3.3.Timeout Counter
3.4.Rx Handling
3.5.Tx Handling
3.6.FIFO Acknowledge Handling
3.7. Configuring the CAN Bit Timing

3.1. Operating Modes

This section explains the operating modes of the CAN FD Controller.

3.1.1.Software Initialization
3.1.2.Normal Operation
3.1.3.CAN FD Operation
3.1.4.Transceiver Delay Compensation
3.1.5.Restricted Operation Mode
3.1.6.Bus Monitoring Mode
3.1.7.Disabled Automatic Retransmission
3.1.8.Power Down (Sleep Mode)
3.1.9.Test Modes

3.1.1. Software Initialization

Setting/Resetting the Initialization Bit (CCCR.INIT)

Software initialization is started by setting bit Initialization (CCCR.INIT),

- either by software or by a hardware reset,
- when an uncorrected bit error was detected in the Message RAM,
- or by going Bus_Off.

While CCCR.INIT is set,

- message transfer from and to the CAN bus is stopped,
- the status of the CAN bus output canfd_tx is recessive,
- the protocol error counters are unchanged.

Setting CCCR.INIT does not change any configuration register.

Resetting CCCR.INIT finishes the software initialization. Afterwards the CAN FD Controller synchronizes itself to the data transfer on the CAN bus by waiting for the occurrence of a sequence of

11 consecutive recessive bits (≡ Bus_Idle) before it can take part in bus activities and start the message transfer.

Notes:

- *In case of setting CCCR.INIT to "1" during Normal Operation, set CCCR.INIT to "1" by issuing a Clock Stop Request via CCCR.CSR = 1. Before resetting CCCR.INIT first reset CCCR.CSR.*
- *When halting transmissions or receptions via CCCR.INIT = 1, cancel any pending transmissions (Tx Buffer Request Pending Register TXBRP ≠ 0x00000000) by setting Tx Buffer Cancellation Request Register TXBCR to 0xFFFF_FFFF.*

Access/Set/Reset Properties of Registers Affected by Configuration Change Enable (CCCR.CCE)

Access to the configuration registers is only enabled when both bits CCCR.INIT and Configuration Change Enable (CCCR.CCE) are set (protected write). CCCR.CCE can only be set/reset while CCCR.INIT = 1. CCCR.CCE is automatically reset when CCCR.INIT is reset.

The following registers are reset when CCCR.CCE is set

- HPMS - High Priority Message Status
- RXF0S - Rx FIFO 0 Status
- RXF1S - Rx FIFO 1 Status
- TXFQS - Tx FIFO/Queue Status
- TXBRP - Tx Buffer Request Pending
- TXBTO - Tx Buffer Transmission Occurred
- TXBCF - Tx Buffer Cancellation Finished
- TXEFS - Tx Event FIFO Status

In addition

- the Timeout Counter value (TOCV.TOC[15:0]) is preset to the value configured by the Timeout Period (TOCC.TOP[15:0]) when CCCR.CCE is set.
- the state machines of the Tx Handler and Rx Handler are held in idle state while CCCR.CCE = 1.

The following registers are only writeable while CCCR.CCE = 0.

- TXBAR - Tx Buffer Add Request
- TXBCR - Tx Buffer Cancellation Request

Test Mode Enable (CCCR.TEST) and Bus Monitoring Mode (CCCR.MON) can only be set by the CPU while CCCR.INIT = 1 and CCCR.CCE = 1. Both bits may be reset at any time.

Disable Automatic Retransmission (CCCR.DAR) can only be set/reset while CCCR.INIT = 1 and CCCR.CCE = 1.

Note:

- *To set CCCR.INIT and CCCR.CCE to "1", the CPU must follow the procedure below. In case of setting CCCR.CCE to "1" just after hardware reset, it isn't necessary to follow the procedure.*

  1. *Cancel all pending transmission requests by setting Tx Buffer Cancellation Request Register TXBCR to 0xFFFF_FFFF (the Tx Buffer Request Pending Register TXBRP will be reset to "0")*

2. *Issue a clock stop request by setting Clock Stop Request (CCCR.CSR) to "1"*
3. *Wait till both CCCR.INIT and Clock Stop Acknowledge (CCCR.CSA) are "1"*
4. *First reset CCCR.CSR*
5. *Then reset CCCR.INIT*
6. *Wait till CCCR.INIT is "0"*
7. *Issue a second clock stop request by setting CCCR.CSR to "1"*
8. *Wait till both CCCR.INIT and CCCR.CSA are "1"*
9. *Set CCCR.CCE to "1" and reset CCCR.CSR*

Message RAM Initialization

The Message RAM should be zeroized before configuration of the CAN FD Controller in order to prevent Message RAM bit errors when reading uninitialized words, and also to avoid unexpected filter element configurations in the Message RAM.

3.1.2. Normal Operation

Once the CAN FD Controller is initialized and Initialization (CCCR.INIT) is reset to "0", the CAN FD Controller synchronizes itself to the CAN bus and is ready for communication.

After passing the acceptance filtering, received messages including Message ID and DLC (Data Length Code) are stored into a dedicated Rx Buffer or into Rx FIFO 0 or Rx FIFO 1.

For messages to be transmitted, dedicated Tx Buffers and/or a Tx FIFO or a Tx Queue can be initialized or updated. Automated transmission on reception of remote frames is not implemented.

3.1.3. CAN FD Operation

There are two variants in the CAN FD frame format,

- the CAN FD frame without bit rate switching where the data field of a CAN frame may be longer than 8 bytes.
- the CAN FD frame where control field, data field, and CRC field of a CAN frame are transmitted with a higher bit rate than the beginning and the end of the frame.

Enabling the CAN Operation Mode via CAN Mode Enable (CCCR.CME[1:0])

The CAN operation mode is enabled by programming CAN Mode Enable (CCCR.CME[1:0]). The transmission properties of the enabled CAN Mode can be changed via CAN Mode Request (CCCR.CMR[1:0]. See next paragraph for details).

- In case CCCR.CME[1:0] = 0b00, transmission and reception of CAN frames according to ISO 11898-1 is enabled.
- In case CCCR.CME[1:0] = 0b01, transmission of long CAN FD frames and reception of long and fast CAN FD frames is enabled.
- With CCCR.CME[1:0] = 0b10/0b11, transmission and reception of long and fast CAN FD frames is enabled.

CCCR.CME[1:0] can only be changed while Initialization (CCCR.INIT) and Configuration Change Enable (CCCR.CCE) are both set.

Changing CAN Operation Modes via CAN Mode Request (CCCR.CMR[1:0])

When initialization is left (CCCR.INIT set to "0"), a mode change to the CAN FD protocol option has to be requested by writing to CAN Mode Request (CCCR.CMR[1:0]).

A mode change requested by writing to CCCR.CMR[1:0] is allowed only when no transmission requests are pending (Tx Buffer Request Pending Register TXBRP = 0x00000000). If necessary, cancel pending transmission requests before changing the CAN operation mode. When the CAN FD Controller reaches idle phase after such mode change request, CCCR.CMR[1:0] is reset to 0b00 and the status flags CAN FD Bit Rate Switching (CCCR.FDBS) and CAN FD Operation (CCCR.FDO) are set accordingly. In case the requested CAN operation mode is not enabled, the value written to CCCR.CMR[1:0] is retained until it is overwritten by the next mode change request. Default is CAN operation according to ISO11898-1.

It is not necessary to change the CAN operation mode after system startup. A mode change during CAN operation is only recommended under the following conditions:

- The failure rate in the CAN FD data phase is significantly higher than in the CAN FD arbitration phase. In this case disable the CAN FD bit rate switching option for transmissions.
- During system startup all nodes are transmitting according to ISO11898-1 until it is verified that they are able to communicate in CAN FD format. If this is true, all nodes switch to CAN FD operation.
- End-of-line programming in case not all nodes are CAN FD capable. Non CAN FD nodes are held in Bus Monitoring mode until programming has completed. Then all nodes switch back to CAN communication according ISO11898-1.
- Wake-up messages in CAN Partial Networking have to be transmitted in CAN format.

Interpretation of Received Frames

When CCCR.CME[1:0] ≠ 0b00, received CAN FD frames are interpreted according to the CAN FD Specification. The reserved bit in CAN frames with 11-bit identifiers and the first reserved bit in CAN frames with 29-bit identifiers will be decoded as EDL bit. EDL = recessive signifies a CAN FD frame, EDL = dominant signifies a standard CAN frame. In a CAN FD frame, the two bits following EDL, r0 and BRS, decide whether the bit rate inside of this CAN FD frame is switched. A CAN FD bit rate switch is signified by r0 = dominant and BRS = recessive. The coding of r0 = recessive in a CAN FD frame is used to detect Protocol Exception Events (see "3.4.5. Protocol Exception Event").

Reception of CAN frames according to ISO 11898-1 is possible in all CAN operation modes.

Format of Transmitted Frames

The status bits CAN FD Operation (CCCR.FDO) and CAN FD Bit Rate Switching (CCCR.FDBS) indicate the format of transmitted frames.

- When CCCR.FDO is set, frames will be transmitted in CAN FD format with EDL = recessive.
- When both CCCR.FDO and CCCR.FDBS are set, frames will be transmitted in CAN FD format with bit rate switching and both bits EDL and BRS = recessive.

The DLC Field

In the CAN FD format, the coding of the DLC differs from the CAN format. The DLC codes 0 to 8 have the same coding as in CAN, the codes 9 to 15, which in CAN all code a data field of 8 bytes, are coded according to Table 3-1 below.

Table 3-1 Coding of DLC in CAN FD

DLC	9	10	11	12	13	14	15
Number of Data Bytes	12	16	20	24	32	48	64

Bit Rate Switching

In CAN FD frames, the bit timing will be switched inside the frame, after the BRS (Bit Rate Switch) bit, if this bit is recessive. Before the BRS bit, in the CAN FD arbitration phase, the nominal bit timing is used as defined by the Bit Timing & Prescaler Register (BTP). In the following CAN FD data phase, the fast bit timing is used as defined by the Fast Bit Timing & Prescaler Register (FBTP). The bit timing is switched back from the fast timing at the CRC delimiter or when an error is detected, whichever occurs first.

The maximum configurable bit rate in the CAN FD data phase depends on the CAN clock frequency (canfd_cclk).

Example:

- *With a CAN clock frequency of 20MHz and the shortest configurable bit time of 4 tq, the bit rate in the data phase is 5 Mbit/s.*

The Error Status Indicator ESI

In both data frame formats, CAN FD long and CAN FD fast, the value of the bit ESI (Error Status Indicator) is determined by the transmitter's error state at the start of the transmission. If the transmitter is error passive, ESI is transmitted recessive, else it is transmitted dominant.

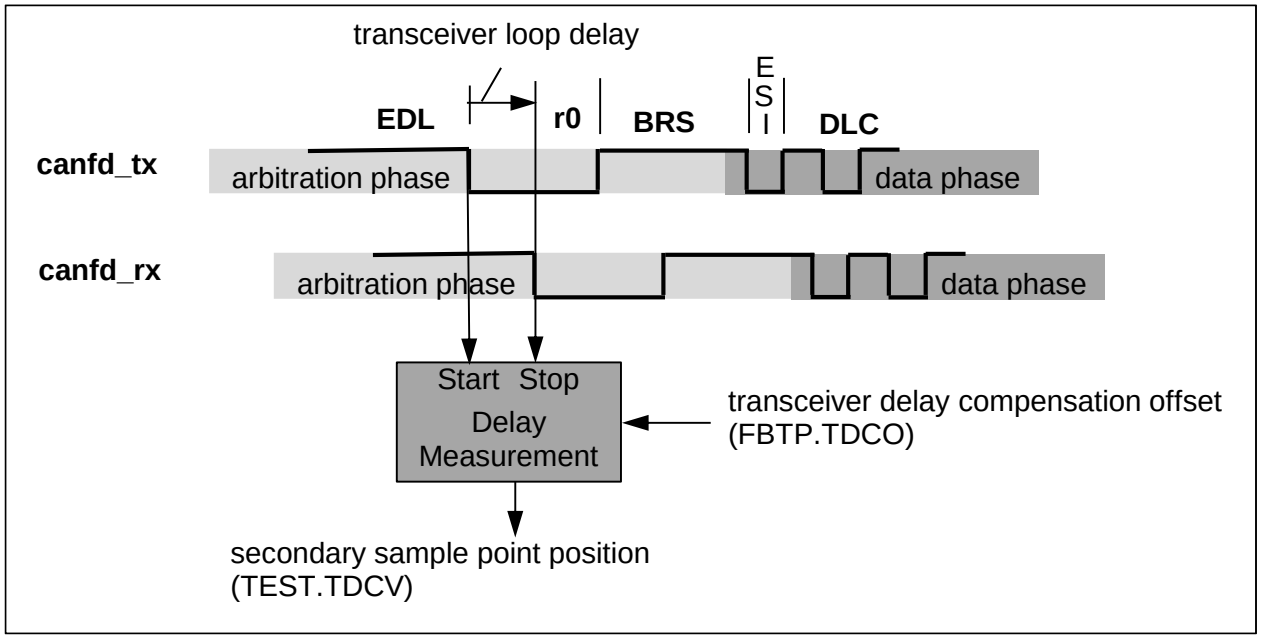
3.1.4. Transceiver Delay Compensation

During the data phase of a CAN FD transmission only one node is transmitting, all others are receivers. The length of the bus line has no impact. When transmitting via pin canfd_tx the protocol controller receives the transmitted data from its local CAN transceiver via pin canfd_rx. The received data is delayed by the CAN transceiver's loop delay. In case this delay is greater than the time segment before the sample point of a data phase bit time, a bit error is detected. In order to enable a data phase bit time (fast bit time) that is even shorter than the transceiver loop delay, the delay compensation is introduced. Without transceiver delay compensation, the bit rate in the data phase of a CAN FD frame is limited by the transceiver's loop delay.

3.1.4.1 Description

The CAN FD protocol unit has implemented a delay compensation mechanism to compensate the CAN transceiver's loop delay, thereby enabling transmission with higher bit rates during the CAN FD data phase independent of the delay of a specific CAN transceiver. Figure 3-1 below describes how the transceiver loop delay is measured.

Figure 3-1 Transceiver Delay Measurement



Within each CAN FD frame, the transmitter measures the delay between the data transmitted at pin canfd_tx and the data received at pin canfd_rx. The measurement is done once, at the falling edge of bit EDL to bit r0. The delay is measured in canfd_cclk periods.

A secondary sample point position is calculated by adding a configurable transceiver delay compensation offset (FBTP.TDCO[4:0]) to the measured transceiver delay. This transceiver delay compensation value (TEST.TDCV[5:0]) is the sum of the measured transceiver delay and the transceiver delay compensation offset. The transceiver delay compensation offset is chosen to adjust

the secondary sample point inside the bit time (e.g. half of the bit time in the data phase). The position of the secondary sample point is rounded down to the next integer number of time quanta tq.

To check for bit errors during the data phase, the delayed transmit data is compared against the received data at the secondary sample point. If a bit error is detected at the secondary sample point, the transmitter will react to this bit error at the next following regular sample point. During arbitration phase the delay compensation is always disabled.

For the transceiver delay compensation, the sum of the measured delay from canfd_tx to canfd_rx and the configured transceiver delay compensation offset (FBTP.TDCO[4:0]) must meet both of the following boundary conditions:

- has to be less than 3 bit times in the data phase.
- has to be less or equal 63 canfd_cclk periods. In case this sum exceeds 63 canfd_cclk periods, the maximum value of 63 canfd_cclk periods is used for transceiver delay compensation.

The actual delay compensation value is monitored by reading the Transceiver Delay Compensation Value (TEST.TDCV[5:0]).

3.1.4.2 Configuration and Status

Compensation for the transceiver loop delay by the CAN FD Controller is enabled via the Transceiver Delay Compensation bit (FBTP.TDC). The transceiver delay compensation offset is configured via the Transceiver Delay Compensation Offset field (FBTP.TDCO[4:0]). The actual delay compensation value applied by the CAN FD Controller's protocol engine (i.e. the measured delay plus the offset) can be read from the Transceiver Delay Compensation Value field (TEST.TDCV[5:0]).

3.1.5. Restricted Operation Mode

In Restricted Operation Mode the node is able to receive data and remote frames and to give acknowledge to valid frames, but it does not send data frames, remote frames, active error frames, or overload frames. In case of an error condition or overload condition, it does not send dominant bits, instead it waits for the occurrence of bus idle condition to resynchronize itself to the CAN communication. The error counters are not incremented.

The CPU can set the CAN FD Controller into Restricted Operation Mode by setting the Restricted Operation Mode bit (CCCR.ASM). CCCR.ASM can only be set by the CPU when both Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) are set to "1". CCCR.ASM can be reset by the CPU at any time.

Restricted Operation Mode is automatically entered when the Tx Handler was not able to read data from the Message RAM in time. To leave Restricted Operation Mode, the CPU must follow the procedure below:

1. Cancel all pending transmission requests by setting Tx Buffer Cancellation Request Register TXBCR to 0xFFFF_FFFF (the Tx Buffer Request Pending Register TXBRP will be reset to "0")
2. Issue a clock stop request by setting Clock Stop Request (CCCR.CSR) to "1"
3. Wait till both CCCR.INIT and Clock Stop Acknowledge (CCCR.CSA) are "1"
4. First reset CCCR.CSR
5. Then reset CCCR.INIT
6. Wait till CCCR.INIT is "0"
7. Issue a second clock stop request by setting CCCR.CSR to "1"

8. Wait till both CCCR.INIT and CCCR.CSA are "1"
9. Set CCCR.CCE to "1" and reset CCCR.CSR and CCCR.ASM
10. Restart the CAN FD Controller by setting CCCR.INIT to "0"
11. Wait till CCCR.INIT is "0"
12. Reconfigure the CAN Operation Modes CAN FD Operation (CCCR.FDO) and CAN FD Bit Rate Switching (CCCR.FDBS) via CAN Mode Request (CCCR.CMR[1:0])
13. Request the transmissions cancelled in the first step

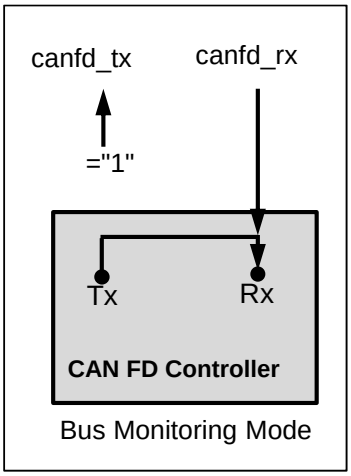
The Restricted Operation Mode can be used in applications that adapt themselves to different CAN bit rates. In this case the application tests different bit rates and leaves the Restricted Operation Mode after it has received a valid frame.

3.1.6. Bus Monitoring Mode

The CAN FD Controller is set in Bus Monitoring Mode by programming the Bus Monitoring Mode bit (CCCR.MON) to one. In Bus Monitoring Mode (see ISO11898-1, 10.12 Bus monitoring), the CAN FD Controller is able to receive valid data frames and valid remote frames, but cannot start a transmission. In this mode, it sends only recessive bits on the CAN bus. If the CAN FD Controller is required to send a dominant bit (ACK bit, overload flag, active error flag), the bit is rerouted internally so that the CAN FD Controller monitors this dominant bit, although the CAN bus may remain in recessive state. The Tx Buffer Request Pending register (TXBRP) will be cleared when setting Configuration Change Enable CCCR.CCE = 1 upon entering Bus Monitoring Mode via CCCR.MON = 1, and will be held in reset state while it is in this mode.

The Bus Monitoring Mode can be used to analyze the traffic on a CAN bus without affecting it by the transmission of dominant bits. Figure 3-2 shows the connection of signals canfd_tx and canfd_rx to the CAN FD Controller in Bus Monitoring Mode.

Figure 3-2 Pin Control in Bus Monitoring Mode



3.1.7. Disabled Automatic Retransmission

According to the CAN Specification (see ISO11898-1, 6.3.3 Recovery Management), the CAN FD Controller provides means for automatic retransmission of frames that have lost arbitration or that have been disturbed by errors during transmission. By default automatic retransmission is enabled. The automatic retransmission may be disabled via the Disable Automatic Retransmission bit (CCCR.DAR).

In DAR mode all transmissions are automatically cancelled after they started on the CAN bus. In this mode the CAN FD Controller will internally reset the corresponding Tx Buffer's Tx Request Pending bit (TXBRP.TRPn) when one of the following conditions occur:

- A successful transmission has occurred. This can be observed as
  Corresponding Tx Buffer Transmission Occurred bit TXBTO.TOn set
  Corresponding Tx Buffer Cancellation Finished bit TXBCF.CFn not set

  or, in case a successful transmission was executed in spite of cancellation,

  Corresponding Tx Buffer Transmission Occurred bit TXBTO.TOn set
  Corresponding Tx Buffer Cancellation Finished bit TXBCF.CFn set
- A transmission has not yet been started at the point of cancellation
- A transmission has been aborted due to lost arbitration or frame transmission disturbed. This can be observed by
  Corresponding Tx Buffer Transmission Occurred bit TXBTO.TOn not set
  Corresponding Tx Buffer Cancellation Finished bit TXBCF.CFn set

Note:

- *Do not use the same Tx Buffer for consecutive DAR transmissions or wait at least for 4 nominal bit times after successful transmission before requesting the next transmission from the same Tx Buffer.*

3.1.8. Power Down (Sleep Mode)

The CAN FD Controller can be set into power down mode via Clock Stop Request (CCCR.CSR).

When all pending transmission requests have completed (when Tx Buffer Request Pending Register TXBRP = 0x00000000), the CAN FD Controller waits until bus idle state is detected. Then the CAN FD Controller sets Initialization (CCCR.INIT) to one to prevent any further CAN transfers. Now the CAN FD Controller acknowledges that it is ready for power down by setting Clock Stop Acknowledge (CCCR.CSA) to one. In this state, before the clocks are switched off, further register accesses can be made. A write access to CCCR.INIT will have no effect. Now the CAN FD Controller clock inputs canfd_bclk and canfd_cclk may be switched off.

To leave power down mode, the application has to turn on the CAN FD Controller clocks before resetting Clock Stop Request (CCCR.CSR). The CAN FD Controller will acknowledge this by resetting CCCR.CSA. Afterwards, the application can restart CAN communication by resetting bit CCCR.INIT.

3.1.9. Test Modes

To enable write access to the Test register TEST, Test Mode Enable (CCCR.TEST) has to be set to one. This allows the configuration of the test modes and test functions.

Four output functions are available for the CAN transmit pin canfd_tx by programming Control of Transmit Pin (TEST.TX). These are

- the default function – the serial data output
- drive the CAN Sample Point signal to monitor the CAN FD Controller's bit timing
- drive constant dominant values
- drive constant recessive values.

The actual value at pin canfd_rx can be read from Receive Pin (TEST.RX). Both functions can be used to check the CAN bus' physical layer.

Due to the synchronization mechanism between the CAN clock and Bus clock domains, there may be a delay of several Bus clock periods between writing to TEST.TX until the new configuration is visible at output pin canfd_tx. This applies also when reading input pin canfd_rx via TEST.RX.

Note:

- *Test modes should be used for self test only. The software control for pin canfd_tx interferes with all CAN protocol functions. It is not recommended to use test modes for application.*

3.1.9.1 External Loop Back Mode

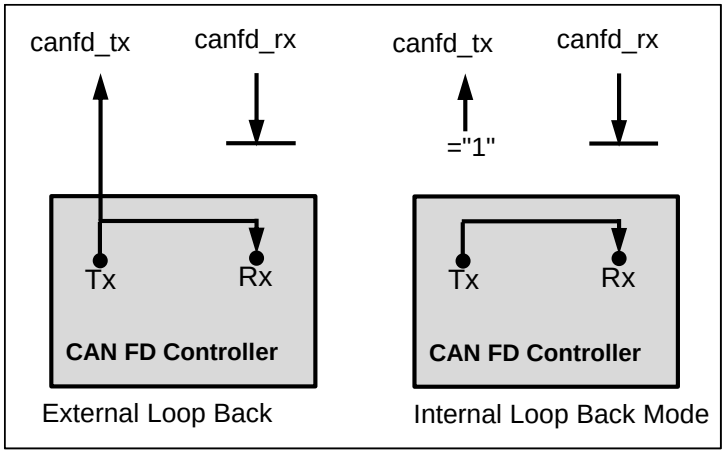
The CAN FD Controller can be set in External Loop Back Mode by programming the Loop Back Mode bit (TEST.LBCK) to one. In External Loop Back Mode, the CAN FD Controller treats its own transmitted messages as received messages and stores them (if they pass acceptance filtering) into a dedicated Rx Buffer or an Rx FIFO. Figure 3-3 shows the connection of signals canfd_tx and canfd_rx to the CAN FD Controller in External Loop Back Mode.

This mode is provided for hardware self test. The CAN FD Controller ignores acknowledge errors in External Loop Back Mode. In this mode the CAN FD Controller performs an internal feedback from its Tx output to its Rx input. The actual value of the canfd_rx input pin is disregarded by the CAN FD Controller. The transmitted messages can be monitored at the canfd_tx pin.

3.1.9.2 Internal Loop Back Mode

Internal Loop Back Mode is entered by programming bits Loop Back Mode (TEST.LBCK) and Bus Monitoring Mode (CCCR.MON) to one. This mode can be used for a "Hot Self test", meaning the CAN FD Controller can be tested without affecting a running CAN system connected to the pins canfd_tx and canfd_rx. In this mode pin canfd_rx is disconnected from the CAN FD Controller and pin canfd_tx is held recessive. Figure 3-3 shows the connection of canfd_tx and canfd_rx to the CAN FD Controller in case of Internal Loop Back Mode.

Figure 3-3 Pin Control in Loop Back Modes



3.2. Timestamp Generation

For timestamp generation the CAN FD Controller supplies a 16-bit wrap-around counter. On start of frame reception/transmission the counter value is captured and stored into the timestamp section of an Rx Buffer and FIFO or Tx Event FIFO element (fields RXTS[15:0] and TXTS[15:0] of the respective elements).

For more details, see "5.8.Timestamp Counter Configuration (MCG_CANFDx_TSCC, CPG_CANFDx_TSCC) " and "5.9.Timestamp Counter Value (MCG_CANFDx_TSCV, CPG_CANFDx_TSCV)".

3.3. Timeout Counter

To signal timeout conditions, the CAN FD Controller supplies a 16-bit Timeout Counter. For Rx FIFO 0, Rx FIFO 1, and Tx Event FIFO triggered operations, it starts down counting when an element is stored, and stops counting when the FIFO is emptied by the CPU. The Timeout Counter may also be triggered by a "0" write to the Initialization bit CCCR.INIT, and count down repeatedly until CCCR.INIT is set back to "1".

The Timeout Counter operates as down-counter and uses the same prescaler controlled by Timestamp Counter Prescaler (TSCC.TCP[3:0]) as the Timestamp Counter. The Timeout Counter is configured via register Timeout Counter Configuration (TOCC). The actual counter value can be read from Timeout Counter (TOCV.TOC[15:0]). The Timeout Counter is operable only when CCCR.INIT = 0. It is stopped when Initialization CCCR.INIT = 1, e.g. when the CAN FD Controller enters Bus_Off state.

The operation mode is selected by the Timeout Select field TOCC.TOS[1:0]. For details, see "5.11.Timeout Counter Value (MCG_CANFDx_TOCV, CPG_CANFDx_TOCV)".

Note:

- *When Timestamp Select TSCC.TSS[1:0] = 0b01, the clock signal for the Timeout Counter is derived from the CAN Core's sample point signal. Therefore, if the baud rate switch feature in CAN FD is used, the timeout counter is clocked differently in arbitration and data field.*

3.4. Rx Handling

The Rx Handler controls the acceptance filtering, the transfer of received messages to the dedicated Rx Buffers or to one of the two Rx FIFOs, as well as the Rx FIFO's Put and Get Indices.

3.4.1.Acceptance Filtering

3.4.2.Rx FIFOs

3.4.3.Dedicated Rx Buffers

3.4.4.Debug on CAN Support

3.4.5. Protocol Exception Event

3.4.1. Acceptance Filtering

The CAN FD Controller offers the possibility to configure two sets of acceptance filters, one for standard identifiers and one for extended identifiers. These filters can be assigned to a dedicated Rx Buffer or to Rx FIFO 0, 1. For acceptance filtering each list of filters is executed from element #0 until the first matching element. Acceptance filtering stops at the first matching element. The following filter elements are not evaluated for this message.

Main Features

- Each filter element can be configured as
  - range filter (from - to)
  - filter for one or two specific IDs (dual filter)
  - classic bit mask filter
  - filter for a single dedicated ID
- Each filter element is configurable for acceptance or rejection filtering
- Each filter element can be enabled/disabled individually
- Filters are checked sequentially from filter element 0, execution stops with the first matching filter element

Related Configuration Registers

- Global Filter Configuration GFC
- Standard ID Filter Configuration SIDFC
- Extended ID Filter Configuration XIDFC
- Extended ID AND Mask XIDAM

Filter Properties

Depending on the configuration of the filter element configuration fields (SFEC[2:0]/EFEC[2:0]) a match triggers one of the following actions:

- Store received frame in Rx FIFO 0 or Rx FIFO 1
- Store received frame in dedicated Rx Buffer
- Reject received frame
- Set High Priority Message interrupt flag IR.HPM
- Set High Priority Message interrupt flag IR.HPM and store received frame in Rx FIFO 0 or Rx FIFO 1

Description

Acceptance filtering is started after the complete identifier has been received. After acceptance filtering has completed, and if a matching dedicated Rx Buffer or Rx FIFO has been found, the Message Handler starts writing the received message data in portions of 32 bits to the matching dedicated Rx Buffer or Rx FIFO.

If the CAN protocol controller has detected an error condition (e.g. CRC error), this message is discarded with the following impact on the affected dedicated Rx Buffer or Rx FIFO:

- Dedicated Rx Buffer

  New Data flag NDAT1/2.NDn of matching Rx Buffer is not set, but Rx Buffer (partly) overwritten with received data. For error type see Last Error Code (PSR.LEC[2:0]) respectively Fast Last Error Code (PSR.FLEC[2:0]).

- Rx FIFO

  Put index of matching Rx FIFO RXFnS.FnPI[5:0] is not updated, but related Rx FIFO element (partly) overwritten with received data. For error type see PSR.LEC[2:0] respectively PSR.FLEC[2:0].

In case the matching Rx FIFO is operated in overwrite mode, the boundary conditions described in "3.4.2.2. Rx FIFO Overwrite Mode" have to be considered.

3.4.1.1 Filter Types

Range Filter

A range filter matches all received frames with Message IDs in the range defined by the Standard Message ID Filter Element SFID1[10:0]/SFID2[10:0] fields resp. Extended Message ID Filter Element EFID1[28:0]/EFID2[28:0] fields (the range includes the end points SFID1[10:0]/SFID2[10:0] resp. EFID1[28:0]/EFID2[28:0]).

There are two possibilities for the Extended Filter Type of the Extended Message ID Filter Element (EFT[1:0]) when range filtering is used together with extended frames:

- EFT[1:0] = 0b00:

  The Message ID of received frames is ANDed with the Extended ID AND Mask (XIDAM) before the range filter is applied

- EFT[1:0] = 0b11:

  The Extended ID AND Mask (XIDAM) is not used for range filtering

Filter for Specific IDs (Dual Filter)

A filter element can be configured to filter for one or two specific Message IDs. To filter for one specific Message ID, the filter element has to be configured with SFID1[10:0] = SFID2[10:0] resp. EFID1[28:0] = EFID2[28:0].

Classic Bit Mask Filter

Classic bit mask filtering is intended to filter groups of Message IDs by masking single bits of a received Message ID. With classic bit mask filtering SFID1[10:0]/EFID1[28:0] is used as Message ID filter, while SFID2[10:0]/EFID2[28:0] is used as filter mask.

A zero bit at the filter mask will mask out the corresponding bit position of the configured ID filter, e.g. the value of the received Message ID at that bit position is not relevant for acceptance filtering. Only those bits of the received Message ID where the corresponding mask bits are one are relevant for acceptance filtering.

In case all mask bits are one, a match occurs only when the received Message ID and the Message ID filter are identical. If all mask bits are zero, all Message IDs match.

Filter for a Single Dedicated ID

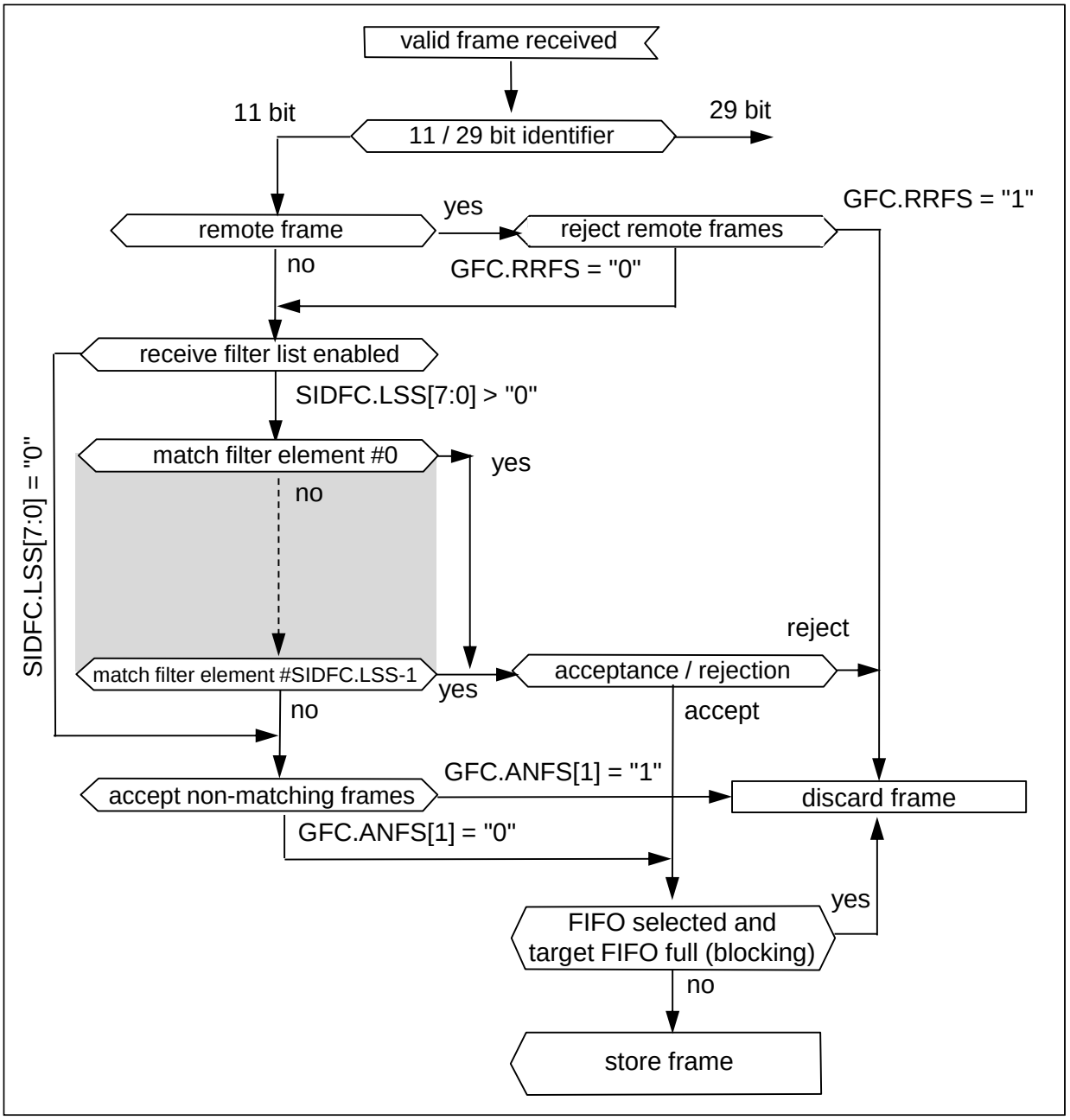
A filter for a single dedicated ID stores received messages in dedicated Rx Buffers if the received ID matches SFID1[10:0]/EFID1[28:0]. SFID2[10:0]/EFID2[28:0] is used to configure the offset to the Rx Buffer Start Address (RXBC.RBSA[15:2]) of where the message is to be stored, and also to define how the message is to be treated (Debug message or Dedicated Rx message).

3.4.1.2 Standard Message ID Filtering

Figure 3-4 below shows the flow for standard Message ID (11-bit Identifier) filtering.

Controlled by the Global Filter Configuration GFC and the Standard ID Filter Configuration SIDFC, Message ID, Remote Transmission Request bit (RTR), and the Identifier Extension bit (IDE) of received frames are compared against the list of configured filter elements.

Figure 3-4 Standard Message ID Filter Path



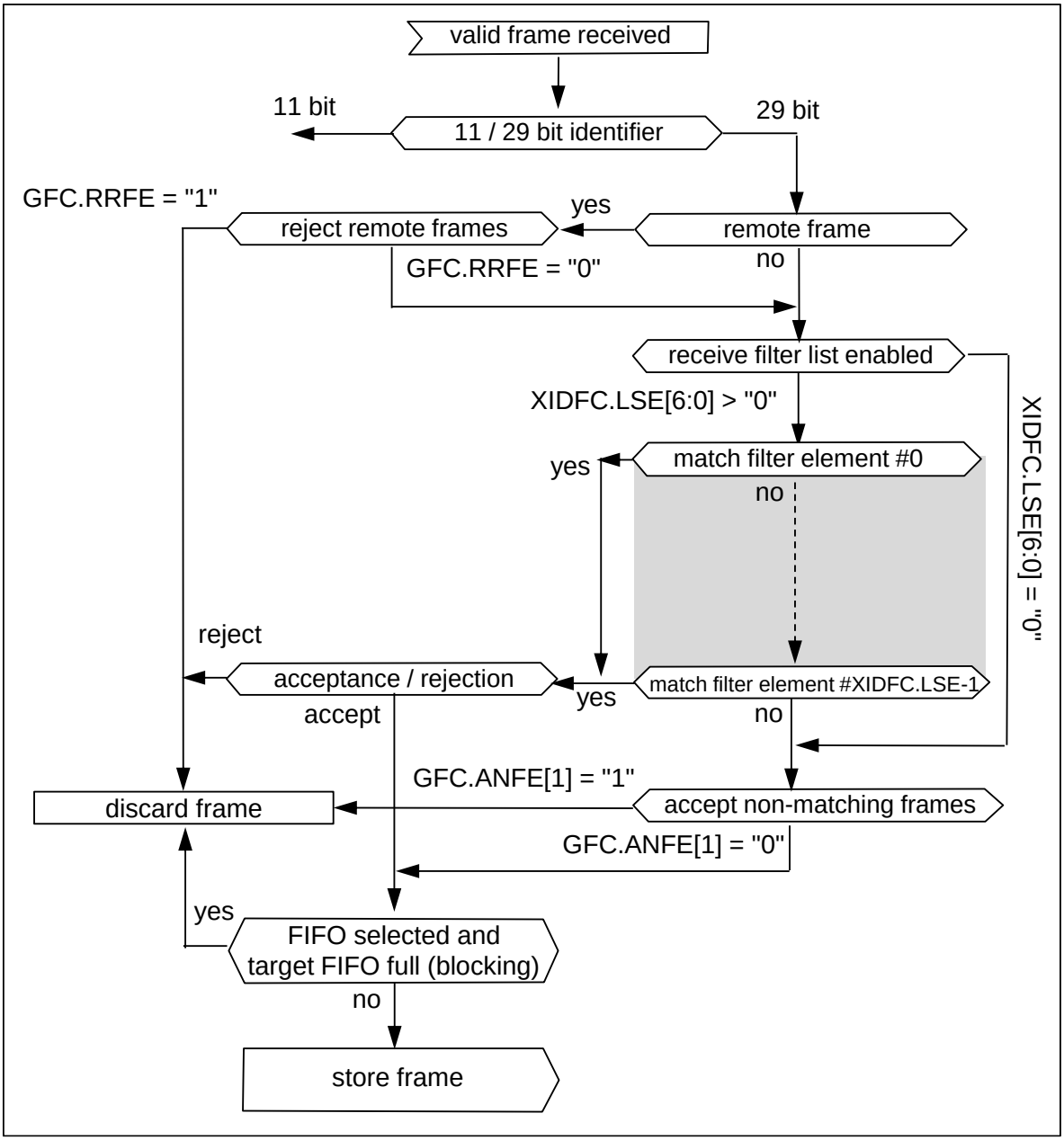
3.4.1.3 Extended Message ID Filtering

Figure 3-5 below shows the flow for extended Message ID (29-bit Identifier) filtering.

Controlled by the Global Filter Configuration GFC and the Extended ID Filter Configuration XIDFC, Message ID, Remote Transmission Request bit (RTR), and the Identifier Extension bit (IDE) of received frames are compared against the list of configured filter elements.

The Extended ID AND Mask XIDAM.EIDM[28:0] is ANDed with the received identifier before the filter list is executed.

Figure 3-5 Extended Message ID Filter Path



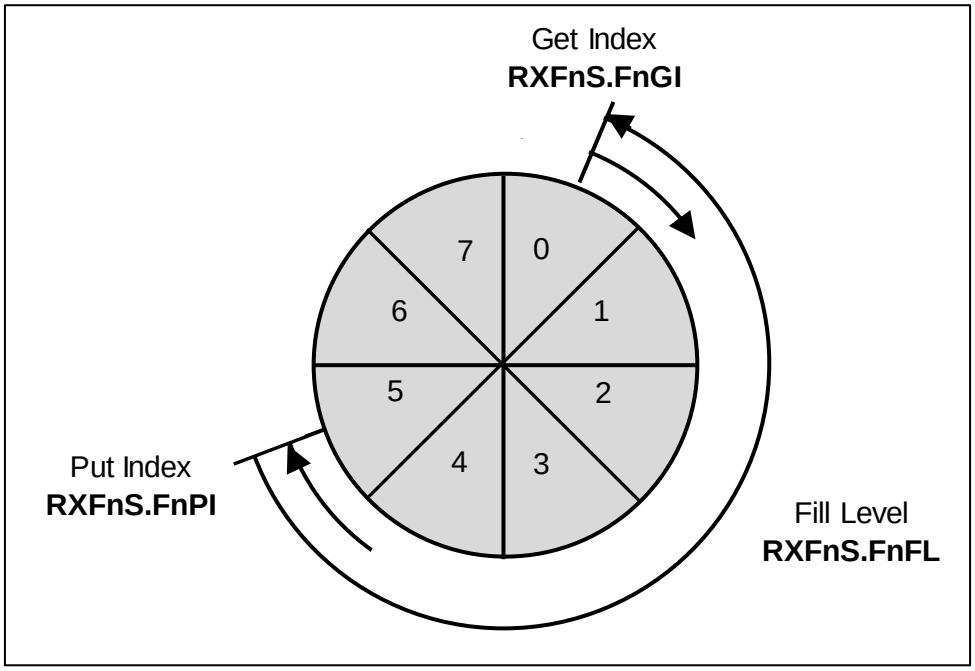
3.4.2. Rx FIFOs

Rx FIFO 0 and Rx FIFO 1 can be configured to hold up to 64 elements each. Configuration of the two Rx FIFOs is done via the Rx FIFO 0 and 1 Configuration registers RXF0C and RXF1C.

Received messages that passed acceptance filtering are transferred to the Rx FIFO as configured by the matching filter element.

To avoid an Rx FIFO overflow, the Rx FIFO watermark can be used. When the Rx FIFO fill level RXFnS.FnFL[6:0] reaches the Rx FIFO watermark configured by Rx FIFO n Watermark (RXFnC.FnWM[6:0]), interrupt flag Rx FIFO n Watermark Reached (IR.RFnW) is set. When the Rx FIFO Put Index reaches the Rx FIFO Get Index an Rx FIFO Full condition is signaled by Rx FIFO n Full (RXFnS.FnF). In addition interrupt flag Rx FIFO n Full (IR.RFnF) is set.

Figure 3-6 RX FIFO Status



Addressing Rx Buffers

An Rx Buffer allocates Element Size 32-bit words in the Message RAM (see Table 3-2).Therefore when reading from an Rx FIFO, the start address of an Rx Buffer will be

Rx FIFO Get Index (RXFnS.FnGI[5:0]) × Element Size
+ corresponding Rx FIFO start address (RXFnC.FnSA[15:2]).

Table 3-2 Rx Buffer/FIFO Element Size

RXESC.RBDS[2:0] RXESC.FnDS[2:0]	Data Field [Bytes]	Element Size [RAM Words]
000	8	4
001	12	5
010	16	6
011	20	7

RXESC.RBDS[2:0] RXESC.FnDS[2:0]	Data Field [Bytes]	Element Size [RAM Words]
100	24	8
101	32	10
110	48	14
111	64	18

3.4.2.1 Rx FIFO Blocking Mode

The Rx FIFO blocking mode is configured by FIFO n Operation Mode RXFnC.FnOM = 0. This is the default operation mode for the Rx FIFOs.

When an Rx FIFO full condition is reached (RXFnS.FnPI[5:0] = RXFnS.FnGI[5:0]), no further messages are written to the corresponding Rx FIFO until at least one message has been read out and the Rx FIFO Get Index has been incremented. An Rx FIFO full condition is signaled by Rx FIFO n Full (RXFnS.FnF = 1). In addition interrupt flag Rx FIFO n Full (IR.RFnF) is set.

In case a message is received while the corresponding Rx FIFO is full, this message is discarded and the message lost condition is signaled by Rx FIFO n Message Lost RXFnS.RFnL = 1. In addition interrupt flag Rx FIFO n Message Lost (IR.RFnL) is set.

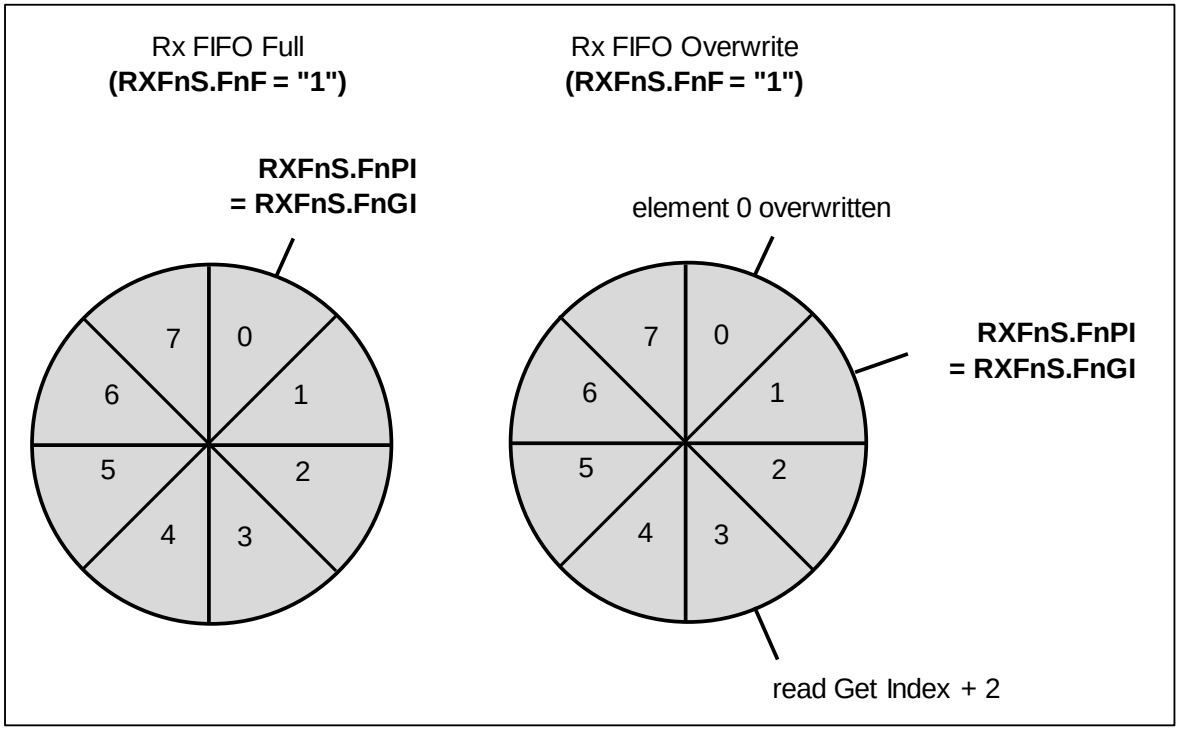
3.4.2.2 Rx FIFO Overwrite Mode

The Rx FIFO overwrite mode is configured by FIFO n Operation Mode RXFnC.FnOM = 1.

When an Rx FIFO full condition (RXFnS.FnPI[5:0] = RXFnS.FnGI[5:0]) is signaled by Rx FIFO n Full RXFnS.FnF = 1, the next message accepted for the Rx FIFO will overwrite the oldest Rx FIFO message. Put and get index are both incremented by one.

When an Rx FIFO is operated in overwrite mode and an Rx FIFO full condition is signaled, reading of the Rx FIFO elements should start at least at get index + 1. The reason for that is, that it might happen, that a received message is written to the Message RAM (put index) while the CPU is reading from the Message RAM (get index). In this case inconsistent data may be read from the respective Rx FIFO element. Adding an offset to the get index when reading from the Rx FIFO avoids this problem. The offset depends on how fast the CPU accesses the Rx FIFO. Figure 3-7 shows an offset of two with respect to the get index when reading the Rx FIFO. In this case the two messages stored in element 1 and 2 are lost.

Figure 3-7 RX FIFO Overflow Handling



After reading from the Rx FIFO, the number of the last element read has to be written to the Rx FIFO Acknowledge Index (RXFnA.FnAI[5:0]). This increments the get index to that element number. In case the put index has not been incremented to this Rx FIFO element, the Rx FIFO full condition is reset (Rx FIFO n Full RXFnS.FnF = 0).

3.4.3. Dedicated Rx Buffers

The CAN FD Controller supports up to 64 dedicated Rx Buffers. The start address of the dedicated Rx Buffer section is configured via the Rx Buffer Start Address field (RXBC.RBSA[15:2]).

For each dedicated Rx Buffer a Standard or Extended Message ID Filter Element with SFEC[2:0]/EFEC[2:0] = 0b111 and SFID2[10:9]/EFID2[10:9] = 0b00 has to be configured.

After a received message has been accepted by a filter element, the message is stored into the Rx Buffer in the Message RAM referenced by the filter element. The format is the same as for an Rx FIFO element. In addition the flag IR.DRX (Message stored to Dedicated Rx Buffer) in the interrupt register is set.

Table 3-3 Example Filter Configuration for Rx Buffers

Filter Element	SFID1[10:0] EFID1[28:0]	SFID2[10:9] EFID2[10:9]	SFID2[5:0] EFID2[5:0]
0	ID message 1	00	00 0000
1	ID message 2	00	00 0001
2	ID message 3	00	00 0010

After the last word of a matching received message has been written to the Message RAM, the respective New Data flag in register NDAT1, 2 is set. As long as the New Data flag is set, the respective dedicated Rx Buffer is locked against updates from received matching frames. The New Data flags have to be reset by the CPU by writing a "1" to the respective bit position.

While a dedicated Rx Buffer's New Data flag is set, a Message ID Filter Element referencing this specific dedicated Rx Buffer will not match, causing the acceptance filtering to continue. Following Message ID Filter Elements may cause the received message to be stored into another dedicated Rx Buffer, or into an Rx FIFO, or the message may be rejected, depending on filter configuration.

Addressing Dedicated Rx Buffers

A Dedicated Rx Buffer allocates Element Size 32-bit words in the Message RAM (see Table 3-2). Therefore the start address of a dedicated Rx Buffer in the Message RAM is calculated by

(Offset defined in Message ID Filter Element SFID2[5:0] resp. EFID2[5:0]) × Element Size + Rx Buffer Start Address (RXBC.RBSA[15:2]).

3.4.4. Debug on CAN Support

For debug handling, three consecutive Rx buffers (e.g. #61, #62, #63) can be used for storage of three specific debug messages A, B, and C. The format is the same as for a dedicated Rx Buffer or an Rx FIFO element.

For filtering of debug messages Standard/Extended Filter Elements with Standard/Extended Filter Configuration SFEC[2:0]/EFEC[2:0] = 0b111 have to be set up. Messages matching these filter elements are stored into the Rx Buffers addressed by the lower six bits of the Standard/Extended Message ID Filter Element fields SFID2[5:0]/EFID2[5:0]. The upper two bits SFID2[10:9]/EFID2[10:9] are used to define the Debug Message that is to be stored in the respective Rx Buffer.

When a debug message is stored, neither the respective New Data flag (NDAT1.NDn / NDAT2.NDn) nor the interrupt IR.DRX (Message stored to Dedicated Rx Buffer) are set.

After the three messages have been received in correct order, a DMA transfer is requested. After the DMA transfer has completed, the CAN FD Controller is prepared to receive the next set of debug messages.

Table 3-4 Example Filter Configuration for Debug Messages

Filter Element	SFID1[10:0] EFID1[28:0]	SFID2[10:9] EFID2[10:9]	SFID2[5:0] EFID2[5:0]
0	ID debug message A	01	11 1101
1	ID debug message B	10	11 1110
2	ID debug message C	11	11 1111

Notes:

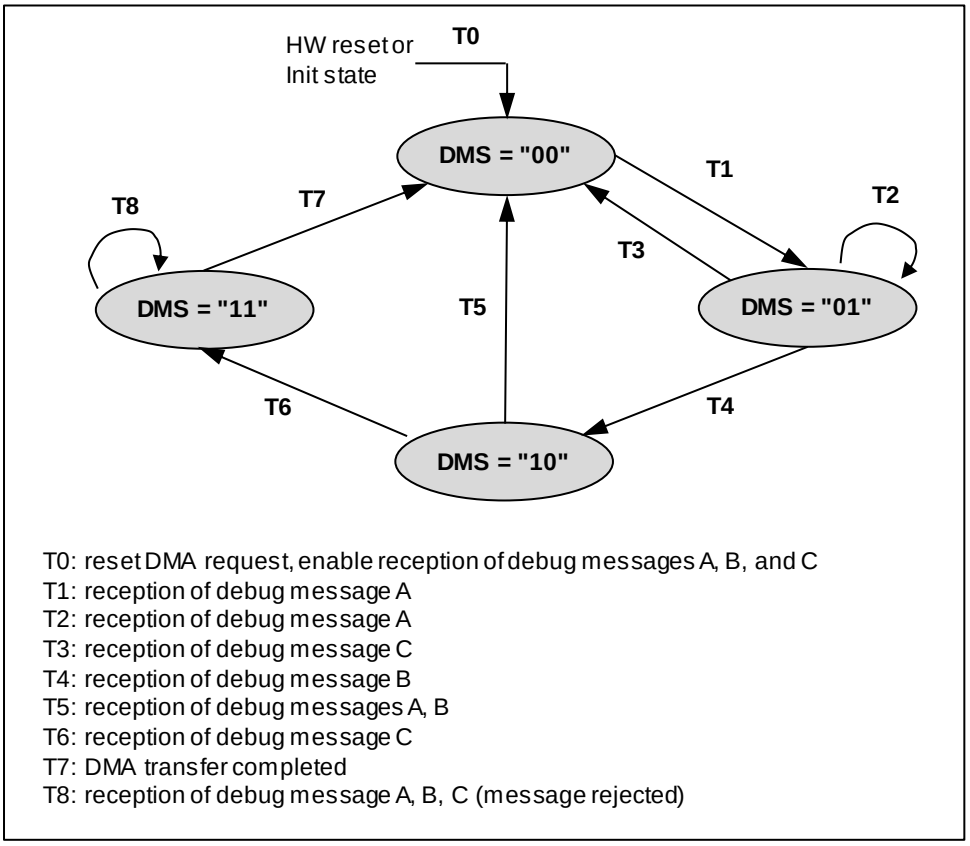
- *Debug on CAN feature is not supported for TRAVEO™ T1G Platform.*
- *DMA transfer request is not supported also.*

3.4.4.1 Debug Message Handling

The debug message handling state machine assures that debug messages are stored to three consecutive Rx Buffers in correct order. In case of missing messages the process is restarted. The DMA request is activated only when all three debug messages A, B, C have been received in correct order.

The status of the debug message handling state machine is signaled via the Debug Message Status field (RXF1S.DMS[1:0]).

Figure 3-8 Debug Message Handling State Machine



Notes:

- *Debug message is used for Debug on CAN feature.*
- *Debug on CAN feature is not supported for TRAVEO™ T1G Platform.*

3.4.5. Protocol Exception Event

When the CAN FD Controller is enabled for CAN FD operation (CCCR.CME ≠ 0b00) and the CAN FD Controller detects the r0 bit following the EDL bit in a received message to be recessive, it treats this detection as a Protocol Exception Event. As reaction to the Protocol Exception Event,

- the error counters (ECR.REC[6:0] and ECR.TEC[7:0]) are not changed
- hard synchronization is enabled
- the CAN FD Controller will send recessive bits

The CAN FD Controller will also start an internal bit counter that counts the number of recessive bits detected on the CAN bus. If the counter detects eleven consecutive recessive bits, the CAN FD Controller will recover from the Protocol Exception Event and enter an idle state. In case an edge that causes synchronization is detected during the wait period, the CAN FD Controller will reset the internal bit counter and restart counting the number of recessive bits detected.

If the CAN FD Controller

- receives a message directly after the Protocol Exception Event, interrupt flag Message RAM Access Failure (IR.MRAF) will be asserted although the message has been received correctly.
- transmits a message directly after the Protocol Exception Event, IR.MRAF will not be asserted, but that frame will be transmitted with faulty frame format and cause an error frame.

Only the first message after a Protocol Exception Event is affected, all following messages (received or transmitted) will have no problem.

3.5. Tx Handling

The Tx Handler handles transmission requests for the dedicated Tx Buffers, the Tx FIFO, and the Tx Queue. It controls the transfer of transmit messages to the CAN Core, the Put and Get Indices, and the Tx Event FIFO. Up to 32 Tx Buffers can be set up for message transmission. The Tx Buffer element is described in "6.3. Tx Buffer Element".

The Tx Handler starts a Tx scan to check for the highest priority pending Tx request (Tx Buffer with lowest Message ID) when the Tx Buffer Request Pending register TXBRP is updated, or when a transmission has been started.

3.5.1.Transmit Pause

3.5.2.Dedicated Tx Buffers

3.5.3.Tx FIFO

3.5.4.Tx Queue

3.5.5.Mixed Dedicated Tx Buffers/Tx FIFO

3.5.6.Mixed Dedicated Tx Buffers/Tx Queue

3.5.7.Transmit Cancellation

3.5.8.Tx Event Handling

3.5.1. Transmit Pause

The transmit pause feature is intended for use in CAN systems where the CAN message identifiers are (permanently) specified to specific values and cannot easily be changed. These message identifiers may have a higher CAN arbitration priority than other defined messages, while in a specific application their relative arbitration priority should be inverse. This may lead to a case where one ECU (Electronic Control Unit) sends a burst of CAN messages that cause another ECU’s CAN messages to be delayed because that other messages have a lower CAN arbitration priority.

If e.g. CAN ECU-1 has the transmit pause feature enabled and is requested by its application software to transmit four messages, it will, after the first successful message transmission, wait for two nominal bit times of bus idle before it is allowed to start the next requested message. If there are other ECUs with pending messages, those messages are started in the idle time, they would not need to arbitrate with the next message of ECU-1. After having received a message, ECU-1 is allowed to start its next transmission as soon as the received message releases the CAN bus.

The transmit pause feature is controlled by the Transmit Pause bit (CCCR.TXP). If the bit is set, the CAN FD Controller will, each time it has successfully transmitted a message, pause for two nominal bit times before starting the next transmission. This enables other CAN nodes in the network to transmit messages even if their messages have lower prior identifiers. Default is transmit pause disabled (CCCR.TXP = 0).

This feature lose up burst transmissions coming from a single node and it protects against "babbling idiot" scenarios where the application program erroneously requests too many transmissions.

3.5.2. Dedicated Tx Buffers

Dedicated Tx Buffers are intended for message transmission under complete control of the CPU. Each Dedicated Tx Buffer is configured with a specific Message ID. In case that multiple Tx Buffers are configured with the same Message ID, the Tx Buffer with the lowest buffer number is transmitted first. These Tx buffers shall be requested in ascending order with lowest buffer number first. Alternatively all Tx buffers configured with the same Message ID can be requested simultaneously by a single write access to TXBAR.

If the data section has been updated, a transmission is requested by an Add Request via TXBAR.ARn. The requested messages arbitrate internally with messages from an optional Tx FIFO or Tx Queue and externally with messages on the CAN bus, and are sent out according to their Message ID.

Addressing Dedicated Tx Buffers

A Dedicated Tx Buffer allocates Element Size 32-bit words in the Message RAM (see Table 3-5). Therefore the start address of a dedicated Tx Buffer in the Message RAM is calculated by

transmit buffer index (0 to 31) × Element Size + Tx Buffers Start Address (TXBC.TBSA[15:2]).

Table 3-5 Tx Buffer/FIFO/Queue Element Size

TXESC.TBDS[2:0]	Data Field [Bytes]	Element Size [RAM Words]
000	8	4
001	12	5
010	16	6
011	20	7
100	24	8
101	32	10
110	48	14
111	64	18

3.5.3. Tx FIFO

Tx FIFO operation is configured by programming the Tx FIFO/Queue Mode bit (TXBC.TFQM) to "0". Messages stored in the Tx FIFO are transmitted starting with the message referenced by the Tx FIFO Get Index (TXFQS.TFGI[4:0]). After each transmission, the Get Index is incremented cyclically until the Tx FIFO is empty. The Tx FIFO enables transmission of messages with the same Message ID from different Tx Buffers in the order these messages have been written to the Tx FIFO. The CAN FD Controller calculates the number of available (free) Tx FIFO elements TXFQS.TFFL[5:0] (Tx FIFO Free Level) as difference between Tx FIFO Get Index (TXFQS.TFGI[4:0]) and Tx FIFO/Queue Put Index (TXFQS.TFQPI[4:0]).

Adding Messages and Requesting Transmissions

New transmit messages have to be written to the Tx FIFO starting with the Tx Buffer referenced by the Put Index (TXFQS.TFQPI[4:0]). An Add Request increments the Put Index to the next free Tx FIFO element. When the Put Index reaches the Get Index, Tx FIFO Full (TXFQS.TFQF = 1) is signaled. In this case no further messages should be written to the Tx FIFO until the next message has been transmitted and the Get Index has been incremented.

When a single message is added to the Tx FIFO, the transmission is requested by writing a "1" to the Add Request bit (TXBAR.ARn. n is the index of the requested Tx Buffer) related to the Tx Buffer referenced by the Tx FIFO's Put Index.

When multiple (n) messages are added to the Tx FIFO, they are written to n consecutive Tx Buffers starting with the Put Index. The transmissions are then requested via the Tx Buffer Add Request register (TXBAR) by requesting all n Tx FIFO Buffers in a single command. The Put Index is then cyclically incremented by n. The number of requested Tx buffers should not exceed the number of free Tx Buffers as indicated by the Tx FIFO Free Level (TXFQS.TFFL[5:0]). Transmission should not be requested to Tx Buffers outside the Put Index since this will cause the TXFQS.TFFL[5:0] to return a false value.

Cancelling Transmissions

Transmit Cancellation is intended for Tx Queues and Dedicated Tx Buffers. Cancelling requested Tx FIFO Buffer transmissions are not supported and is not recommended, except for the case where all Tx Buffers within the Tx FIFO are cancelled simultaneously (with a single command).

In the case that a transmission request for the Tx Buffer referenced by the Get Index is cancelled, the Get Index is incremented to the next Tx Buffer with pending transmission request and the Tx FIFO Free Level is recalculated. When transmission cancellation is applied to any other Tx Buffer, the Get Index and the FIFO Free Level remain unchanged.

Addressing Tx Buffers in the Tx FIFO

A Tx FIFO element allocates Element Size 32-bit words in the Message RAM (see Table 3-5). Therefore the start address of the next available (free) Tx FIFO Buffer is calculated by

Tx FIFO/Queue Put Index (TXFQS.TFQPI[4:0]) (0 to 31) × Element Size + Tx Buffers Start Address (TXBC.TBSA[15:2]).

3.5.4. Tx Queue

Tx Queue operation is configured by programming the Tx FIFO/Queue Mode bit TXBC.TFQM to "1". Messages stored in the Tx Queue are transmitted starting with the message with the lowest Message ID (highest priority). If multiple Tx Queue buffers are configured with the same Message ID, the transmission order depends on numbers of the buffers where the messages were stored for transmission. As these buffer numbers depend on the then current states of the PUT index, a prediction of the transmission order is not possible.

Adding Messages and Requesting Transmissions

New messages have to be written to the Tx Buffer referenced by the Tx FIFO/Queue Put Index TXFQS.TFQPI[4:0]. The PUT Index always points to that free buffer of the Tx Queue with the lowest buffer number. In case that the Tx Queue is full (TXFQS.TFQF = 1), the Put Index is not valid and no further message should be written to the Tx Queue until at least one of the requested messages has been sent out or a pending transmission request has been cancelled.

The application may use the Tx Buffer Request Pending register TXBRP instead of the Put Index and may place messages to any Tx Buffer without pending transmission request.

Addressing Tx Buffers in the Tx Queue

A Tx Queue Buffer allocates Element Size 32-bit words in the Message RAM (see Table 3-5). Therefore the start address of the next available (free) Tx Queue Buffer is calculated by

Tx FIFO/Queue Put Index (TXFQS.TFQPI[4:0]) (0 to 31) × Element Size + Tx Buffers Start Address (TXBC.TBSA[15:2]).

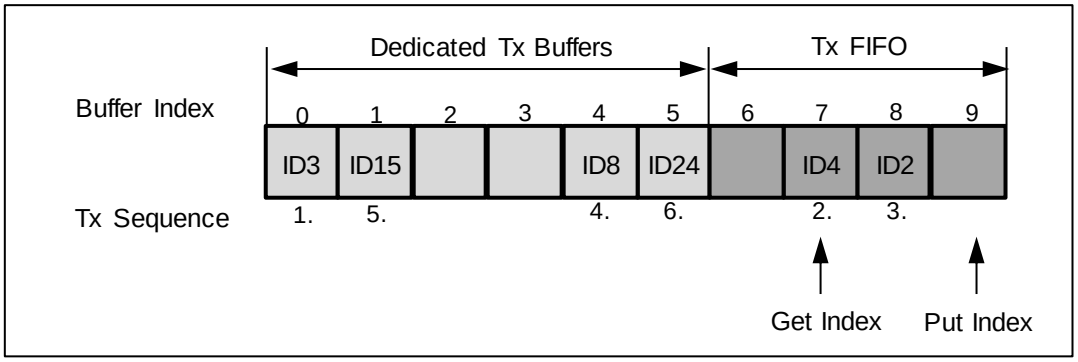
3.5.5. Mixed Dedicated Tx Buffers/Tx FIFO

In this case the Tx Buffers section in the Message RAM is subdivided into a set of Dedicated Tx Buffers and a Tx FIFO.

The number of Dedicated Tx Buffers is configured by TXBC.NDTB[5:0].

The number of Tx Buffers assigned to the Tx FIFO is configured by the Tx FIFO/Queue Size TXBC.TFQS[5:0]. In case TXBC.TFQS[5:0] is programmed to zero, only Dedicated Tx Buffers are used.

Figure 3-9 Example of Mixed Configuration Dedicated Tx Buffers/Tx FIFO



Tx Prioritization:

- Scan Dedicated Tx Buffers and oldest pending Tx FIFO Buffer (referenced by the Tx FIFO Get Index TXFQS.TFGI[4:0])
- Buffer with lowest Message ID gets highest priority and is transmitted next

Note:

- *Mixed Dedicated Tx Buffers/Tx FIFO isn't supported. Don't use combination Dedicated Tx Buffers and Tx FIFO.*

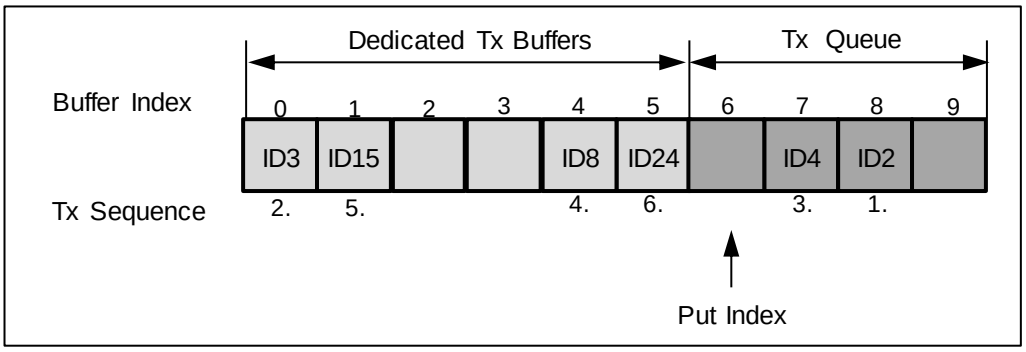
3.5.6. Mixed Dedicated Tx Buffers/Tx Queue

In this case the Tx Buffers section in the Message RAM is subdivided into a set of Dedicated Tx Buffers and a Tx Queue.

The number of Dedicated Tx Buffers is configured by TXBC.NDTB[5:0].

The number of Tx Queue Buffers is configured by the Tx FIFO/Queue Size TXBC.TFQS[5:0]. In case TXBC.TFQS[5:0] is programmed to zero, only Dedicated Tx Buffers are used.

Figure 3-10 Example of Mixed Configuration Dedicated Tx Buffers/Tx Queue



Tx Prioritization:

- Scan all Tx Buffers with activated transmission request
- Tx Buffer with lowest Message ID gets highest priority and is transmitted next

3.5.7. Transmit Cancellation

The CAN FD Controller supports transmit cancellation. To cancel a requested transmission from a dedicated Tx Buffer or a Tx Queue Buffer the CPU has to write a "1" to the corresponding bit position of the Tx Buffer Cancellation Request register TXBCR. Transmit cancellation is not intended for Tx FIFO operation.

Successful cancellation is signaled by setting the corresponding bit of the Tx Buffer Cancellation Finished register TXBCF to "1".

A cancellation request resets the corresponding Transmission Request Pending bit (TXBRP.TRPn). In case a transmit cancellation is requested while a transmission from a Tx Buffer is already ongoing, the corresponding TXBRP.TRPn bit remains set as long as the transmission is in progress.

- If the transmission was successful, the corresponding Transmission Occurred (TXBTO.TOn) and Cancellation Finished (TXBCF.CFn) bits are set.
- If the transmission was not successful (the transmission has been aborted due to lost arbitration, error occurred during frame transmission), it is not repeated and only the corresponding Cancellation Finished bit (TXBCF.CFn) is set.

TXBCF.CFn is also set when the transmission has not yet been started at the point of cancellation,

The cancellation request bits (TXBCR.CRn) are reset directly after the corresponding Transmission Request Pending bit (TXBRP.TRPn) has been reset.

Note:

- *In case a pending transmission is cancelled immediately before this transmission could have been started, there follows a short time window where no transmission is started even if another message is also pending in this node. This may enable another node to transmit a message which may have a lower priority (a greater message ID) than the second message in this node.*

3.5.8. Tx Event Handling

To support Tx event handling the CAN FD Controller has implemented a Tx Event FIFO. After the CAN FD Controller has transmitted a message on the CAN bus, Message ID and timestamp are stored in a Tx Event FIFO element. To link a Tx event to a Tx Event FIFO element, the Message Marker from the transmitted Tx Buffer MM[7:0] is copied into the Tx Event FIFO element.

The Tx Event FIFO can be configured by Event FIFO Size (TXEFC.EFS[5:0]) to a maximum of 32 elements.

When a Tx Event FIFO full condition is signaled by interrupt flag Tx Event FIFO Full (IR.TEFF), no further elements are written to the Tx Event FIFO until at least one element has been read out and the Tx Event FIFO Get Index TXEFS.EFGI[4:0] has been incremented. In case a Tx event occurs while the Tx Event FIFO is full, this event is discarded and interrupt flag Tx Event FIFO Element Lost IR.TEFL is set.

To avoid a Tx Event FIFO overflow, the Tx Event FIFO watermark can be used. When the Tx Event FIFO fill level TXEFS.EFFL[5:0] reaches the Tx Event FIFO watermark configured by TXEFC.EFWM[5:0], interrupt flag Tx Event FIFO Watermark Reached IR.TEFW is set.

Addressing Tx Events in the Tx Event FIFO

The start address when reading from the Tx Event FIFO will be

2 × Tx Event FIFO Get Index TXEFS.EFGI[4:0] + Tx Event FIFO start address TXEFC.EFSA[15:2].

3.6. FIFO Acknowledge Handling

The Get Indices of Rx FIFO 0, Rx FIFO 1, and the Tx Event FIFO (RXF0S.F0GI[5:0], RXF1S.F1GI[5:0], and TXEFS.EFGI[4:0], respectively) are controlled by writing to the corresponding FIFO Acknowledge Index (RXF0A.F0AI[5:0], RXF1A.F1AI[5:0], and TXEFA.EFAI[4:0], respectively). Writing to the FIFO Acknowledge Index will set the FIFO Get Index to the FIFO Acknowledge Index plus one and thereby updates the FIFO Fill Level. There are two use cases:

- When only a single element has been read from the FIFO (the one being pointed to by the Get Index), this Get Index value is written to the FIFO Acknowledge Index. (Multiple elements may be read by repeating this operation)
- When a sequence of elements has been read from the FIFO, it is sufficient to write the FIFO Acknowledge Index only once at the end of that read sequence (value: Index of the last element read), to update the FIFO's Get Index.

Due to the fact that the CPU has free access to the Message RAM, special care has to be taken when reading FIFO elements in an arbitrary order (Get Index not considered). This might be useful when reading a High Priority Message from one of the two Rx FIFOs. In this case the Rx FIFO's Acknowledge Index should not be written because this would set the Get Index to a wrong position, alter the Rx FIFO's Fill Level, and also cause some of the older Rx FIFO elements to be lost.

Note:

- *The application has to ensure that a valid value is written to the FIFO Acknowledge Index. The CAN FD Controller does not check for erroneous values.*

3.7. Configuring the CAN Bit Timing

Each CAN node in the CAN network has its own clock generator (usually a quartz oscillator). The time parameter of the bit time can be configured individually for each CAN node. Even if each CAN node's oscillator has a different period, a common bit rate can be generated.

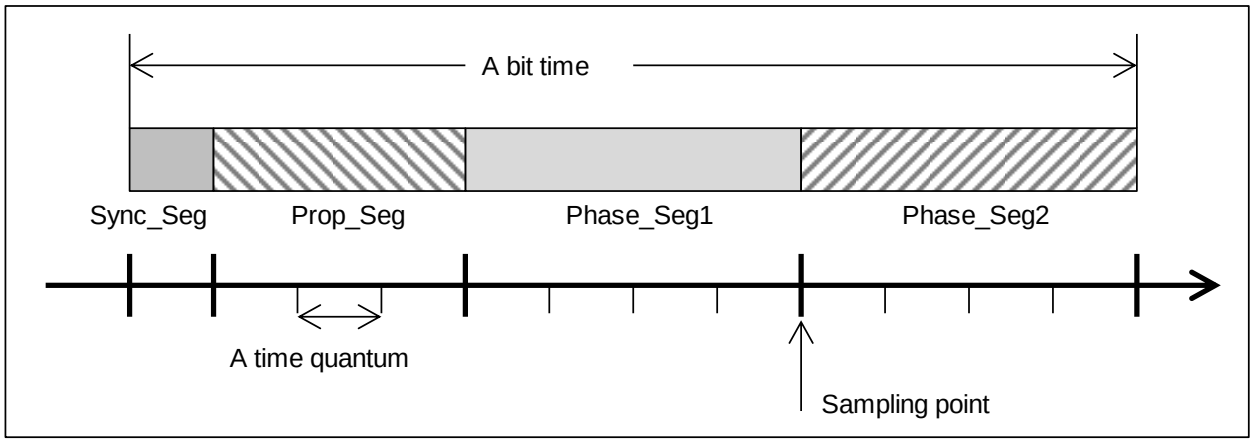
The oscillator frequencies vary slightly because of changes in temperature or voltage, or deterioration of components. As long as the frequencies vary only within the tolerance range of the oscillators, the CAN nodes can compensate for the different bit rates by resynchronizing to the bit stream.

3.7.1. CAN Bit Timing

The CAN FD specification V1.0 defines two bit times - nominal bit time and fast bit time (the CAN FD specification V1.0 calls these nominal bit time and data bit time). The nominal bit time is for the arbitration phase. The fast bit time has an equal or shorter length and can be used to accelerate the data phase (see Bit Rate Switching in "3.1.3.CAN FD Operation" for details).

The basic construction of a bit time is shared with both the nominal and fast bit times. The bit time can be divided into four segments according to the CAN specifications (see Figure 3-11): the synchronization segment (Sync_Seg), the propagation time segment (Prop_Seg), the phase buffer segment 1 (Phase_Seg1), and the phase buffer segment 2 (Phase_Seg2). The sample point, the point of time at which the bus level is read and interpreted as the value of that respective bit, is located at the end of Phase_Seg1.

Figure 3-11 Bit Time Construction



Each segment consists of a programmable number of time quanta, which is a multiple of the time quantum that is defined by canfd_cclk and a prescaler. The values and prescalers used to define these parameters differ for the nominal and fast bit times, and are configured by BTP (Bit Timing & Prescaler Register) and FBTP (Fast Bit Timing & Prescaler Register) as shown in the table below (Table 3-6).

Table 3-6 Bit Time Parameters

Parameter	Description
Time quantum tq (nominal) and tqf (fast)	Time quantum. Derived by multiplying the basic unit time quanta (i.e. the canfd_cclk period) with the respective prescaler. The time quantum is configured by the CAN FD Controller as nominal: tq = (BTP.BRP[9:0] +1) × canfd_cclk period fast: tqf = (FBTP.FBRP[4:0] + 1) × canfd_cclk period
Sync_Seg	Sync_Seg is fixed to 1 time quantum as defined by the CAN specifications and is therefore not configurable (inherently built into the CAN FD Controller). nominal: 1 tq fast: 1 tqf
Prop_Seg	Prop_Seg is the part of the bit time that is used to compensate for the physical delay times within the network. The CAN FD Controller configures the sum of Prop_Seg and Phase_Seg1 with a single parameter, i.e. nominal: Prop_Seg + Phase_Seg1 = BTP.TSEG1[5:0] + 1 fast: Prop_Seg + Phase_Seg1 = FBTP.FTSEG1[3:0] + 1
Phase_Seg1	Phase_Seg1 is used to compensate for edge phase errors before the sampling point. Can be lengthened by the resynchronization jump width. The sum of Prop_Seg and Phase_Seg1 is configured by the CAN FD Controller as nominal: BTP.TSEG1[5:0] + 1 fast: FBTP.FTSEG1[3:0] + 1

Parameter	Description
Phase_Seg2	Phase_Seg2 is used to compensate for edge phase errors after the sampling point. Can be shortened by the resynchronization jump width. Phase_Seg2 is configured by the CAN FD Controller as nominal: BTP.TSEG2[3:0] + 1 fast: FBTP.FTSEG2[2:0] + 1
SJW	Resynchronization Jump Width. Used to adjust the length of Phase_Seg1 and Phase_Seg2. SJW will not be longer than either Phase_Seg1 or Phase_Seg2. SJW is configured by the CAN FD Controller as nominal: BTP.SJW[3:0] + 1 fast: FBTP.FSJW[1:0] + 1

These relations result in the following equations for the nominal and fast bit times:

Nominal Bit time

= [Sync_Seg + Prop_Seg + Phase_Seg1 + Phase_Seg2] × tq
= [1 + (BTP.TSEG1[5:0] + 1) + (BTP.TSEG2[3:0] + 1)] × [(BTP.BRP[9:0] +1) × canfd_cclk period]

and for the fast bit time

= [1 + (FBTP.FTSEG1[3:0] + 1) + (FBTP.FTSEG2[2:0] + 1)] × [(FBTP.FBRP[4:0] +1) × canfd_cclk period]

Note:

- *The Information Processing Time (IPT) of the CAN FD Controller is zero, meaning the data for the next bit is available at the first CAN clock edge after the sample point. Therefore, the IPT does not have to be accounted for when configuring Phase_Seg2, which is the maximum of Phase_Seg1 and the IPT.*

3.7.2. CAN Bit Rates

Since the bit rate is the inverse of the bit time, the nominal bit rate is

1 / [{1 + (BTP.TSEG1[5:0] + 1) + (BTP.TSEG2[3:0] + 1)} × {(BTP.BRP[9:0] +1) × canfd_cclk period}]

and the FD bit rate is

1 / [{1 + (FBTP.FTSEG1[3:0] + 1) + (FBTP.FTSEG2[2:0] + 1)} × {(FBTP.FBRP[4:0] +1) × canfd_cclk period}]

From these formula we can see that the bit rates of the CAN FD Controller will depend on the CAN clock (i.e. canfd_cclk) period, and the range each parameter can be configured to. The tables below lists examples of the configurable bit rates at varying CAN clock frequencies. Empty boxes indicate that the desired bit rate cannot be configured at the specified input CAN clock frequency.

Table 3-7 Example Configurations for Nominal Bit Rates

CAN clock frequency	8MHz		10MHz		16MHz		20MHz		32MHz		40MHz	
configuration / nominal bit rate	# of tqs	BTP.BRP + 1	# of tqs	BTP.BRP + 1	# of tqs	BTP.BRP + 1	# of tqs	BTP.BRP + 1	# of tqs	BTP.BRP + 1	# of tqs	BTP.BRP + 1
125kbps	64tq 32tq 16tq 8tq	1 2 4 8	80tq 40tq 20tq 10tq	1 2 4 8	64tq 32tq 16tq 8tq	2 4 8 16	80tq 40tq 20tq 10tq	2 4 8 16	64tq 32tq 16tq 8tq	4 8 16 32	80tq 40tq 20tq 10tq	4 8 16 32
250kbps	32tq 16tq 8tq	1 2 4	40tq 20tq 10tq	1 2 4	64tq 32tq 16tq 8tq	1 2 4 8	80tq 40tq 20tq 10tq	1 2 4 8	64tq 32tq 16tq 8tq	2 4 8 16	80tq 40tq 20tq 10tq	2 4 8 16
500kbps	16tq 8tq	1 2	20tq 10tq	1 2	32tq 16tq 8tq	1 2 4	40tq 20tq 10tq	1 2 4	64tq 32tq 16tq 8tq	1 2 4 8	80tq 40tq 20tq 10tq	1 2 4 8
1Mbps	8tq	1	10tq	1	16tq 8tq	1 2	20tq 10tq	1 2	32tq 16tq 8tq	1 2 4	40tq 20tq 10tq	1 2 4

Table 3-8 Example Configurations for FD Bit Rates

CAN clock frequency	8MHz		10MHz		16MHz		20MHz		32MHz		40MHz	
configuration / FD bit rate	# of tqfs	FBTP.FBRP+1	# of tqfs	FBTP.FBRP+1	# of tqfs	FBTP.FBRP+1	# of tqfs	FBTP.FBRP+1	# of tqfs	FBTP.FBRP+1	# of tqfs	FBTP.FBRP+1
500kbps	16tqf 8tqf	1 2	20tqf 10tqf	1 2	16tqf 8tqf	2 4	20tqf 10tqf	2 4	16tqf 8tqf	4 8	20tqf 10tqf	4 8
1Mbps	8tqf	1	10tqf	1	16tqf 8tqf	1 2	20tqf 10tqf	1 2	16tqf 8tqf	2 4	20tqf 10tqf	2 4
2Mbps					8tqf	1	10tqf	1	16tqf 8tqf	1 2	20tqf 10tqf	1 2
4Mbps									8tqf	1	10tqf	1
5Mbps											8tqf	1

Note:

- *The user must configure the CAN bit timings so they comply with the corresponding CAN standards to ensure proper communication on the CAN bus.*

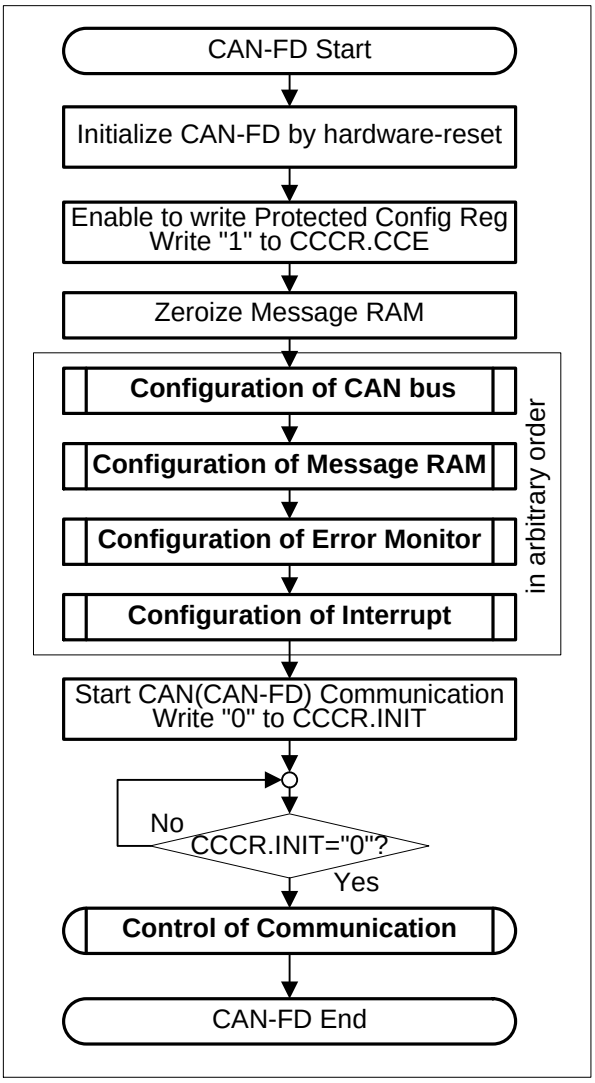
4. Setup Procedure Examples

This section provides examples of setup procedures for the CAN FD Controller.

4.1. Configuration of CAN Bus
4.2. Configuration of Message RAM
4.2.1. Configuration of ID Filter List
4.2.2. Configuration of Rx Buffer and Rx FIFO
4.2.3. Configuration of Tx Buffer and Tx FIFO/Queue
4.2.4. Configuration of ID Filter
4.3. Configuration of Error Monitor
4.4. Configuration of Interrupt
4.5. Control of Communication
4.5.1. CAN Transmit Mode Change
4.5.2. Configuration of Transmission Frame
4.6. Interrupt Handling Operation
4.6.1. Bus_Off Status Handling Operation
4.6.2. Message RAM Access Failure Handling Operation
4.6.2.1. CCCR.CCE Setting Operation
4.6.3. Bit Error Handling Operation
4.6.4. Tx Event FIFO Handling Operation
4.6.5. Dedicated Rx Buffer Handling Operation
4.6.6. High Priority Message Handling Operation
4.6.7. Rx FIFO Handling Operation

Figure 4-1 shows the general flow of a program. The following sections will describe each step in more detail.

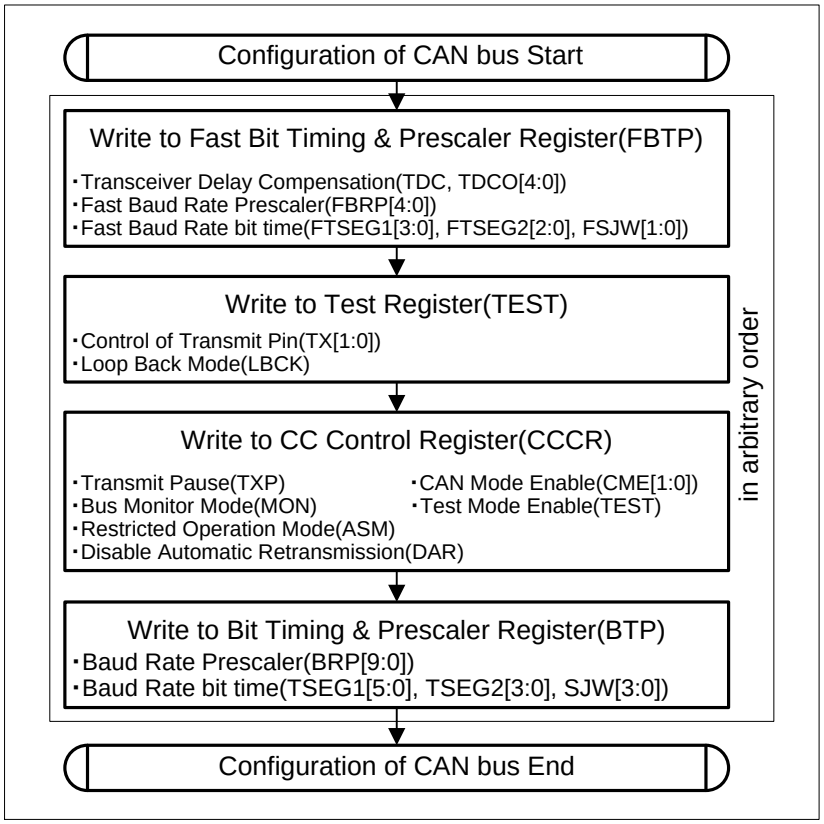
Figure 4-1 General Program Flow



4.1. Configuration of CAN Bus

Figure 4-2 lists what to configure in order to set up the CAN bus related functions.

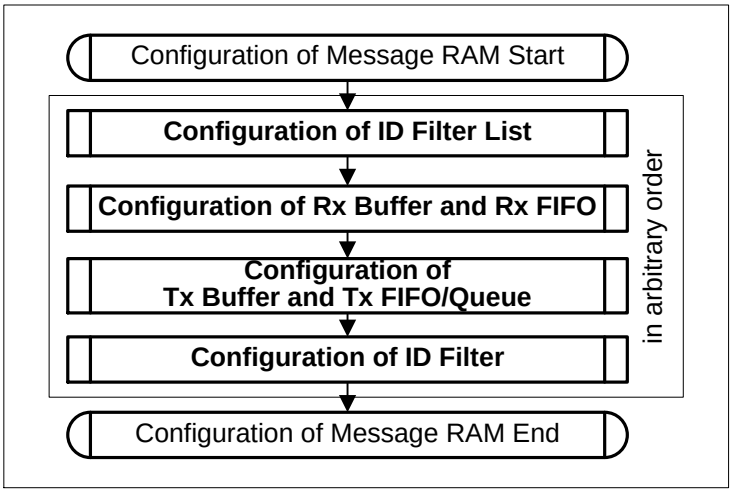
Figure 4-2 Flow Diagram "Configuration of CAN Bus"



4.2. Configuration of Message RAM

Figure 4-3 shows the overall configuration procedure of the Message RAM.

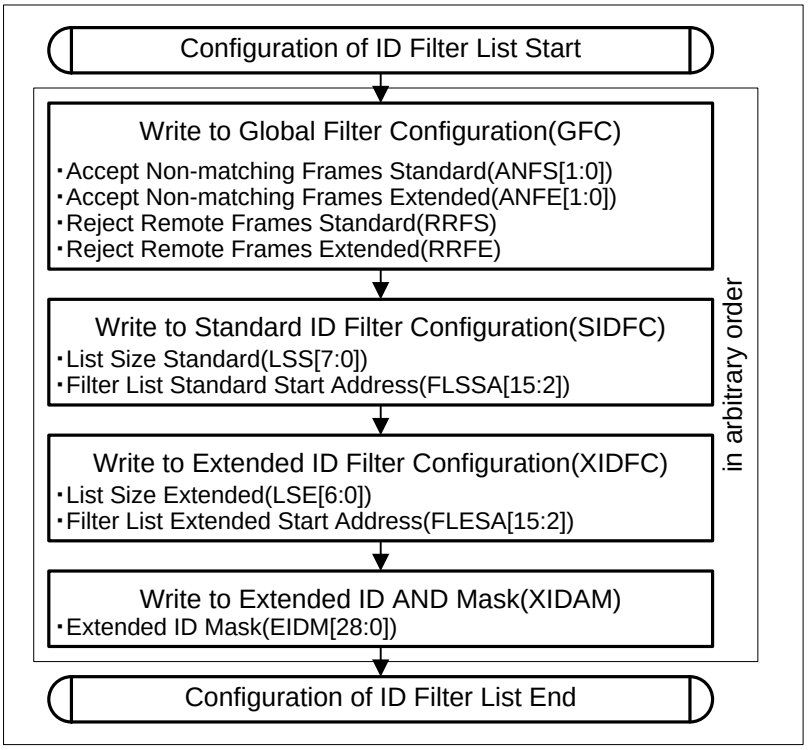
Figure 4-3 Flow Diagram "Configuration of Message RAM"



4.2.1. Configuration of ID Filter List

Figure 4-4 lists what to configure in order to set up the ID Filter lists.

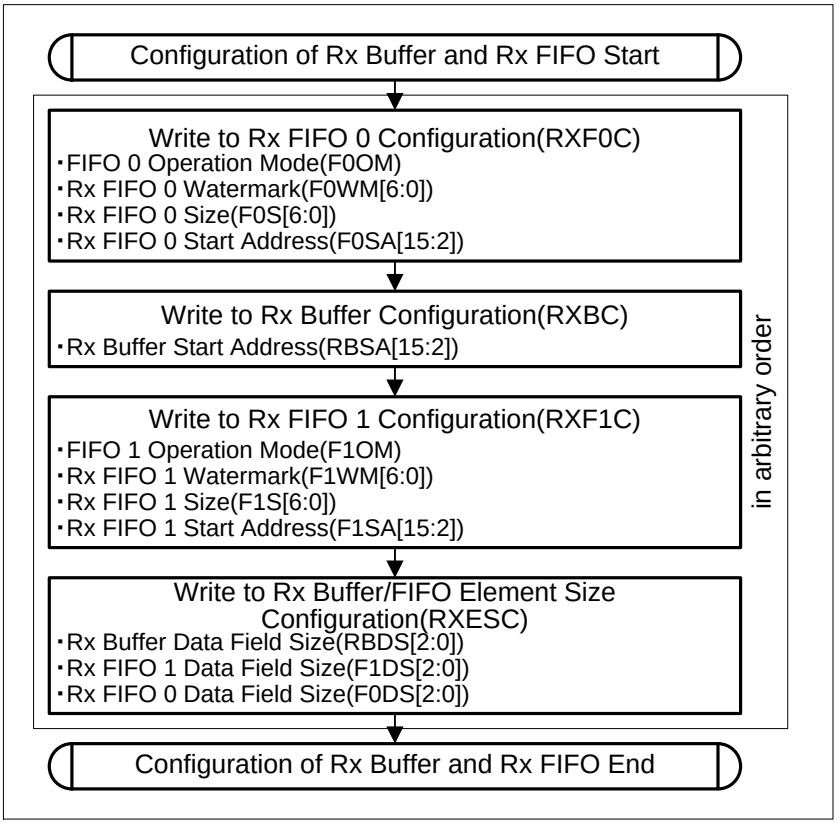
Figure 4-4 Flow Diagram "Configuration of ID Filter List"



4.2.2. Configuration of Rx Buffer and Rx FIFO

Figure 4-5 lists what to configure in order to set up the Dedicated Rx Buffers and Rx FIFOs.

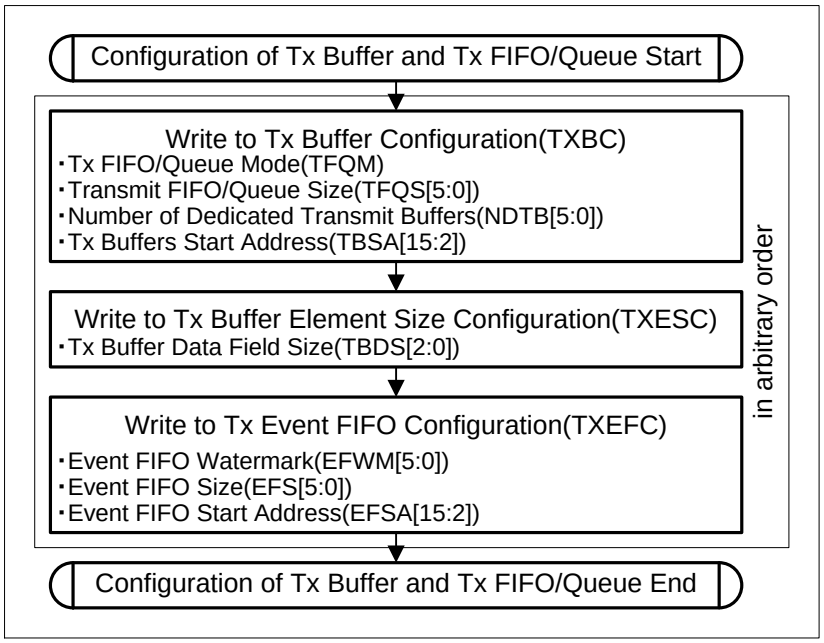
Figure 4-5 Flow Diagram "Configuration of Rx Buffer and Rx FIFO"



4.2.3. Configuration of Tx Buffer and Tx FIFO/Queue

Figure 4-6 lists what to configure in order to set up the Tx Buffers and Tx Event FIFO.

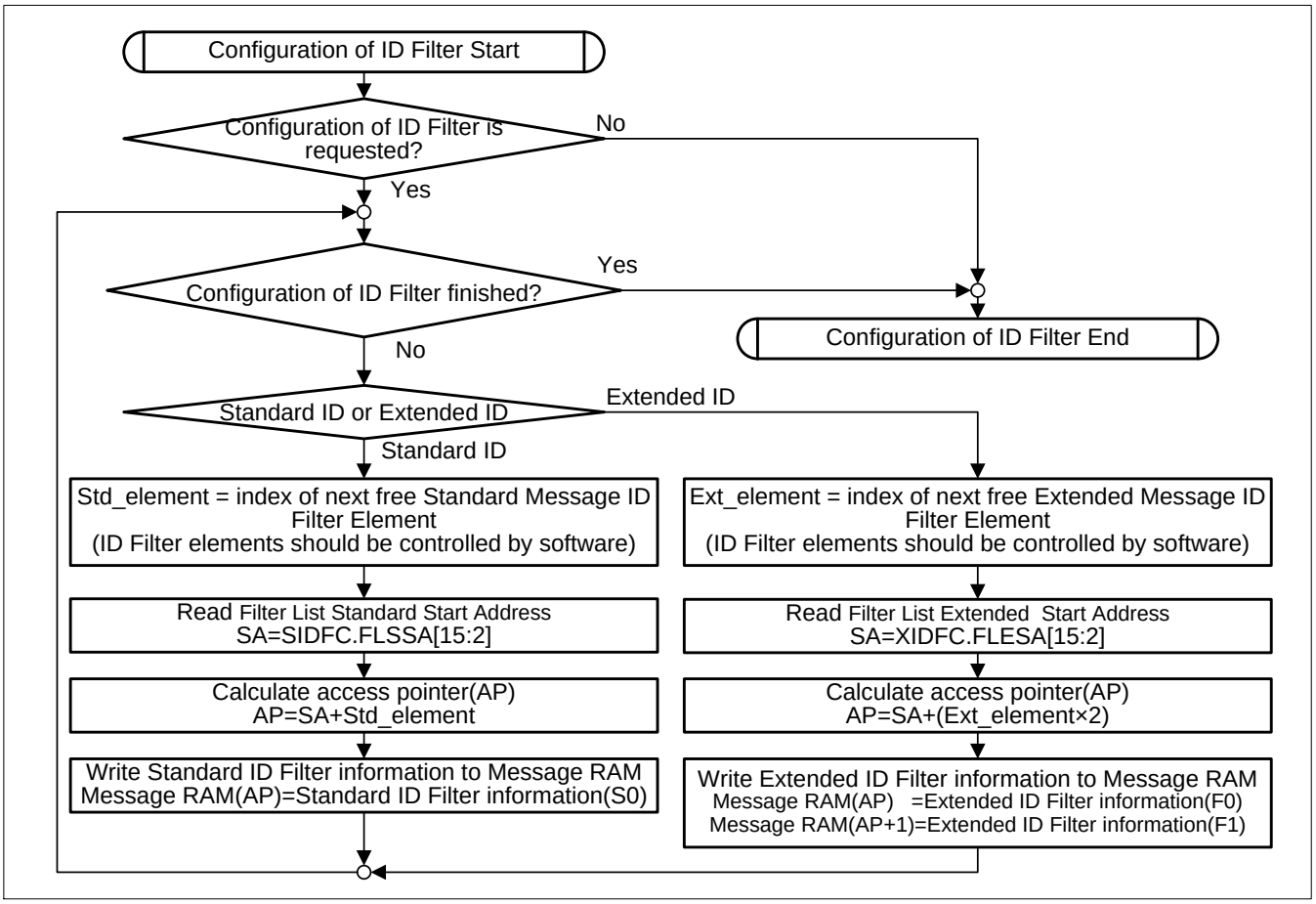
Figure 4-6 Flow Diagram "Configuration of Tx Buffer and Tx FIFO/Queue"



4.2.4. Configuration of ID Filter

Figure 4-7 shows the steps to set up an ID Filter.

Figure 4-7 Flow Diagram "Configuration of ID Filter"



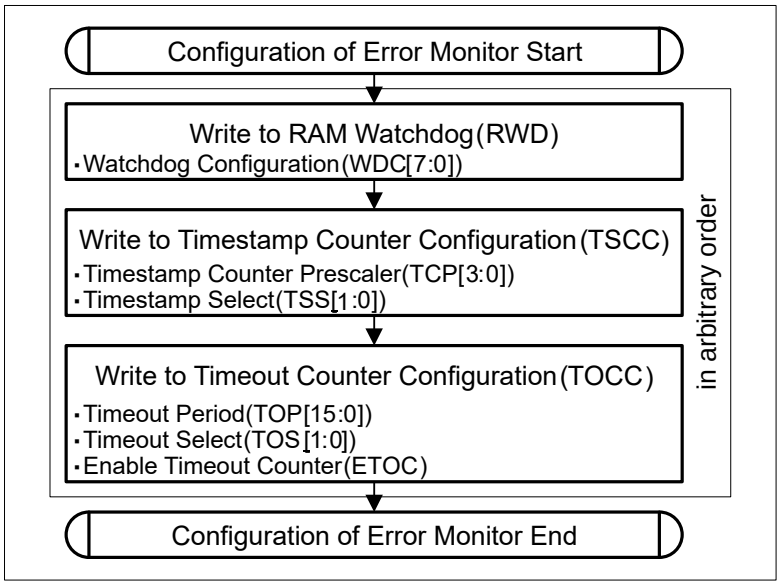
Note:

- *Use 32-bit word address values when calculating the pointers in the flow diagram above.*

4.3. Configuration of Error Monitor

Figure 4-8 lists the error monitoring functions that can be used.

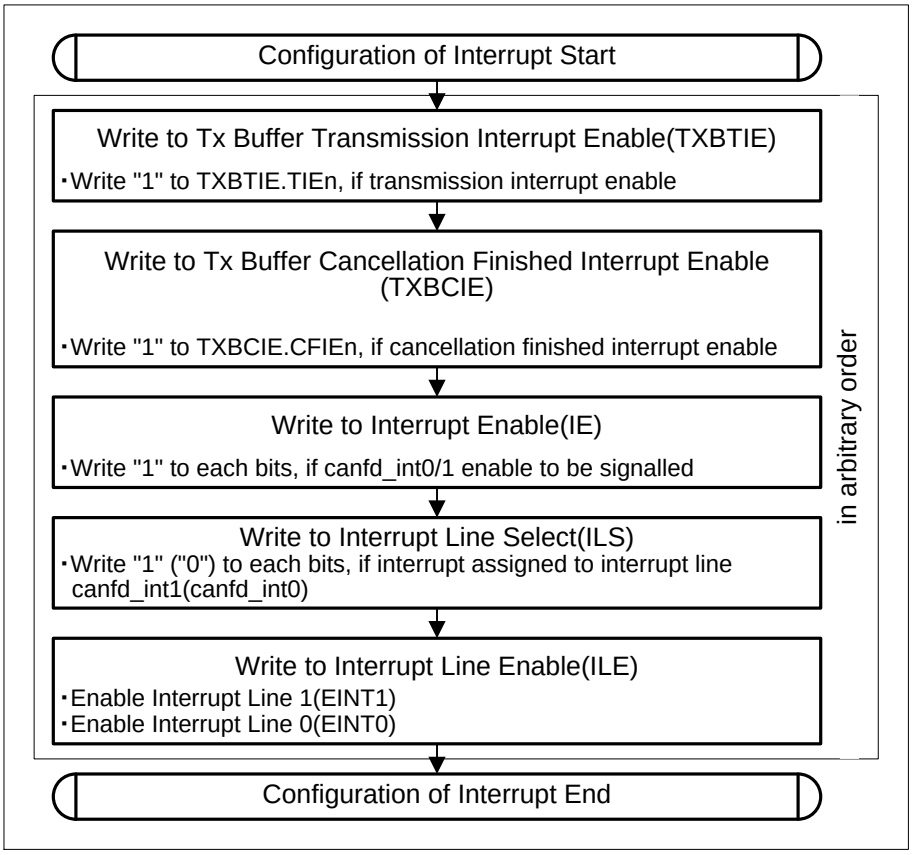
Figure 4-8 Flow Diagram "Configuration of Error Monitor"



4.4. Configuration of Interrupt

Figure 4-9 lists what to configure in order to set up the Interrupt related logic.

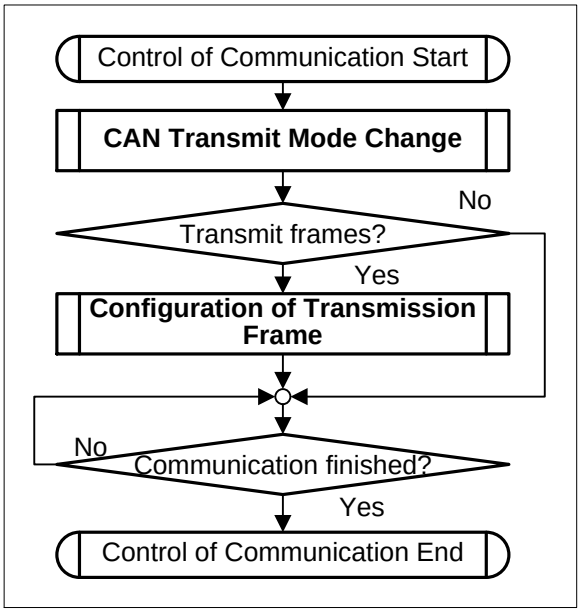
Figure 4-9 Flow Diagram "Configuration of Interrupt"



4.5. Control of Communication

Figure 4-10 shows the overall configuration procedure for communication on the CAN bus.

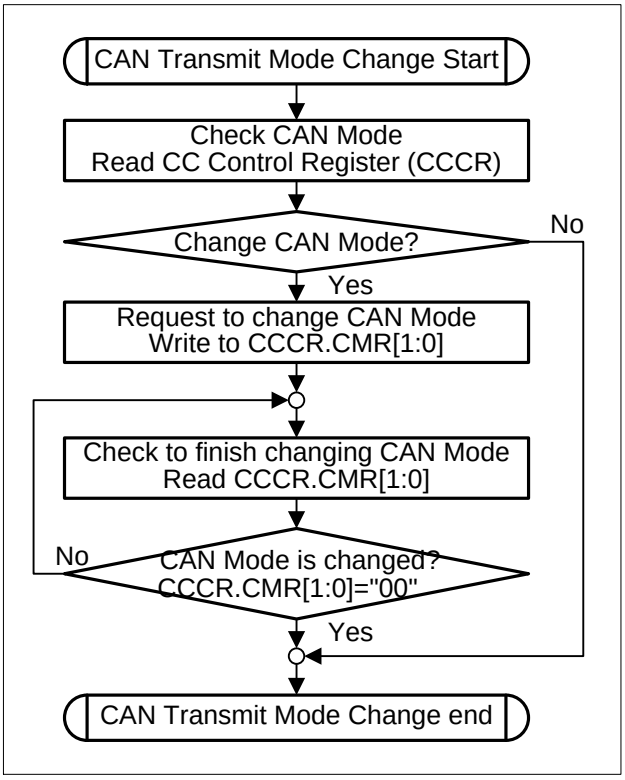
Figure 4-10 Flow Diagram "Control of Communication"



4.5.1. CAN Transmit Mode Change

Figure 4-11 shows how the CAN transmission properties CCCR.FDBS (CAN FD Bit Rate Switching) and CCCR.FDO (CAN FD Operation) can be changed for a specific CAN Mode Enable (CCCR.CME[1:0]) setting.

Figure 4-11 Flow Diagram "CAN Transmit Mode Change"



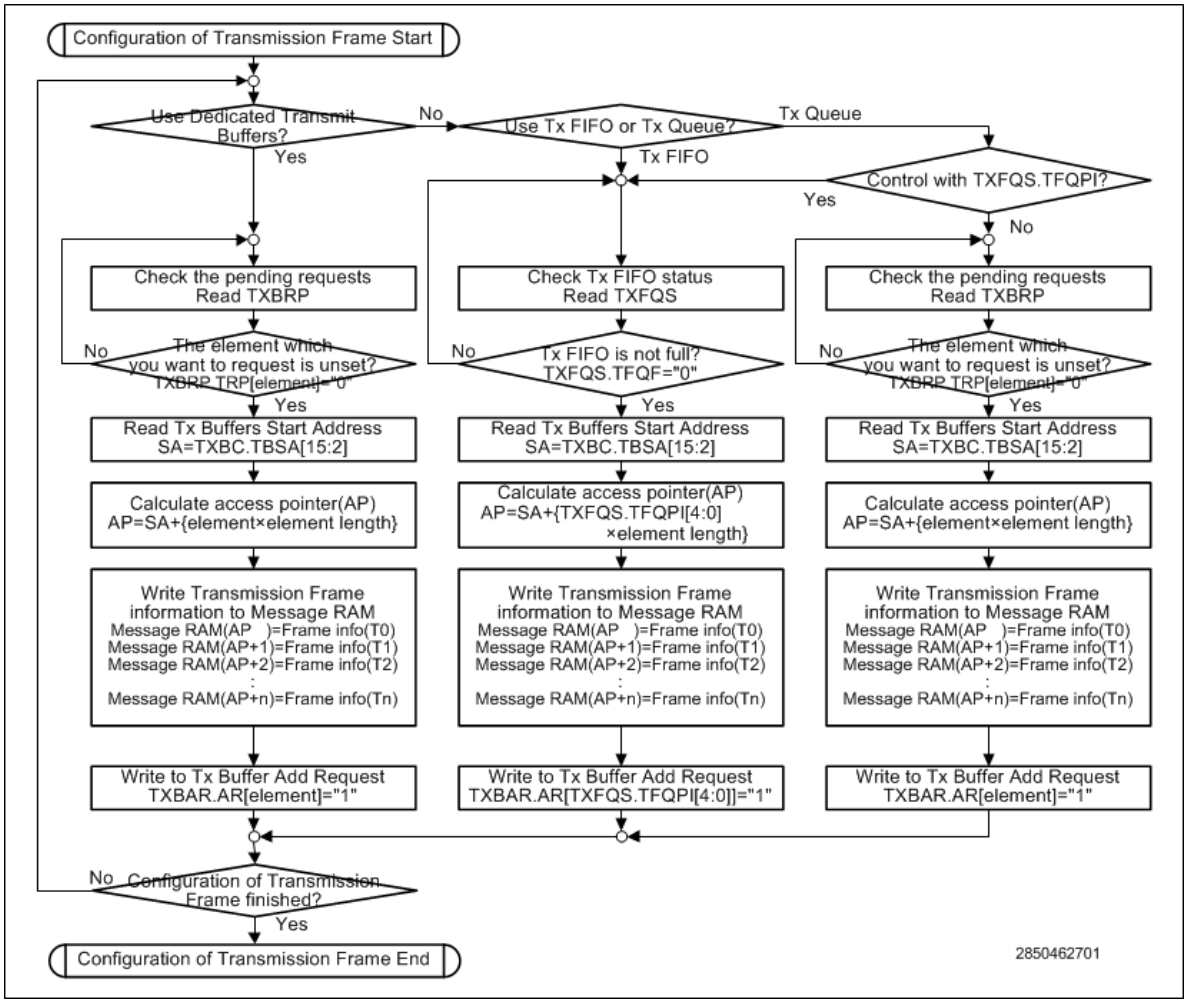
Note:

- *A mode change requested by writing to CCCR.CMR[1:0] is allowed only when no transmission requests are pending. If necessary, cancel pending transmission requests before changing the CAN operation mode.*

4.5.2. Configuration of Transmission Frame

Figure 4-12 shows how to set and transmit frames.

Figure 4-12 Flow Diagram "Configuration of Transmission Frame"



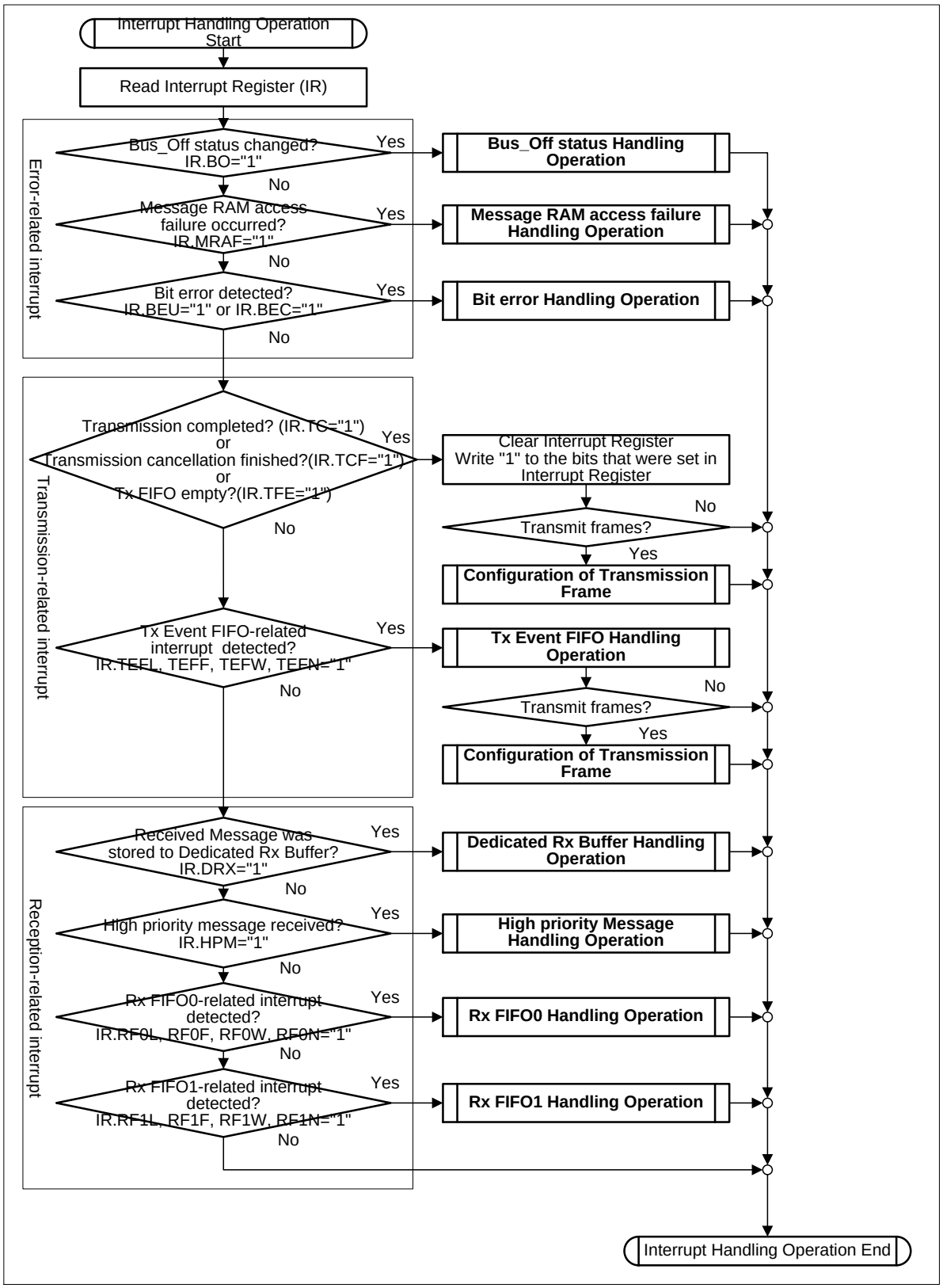
Notes:

- *Use 32-bit word address values when calculating the pointers in the flow diagram above.*
- *Dedicated Transmit Buffers are normally configured with a single specific identifier.*

4.6. Interrupt Handling Operation

Figure 4-13 shows the overall picture of how interrupts are processed.

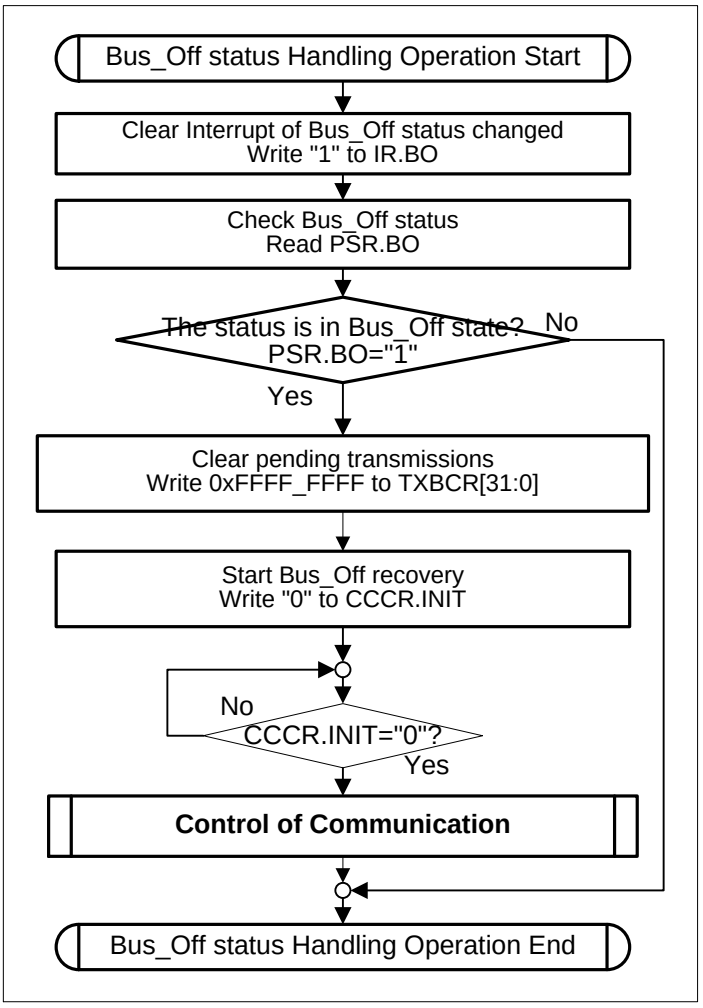
Figure 4-13 Flow Diagram "Interrupt Handling Operation"



4.6.1. Bus_Off Status Handling Operation

Figure 4-14 shows how to process a "Bus_Off Status" interrupt (IR.BO).

Figure 4-14 Flow Diagram "Bus_Off Status Handling Operation"



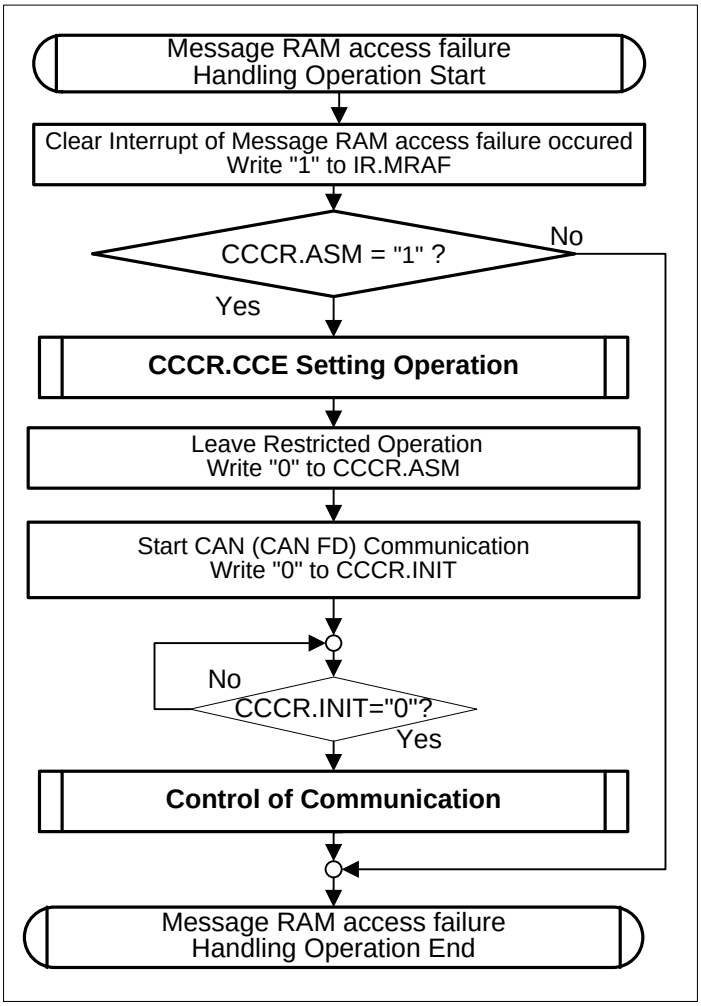
Note:

- *Once the Initialization bit (CCCR.INIT) has been cleared by the CPU, the CAN FD Controller will then wait for 129 occurrences of Bus Idle (129 × 11 consecutive recessive bits) before resuming normal operation.*

4.6.2. Message RAM Access Failure Handling Operation

Figure 4-15 shows how to process a "Message RAM Access Failure" interrupt (IR.MRAF).

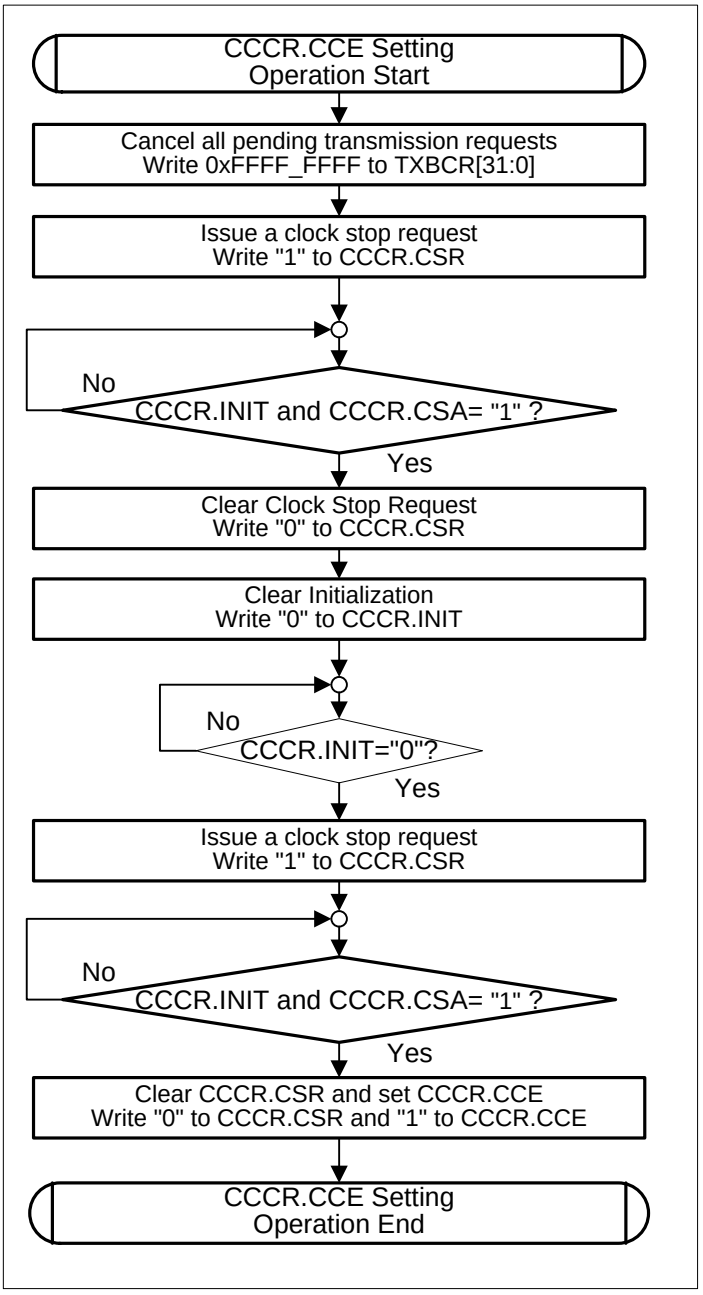
Figure 4-15 Flow Diagram "Message RAM Access Failure Handling Operation"



4.6.2.1 CCCR.CCE Setting Operation

Figure 4-16 shows how to process a "Setting CCCR.CCE".

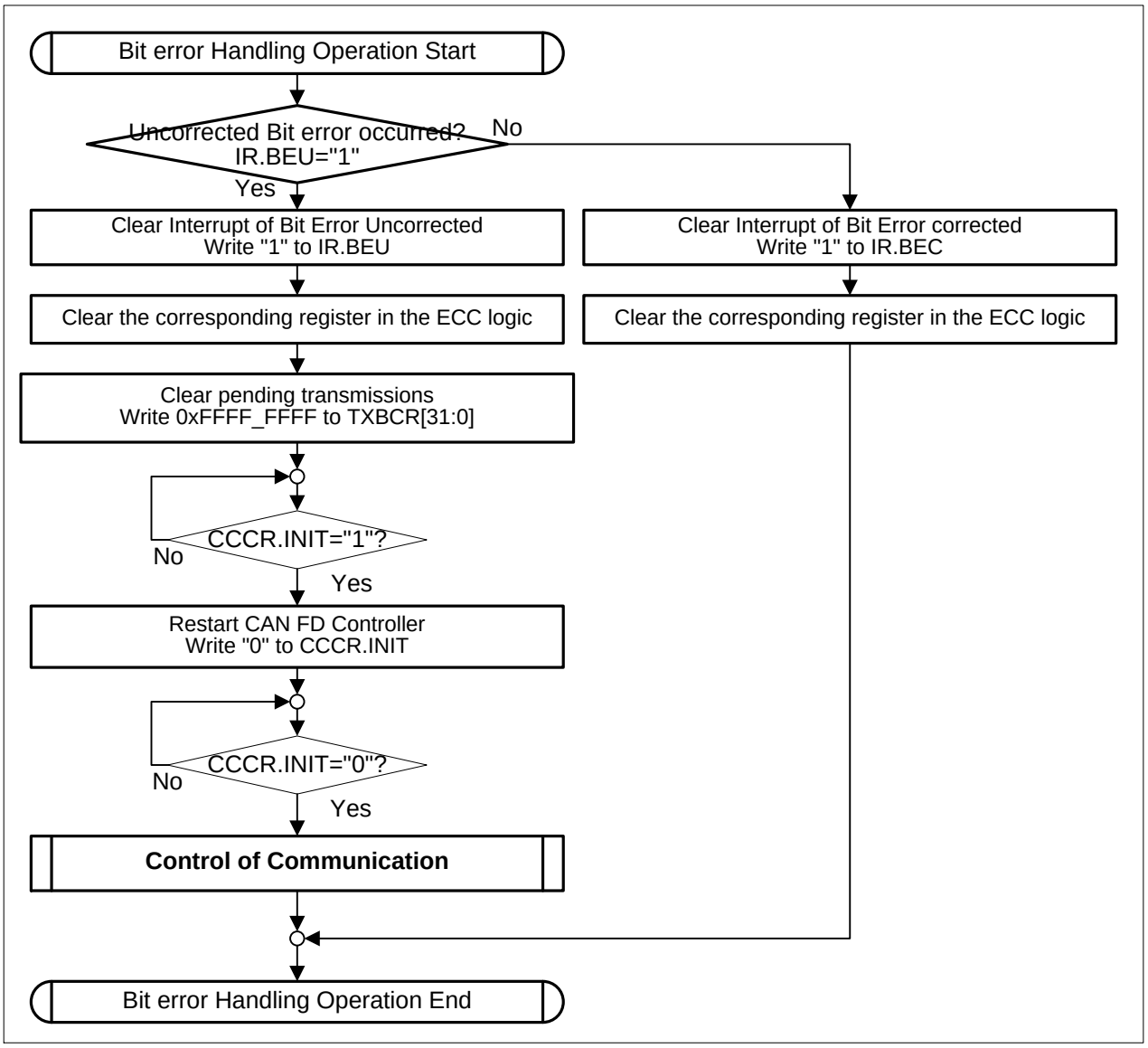
Figure 4-16 Flow Diagram "CCCR.CCE Setting Operation"



4.6.3. Bit Error Handling Operation

Figure 4-17 shows how to process the "Bit Error" interrupts (IR.BEU, IR.BEC).

Figure 4-17 Flow Diagram "Bit Error Handling Operation"



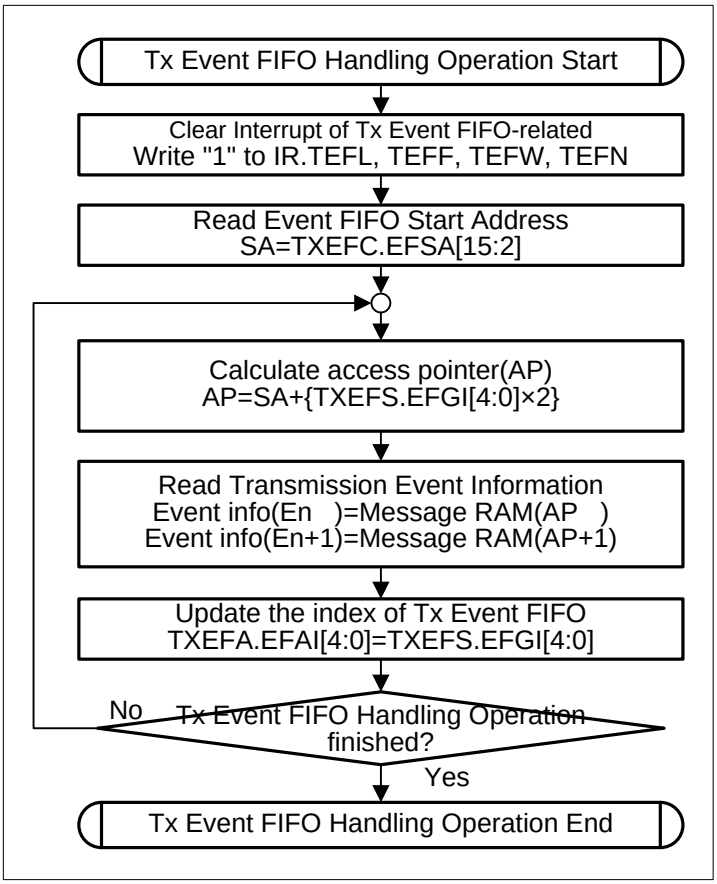
Note:

- *In case Bit Error Uncorrected IR.BEU occurs, due to the synchronization mechanism between the Bus clock and the CAN clock, there may be a delay until Initialization CCCR.INIT set to "1". Therefore the programmer has to assure that CCCR.INIT has been set to "1" by reading CCCR.INIT before resetting CCCR.INIT to "0".*

4.6.4. Tx Event FIFO Handling Operation

Figure 4-18 shows how to process Tx Event FIFO related interrupts.

Figure 4-18 Flow Diagram "Tx Event FIFO Handling Operation"



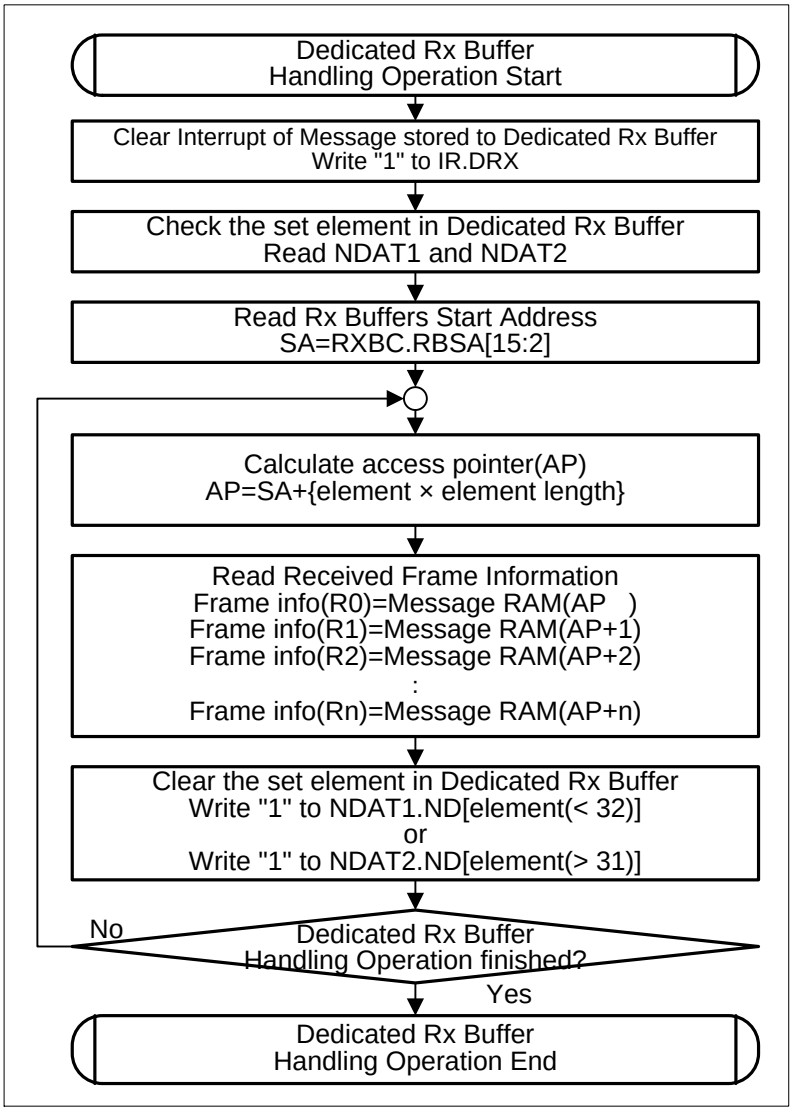
Note:

- *Use 32-bit word address values when calculating the pointers in the flow diagram above.*

4.6.5. Dedicated Rx Buffer Handling Operation

Figure 4-19 shows how to process a "Message stored to Dedicated Rx Buffer" interrupt (IR.DRX).

Figure 4-19 Flow Diagram "Dedicated Rx Buffer Handling Operation"



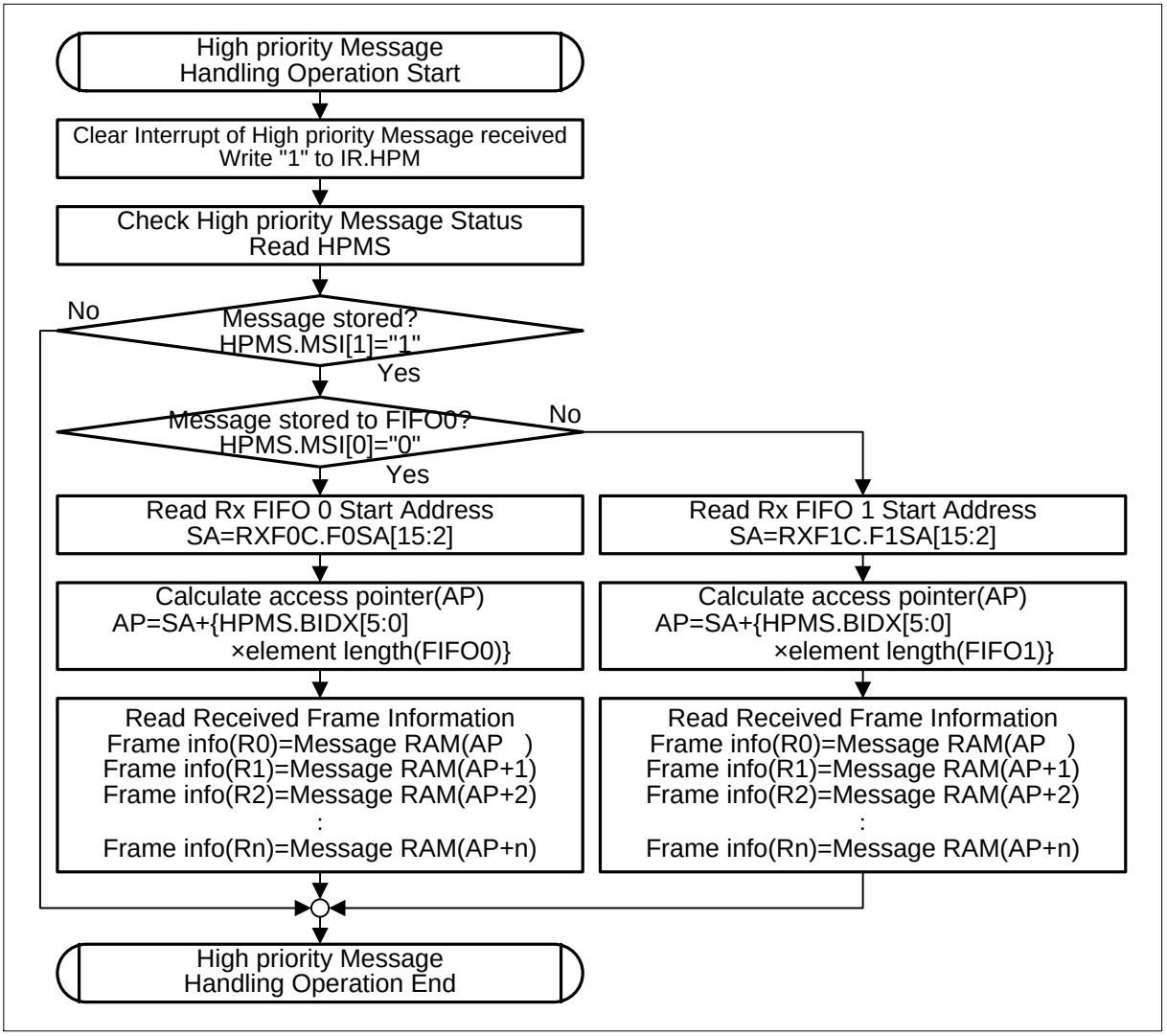
Note:

- *Use 32-bit word address values when calculating the pointers in the flow diagram above.*

4.6.6. High Priority Message Handling Operation

Figure 4-20 shows how to process a High Priority Message interrupt (IR.HPM).

Figure 4-20 Flow Diagram "High Priority Message Handling Operation"



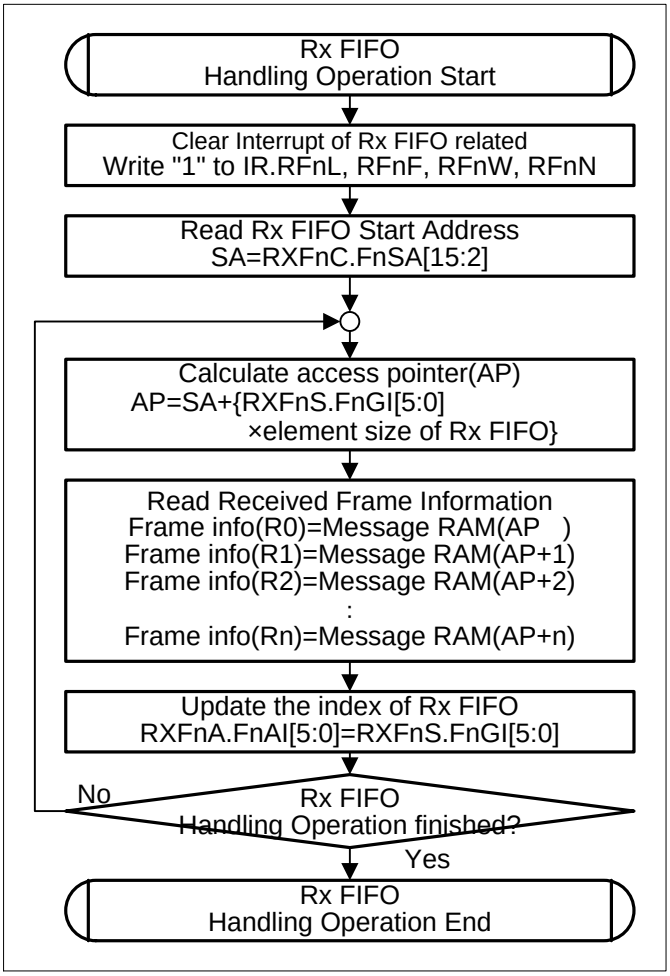
Note:

- *Use 32-bit word address values when calculating the pointers in the flow diagram above.*

4.6.7. Rx FIFO Handling Operation

Figure 4-21 shows how to process Rx FIFO related interrupts.

Figure 4-21 Flow Diagram "Rx FIFO Handling Operation"



Notes:

- *Use 32-bit word address values when calculating the pointers in the flow diagram above.*
- *The "n"s in the register/bit names (e.g. "RXFnS.FnGI") stand for "0" or "1".*
- *If a high priority message is received and processed by the "High priority Message Handling Operation" flow, this flow (i.e. Rx FIFO Handling Operation) may be skipped for that particular message. The following message received by the Rx FIFO may resume with this flow again.*

5. Registers

This section explains the configuration and functions of the registers used for the CAN FD Controller.

Hardware Reset Description

After hardware reset, the registers of the CAN FD Controller hold the reset values as indicated in each relative register description. Additionally the Bus_Off state is reset and the output canfd_tx is set to recessive. Setting the Initialization bit CCCR.INIT to "1" in the CC Control Register enables software initialization. The CAN FD Controller does not influence the CAN bus until the CPU resets CCCR.INIT to "0".

Register Map

The CAN FD Controller allocates an address space of 256 bytes for the registers.

All registers are organized as 32-bit registers. The registers are accessible by the CPU using a data width of 8 bit (byte access), 16 bit (half-word access), or 32 bit (word access).

Note:

- *As an exception, the Protocol Status Register (PSR) does not allow byte accessing.*

Write access by the CPU to several of the registers/bits that configure the basic operation of the CAN FD Controller is possible only with Configuration Change Enable CCCR.CCE= 1 AND Initialization CCCR.INIT= 1.

There is a delay from writing to a command register until the update of the related status register bits due to clock domain crossing.

Table 5-1 List of Registers for the CAN FD Controller

Abbreviation	Register Name	Reference
CREL	Core Release Register	5.1
ENDN	Endian Register	5.2
FBTP	Fast Bit Timing & Prescaler Register	5.3
TEST	Test Register	5.4
RWD	RAM Watchdog	5.5
CCCR	CC Control Register	5.6
BTP	Bit Timing & Prescaler Register	5.7
TSCC	Timestamp Counter Configuration	5.8
TSCV	Timestamp Counter Value	5.9
TOCC	Timeout Counter Configuration	5.10
TOCV	Timeout Counter Value	5.11
ECR	Error Counter Register	5.12
PSR	Protocol Status Register	5.13
IR	Interrupt Register	5.14
IE	Interrupt Enable	5.15
ILS	Interrupt Line Select	5.16
ILE	Interrupt Line Enable	5.17
GFC	Global Filter Configuration	5.18
SIDFC	Standard ID Filter Configuration	5.19
XIDFC	Extended ID Filter Configuration	5.20

Abbreviation	Register Name	Reference
XIDAM	Extended ID AND Mask	5.21
HPMS	High Priority Message Status	5.22
NDAT1	New Data 1	5.23
NDAT2	New Data 2	5.24
RXF0C	Rx FIFO 0 Configuration	5.25
RXF0S	Rx FIFO 0 Status	5.26
RXF0A	Rx FIFO 0 Acknowledge	5.27
RXBC	Rx Buffer Configuration	5.28
RXF1C	Rx FIFO 1 Configuration	5.29
RXF1S	Rx FIFO 1 Status	5.30
RXF1A	Rx FIFO 1 Acknowledge	5.31
RXESC	Rx Buffer/FIFO Element Size Configuration	5.32
TXBC	Tx Buffer Configuration	5.33
TXFQS	Tx FIFO/Queue Status	5.34
TXESC	Tx Buffer Element Size Configuration	5.35
TXBRP	Tx Buffer Request Pending	5.36
TXBAR	Tx Buffer Add Request	5.37
TXBCR	Tx Buffer Cancellation Request	5.38
TXBTO	Tx Buffer Transmission Occurred	5.39
TXBCF	Tx Buffer Cancellation Finished	5.40
TXBTIE	Tx Buffer Transmission Interrupt Enable	5.41
TXBCIE	Tx Buffer Cancellation Finished Interrupt Enable	5.42
TXEFC	Tx Event FIFO Configuration	5.43
TXEFS	Tx Event FIFO Status	5.44
TXEFA	Tx Event FIFO Acknowledge	5.45

Table 5-2 Register Map CAN FD CONTROLLER

MCU Config Group (Channel No.: 0, 1, and 2)

Offset	Register Name / Initial Value			
	+3	+2	+1	+0
0000_0000	MCG_CANFDx_CREL 00110000 00010011 00000101 00000110			
0000_0004	MCG_CANFDx_ENDN 10000111 01100101 01000011 00100001			
0000_0008	-			
0000_000C	MCG_CANFDx_FBTP 00000000 00000000 00001010 00110011			
0000_0010	MCG_CANFDx_TEST 00000000 00000000 00000000 *0000000			
0000_0014	MCG_CANFDx_RWD 00000000 00000000 00000000 00000000			
0000_0018	MCG_CANFDx_CCCR 00000000 00000000 00000000 00000001			
0000_001C	MCG_CANFDx_BTP 00000000 00000000 00001010 00110011			
0000_0020	MCG_CANFDx_TSCC 00000000 00000000 00000000 00000000			
0000_0024	MCG_CANFDx_TSCV 00000000 00000000 00000000 00000000			
0000_0028	MCG_CANFDx_TOCC 11111111 11111111 00000000 00000000			
0000_002C	MCG_CANFDx_TOCV 00000000 00000000 11111111 11111111			
0000_0030 | 0000_003C	-			
0000_0040	MCG_CANFDx_ECR XXXXXXXX 00000000 00000000 00000000			
0000_0044	MCG_CANFDx_PSR XXXXXXXX XXXXXXXX XX000111 00000111			
0000_0048 0000_004C	-			
0000_0050	MCG_CANFDx_IR 00000000 00000000 00000000 00000000			
0000_0054	MCG_CANFDx_IE 00000000 00000000 00000000 00000000			
0000_0058	MCG_CANFDx_ILS 00000000 00000000 00000000 00000000			
0000_005C	MCG_CANFDx_ILE 00000000 00000000 00000000 00000000			

Offset	Register Name / Initial Value			
	+3	+2	+1	+0
0000_0060 0000_007C	-			
0000_0080	MCG_CANFDx_GFC 00000000 00000000 00000000 00000000			
0000_0084	MCG_CANFDx_SIDFC 00000000 00000000 00000000 00000000			
0000_0088	MCG_CANFDx_XIDFC 00000000 00000000 00000000 00000000			
0000_008C	-			
0000_0090	MCG_CANFDx_XIDAM 00011111 11111111 11111111 11111111			
0000_0094	MCG_CANFDx_HPMS XXXXXXXX XXXXXXXX 00000000 00000000			
0000_0098	MCG_CANFDx_NDAT1 00000000 00000000 00000000 00000000			
0000_009C	MCG_CANFDx_NDAT2 00000000 00000000 00000000 00000000			
0000_00A0	MCG_CANFDx_RXF0C 00000000 00000000 00000000 00000000			
0000_00A4	MCG_CANFDx_RXF0S XXXXXX00 XX000000 XX000000 X0000000			
0000_00A8	MCG_CANFDx_RXF0A 00000000 00000000 00000000 00000000			
0000_00AC	MCG_CANFDx_RXBC 00000000 00000000 00000000 00000000			
0000_00B0	MCG_CANFDx_RXF1C 00000000 00000000 00000000 00000000			
0000_00B4	MCG_CANFDx_RXF1S 00XXXX00 XX000000 XX000000 X0000000			
0000_00B8	MCG_CANFDx_RXF1A 00000000 00000000 00000000 00000000			
0000_00BC	MCG_CANFDx_RXESC 00000000 00000000 00000000 00000000			
0000_00C0	MCG_CANFDx_TXBC 00000000 00000000 00000000 00000000			
0000_00C4	MCG_CANFDx_TXFQS XXXXXXXX XX000000 XXX00000 XX000000			
0000_00C8	MCG_CANFDx_TXESC 00000000 00000000 00000000 00000000			
0000_00CC	MCG_CANFDx_TXBRP 00000000 00000000 00000000 00000000			

Offset	Register Name / Initial Value			
	+3	+2	+1	+0
0000_00D0	MCG_CANFDx_TXBAR 00000000 00000000 00000000 00000000			
0000_00D4	MCG_CANFDx_TXBCR 00000000 00000000 00000000 00000000			
0000_00D8	MCG_CANFDx_TXBTO 00000000 00000000 00000000 00000000			
0000_00DC	MCG_CANFDx_TXBCF 00000000 00000000 00000000 00000000			
0000_00E0	MCG_CANFDx_TXBTIE 00000000 00000000 00000000 00000000			
0000_00E4	MCG_CANFDx_TXBCIE 00000000 00000000 00000000 00000000			
0000_00E8 0000_00EC	-			
0000_00F0	MCG_CANFDx_TXEFC 00000000 00000000 00000000 00000000			
0000_00F4	MCG_CANFDx_TXEFS XXXXXX00 XXX00000 XXX00000 XX000000			
0000_00F8	MCG_CANFDx_TXEFA 00000000 00000000 00000000 00000000			
0000_00FC 0000_01FC	-			
0000_0200	MCG_CANFDx_FDSEAR 00000000 00000000		MCG_CANFDx_FDESR 00000000	MCG_CANFDx_FDECR 00000000
0000_0204	MCG_CANFDx_FDDEAR 00000000 00000000		MCG_CANFDx_FDESCR 00000000	-
0000_0208 0000_02FC	-			
0000_0300 0000_7FFC	-			
0000_8000 0000_BFFC	CAN_RAM chx			
0000_C000 0000_FFFC	-			

Notes:

- *x is the channel number. (0 to 2)*
- *"*": Initial value "0" or "1" according to the setting (designed by higher layer)*
- *The number of circuits is designed by a higher layer.*

Common Peripheral #0 Group (Channel No.: 0, 1, 2, 3, and 4)

Offset	Register Name / Initial Value			
	+3	+2	+1	+0
0000_0000	CPG_CANFDx_CREL 00110000 00010011 00000101 00000110			
0000_0004	CPG_CANFDx_ENDN 10000111 01100101 01000011 00100001			
0000_0008	-			
0000_000C	CPG_CANFDx_FBTP 00000000 00000000 00001010 00110011			
0000_0010	CPG_CANFDx_TEST 00000000 00000000 00000000 *0000000			
0000_0014	CPG_CANFDx_RWD 00000000 00000000 00000000 00000000			
0000_0018	CPG_CANFDx_CCCR 00000000 00000000 00000000 00000001			
0000_001C	CPG_CANFDx_BTP 00000000 00000000 00001010 00110011			
0000_0020	CPG_CANFDx_TSCC 00000000 00000000 00000000 00000000			
0000_0024	CPG_CANFDx_TSCV 00000000 00000000 00000000 00000000			
0000_0028	CPG_CANFDx_TOCC 11111111 11111111 00000000 00000000			
0000_002C	CPG_CANFDx_TOCV 00000000 00000000 11111111 11111111			
0000_0030 | 0000_003C	-			
0000_0040	CPG_CANFDx_ECR XXXXXXXX 00000000 00000000 00000000			
0000_0044	CPG_CANFDx_PSR XXXXXXXX XXXXXXXX XX000111 00000111			
0000_0048 0000_004C	-			
0000_0050	CPG_CANFDx_IR 00000000 00000000 00000000 00000000			
0000_0054	CPG_CANFDx_IE 00000000 00000000 00000000 00000000			
0000_0058	CPG_CANFDx_ILS 00000000 00000000 00000000 00000000			
0000_005C	CPG_CANFDx_ILE 00000000 00000000 00000000 00000000			
0000_0060 0000_007C	-			

Offset	Register Name / Initial Value			
	+3	+2	+1	+0
0000_0080	CPG_CANFDx_GFC 00000000 00000000 00000000 00000000			
0000_0084	CPG_CANFDx_SIDFC 00000000 00000000 00000000 00000000			
0000_0088	CPG_CANFDx_XIDFC 00000000 00000000 00000000 00000000			
0000_008C	-			
0000_0090	CPG_CANFDx_XIDAM 00011111 11111111 11111111 11111111			
0000_0094	CPG_CANFDx_HPMS XXXXXXXX XXXXXXXX 00000000 00000000			
0000_0098	CPG_CANFDx_NDAT1 00000000 00000000 00000000 00000000			
0000_009C	CPG_CANFDx_NDAT2 00000000 00000000 00000000 00000000			
0000_00A0	CPG_CANFDx_RXF0C 00000000 00000000 00000000 00000000			
0000_00A4	CPG_CANFDx_RXF0S XXXXXX00 XX000000 XX000000 X0000000			
0000_00A8	CPG_CANFDx_RXF0A 00000000 00000000 00000000 00000000			
0000_00AC	CPG_CANFDx_RXBC 00000000 00000000 00000000 00000000			
0000_00B0	CPG_CANFDx_RXF1C 00000000 00000000 00000000 00000000			
0000_00B4	CPG_CANFDx_RXF1S 00XXXX00 XX000000 XX000000 X0000000			
0000_00B8	CPG_CANFDx_RXF1A 00000000 00000000 00000000 00000000			
0000_00BC	CPG_CANFDx_RXESC 00000000 00000000 00000000 00000000			
0000_00C0	CPG_CANFDx_TXBC 00000000 00000000 00000000 00000000			
0000_00C4	CPG_CANFDx_TXFQS XXXXXXXX XX000000 XXX00000 XX000000			
0000_00C8	CPG_CANFDx_TXESC 00000000 00000000 00000000 00000000			
0000_00CC	CPG_CANFDx_TXBRP 00000000 00000000 00000000 00000000			
0000_00D0	CPG_CANFDx_TXBAR 00000000 00000000 00000000 00000000			
0000_00D4	CPG_CANFDx_TXBCR 00000000 00000000 00000000 00000000			

Offset	Register Name / Initial Value			
	+3	+2	+1	+0
0000_00D8	CPG_CANFDx_TXBTO 00000000 00000000 00000000 00000000			
0000_00DC	CPG_CANFDx_TXBCF 00000000 00000000 00000000 00000000			
0000_00E0	CPG_CANFDx_TXBTIE 00000000 00000000 00000000 00000000			
0000_00E4	CPG_CANFDx_TXBCIE 00000000 00000000 00000000 00000000			
0000_00E8 0000_00EC	-			
0000_00F0	CPG_CANFDx_TXEFC 00000000 00000000 00000000 00000000			
0000_00F4	CPG_CANFDx_TXEFS XXXXXX00 XXX00000 XXX00000 XX000000			
0000_00F8	CPG_CANFDx_TXEFA 00000000 00000000 00000000 00000000			
0000_00FC 0000_01FC	-			
0000_0200	CPG_CANFDx_FDSEAR 00000000 00000000		CPG_CANFDx_FDESR 00000000	CPG_CANFDx_FDECR 00000000
0000_0204	CPG_CANFDx_FDDEAR 00000000 00000000		CPG_CANFDx_FDESCR 00000000	-
0000_0208 0000_02FC	-			
0000_0300 0000_7FFC	-			
0000_8000 0000_BFFC	CAN_RAM chx			
0000_C000 0000_FFFC	-			

Notes:

- *x is the channel number. (0 to 4)*
- *"*": Initial value "0" or "1" according to the setting (designed by higher layer)*
- *The number of circuits is designed by a higher layer.*

Table 5-3 Register Mirror Area CAN FD CONTROLLER

Address of Register Mirror Area	Mirrored Peripheral	Mirror Area Behavior	PPU
0000_C000 to 0000_FFFF	MCU_CONFIG CAN FD ch.0	Accessing this region results to the access to MCU_CONFIG CAN FD ch.0 Message RAM area (0000_0000 to 0000_BFFF)	This area is covered by PPU #47 as well as the mirrored peripheral.
0000_C000 to 0000_FFFF	MCU_CONFIG CAN FD ch.1	Accessing this region results to the access to MCU_CONFIG CAN FD ch.1 Message RAM area (0000_0000 to 0000_BFFF)	This area is covered by PPU #48 as well as the mirrored peripheral.
0000_C000 to 0000_FFFF	MCU_CONFIG CAN FD ch.2	Accessing this region results to the access to MCU_CONFIG CAN FD ch.2 Message RAM area (0000_0000 to 0000_BFFF)	This area is covered by PPU #49 as well as the mirrored peripheral.
0000_C000 to 0000_FFFF	CommonPERI#0 CAN FD ch.0	Accessing this region results to the access to CommonPERI#0 CAN FD ch.0 Message RAM area (0000_0000 to 0000_BFFF)	This area is covered by PPU #256 as well as the mirrored peripheral.
0000_C000 to 0000_FFFF	CommonPERI#0 CAN FD ch.1	Accessing this region results to the access to CommonPERI#0 CAN FD ch.1 Message RAM area (0000_0000 to 0000_BFFF)	This area is covered by PPU #257 as well as the mirrored peripheral.
0000_C000 to 0000_FFFF	CommonPERI#0 CAN FD ch.2	Accessing this region results to the access to CommonPERI#0 CAN FD ch.2 Message RAM area (0000_0000 to 0000_BFFF)	This area is covered by PPU #258 as well as the mirrored peripheral.
0000_C000 to 0000_FFFF	CommonPERI#0 CAN FD ch.3	Accessing this region results to the access to CommonPERI#0 CAN FD ch.3 Message RAM area (0000_0000 to 0000_BFFF)	This area is covered by PPU #259 as well as the mirrored peripheral.
0000_C000 to 0000_FFFF	CommonPERI#0 CAN FD ch.4	Accessing this region results to the access to CommonPERI#0 CAN FD ch.4 Message RAM area (0000_0000 to 0000_BFFF)	This area is covered by PPU #260 as well as the mirrored peripheral.

5.1. Core Release Register (MCG_CANFDx_CREL, CPG_CANFDx_CREL)

The Core Release Register displays the revision of the CAN FD Controller.

bit	31	30	29	28	27	26	25	24
Field	REL[3:0]				STEP[3:0]			
R/W Attribute	R,WX				R,WX			
Protection Attribute	-				-			
Initial value	0x3				0x0			

bit	23	22	21	20	19	18	17	16
Field	SUBSTEP[3:0]				YEAR[3:0]			
R/W Attribute	R,WX				R,WX			
Protection Attribute	-				-			
Initial value	0x1				0x3			

bit	15	14	13	12	11	10	9	8
Field	MON[7:0]							
R/W Attribute	R,WX							
Protection Attribute	-							
Initial value	0x05							

bit	7	6	5	4	3	2	1	0
Field	DAY[7:0]							
R/W Attribute	R,WX							
Protection Attribute	-							
Initial value	0x06							

[bit31:28] REL[3:0]: Core Release
One digit, BCD-coded.

[bit27:24] STEP[3:0]: Step of Core Release
One digit, BCD-coded.

[bit23:20] SUBSTEP[3:0]: Sub-step of Core Release
One digit, BCD-coded.

[bit19:16] YEAR[3:0]: Time Stamp Year
One digit, BCD-coded.

[bit15:8] MON[7:0]: Time Stamp Month
Two digits, BCD-coded.

[bit7:0] DAY[7:0]: Time Stamp Day
Two digits, BCD-coded.

Table 5-4 Example for Coding of Revisions

Release	Step	SubStep	Year	Month	Day	Name
3	0	1	3	05	06	Revision 3.0.1, Date 2013/05/06

5.2. Endian Register (MCG_CANFDx_ENDN, CPG_CANFDx_ENDN)

The Endian Register can be used to check the endian-ness of the CAN FD Controller when accessed by the CPU.

bit	31	30	29	28	27	26	25	24
Field	ETV[31:24]							
R/W Attribute	R,WX							
Protection Attribute	-							
Initial value	0x87							

bit	23	22	21	20	19	18	17	16
Field	ETV[23:16]							
R/W Attribute	R,WX							
Protection Attribute	-							
Initial value	0x65							

bit	15	14	13	12	11	10	9	8
Field	ETV[15:8]							
R/W Attribute	R,WX							
Protection Attribute	-							
Initial value	0x43							

bit	7	6	5	4	3	2	1	0
Field	ETV[7:0]							
R/W Attribute	R,WX							
Protection Attribute	-							
Initial value	0x21							

[bit31:0] ETV[31:0]: Endian-ness Test Value

The endian-ness test value is 0x87654321.

5.3. Fast Bit Timing & Prescaler Register (MCG_CANFDx_FBTP, CPG_CANFDx_FBTP)

The Fast Bit Timing & Prescaler Register configures the fast bit time and the offset value for Transceiver Delay Compensation.

This register is only writable if bits Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) are set.

See "3.1.4.Transceiver Delay Compensation" for a description of the TDCO[4:0] and TDC fields.

See "3.7.Configuring the CAN Bit Timing" for a description of all other fields.

Notes:

- *With a CAN clock (canfd_cclk) of 8 MHz, the reset value of 0x00000A33 configures the CAN FD Controller for a FD bit rate of 500 kBit/s.*
- *The bit rate configured for the CAN FD data phase via the Fast Bit Timing & Prescaler Register (FBTP) must be higher or equal to the bit rate configured for the arbitration phase via the Bit Timing & Prescaler Register (BTP).*

bit	31	30	29	28	27	26	25	24
Field	Reserved			TDCO[4:0]				
R/W Attribute	R0,W0			R/W				
Protection Attribute	-			-				
Initial value	000			00000				

bit	23	22	21	20	19	18	17	16
Field	TDC	Reserved		FBRP[4:0]				
R/W Attribute	R/W	R0,W0		R/W				
Protection Attribute	-	-		-				
Initial value	0	00		00000				

Bit	15	14	13	12	11	10	9	8
Field	Reserved				FTSEG1[3:0]			
R/W Attribute	R0,W0				R/W			
Protection Attribute	-				-			
Initial value	0000				0xA			

bit	7	6	5	4	3	2	1	0
Field	Reserved	FTSEG2[2:0]			Reserved		FSJW[1:0]	
R/W Attribute	R0,W0	R/W			R0,W0		R/W	
Protection Attribute	-	-			-		-	
Initial value	0	011			00		11	

[bit31:29] Reserved: Reserved bits
When writing, always write "0". When reading, "0" is always read.

[bit28:24] TDCO[4:0]: Transceiver Delay Compensation Offset

TDCO[4:0]	Description
0x00-0x1F	Offset value defining the distance between the measured delay (the delay from canfd_tx to canfd_rx) and the secondary sample point. Valid values are 0 to 31 canfd_cclk periods.

[bit23] TDC: Transceiver Delay Compensation

Bit	Description
0	Transceiver Delay Compensation disabled.
1	Transceiver Delay Compensation enabled.

[bit22:21] Reserved: Reserved bits
When writing, always write "0". When reading, "0" is always read.

[bit20:16] FBRP[4:0]: Fast Baud Rate Prescaler

FBRP[4:0]	Description
0x00-0x1F	The value by which the oscillator frequency is divided for generating the fast bit time quanta. The fast bit time is built up from a multiple of this quanta. Valid values for the Fast Baud Rate Prescaler are 0 to 31. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used.

[bit15:12] Reserved: Reserved bits
When writing, always write "0". When reading, "0" is always read.

[bit11:8] FTSEG1[3:0]: Fast time segment before sample point

FTSEG1[3:0]	Description
0x1-0xF	Valid values are 1 to 15. The value 0 must not be used. The actual interpretation by the hardware of this value is such that one more than the programmed value is used.

[bit7] Reserved: Reserved bit
When writing, always write "0". When reading, "0" is always read.

[bit6:4] FTSEG2[2:0]: Fast time segment after sample point

FTSEG2[2:0]	Description
0x0-0x7	Valid values are 0 to 7. The actual interpretation by the hardware of this value is such that one more than the programmed value is used.

[bit3:2] Reserved: Reserved bits

When writing, always write "0". When reading, "0" is always read.

[bit1:0] FSJW[1:0]: Fast (Re) Synchronization Jump Width

FSJW[1:0]	Description
0x0-0x3	Valid values are 0 to 3. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used.

5.4. Test Register (MCG_CANFDx_TEST, CPG_CANFDx_TEST)

The Test Register monitors the canfd_rx/canfd_tx pins and displays the Transceiver Delay Compensation value. It is also used to enable the Loop Back Modes.

Write access to the Test Register has to be enabled by setting the Test Mode Enable bit CCCR.TEST to "1".

All Test Register functions are set to their reset values when bit CCCR.TEST is reset.

Loop Back Mode and software control of pin canfd_tx are hardware test modes. Programming of TX ≠ 0b00 may disturb the message transfer on the CAN bus.

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

bit	23	22	21	20	19	18	17	16
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

bit	15	14	13	12	11	10	9	8
Field	Reserved		TDCV[5:0]					
R/W Attribute	R0,W0		R,WX					
Protection Attribute	-		-					
Initial value	00		000000					

bit	7	6	5	4	3	2	1	0
Field	RX	TX[1:0]		LBCK	Reserved			
R/W Attribute	R,WX	R/W		R/W	R0,W0			
Protection Attribute	-	-		-	-			
Initial value	U	00		0	0000			

U = undefined. The RX bit will reflect the actual level at pin canfd_rx.

[bit31:14] Reserved: Reserved bits

When writing, always write "0". When reading, "0" is always read.

[bit13:8] TDCV[5:0]: Transceiver Delay Compensation Value

TDCV[5:0]	Description
0x00-0x3F	Position of the secondary sample point, defined by the sum of the measured delay from canfd_tx to canfd_rx and the Transceiver Delay Compensation Offset (FBTP.TDCO[4:0]). Valid values are 0 to 63 canfd_cclk periods.

[bit7] RX: Receive Pin

Monitors the actual value of pin canfd_rx

Bit	Description
0	The CAN bus is dominant (dominant level at pin canfd_rx).
1	The CAN bus is recessive (recessive level at pin canfd_rx).

[bit6:5] TX[1:0]: Control of Transmit Pin

Bits	Description
00	Reset value. canfd_tx controlled by the CAN Core, updated at the end of the bit time.
01	Sample Point can be monitored at pin canfd_tx.
10	Dominant level at pin canfd_tx.
11	Recessive level at pin canfd_tx.

[bit4] LBCK: Loop Back Mode

Bit	Description
0	Reset value, Loop Back Mode is disabled.
1	Loop Back Mode is enabled.

[bit3:0] Reserved: Reserved bits

When writing, always write "0". When reading, "0" is always read.

5.5. RAM Watchdog (MCG_CANFDx_RWD, CPG_CANFDx_RWD)

The RAM Watchdog monitors the Message RAM to see if it is ready to be accessed or not.

A Message RAM access starts the RAM Watchdog Counter with the value configured by the Watchdog Configuration (RWD.WDC[7:0]). The counter is reloaded with RWD.WDC when a successful access to the Message RAM has been completed. In case there is no response from the Message RAM until the counter has counted down to zero, the counter stops and interrupt flag Watchdog Interrupt (IR.WDI) is set.

The RAM Watchdog Counter is clocked by the Bus clock (canfd_bclk).

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

bit	23	22	21	20	19	18	17	16
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

bit	15	14	13	12	11	10	9	8
Field	WDV[7:0]							
R/W Attribute	R,WX							
Protection Attribute	-							
Initial value	0x00							

bit	7	6	5	4	3	2	1	0
Field	WDC[7:0]							
R/W Attribute	R/W							
Protection Attribute	-							
Initial value	0x00							

[bit31:16] Reserved: Reserved bits

When writing, always write "0". When reading, "0" is always read.

[bit15:8] WDV[7:0]: Watchdog Value

WDV[7:0]	Description
0x00-0xFF	Actual Message RAM Watchdog Counter Value

[bit7:0] WDC[7:0]: Watchdog Configuration

Start value of the Message RAM Watchdog Counter. With the reset value of 0x00 the counter is disabled.

Write access to this field is possible only when the Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) bits are both "1".

5.6. CC Control Register (MCG_CANFDx_CCCR, CPG_CANFDx_CCCR)

The CC Control Register configures various operating modes of the CAN FD Controller.

For details about setting and resetting of single bits see "3.1.1. Software Initialization".

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

bit	23	22	21	20	19	18	17	16
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

bit	15	14	13	12	11	10	9	8
Field	Reserved	TXP	FDBS	FDO	CMR[1:0]		CME[1:0]	
R/W Attribute	R0,W0	R/W	R,WX	R,WX	R/W		R/W	
Protection Attribute	-	-	-	-	-		-	
Initial value	0	0	0	0	00		00	

bit	7	6	5	4	3	2	1	0
Field	TEST	DAR	MON	CSR	CSA	ASM	CCE	INIT
R/W Attribute	R/W	R/W	R/W	R/W	R,WX	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	1

[bit31:15] Reserved: Reserved bits

When writing, always write "0". When reading, "0" is always read.

[bit14] TXP: Transmit Pause

If this bit is set, the CAN FD Controller pauses for two nominal bit times before starting the next transmission after it has successfully transmitted a frame.

Write access to this bit is possible only when the Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) bits are both "1".

Bit	Description
0	Transmit pause disabled.
1	Transmit pause enabled.

[bit13] FDBS: CAN FD Bit Rate Switching

This bit is configured via CCCR.CMR[1:0]. FDBS is reset to "0" when Initialization (CCCR.INIT) is set to "1".

Bit	Description
0	This node transmits no frames with bit rate switching.
1	This node transmits all frames (excl. remote frames) with bit rate switching.

[bit12] FDO: CAN FD Operation

This bit is configured via CCCR.CMR[1:0]. FDO is reset to "0" when Initialization (CCCR.INIT) is set to "1".

Bit	Description
0	This node transmits all frames in CAN format according to ISO11898-1.
1	This node transmits all frames (excl. remote frames) in CAN FD format.

[bit11:10] CMR[1:0]: CAN Mode Request

A change of the CAN operation mode is requested by writing to this bit field. After change to the requested operation mode the bit field is reset to 0b00 and the status flags CAN FD Bit Rate Switching (CCCR.FDBS) and CAN FD Operation (CCCR.FDO) are set accordingly.

In case the requested CAN operation mode is not enabled, the value written to CMR[1:0] is retained until it is overwritten by the next mode change request.

In case CAN Mode Enable CCCR.CME[1:0] = 0b01/0b10/0b11 a change to CAN operation according to ISO 11898-1 is always possible.

Default is CAN operation according to ISO11898-1 (CCCR.FDBS = 0 and CCCR.FDO = 0).

Bits	Description
00	Unchanged.
01	Request CAN FD operation.
10	Request CAN FD operation with bit rate switching.
11	Request CAN operation according ISO11898-1.

[bit9:8] CME[1:0]: CAN Mode Enable

Write access to this bit is possible only when the Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) bits are both "1".

Bits	Description
00	CAN operation according to ISO11898-1 enabled.
01	CAN FD operation enabled.
10	CAN FD operation with bit rate switching enabled.
11	CAN FD operation with bit rate switching enabled.

Note:

- *When CME[1:0] = 0b00, received frames are strictly interpreted according to ISO11898-1, which leads to the transmission of an error frame when receiving a CAN FD frame.*
  In case CME[1:0] = 0b01, transmission of long CAN FD frames and reception of long and fast CAN FD frames is enabled.

With CME[1:0] = 0b10/0b11, transmission and reception of long and fast CAN FD frames is enabled.

[bit7] TEST: Test Mode Enable

Bit TEST can only be set by the CPU when the Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) bits are both "1".

The bit can be reset by the CPU at any time.

Bit	Description
0	Normal operation, register TEST holds reset values.
1	Test Mode, write access to register TEST enabled.

[bit6] DAR: Disable Automatic Retransmission

Write access to this bit is possible only when the Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) bits are both "1". For a description of DAR mode, see "3.1.7. Disabled Automatic Retransmission".

Bit	Description
0	Automatic retransmission of messages not transmitted successfully enabled.
1	Automatic retransmission disabled.

Note:

- *When CCCR.DAR is set "1", always set "00" to first two identifier (Arbitration field) bits on transmission frame configuration.*

[bit5] MON: Bus Monitoring Mode

Bit MON can only be set by the CPU when the Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) bits are both "1".

The bit can be reset by the CPU at any time.

Bit	Description
0	Bus Monitoring Mode is disabled.
1	Bus Monitoring Mode is enabled.

[bit4] CSR: Clock Stop Request

For a description of Power Down, see "3.1.8. Power Down (Sleep Mode)".

Bit	Description
0	No clock stop is requested.
1	Clock stop requested. When clock stop is requested, first CCCR.INIT and then CCCR.CSA will be set after all pending transfer requests have been completed and the CAN bus reached idle.

[bit3] CSA: Clock Stop Acknowledge

For a description of Power Down, see "3.1.8.Power Down (Sleep Mode)".

Bit	Description
0	No clock stop acknowledged.
1	CAN FD Controller may be set in power down by stopping canfd_bclk and canfd_cclk.

[bit2] ASM: Restricted Operation Mode

Bit ASM can only be set by the CPU when the Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) bits are both "1"

The bit can be reset by the CPU at any time, but must follow the procedure as described in "4.6.2. Message RAM Access Failure Handling Operation".

For a description of the Restricted Operation Mode see "3.1.5. Restricted Operation Mode".

Bit	Description
0	Normal CAN operation.
1	Restricted Operation Mode active.

[bit1] CCE: Configuration Change Enable

Write access to this bit is possible only when the Initialization (CCCR.INIT) bit is "1". This bit (CCCR.CCE) is reset to "0" when CCCR.INIT is set to "0".

Bit	Description
0	The CPU has no write access to the protected configuration registers.
1	The CPU has write access to the protected configuration registers (while bit Initialization CCCR.INIT = 1).

Note:

- *The CPU must follow the procedure as described in "3.1.1. Software Initialization" to set CCCR.CCE, but it isn't necessary to follow the procedure, in case of setting CCCR.CCE just after hardware reset.*

[bit0] INIT: Initialization

Bit	Description
0	Normal Operation.
1	Initialization is started.

Notes:

- *Due to the synchronization mechanism between the two clock domains, there may be a delay until the value written to Initialization CCCR.INIT can be read back (the maximum length of the delay is four Bus clocks plus five CAN clocks). Therefore the programmer has to assure that the previous value written to CCCR.INIT has been accepted by reading CCCR.INIT before setting CCCR.INIT to a new value.*
- *In case of setting CCCR.INIT to "1" during Normal Operation, set CCCR.INIT to "1" by issuing a Clock Stop Request via CCCR.CSR = 1. Before resetting CCCR.INIT first reset CCCR.CSR.*

5.7. Bit Timing & Prescaler Register (MCG_CANFDx_BTP, CPG_CANFDx_BTP)

The Bit Timing & Prescaler Register configures the nominal bit time of the CAN FD Controller.

This register is only writable if bits Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) are set.

See "3.7.Configuring the CAN Bit Timing" for a description of the Bit Timing.

Note:

- *With a CAN clock (canfd_cclk) of 8 MHz, the reset value of 0x00000A33 configures the CAN FD Controller for a bit rate of 500 kBit/s.*

Note:

- *In environments that include CAN FD enabled nodes, do not use bit timing configurations where Time Segment after sample point and the Baud Rate Prescaler are both zero (i.e. BTP.TSEG2[3:0] = 0x0 and BTP.BRP[9:0] = 0x000). Such settings will cause the r1 bit (for CAN format frames) or EDL bit (for CAN FD format frames) to be transmitted opposite the protocol defined level.*
  In a CAN only environment, a bit timing configuration where Time Segment after sample point and the Baud Rate Prescaler are both zero (i.e. BTP.TSEG2[3:0] = 0x0 and BTP.BRP[9:0] = 0x000) may be used although this configuration will cause the r1 bit to be transmitted opposite the protocol defined level.

Details
The r1 bit (for CAN format frames) or EDL bit (for CAN FD format frames) will be transmitted opposite the protocol defined level for the following cases (all conditions within each case must be fulfilled).

Case 1:

- *The CAN FD Controller is configured to CAN operation mode according to ISO 11898-1 (CAN FD Bit Rate Switching (CCCR.FDBS) and CAN FD Operation (CCCR.FDO) are both zero)*
- *Time Segment after sample point and the Baud Rate Prescaler are both zero (i.e. BTP.TSEG2[3:0] = 0x0 and BTP.BRP[9:0] = 0x000)*
- *An extended frame format message with the MSB of the Identifier (ID28) set to "1" is transmitted (XTD bit = 1 and ID[28] = 1 in the corresponding Tx Buffer Element)*

Case 2:

- *The CAN FD Controller is configured to CAN FD operation mode (CAN FD Operation (CCCR.FDO) is "1")*
- *Time Segment after sample point and the Baud Rate Prescaler are both zero (i.e. BTP.TSEG2[3:0] = 0x0 and BTP.BRP[9:0] = 0x000)*
- *A Standard Frame format message is transmitted (XTD bit = 0 in the corresponding Tx Buffer Element)*

Case 3:

- *The CAN FD Controller is configured to CAN FD operation mode (CAN FD Operation (CCCR.FDO) is "1")*

- *Time Segment after sample point and the Baud Rate Prescaler are both zero (i.e. BTP.TSEG2[3:0] = 0x0 and BTP.BRP[9:0] = 0x000)*
- *An extended frame format message with the MSB of the Identifier (ID28) set to "0" is transmitted (XTD bit = 1 and ID[28] = 0 in the corresponding Tx Buffer Element)*

bit	31	30	29	28	27	26	25	24
Field	Reserved						BRP[9:8]	
R/W Attribute	R0,W0						R/W	
Protection Attribute	-						-	
Initial value	000000						00	

bit	23	22	21	20	19	18	17	16
Field	BRP[7:0]							
R/W Attribute	R/W							
Protection Attribute	-							
Initial value	0x00							

Bit	15	14	13	12	11	10	9	8
Field	Reserved		TSEG1[5:0]					
R/W Attribute	R0,W0		R/W					
Protection Attribute	-		-					
Initial value	00		001010					

Bit	7	6	5	4	3	2	1	0
Field	TSEG2[3:0]				SJW[3:0]			
R/W Attribute	R/W				R/W			
Protection Attribute	-				-			
Initial value	0x3				0x3			

[bit31:26] Reserved: Reserved bits

When writing, always write "0". When reading, "0" is always read.

[bit25:16] BRP[9:0]: Baud Rate Prescaler

BRP[9:0]	Description
0x000-0x3FF	The value by which the oscillator frequency is divided for generating the bit time quanta. The bit time is built up from a multiple of this quanta. Valid values for the Baud Rate Prescaler are 0 to 1023. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used.

[bit15:14] Reserved: Reserved bits

When writing, always write "0". When reading, "0" is always read.

[bit13:8] TSEG1[5:0]: Time segment before sample point

TSEG1[5:0]	Description
0x01-0x3F	Valid values are 1 to 63. The value 0 must not be used. The actual interpretation by the hardware of this value is such that one more than the programmed value is used.

[bit7:4] TSEG2[3:0]: Time segment after sample point

TSEG2[3:0]	Description
0x0-0xF	Valid values are 0 to 15. The actual interpretation by the hardware of this value is such that one more than the programmed value is used.

[bit3:0] SJW[3:0]: (Re) Synchronization Jump Width

SJW[3:0]	Description
0x0-0xF	Valid values are 0 to 15. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used.

5.8. Timestamp Counter Configuration (MCG_CANFDx_TSCC, CPG_CANFDx_TSCC)

The Timestamp Counter Configuration holds the settings for Timestamp Generation.

Write access to the Timestamp Counter Configuration (TSCC) is possible only when the Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) bits are both "1".

For a description of the Timestamp Counter see "3.2. Timestamp Generation".

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

bit	23	22	21	20	19	18	17	16
Field	Reserved				TCP[3:0]			
R/W Attribute	R0,W0				R/W			
Protection Attribute	-				-			
Initial value	0x0				0x0			

bit	15	14	13	12	11	10	9	8
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

Bit	7	6	5	4	3	2	1	0
Field	Reserved						TSS[1:0]	
R/W Attribute	R0,W0						R/W	
Protection Attribute	-						-	
Initial value	000000						00	

[bit31:20] Reserved: Reserved bits

When writing, always write "0". When reading, "0" is always read.

[bit19:16] TCP[3:0]: Timestamp Counter Prescaler

TCP[3:0]	Description
0x0-0xF	Configures the timestamp and timeout counters time unit in multiples of bit times [1 to 16]. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used.

Note:

- *For each timestamp and timeout counter time unit tick, the CAN FD Controller counts the number of bit times that have elapsed with an internal counter, up to the configured TSCC.TCP[3:0] value. This internal counter is not initialized by Initialization (CCCR.INIT), but only by a hardware reset.*

[bit15:2] Reserved: Reserved bits

When writing, always write "0". When reading, "0" is always read.

[bit1:0] TSS[1:0]: Timestamp Select

Bits	Description
00	Timestamp counter (TSCV.TSC[15:0]) value always 0x0000.
01	Timestamp counter value incremented according to Timestamp Counter Prescaler (TSCC.TCP[3:0]).
10	Timestamp counter value of a counter external to the CAN FD Controller used.
11	Same as 0b00.

Note:

- *When Timestamp Select TSCC.TSS[1:0] = 0b01, the clock signal for the Timeout Counter is derived from the CAN Core's sample point signal. Therefore, if the baud rate switch feature in CAN FD is used, the timeout counter is clocked differently in arbitration and data field.*

5.9. Timestamp Counter Value (MCG_CANFDx_TSCV, CPG_CANFDx_TSCV)

Holds the value of the Timestamp Counter.

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

bit	23	22	21	20	19	18	17	16
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

bit	15	14	13	12	11	10	9	8
Field	TSC[15:8]							
R/W Attribute	R,W							
Protection Attribute	-							
Initial value	0x00							

bit	7	6	5	4	3	2	1	0
Field	TSC[7:0]							
R/W Attribute	R,W							
Protection Attribute	-							
Initial value	0x00							

[bit31:16] Reserved: Reserved bits

When writing, always write "0". When reading, "0" is always read.

[bit15:0] TSC[15:0]: Timestamp Counter

The internal/external Timestamp Counter value is captured on start of frame (both Rx and Tx).

- When Timestamp Select TSCC.TSS[1:0] = 0b01, the Timestamp Counter TSCV.TSC[15:0] is incremented in multiples of bit times [1 to 16] depending on the configuration of the Timestamp Counter Prescaler (TSCC.TCP[3:0]). A wrap around sets interrupt flag Timestamp Wraparound (IR.TSW). Write access to TSCV resets the counter to zero.
- When Timestamp Select TSCC.TSS[1:0] = 0b10, TSCV.TSC[15:0] reflects a 16-bit Timestamp Counter value obtained from a counter external to the CAN FD Controller. For this case, a write access has no impact and will not clear the counter value.

Note:

- *A “wrap around” is a change of the Timestamp Counter value from non-zero to zero not caused by write access to Timestamp Counter Value (TSCV). Such write access will not set IR.TSW to "1".*

5.10. Timeout Counter Configuration (MCG_CANFDx_TOCC, CPG_CANFDx_TOCC)

The Timeout Counter Configuration holds the settings for the Timeout Counter.

Write access to the Timeout Counter Configuration (TOCC) is possible only when the Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) bits are both "1".

For a description of the Timeout Counter see "3.3. Timeout Counter".

bit	31	30	29	28	27	26	25	24
Field	TOP[15:8]							
R/W Attribute	R/W							
Protection Attribute	-							
Initial value	0xFF							

bit	23	22	21	20	19	18	17	16
Field	TOP[7:0]							
R/W Attribute	R/W							
Protection Attribute	-							
Initial value	0xFF							

bit	15	14	13	12	11	10	9	8
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

Bit	7	6	5	4	3	2	1	0
Field	Reserved					TOS[1:0]		ETOC
R/W Attribute	R0,W0					R/W		R/W
Protection Attribute	-					-		-
Initial value	00000					00		0

[bit31:16] TOP[15:0]: Timeout Period

Start value of the Timeout Counter (down-counter). Configures the Timeout Period.

[bit15:3] Reserved: Reserved bits

When writing, always write "0". When reading, "0" is always read.

[bit2:1] TOS[1:0]: Timeout Select

When operating in Continuous mode (TOS = 0b00), a write to Timeout Counter Value (TOCV) presets the counter to the value configured by Timeout Period (TOCC.TOP[15:0]) and continues down-counting.

When the Timeout Counter is controlled by one of the FIFOs (TOS = 0b01/0b10/0b11), an empty FIFO presets the counter to the value configured by Timeout Period (TOCC.TOP[15:0]). Down-counting is started when the first FIFO element is stored.

Bits	Description
00	Continuous operation.
01	Timeout controlled by Tx Event FIFO.
10	Timeout controlled by Rx FIFO 0.
11	Timeout controlled by Rx FIFO 1.

[bit0] ETOC: Enable Timeout Counter

Bit	Description
0	Timeout Counter disabled.
1	Timeout Counter enabled.

5.11. Timeout Counter Value (MCG_CANFDx_TOCV, CPG_CANFDx_TOCV)

Holds the value of the Timeout Counter.

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

bit	23	22	21	20	19	18	17	16
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

Bit	15	14	13	12	11	10	9	8
Field	TOC[15:8]							
R/W Attribute	R,W							
Protection Attribute	-							
Initial value	0xFF							

bit	7	6	5	4	3	2	1	0
Field	TOC[7:0]							
R/W Attribute	R,W							
Protection Attribute	-							
Initial value	0xFF							

[bit31:16] Reserved: Reserved bits

When writing, always write "0". When reading, "0" is always read.

[bit15:0] TOC[15:0]: Timeout Counter

The Timeout Counter is decremented in multiples of bit times [1 to 16] depending on the configuration of Timestamp Counter Prescaler (TSCC.TCP[3:0]). Setting Configuration Change Enable (CCCR.CCE) will preset the Timeout Counter (TOCV.TOC[15:0]) to the Timeout Period value (TOCC.TOP[15:0]).

- When Timeout Select TOCC.TOS[1:0] = 0b00 (Continuous Mode), the counter starts when Initialization CCCR.INIT is reset. A write to the Timeout Counter Value register TOCV presets the counter to the value configured by Timeout Period (TOCC.TOP[15:0]) and continues down-counting. When the counter reaches zero, interrupt flag Timeout Occurred (IR.TOO) is set, and the counter is immediately restarted at Timeout Period (TOCC.TOP[15:0]).

- When Timeout Select TOCC.TOS ≠ 0b00 (counter is controlled by one of the FIFOs), an empty FIFO presets the counter to the value configured by Timeout Period (TOCC.TOP[15:0]). Down-counting is started when the first FIFO element is stored. In this setting, writing to Timeout Counter Value (TOCV) has no effect. When the counter reaches zero, interrupt flag Timeout Occurred (IR.TOO) is set, and the Timeout Counter is stopped.

5.12. Error Counter Register (MCG_CANFDx_ECR, CPG_CANFDx_ECR)

Holds the values of the Error Counters.

Note:

- *When Restricted Operation Mode (CCCR.ASM) is set, the CAN protocol controller does not increment the Transmit Error Counter (ECR.TEC[7:0]) and Receive Error Counter (ECR.REC[6:0]) when a CAN protocol error is detected, but CAN Error Logging (ECR.CEL[7:0]) is still incremented.*

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	RX,W0							
Protection Attribute	-							
Initial value	0xXX							

bit	23	22	21	20	19	18	17	16
Field	CEL[7:0]							
R/W Attribute	R,WX							
Protection Attribute	-							
Initial value	0x00							

bit	15	14	13	12	11	10	9	8
Field	RP	REC[6:0]						
R/W Attribute	R,WX	R,WX						
Protection Attribute	-	-						
Initial value	0	0000000						

bit	7	6	5	4	3	2	1	0
Field	TEC[7:0]							
R/W Attribute	R,WX							
Protection Attribute	-							
Initial value	0x00							

[bit31:24] Reserved: Reserved bits

When writing, always write "0". The read value is undefined.

[bit23:16] CEL[7:0]: CAN Error Logging

The counter is incremented each time when a CAN protocol error causes the Transmit Error Counter or the Receive Error Counter to be incremented. It is reset by read access to CEL.

The counter stops at 0xFF; the next increment of Transmit Error Counter (ECR.TEC[7:0]) or Receive Error Counter (ECR.REC[6:0]) sets interrupt flag Error Logging Overflow (IR.ELO).

[bit15] RP: Receive Error Passive

Bit	Description
0	The Receive Error Counter is below the error passive level of 128.
1	The Receive Error Counter has reached the error passive level of 128.

[bit14:8] REC[6:0]: Receive Error Counter

REC[6:0]	Description
0-127	Actual state of the Receive Error Counter, values between 0 and 127.

[bit7:0] TEC[7:0]: Transmit Error Counter

TEC[7:0]	Description
0-255	Actual state of the Transmit Error Counter, values between 0 and 255.

5.13. Protocol Status Register (MCG_CANFDx_PSR, CPG_CANFDx_PSR)

The Protocol Status Register displays the CAN protocol status of the CAN FD Controller.

Note:

- *The PSR register accepts only 16 bit half-word and 32 bit word accesses. 8 bit byte accesses to this register are prohibited.*

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	RX,W0							
Protection Attribute	-							
Initial value	0xXX							

Bit	23	22	21	20	19	18	17	16
Field	Reserved							
R/W Attribute	RX,W0							
Protection Attribute	-							
Initial value	0xXX							

bit	15	14	13	12	11	10	9	8
Field	Reserved		REDL	RBRS	RESI	FLEC[2:0]		
R/W Attribute	RX,W0		R,WX	R,WX	R,WX	R,WX		
Protection Attribute	-		-	-	-	-		
Initial value	XX		0	0	0	111		

Bit	7	6	5	4	3	2	1	0
Field	BO	EW	EP	ACT[1:0]		LEC[2:0]		
R/W Attribute	R,WX	R,WX	R,WX	R,WX		R,WX		
Protection Attribute	-	-	-	-		-		
Initial value	0	0	0	00		111		

[bit31:14] Reserved: Reserved bits

When writing, always write "0". The read value is undefined.

[bit13] REDL: Received a CAN FD Message

This bit is set independent of acceptance filtering. A read access will reset this bit to "0".

Bit	Description
0	Since this bit was reset by the CPU, no CAN FD message has been received.
1	Message in CAN FD format with EDL flag set has been received.

Note:

- *Once set to "1", REDL will be held at "1" until it is reset to "0" by the CPU.*

[bit12] RBRS: BRS flag of last received CAN FD Message

This bit is set together with "Received a CAN FD Message" (PSR.REDL), independent of acceptance filtering. A read access will reset this bit to "0".

Bit	Description
0	Last received CAN FD message did not have its BRS flag set.
1	Last received CAN FD message had its BRS flag set.

[bit11] RESI: ESI flag of last received CAN FD Message

This bit is set together with "Received a CAN FD Message" (PSR.REDL), independent of acceptance filtering. A read access will reset this bit to "0".

Bit	Description
0	Last received CAN FD message did not have its ESI flag set.
1	Last received CAN FD message had its ESI flag set.

[bit10:8] FLEC[2:0]: Fast Last Error Code

Type of last error that occurred in the data phase of a CAN FD format frame with its BRS flag set. Coding is the same as for Last Error Code (PSR.LEC[2:0]).

This field will be cleared to zero when a CAN FD format frame with its BRS flag set has been transferred (reception or transmission) without error.

A read access will set this field to 7.

[bit7] BO: Bus_Off Status

Bit	Description
0	The CAN FD Controller is not Bus_Off.
1	The CAN FD Controller is in Bus_Off state.

[bit6] EW: Warning Status

Bit	Description
0	Both error counters are below the Error_Warning limit of 96.
1	At least one of error counter has reached the Error_Warning limit of 96.

[bit5] EP: Error Passive

Bit	Description
0	The CAN FD Controller is in the Error_Active state. It normally takes part in bus communication and sends an active error flag when an error has been detected.
1	The CAN FD Controller is in the Error_Passive state.

[bit4:3] ACT[1:0]: Activity

Monitors the CAN communication state. This field is reset to 0b00 when Initialize (CCCR.INIT) is set to "1".

Bits	Description
00	Synchronizing - node is synchronizing on CAN communication.
01	Idle - node is neither receiver nor transmitter.
10	Receiver - node is operating as receiver.
11	Transmitter - node is operating as transmitter.

[bit2:0] LEC[2:0]: Last Error Code

The LEC indicates the type of the last error to occur on the CAN bus.

This field will be cleared to 0 when a message has been transferred (reception or transmission) without error.

LEC[2:0]	Description
0	No Error: No error occurred since LEC has been reset by successful reception or transmission.
1	Stuff Error: More than 5 equal bits in a sequence have occurred in a part of a received message where this is not allowed.
2	Form Error: A fixed format part of a received frame has the wrong format.
3	AckError: The message transmitted by the CAN FD Controller was not acknowledged by another node.
4	Bit1Error: During the transmission of a message (with the exception of the arbitration field), the CAN FD Controller wanted to send a recessive level, but the monitored bus value was dominant.
5	Bit0Error: During the transmission of a message (or acknowledge bit, or active error flag, or overload flag), the CAN FD Controller wanted to send a dominant level, but the monitored bus value was recessive. During Bus_Off recovery this status is set each time a sequence of 11 recessive bits has been monitored. This enables the CPU to monitor the proceeding of the Bus_Off recovery sequence (indicating the bus is not stuck at dominant or continuously disturbed).
6	CRCError: The CRC check sum of a received message was incorrect. The CRC of an incoming message does not match with the CRC calculated from the received data.
7	NoChange: Any read access to the Protocol Status Register (PSR) re-initializes the Last Error Code field (LEC[2:0]) to 7. When the LEC[2:0] shows the value 7, no CAN bus event was detected since the last CPU read access to the Protocol Status Register.

Notes:

- *When a frame in CAN FD format has reached the data phase with BRS flag set, the next CAN event (error or valid frame) will be shown in the Fast Last Error Code field (PSR.FLEC[2:0]) instead of the Last Error Code field (PSR.LEC[2:0]). An error in a fixed stuff bit of a CAN FD CRC sequence will be shown as a Form Error, not Stuff Error.*
- *The Bus_Off recovery sequence (see CAN Specification Rev. 2.0 or ISO11898-1) cannot be shortened by setting or resetting Initialization (CCCR.INIT). If the CAN FD Controller goes Bus_Off, it will set CCCR.INIT of its own accord, stopping all bus activities. Once CCCR.INIT has been cleared by the CPU, the CAN FD Controller will then wait for 129 occurrences of Bus*

Idle (129 × 11 consecutive recessive bits) before resuming normal operation. At the end of the Bus_Off recovery sequence, the Receive and Transmit Error Counters (ECR.REC[6:0] and ECR.TEC[7:0]) will be reset. During the waiting time after the resetting of CCCR.INIT, each time a sequence of 11 recessive bits has been monitored, a Bit0Error code is written to the Last Error Code field (PSR.LEC[2:0]), enabling the CPU to readily check up whether the CAN bus is stuck at dominant or continuously disturbed, and to monitor the Bus_Off recovery sequence. The Receive Error Counter (ECR.REC[6:0]) is used to count these sequences

5.14. Interrupt Register (MCG_CANFDx_IR, CPG_CANFDx_IR)

The Interrupt Register holds the flags that are set when one of the listed conditions is detected (edge-sensitive). The flags remain set until the CPU clears them. A flag is cleared by writing a "1" to the corresponding bit position. Writing a "0" has no effect. A hard reset will clear the register.

The configuration of Interrupt Enable (IE) controls whether an interrupt is generated.

The configuration of Interrupt Line Select (ILS) controls on which interrupt line (canfd_int0/1) an interrupt is signaled.

Note:

- *The TEFW, RF1W, and RF0W interrupts are asserted only when their respective FIFO levels are equal to their configured watermarks. If at this point the interrupt is cleared, and the respective FIFO continues to be filled, the interrupt will not be re-asserted even if the respective FIFO level is greater than the configured watermark.*

bit	31	30	29	28	27	26	25	24
Field	STE	FOE	ACKE	BE	CRCE	WDI	BO	EW
R/W Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	EP	ELO	BEU	BEC	DRX	TOO	MRAF	TSW
R/W Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	TEFL	TEFF	TEFW	TEFN	TFE	TCF	TC	HPM
R/W Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	RF1L	RF1F	RF1W	RF1N	RF0L	RF0F	RF0W	RF0N
R/W Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

[bit31] STE: Stuff Error

Bit	Description
0	No Stuff Error detected.
1	More than 5 equal bits in a sequence occurred.

[bit30] FOE: Format Error

Bit	Description
0	No Format Error detected.
1	A fixed format part of a received frame has the wrong format.

[bit29] ACKE: Acknowledge Error

Bit	Description
0	No Acknowledge Error detected.
1	A transmitted message was not acknowledged by another node.

[bit28] BE: Bit Error

Bit	Description
0	No Bit0Error/Bit1Error detected.
1	CAN FD Controller wanted to send a recessive/dominant level, but monitored bus level was dominant/recessive.

[bit27] CRCE: CRC Error

Bit	Description
0	No CRC Error detected.
1	Received CRC did not match the calculated CRC.

[bit26] WDI: Watchdog Interrupt

Bit	Description
0	No Message RAM Watchdog event occurred.
1	Message RAM Watchdog event due to incomplete access of Message RAM.

[bit25] BO: Bus_Off Status

Bit	Description
0	Bus_Off status unchanged.
1	Bus_Off status changed.

[bit24] EW: Warning Status

Bit	Description
0	Error_Warning status unchanged.
1	Error_Warning status changed.

[bit23] EP: Error Passive

Bit	Description
0	Error_Passive status unchanged.
1	Error_Passive status changed.

[bit22] ELO: Error Logging Overflow

Bit	Description
0	CAN Error Logging Counter (ECR.CEL[7:0]) did not overflow.
1	Overflow of CAN Error Logging Counter (ECR.CEL[7:0]) occurred.

[bit21] BEU: Bit Error Uncorrected

Message RAM bit error during access from CAN FD Controller detected, but could not be corrected by the ECC (Error Correction Code) logic attached to the Message RAM.

An uncorrected Message RAM bit error sets Initialization (CCCR.INIT) to "1". This is done to avoid transmission of corrupted data.

Bit	Description
0	No bit error detected when reading from Message RAM.
1	Bit error detected, but could not be corrected.

Note:

- *The corresponding bit in the ECC logic must also be cleared when clearing IR.BEU.*

[bit20] BEC: Bit Error Corrected

Message RAM bit error during access from CAN FD Controller detected and corrected by the ECC logic attached to the Message RAM.

Bit	Description
0	No bit error detected when reading from Message RAM.
1	Bit error detected and corrected by ECC logic.

Note:

- *The corresponding bit in the ECC logic must also be cleared when clearing IR.BEC.*

[bit19] DRX: Message stored to Dedicated Rx Buffer

The flag is set whenever a received message has been stored into a dedicated Rx Buffer. This flag is not set by reception of a Debug message.

Bit	Description
0	No Rx Buffer updated.
1	At least one received message stored into an Rx Buffer.

[bit18] TOO: Timeout Occurred

Bit	Description
0	No timeout.
1	Timeout reached.

[bit17] MRAF: Message RAM Access Failure

The flag is set, when the Rx Handler

- has not completed acceptance filtering or storage of an accepted message until the arbitration field of the following message has been received. In this case acceptance filtering or message storage is aborted and the Rx Handler starts processing of the following message. Such behavior will occur also when a Protocol Exception Event is detected, since this will cause the Rx Handler to halt for that frame (see "3.4.5. Protocol Exception Event").
- completed message reception operation, but was not able to write the message to the Message RAM. In this case message storage is aborted.

In both cases the Rx FIFO put index is not updated resp. the New Data flag for a dedicated Rx Buffer is not set, a partly stored message is overwritten when the next message is stored to this location.

The flag is also set when the Tx Handler was not able to read a message from the Message RAM in time. In this case message transmission is aborted. In case of a Tx Handler access failure the CAN FD Controller is switched into Restricted Operation Mode (see "3.1.5. Restricted Operation Mode"). To leave Restricted Operation Mode, the CPU has to reset the Restricted Operation Mode bit (CCCR.ASM). See "4.6.2. Message RAM Access Failure Handling Operation" for a detailed procedure of how to leave the Restricted Operation Mode.

Bit	Description
0	No Message RAM access failure occurred.
1	Message RAM access failure occurred.

[bit16] TSW: Timestamp Wraparound

Bit	Description
0	No timestamp counter wrap-around.
1	Timestamp counter wrapped around.

[bit15] TEFL: Tx Event FIFO Element Lost

Bit	Description
0	No Tx Event FIFO element lost.
1	Tx Event FIFO element lost. Also set after write attempt to Tx Event FIFO of size zero.

[bit14] TEFF: Tx Event FIFO Full

Bit	Description
0	Tx Event FIFO not full.
1	Tx Event FIFO full.

[bit13] TEFW: Tx Event FIFO Watermark Reached

Bit	Description
0	Tx Event FIFO fill level below watermark.
1	Tx Event FIFO fill level reached watermark.

[bit12] TEFN: Tx Event FIFO New Entry

Bit	Description
0	Tx Event FIFO unchanged.
1	Tx Handler wrote Tx Event FIFO element.

[bit11] TFE: Tx FIFO Empty

Bit	Description
0	Tx FIFO non-empty.
1	Tx FIFO empty.

Note:

- *The TFE bit will not be asserted for an empty Tx Queue.*

[bit10] TCF: Transmission Cancellation Finished

Bit	Description
0	No transmission cancellation finished.
1	Transmission cancellation finished.

[bit9] TC: Transmission Completed

Bit	Description
0	No transmission completed.
1	Transmission completed.

[bit8] HPM: High Priority Message

Bit	Description
0	No high priority message received.
1	High priority message received.

[bit7] RF1L: Rx FIFO 1 Message Lost

Bit	Description
0	No Rx FIFO 1 message lost.
1	Rx FIFO 1 message lost. Also set after write attempt to Rx FIFO 1 of size zero.

[bit6] RF1F: Rx FIFO 1 Full

Bit	Description
0	Rx FIFO 1 not full.
1	Rx FIFO 1 full.

[bit5] RF1W: Rx FIFO 1 Watermark Reached

Bit	Description
0	Rx FIFO 1 fill level below watermark.
1	Rx FIFO 1 fill level reached watermark.

[bit4] RF1N: Rx FIFO 1 New Message

Bit	Description
0	No new message written to Rx FIFO 1.
1	New message written to Rx FIFO 1.

[bit3] RF0L: Rx FIFO 0 Message Lost

Bit	Description
0	No Rx FIFO 0 message lost.
1	Rx FIFO 0 message lost. Also set after write attempt to Rx FIFO 0 of size zero.

[bit2] RF0F: Rx FIFO 0 Full

Bit	Description
0	Rx FIFO 0 not full.
1	Rx FIFO 0 full.

[bit1] RF0W: Rx FIFO 0 Watermark Reached

Bit	Description
0	Rx FIFO 0 fill level below watermark.
1	Rx FIFO 0 fill level reached watermark.

[bit0] RF0N: Rx FIFO 0 New Message

Bit	Description
0	No new message written to Rx FIFO 0.
1	New message written to Rx FIFO 0.

5.15. Interrupt Enable (MCG_CANFDx_IE, CPG_CANFDx_IE)

The settings in the Interrupt Enable register determine which status changes in the Interrupt Register (IR) will be signaled on an interrupt line.

Bit	Description
0	Interrupt disabled.
1	Interrupt enabled.

bit	31	30	29	28	27	26	25	24
Field	STEE	FOEE	ACKEE	BEE	CRCEE	WDIE	BOE	EWE
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	EPE	ELOE	BEUE	BECE	DRXE	TOOE	MRAFE	TSWE
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	TEFLE	TEFFE	TEFWE	TEFNE	TFEE	TCFE	TCE	HPME
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	RF1LE	RF1FE	RF1WE	RF1NE	RF0LE	RF0FE	RF0WE	RF0NE
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

[bit31] STEE: Stuff Error Interrupt Enable

[bit30] FOEE: Format Error Interrupt Enable

[bit29] ACKEE: Acknowledge Error Interrupt Enable

[bit28] BEE: Bit Error Interrupt Enable

[bit27] CRCEE: CRC Error Interrupt Enable

[bit26] WDIE: Watchdog Interrupt Enable

[bit25] BOE: Bus_Off Status Interrupt Enable

[bit24] EWE: Warning Status Interrupt Enable

[bit23] EPE: Error Passive Interrupt Enable

[bit22] ELOE: Error Logging Overflow Interrupt Enable

[bit21] BEUE: Bit Error Uncorrected Interrupt Enable

[bit20] BECE: Bit Error Corrected Interrupt Enable

[bit19] DRXE: Message stored to Dedicated Rx Buffer Interrupt Enable

[bit18] TOOE: Timeout Occurred Interrupt Enable

[bit17] MRAFE: Message RAM Access Failure Interrupt Enable

[bit16] TSWE: Timestamp Wraparound Interrupt Enable

[bit15] TEFLE: Tx Event FIFO Event Lost Interrupt Enable

[bit14] TEFFE: Tx Event FIFO Full Interrupt Enable

[bit13] TEFWE: Tx Event FIFO Watermark Reached Interrupt Enable

[bit12] TEFNE: Tx Event FIFO New Entry Interrupt Enable

[bit11] TFEE: Tx FIFO Empty Interrupt Enable

[bit10] TCFE: Transmission Cancellation Finished Interrupt Enable

[bit9] TCE: Transmission Completed Interrupt Enable

[bit8] HPME: High Priority Message Interrupt Enable

[bit7] RF1LE: Rx FIFO 1 Message Lost Interrupt Enable

[bit6] RF1FE: Rx FIFO 1 Full Interrupt Enable

[bit5] RF1WE: Rx FIFO 1 Watermark Reached Interrupt Enable

[bit4] RF1NE: Rx FIFO 1 New Message Interrupt Enable

[bit3] RF0LE: Rx FIFO 0 Message Lost Interrupt Enable

[bit2] RF0FE: Rx FIFO 0 Full Interrupt Enable

[bit1] RF0WE: Rx FIFO 0 Watermark Reached Interrupt Enable

[bit0] RF0NE: Rx FIFO 0 New Message Interrupt Enable

5.16. Interrupt Line Select (MCG_CANFDx_ILS, CPG_CANFDx_ILS)

The Interrupt Line Select register assigns an interrupt generated by a specific interrupt flag from the Interrupt Register (IR) to one of the two CAN FD Controller interrupt lines (canfd_int0/1).

Bit	Description
0	Interrupt assigned to interrupt line canfd_int0.
1	Interrupt assigned to interrupt line canfd_int1.

bit	31	30	29	28	27	26	25	24
Field	STEL	FOEL	ACKEL	BEL	CRCEL	WDIL	BOL	EWL
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	EPL	ELOL	BEUL	BECL	DRXL	TOOL	MRAFL	TSWL
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	TEFLL	TEFFL	TEFWL	TEFNL	TFEL	TCFL	TCL	HPML
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	RF1LL	RF1FL	RF1WL	RF1NL	RF0LL	RF0FL	RF0WL	RF0NL
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

[bit31] STEL: Stuff Error Interrupt Line

[bit30] FOEL: Format Error Interrupt Line

[bit29] ACKEL: Acknowledge Error Interrupt Line

[bit28] BEL: Bit Error Interrupt Line

[bit27] CRCEL: CRC Error Interrupt Line

[bit26] WDIL: Watchdog Interrupt Line

[bit25] BOL: Bus_Off Status Interrupt Line

[bit24] EWL: Warning Status Interrupt Line

[bit23] EPL: Error Passive Interrupt Line

[bit22] ELOL: Error Logging Overflow Interrupt Line

[bit21] BEUL: Bit Error Uncorrected Interrupt Line

[bit20] BECL: Bit Error Corrected Interrupt Line

[bit19] DRXL: Message stored to Dedicated Rx Buffer Interrupt Line

[bit18] TOOL: Timeout Occurred Interrupt Line

[bit17] MRAFL: Message RAM Access Failure Interrupt Line

[bit16] TSWL: Timestamp Wraparound Interrupt Line

[bit15] TEFLL: Tx Event FIFO Event Lost Interrupt Line

[bit14] TEFFL: Tx Event FIFO Full Interrupt Line

[bit13] TEFWL: Tx Event FIFO Watermark Reached Interrupt Line

[bit12] TEFNL: Tx Event FIFO New Entry Interrupt Line

[bit11] TFEL: Tx FIFO Empty Interrupt Line

[bit10] TCFL: Transmission Cancellation Finished Interrupt Line

[bit9] TCL: Transmission Completed Interrupt Line

[bit8] HPML: High Priority Message Interrupt Line

[bit7] RF1LL: Rx FIFO 1 Message Lost Interrupt Line

[bit6] RF1FL: Rx FIFO 1 Full Interrupt Line

[bit5] RF1WL: Rx FIFO 1 Watermark Reached Interrupt Line

[bit4] RF1NL: Rx FIFO 1 New Message Interrupt Line

[bit3] RF0LL: Rx FIFO 0 Message Lost Interrupt Line

[bit2] RF0FL: Rx FIFO 0 Full Interrupt Line

[bit1] RF0WL: Rx FIFO 0 Watermark Reached Interrupt Line

[bit0] RF0NL: Rx FIFO 0 New Message Interrupt Line

5.17. Interrupt Line Enable (MCG_CANFDx_ILE, CPG_CANFDx_ILE)

Interrupt Line Enable can separately enable/disable each of the two interrupt lines to the CPU by the values set to Enable Interrupt Line 0 (ILE.EINT0) and Enable Interrupt Line 1 (ILE.EINT1).

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

Bit	23	22	21	20	19	18	17	16
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

Bit	15	14	13	12	11	10	9	8
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

Bit	7	6	5	4	3	2	1	0
Field	Reserved						EINT1	EINT0
R/W Attribute	R0,W0						R/W	R/W
Protection Attribute	-						-	-
Initial value	000000						0	0

[bit31:2] Reserved: Reserved bits

When writing, always write "0". When reading, "0" is always read.

[bit1] EINT1: Enable Interrupt Line 1

Bit	Description
0	Interrupt line canfd_int1 disabled.
1	Interrupt line canfd_int1 enabled.

[bit0] EINT0: Enable Interrupt Line 0

Bit	Description
0	Interrupt line canfd_int0 disabled.
1	Interrupt line canfd_int0 enabled.

5.18. Global Filter Configuration (MCG_CANFDx_GFC, CPG_CANFDx_GFC)

Global settings for Message ID filtering. The Global Filter Configuration (GFC) controls the filter path for standard and extended messages as described in Figure 3-4 and Figure 3-5.

Write access to the Global Filter Configuration (GFC) is possible only when the Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) bits are both "1".

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

Bit	23	22	21	20	19	18	17	16
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

Bit	15	14	13	12	11	10	9	8
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

Bit	7	6	5	4	3	2	1	0
Field	Reserved		ANFS[1:0]		ANFE[1:0]		RRFS	RRFE
R/W Attribute	R0,W0		R/W		R/W		R/W	R/W
Protection Attribute	-		-		-		-	-
Initial value	00		00		00		0	0

[bit31:6] Reserved: Reserved bits

When writing, always write "0". When reading, "0" is always read.

[bit5:4] ANFS[1:0]: Accept Non-matching Frames Standard

Defines how received messages with 11-bit IDs that do not match any element of the filter list are treated.

Bits	Description
00	Accept in Rx FIFO 0.
01	Accept in Rx FIFO 1.
10	Reject.

11	Reject.

[bit3:2] ANFE[1:0]: Accept Non-matching Frames Extended

Defines how received messages with 29-bit IDs that do not match any element of the filter list are treated.

Bits	Description
00	Accept in Rx FIFO 0.
01	Accept in Rx FIFO 1.
10	Reject.
11	Reject.

[bit1] RRFS: Reject Remote Frames Standard

Bit	Description
0	Filter remote frames with 11-bit standard IDs.
1	Reject all remote frames with 11-bit standard IDs.

[bit0] RRFE: Reject Remote Frames Extended

Bit	Description
0	Filter remote frames with 29-bit extended IDs.
1	Reject all remote frames with 29-bit extended IDs.

5.19. Standard ID Filter Configuration (MCG_CANFDx_SIDFC, CPG_CANFDx_SIDFC)

Settings for 11-bit standard Message ID filtering. The Standard ID Filter Configuration controls the filter path for standard messages as described in Figure 3-4.

Write access to the Standard ID Filter Configuration (SIDFC) is possible only when the Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) bits are both "1".

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

bit	23	22	21	20	19	18	17	16
Field	LSS[7:0]							
R/W Attribute	R/W							
Protection Attribute	-							
Initial value	0x00							

Bit	15	14	13	12	11	10	9	8
Field	FLSSA[15:8]							
R/W Attribute	R/W							
Protection Attribute	-							
Initial value	00000000							

bit	7	6	5	4	3	2	1	0
Field	FLSSA[7:2]						Reserved	
R/W Attribute	R/W						R0,W0	
Protection Attribute	-						-	
Initial value	000000						00	

[bit31:24] Reserved: Reserved bits

When writing, always write "0". When reading, "0" is always read.

[bit23:16] LSS[7:0]: List Size Standard

LSS[7:0]	Description
0	No standard Message ID filter.
1 to 128	Number of standard Message ID filter elements.
>128	Values greater than 128 are interpreted as 128.

[bit15:2] FLSSA[15:2]: Filter List Standard Start Address
Start address of standard Message ID filter list (32-bit word address, see Figure 6-1).

[bit1:0] Reserved: Reserved bits
When writing, always write "0". When reading, "0" is always read.

5.20. Extended ID Filter Configuration (MCG_CANFDx_XIDFC, CPG_CANFDx_XIDFC)

Settings for 29-bit extended Message ID filtering. The Extended ID Filter Configuration controls the filter path for extended messages as described in Figure 3-5.

Write access to the Extended ID Filter Configuration (XIDFC) is possible only when the Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) bits are both "1".

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

Bit	23	22	21	20	19	18	17	16
Field	Reserved	LSE[6:0]						
R/W Attribute	R0,W0	R/W						
Protection Attribute	-	-						
Initial value	0	0000000						

Bit	15	14	13	12	11	10	9	8
Field	FLESA[15:8]							
R/W Attribute	R/W							
Protection Attribute	-							
Initial value	00000000							

Bit	7	6	5	4	3	2	1	0
Field	FLESA[7:2]						Reserved	
R/W Attribute	R/W						R0,W0	
Protection Attribute	-						-	
Initial value	000000						00	

[bit31:23] Reserved: Reserved bits

When writing, always write "0". When reading, "0" is always read.

[bit22:16] LSE[6:0]: List Size Extended

LSE[6:0]	Description
0	No extended Message ID filter.
1 to 64	Number of extended Message ID filter elements.
>64	Values greater than 64 are interpreted as 64.

[bit15:2] FLESA[15:2]: Filter List Extended Start Address
Start address of extended Message ID filter list (32-bit word address, see Figure 6-1).

[bit1:0] Reserved: Reserved bits
When writing, always write "0". When reading, "0" is always read.

5.21. Extended ID AND Mask (MCG_CANFDx_XIDAM, CPG_CANFDx_XIDAM)

The Extended ID AND Mask defines the valid bits of a 29-bit ID for acceptance filtering.

Write access to the Extended ID AND Mask (XIDAM) is possible only when the Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) bits are both "1".

bit	31	30	29	28	27	26	25	24
Field	Reserved			EIDM[28:24]				
R/W Attribute	R0,W0			R/W				
Protection Attribute	-			-				
Initial value	000			11111				

Bit	23	22	21	20	19	18	17	16
Field	EIDM[23:16]							
R/W Attribute	R/W							
Protection Attribute	-							
Initial value	0xFF							

Bit	15	14	13	12	11	10	9	8
Field	EIDM[15:8]							
R/W Attribute	R/W							
Protection Attribute	-							
Initial value	0xFF							

bit	7	6	5	4	3	2	1	0
Field	EIDM[7:0]							
R/W Attribute	R/W							
Protection Attribute	-							
Initial value	0xFF							

[bit31:29] Reserved: Reserved bits

When writing, always write "0". When reading, "0" is always read.

[bit28:0] EIDM[28:0]: Extended ID Mask

For acceptance filtering of extended frames the Extended ID AND Mask is ANDed with the Message ID of a received frame.

With the reset value of all bits set to one the mask is not active.

5.22. High Priority Message Status (MCG_CANFDx_HPMS, CPG_CANFDx_HPMS)

This register is updated every time a Message ID filter element configured to generate a priority event matches. This can be used to monitor the status of incoming high priority messages and to enable fast access to these messages.

Note:

- *If all the following conditions are fulfilled, set at least 1 to both List Size Standard (SIDFC.LSS[7:0]) and List Size Extended (XIDFC.LSE[6:0]).*
  - *To use high priority events*
  - *To use both standard and extended frame format*

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	RX,W0							
Protection Attribute	-							
Initial value	0xXX							

Bit	23	22	21	20	19	18	17	16
Field	Reserved							
R/W Attribute	RX,W0							
Protection Attribute	-							
Initial value	0xXX							

Bit	15	14	13	12	11	10	9	8
Field	FLST	FIDX[6:0]						
R/W Attribute	R,WX	R,WX						
Protection Attribute	-	-						
Initial value	0	0000000						

Bit	7	6	5	4	3	2	1	0
Field	MSI[1:0]		BIDX[5:0]					
R/W Attribute	R,WX		R,WX					
Protection Attribute	-		-					
Initial value	00		000000					

[bit31:16] Reserved: Reserved bits

When writing, always write "0". The read value is undefined.

[bit15] FLST: Filter List

Indicates the filter list of the matching filter element.

Bit	Description
0	Standard Filter List.
1	Extended Filter List.

[bit14:8] FIDX[6:0]: Filter Index

FIDX[6:0]	Description
0 to 127	Index of matching Rx acceptance filter element. Range is 0 to List Size Standard/Extended minus 1 (i.e. SIDFC.LSS[7:0] - 1 resp. XIDFC.LSE[6:0] - 1).

[bit7:6] MSI[1:0]: Message Storage Indicator

Bits	Description
00	No Rx FIFO selected.
01	Rx FIFO message lost.
10	Message stored in Rx FIFO 0.
11	Message stored in Rx FIFO 1.

[bit5:0] BIDX[5:0]: Buffer Index

Index of Rx FIFO element to which the message was stored. Only valid when bit[1] of the Message Storage Indicator MSI[1] = 1.

5.23. New Data 1 (MCG_CANFDx_NDAT1, CPG_CANFDx_NDAT1)

New Data 1 holds flags that are set when the respective dedicated Rx Buffer receives a frame.

bit	31	30	29	28	27	26	25	24
Field	ND31	ND30	ND29	ND28	ND27	ND26	ND25	ND24
R/W Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	ND23	ND22	ND21	ND20	ND19	ND18	ND17	ND16
R/W Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	ND15	ND14	ND13	ND12	ND11	ND10	ND9	ND8
R/W Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	ND7	ND6	ND5	ND4	ND3	ND2	ND1	ND0
R/W Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

[bit31:0] ND[31:0]: New Data

The register holds the New Data flags of dedicated Rx Buffers 0 to 31.

The flags are set when the respective dedicated Rx Buffer has been updated from a received frame. The flags remain set until the CPU clears them.

A flag is cleared by writing a "1" to the corresponding bit position. Writing a "0" has no effect. A hard reset will clear the register.

Bit	Description
0	Rx Buffer not updated.
1	Rx Buffer updated from new message.

5.24. New Data 2 (MCG_CANFDx_NDAT 2, CPG_CANFDx_NDAT 2)

New Data 2 holds flags that are set when the respective dedicated Rx Buffer receives a frame.

bit	31	30	29	28	27	26	25	24
Field	ND63	ND62	ND61	ND60	ND59	ND58	ND57	ND56
R/W Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	ND55	ND54	ND53	ND52	ND51	ND50	ND49	ND48
R/W Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	ND47	ND46	ND45	ND44	ND43	ND42	ND41	ND40
R/W Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	ND39	ND38	ND37	ND36	ND35	ND34	ND33	ND32
R/W Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

[bit31:0] ND[63:32]: New Data

The register holds the New Data flags of dedicated Rx Buffers 32 to 63.

The flags are set when the respective dedicated Rx Buffer has been updated from a received frame. The flags remain set until the CPU clears them.

A flag is cleared by writing a "1" to the corresponding bit position. Writing a "0" has no effect. A hard reset will clear the register.

Bit	Description
0	Rx Buffer not updated.
1	Rx Buffer updated from new message.

5.25. Rx FIFO 0 Configuration (MCG_CANFDx_RXF0C, CPG_CANFDx_RXF0C)

Settings for the Rx FIFO 0.

Write access to the Rx FIFO 0 Configuration (RXF0C) is possible only when the Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) bits are both "1".

bit	31	30	29	28	27	26	25	24
Field	F0OM	F0WM[6:0]						
R/W Attribute	R/W	R/W						
Protection Attribute	-	-						
Initial value	0	0000000						

bit	23	22	21	20	19	18	17	16
Field	Reserved	F0S[6:0]						
R/W Attribute	R0,W0	R/W						
Protection Attribute	-	-						
Initial value	0	0000000						

Bit	15	14	13	12	11	10	9	8
Field	F0SA[15:8]							
R/W Attribute	R/W							
Protection Attribute	-							
Initial value	00000000							

Bit	7	6	5	4	3	2	1	0
Field	F0SA[7:2]						Reserved	
R/W Attribute	R/W						R0,W0	
Protection Attribute	-						-	
Initial value	000000						00	

[bit31] F0OM: FIFO 0 Operation Mode

Rx FIFO 0 can be operated in blocking or in overwrite mode (see "3.4.2. Rx FIFOs").

Bit	Description
0	Rx FIFO 0 blocking mode.
1	Rx FIFO 0 overwrite mode.

[bit30:24] F0WM[6:0]: Rx FIFO 0 Watermark

F0WM[6:0]	Description
0	Watermark interrupt disabled.
1 to 64	Level for Rx FIFO 0 Watermark Reached interrupt (IR.RF0W).
>64	Watermark interrupt disabled.

[bit23] Reserved: Reserved bit

When writing, always write "0". When reading, "0" is always read.

[bit22:16] F0S[6:0]: Rx FIFO 0 Size

The Rx FIFO 0 elements are indexed from 0 to F0S[6:0] - 1.

F0S[6:0]	Description
0	No Rx FIFO 0.
1 to 64	Number of Rx FIFO 0 elements.
>64	Values greater than 64 are interpreted as 64.

[bit15:2] F0SA[15:2]: Rx FIFO 0 Start Address

Start address of Rx FIFO 0 in Message RAM (32-bit word address, see Figure 6-1).

[bit1:0] Reserved: Reserved bits

When writing, always write "0". When reading, "0" is always read.

5.26. Rx FIFO 0 Status (MCG_CANFDx_RXF0S, CPG_CANFDx_RXF0S)

Status of the Rx FIFO 0.

bit	31	30	29	28	27	26	25	24
Field	Reserved						RF0L	F0F
R/W Attribute	RX,W0						R,WX	R,WX
Protection Attribute	-						-	-
Initial value	XXXXXX						0	0

Bit	23	22	21	20	19	18	17	16
Field	Reserved		F0PI[5:0]					
R/W Attribute	RX,W0		R,WX					
Protection Attribute	-		-					
Initial value	XX		000000					

Bit	15	14	13	12	11	10	9	8
Field	Reserved		F0GI[5:0]					
R/W Attribute	RX,W0		R,WX					
Protection Attribute	-		-					
Initial value	XX		000000					

Bit	7	6	5	4	3	2	1	0
Field	Reserved	F0FL[6:0]						
R/W Attribute	RX,W0	R,WX						
Protection Attribute	-	-						
Initial value	X	0000000						

[bit31:26] Reserved: Reserved bits

When writing, always write "0". The read value is undefined.

[bit25] RF0L: Rx FIFO 0 Message Lost

This bit is a copy of interrupt flag IR.RF0L (Rx FIFO 0 Message Lost).

When IR.RF0L is reset, this bit is also reset.

Bit	Description
0	No Rx FIFO 0 message lost.
1	Rx FIFO 0 message lost. Also set after write attempt to Rx FIFO 0 of size zero.

Note:

- *Overwriting a message when the FIFO is in overwrite mode (RXF0C.F0OM = 1) will not set this flag.*

[bit24] F0F: Rx FIFO 0 Full

Bit	Description
0	Rx FIFO 0 not full.
1	Rx FIFO 0 full.

[bit23:22] Reserved: Reserved bits

When writing, always write "0". The read value is undefined.

[bit21:16] F0PI[5:0]: Rx FIFO 0 Put Index

F0PI[5:0]	Description
0 to 63	Rx FIFO 0 write index pointer, range 0 to 63.

[bit15:14] Reserved: Reserved bits

When writing, always write "0". The read value is undefined.

[bit13:8] F0GI[5:0]: Rx FIFO 0 Get Index

F0GI[5:0]	Description
0 to 63	Rx FIFO 0 read index pointer, range 0 to 63.

[bit7] Reserved: Reserved bit

When writing, always write "0". The read value is undefined.

[bit6:0] F0FL[6:0]: Rx FIFO 0 Fill Level

F0FL[6:0]	Description
0 to 64	Number of elements stored in Rx FIFO 0, range 0 to 64.

5.27. Rx FIFO 0 Acknowledge (MCG_CANFDx_RXF0A, CPG_CANFDx_RXF0A)

The Rx FIFO 0 Acknowledge is used to acknowledge that the CPU has read a message or a sequence of messages from the Rx FIFO 0 to indicate to the CAN FD Controller that the corresponding Message RAM area may be released. See "3.6. FIFO Acknowledge Handling" for details.

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

Bit	23	22	21	20	19	18	17	16
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

Bit	15	14	13	12	11	10	9	8
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

Bit	7	6	5	4	3	2	1	0
Field	Reserved		F0AI[5:0]					
R/W Attribute	R0,W0		R/W					
Protection Attribute	-		-					
Initial value	00		000000					

[bit31:6] Reserved: Reserved bits
When writing, always write "0". When reading, "0" is always read.

[bit5:0] F0AI[5:0]: Rx FIFO 0 Acknowledge Index
After the CPU has read a message or a sequence of messages from Rx FIFO 0, it has to write the buffer index of the last element read from Rx FIFO 0 to this field (F0AI[5:0]). This will set the Rx FIFO 0 Get Index RXF0S.F0GI[5:0] to F0AI[5:0] + 1 and update the FIFO 0 Fill Level RXF0S.F0FL[6:0].

5.28. Rx Buffer Configuration (MCG_CANFDx_RXBC, CPG_CANFDx_RXBC)

Defines the start address of the Rx Buffers section in the Message RAM.

Write access to the Rx Buffer Configuration (RXBC) is possible only when the Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) bits are both "1".

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

bit	23	22	21	20	19	18	17	16
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

Bit	15	14	13	12	11	10	9	8
Field	RBSA[15:8]							
R/W Attribute	R/W							
Protection Attribute	-							
Initial value	00000000							

bit	7	6	5	4	3	2	1	0
Field	RBSA[7:2]						Reserved	
R/W Attribute	R/W						R0,W0	
Protection Attribute	-						-	
Initial value	000000						00	

[bit31:16] Reserved: Reserved bits

When writing, always write "0". When reading, "0" is always read.

[bit15:2] RBSA[15:2]: Rx Buffer Start Address

Configures the start address of the Rx Buffers section in the Message RAM (32-bit word address).

Also used to reference debug messages A, B, C.

[bit1:0] Reserved: Reserved bits

When writing, always write "0". When reading, "0" is always read.

5.29. Rx FIFO 1 Configuration (MCG_CANFDx_RXF1C, CPG_CANFDx_RXF1C)

Settings for the Rx FIFO 1.

Write access to the Rx FIFO 1 Configuration (RXF1C) is possible only when the Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) bits are both "1".

bit	31	30	29	28	27	26	25	24
Field	F1OM	F1WM[6:0]						
R/W Attribute	R/W	R/W						
Protection Attribute	-	-						
Initial value	0	0000000						

bit	23	22	21	20	19	18	17	16
Field	Reserved	F1S[6:0]						
R/W Attribute	R0,W0	R/W						
Protection Attribute	-	-						
Initial value	0	0000000						

bit	15	14	13	12	11	10	9	8
Field	F1SA[15:8]							
R/W Attribute	R/W							
Protection Attribute	-							
Initial value	00000000							

bit	7	6	5	4	3	2	1	0
Field	F1SA[7:2]						Reserved	
R/W Attribute	R/W						R0,W0	
Protection Attribute	-						-	
Initial value	000000						00	

[bit31] F1OM: FIFO 1 Operation Mode

Rx FIFO 1 can be operated in blocking or in overwrite mode (see "3.4.2. Rx FIFOs").

bit	Description
0	Rx FIFO 1 blocking mode.
1	Rx FIFO 1 overwrite mode.

[bit30:24] F1WM[6:0]: Rx FIFO 1 Watermark

F1WM[6:0]	Description
0	Watermark interrupt disabled.
1 to 64	Level for Rx FIFO 1 Watermark Reached interrupt (IR.RF1W).
>64	Watermark interrupt disabled.

[bit23] Reserved: Reserved bit

When writing, always write "0". When reading, "0" is always read.

[bit22:16] F1S[6:0]: Rx FIFO 1 Size

The Rx FIFO 1 elements are indexed from 0 to F1S[6:0] - 1.

F1S[6:0]	Description
0	No Rx FIFO 1.
1 to 64	Number of Rx FIFO 1 elements.
>64	Values greater than 64 are interpreted as 64.

[bit15:2] F1SA[15:2]: Rx FIFO 1 Start Address

Start address of Rx FIFO 1 in Message RAM (32-bit word address, see Figure 6-1).

[bit1:0] Reserved: Reserved bits

When writing, always write "0". When reading, "0" is always read.

5.30. Rx FIFO 1 Status (MCG_CANFDx_RXF1S, CPG_CANFDx_RXF1S)

Status of the Rx FIFO 1.

bit	31	30	29	28	27	26	25	24
Field	DMS[1:0]		Reserved				RF1L	F1F
R/W Attribute	R,WX		RX,W0				R,WX	R,WX
Protection Attribute	-		-				-	-
Initial value	00		XXXX				0	0

bit	23	22	21	20	19	18	17	16
Field	Reserved		F1PI[5:0]					
R/W Attribute	RX,W0		R,WX					
Protection Attribute	-		-					
Initial value	XX		000000					

Bit	15	14	13	12	11	10	9	8
Field	Reserved		F1GI[5:0]					
R/W Attribute	RX,W0		R,WX					
Protection Attribute	-		-					
Initial value	XX		000000					

Bit	7	6	5	4	3	2	1	0
Field	Reserved	F1FL[6:0]						
R/W Attribute	RX,W0	R,WX						
Protection Attribute	-	-						
Initial value	X	0000000						

[bit31:30] DMS[1:0]: Debug Message Status

Bits	Description
00	Idle state, wait for reception of debug messages, DMA request is cleared.
01	Debug message A received.
10	Debug messages A, B received.
11	Debug messages A, B, C received, DMA request is set.

Notes:

- *Debug message is used for Debug on CAN feature.*
- *Debug on CAN feature is not supported for TRAVEO™ T1G Platform.*

[bit29:26] Reserved: Reserved bits

When writing, always write "0". The read value is undefined.

[bit25] RF1L: Rx FIFO 1 Message Lost

This bit is a copy of interrupt flag IR.RF1L (Rx FIFO 1 Message Lost).

When IR.RF1L is reset, this bit is also reset.

Bit	Description
0	No Rx FIFO 1 message lost.
1	Rx FIFO 1 message lost. Also set after write attempt to Rx FIFO 1 of size zero.

Note:

- *Overwriting a message when the FIFO is in overwrite mode (RXF1C.F1OM = 1) will not set this flag.*

[bit24] F1F: Rx FIFO 1 Full

Bit	Description
0	Rx FIFO 1 not full.
1	Rx FIFO 1 full.

[bit23:22] Reserved: Reserved bits

When writing, always write "0". The read value is undefined.

[bit21:16] F1PI[5:0]: Rx FIFO 1 Put Index

F1PI[5:0]	Description
0 to 63	Rx FIFO 1 write index pointer, range 0 to 63.

[bit15:14] Reserved: Reserved bits

When writing, always write "0". The read value is undefined.

[bit13:8] F1GI[5:0]: Rx FIFO 1 Get Index

F1GI[5:0]	Description
0 to 63	Rx FIFO 1 read index pointer, range 0 to 63.

[bit7] Reserved: Reserved bit

When writing, always write "0". The read value is undefined.

[bit6:0] F1FL[6:0]: Rx FIFO 1 Fill Level

F1FL[6:0]	Description
0 to 64	Number of elements stored in Rx FIFO 1, range 0 to 64.

5.31. Rx FIFO 1 Acknowledge (MCG_CANFDx_RXF1A, CPG_CANFDx_RXF1A)

The Rx FIFO 1 Acknowledge is used to acknowledge that the CPU has read a message or a sequence of messages from the Rx FIFO 1 to indicate to the CAN FD Controller that the corresponding Message RAM area may be released. See "3.6. FIFO Acknowledge Handling" for details.

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

Bit	23	22	21	20	19	18	17	16
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

Bit	15	14	13	12	11	10	9	8
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

Bit	7	6	5	4	3	2	1	0
Field	Reserved		F1AI[5:0]					
R/W Attribute	R0,W0		R/W					
Protection Attribute	-		-					
Initial value	00		000000					

[bit31:6] Reserved: Reserved bits
When writing, always write "0". When reading, "0" is always read.

[bit5:0] F1AI[5:0]: Rx FIFO 1 Acknowledge Index
After the CPU has read a message or a sequence of messages from Rx FIFO 1, it has to write the buffer index of the last element read from Rx FIFO 1 to this field (F1AI[5:0]). This will set the Rx FIFO 1 Get Index RXF1S.F1GI[5:0] to F1AI[5:0] + 1 and update the FIFO 1 Fill Level RXF1S.F1FL[6:0].

5.32. Rx Buffer/FIFO Element Size Configuration (MCG_CANFDx_RXESC, CPG_CANFDx_RXESC)

Configures the number of data bytes belonging to an Rx Buffer and FIFO element. Data field sizes > 8 bytes are intended for CAN FD operation only.

Write access to the Rx Buffer/FIFO Element Size Configuration (RXESC) is possible only when the Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) bits are both "1".

Note:

- *In case the data field size of an accepted CAN frame exceeds the data field size configured for the matching Rx Buffer or Rx FIFO, only the number of bytes as configured by this register (i.e. RXESC) are stored to the Rx Buffer resp. Rx FIFO element. The rest of the frame's data field is ignored.*

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

Bit	23	22	21	20	19	18	17	16
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

Bit	15	14	13	12	11	10	9	8
Field	Reserved					RBDS[2:0]		
R/W Attribute	R0,W0					R/W		
Protection Attribute	-					-		
Initial value	00000					000		

Bit	7	6	5	4	3	2	1	0
Field	Reserved	F1DS[2:0]			Reserved	F0DS[2:0]		
R/W Attribute	R0,W0	R/W			R0,W0	R/W		
Protection Attribute	-	-			-	-		
Initial value	0	000			0	000		

[bit31:11] Reserved: Reserved bits

When writing, always write "0". When reading, "0" is always read.

[bit10:8] RBDS[2:0]: Rx Buffer Data Field Size

Bits	Description
000	8 byte data field.
001	12 byte data field.
010	16 byte data field.
011	20 byte data field.
100	24 byte data field.
101	32 byte data field.
110	48 byte data field.
111	64 byte data field.

[bit7] Reserved: Reserved bit

When writing, always write "0". When reading, "0" is always read.

[bit6:4] F1DS[2:0]: Rx FIFO 1 Data Field Size

Bits	Description
000	8 byte data field.
001	12 byte data field.
010	16 byte data field.
011	20 byte data field.
100	24 byte data field.
101	32 byte data field.
110	48 byte data field.
111	64 byte data field.

[bit3] Reserved: Reserved bit

When writing, always write "0". When reading, "0" is always read.

[bit2:0] F0DS[2:0]: Rx FIFO 0 Data Field Size

Bits	Description
000	8 byte data field.
001	12 byte data field.
010	16 byte data field.
011	20 byte data field.
100	24 byte data field.
101	32 byte data field.
110	48 byte data field.
111	64 byte data field.

5.33. Tx Buffer Configuration (MCG_CANFDx_TXBC, CPG_CANFDx_TXBC)

Settings for Tx Buffers stored in the Message RAM.

Write access to the Tx Buffer Configuration (TXBC) is possible only when the Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) bits are both "1".

Note:

- *Be aware that the sum of Transmit FIFO/Queue Size (TXBC.TFQS[5:0]) and Number of Dedicated Transmit Buffers (TXBC,NDTB[5:0]) may be not greater than 32. There is no check for erroneous configurations.*
- *The Tx Buffers section in the Message RAM starts with the dedicated Tx Buffers.*
- *Don't use combination Tx FIFO and Dedicated Tx Buffers. In case of setting "1" to Tx FIFO/Queue Mode (TXBC.TFQM), set 0 to TXBC.NDTB[5:0].*

bit	31	30	29	28	27	26	25	24
Field	Reserved	TFQM	TFQS[5:0]					
R/W Attribute	R0,W0	R/W	R/W					
Protection Attribute	-	-	-					
Initial value	0	0	000000					

bit	23	22	21	20	19	18	17	16
Field	Reserved		NDTB[5:0]					
R/W Attribute	R0,W0		R/W					
Protection Attribute	-		-					
Initial value	00		000000					

bit	15	14	13	12	11	10	9	8
Field	TBSA[15:8]							
R/W Attribute	R/W							
Protection Attribute	-							
Initial value	00000000							

bit	7	6	5	4	3	2	1	0
Field	TBSA[7:2]						Reserved	
R/W Attribute	R/W						R0,W0	
Protection Attribute	-						-	
Initial value	000000						00	

[bit31] Reserved: Reserved bit

When writing, always write "0". When reading, "0" is always read.

[bit30] TFQM: Tx FIFO/Queue Mode

Bit	Description
0	Tx FIFO operation.
1	Tx Queue operation.

[bit29:24] TFQS[5:0]: Transmit FIFO/Queue Size

TFQS[5:0]	Description
0	No Tx FIFO/Queue.
1 to 32	Number of Tx Buffers used for Tx FIFO/Queue.
>32	Values greater than 32 are interpreted as 32.

[bit23:22] Reserved: Reserved bits

When writing, always write "0". When reading, "0" is always read.

[bit21:16] NDTB[5:0]: Number of Dedicated Transmit Buffers

NDTB[5:0]	Description
0	No Dedicated Tx Buffers.
1 to 32	Number of Dedicated Tx Buffers.
>32	Values greater than 32 are interpreted as 32.

[bit15:2] TBSA[15:2]: Tx Buffers Start Address

Start address of Tx Buffers section in Message RAM (32-bit word address, see Figure 6-1).

[bit1:0] Reserved: Reserved bits

When writing, always write "0". When reading, "0" is always read.

5.34. Tx FIFO/Queue Status (MCG_CANFDx_TXFQS, CPG_CANFDx_TXFQS)

The Tx FIFO/Queue status is related to the pending Tx requests listed in the Tx Buffer Request Pending register TXBRP. Therefore the effect of Add/Cancellation requests may be delayed due to a running Tx scan (TXBRP not yet updated).

Note:

- *In case of mixed configurations where dedicated Tx Buffers are combined with a Tx FIFO or a Tx Queue, the Put and Get Indices (TXFQS.TFQPI[4:0] resp. TXFQS.TFGI[4:0]) indicate the number of the Tx Buffer starting with the first dedicated Tx Buffers.*

Example:

- *For a configuration of 12 dedicated Tx Buffers and a Tx FIFO of 20 Buffers a Put Index of 15 points to the fourth buffer of the Tx FIFO.*

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	RX,W0							
Protection Attribute	-							
Initial value	0xXX							

bit	23	22	21	20	19	18	17	16
Field	Reserved		TFQF	TFQPI[4:0]				
R/W Attribute	RX,W0		R,WX	R,WX				
Protection Attribute	-		-	-				
Initial value	XX		0	00000				

Bit	15	14	13	12	11	10	9	8
Field	Reserved			TFGI[4:0]				
R/W Attribute	RX,W0			R,WX				
Protection Attribute	-			-				
Initial value	XXX			00000				

Bit	7	6	5	4	3	2	1	0
Field	Reserved		TFFL[5:0]					
R/W Attribute	RX,W0		R,WX					
Protection Attribute	-		-					
Initial value	XX		000000					

[bit31:22] Reserved: Reserved bits

When writing, always write "0". The read value is undefined.

[bit21] TFQF: Tx FIFO/Queue Full

Bit	Description
0	Tx FIFO/Queue not full.
1	Tx FIFO/Queue full.

[bit20:16] TFQPI[4:0]: Tx FIFO/Queue Put Index

TFQPI[4:0]	Description
0 to 31	Tx FIFO/Queue write index pointer, range 0 to 31.

[bit15:13] Reserved: Reserved bits

When writing, always write "0". The read value is undefined.

[bit12:8] TFGI[4:0]: Tx FIFO Get Index

TFGI[4:0]	Description
0 to 31	Tx FIFO read index pointer, range 0 to 31. Read as zero when Tx Queue operation is configured (Tx FIFO/Queue Mode TXBC.TFQM = 1).

[bit7:6] Reserved: Reserved bits

When writing, always write "0". The read value is undefined.

[bit5:0] TFFL[5:0]: Tx FIFO Free Level

TFFL[5:0]	Description
0 to 32	Number of consecutive free Tx FIFO elements starting from the Tx FIFO Get Index (TXFQS.TFGI[4:0]), range 0 to 32. Read as zero when Tx Queue operation is configured (Tx FIFO/Queue Mode TXBC.TFQM = 1).

5.35. Tx Buffer Element Size Configuration (MCG_CANFDx_TXESC, CPG_CANFDx_TXESC)

Configures the number of data bytes belonging to a Tx Buffer element. Data field sizes > 8 bytes are intended for CAN FD operation only.

Write access to the Tx Buffer Element Size Configuration (TXESC) is possible only when the Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) bits are both "1".

Note:

- *In case the data length code DLC of a Tx Buffer element is configured to a value higher than the Tx Buffer data field size TXESC.TBDS[2:0], the bytes not defined by the Tx Buffer are transmitted as 0xCC (padding bytes).*

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

Bit	23	22	21	20	19	18	17	16
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

Bit	15	14	13	12	11	10	9	8
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

Bit	7	6	5	4	3	2	1	0
Field	Reserved					TBDS[2:0]		
R/W Attribute	R0,W0					R/W		
Protection Attribute	-					-		
Initial value	00000					000		

[bit31:3] Reserved: Reserved bits

When writing, always write "0". When reading, "0" is always read.

[bit2:0] TBDS[2:0]: Tx Buffer Data Field Size

Bits	Description
000	8 byte data field.
001	12 byte data field.
010	16 byte data field.
011	20 byte data field.
100	24 byte data field.
101	32 byte data field.
110	48 byte data field.
111	64 byte data field.

5.36. Tx Buffer Request Pending (MCG_CANFDx_TXBRP, CPG_CANFDx_TXBRP)

Tx Buffer Request Pending holds the status of the transmission requests of each corresponding Tx Buffer.

bit	31	30	29	28	27	26	25	24
Field	TRP31	TRP30	TRP29	TRP28	TRP27	TRP26	TRP25	TRP24
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	TRP23	TRP22	TRP21	TRP20	TRP19	TRP18	TRP17	TRP16
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	TRP15	TRP14	TRP13	TRP12	TRP11	TRP10	TRP9	TRP8
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	TRP7	TRP6	TRP5	TRP4	TRP3	TRP2	TRP1	TRP0
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

[bit31:0] TRP[31:0]: Transmission Request Pending

Each Tx Buffer has its own Transmission Request Pending bit. The bits are set via register TXBAR (Tx Buffer Add Request). The bits are reset after a requested transmission has completed or has been cancelled via register TXBCR (Tx Buffer Cancellation Request).

TXBRP bits are set only for those Tx Buffers that are configured by the TXBC.TFQS[5:0] (Transmit FIFO/Queue Size) and TXBC.NDTB[5:0] (Number of Dedicated Transmit Buffers) fields. After a TXBRP bit has been set, a Tx scan (see "3.5. Tx Handling") is started to check for the pending Tx request with the highest priority (Tx Buffer with lowest Message ID).

Bit	Description
0	No transmission request pending.
1	Transmission request pending.

Note:

- *TXBRP bits which are set while a Tx scan is in progress are not considered during this particular Tx scan. In case a cancellation is requested for such a Tx Buffer, this Add Request is cancelled immediately, the corresponding TXBRP bit is reset.*

5.37. Tx Buffer Add Request (MCG_CANFDx_TXBAR, CPG_CANFDx_TXBAR)

Tx Buffer Add Request is used to request the transmission of each corresponding Tx Buffer.

bit	31	30	29	28	27	26	25	24
Field	AR31	AR30	AR29	AR28	AR27	AR26	AR25	AR24
R/W Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	AR23	AR22	AR21	AR20	AR19	AR18	AR17	AR16
R/W Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	AR15	AR14	AR13	AR12	AR11	AR10	AR9	AR8
R/W Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	AR7	AR6	AR5	AR4	AR3	AR2	AR1	AR0
R/W Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

[bit31:0] AR[31:0]: Add Request

Each Tx Buffer has its own Add Request bit. Writing a "1" will set the corresponding Add Request bit; writing a "0" has no impact. This enables the CPU to set transmission requests for multiple Tx Buffers with one write to TXBAR.

TXBAR bits are set only for those Tx Buffers that are configured by the TXBC.TFQS[5:0] (Transmit FIFO/Queue Size) and TXBC.NDTB[5:0] (Number of Dedicated Transmit Buffers) fields.

When no Tx scan is running, the bits are reset immediately, else the bits remain set until the Tx scan process has completed.

Bit	Description
0	No transmission request added.
1	Transmission requested added.

Note:

- *If an add request is applied for a Tx Buffer with pending transmission request (corresponding TXBRP bit already set), this add request is ignored.*

5.38. Tx Buffer Cancellation Request (MCG_CANFDx_TXBCR, CPG_CANFDx_TXBCR)

Used to cancel transmission requests of each corresponding Tx Buffer.

bit	31	30	29	28	27	26	25	24
Field	CR31	CR30	CR29	CR28	CR27	CR26	CR25	CR24
R/W Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	CR23	CR22	CR21	CR20	CR19	CR18	CR17	CR16
R/W Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	CR15	CR14	CR13	CR12	CR11	CR10	CR9	CR8
R/W Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	CR7	CR6	CR5	CR4	CR3	CR2	CR1	CR0
R/W Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

[bit31:0] CR[31:0]: Cancellation Request

Each Tx Buffer has its own Cancellation Request bit. Writing a "1" will set the corresponding Cancellation Request bit; writing a "0" has no impact. This enables the CPU to set cancellation requests for multiple Tx Buffers with one write to TXBCR.

TXBCR bits are set only for those Tx Buffers that are configured by the TXBC.TFQS[5:0] (Transmit FIFO/Queue Size) and TXBC.NDTB[5:0] (Number of Dedicated Transmit Buffers) fields. The bits remain set until the corresponding bit of TXBRP (Tx Buffer Request Pending) is reset.

Bit	Description
0	No cancellation pending.
1	Cancellation pending.

5.39. Tx Buffer Transmission Occurred (MCG_CANFDx_TXBTO, CPG_CANFDx_TXBTO)

Displays the status of whether the corresponding Tx Buffer has been transmitted or not.

bit	31	30	29	28	27	26	25	24
Field	TO31	TO30	TO29	TO28	TO27	TO26	TO25	TO24
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	TO23	TO22	TO21	TO20	TO19	TO18	TO17	TO16
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	TO15	TO14	TO13	TO12	TO11	TO10	TO9	TO8
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	TO7	TO6	TO5	TO4	TO3	TO2	TO1	TO0
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

[bit31:0] TO[31:0]: Transmission Occurred

Each Tx Buffer has its own Transmission Occurred bit.

The bits are set when the corresponding TXBRP (Tx Buffer Request Pending) bit is cleared after a successful transmission.

The bits are reset when a new transmission is requested by writing a "1" to the corresponding bit of register TXBAR (Tx Buffer Add Request).

Bit	Description
0	No transmission occurred.
1	Transmission occurred.

5.40. Tx Buffer Cancellation Finished (MCG_CANFDx_TXBCF, CPG_CANFDx_TXBCF)

Signals whether the cancellation request of the corresponding Tx Buffer has been successful or not.

bit	31	30	29	28	27	26	25	24
Field	CF31	CF30	CF29	CF28	CF27	CF26	CF25	CF24
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	CF23	CF22	CF21	CF20	CF19	CF18	CF17	CF16
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	CF15	CF14	CF13	CF12	CF11	CF10	CF9	CF8
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	CF7	CF6	CF5	CF4	CF3	CF2	CF1	CF0
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

[bit31:0] CF[31:0]: Cancellation Finished

Each Tx Buffer has its own Cancellation Finished bit.

The bits are set when the corresponding TXBRP (Tx Buffer Request Pending) bit is cleared after a cancellation was requested via TXBCR (Tx Buffer Cancellation Request). In case the corresponding TXBRP bit was not set at the point of cancellation, CF is set immediately. CF is also set for an unsuccessful transmission when in DAR mode.

The bits are reset when a new transmission is requested by writing a "1" to the corresponding bit of register TXBAR (Tx Buffer Add Request).

Bit	Description
0	No transmit buffer cancellation.
1	Transmit buffer cancellation finished.

5.41. Tx Buffer Transmission Interrupt Enable (MCG_CANFDx_TXBTIE, CPG_CANFDx_TXBTIE)

The settings in the Tx Buffer Transmission Interrupt Enable determine which Tx Buffer will assert an interrupt upon transmission.

bit	31	30	29	28	27	26	25	24
Field	TIE31	TIE30	TIE29	TIE28	TIE27	TIE26	TIE25	TIE24
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	TIE23	TIE22	TIE21	TIE20	TIE19	TIE18	TIE17	TIE16
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	TIE15	TIE14	TIE13	TIE12	TIE11	TIE10	TIE9	TIE8
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	TIE7	TIE6	TIE5	TIE4	TIE3	TIE2	TIE1	TIE0
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

[bit31:0] TIE[31:0]: Transmission Interrupt Enable

Each Tx Buffer has its own Transmission Interrupt Enable bit.

Bit	Description
0	Transmission interrupt disabled.
1	Transmission interrupt enable.

5.42. Tx Buffer Cancellation Finished Interrupt Enable (MCG_CANFDx_TXBCIE, CPG_CANFDx_TXBCIE)

The settings in the Tx Buffer Cancellation Finished Interrupt Enable determine which Tx Buffer will assert an interrupt upon completion of a transmission cancellation request.

bit	31	30	29	28	27	26	25	24
Field	CFIE31	CFIE30	CFIE29	CFIE28	CFIE27	CFIE26	CFIE25	CFIE24
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	CFIE23	CFIE22	CFIE21	CFIE20	CFIE19	CFIE18	CFIE17	CFIE16
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	CFIE15	CFIE14	CFIE13	CFIE12	CFIE11	CFIE10	CFIE9	CFIE8
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	CFIE7	CFIE6	CFIE5	CFIE4	CFIE3	CFIE2	CFIE1	CFIE0
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

[bit31:0] CFIE[31:0]: Cancellation Finished Interrupt Enable

Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit.

Bit	Description
0	Cancellation finished interrupt disabled.
1	Cancellation finished interrupt enabled.

5.43. Tx Event FIFO Configuration (MCG_CANFDx_TXEFC, CPG_CANFDx_TXEFC)

Settings for the Tx Event FIFO.

Write access to the Tx Event FIFO Configuration (TXEFC) is possible only when the Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) bits are both "1".

bit	31	30	29	28	27	26	25	24
Field	Reserved		EFWM[5:0]					
R/W Attribute	R0,W0		R/W					
Protection Attribute	-		-					
Initial value	00		000000					

Bit	23	22	21	20	19	18	17	16
Field	Reserved		EFS[5:0]					
R/W Attribute	R0,W0		R/W					
Protection Attribute	-		-					
Initial value	00		000000					

Bit	15	14	13	12	11	10	9	8
Field	EFSA[15:8]							
R/W Attribute	R/W							
Protection Attribute	-							
Initial value	00000000							

Bit	7	6	5	4	3	2	1	0
Field	EFSA[7:2]						Reserved	
R/W Attribute	R/W						R0,W0	
Protection Attribute	-						-	
Initial value	000000						00	

[bit31:30] Reserved: Reserved bits

When writing, always write "0". When reading, "0" is always read.

[bit29:24] EFWM[5:0]: Event FIFO Watermark

EFWM[5:0]	Description
0	Watermark interrupt disabled.
1 to 32	Level for Tx Event FIFO Watermark Reached interrupt (IR.TEFW).
>32	Watermark interrupt disabled.

[bit23:22] Reserved: Reserved bits
When writing, always write "0". When reading, "0" is always read.

[bit21:16] EFS[5:0]: Event FIFO Size
The Tx Event FIFO elements are indexed from 0 to EFS[5:0] - 1.

EFS[5:0]	Description
0	Tx Event FIFO disabled.
1 to 32	Number of Tx Event FIFO elements.
>32	Values greater than 32 are interpreted as 32.

[bit15:2] EFSA[15:2]: Event FIFO Start Address
Start address of Tx Event FIFO in Message RAM (32-bit word address, see Figure 6-1).

[bit1:0] Reserved: Reserved bits
When writing, always write "0". When reading, "0" is always read.

5.44. Tx Event FIFO Status (MCG_CANFDx_TXEFS, CPG_CANFDx_TXEFS)

Status of the Tx Event FIFO.

bit	31	30	29	28	27	26	25	24
Field	Reserved						TEFL	EFF
R/W Attribute	RX,W0						R,WX	R,WX
Protection Attribute	-						-	-
Initial value	XXXXXX						0	0

bit	23	22	21	20	19	18	17	16
Field	Reserved			EFPI[4:0]				
R/W Attribute	RX,W0			R,WX				
Protection Attribute	-			-				
Initial value	XXX			00000				

Bit	15	14	13	12	11	10	9	8
Field	Reserved			EFGI[4:0]				
R/W Attribute	RX,W0			R,WX				
Protection Attribute	-			-				
Initial value	XXX			00000				

bit	7	6	5	4	3	2	1	0
Field	Reserved		EFFL[5:0]					
R/W Attribute	RX,W0		R,WX					
Protection Attribute	-		-					
Initial value	XX		000000					

[bit31:26] Reserved: Reserved bits

When writing, always write "0". The read value is undefined.

[bit25] TEFL: Tx Event FIFO Element Lost

This bit is a copy of interrupt flag IR.TEFL (Tx Event FIFO Element Lost). When IR.TEFL is reset, this bit is also reset.

Bit	Description
0	No Tx Event FIFO element lost.
1	Tx Event FIFO element lost. Also set after write attempt to Tx Event FIFO of size zero.

[bit24] EFF: Event FIFO Full

Bit	Description
0	Tx Event FIFO not full.
1	Tx Event FIFO full.

[bit23:21] Reserved: Reserved bits

When writing, always write "0". The read value is undefined.

[bit20:16] EFPI[4:0]: Event FIFO Put Index

EFPI[4:0]	Description
0 to 31	Tx Event FIFO write index pointer, range 0 to 31.

[bit15:13] Reserved: Reserved bits

When writing, always write "0". The read value is undefined.

[bit12:8] EFGI[4:0]: Event FIFO Get Index

EFGI[4:0]	Description
0 to 31	Tx Event FIFO read index pointer, range 0 to 31.

[bit7:6] Reserved: Reserved bits

When writing, always write "0". The read value is undefined.

[bit5:0] EFFL[5:0]: Event FIFO Fill Level

EFFL[5:0]	Description
0 to 32	Number of elements stored in Tx Event FIFO, range 0 to 32.

5.45. Tx Event FIFO Acknowledge (MCG_CANFDx_TXEFA, CPG_CANFDx_TXEFA)

The Tx Event FIFO Acknowledge is used to acknowledge that the CPU has read an event from the Tx Event FIFO to indicate to the CAN FD Controller that the corresponding Message RAM area may be released. See "3.6. FIFO Acknowledge Handling" for details.

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

Bit	23	22	21	20	19	18	17	16
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

Bit	15	14	13	12	11	10	9	8
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

Bit	7	6	5	4	3	2	1	0
Field	Reserved			EFAI[4:0]				
R/W Attribute	R0,W0			R/W				
Protection Attribute	-			-				
Initial value	000			00000				

[bit31:5] Reserved: Reserved bits

When writing, always write "0". When reading, "0" is always read.

[bit4:0] EFAI[4:0]: Event FIFO Acknowledge Index

After the CPU has read an element or a sequence of elements from the Tx Event FIFO, it has to write the index of the last element read from Tx Event FIFO to EFAI[4:0]. This will set the Tx Event FIFO Get Index TXEFS.EFGI[4:0] to EFAI[4:0] + 1 and update the FIFO 0 Fill Level TXEFS.EFFL[5:0].

6. Message RAM

The Message RAM stores Rx/Tx messages and filter configurations.

6.1.Message RAM Configuration

6.2.Rx Buffer and FIFO Element

6.3.Tx Buffer Element

6.4.Tx Event FIFO Element

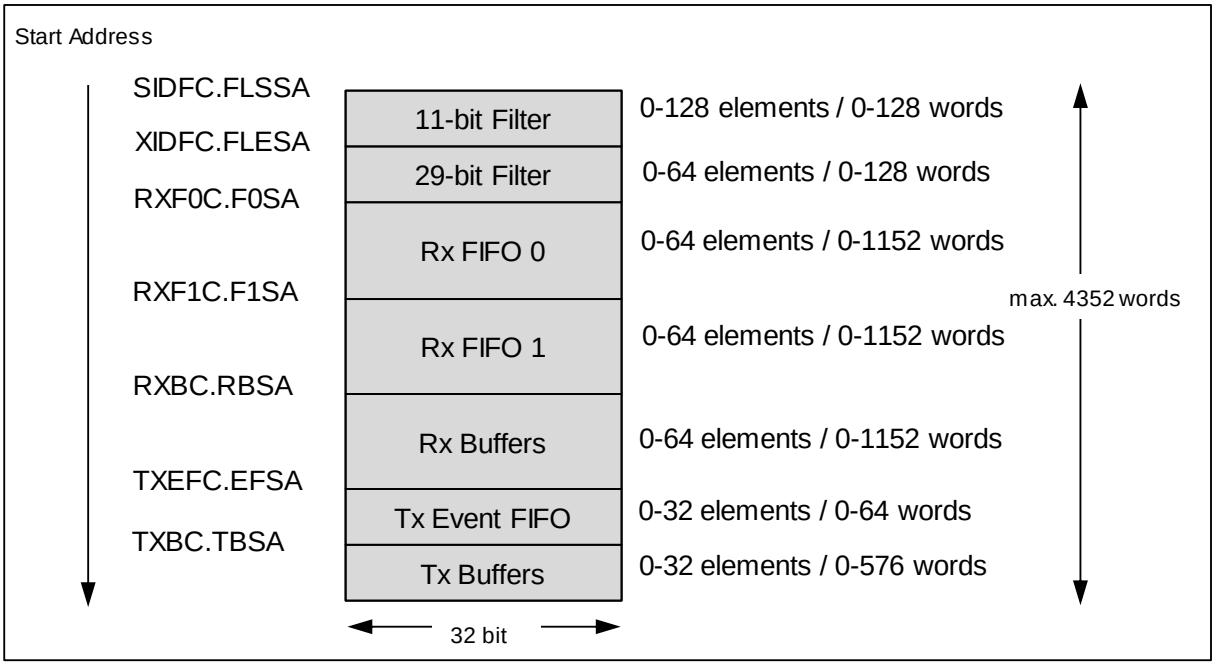
6.5.Standard Message ID Filter Element

6.6.Extended Message ID Filter Element

6.1. Message RAM Configuration

The Message RAM has a width of 32 bits. The CAN FD Controller can be configured to allocate up to 4352 words in the Message RAM (note that the number of words that can be used will be limited by the size of the actual Message RAM). It is not necessary to configure each of the sections listed in Figure 6-1, nor is there any restriction with respect to the sequence of the sections.

Figure 6-1 Message RAM Configuration



The CAN FD Controller addresses the Message RAM in 32-bit words, not single bytes. The configurable start addresses are 32-bit word addresses i.e. only bits 15 to 2 are evaluated, the two least significant bits are ignored.

Notes:

- *The CAN FD Controller does not check for erroneous configuration of the Message RAM. Especially the configuration of the start addresses of the different sections and the number of elements of each section has to be done carefully to avoid falsification or loss of data.*
- *Message RAM accesses by the CPU will take two to four Bus clock cycles.*

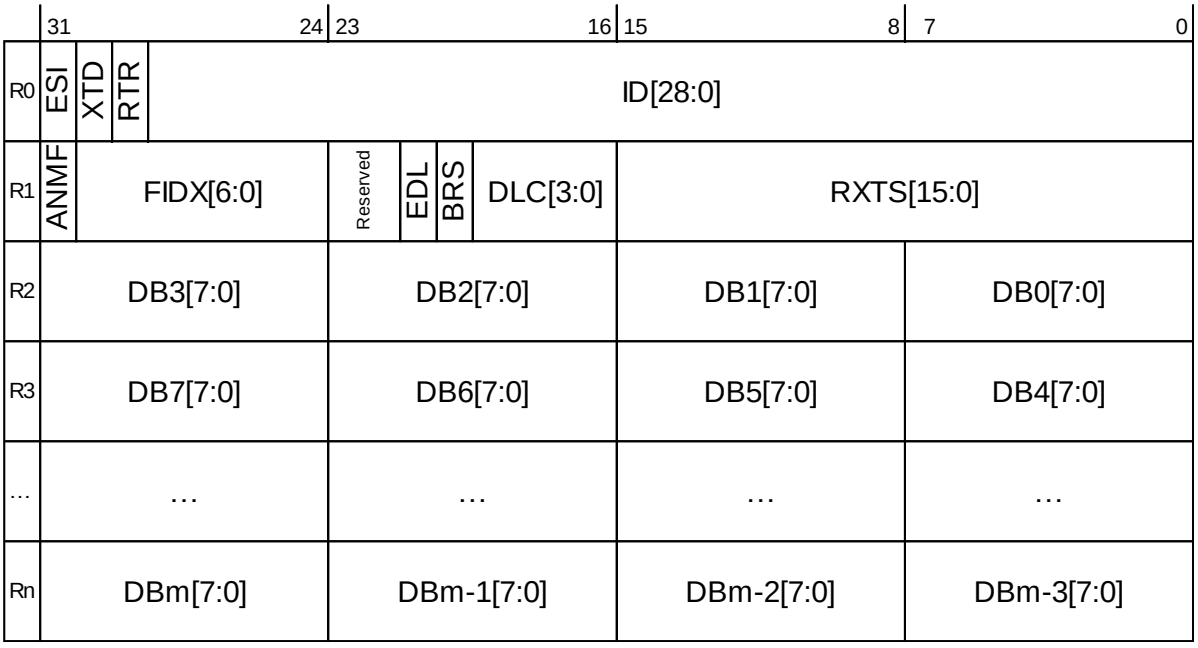
List of Message RAM Elements

Abbreviation	Reference
Rx Buffer and FIFO Element	6.2
Tx Buffer Element	6.3
Tx Event FIFO Element	6.4
Standard Message ID Filter Element	6.5
Extended Message ID Filter Element	6.6

6.2. Rx Buffer and FIFO Element

An Rx Buffer and FIFO Element is a block of 32-bit words that holds the data and status of a received frame that was stored in the Message RAM.

Up to 64 Rx Buffers and two Rx FIFOs can be configured in the Message RAM. Each Rx FIFO section can be configured to store up to 64 received messages. The structure of an Rx Buffer and FIFO element is shown in the figure below. The element size can be configured for storage of CAN FD messages with up to 64 bytes data field via register RXESC (Rx Buffer/FIFO Element Size Configuration).



R0 [bit31] ESI: Error State Indicator

Bit	Description
0	Transmitting node is error active.
1	Transmitting node is error passive.

R0 [bit30] XTD: Extended Identifier

Signals to the CPU whether the received frame has a standard or extended identifier.

Bit	Description
0	11-bit standard identifier.
1	29-bit extended identifier.

R0 [bit29] RTR: Remote Transmission Request

Signals to the CPU whether the received frame is a data frame or a remote frame.

Bit	Description
0	Received frame is a data frame.

1	Received frame is a remote frame.

Note:

- *There are no remote frames in CAN FD format. In case a CAN FD frame was received (EDL = 1), bit RTR reflects the state of the reserved bit r1.*

R0 [bit28:0] ID[28:0]: Identifier

Standard or extended identifier depending on bit XTD. A standard identifier is stored into ID[28:18].

R1 [bit31] ANMF: Accepted Non-matching Frame

Acceptance of non-matching frames may be enabled via GFC.ANFS[1:0] (Accept Non-matching Frames Standard) and GFC.ANFE[1:0] (Accept Non-matching Frames Extended).

Bit	Description
0	Received frame matching filter index FIDX.
1	Received frame did not match any Rx filter element.

R1 [bit30:24] FIDX[6:0]: Filter Index

FIDX[6:0]	Description
0 to 127	Index of matching Rx acceptance filter element (invalid if ANMF = 1). Range is 0 to List Size Standard/Extended minus 1 (i.e. SIDFC.LSS - 1 resp. XIDFC.LSE - 1).

R1 [bit23:22] Reserved: Reserved bits

When writing, always write "0". The read value is undefined.

R1 [bit21] EDL: Extended Data Length

Bit	Description
0	Standard frame format.
1	CAN FD frame format.

R1 [bit20] BRS: Bit Rate Switch

Bit	Description
0	Frame received without bit rate switching.
1	Frame received with bit rate switching.

R1 [bit19:16] DLC[3:0]: Data Length Code

DLC[3:0]	Description
0 to 8	CAN + CAN FD: received frame has 0-8 data bytes.

9 to 15	CAN: received frame has 8 data bytes. CAN FD: received frame has 12/16/20/24/32/48/64 data bytes. See Table 3-1 for details.

R1 [bit15:0] RXTS[15:0]: Rx Timestamp

Timestamp Counter value captured on start of frame reception. Resolution depending on configuration of the Timestamp Counter Prescaler TSCC.TCP[3:0].

R2 [bit31:24]	DB3[7:0]:	Data Byte 3
R2 [bit23:16]	DB2[7:0]:	Data Byte 2
R2 [bit15:8]	DB1[7:0]:	Data Byte 1
R2 [bit7:0]	DB0[7:0]:	Data Byte 0
R3 [bit31:24]	DB7[7:0]:	Data Byte 7
R3 [bit23:16]	DB6[7:0]:	Data Byte 6
R3 [bit15:8]	DB5[7:0]:	Data Byte 5
R3 [bit7:0]	DB4[7:0]:	Data Byte 4
…	…	…
Rn [bit31:24]	DBm[7:0]:	Data Byte m
Rn [bit23:16]	DBm-1[7:0]:	Data Byte m-1
Rn [bit15:8]	DBm-2[7:0]:	Data Byte m-2
Rn [bit7:0]	DBm-3[7:0]:	Data Byte m-3

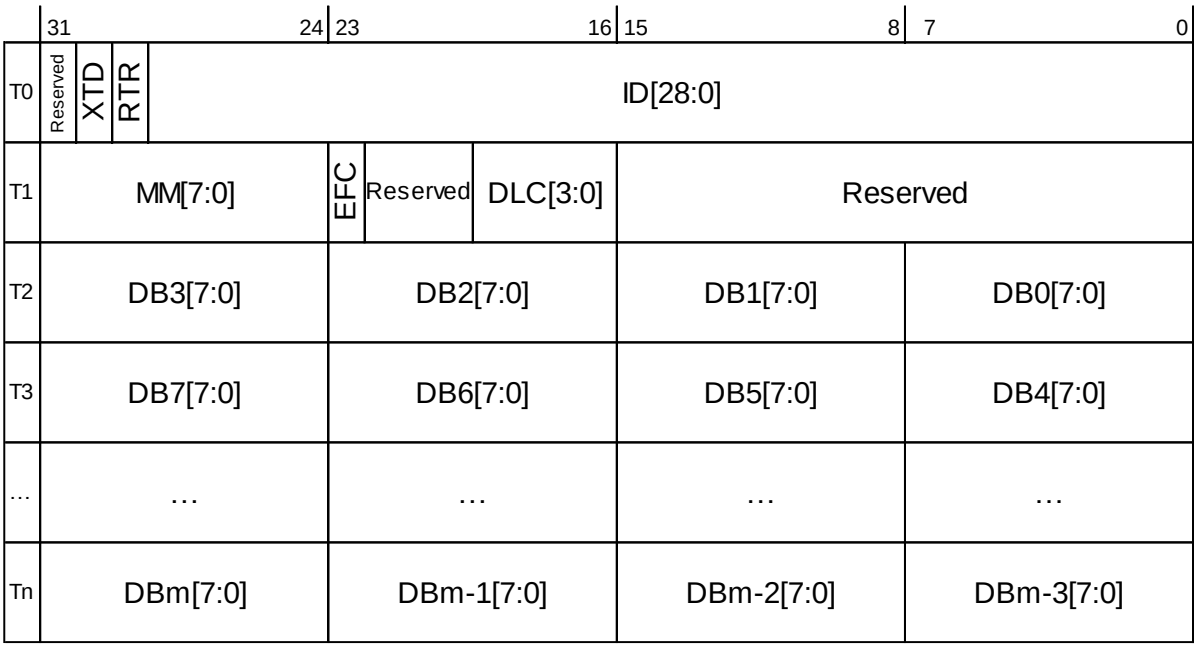
Notes:

- *Depending on the configuration of the element size (defined by Rx Buffer/FIFO Element Size Configuration (RXESC)), Rn will vary from n = 3 to 17.*
- *m is a function of n, m = (n - 1) × 4 - 1.*
- *The number of valid data bytes are defined by the Data Length Code.*

6.3. Tx Buffer Element

A Tx Buffer Element is a block of 32-bit words stored in the Message RAM that holds data and control information of a frame to be transmitted by the CAN FD Controller.

The Tx Buffers section can be configured to hold dedicated Tx Buffers as well as a Tx FIFO/Tx Queue. In case that the Tx Buffers section is shared by dedicated Tx buffers and a Tx FIFO/Tx Queue, the dedicated Tx Buffers start at the beginning of the Tx Buffers section followed by the buffers assigned to the Tx FIFO or Tx Queue. The Tx Handler distinguishes between dedicated Tx Buffers and Tx FIFO/Tx Queue by evaluating the Tx Buffer configuration TXBC.TFQS[5:0] (Transmit FIFO/Queue Size) and TXBC.NDTB[5:0] (Number of Dedicated Transmit Buffers). The element size can be configured for storage of CAN FD messages with up to 64 bytes data field via register TXESC (Tx Buffer Element Size Configuration).



T0 [bit31] Reserved: Reserved bit

When writing, always write "0". The read value is undefined.

T0 [bit30] XTD: Extended Identifier

Bit	Description
0	11-bit standard identifier.
1	29-bit extended identifier.

T0 [bit29] RTR: Remote Transmission Request

Bit	Description
0	Transmit data frame.
1	Transmit remote frame.

Note:

- *When RTR = 1, the CAN FD Controller transmits a remote frame according to ISO11898-1, even if the CAN Mode Enable field (CCCR.CME[1:0]) enables the transmission in CAN FD format.*

T0 [bit28:0] ID[28:0]: Identifier

Standard or extended identifier depending on bit XTD. A standard identifier has to be written to ID[28:18].

T1 [bit31:24] MM[7:0]: Message Marker

Written by CPU during Tx Buffer configuration. Copied into Tx Event FIFO element for identification of Tx message status.

T1 [bit23] EFC: Event FIFO Control

Bit	Description
0	Don’t store Tx events.
1	Store Tx events.

T1 [bit22:20] Reserved: Reserved bits

When writing, always write "0". The read value is undefined.

T1 [bit19:16] DLC[3:0]: Data Length Code

DLC[3:0]	Description
0 to 8	CAN + CAN FD: transmit frame has 0-8 data bytes.
9 to 15	CAN: transmit frame has 8 data bytes. CAN FD: transmit frame has 12/16/20/24/32/48/64 data bytes. See Table 3-1 for details.

T1 [bit15:0] Reserved: Reserved bits

When writing, always write "0". The read value is undefined.

T2 [bit31:24]	DB3[7:0]:	Data Byte 3
T2 [bit23:16]	DB2[7:0]:	Data Byte 2
T2 [bit15:8]	DB1[7:0]:	Data Byte 1
T2 [bit7:0]	DB0[7:0]:	Data Byte 0
T3 [bit31:24]	DB7[7:0]:	Data Byte 7
T3 [bit23:16]	DB6[7:0]:	Data Byte 6
T3 [bit15:8]	DB5[7:0]:	Data Byte 5
T3 [bit7:0]	DB4[7:0]:	Data Byte 4

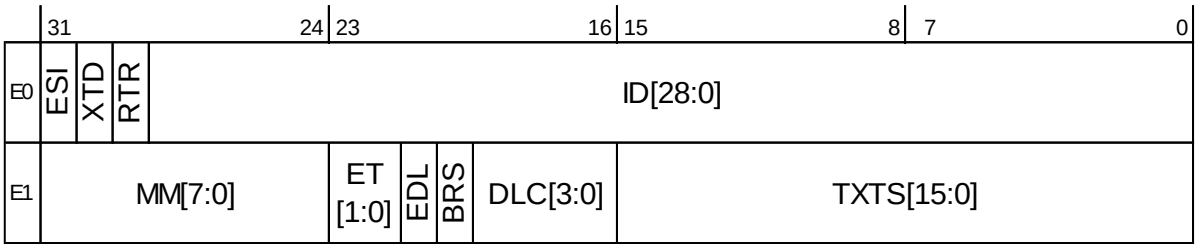
…	…	…
Tn [bit31:24]	DBm[7:0]:	Data Byte m
Tn [bit23:16]	DBm-1[7:0]:	Data Byte m-1
Tn [bit15:8]	DBm-2[7:0]:	Data Byte m-2
Tn [bit7:0]	DBm-3[7:0]:	Data Byte m-3

Notes:

- *Depending on the configuration of the element size (TXESC), Tn will vary from n = 3 to 17.*
- *m is a function of n: m = (n - 1) × 4 - 1.*

6.4. Tx Event FIFO Element

Each Tx Event FIFO Element stores information about transmitted messages. By reading the Tx Event FIFO the CPU gets this information in the order the messages were transmitted. Status information about the Tx Event FIFO can be obtained from register TXEFS (Tx Event FIFO Status).



E0 [bit31] ESI: Error State Indicator

Bit	Description
0	Transmitting node is error active.
1	Transmitting node is error passive.

E0 [bit30] XTD: Extended Identifier

Bit	Description
0	11-bit standard identifier.
1	29-bit extended identifier.

E0 [bit29] RTR: Remote Transmission Request

Bit	Description
0	Data frame transmitted.
1	Remote frame transmitted.

E0 [bit28:0] ID[28:0]: Identifier

Standard or extended identifier depending on bit XTD. A standard identifier is stored into ID[28:18].

E1 [bit31:24] MM[7:0]: Message Marker

Copied from Tx Buffer into Tx Event FIFO element for identification of Tx message status.

E1 [bit23:22] ET[1:0]: Event Type

Bits	Description
00	Reserved.
01	Tx event.
10	Transmission in spite of cancellation.
11	Reserved.

E1 [bit21] EDL: Extended Data Length

Bit	Description
0	Standard frame format.
1	CAN FD frame format (new DLC-coding and CRC).

E1 [bit20] BRS: Bit Rate Switch

Bit	Description
0	Frame transmitted without bit rate switching.
1	Frame transmitted with bit rate switching.

E1 [bit19:16] DLC[3:0]: Data Length Code

DLC[3:0]	Description
0 to 8	CAN + CAN FD: frame with 0-8 data bytes transmitted.
9 to 15	CAN: frame with 8 data bytes transmitted. CAN FD: frame with 12/16/20/24/32/48/64 data bytes transmitted. See Table 3-1 for details.

E1 [bit15:0] TXTS[15:0]: Tx Timestamp

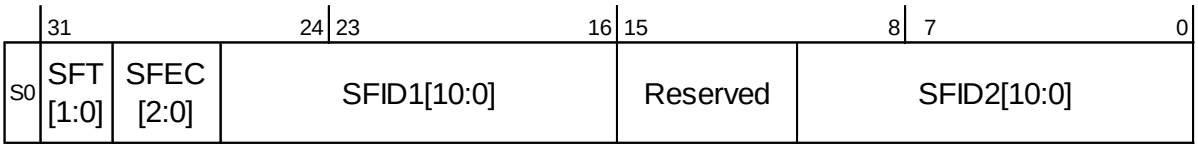
Timestamp Counter value captured on start of frame transmission. Resolution depending on configuration of the Timestamp Counter Prescaler (TSCC.TCP[3:0]).

6.5. Standard Message ID Filter Element

A Standard Message ID Filter Element consists of a single 32-bit word, and can be configured as a range filter, dual filter, classic bit mask filter, or filter for a single dedicated ID, for messages with 11-bit standard IDs.

Up to 128 filter elements can be configured for 11-bit standard IDs. When accessing a Standard Message ID Filter element, its address is

Filter List Standard Start Address (SIDFC.FLSSA[15:2]) + index of the filter element (0 to 127).



S0 [bit31:30] SFT[1:0]: Standard Filter Type

Bits	Description
00	Range filter from SFID1[10:0] to SFID2[10:0] (SFID2[10:0] ≥ received ID ≥ SFID1[10:0]).
01	Dual ID filter for SFID1[10:0] or SFID2[10:0].
10	Classic filter: SFID1[10:0] = filter, SFID2[10:0] = mask. Only those bits of SFID1[10:0] where the corresponding SFID2[10:0] bits are "1" are relevant.
11	Reserved.

Note:

- *The setting SFT[1:0] = 0b11 is reserved and should not be used. The setting SFT[1:0] = 0b11 will disable the filter element, but use SFEC[2:0] = 0b000 instead.*

S0 [bit29:27] SFEC[2:0]: Standard Filter Element Configuration

All enabled filter elements are used for acceptance filtering of standard frames. Acceptance filtering stops at the first matching enabled filter element or when the end of the filter list is reached.

If SFEC[2:0] = 0b100, 0b101, or 0b110 a match sets interrupt flag IR.HPM (High Priority Message) and, if enabled, an interrupt is generated. In this case register HPMS (High Priority Message Status) is updated with the status of the priority match.

Bits	Description
000	Disable filter element.
001	Store in Rx FIFO 0 if filter matches.
010	Store in Rx FIFO 1 if filter matches.
011	Reject ID if filter matches.
100	Set priority if filter matches.
101	Set priority and store in Rx FIFO 0 if filter matches.
110	Set priority and store in Rx FIFO 1 if filter matches.
111	Store into dedicated Rx Buffer or as debug message, configuration of SFT[1:0] ignored.

S0 [bit26:16] SFID1[10:0]: Standard Filter ID 1

This bit field has a different meaning depending on the configuration of SFEC[2:0]:

- SFEC[2:0] = 0b001 to 0b110

  Set SFID1[10:0] according to the SFT[1:0] setting.

- SFEC[2:0] = 0b111

  SFID1[10:0] defines the ID of a standard dedicated Rx Buffer or debug message to be stored. The received identifiers must match exactly, no masking mechanism is used.

S0 [bit15:11] Reserved: Reserved bits

When writing, always write "0". The read value is undefined.

S0 [bit10:0] SFID2[10:0]: Standard Filter ID 2

This bit field has a different meaning depending on the configuration of SFEC[2:0]:

- SFEC[2:0] = 0b001 to 0b110

  Set SFID2[10:0] according to the SFT[1:0] setting

- SFEC[2:0] = 0b111

  Filter for dedicated Rx Buffers or for debug messages

  SFID2[10:9] decides whether the received message is stored into a dedicated Rx Buffer or treated as message A, B, or C of the debug message sequence.

SFID2[10:9]	Description
00	Store message into a dedicated Rx Buffer.
01	Debug Message A.
10	Debug Message B.
11	Debug Message C.

  SFID2[8:6] are reserved bits. When writing, always write "0". The read value is undefined.

  SFID2[5:0] defines the offset to the Rx Buffer Start Address RXBC.RBSA[15:2] for storage of a matching message.

Notes:

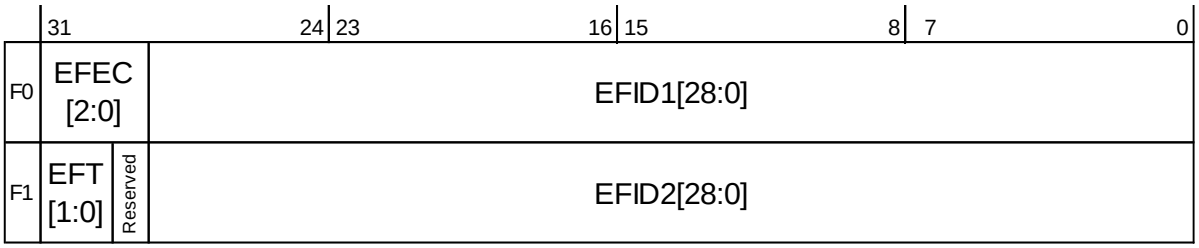
- *Debug message is used for Debug on CAN feature.*
- *Debug on CAN feature is not supported for TRAVEO™ T1G Platform.*

6.6. Extended Message ID Filter Element

An Extended Message ID Filter Element consists of two 32-bit words, and can be configured as a range filter, dual filter, classic bit mask filter, or filter for a single dedicated ID, for messages with 29-bit extended IDs.

Up to 64 filter elements can be configured for 29-bit extended IDs. When accessing an Extended Message ID Filter element, its address is

Filter List Extended Start Address (XIDFC.FLESA[15:2]) + 2 × index of the filter element (0 to 63).



F0 [bit31:29] EFEC[2:0]: Extended Filter Element Configuration

All enabled filter elements are used for acceptance filtering of extended frames. Acceptance filtering stops at the first matching enabled filter element or when the end of the filter list is reached.

If EFEC[2:0] = 0b100, 0b101, or 0b110 a match sets interrupt flag IR.HPM (High Priority Message) and, if enabled, an interrupt is generated. In this case register HPMS (High Priority Message Status) is updated with the status of the priority match.

Bits	Description
000	Disable filter element.
001	Store in Rx FIFO 0 if filter matches.
010	Store in Rx FIFO 1 if filter matches.
011	Reject ID if filter matches.
100	Set priority if filter matches.
101	Set priority and store in Rx FIFO 0 if filter matches.
110	Set priority and store in Rx FIFO 1 if filter matches.
111	Store into dedicated Rx Buffer or as debug message, configuration of EFT[1:0] ignored.

F0 [bit28:0] EFID1[28:0]: Extended Filter ID 1

This bit field has a different meaning depending on the configuration of EFEC[2:0].

- EFEC[2:0] = 0b001 to 0b110

  Set EFID1[28:0] according to the EFT[1:0] setting.

- EFEC[2:0] = 0b111

  EFID1[28:0] defines the ID of an extended dedicated Rx Buffer or debug message to be stored. The received identifiers must match exactly, only XIDAM masking mechanism (see "3.4.1.3. Extended Message ID Filtering") is used.

F1 [bit31:30] EFT[1:0]: Extended Filter Type

Bits	Description
00	Range filter from EFID1[28:0] to EFID2[28:0] (EFID2[28:0] ≥ received ID ANDed with XIDAM ≥ EFID1[28:0]).
01	Dual ID filter Matches when EFID1[28:0] or EFID2[28:0] is equal to received ID ANDed with XIDAM.
10	Classic filter: EFID1[28:0] = filter, EFID2[28:0] = mask. Only those bits of EFID1[28:0] where the corresponding EFID2[28:0] bits are "1" are relevant. Matches when the received ID ANDed with XIDAM is equal to EFID1[28:0] masked by EFID2[28:0].
11	Range filter from EFID1[28:0] to EFID2[28:0] (EFID2[28:0] ≥ EFID1[28:0]), XIDAM mask not applied.

F1 [bit29] Reserved: Reserved bit

When writing, always write "0". The read value is undefined.

F1 [bit28:0] EFID2[28:0]: Extended Filter ID 2

This bit field has a different meaning depending on the configuration of EFEC[2:0]:

- EFEC[2:0] = 0b001 to 0b110

  Set EFID2[28:0] according to the EFT[1:0] setting

- EFEC[2:0] = 0b111

  EFID2[28:0] is used to configure this filter for dedicated Rx Buffers or for debug messages

  EFID2[28:11] are reserved bits. When writing, always write "0". The read value is undefined.

  EFID2[10:9] decides whether the received message is stored into a dedicated Rx Buffer or treated as message A, B, or C of the debug message sequence.

Bits	Description
00	Store message into a dedicated Rx Buffer.
01	Debug Message A.
10	Debug Message B.
11	Debug Message C.

  EFID2[8:6] are reserved bits. When writing, always write "0". The read value is undefined.

  EFID2[5:0] defines the offset to the Rx Buffer Start Address RXBC.RBSA[15:2] for storage of a matching message.

Notes:

- *Debug message is used for Debug on CAN feature.*
- *Debug on CAN feature is not supported for TRAVEO™ T1G Platform.*

CHAPTER 31: CAN FD Controller(MCAN3.2)

This chapter explains the functions and operations of the CAN FD Controller.

1. Overview
2. Configuration
3. Explanation of Operations
4. Setup Procedure Examples
5. Registers
6. Message RAM

1. Overview

The CAN FD Controller performs communication according to ISO11898-1 (CAN specification Rev. 2.0 part A, B) and to the CAN FD specification V1.0.

All functions concerning the handling of messages are implemented by the Rx Handler and the Tx Handler. The Rx Handler manages message acceptance filtering, the transfer of received messages from the CAN Core to a Message RAM as well as providing receive message status information. The Tx Handler is responsible for the transfer of transmit messages from the Message RAM to the CAN Core as well as providing transmit status information.

Acceptance filtering is implemented by a combination of up to 192 filter elements where each one can be configured as a range, as a bit mask, or as a dedicated ID filter.

The CAN FD Controller is accessible by the CPU using a data width of 8/16/32-bits. The CAN FD Controller's clock domain concept allows the separation between the two input clocks, the CAN clock and the Bus clock.

Features of the CAN FD Controller

- Conform with CAN protocol version 2.0 part A, B and ISO 11898-1
- CAN FD with up to 64 data bytes supported
- CAN Error Logging
- Improved acceptance filtering
- Two configurable Receive FIFOs
- Separate signalling on reception of High Priority Messages
- Up to 64 dedicated Receive Buffers
- Up to 32 dedicated Transmit Buffers
- Configurable Transmit FIFO
- Configurable Transmit Queue
- Configurable Transmit Event FIFO
- Direct Message RAM access for CPU
- Programmable loop-back test mode
- Maskable interrupts
- Two clock domains (CAN clock and Bus clock)
- Power-down support
- Debug on CAN support

Note:

- *Debug on CAN feature is not supported for TRAVEO™ T1G Platform.*

Terms and Abbreviations

This document uses the following terms and abbreviations.

Term	Meaning
mtq	minimum time quantum = CAN clock period (canfd_cclk)
tq	time quantum
SSP	Secondary Sample Point
TSEG1	Time Segment before Sample Point
TSEG2	Time Segment after Sample Point

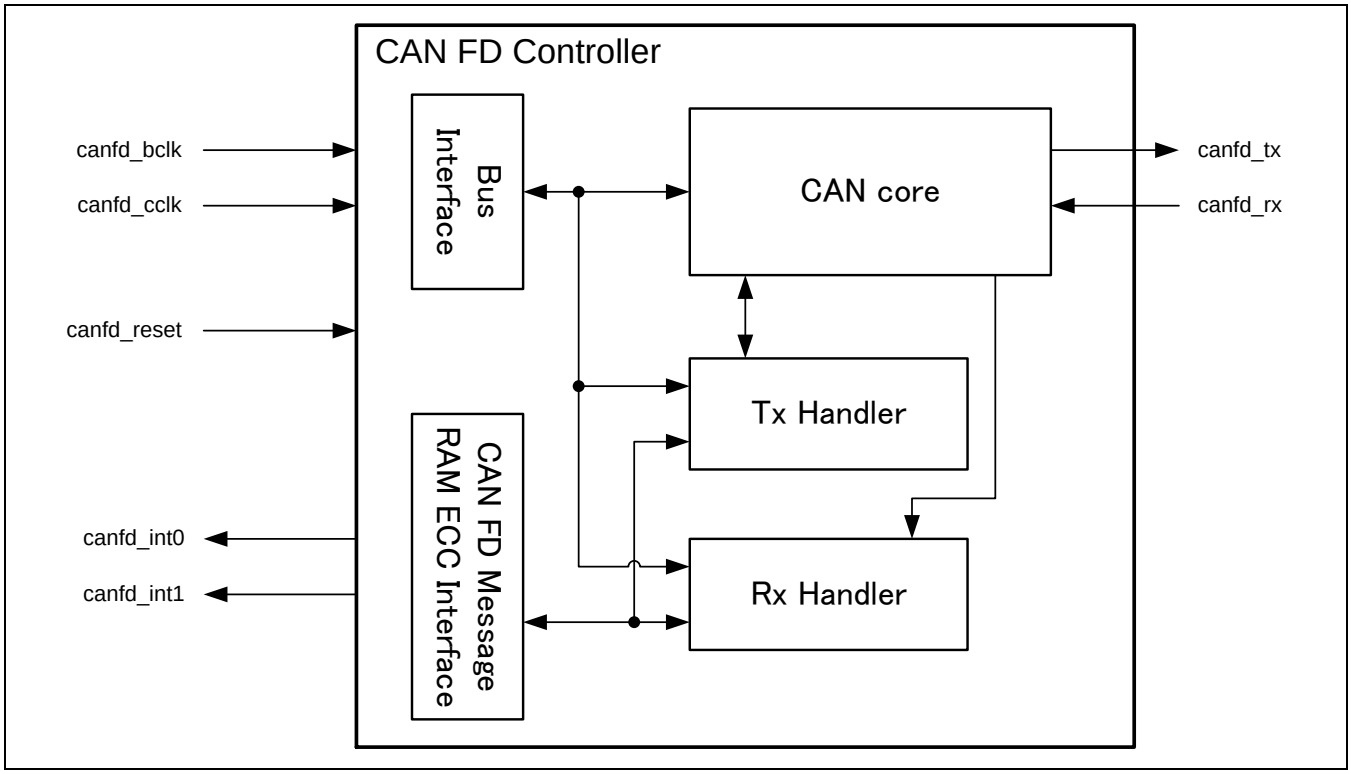
2. Configuration

This section provides configuration related information about the CAN FD controller.

2.1. Block Diagram
2.2. Dual Clock Sources
2.3. Dual Interrupt Lines

2.1. Block Diagram

Figure 2-1 CAN FD Controller Block Diagram



CAN Core

CAN Protocol Controller. Handles all ISO 11898-1 protocol functions. Supports 11-bit and 29-bit identifiers.

CAN FD Message RAM ECC Interface

Interface with the CAN FD Message RAM ECC module. This module contains

- a Message RAM that stores messages to be transmitted, received messages, acceptance filter elements, and Tx Events, and
- ECC (Error Correction Code) logic for the Message RAM.

Bus Interface

CPU interface bus with a data width of 8/16/32-bits.

Tx Handler

Controls the message transfer from the Message RAM to the CAN Core. A maximum of 32 Tx Buffers can be configured for transmission. Tx buffers can be used as dedicated Tx Buffers, as Tx FIFO, part of a Tx Queue, or as a combination of them. A Tx Event FIFO stores Tx timestamps together with the corresponding Message ID. Transmit cancellation is also supported.

Rx Handler

Controls the transfer of received messages from the CAN Core to the Message RAM. The Rx Handler supports two Receive FIFOs, each of configurable size, and up to 64 dedicated Rx Buffers for storage of all messages that have passed acceptance filtering. A dedicated Rx Buffer, in contrast to a Receive FIFO, is used to store only messages with a specific identifier. An Rx timestamp is stored together with each message. Up to 128 filters can be defined for 11-bit IDs and up to 64 filters for 29-bit IDs.

2.2. Dual Clock Sources

The two clocks of the CAN FD Controller must be configured to meet the following requirement in order to guarantee proper operation:

Bus clock (canfd_bclk) frequency > CAN clock (canfd_cclk) frequency

Note:

- *Message RAM Access Failure (see "5.15 Interrupt Register (MCG_CANFDx_IR, CPG_CANFDx_IR)") may be generated depending on between Bus clock (canfd_bclk) and the bit rate of CAN (or CAN FD). Therefore we show the examples of Bus clock frequency to the following tables for preventing Message RAM Access Failure. Bus clock frequency must be input more than the frequency shown in the following tables.*

Table 2-1 Required Frequencies [min] for CAN

	Bit Rate		
	250kbps	**500kbps**	**1Mbps**
Required Frequency (Bus Clock: canfd_bclk)	5.1MHz	10.1MHz	20.3MHz

Table 2-2 Required Frequencies [min] for CAN FD

	Bit Rate					
	250kbps (nominal) 1Mbps (FD)	250kbps (nominal) 2Mbps (FD)	500kbps (nominal) 2Mbps (FD)	500kbps (nominal) 4Mbps (FD)	500kbps (nominal) 5Mbps (FD)	1Mbps (nominal) 2Mbps (FD)
Required Frequency (Bus Clock: canfd_bclk)	9.4MHz	12.0MHz	18.8MHz	23.9MHz	25.3MHz	26.3MHz

2.3. Dual Interrupt Lines

The CAN FD Controller provides two interrupt lines. Interrupts can be routed either to canfd_int0 or to canfd_int1. By default all interrupts are routed to interrupt line canfd_int0. By programming bits Enable Interrupt Line 0 (ILE.EINT0) and Enable Interrupt Line 1 (ILE.EINT1) the interrupt lines can be enabled or disabled separately.

3. Explanation of Operations

This section explains the operations of the CAN FD Controller.

3.1.Operating Modes

3.2.Timestamp Generation

3.3.Timeout Counter

3.4.Rx Handling

3.5.Tx Handling

3.6.FIFO Acknowledge Handling

3.7. Configuring the CAN Bit Timing

3.1. Operating Modes

This section explains the operating modes of the CAN FD Controller.

3.1.1.Software Initialization

3.1.2.Normal Operation

3.1.3.CAN FD Operation

3.1.4.Transmitter Delay Compensation

3.1.5.Restricted Operation Mode

3.1.6.Bus Monitoring Mode

3.1.7.Disabled Automatic Retransmission

3.1.8.Power Down (Sleep Mode)

3.1.9.Test Modes

3.1.1. Software Initialization

Setting/Resetting the Initialization Bit (CCCR.INIT)

Software initialization is started by setting bit Initialization (CCCR.INIT),

- either by software or by a hardware reset,
- when an uncorrected bit error was detected in the Message RAM,
- or by going Bus_Off.

While CCCR.INIT is set,

- message transfer from and to the CAN bus is stopped,
- the status of the CAN bus output canfd_tx is recessive,
- the protocol error counters are unchanged.

Setting CCCR.INIT does not change any configuration register.

Resetting CCCR.INIT finishes the software initialization. Afterwards the CAN FD Controller synchronizes itself to the data transfer on the CAN bus by waiting for the occurrence of a sequence of 11 consecutive recessive bits (≡ Bus_Idle) before it can take part in bus activities and start the message transfer.

Access/Set/Reset Properties of Registers Affected by Configuration Change Enable (CCCR.CCE)

Access to the configuration registers is only enabled when both bits CCCR.INIT and Configuration Change Enable (CCCR.CCE) are set (protected write). CCCR.CCE can only be set/reset while CCCR.INIT = "1". CCCR.CCE is automatically reset when CCCR.INIT is reset.

The following registers are reset when CCCR.CCE is set

- HPMS - High Priority Message Status
- RXF0S - Rx FIFO 0 Status
- RXF1S - Rx FIFO 1 Status
- TXFQS - Tx FIFO/Queue Status
- TXBRP - Tx Buffer Request Pending
- TXBTO - Tx Buffer Transmission Occurred
- TXBCF - Tx Buffer Cancellation Finished
- TXEFS - Tx Event FIFO Status

In addition

- the Timeout Counter value (TOCV.TOC[15:0]) is preset to the value configured by the Timeout Period (TOCC.TOP[15:0]) when CCCR.CCE is set.
- the state machines of the Tx Handler and Rx Handler are held in idle state while CCCR.CCE = "1".

The following registers are only writeable while CCCR.CCE = "0".

- TXBAR - Tx Buffer Add Request
- TXBCR - Tx Buffer Cancellation Request

Test Mode Enable (CCCR.TEST) and Bus Monitoring Mode (CCCR.MON) can only be set by the CPU while CCCR.INIT = "1" and CCCR.CCE = "1". Both bits may be reset at any time.

Disable Automatic Retransmission (CCCR.DAR) can only be set/reset while CCCR.INIT = "1" and CCCR.CCE = "1".

Message RAM Initialization

The Message RAM should be zeroized before configuration of the CAN FD Controller in order to prevent Message RAM bit errors when reading uninitialized words, and also to avoid unexpected filter element configurations in the Message RAM.

3.1.2. Normal Operation

Once the CAN FD Controller is initialized and Initialization (CCCR.INIT) is reset to "0", the CAN FD Controller synchronizes itself to the CAN bus and is ready for communication.

After passing the acceptance filtering, received messages including Message ID and DLC (Data Length Code) are stored into a dedicated Rx Buffer or into Rx FIFO 0 or Rx FIFO 1.

For messages to be transmitted, dedicated Tx Buffers and/or a Tx FIFO or a Tx Queue can be initialized or updated. Automated transmission on reception of remote frames is not implemented.

3.1.3. CAN FD Operation

There are two variants in the CAN FD frame format,

- the CAN FD frame without bit rate switching.
- the CAN FD frame where control field, data field, and CRC field are transmitted with a higher bit rate than the beginning and the end of the frame.

Interpretation of Received Frames

The previously reserved bit in CAN frames with 11-bit identifiers and the first previously reserved bit in CAN frames with 29-bit identifiers will now be decoded as FDF bit.

- FDF = recessive signifies a CAN FD frame,
- FDF = dominant signifies a Classic CAN frame.

In a CAN FD frame, the two bits following FDF, res and BRS, decide whether the bit rate inside of this CAN FD frame is switched. A CAN FD bit rate switch is signified by res = dominant and BRS = recessive.

The coding of res = recessive is reserved for future expansion of the protocol. In case receiving a frame with FDF = recessive and res = recessive,

- With Protocol Exception Handling enabled (CCCR.PXHD = "0"), it will signal a Protocol Exception Event by setting PSR.PXE. This causes the operation state to change from Receiver (PSR.ACT = "10") to Integrating (PSR.ACT = "00") at the next sample point.
- With Protocol Exception Handling disabled (CCCR.PXHD = "1"), it will treat a recessive res bit as a form error and will respond with an error frame.

Enabling CAN FD Operation via FD Operation Enable (CCCR.FDOE)

CAN FD operation is enabled by programming FD Operation Enable (CCCR.FDOE). In case CCCR.FDOE = "1", transmission and reception of CAN FD frames is enabled. Transmission and reception of Classic CAN frames is always possible. Whether a CAN FD frame or a Classic CAN frame is transmitted can be configured via bit FDF in the respective Tx Buffer element. With CCCR.FDOE = "0", received frames are interpreted as Classic CAN frames, which leads to the transmission of an error frame when receiving a CAN FD frame. When CAN FD operation is disabled, no CAN FD frames are transmitted even if bit FDF of a Tx Buffer element is set. The transmission properties of CAN FD operation can be changed via FD Operation Enable and Bit Rate Switch Enable (CCCR.FDOE and CCCR.BRSE. See next paragraph for details).

- With CCCR.FDOE = "0", the setting of bits FDF and BRS is ignored and frames are transmitted in Classic CAN format.
- With CCCR.FDOE = "1" and CCCR.BRSE = "0", only bit FDF of a Tx Buffer element is evaluated.
- With CCCR.FDOE = "1" and CCCR.BRSE = "1", transmission of CAN FD frames with bit rate switching is enabled. All Tx Buffer elements with bits FDF and BRS set are transmitted in CAN FD format with bit rate switching.

CCCR.FDOE and CCCR.BRSE can only be changed while CCCR.INIT and CCCR.CCE are both set.

Changing a Mode during CAN Operation

A mode change during CAN operation is only recommended under the following conditions:

- The failure rate in the CAN FD data phase is significantly higher than in the CAN FD arbitration phase. In this case disable the CAN FD bit rate switching option for transmissions.
- During system startup all nodes are transmitting Classic CAN messages until it is verified that they are able to communicate in CAN FD format. If this is true, all nodes switch to CAN FD operation.
- Wake-up messages in CAN Partial Networking have to be transmitted in Classic CAN format.
- End-of-line programming in case not all nodes are CAN FD capable. Non CAN FD nodes are held in Bus Monitoring mode until programming has completed. Then all nodes switch back to Classic CAN communication.

The DLC Field

In the CAN FD format, the coding of the DLC differs from the Classic CAN format. The DLC codes 0 to 8 have the same coding as in Classic CAN, the codes 9 to 15, which in Classic CAN all code a data field of 8 bytes, are coded according to Table 3-1 below.

Table 3-1 Coding of DLC in CAN FD

DLC	9	10	11	12	13	14	15
Number of Data Bytes	12	16	20	24	32	48	64

Bit Rate Switching

In CAN FD frames, the bit timing will be switched inside the frame, after the BRS (Bit Rate Switch) bit, if this bit is recessive. Before the BRS bit, in the CAN FD arbitration phase, the nominal CAN bit timing is used as defined by the Nominal Bit Timing & Prescaler Register (NBTP). In the following CAN FD data phase, the data phase bit timing is used as defined by the Data Bit Timing & Prescaler Register (DBTP). The bit timing is switched back from the data phase timing at the CRC delimiter or when an error is detected, whichever occurs first.

The maximum configurable bit rate in the CAN FD data phase depends on the CAN clock frequency (canfd_cclk).

Example:

- *With a CAN clock frequency of 20MHz and the shortest configurable bit time of 4 tq, the bit rate in the data phase is 5 Mbit/s.*

The Error Status Indicator ESI

In both data frame formats, CAN FD and CAN FD with bit rate switching, the value of the bit ESI (Error Status Indicator) is determined by the transmitter's error state at the start of the transmission. If the transmitter is error passive, ESI is transmitted recessive, else it is transmitted dominant.

3.1.4. Transmitter Delay Compensation

During the data phase of a CAN FD transmission only one node is transmitting, all others are receivers. The length of the bus line has no impact. When transmitting via pin canfd_tx the protocol controller receives the transmitted data from its local CAN transceiver via pin canfd_rx. The received data is delayed by the transmitter delay. In case this delay is greater than the time segment before the sample point of a data phase bit time, a bit error is detected. In order to enable a data phase bit time that is even shorter than the transmitter delay, the delay compensation is introduced. Without transmitter delay compensation, the bit rate in the data phase of a CAN FD frame is limited by the transmitter delay.

3.1.4.1 Description

The CAN FD protocol unit has implemented a delay compensation mechanism to compensate the transmitter delay, thereby enabling transmission with higher bit rates during the CAN FD data phase independent of the delay of a specific CAN transceiver.

To check for bit errors during the data phase of transmitting nodes, the delayed transmit data is compared against the received data at the Secondary Sample Point SSP. If a bit error is detected, the transmitter will react on this bit error at the next following regular sample point. During arbitration phase the delay compensation is always disabled.

The transmitter delay compensation enables configurations where the data bit time is shorter than the transmitter delay, it is described in detail in the new ISO11898-1. It is enabled by setting bit Transmitter Delay Compensation (DBTP.TDC).

The received bit is compared against the transmitted bit at the SSP. The SSP position is defined as the sum of the measured delay from the transmit output canfd_tx through the transceiver to the receive input canfd_rx plus the transmitter delay compensation offset as configured by Transmitter Delay Compensation Offset (TDCR.TDCO[6:0]). The transmitter delay compensation offset is used to adjust the position of the SSP inside the received bit (e.g. half of the bit time in the data phase). The position of the secondary sample point is rounded down to the next integer number of mtq (canfd_cclk period).

Transmitter Delay Compensation Value (PSR.TDCV[6:0]) shows the actual transmitter delay compensation value. PSR.TDCV[6:0] is cleared when CCCR.INIT is set and is updated at each transmission of an FD frame while DBTP.TDC is set.

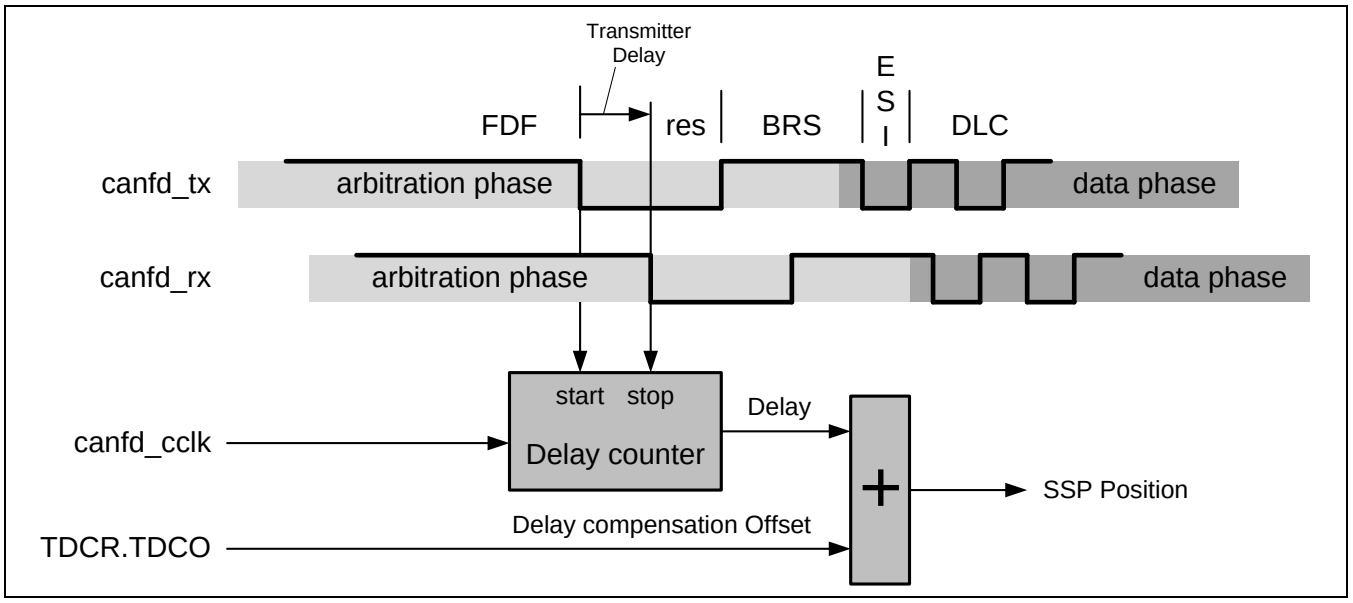
The following boundary conditions have to be considered for the transmitter delay compensation implemented in the CAN FD Controller:

- The sum of the measured delay from canfd_tx to can_fd_rx and the configured transmitter delay compensation offset TDCR.TDCO[6:0] has to be less than 6 bit times in the data phase.
- The sum of the measured delay from canfd_tx to canfd_rx and the configured transmitter delay compensation offset TDCR.TDCO[6:0] has to be less or equal 127 mtq. In case this sum exceeds 127 mtq, the maximum value of 127 mtq is used for transmitter delay compensation.
- The data phase ends at the sample point of the CRC delimiter, that stops checking of receive bits at the SSPs.

3.1.4.2 Transmitter Delay Compensation Measurement

Figure 3-1 below describes how the transmitter delay is measured.

Figure 3-1 Transmitter Delay Measurement



If transmitter delay compensation is enabled by programming Transmitter Delay Compensation (DBTP.TDC) = "1", the measurement is started within each transmitted CAN FD frame at the falling edge of bit FDF to bit res. The measurement is stopped when this edge is seen at the receive input canfd_rx of the transmitter. The resolution of this measurement is one mtq.

To avoid that a dominant glitch inside the received FDF bit ends the delay compensation measurement before the falling edge of the received res bit, resulting in a to early SSP position, the use of a transmitter delay compensation filter window can be enabled by programming Transmitter Delay Compensation Filter Window Length (TDCR.TDCF[6:0]). This defines a minimum value for the SSP position. Dominant edges on canfd_rx, that would result in an earlier SSP position are ignored for transmitter delay measurement. The measurement is stopped when the SSP position is at least TDCR.TDCF[6:0] AND canfd_rx is low.

3.1.5. Restricted Operation Mode

In Restricted Operation Mode the node is able to receive data and remote frames and to give acknowledge to valid frames, but it does not send data frames, remote frames, active error frames, or overload frames. In case of an error condition or overload condition, it does not send dominant bits, instead it waits for the occurrence of bus idle condition to resynchronize itself to the CAN communication. The error counters (ECR.REC[6:0], ECR.TEC[7:0]) are frozen while Error Logging (ECR.CEL) is active.

The CPU can set the CAN FD Controller into Restricted Operation Mode by setting the Restricted Operation Mode bit (CCCR.ASM). CCCR.ASM can only be set by the CPU when both Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) are set to "1". CCCR.ASM can be reset by the CPU at any time.

Restricted Operation Mode is automatically entered when the Tx Handler was not able to read data from the Message RAM in time. To leave Restricted Operation Mode, the CPU has to reset CCCR.ASM.

The Restricted Operation Mode can be used in applications that adapt themselves to different CAN bit rates. In this case the application tests different bit rates and leaves the Restricted Operation Mode after it has received a valid frame.

Note:

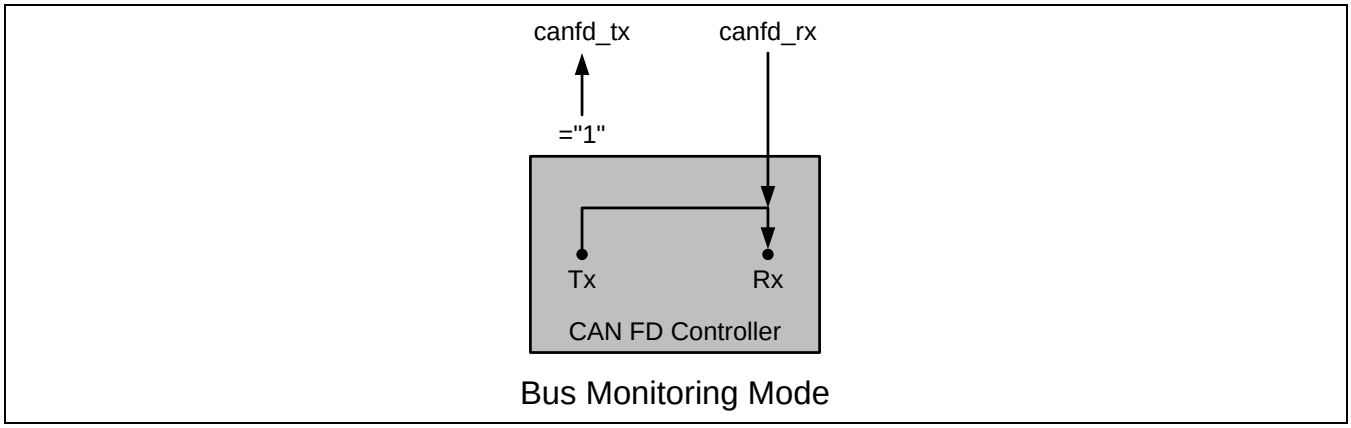
- *The Restricted Operation Mode must not be combined with the Loop Back Mode (internal or external).*

3.1.6. Bus Monitoring Mode

The CAN FD Controller is set in Bus Monitoring Mode by programming the Bus Monitoring Mode bit (CCCR.MON) to one. In Bus Monitoring Mode (see ISO11898-1, 10.12 Bus monitoring), the CAN FD Controller is able to receive valid data frames and valid remote frames, but cannot start a transmission. In this mode, it sends only recessive bits on the CAN bus. If the CAN FD Controller is required to send a dominant bit (ACK bit, overload flag, active error flag), the bit is rerouted internally so that the CAN FD Controller monitors this dominant bit, although the CAN bus may remain in recessive state. The Tx Buffer Request Pending register (TXBRP) will be cleared when setting Configuration Change Enable CCCR.CCE = "1" upon entering Bus Monitoring Mode via CCCR.MON = "1", and will be held in reset state while it is in this mode.

The Bus Monitoring Mode can be used to analyze the traffic on a CAN bus without affecting it by the transmission of dominant bits. Figure 3-2 shows the connection of signals canfd_tx and canfd_rx to the CAN FD Controller in Bus Monitoring Mode.

Figure 3-2 Pin Control in Bus Monitoring Mode



3.1.7. Disabled Automatic Retransmission

According to the CAN Specification (see ISO11898-1, 6.3.3 Recovery Management), the CAN FD Controller provides means for automatic retransmission of frames that have lost arbitration or that have been disturbed by errors during transmission. By default automatic retransmission is enabled. The automatic retransmission may be disabled via the Disable Automatic Retransmission bit (CCCR.DAR).

In DAR mode all transmissions are automatically cancelled after they started on the CAN bus. In this mode the CAN FD Controller will internally reset the corresponding Tx Buffer's Tx Request Pending bit (TXBRP.TRPn) when one of the following conditions occur:

- A successful transmission has occurred. This can be observed as
  Corresponding Tx Buffer Transmission Occurred bit TXBTO.TOn set
  Corresponding Tx Buffer Cancellation Finished bit TXBCF.CFn not set

  or, in case a successful transmission was executed in spite of cancellation,

  Corresponding Tx Buffer Transmission Occurred bit TXBTO.TOn set
  Corresponding Tx Buffer Cancellation Finished bit TXBCF.CFn set

  In both these cases, if storage of Tx events is enabled, a Tx Event FIFO elements is written with Event Type ET "10" (transmission in spite of cancellation).
- A transmission has not yet been started at the point of cancellation
- A transmission has been aborted due to lost arbitration or frame transmission disturbed. This can be observed by
  Corresponding Tx Buffer Transmission Occurred bit TXBTO.TOn not set
  Corresponding Tx Buffer Cancellation Finished bit TXBCF.CFn set

3.1.8. Power Down (Sleep Mode)

The CAN FD Controller can be set into power down mode via Clock Stop Request (CCCR.CSR).

When all pending transmission requests have completed (when Tx Buffer Request Pending Register TXBRP = 0x00000000), the CAN FD Controller waits until bus idle state is detected. Then the CAN FD Controller sets Initialization (CCCR.INIT) to one to prevent any further CAN transfers. Now the CAN FD Controller acknowledges that it is ready for power down by setting Clock Stop Acknowledge (CCCR.CSA) to one. In this state, before the clocks are switched off, further register accesses can be made. A write access to CCCR.INIT will have no effect. Now the CAN FD Controller clock inputs canfd_bclk and canfd_cclk may be switched off.

To leave power down mode, the application has to turn on the CAN FD Controller clocks before resetting Clock Stop Request (CCCR.CSR). The CAN FD Controller will acknowledge this by resetting CCCR.CSA. Afterwards, the application can restart CAN communication by resetting bit CCCR.INIT.

3.1.9. Test Modes

To enable write access to the Test register TEST, Test Mode Enable (CCCR.TEST) has to be set to one. This allows the configuration of the test modes and test functions.

Four output functions are available for the CAN transmit pin canfd_tx by programming Control of Transmit Pin (TEST.TX). These are

- the default function – the serial data output
- drive the CAN Sample Point signal to monitor the CAN FD Controller's bit timing
- drive constant dominant values
- drive constant recessive values.

The actual value at pin canfd_rx can be read from Receive Pin (TEST.RX). Both functions can be used to check the CAN bus' physical layer.

Due to the synchronization mechanism between the CAN clock and Bus clock domains, there may be a delay of several Bus clock periods between writing to TEST.TX until the new configuration is visible at output pin canfd_tx. This applies also when reading input pin canfd_rx via TEST.RX.

Note:

- *Test modes should be used for self test only. The software control for pin canfd_tx interferes with all CAN protocol functions. It is not recommended to use test modes for application.*

3.1.9.1 External Loop Back Mode

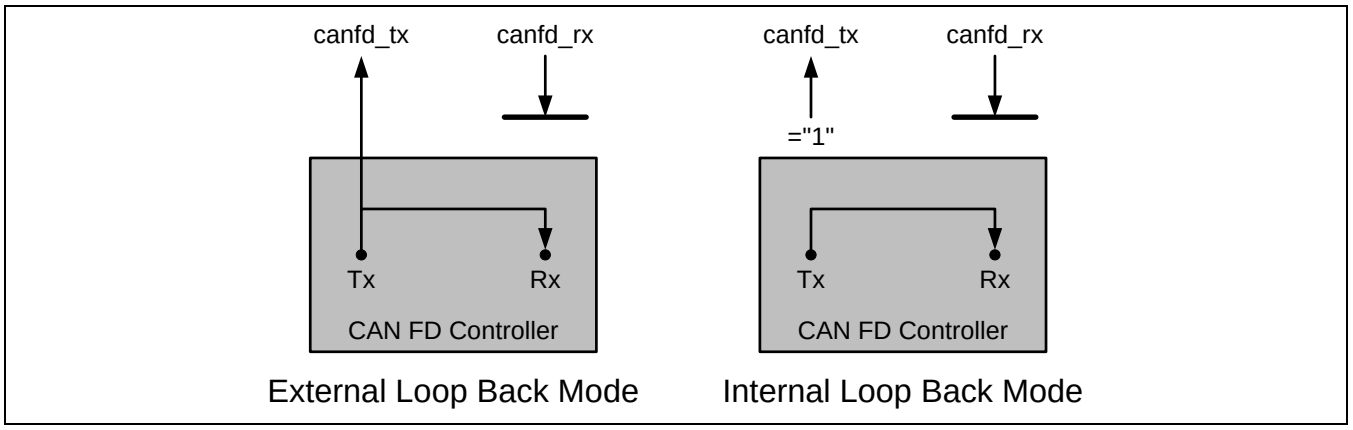
The CAN FD Controller can be set in External Loop Back Mode by programming the Loop Back Mode bit (TEST.LBCK) to one. In External Loop Back Mode, the CAN FD Controller treats its own transmitted messages as received messages and stores them (if they pass acceptance filtering) into a dedicated Rx Buffer or an Rx FIFO. Figure 3-3 shows the connection of signals canfd_tx and canfd_rx to the CAN FD Controller in External Loop Back Mode.

This mode is provided for hardware self test. The CAN FD Controller ignores acknowledge errors in External Loop Back Mode. In this mode the CAN FD Controller performs an internal feedback from its Tx output to its Rx input. The actual value of the canfd_rx input pin is disregarded by the CAN FD Controller. The transmitted messages can be monitored at the canfd_tx pin.

3.1.9.2 Internal Loop Back Mode

Internal Loop Back Mode is entered by programming bits Loop Back Mode (TEST.LBCK) and Bus Monitoring Mode (CCCR.MON) to one. This mode can be used for a "Hot Self test", meaning the CAN FD Controller can be tested without affecting a running CAN system connected to the pins canfd_tx and canfd_rx. In this mode pin canfd_rx is disconnected from the CAN FD Controller and pin canfd_tx is held recessive. Figure 3-3 shows the connection of canfd_tx and canfd_rx to the CAN FD Controller in case of Internal Loop Back Mode.

Figure 3-3 Pin Control in Loop Back Modes



3.2. Timestamp Generation

For timestamp generation the CAN FD Controller supplies a 16-bit wrap-around counter. On start of frame reception/transmission the counter value is captured and stored into the timestamp section of an Rx Buffer and FIFO or Tx Event FIFO element (fields RXTS[15:0] and TXTS[15:0] of the respective elements).

For more details, see "5.8.Timestamp Counter Configuration (MCG_CANFDx_TSCC, CPG_CANFDx_TSCC)" and "5.9.Timestamp Counter Value (MCG_CANFDx_TSCV, CPG_CANFDx_TSCV)".

3.3. Timeout Counter

To signal timeout conditions, the CAN FD Controller supplies a 16-bit Timeout Counter. For Rx FIFO 0, Rx FIFO 1, and Tx Event FIFO triggered operations, it starts down counting when an element is stored, and stops counting when the FIFO is emptied by the CPU. The Timeout Counter may also be triggered by a "0" write to the Initialization bit CCCR.INIT, and count down repeatedly until CCCR.INIT is set back to "1".

The Timeout Counter operates as down-counter and uses the same prescaler controlled by Timestamp Counter Prescaler (TSCC.TCP[3:0]) as the Timestamp Counter. The Timeout Counter is configured via register Timeout Counter Configuration (TOCC). The actual counter value can be read from Timeout Counter (TOCV.TOC[15:0]). The Timeout Counter is operable only when CCCR.INIT = "0". It is stopped when Initialization CCCR.INIT = "1", e.g. when the CAN FD Controller enters Bus_Off state.

The operation mode is selected by the Timeout Select field TOCC.TOS[1:0]. For details, see "5.11.Timeout Counter Value (MCG_CANFDx_TOCV, CPG_CANFDx_TOCV) ".

Note:

- *When Timestamp Select TSCC.TSS[1:0] = "01", the clock signal for the Timeout Counter is derived from the CAN Core's sample point signal. Therefore, if the bit rate switch feature in CAN FD is used, the timeout counter is clocked differently in arbitration and data field.*

3.4. Rx Handling

The Rx Handler controls the acceptance filtering, the transfer of received messages to the dedicated Rx Buffers or to one of the two Rx FIFOs, as well as the Rx FIFO's Put and Get Indices.

3.4.1.Acceptance Filtering

3.4.2.Rx FIFOs

3.4.3.Dedicated Rx Buffers

3.4.4.Debug on CAN Support

3.4.5.Protocol Exception Event

3.4.1. Acceptance Filtering

The CAN FD Controller offers the possibility to configure two sets of acceptance filters, one for standard identifiers and one for extended identifiers. These filters can be assigned to a dedicated Rx Buffer or to Rx FIFO 0, 1. For acceptance filtering each list of filters is executed from element #0 until the first matching element. Acceptance filtering stops at the first matching element. The following filter elements are not evaluated for this message.

Main Features

- Each filter element can be configured as
  - range filter (from - to)
  - filter for one or two specific IDs (dual filter)
  - classic bit mask filter
  - filter for a single dedicated ID
- Each filter element is configurable for acceptance or rejection filtering
- Each filter element can be enabled/disabled individually
- Filters are checked sequentially from filter element 0, execution stops with the first matching filter element

Related Configuration Registers

- Global Filter Configuration GFC
- Standard ID Filter Configuration SIDFC
- Extended ID Filter Configuration XIDFC
- Extended ID AND Mask XIDAM

Filter Properties

Depending on the configuration of the filter element configuration fields (SFEC[2:0]/EFEC[2:0]) a match triggers one of the following actions:

- Store received frame in Rx FIFO 0 or Rx FIFO 1
- Store received frame in dedicated Rx Buffer
- Reject received frame
- Set High Priority Message interrupt flag IR.HPM
- Set High Priority Message interrupt flag IR.HPM and store received frame in Rx FIFO 0 or Rx FIFO 1

Description

Acceptance filtering is started after the complete identifier has been received. After acceptance filtering has completed, and if a matching dedicated Rx Buffer or Rx FIFO has been found, the Message Handler starts writing the received message data in portions of 32 bits to the matching dedicated Rx Buffer or Rx FIFO.

If the CAN protocol controller has detected an error condition (e.g. CRC error), this message is discarded with the following impact on the affected dedicated Rx Buffer or Rx FIFO:

- Dedicated Rx Buffer

  New Data flag NDAT1/2.NDn of matching Rx Buffer is not set, but Rx Buffer (partly) overwritten with received data. For error type see Last Error Code (PSR.LEC[2:0]) respectively Data Phase Last Error Code (PSR.DLEC[2:0]).

- Rx FIFO

Put index of matching Rx FIFO RXFnS.FnPI[5:0] is not updated, but related Rx FIFO element (partly) overwritten with received data. For error type see PSR.LEC[2:0] respectively PSR.DLEC[2:0].

In case the matching Rx FIFO is operated in overwrite mode, the boundary conditions described in "3.4.2.2. Rx FIFO Overwrite Mode" have to be considered.

3.4.1.1 Filter Types

Range Filter

A range filter matches all received frames with Message IDs in the range defined by the Standard Message ID Filter Element SFID1[10:0]/SFID2[10:0] fields resp. Extended Message ID Filter Element EFID1[28:0]/EFID2[28:0] fields (the range includes the end points SFID1[10:0]/SFID2[10:0] resp. EFID1[28:0]/EFID2[28:0]).

There are two possibilities for the Extended Filter Type of the Extended Message ID Filter Element (EFT[1:0]) when range filtering is used together with extended frames:

- EFT[1:0] = "00":

  The Message ID of received frames is ANDed with the Extended ID AND Mask (XIDAM) before the range filter is applied

- EFT[1:0] = "11":

  The Extended ID AND Mask (XIDAM) is not used for range filtering

Filter for Specific IDs (Dual Filter)

A filter element can be configured to filter for one or two specific Message IDs. To filter for one specific Message ID, the filter element has to be configured with SFID1[10:0] = SFID2[10:0] resp. EFID1[28:0] = EFID2[28:0].

Classic Bit Mask Filter

Classic bit mask filtering is intended to filter groups of Message IDs by masking single bits of a received Message ID. With classic bit mask filtering SFID1[10:0]/EFID1[28:0] is used as Message ID filter, while SFID2[10:0]/EFID2[28:0] is used as filter mask.

A zero bit at the filter mask will mask out the corresponding bit position of the configured ID filter, e.g. the value of the received Message ID at that bit position is not relevant for acceptance filtering. Only those bits of the received Message ID where the corresponding mask bits are one are relevant for acceptance filtering.

In case all mask bits are one, a match occurs only when the received Message ID and the Message ID filter are identical. If all mask bits are zero, all Message IDs match.

Filter for a Single Dedicated ID

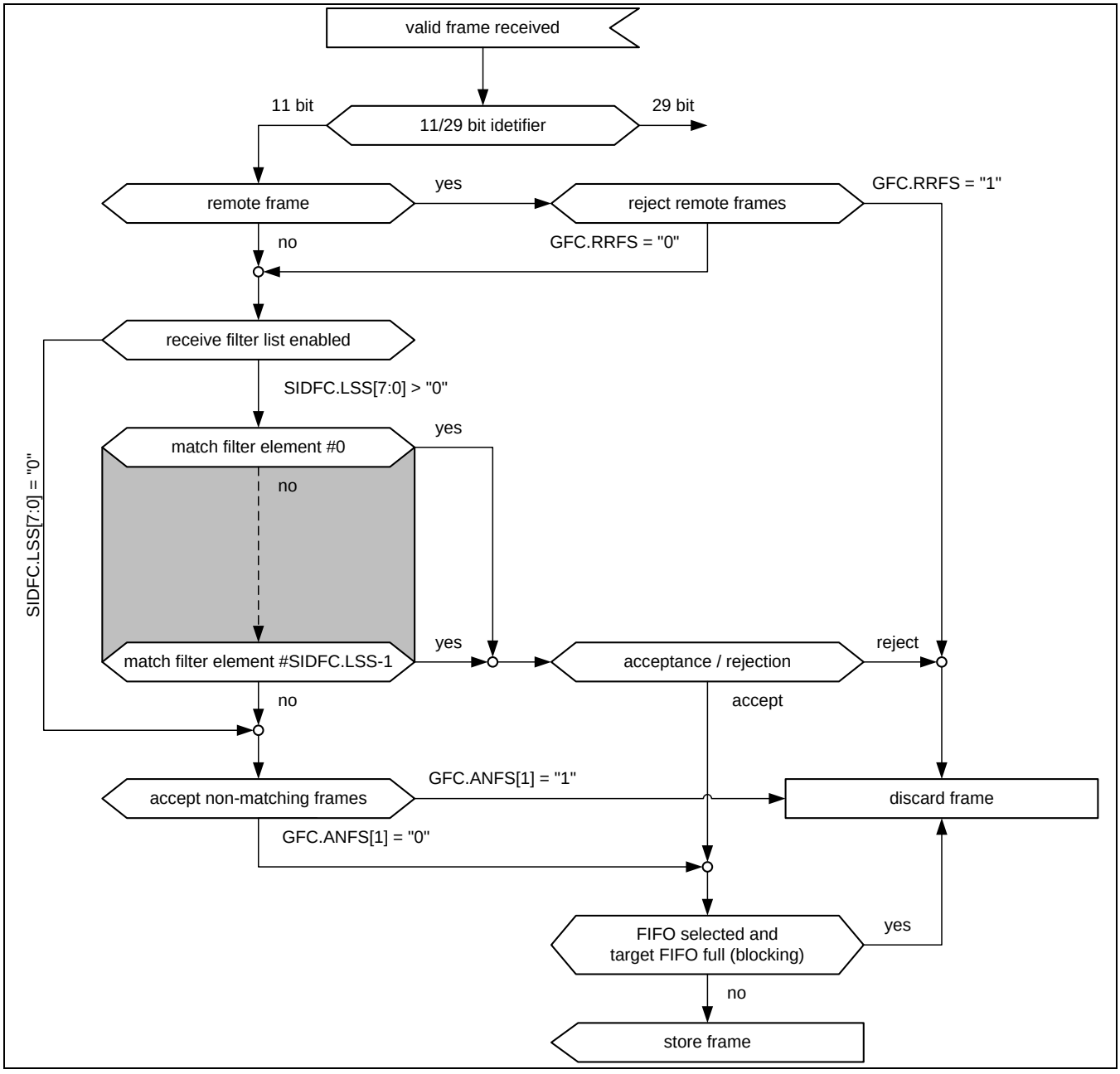
A filter for a single dedicated ID stores received messages in dedicated Rx Buffers if the received ID matches SFID1[10:0]/EFID1[28:0]. SFID2[10:0]/EFID2[28:0] is used to configure the offset to the Rx Buffer Start Address (RXBC.RBSA[15:2]) of where the message is to be stored, and also to define how the message is to be treated (Debug message or Dedicated Rx message).

3.4.1.2 Standard Message ID Filtering

Figure 3-4 below shows the flow for standard Message ID (11-bit Identifier) filtering.

Controlled by the Global Filter Configuration GFC and the Standard ID Filter Configuration SIDFC, Message ID, Remote Transmission Request bit (RTR), and the Identifier Extension bit (IDE) of received frames are compared against the list of configured filter elements.

Figure 3-4 Standard Message ID Filter Path



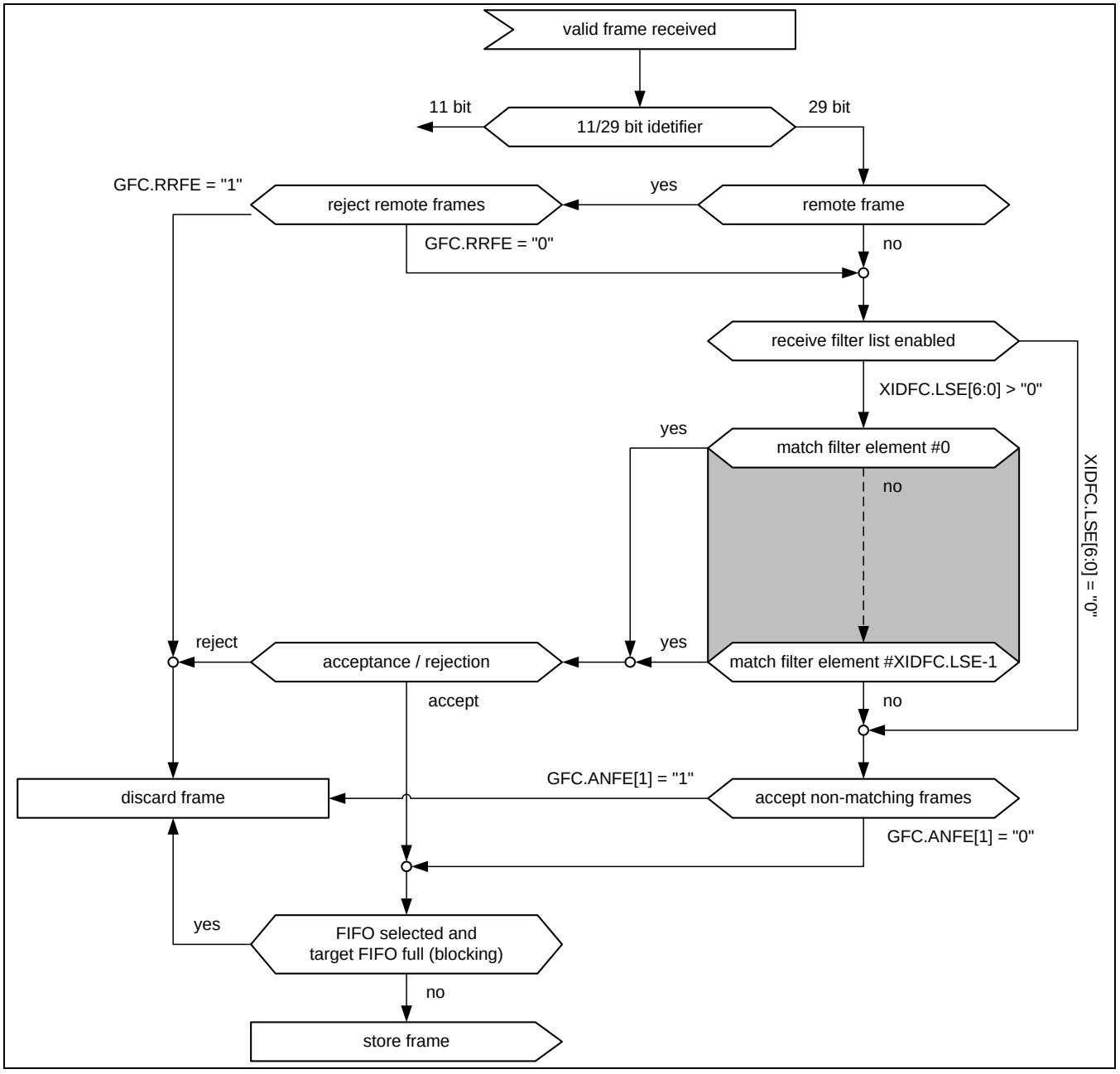
3.4.1.3 Extended Message ID Filtering

Figure 3-5 below shows the flow for extended Message ID (29-bit Identifier) filtering.

Controlled by the Global Filter Configuration GFC and the Extended ID Filter Configuration XIDFC, Message ID, Remote Transmission Request bit (RTR), and the Identifier Extension bit (IDE) of received frames are compared against the list of configured filter elements.

The Extended ID AND Mask XIDAM.EIDM[28:0] is ANDed with the received identifier before the filter list is executed.

Figure 3-5 Extended Message ID Filter Path



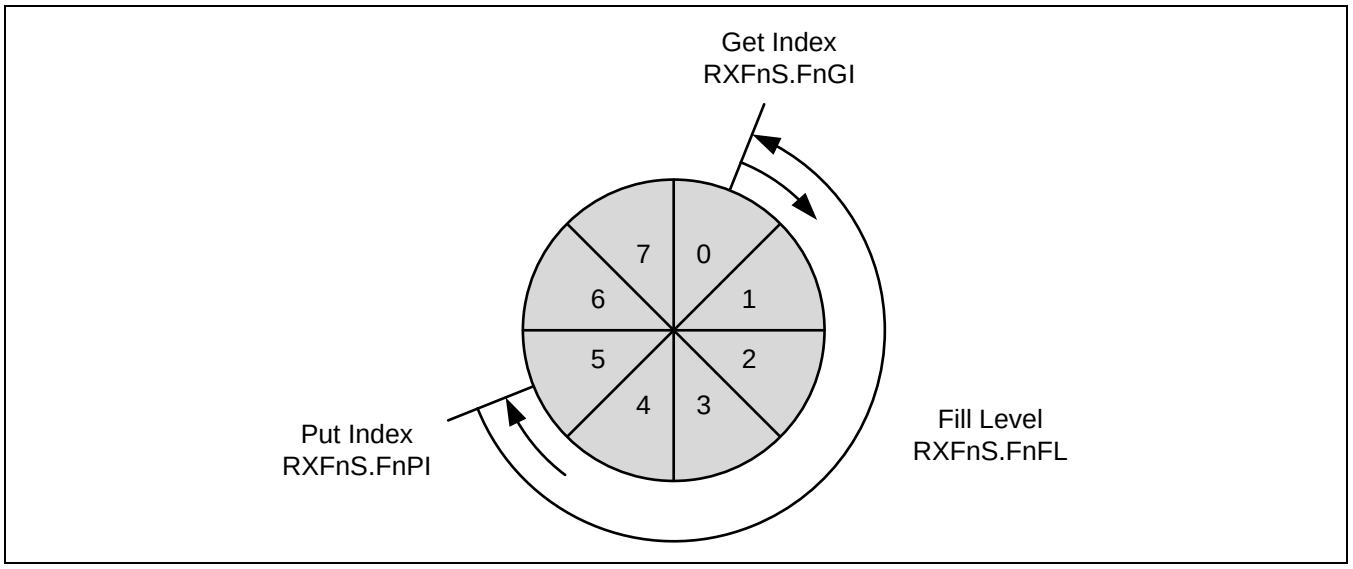
3.4.2. Rx FIFOs

Rx FIFO 0 and Rx FIFO 1 can be configured to hold up to 64 elements each. Configuration of the two Rx FIFOs is done via the Rx FIFO 0 and 1 Configuration registers RXF0C and RXF1C.

Received messages that passed acceptance filtering are transferred to the Rx FIFO as configured by the matching filter element.

To avoid an Rx FIFO overflow, the Rx FIFO watermark can be used. When the Rx FIFO fill level RXFnS.FnFL[6:0] reaches the Rx FIFO watermark configured by Rx FIFO n Watermark (RXFnC.FnWM[6:0]), interrupt flag Rx FIFO n Watermark Reached (IR.RFnW) is set. When the Rx FIFO Put Index reaches the Rx FIFO Get Index an Rx FIFO Full condition is signalled by Rx FIFO n Full (RXFnS.FnF). In addition interrupt flag Rx FIFO n Full (IR.RFnF) is set.

Figure 3-6 RX FIFO Status



Addressing Rx Buffers

An Rx Buffer allocates Element Size 32-bit words in the Message RAM (see Table 3-2).Therefore when reading from an Rx FIFO, the start address of an Rx Buffer will be

Rx FIFO Get Index (RXFnS.FnGI[5:0]) × Element Size
+ corresponding Rx FIFO start address (RXFnC.FnSA[15:2]).

Table 3-2 Rx Buffer/FIFO Element Size

RXESC.RBDS[2:0] RXESC.FnDS[2:0]	Data Field [Bytes]	Element Size [RAM words]
000	8	4
001	12	5
010	16	6
011	20	7
100	24	8
101	32	10
110	48	14
111	64	18

3.4.2.1 Rx FIFO Blocking Mode

The Rx FIFO blocking mode is configured by FIFO n Operation Mode RXFnC.FnOM = "0". This is the default operation mode for the Rx FIFOs.

When an Rx FIFO full condition is reached (RXFnS.FnPI[5:0] = RXFnS.FnGI[5:0]), no further messages are written to the corresponding Rx FIFO until at least one message has been read out and the Rx FIFO Get Index has been incremented. An Rx FIFO full condition is signalled by Rx FIFO n Full (RXFnS.FnF = "1"). In addition interrupt flag Rx FIFO n Full (IR.RFnF) is set.

In case a message is received while the corresponding Rx FIFO is full, this message is discarded and the message lost condition is signalled by Rx FIFO n Message Lost RXFnS.RFnL = "1". In addition interrupt flag Rx FIFO n Message Lost (IR.RFnL) is set.

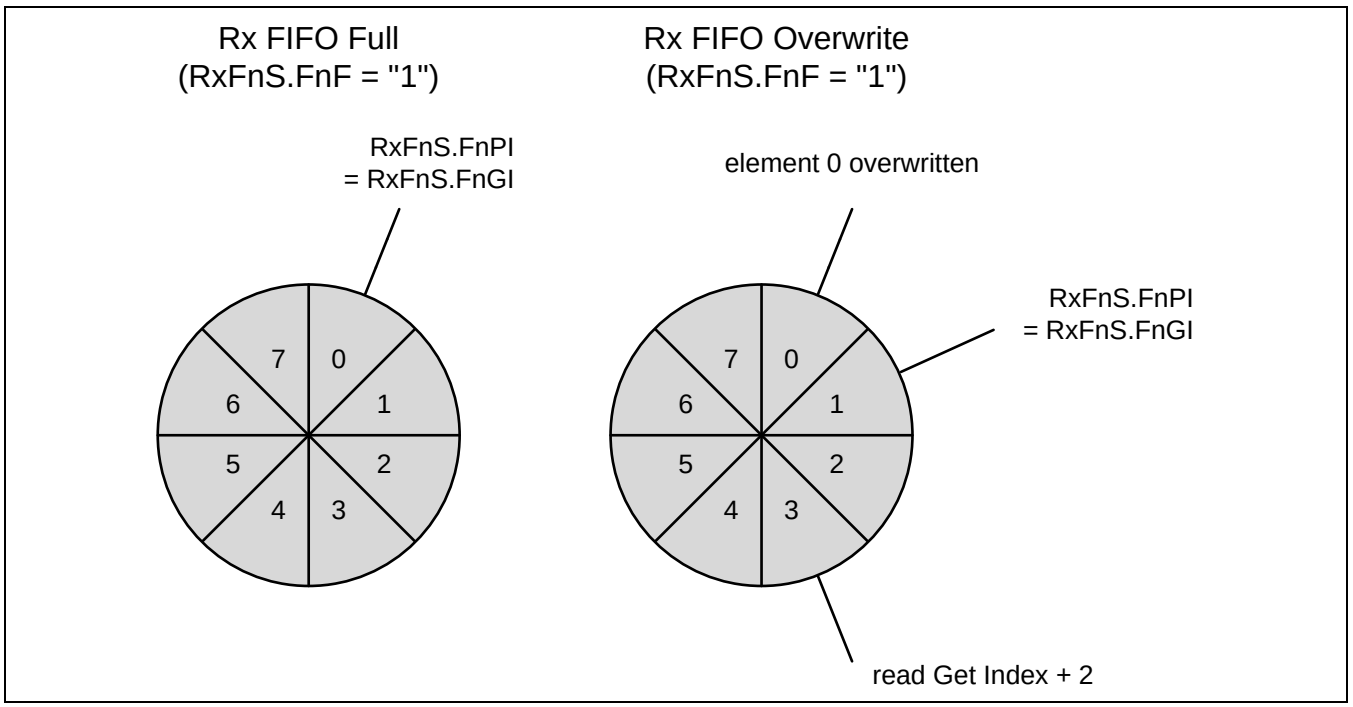
3.4.2.2 Rx FIFO Overwrite Mode

The Rx FIFO overwrite mode is configured by FIFO n Operation Mode RXFnC.FnOM = "1".

When an Rx FIFO full condition (RXFnS.FnPI[5:0] = RXFnS.FnGI[5:0]) is signalled by Rx FIFO n Full RXFnS.FnF = "1", the next message accepted for the Rx FIFO will overwrite the oldest Rx FIFO message. Put and get index are both incremented by one.

When an Rx FIFO is operated in overwrite mode and an Rx FIFO full condition is signalled, reading of the Rx FIFO elements should start at least at get index + 1. The reason for that is, that it might happen, that a received message is written to the Message RAM (put index) while the CPU is reading from the Message RAM (get index). In this case inconsistent data may be read from the respective Rx FIFO element. Adding an offset to the get index when reading from the Rx FIFO avoids this problem. The offset depends on how fast the CPU accesses the Rx FIFO. Figure 3-7 shows an offset of two with respect to the get index when reading the Rx FIFO. In this case the two messages stored in element 1 and 2 are lost.

Figure 3-7 RX FIFO Overflow Handling



After reading from the Rx FIFO, the number of the last element read has to be written to the Rx FIFO Acknowledge Index (RXFnA.FnA[5:0]). This increments the get index to that element number. In case the put index has not been incremented to this Rx FIFO element, the Rx FIFO full condition is reset (Rx FIFO n Full RXFnS.FnF = "0").

3.4.3. Dedicated Rx Buffers

The CAN FD Controller supports up to 64 dedicated Rx Buffers. The start address of the dedicated Rx Buffer section is configured via the Rx Buffer Start Address field (RXBC.RBSA[15:2]).

For each dedicated Rx Buffer a Standard or Extended Message ID Filter Element with SFEC[2:0]/EFEC[2:0] = "111" and SFID2[10:9]/EFID2[10:9] = "00" has to be configured.

After a received message has been accepted by a filter element, the message is stored into the Rx Buffer in the Message RAM referenced by the filter element. The format is the same as for an Rx FIFO element. In addition the flag IR.DRX (Message stored to Dedicated Rx Buffer) in the interrupt register is set.

Table 3-3 Example Filter Configuration for Rx Buffers

Filter Element	SFID1[10:0] EFID1[28:0]	SFID2[10:9] EFID2[10:9]	SFID2[5:0] EFID2[5:0]
0	ID message 1	00	00 0000
1	ID message 2	00	00 0001
2	ID message 3	00	00 0010

After the last word of a matching received message has been written to the Message RAM, the respective New Data flag in register NDAT1, 2 is set. As long as the New Data flag is set, the respective dedicated Rx Buffer is locked against updates from received matching frames. The New Data flags have to be reset by the CPU by writing a "1" to the respective bit position.

While a dedicated Rx Buffer's New Data flag is set, a Message ID Filter Element referencing this specific dedicated Rx Buffer will not match, causing the acceptance filtering to continue. Following Message ID Filter Elements may cause the received message to be stored into another dedicated Rx Buffer, or into an Rx FIFO, or the message may be rejected, depending on filter configuration.

Addressing Dedicated Rx Buffers

A Dedicated Rx Buffer allocates Element Size 32-bit words in the Message RAM (see Table 3-2). Therefore the start address of a dedicated Rx Buffer in the Message RAM is calculated by

(Offset defined in Message ID Filter Element SFID2[5:0] resp. EFID2[5:0]) × Element Size + Rx Buffer Start Address (RXBC.RBSA[15:2]).

3.4.4. Debug on CAN Support

For debug handling, three consecutive Rx buffers (e.g. #61, #62, #63) can be used for storage of three specific debug messages A, B, and C. The format is the same as for a dedicated Rx Buffer or an Rx FIFO element.

For filtering of debug messages Standard/Extended Filter Elements with Standard/Extended Filter Configuration SFEC[2:0]/EFEC[2:0] = "111" have to be set up. Messages matching these filter elements are stored into the Rx Buffers addressed by the lower six bits of the Standard/Extended Message ID Filter Element fields SFID2[5:0]/EFID2[5:0]. The upper two bits SFID2[10:9]/EFID2[10:9] are used to define the Debug Message that is to be stored in the respective Rx Buffer.

When a debug message is stored, neither the respective New Data flag (NDAT1.NDn / NDAT2.NDn) nor the interrupt IR.DRX (Message stored to Dedicated Rx Buffer) are set.

After the three messages have been received in correct order, a DMA transfer is requested. After the DMA transfer has completed, the CAN FD Controller is prepared to receive the next set of debug messages.

Table 3-4 Example Filter Configuration for Debug Messages

Filter Element	SFID1[10:0] EFID1[28:0]	SFID2[10:9] EFID2[10:9]	SFID2[5:0] EFID2[5:0]
0	ID debug message A	01	11 1101
1	ID debug message B	10	11 1110
2	ID debug message C	11	11 1111

Notes:

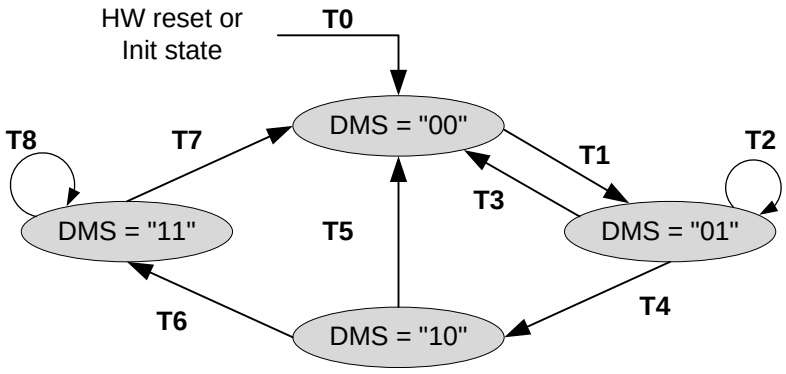
- *Debug on CAN feature is not supported for TRAVEO™ T1G Platform.*
- *DMA transfer request is not supported also.*

3.4.4.1 Debug Message Handling

The debug message handling state machine assures that debug messages are stored to three consecutive Rx Buffers in correct order. In case of missing messages the process is restarted. The DMA request is activated only when all three debug messages A, B, C have been received in correct order.

The status of the debug message handling state machine is signalled via the Debug Message Status field (RXF1S.DMS[1:0]).

Figure 3-8 Debug Message Handling State Machine



T0: reset DMA request, enable reception of debug messages A, B, and C
T1: reception of debug message A
T2: reception of debug message A
T3: reception of debug message C
T4: reception of debug message B
T5: reception of debug messages A, B
T6: reception of debug message C
T7: DMA transfer completed
T8: reception of debug message A, B, C (message rejected)

Notes:

- *Debug message is used for Debug on CAN feature.*
- *Debug on CAN feature is not supported for TRAVEO™ T1G Platform.*

3.4.5. Protocol Exception Event

When the CAN FD Controller is enabled both CAN FD operation (CCCR.FDOE = "1") and Protocol exception handling (CCCR.PXHD = "0"), the CAN FD Controller detects the res bit following the FDF bit in a received message to be recessive, it treats this detection as a Protocol Exception Event. As reaction to the Protocol Exception Event,

- the error counters (ECR.REC[6:0] and ECR.TEC[7:0]) are not changed
- hard synchronization is enabled
- the CAN FD Controller will send recessive bits
- Protocol Exception Event (PSR.PXE) bit is set, then the operation state will change from Receiver (PSR.ACT[1:0] = "10") to Integrating (PSR.ACT[1:0] = "00") at the next sample point

The CAN FD Controller will also start an internal bit counter that counts the number of recessive bits detected on the CAN bus. If the counter detects eleven consecutive recessive bits, the CAN FD Controller will recover from the Protocol Exception Event and enter an idle state. In case an edge that causes synchronization is detected during the wait period, the CAN FD Controller will reset the internal bit counter and restart counting the number of recessive bits detected.

3.5. Tx Handling

The Tx Handler handles transmission requests for the dedicated Tx Buffers, the Tx FIFO, and the Tx Queue. It controls the transfer of transmit messages to the CAN Core, the Put and Get Indices, and the Tx Event FIFO. Up to 32 Tx Buffers can be set up for message transmission. The CAN mode for transmission (Classic CAN or CAN FD) can be configured separately for each Tx Buffer element. The Tx Buffer element is described in "6.3. Tx Buffer Element". Table 3-5 below describes the possible configurations for frame transmission.

Table 3-5 Possible Configurations for Frame Transmission

CCCR		Tx Buffer Element		Frame Transmission
BRSE	FDOE	FDF	BRS	
ignored	0	ignored	ignored	Classic CAN
0	1	0	ignored	Classic CAN
0	1	1	ignored	FD without bit rate switching
1	1	0	ignored	Classic CAN
1	1	1	0	FD without bit rate switching
1	1	1	1	FD with bit rate switching

The Tx Handler starts a Tx scan to check for the highest priority pending Tx request (Tx Buffer with lowest Message ID) when the Tx Buffer Request Pending register TXBRP is updated, or when a transmission has been started.

3.5.1.Transmit Pause

3.5.2.Dedicated Tx Buffers

3.5.3.Tx FIFO

3.5.4.Tx Queue

3.5.5.Mixed Dedicated Tx Buffers/Tx FIFO

3.5.6.Mixed Dedicated Tx Buffers/Tx Queue

3.5.7.Transmit Cancellation

3.5.8.Tx Event Handling

3.5.1. Transmit Pause

The transmit pause feature is intended for use in CAN systems where the CAN message identifiers are (permanently) specified to specific values and cannot easily be changed. These message identifiers may have a higher CAN arbitration priority than other defined messages, while in a specific application their relative arbitration priority should be inverse. This may lead to a case where one ECU (Electronic Control Unit) sends a burst of CAN messages that cause another ECU's CAN messages to be delayed because that other messages have a lower CAN arbitration priority.

If e.g. CAN ECU-1 has the transmit pause feature enabled and is requested by its application software to transmit four messages, it will, after the first successful message transmission, wait for two nominal bit times of bus idle before it is allowed to start the next requested message. If there are other ECUs with pending messages, those messages are started in the idle time, they would not need to arbitrate

with the next message of ECU-1. After having received a message, ECU-1 is allowed to start its next transmission as soon as the received message releases the CAN bus.

The transmit pause feature is controlled by the Transmit Pause bit (CCCR.TXP). If the bit is set, the CAN FD Controller will, each time it has successfully transmitted a message, pause for two nominal bit times before starting the next transmission. This enables other CAN nodes in the network to transmit messages even if their messages have lower prior identifiers. Default is transmit pause disabled (CCCR.TXP = "0").

This feature looses up burst transmissions coming from a single node and it protects against "babbling idiot" scenarios where the application program erroneously requests too many transmissions.

3.5.2. Dedicated Tx Buffers

Dedicated Tx Buffers are intended for message transmission under complete control of the CPU. Each Dedicated Tx Buffer is configured with a specific Message ID. In case that multiple Tx Buffers are configured with the same Message ID, the Tx Buffer with the lowest buffer number is transmitted first.

If the data section has been updated, a transmission is requested by an Add Request via TXBAR.ARn. The requested messages arbitrate internally with messages from an optional Tx FIFO or Tx Queue and externally with messages on the CAN bus, and are sent out according to their Message ID.

Addressing Dedicated Tx Buffers

A Dedicated Tx Buffer allocates Element Size 32-bit words in the Message RAM (see Table 3-6). Therefore the start address of a dedicated Tx Buffer in the Message RAM is calculated by

transmit buffer index (0 to 31) × Element Size + Tx Buffers Start Address (TXBC.TBSA[15:2]).

Table 3-6 Tx Buffer/FIFO/Queue Element Size

TXESC.TBDS[2:0]	Data Field [Bytes]	Element Size [RAM words]
000	8	4
001	12	5
010	16	6
011	20	7
100	24	8
101	32	10
110	48	14
111	64	18

3.5.3. Tx FIFO

Tx FIFO operation is configured by programming the Tx FIFO/Queue Mode bit (TXBC.TFQM) to "0". Messages stored in the Tx FIFO are transmitted starting with the message referenced by the Tx FIFO Get Index (TXFQS.TFGI[4:0]). After each transmission the Get Index is incremented cyclically until the Tx FIFO is empty. The Tx FIFO enables transmission of messages with the same Message ID from different Tx Buffers in the order these messages have been written to the Tx FIFO. The CAN FD Controller calculates the number of available (free) Tx FIFO elements TXFQS.TFFL[5:0] (Tx FIFO Free Level) as difference between Tx FIFO Get Index (TXFQS.TFGI[4:0]) and Tx FIFO/Queue Put Index (TXFQS.TFQPI[4:0]).

Adding Messages and Requesting Transmissions

New transmit messages have to be written to the Tx FIFO starting with the Tx Buffer referenced by the Put Index (TXFQS.TFQPI[4:0]). An Add Request increments the Put Index to the next free Tx FIFO element. When the Put Index reaches the Get Index, Tx FIFO Full (TXFQS.TFQF = "1") is signalled. In this case no further messages should be written to the Tx FIFO until the next message has been transmitted and the Get Index has been incremented.

When a single message is added to the Tx FIFO, the transmission is requested by writing a "1" to the Add Request bit (TXBAR.ARn. n is the index of the requested Tx Buffer) related to the Tx Buffer referenced by the Tx FIFO's Put Index.

When multiple (n) messages are added to the Tx FIFO, they are written to n consecutive Tx Buffers starting with the Put Index. The transmissions are then requested via the Tx Buffer Add Request register (TXBAR) by requesting all n Tx FIFO Buffers in a single command. The Put Index is then cyclically incremented by n. The number of requested Tx buffers should not exceed the number of free Tx Buffers as indicated by the Tx FIFO Free Level (TXFQS.TFFL[5:0]). Transmission should not be requested to Tx Buffers outside the Put Index since this will cause the TXFQS.TFFL[5:0] to return a false value.

Cancelling Transmissions

Transmit Cancellation is intended for Tx Queues and Dedicated Tx Buffers. Cancelling requested Tx FIFO Buffer transmissions are not supported and is not recommended, except for the case where all Tx Buffers within the Tx FIFO are cancelled simultaneously (with a single command).

In the case that a transmission request for the Tx Buffer referenced by the Get Index is cancelled, the Get Index is incremented to the next Tx Buffer with pending transmission request and the Tx FIFO Free Level is recalculated. When transmission cancellation is applied to any other Tx Buffer, the Get Index and the FIFO Free Level remain unchanged.

Addressing Tx Buffers in the Tx FIFO

A Tx FIFO element allocates Element Size 32-bit words in the Message RAM (see Table 3-6). Therefore the start address of the next available (free) Tx FIFO Buffer is calculated by

Tx FIFO/Queue Put Index (TXFQS.TFQPI[4:0]) (0 to 31) × Element Size
+ Tx Buffers Start Address (TXBC.TBSA[15:2]).

3.5.4. Tx Queue

Tx Queue operation is configured by programming the Tx FIFO/Queue Mode bit TXBC.TFQM to "1". Messages stored in the Tx Queue are transmitted starting with the message with the lowest Message ID (highest priority). In case that multiple Queue Buffers are configured with the same Message ID, the Queue Buffer with the lowest buffer number is transmitted first.

Adding Messages and Requesting Transmissions

New messages have to be written to the Tx Buffer referenced by the Tx FIFO/Queue Put Index TXFQS.TFQPI[4:0]. An Add Request cyclically increments the Put Index to the next free Tx Buffer. In case that the Tx Queue is full (TXFQS.TFQF = "1"), the Put Index is not valid and no further message should be written to the Tx Queue until at least one of the requested messages has been sent out or a pending transmission request has been cancelled.

The application may use the Tx Buffer Request Pending register TXBRP instead of the Put Index and may place messages to any Tx Buffer without pending transmission request.

Addressing Tx Buffers in the Tx Queue

A Tx Queue Buffer allocates Element Size 32-bit words in the Message RAM (see Table 3-6). Therefore the start address of the next available (free) Tx Queue Buffer is calculated by

Tx FIFO/Queue Put Index (TXFQS.TFQPI[4:0]) (0 to 31) × Element Size
+ Tx Buffers Start Address (TXBC.TBSA[15:2]).

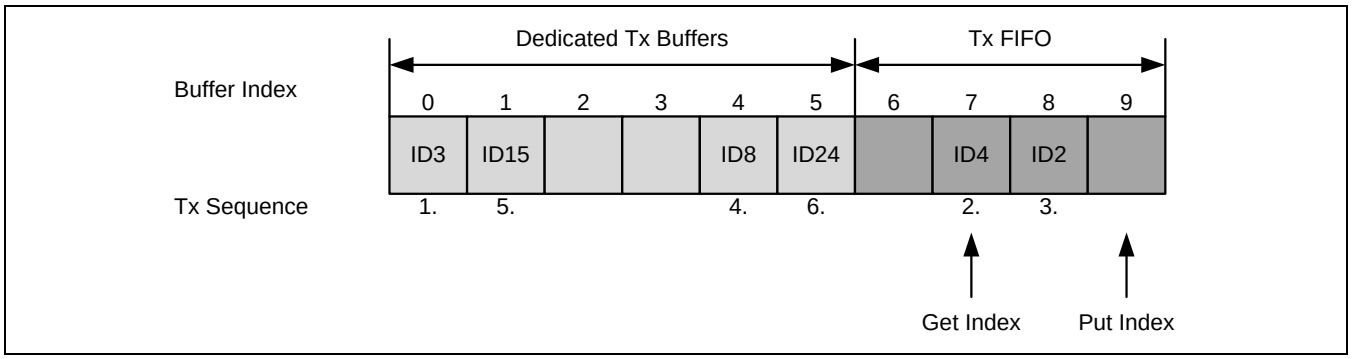
3.5.5. Mixed Dedicated Tx Buffers/Tx FIFO

In this case the Tx Buffers section in the Message RAM is subdivided into a set of Dedicated Tx Buffers and a Tx FIFO.

The number of Dedicated Tx Buffers is configured by TXBC.NDTB[5:0].

The number of Tx Buffers assigned to the Tx FIFO is configured by the Tx FIFO/Queue Size TXBC.TFQS[5:0]. In case TXBC.TFQS[5:0] is programmed to zero, only Dedicated Tx Buffers are used.

Figure 3-9 Example of Mixed Configuration Dedicated Tx Buffers/Tx FIFO



Tx Prioritization:

- Scan Dedicated Tx Buffers and oldest pending Tx FIFO Buffer (referenced by the Tx FIFO Get Index TXFQS.TFGI[4:0])
- Buffer with lowest Message ID gets highest priority and is transmitted next

Note:

- *Mixed Dedicated Tx Buffers/Tx FIFO isn't supported. Don't use combination Dedicated Tx Buffers and Tx FIFO.*

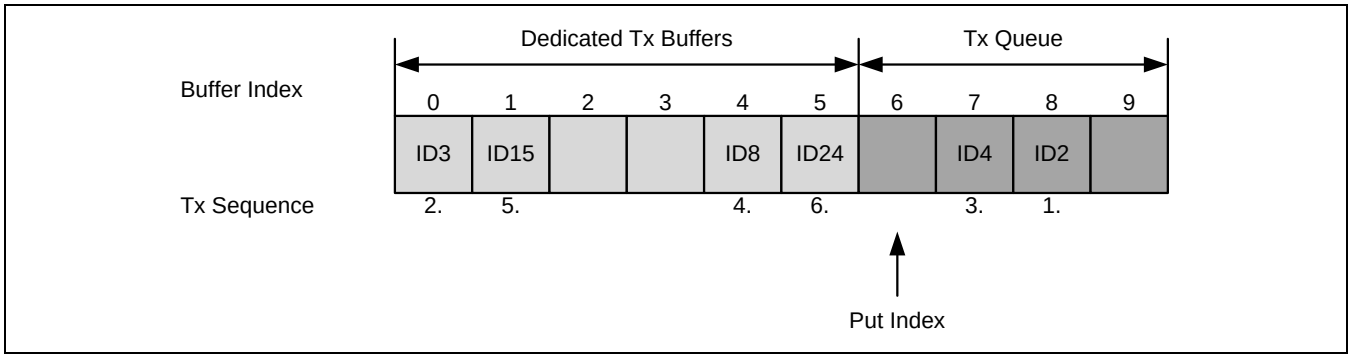
3.5.6. Mixed Dedicated Tx Buffers/Tx Queue

In this case the Tx Buffers section in the Message RAM is subdivided into a set of Dedicated Tx Buffers and a Tx Queue.

The number of Dedicated Tx Buffers is configured by TXBC.NDTB[5:0].

The number of Tx Queue Buffers is configured by the Tx FIFO/Queue Size TXBC.TFQS[5:0]. In case TXBC.TFQS[5:0] is programmed to zero, only Dedicated Tx Buffers are used.

Figure 3-10 Example of Mixed Configuration Dedicated Tx Buffers/Tx Queue



Tx Prioritization:

- Scan all Tx Buffers with activated transmission request
- Tx Buffer with lowest Message ID gets highest priority and is transmitted next

3.5.7. Transmit Cancellation

The CAN FD Controller supports transmit cancellation. To cancel a requested transmission from a dedicated Tx Buffer or a Tx Queue Buffer the CPU has to write a "1" to the corresponding bit position of the Tx Buffer Cancellation Request register TXBCR. Transmit cancellation is not intended for Tx FIFO operation.

Successful cancellation is signalled by setting the corresponding bit of the Tx Buffer Cancellation Finished register TXBCF to "1".

A cancellation request resets the corresponding Transmission Request Pending bit (TXBRP.TRPn). In case a transmit cancellation is requested while a transmission from a Tx Buffer is already ongoing, the corresponding TXBRP.TRPn bit remains set as long as the transmission is in progress.

- If the transmission was successful, the corresponding Transmission Occurred (TXBTO.TOn) and Cancellation Finished (TXBCF.CFn) bits are set.
- If the transmission was not successful (the transmission has been aborted due to lost arbitration, error occurred during frame transmission), it is not repeated and only the corresponding Cancellation Finished bit (TXBCF.CFn) is set.

TXBCF.CFn is also set when the transmission has not yet been started at the point of cancellation,

The cancellation request bits (TXBCR.CRn) are reset directly after the corresponding Transmission Request Pending bit (TXBRP.TRPn) has been reset.

Note:

- *In case a pending transmission is cancelled immediately before this transmission could have been started, there follows a short time window where no transmission is started even if another message is also pending in this node. This may enable another node to transmit a message which may have a lower priority (a greater message ID) than the second message in this node.*

3.5.8. Tx Event Handling

To support Tx event handling the CAN FD Controller has implemented a Tx Event FIFO. After the CAN FD Controller has transmitted a message on the CAN bus, Message ID and timestamp are stored in a Tx Event FIFO element. To link a Tx event to a Tx Event FIFO element, the Message Marker from the transmitted Tx Buffer MM[7:0] is copied into the Tx Event FIFO element.

The Tx Event FIFO can be configured by Event FIFO Size (TXEFC.EFS[5:0]) to a maximum of 32 elements.

The purpose of the Tx Event FIFO is to decouple handling transmit status information from transmit message handling i.e. a Tx Buffer holds only the message to be transmitted, while the transmit status is stored separately in the Tx Event FIFO. This has the advantage, especially when operating a dynamically managed transmit queue, that a Tx Buffer can be used for a new message immediately after successful transmission. There is no need to save transmit status information from a Tx Buffer before overwriting that Tx Buffer.

When a Tx Event FIFO full condition is signalled by interrupt flag Tx Event FIFO Full (IR.TEFF), no further elements are written to the Tx Event FIFO until at least one element has been read out and the Tx Event FIFO Get Index TXEFS.EFGI[4:0] has been incremented. In case a Tx event occurs while the Tx Event FIFO is full, this event is discarded and interrupt flag Tx Event FIFO Element Lost IR.TEFL is set.

To avoid a Tx Event FIFO overflow, the Tx Event FIFO watermark can be used. When the Tx Event FIFO fill level TXEFS.EFFL[5:0] reaches the Tx Event FIFO watermark configured by TXEFC.EFWM[5:0], interrupt flag Tx Event FIFO Watermark Reached IR.TEFW is set.

- Addressing Tx Events in the Tx Event FIFO

The start address when reading from the Tx Event FIFO will be

2 × Tx Event FIFO Get Index TXEFS.EFGI[4:0] + Tx Event FIFO start address TXEFC.EFSA[15:2].

3.6. FIFO Acknowledge Handling

The Get Indices of Rx FIFO 0, Rx FIFO 1, and the Tx Event FIFO (RXF0S.F0GI[5:0], RXF1S.F1GI[5:0], and TXEFS.EFGI[4:0], respectively) are controlled by writing to the corresponding FIFO Acknowledge Index (RXF0A.F0AI[5:0], RXF1A.F1AI[5:0], and TXEFA.EFAI[4:0], respectively). Writing to the FIFO Acknowledge Index will set the FIFO Get Index to the FIFO Acknowledge Index plus one and thereby updates the FIFO Fill Level. There are two use cases:

- When only a single element has been read from the FIFO (the one being pointed to by the Get Index), this Get Index value is written to the FIFO Acknowledge Index. (Multiple elements may be read by repeating this operation)
- When a sequence of elements has been read from the FIFO, it is sufficient to write the FIFO Acknowledge Index only once at the end of that read sequence (value: Index of the last element read), to update the FIFO's Get Index.

Due to the fact that the CPU has free access to the Message RAM, special care has to be taken when reading FIFO elements in an arbitrary order (Get Index not considered). This might be useful when reading a High Priority Message from one of the two Rx FIFOs. In this case the Rx FIFO's Acknowledge Index should not be written because this would set the Get Index to a wrong position, alter the Rx FIFO's Fill Level, and also cause some of the older Rx FIFO elements to be lost.

Note:

- *The application has to ensure that a valid value is written to the FIFO Acknowledge Index. The CAN FD Controller does not check for erroneous values.*

3.7. Configuring the CAN Bit Timing

Each CAN node in the CAN network has its own clock generator (usually a quartz oscillator). The time parameter of the bit time can be configured individually for each CAN node. Even if each CAN node's oscillator has a different period, a common bit rate can be generated.

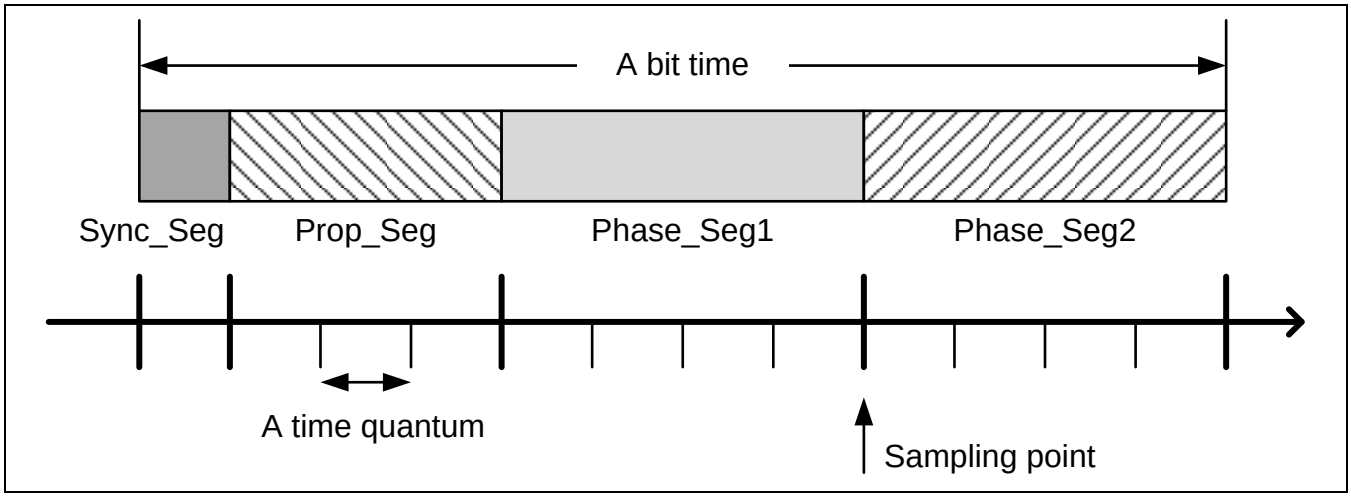
The oscillator frequencies vary slightly because of changes in temperature or voltage, or deterioration of components. As long as the frequencies vary only within the tolerance range of the oscillators, the CAN nodes can compensate for the different bit rates by resynchronizing to the bit stream.

3.7.1. CAN Bit Timing

The CAN FD operation defines two bit times - nominal bit time and data bit time. The nominal bit time is for the arbitration phase. The data bit time has an equal or shorter length and can be used to accelerate the data phase (see Bit Rate Switching in "3.1.3.CAN FD Operation" for details).

The basic construction of a bit time is shared with both the nominal and data bit times. The bit time can be divided into four segments according to the CAN specifications (see Figure 3-11): the synchronization segment (Sync_Seg), the propagation time segment (Prop_Seg), the phase buffer segment 1 (Phase_Seg1), and the phase buffer segment 2 (Phase_Seg2). The sample point, the point of time at which the bus level is read and interpreted as the value of that respective bit, is located at the end of Phase_Seg1.

Figure 3-11 Bit Time Construction



Each segment consists of a programmable number of time quanta, which is a multiple of the time quantum that is defined by canfd_cclk and a prescaler. The values and prescalers used to define these parameters differ for the nominal and data bit times, and are configured by NBTP (Nominal Bit Timing & Prescaler Register) and DBTP (Data Bit Timing & Prescaler Register) as shown in the table below (Table 3-7).

Table 3-7 Bit Time Parameters

Parameter	Description
Time quantum tq (nominal) and tqd (data)	Time quantum. Derived by multiplying the basic unit time quanta (i.e. the canfd_cclk period) with the respective prescaler. The time quantum is configured by the CAN FD Controller as nominal : tq = (NBTP.NBRP[8:0] +1) × canfd_cclk period data : tqd = (DBTP.DBRP[4:0] + 1) × canfd_cclk period
Sync_Seg	Sync_Seg is fixed to 1 time quantum as defined by the CAN specifications and is therefore not configurable (inherently built into the CAN FD Controller). nominal : 1 tq data : 1 tqd
Prop_Seg	Prop_Seg is the part of the bit time that is used to compensate for the physical delay times within the network. The CAN FD Controller configures the sum of Prop_Seg and Phase_Seg1 with a single parameter, i.e. nominal : Prop_Seg + Phase_Seg1 = NBTP.NTSEG1[7:0] + 1 data : Prop_Seg + Phase_Seg1 = DBTP.DTSEG1[4:0] + 1
Phase_Seg1	Phase_Seg1 is used to compensate for edge phase errors before the sampling point. Can be lengthened by the resynchronization jump width. The sum of Prop_Seg and Phase_Seg1 is configured by the CAN FD Controller as nominal : NBTP.NTSEG1[7:0] + 1 data : DBTP.DTSEG1[4:0] + 1
Phase_Seg2	Phase_Seg2 is used to compensate for edge phase errors after the sampling point. Can be shortened by the resynchronization jump width. Phase_Seg2 is configured by the CAN FD Controller as nominal : NBTP.NTSEG2[6:0] + 1 data : DBTP.DTSEG2[3:0] + 1
SJW	Resynchronization Jump Width. Used to adjust the length of Phase_Seg1 and Phase_Seg2. SJW will not be longer than either Phase_Seg1 or Phase_Seg2. SJW is configured by the CAN FD Controller as nominal : NBTP.NSJW[6:0] + 1 data : DBTP.DSJW[3:0] + 1

These relations result in the following equations for the nominal and data bit times:

Nominal Bit time

= [Sync_Seg + Prop_Seg + Phase_Seg1 + Phase_Seg2] × tq
= [1 + (NBTP.NTSEG1[7:0] + 1) + (NBTP.NTSEG2[6:0] + 1)] × [(NBTP.NBRP[8:0] +1) × canfd_cclk period]

and for the data bit time

= [1 + (DBTP.DTSEG1[4:0] + 1) + (DBTP.DTSEG2[3:0] + 1)] × [(DBTP.DBRP[4:0] +1) × canfd_cclk period]

Note:

- *The Information Processing Time (IPT) of the CAN FD Controller is zero, meaning the data for the next bit is available at the first CAN clock edge after the sample point. Therefore, the IPT does not have to be accounted for when configuring Phase_Seg2, which is the maximum of Phase_Seg1 and the IPT.*

3.7.2. CAN Bit Rates

Since the bit rate is the inverse of the bit time, the nominal bit rate is

1 / [{1 + (NBTP.NTSEG1[7:0] + 1) + (NBTP.NTSEG2[6:0] + 1)} × {(NBTP.NBRP[8:0] +1) × canfd_cclk period}]

and the data bit rate is

1 / [{1 + (DBTP.DTSEG1[4:0] + 1) + (DBTP.DTSEG2[3:0] + 1)} × {(DBTP.DBRP[4:0] +1) × canfd_cclk period}]

From these formula we can see that the bit rates of the CAN FD Controller will depend on the CAN clock (i.e. canfd_cclk) period, and the range each parameter can be configured to. The tables below lists examples of the configurable bit rates at varying CAN clock frequencies. Empty boxes indicate that the desired bit rate cannot be configured at the specified input CAN clock frequency.

Table 3-8 Example Configurations for Nominal Bit Rates

CAN clock frequency	8MHz		10MHz		16MHz		20MHz		32MHz		40MHz	
configuration / nominal bit rate	Number of tqs per bit time	NBTP.NBRP + 1	Number of tqs per bit time	NBTP.NBRP + 1	Number of tqs per bit time	NBTP.NBRP + 1	Number of tqs per bit time	NBTP.NBRP + 1	Number of tqs per bit time	NBTP.NBRP + 1	Number of tqs per bit time	NBTP.NBRP + 1
125kbps	64tq 32tq 16tq 8tq	1 2 4 8	80tq 40tq 20tq 10tq	1 2 4 8	128tq 64tq 32tq 16tq 8tq	1 2 4 8 16	160tq 80tq 40tq 20tq 10tq	1 2 4 8 16	256tq 128tq 64tq 32tq 16tq 8tq	1 2 4 8 16 32	320tq 160tq 80tq 40tq 20tq 10tq	1 2 4 8 16 32
250kbps	32tq 16tq 8tq	1 2 4	40tq 20tq 10tq	1 2 4	64tq 32tq 16tq 8tq	1 2 4 8	80tq 40tq 20tq 10tq	1 2 4 8	128tq 64tq 32tq 16tq 8tq	1 2 4 8 16	160tq 80tq 40tq 20tq 10tq	1 2 4 8 16
500kbps	16tq 8tq	1 2	20tq 10tq	1 2	32tq 16tq 8tq	1 2 4	40tq 20tq 10tq	1 2 4	64tq 32tq 16tq 8tq	1 2 4 8	80tq 40tq 20tq 10tq	1 2 4 8
1Mbps	8tq	1	10tq	1	16tq 8tq	1 2	20tq 10tq	1 2	32tq 16tq 8tq	1 2 4	40tq 20tq 10tq	1 2 4

Table 3-9 Example Configurations for Data Bit Rates

CAN clock frequency	8MHz		10MHz		16MHz		20MHz		32MHz		40MHz	
configuration / data bit rate	Number of tqs per bit time	DBTP.DBRP + 1	Number of tqs per bit time	DBTP.DBRP + 1	Number of tqs per bit time	DBTP.DBRP + 1	Number of tqs per bit time	DBTP.DBRP + 1	Number of tqs per bit time	DBTP.DBRP + 1	Number of tqs per bit time	DBTP.DBRP + 1
500kbps	16tqd 8tqd	1 2	20tqd 10tqd	1 2	32tqd 16tqd 8tqd	1 2 4	40tqd 20tqd 10tqd	1 2 4	32tqd 16tqd 8tqd	2 4 8	40tqd 20tqd 10tqd	2 4 8
1Mbps	8tqd	1	10tqd	1	16tqd 8tqd	1 2	20tqd 10tqd	1 2	32tqd 16tqd 8tqd	1 2 4	40tqd 20tqd 10tqd	1 2 4
2Mbps	-	-	-	-	8tqd	1	10tqd	1	16tqd 8tqd	1 2	20tqd 10tqd	1 2
4Mbps	-	-	-	-	-	-	-	-	8tqd	1	10tqd	1
5Mbps	-	-	-	-	-	-	-	-	-	-	8tqd	1

Note:

- *The user must configure the CAN bit timings so that comply with the corresponding CAN standards to ensure proper communication on the CAN bus.*

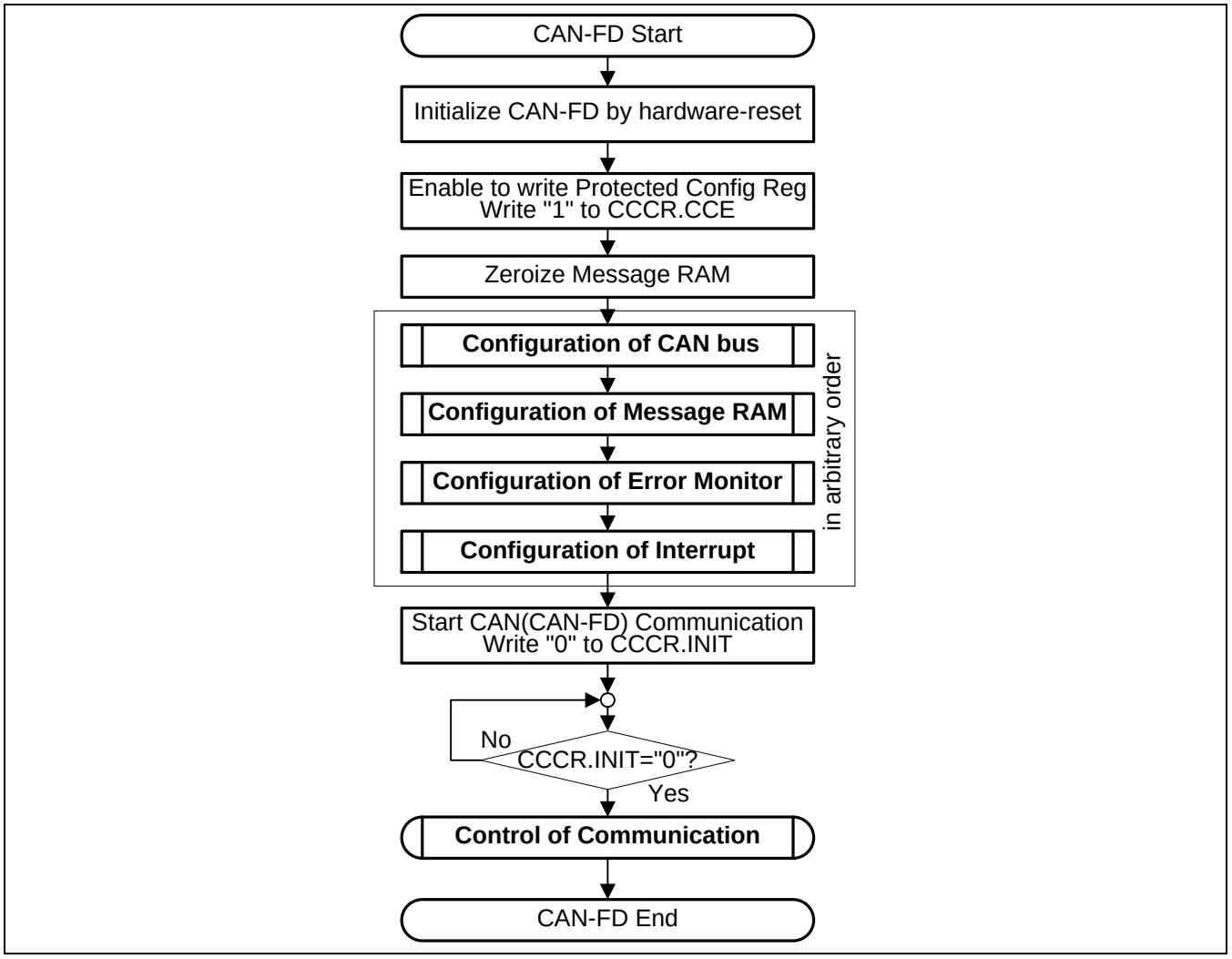
4. Setup Procedure Examples

This section provides examples of setup procedures for the CAN FD Controller.

4.1.Configuration of CAN Bus
4.2.Configuration of Message RAM
4.2.1.Configuration of ID Filter List
4.2.2.Configuration of Rx Buffer and Rx FIFO
4.2.3.Configuration of Tx Buffer and Tx FIFO/Queue
4.2.4.Configuration of ID Filter
4.3.Configuration of Error Monitor
4.4.Configuration of Interrupt
4.5.Control of Communication
4.5.1.Configuration of Transmission Frame
4.6.Interrupt Handling Operation
4.6.1.Bus_Off Status Handling Operation
4.6.2.Message RAM Access Failure Handling Operation
4.6.3.Bit Error Handling Operation
4.6.4.Tx Event FIFO Handling Operation
4.6.5.Dedicated Rx Buffer Handling Operation
4.6.6.High Priority Message Handling Operation
4.6.7.Rx FIFO Handling Operation

Figure 4-1 shows the general flow of a program. The following sections will describe each step in more detail.

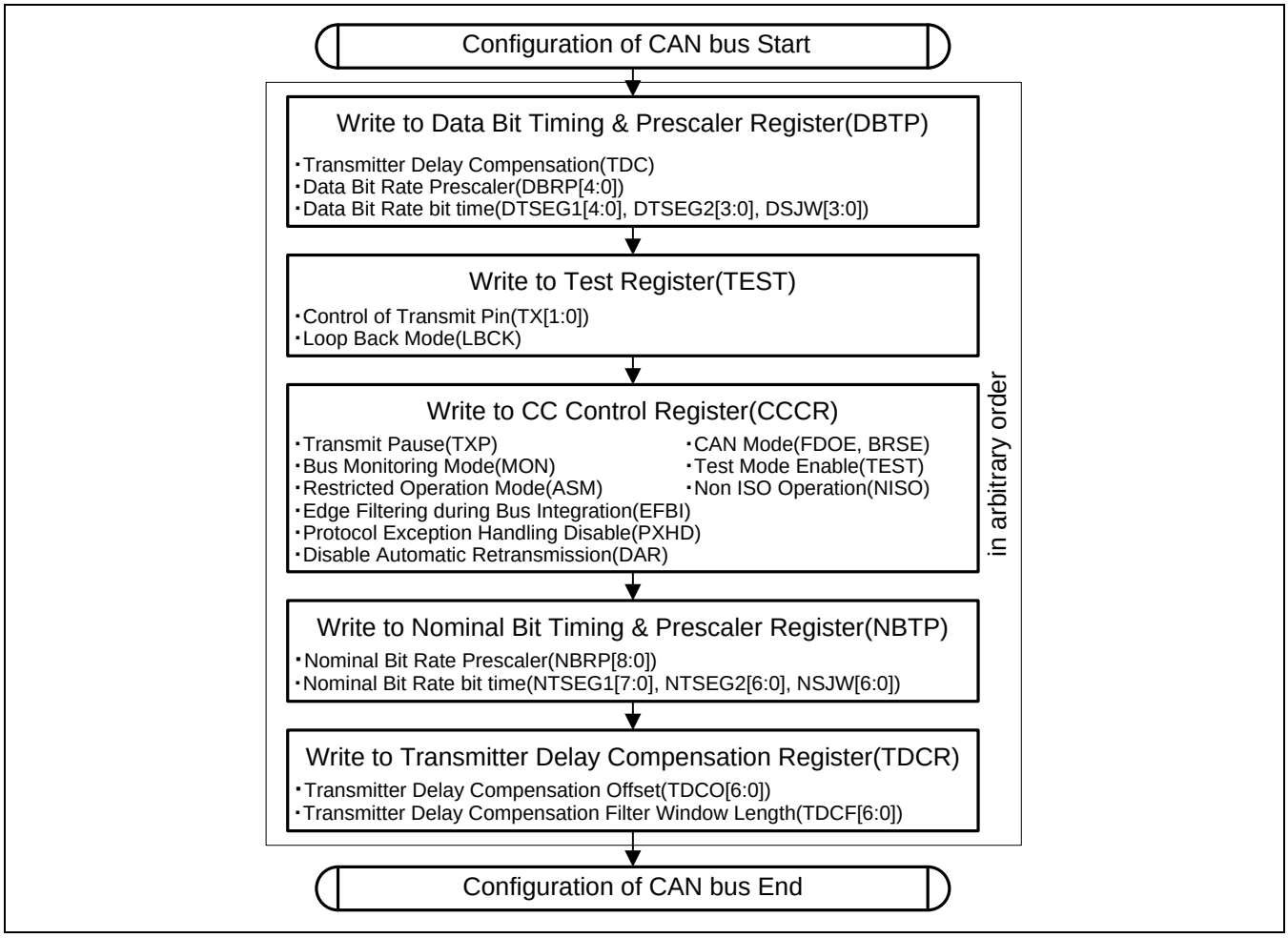
Figure 4-1 General Program Flow



4.1. Configuration of CAN Bus

Figure 4-2 lists what to configure in order to set up the CAN bus related functions.

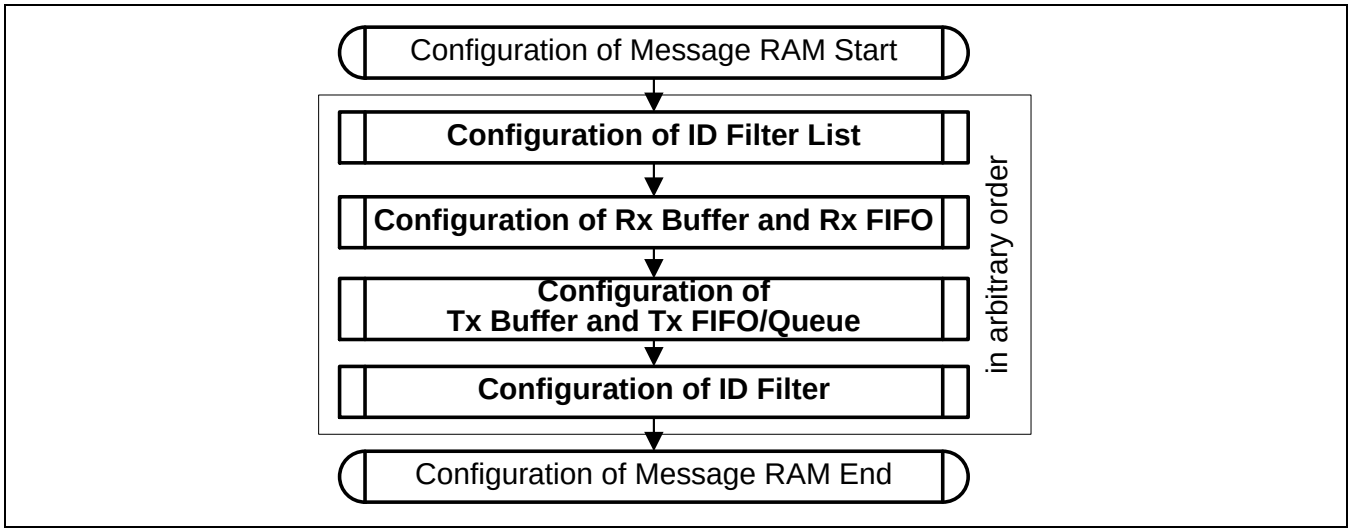
Figure 4-2 Flow Diagram "Configuration of CAN Bus"



4.2. Configuration of Message RAM

Figure 4-3 shows the overall configuration procedure of the Message RAM.

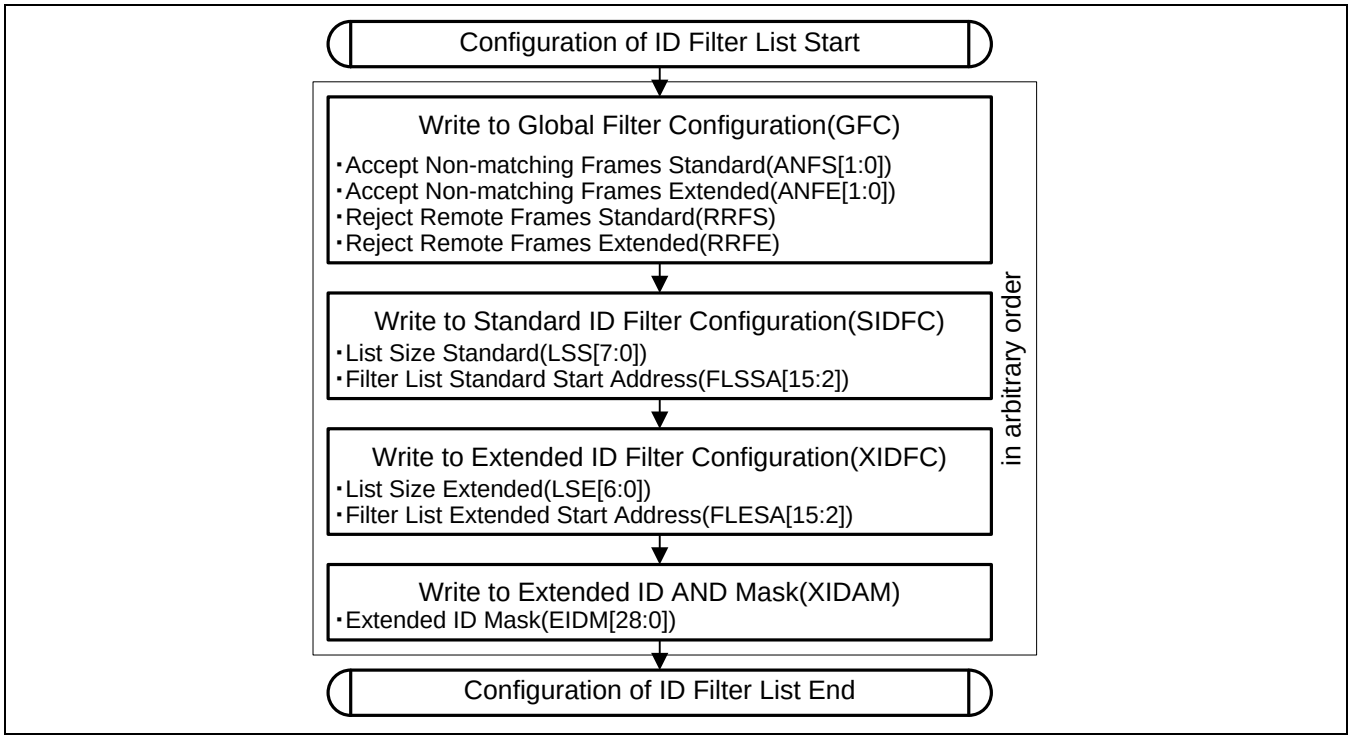
Figure 4-3 Flow Diagram "Configuration of Message RAM"



4.2.1. Configuration of ID Filter List

Figure 4-4 lists what to configure in order to set up the ID Filter lists.

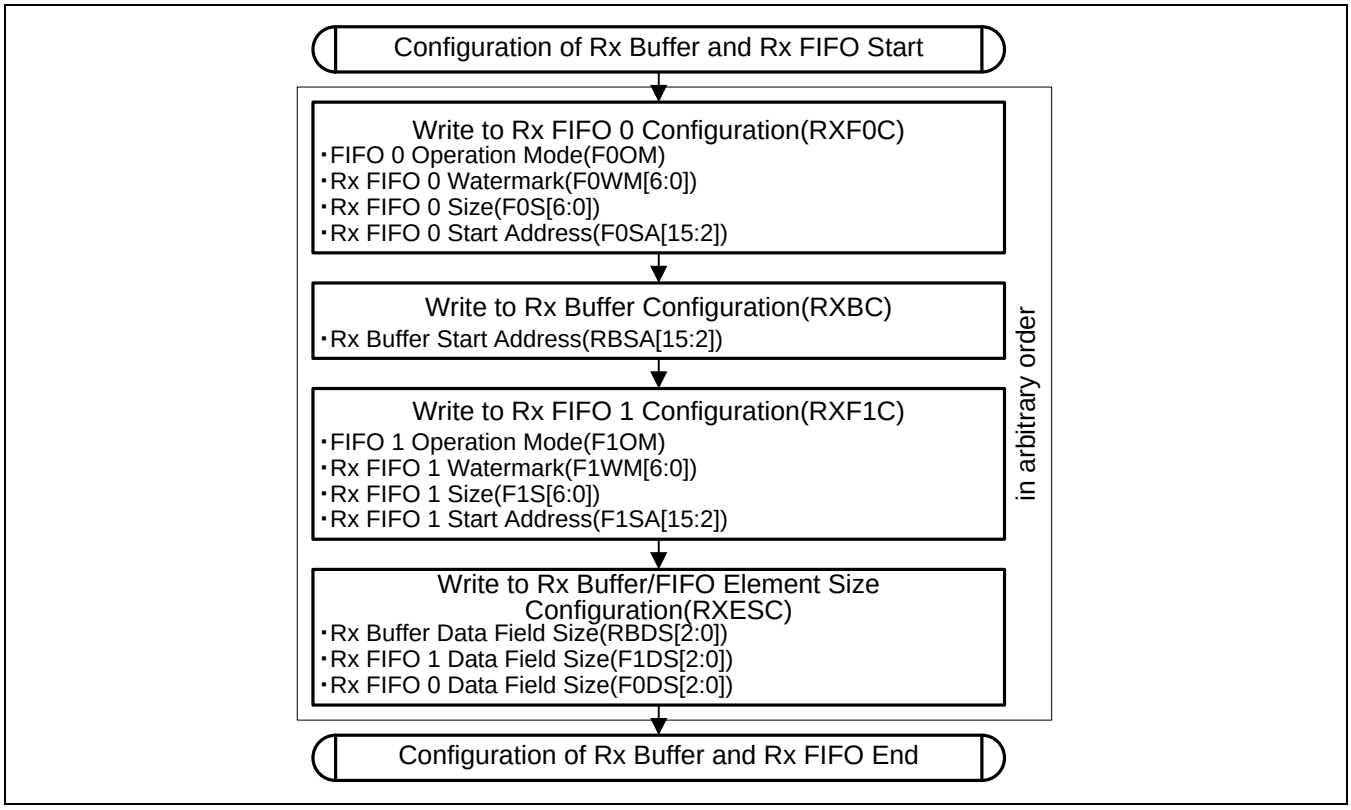
Figure 4-4 Flow Diagram "Configuration of ID Filter List"



4.2.2. Configuration of Rx Buffer and Rx FIFO

Figure 4-5 lists what to configure in order to set up the Dedicated Rx Buffers and Rx FIFOs.

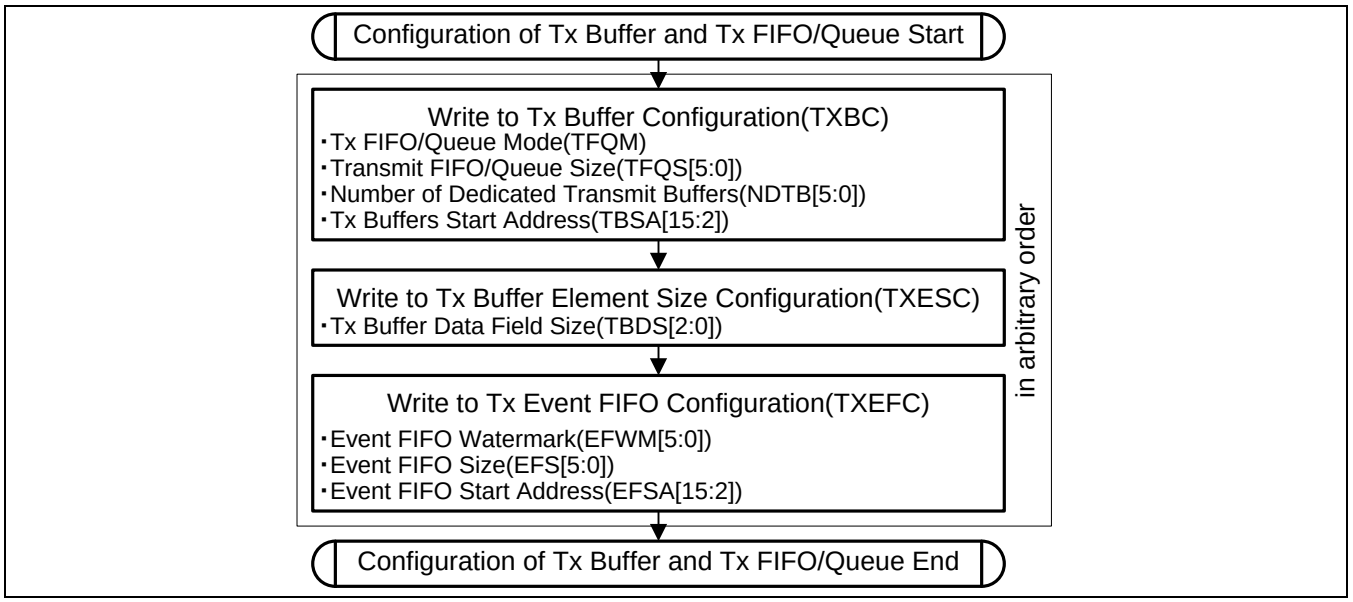
Figure 4-5 Flow Diagram "Configuration of Rx Buffer and Rx FIFO"



4.2.3. Configuration of Tx Buffer and Tx FIFO/Queue

Figure 4-6 lists what to configure in order to set up the Tx Buffers and Tx Event FIFO.

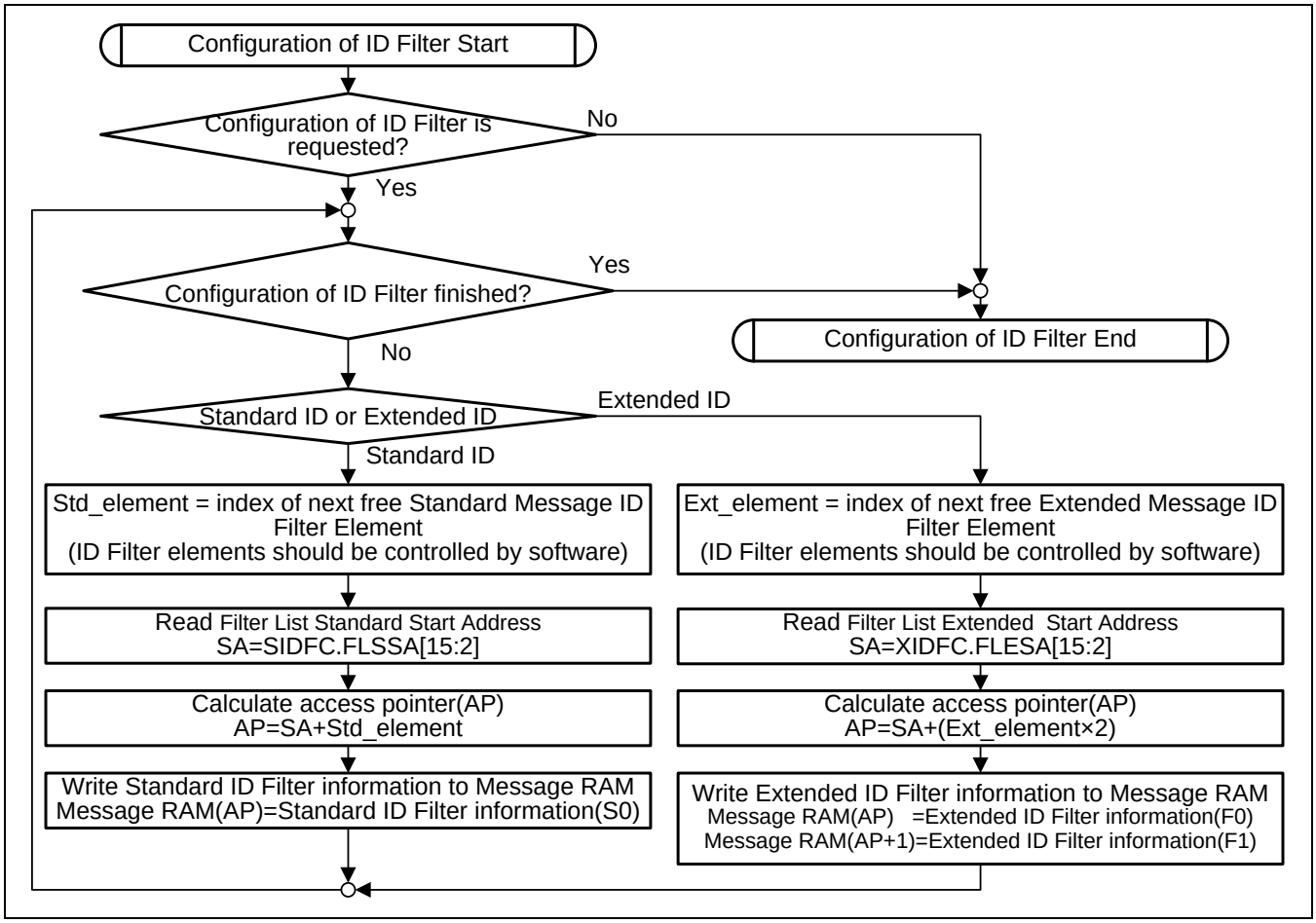
Figure 4-6 Flow Diagram "Configuration of Tx Buffer and Tx FIFO/Queue"



4.2.4. Configuration of ID Filter

Figure 4-7 shows the steps to set up an ID Filter.

Figure 4-7 Flow Diagram "Configuration of ID Filter"



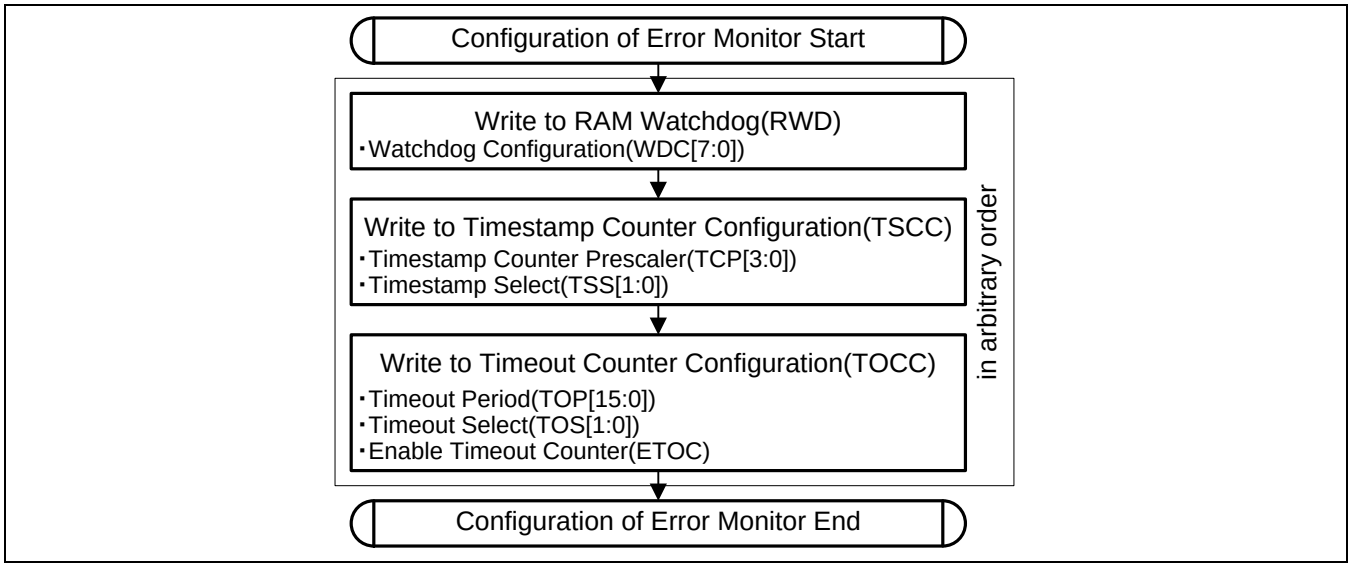
Note:

- *Use 32-bit word address values when calculating the pointers in the flow diagram above.*

4.3. Configuration of Error Monitor

Figure 4-8 lists the error monitoring functions that can be used.

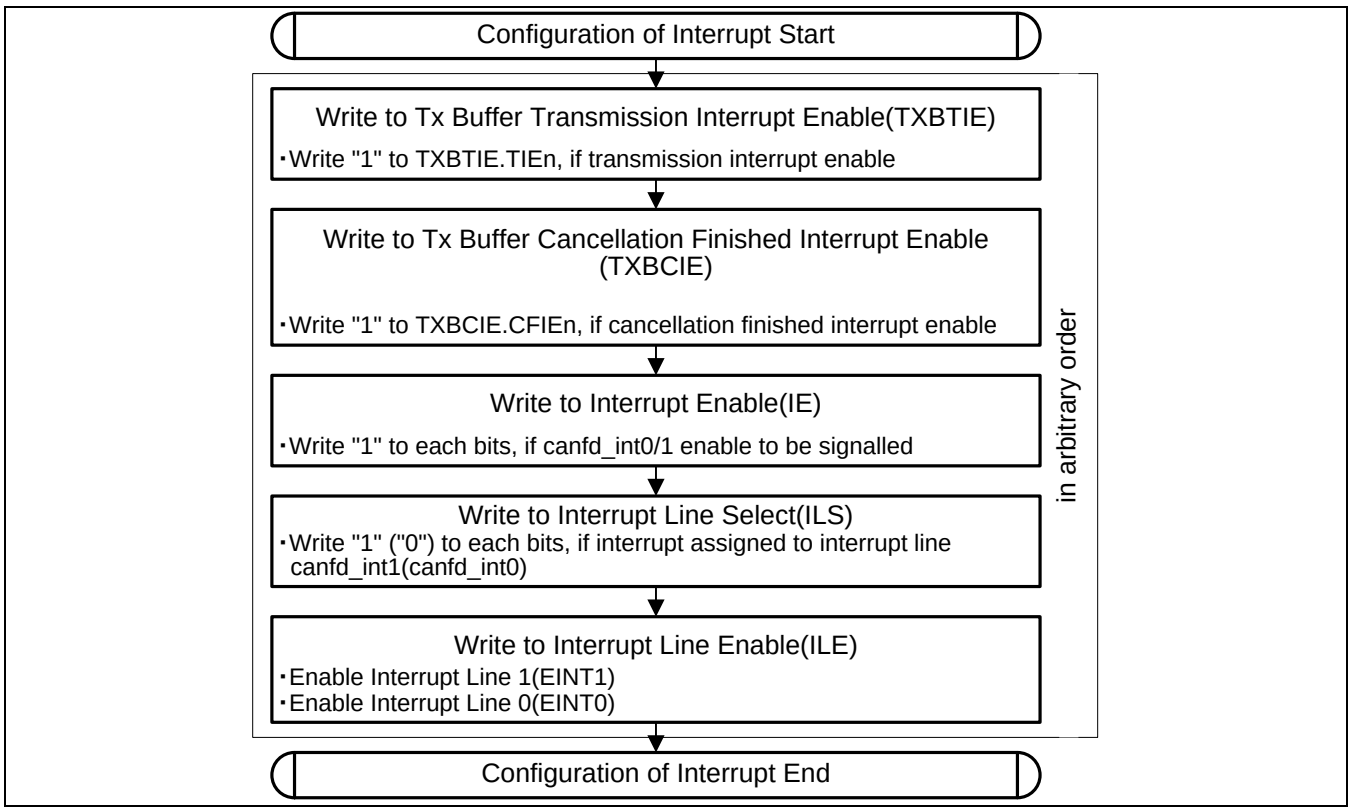
Figure 4-8 Flow Diagram "Configuration of Error Monitor"



4.4. Configuration of Interrupt

Figure 4-9 lists what to configure in order to set up the Interrupt related logic.

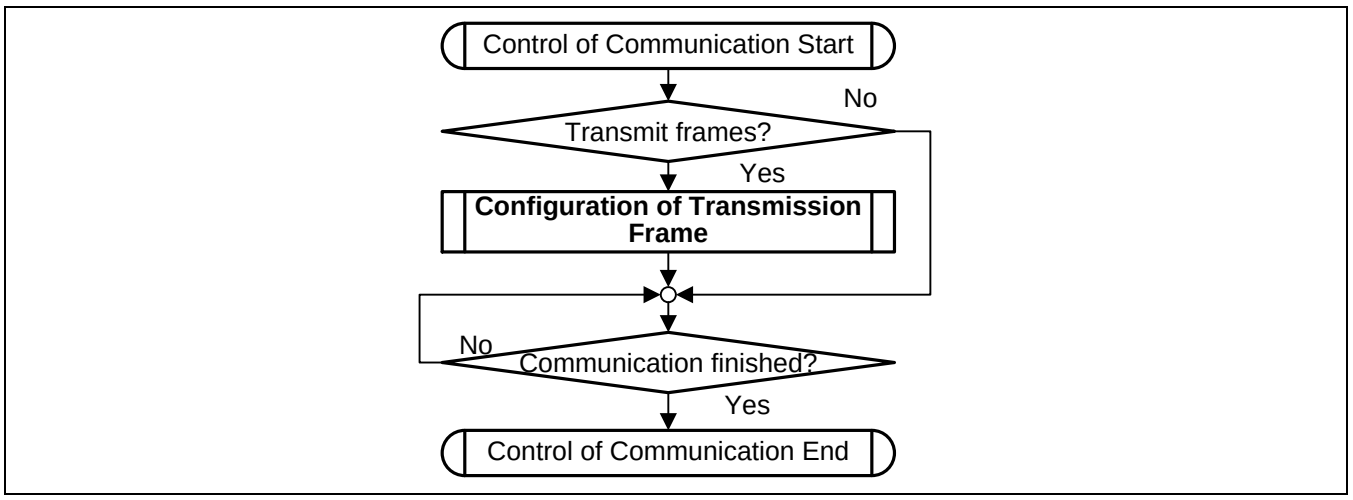
Figure 4-9 Flow Diagram "Configuration of Interrupt"



4.5. Control of Communication

Figure 4-10 shows the overall configuration procedure for communication on the CAN bus.

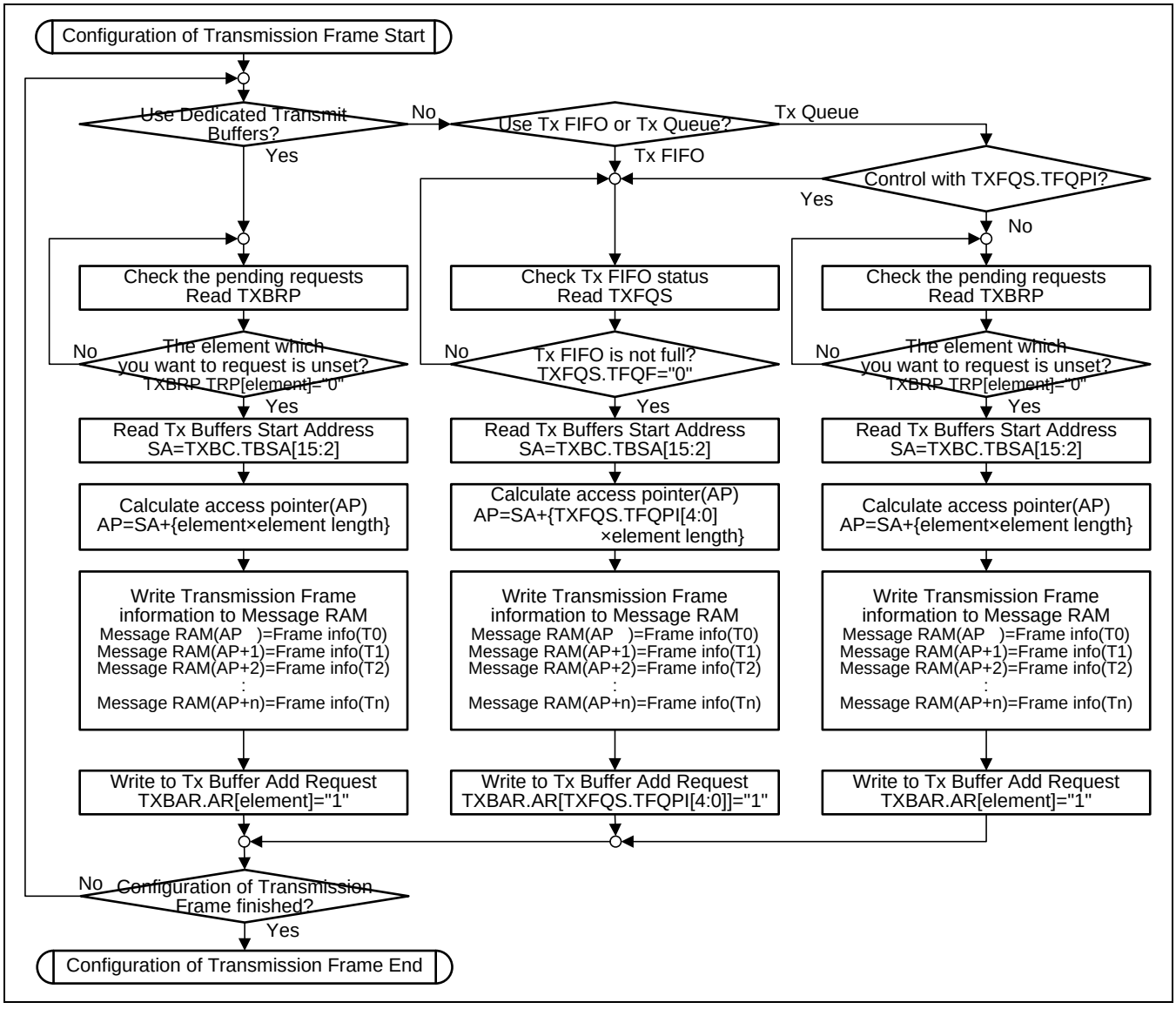
Figure 4-10 Flow Diagram "Control of Communication"



4.5.1. Configuration of Transmission Frame

Figure 4-11 shows how to set and transmit frames.

Figure 4-11 Flow Diagram "Configuration of Transmission Frame"



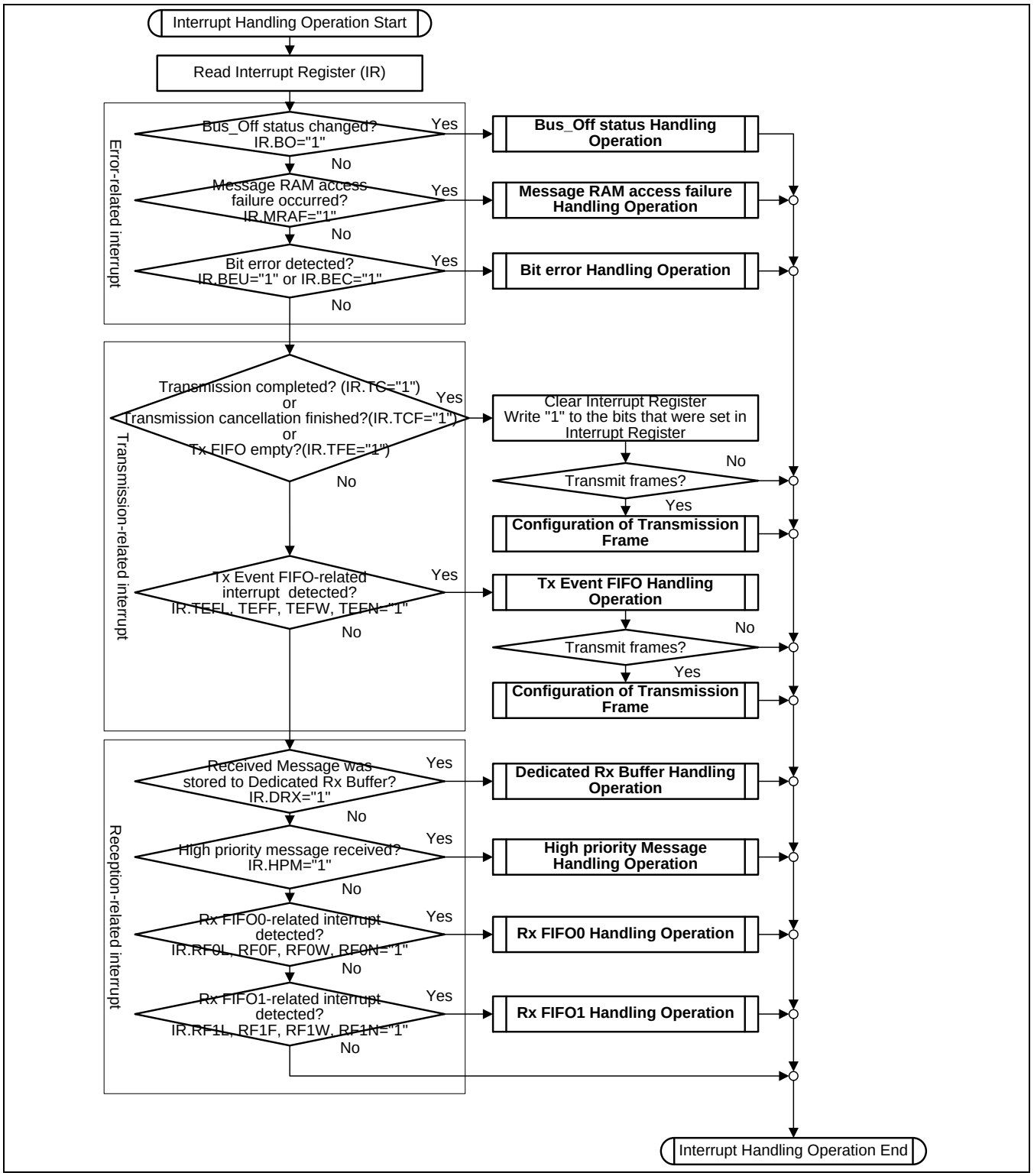
Notes:

- *Use 32-bit word address values when calculating the pointers in the flow diagram above.*
- *Dedicated Transmit Buffers are normally configured with a single specific identifier.*

4.6. Interrupt Handling Operation

Figure 4-12 shows the overall picture of how interrupts are processed.

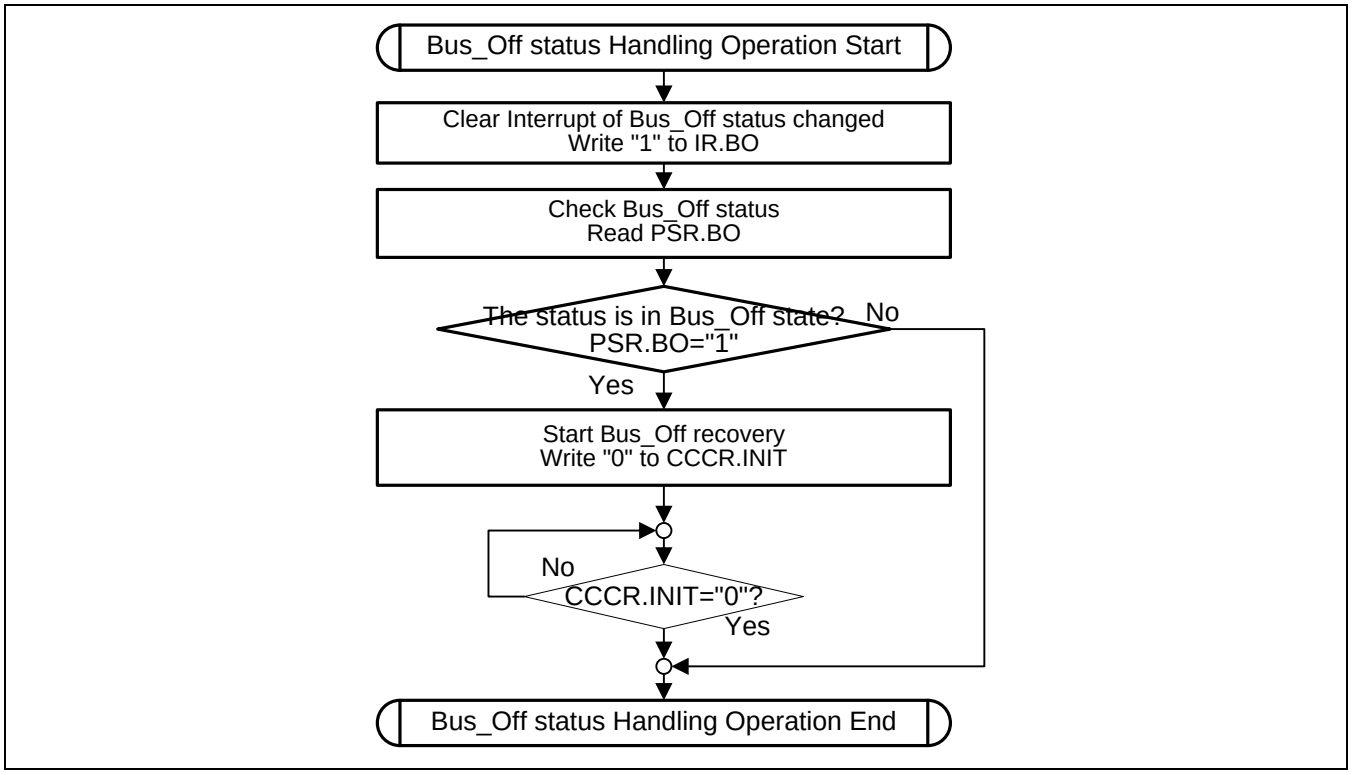
Figure 4-12 Flow Diagram "Interrupt Handling Operation"



4.6.1. Bus_Off Status Handling Operation

Figure 4-13 shows how to process a "Bus_Off Status" interrupt (IR.BO).

Figure 4-13 Flow Diagram "Bus_Off Status Handling Operation"



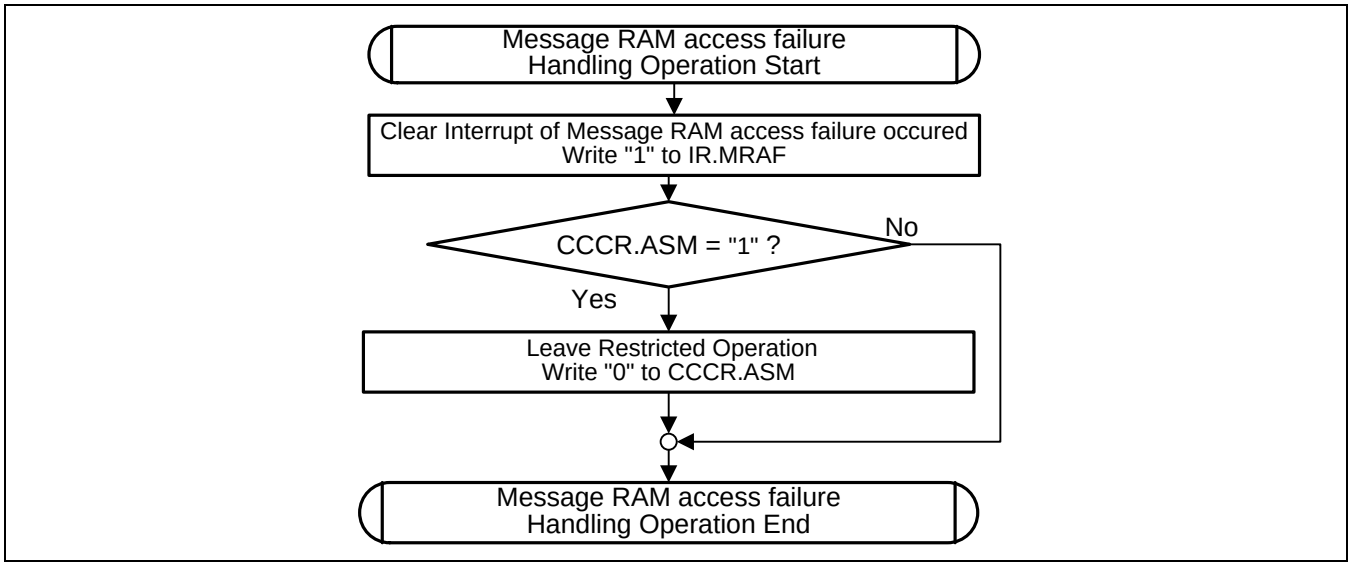
Note:

- *Once the Initialization bit (CCCR.INIT) has been cleared by the CPU, the CAN FD Controller will then wait for 129 occurrences of Bus Idle (129 × 11 consecutive recessive bits) before resuming normal operation.*

4.6.2. Message RAM Access Failure Handling Operation

Figure 4-14 shows how to process a "Message RAM Access Failure" interrupt (IR.MRAF).

Figure 4-14 Flow Diagram "Message RAM Access Failure Handling Operation"



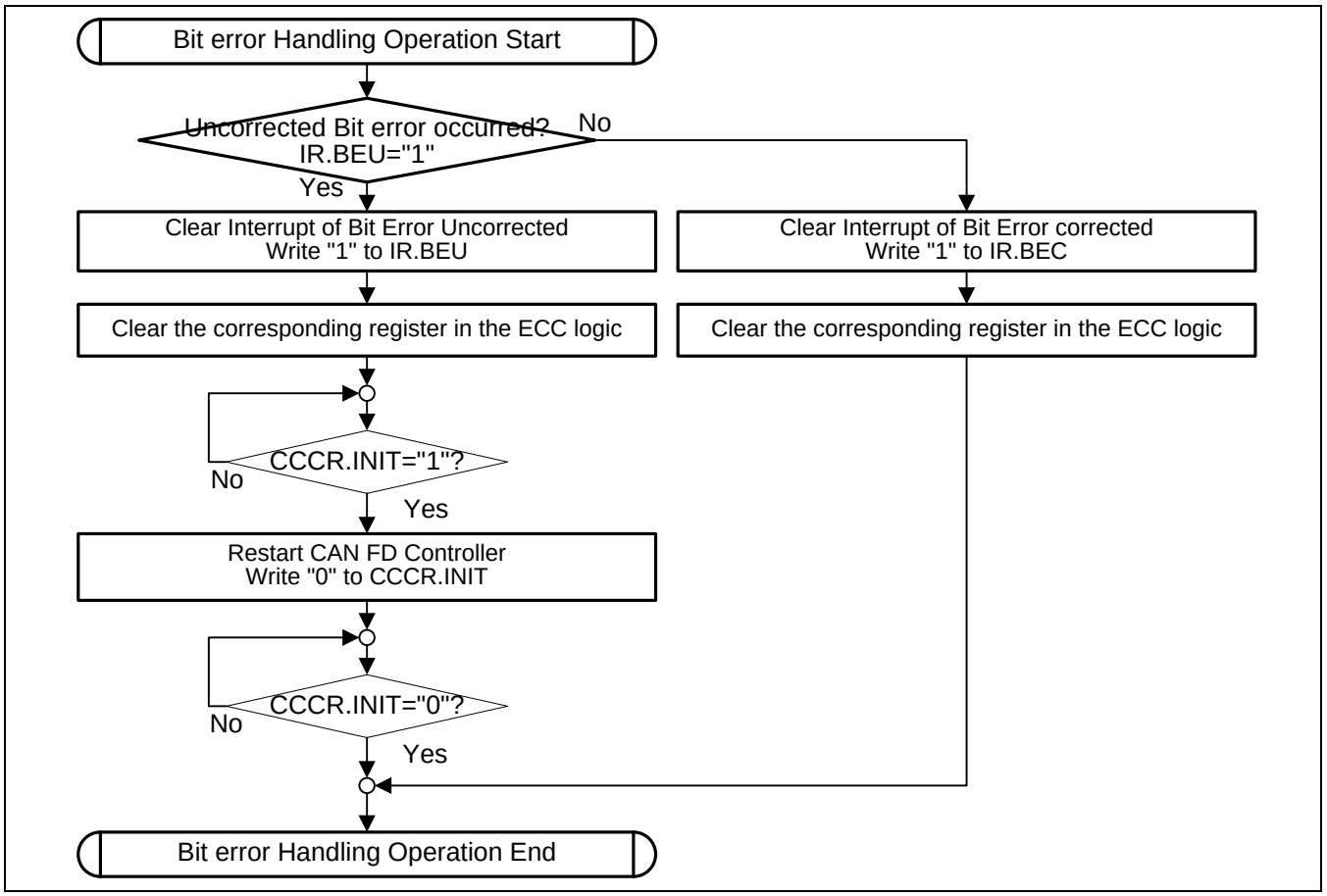
Note:

- *The data of a received frame may be incompletely stored to the Message RAM after Message RAM Access Failure occurs (IR.MRAF = "1") without setting Restricted Operation Mode (CCCR.ASM). In addition, Rx FIFO 0 Status (RXF0S), Rx FIFO 1 Status (RXF1S), or New Data 1/2 (NDAT1/2) is also updated according as the storage location (Rx FIFO0, Rx FIFO1, or an Rx Buffer). Therefore, in this case discard the received frame data.*

4.6.3. Bit Error Handling Operation

Figure 4-15 shows how to process the "Bit Error" interrupts (IR.BEU, IR.BEC).

Figure 4-15 Flow Diagram "Bit Error Handling Operation"



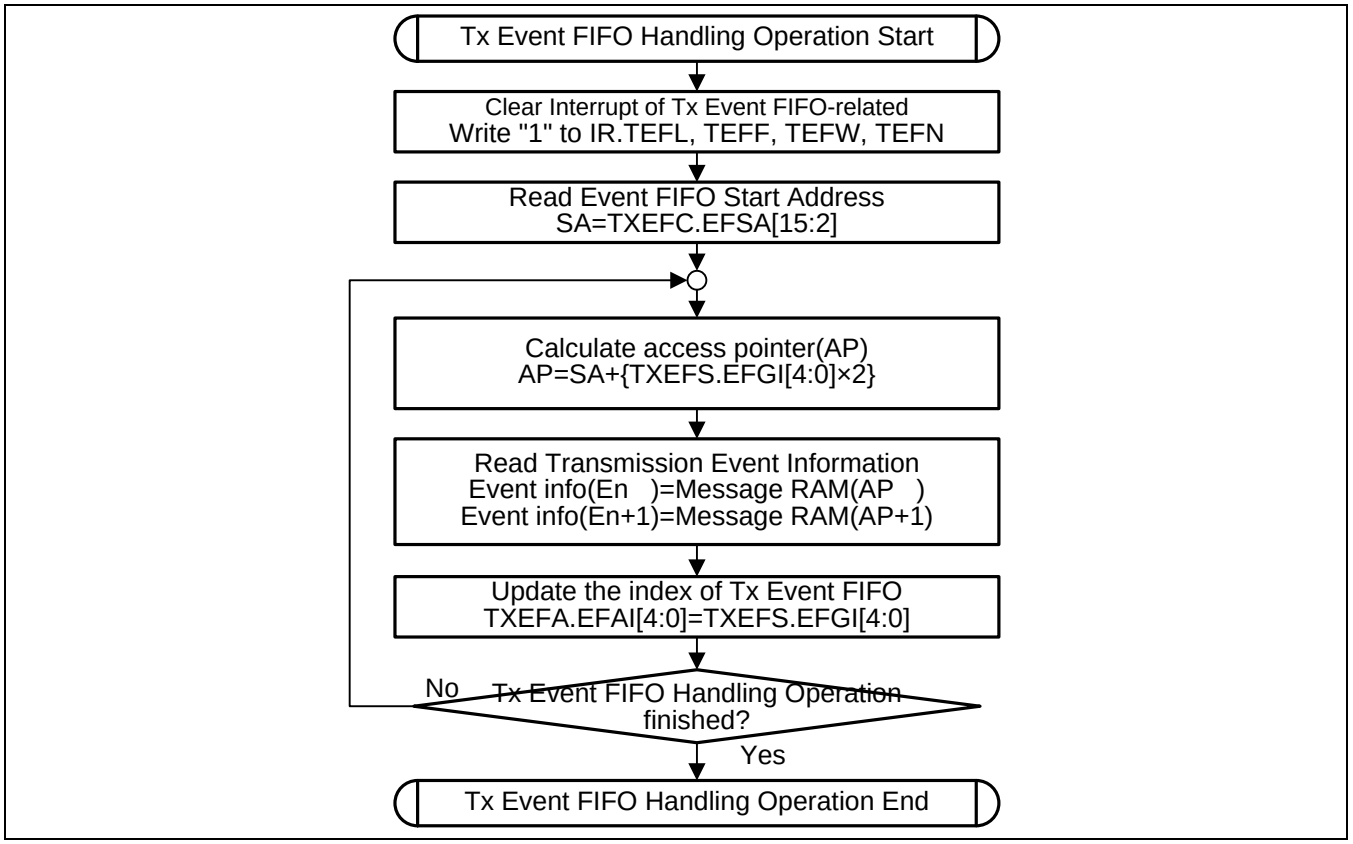
Note:

- *In case Bit Error Uncorrected IR.BEU occurs, due to the synchronization mechanism between the Bus clock and the CAN clock, there may be a delay until Initialization CCCR.INIT set to "1". Therefore the programmer has to assure that CCCR.INIT has been set to "1" by reading CCCR.INIT before resetting CCCR.INIT to "0".*

4.6.4. Tx Event FIFO Handling Operation

Figure 4-16 shows how to process Tx Event FIFO related interrupts.

Figure 4-16 Flow Diagram "Tx Event FIFO Handling Operation"



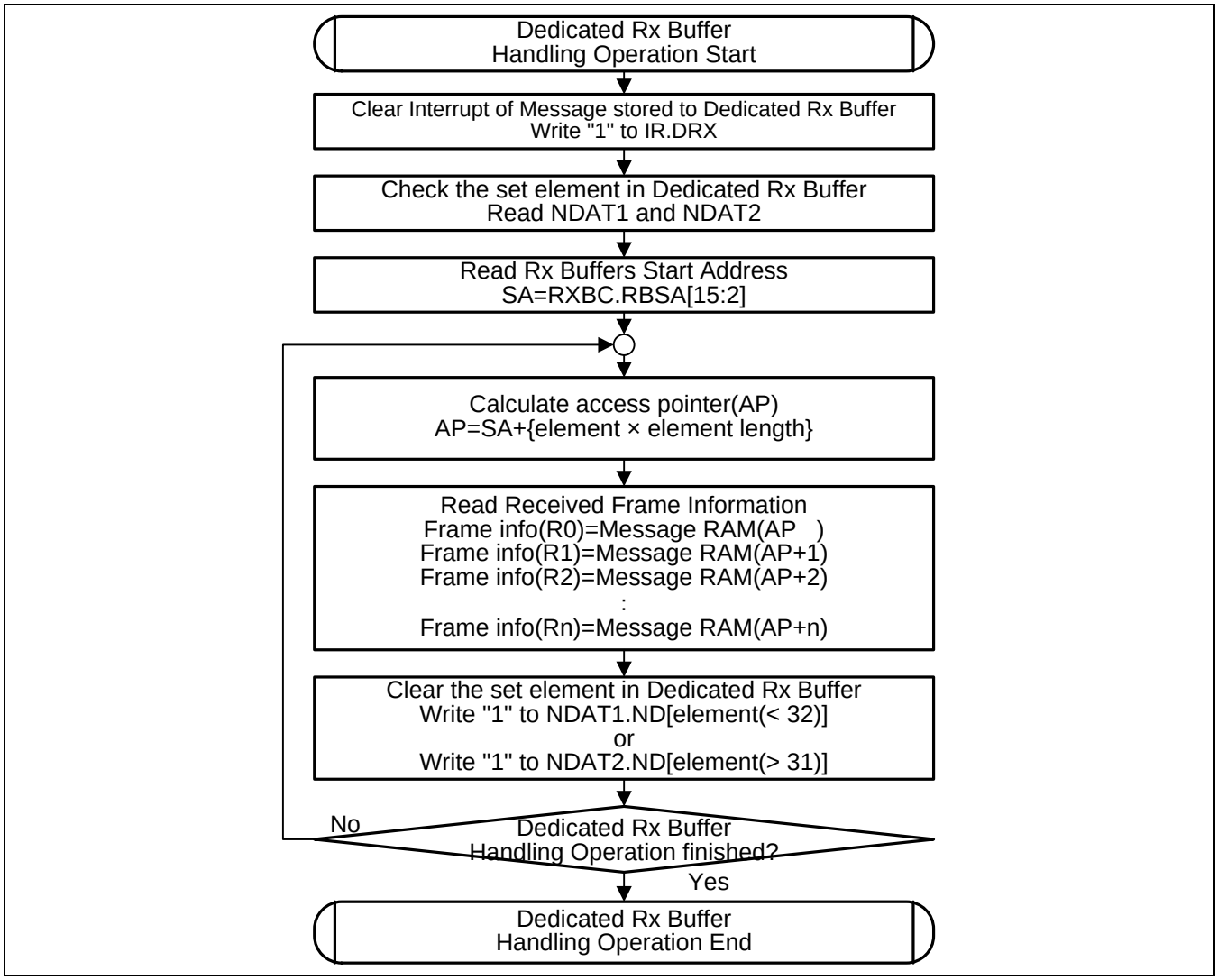
Note:

- *Use 32-bit word address values when calculating the pointers in the flow diagram above.*

4.6.5. Dedicated Rx Buffer Handling Operation

Figure 4-17 shows how to process a "Message stored to Dedicated Rx Buffer" interrupt (IR.DRX).

Figure 4-17 Flow Diagram "Dedicated Rx Buffer Handling Operation"



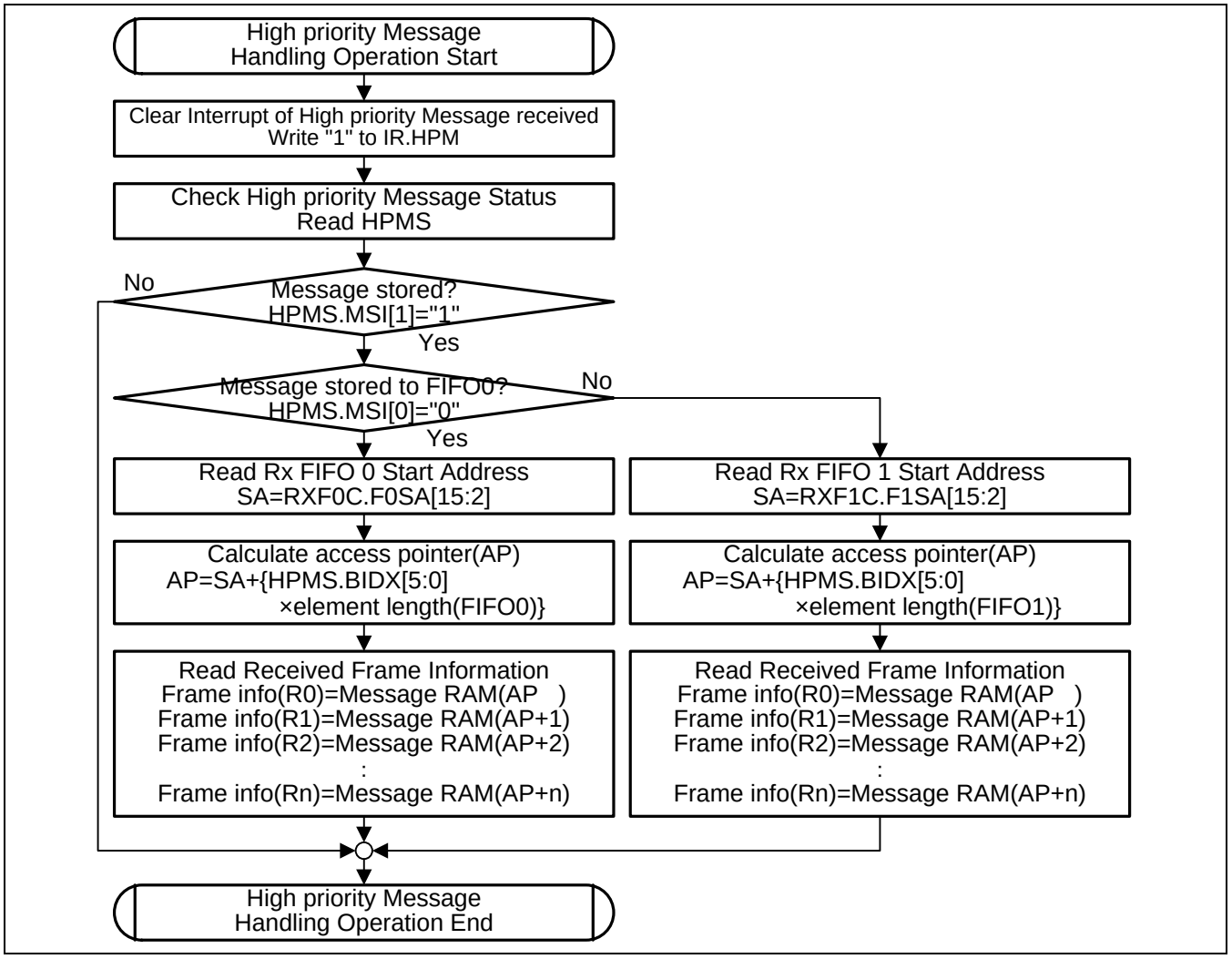
Note:

- *Use 32-bit word address values when calculating the pointers in the flow diagram above.*

4.6.6. High Priority Message Handling Operation

Figure 4-18 shows how to process a High Priority Message interrupt (IR.HPM).

Figure 4-18 Flow Diagram "High Priority Message Handling Operation"



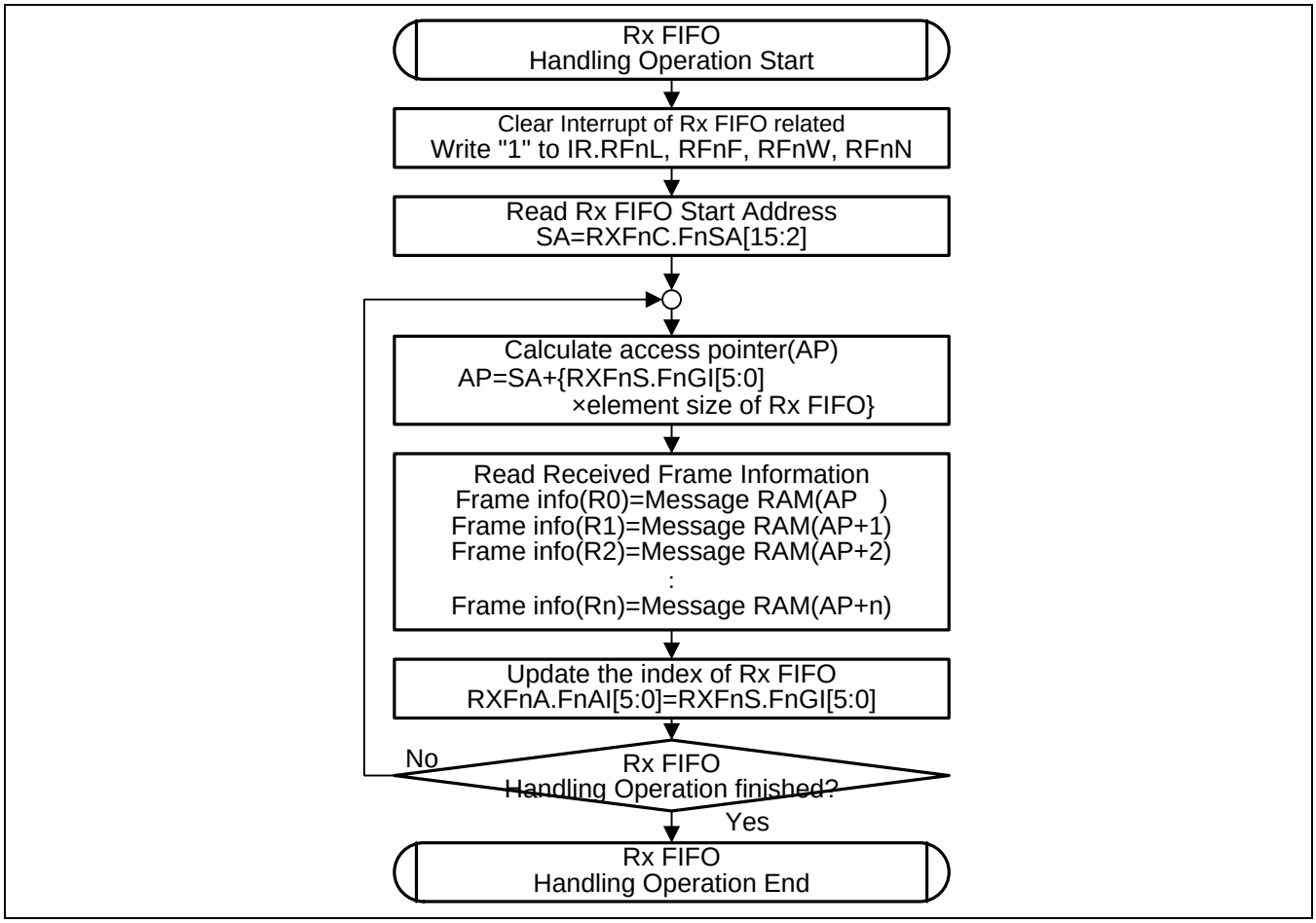
Note:

- *Use 32-bit word address values when calculating the pointers in the flow diagram above.*

4.6.7. Rx FIFO Handling Operation

Figure 4-19 shows how to process Rx FIFO related interrupts.

Figure 4-19 Flow Diagram "Rx FIFO Handling Operation"



Notes:

- *Use 32-bit word address values when calculating the pointers in the flow diagram above.*
- *The "n"s in the register/bit names (e.g. "RXFnS.FnGI") stand for "0" or "1".*
- *If a high priority message is received and processed by the "High priority Message Handling Operation" flow, this flow (i.e. Rx FIFO Handling Operation) may be skipped for that particular message. The following message received by the Rx FIFO may resume with this flow again.*

5. Registers

This section explains the configuration and functions of the registers used for the CAN FD Controller.

Hardware Reset Description

After hardware reset, the registers of the CAN FD Controller hold the reset values as indicated in each relative register description. Additionally the Bus_Off state is reset and the output canfd_tx is set to recessive. Setting the Initialization bit CCCR.INIT to "1" in the CC Control Register enables software initialization. The CAN FD Controller does not influence the CAN bus until the CPU resets CCCR.INIT to "0".

Register Map

The CAN FD Controller allocates an address space of 256 bytes for the registers.

All registers are organized as 32-bit registers. The registers are accessible by the CPU using a data width of 8 bit (byte access), 16 bit (half-word access), or 32 bit (word access).

Note:

- *As an exception, the Protocol Status Register (PSR) does not allow byte accessing.*

Write access by the CPU to several of the registers/bits that configure the basic operation of the CAN FD Controller is possible only with Configuration Change Enable CCCR.CCE="1" AND Initialization CCCR.INIT="1".

There is a delay from writing to a command register until the update of the related status register bits due to clock domain crossing.

Access to Reserved Register Addresses

In case the application software wants to access one of the reserved addresses in the CAN FD Controller register map (read or write access), interrupt flag Access to Reserved Address (IR.ARA) is set, and if enable the interrupt is signalled via the assigned interrupt line (canfd_int0/1). Table 5-1 below describes the reserved addresses, the offsets from the start address of the CAN FD Controller.

Table 5-1 Reserved Addresses in the CAN FD Controller

Offset	Number of Words
0x30-0x3C	4
0x4C	1
0x60-7C	8
0x8C	1
0xE8-EC	2
0xFC	1

Table 5-2 List of Registers for the CAN FD Controller

Abbreviation	Register Name	Reference
CREL	Core Release Register	5.1
ENDN	Endian Register	5.2
DBTP	Data Bit Timing & Prescaler Register	5.3
TEST	Test Register	5.4
RWD	RAM Watchdog	5.5

Abbreviation	Register Name	Reference
CCCR	CC Control Register	5.6
NBTP	Nominal Bit Timing & Prescaler Register	5.7
TSCC	Timestamp Counter Configuration	5.8
TSCV	Timestamp Counter Value	5.9
TOCC	Timeout Counter Configuration	5.10
TOCV	Timeout Counter Value	5.11
ECR	Error Counter Register	5.12
PSR	Protocol Status Register	5.13
TDCR	Transmitter Delay Compensation Register	5.14
IR	Interrupt Register	5.15
IE	Interrupt Enable	5.16
ILS	Interrupt Line Select	5.17
ILE	Interrupt Line Enable	5.18
GFC	Global Filter Configuration	5.19
SIDFC	Standard ID Filter Configuration	5.20
XIDFC	Extended ID Filter Configuration	5.21
XIDAM	Extended ID AND Mask	5.22
HPMS	High Priority Message Status	5.23
NDAT1	New Data 1	5.24
NDAT2	New Data 2	5.25
RXF0C	Rx FIFO 0 Configuration	5.26
RXF0S	Rx FIFO 0 Status	5.27
RXF0A	Rx FIFO 0 Acknowledge	5.28
RXBC	Rx Buffer Configuration	5.29
RXF1C	Rx FIFO 1 Configuration	5.30
RXF1S	Rx FIFO 1 Status	5.31
RXF1A	Rx FIFO 1 Acknowledge	5.32
RXESC	Rx Buffer/FIFO Element Size Configuration	5.33
TXBC	Tx Buffer Configuration	5.34
TXFQS	Tx FIFO/Queue Status	5.35
TXESC	Tx Buffer Element Size Configuration	5.36
TXBRP	Tx Buffer Request Pending	5.37
TXBAR	Tx Buffer Add Request	5.38
TXBCR	Tx Buffer Cancellation Request	5.39
TXBTO	Tx Buffer Transmission Occurred	5.40
TXBCF	Tx Buffer Cancellation Finished	5.41
TXBTIE	Tx Buffer Transmission Interrupt Enable	5.42
TXBCIE	Tx Buffer Cancellation Finished Interrupt Enable	5.43
TXEFC	Tx Event FIFO Configuration	5.44
TXEFS	Tx Event FIFO Status	5.45
TXEFA	Tx Event FIFO Acknowledge	5.46

Table 5-3 Register Map CAN FD CONTROLLER

MCU Config Group (Channel No.: 0, 1, and 2)

Offset	Register Name / Initial Value			
	+3	+2	+1	+0
0000_0000	MCG_CANFDx_CREL 00110010 00000100 00010010 00011000			
0000_0004	MCG_CANFDx_ENDN 10000111 01100101 01000011 00100001			
0000_0008	-			
0000_000C	MCG_CANFDx_DBTP 00000000 00000000 00001010 00110011			
0000_0010	MCG_CANFDx_TEST 00000000 00000000 00000000 *0000000			
0000_0014	MCG_CANFDx_RWD 00000000 00000000 00000000 00000000			
0000_0018	MCG_CANFDx_CCCR 00000000 00000000 00000000 00000001			
0000_001C	MCG_CANFDx_NBTP 00000110 00000000 00001010 00000011			
0000_0020	MCG_CANFDx_TSCC 00000000 00000000 00000000 00000000			
0000_0024	MCG_CANFDx_TSCV 00000000 00000000 00000000 00000000			
0000_0028	MCG_CANFDx_TOCC 11111111 11111111 00000000 00000000			
0000_002C	MCG_CANFDx_TOCV 00000000 00000000 11111111 11111111			
0000_0030 | 0000_003C	-			
0000_0040	MCG_CANFDx_ECR XXXXXXXX 00000000 00000000 00000000			
0000_0044	MCG_CANFDx_PSR XXXXXXXX X0000000 X0000111 00000111			
0000_0048	MCG_CANFDx_TDCR 00000000 00000000 00000000 00000000			
0000_004C	-			
0000_0050	MCG_CANFDx_IR 00000000 00000000 00000000 00000000			
0000_0054	MCG_CANFDx_IE 00000000 00000000 00000000 00000000			
0000_0058	MCG_CANFDx_ILS 00000000 00000000 00000000 00000000			
0000_005C	MCG_CANFDx_ILE 00000000 00000000 00000000 00000000			

Offset	Register Name / Initial Value			
	+3	+2	+1	+0
0000_0060 0000_007C	-			
0000_0080	MCG_CANFDx_GFC 00000000 00000000 00000000 00000000			
0000_0084	MCG_CANFDx_SIDFC 00000000 00000000 00000000 00000000			
0000_0088	MCG_CANFDx_XIDFC 00000000 00000000 00000000 00000000			
0000_008C	-			
0000_0090	MCG_CANFDx_XIDAM 00011111 11111111 11111111 11111111			
0000_0094	MCG_CANFDx_HPMS XXXXXXXX XXXXXXXX 00000000 00000000			
0000_0098	MCG_CANFDx_NDAT1 00000000 00000000 00000000 00000000			
0000_009C	MCG_CANFDx_NDAT2 00000000 00000000 00000000 00000000			
0000_00A0	MCG_CANFDx_RXF0C 00000000 00000000 00000000 00000000			
0000_00A4	MCG_CANFDx_RXF0S XXXXXX00 XX000000 XX000000 X0000000			
0000_00A8	MCG_CANFDx_RXF0A 00000000 00000000 00000000 00000000			
0000_00AC	MCG_CANFDx_RXBC 00000000 00000000 00000000 00000000			
0000_00B0	MCG_CANFDx_RXF1C 00000000 00000000 00000000 00000000			
0000_00B4	MCG_CANFDx_RXF1S 00XXXX00 XX000000 XX000000 X0000000			
0000_00B8	MCG_CANFDx_RXF1A 00000000 00000000 00000000 00000000			
0000_00BC	MCG_CANFDx_RXESC 00000000 00000000 00000000 00000000			
0000_00C0	MCG_CANFDx_TXBC 00000000 00000000 00000000 00000000			
0000_00C4	MCG_CANFDx_TXFQS XXXXXXXX XX000000 XXX00000 XX000000			
0000_00C8	MCG_CANFDx_TXESC 00000000 00000000 00000000 00000000			
0000_00CC	MCG_CANFDx_TXBRP 00000000 00000000 00000000 00000000			
0000_00D0	MCG_CANFDx_TXBAR 00000000 00000000 00000000 00000000			

Offset	Register Name / Initial Value			
	+3	+2	+1	+0
0000_00D4	MCG_CANFDx_TXBCR 00000000 00000000 00000000 00000000			
0000_00D8	MCG_CANFDx_TXBTO 00000000 00000000 00000000 00000000			
0000_00DC	MCG_CANFDx_TXBCF 00000000 00000000 00000000 00000000			
0000_00E0	MCG_CANFDx_TXBTIE 00000000 00000000 00000000 00000000			
0000_00E4	MCG_CANFDx_TXBCIE 00000000 00000000 00000000 00000000			
0000_00E8 0000_00EC	-			
0000_00F0	MCG_CANFDx_TXEFC 00000000 00000000 00000000 00000000			
0000_00F4	MCG_CANFDx_TXEFS XXXXXX00 XXX00000 XXX00000 XX000000			
0000_00F8	MCG_CANFDx_TXEFA 00000000 00000000 00000000 00000000			
0000_00FC 0000_01FC	-			
0000_0200	MCG_CANFDx_FDSEAR 00000000 00000000		MCG_CANFDx_FDESR 00000000	MCG_CANFDx_FDECR 00000000
0000_0204	MCG_CANFDx_FDDEAR 00000000 00000000		MCG_CANFDx_FDESCR 00000000	-
0000_0208 0000_02FC	-			
0000_0300 0000_7FFC	-			
0000_8000 0000_BFFC	CAN_RAM chx			
0000_C000 0000_FFFC	-			

Notes:

- *x is the channel number. (0 to 2)*
- *"*": Initial value "0" or "1" according to the setting (designed by higher layer)*
- *The number of circuits is designed by a higher layer.*

Common Peripheral #0 Group (Channel No.: 0, 1, 2, 3, and 4)

Offset	Register Name / Initial Value			
	+3	+2	+1	+0
0000_0000	CPG_CANFDx_CREL 00110010 00000100 00010010 00011000			
0000_0004	CPG_CANFDx_ENDN 10000111 01100101 01000011 00100001			
0000_0008	-			
0000_000C	CPG_CANFDx_DBTP 00000000 00000000 00001010 00110011			
0000_0010	CPG_CANFDx_TEST 00000000 00000000 00000000 *0000000			
0000_0014	CPG_CANFDx_RWD 00000000 00000000 00000000 00000000			
0000_0018	CPG_CANFDx_CCCR 00000000 00000000 00000000 00000001			
0000_001C	CPG_CANFDx_NBTP 00000110 00000000 00001010 00000011			
0000_0020	CPG_CANFDx_TSCC 00000000 00000000 00000000 00000000			
0000_0024	CPG_CANFDx_TSCV 00000000 00000000 00000000 00000000			
0000_0028	CPG_CANFDx_TOCC 11111111 11111111 00000000 00000000			
0000_002C	CPG_CANFDx_TOCV 00000000 00000000 11111111 11111111			
0000_0030 | 0000_003C	-			
0000_0040	CPG_CANFDx_ECR XXXXXXXX 00000000 00000000 00000000			
0000_0044	CPG_CANFDx_PSR XXXXXXXX X0000000 X0000111 00000111			
0000_0048	CPG_CANFDx_TDCR 00000000 00000000 00000000 00000000			
0000_004C	-			
0000_0050	CPG_CANFDx_IR 00000000 00000000 00000000 00000000			
0000_0054	CPG_CANFDx_IE 00000000 00000000 00000000 00000000			
0000_0058	CPG_CANFDx_ILS 00000000 00000000 00000000 00000000			
0000_005C	CPG_CANFDx_ILE 00000000 00000000 00000000 00000000			

Offset	Register Name / Initial Value			
	+3	+2	+1	+0
0000_0060 0000_007C	-			
0000_0080	CPG_CANFDx_GFC 00000000 00000000 00000000 00000000			
0000_0084	CPG_CANFDx_SIDFC 00000000 00000000 00000000 00000000			
0000_0088	CPG_CANFDx_XIDFC 00000000 00000000 00000000 00000000			
0000_008C	-			
0000_0090	CPG_CANFDx_XIDAM 00011111 11111111 11111111 11111111			
0000_0094	CPG_CANFDx_HPMS XXXXXXXX XXXXXXXX 00000000 00000000			
0000_0098	CPG_CANFDx_NDAT1 00000000 00000000 00000000 00000000			
0000_009C	CPG_CANFDx_NDAT2 00000000 00000000 00000000 00000000			
0000_00A0	CPG_CANFDx_RXF0C 00000000 00000000 00000000 00000000			
0000_00A4	CPG_CANFDx_RXF0S XXXXXX00 XX000000 XX000000 X0000000			
0000_00A8	CPG_CANFDx_RXF0A 00000000 00000000 00000000 00000000			
0000_00AC	CPG_CANFDx_RXBC 00000000 00000000 00000000 00000000			
0000_00B0	CPG_CANFDx_RXF1C 00000000 00000000 00000000 00000000			
0000_00B4	CPG_CANFDx_RXF1S 00XXXX00 XX000000 XX000000 X0000000			
0000_00B8	CPG_CANFDx_RXF1A 00000000 00000000 00000000 00000000			
0000_00BC	CPG_CANFDx_RXESC 00000000 00000000 00000000 00000000			
0000_00C0	CPG_CANFDx_TXBC 00000000 00000000 00000000 00000000			
0000_00C4	CPG_CANFDx_TXFQS XXXXXXXX XX000000 XXX00000 XX000000			
0000_00C8	CPG_CANFDx_TXESC 00000000 00000000 00000000 00000000			
0000_00CC	CPG_CANFDx_TXBRP 00000000 00000000 00000000 00000000			
0000_00D0	CPG_CANFDx_TXBAR 00000000 00000000 00000000 00000000			

Offset	Register Name / Initial Value			
	+3	+2	+1	+0
0000_00D4	CPG_CANFDx_TXBCR 00000000 00000000 00000000 00000000			
0000_00D8	CPG_CANFDx_TXBTO 00000000 00000000 00000000 00000000			
0000_00DC	CPG_CANFDx_TXBCF 00000000 00000000 00000000 00000000			
0000_00E0	CPG_CANFDx_TXBTIE 00000000 00000000 00000000 00000000			
0000_00E4	CPG_CANFDx_TXBCIE 00000000 00000000 00000000 00000000			
0000_00E8 0000_00EC	-			
0000_00F0	CPG_CANFDx_TXEFC 00000000 00000000 00000000 00000000			
0000_00F4	CPG_CANFDx_TXEFS XXXXXX00 XXX00000 XXX00000 XX000000			
0000_00F8	CPG_CANFDx_TXEFA 00000000 00000000 00000000 00000000			
0000_00FC 0000_01FC	-			
0000_0200	CPG_CANFDx_FDSEAR 00000000 00000000		CPG_CANFDx_FDESR 00000000	CPG_CANFDx_FDECR 00000000
0000_0204	CPG_CANFDx_FDDEAR 00000000 00000000		CPG_CANFDx_FDESCR 00000000	-
0000_0208 0000_02FC	-			
0000_0300 0000_7FFC	-			
0000_8000 0000_BFFC	CAN_RAM chx			
0000_C000 0000_FFFC	-			

Notes:

- *x is the channel number. (0 to 4)*
- *"*": Initial value "0" or "1" according to the setting (designed by higher layer)*
- *The number of circuits is designed by a higher layer.*

Table 5-4 Register Mirror Area CAN FD CONTROLLER

Address of Register Mirror Area	Mirrored Peripheral	Mirror Area Behavior	PPU
0000_C000 to 0000_FFFF	MCU_CONFIG CAN FD ch.0	Accessing this region results to the access to MCU_CONFIG CAN FD ch.0 Message RAM area (0000_0000 to 0000_BFFF)	This area is covered by PPU #47 as well as the mirrored peripheral.
0000_C000 to 0000_FFFF	MCU_CONFIG CAN FD ch.1	Accessing this region results to the access to MCU_CONFIG CAN FD ch.1 Message RAM area (0000_0000 to 0000_BFFF)	This area is covered by PPU #48 as well as the mirrored peripheral.
0000_C000 to 0000_FFFF	MCU_CONFIG CAN FD ch.2	Accessing this region results to the access to MCU_CONFIG CAN FD ch.2 Message RAM area (0000_0000 to 0000_BFFF)	This area is covered by PPU #49 as well as the mirrored peripheral.
0000_C000 to 0000_FFFF	CommonPERI#0 CAN FD ch.0	Accessing this region results to the access to CommonPERI#0 CAN FD ch.0 Message RAM area (0000_0000 to 0000_BFFF)	This area is covered by PPU #256 as well as the mirrored peripheral.
0000_C000 to 0000_FFFF	CommonPERI#0 CAN FD ch.1	Accessing this region results to the access to CommonPERI#0 CAN FD ch.1 Message RAM area (0000_0000 to 0000_BFFF)	This area is covered by PPU #257 as well as the mirrored peripheral.
0000_C000 to 0000_FFFF	CommonPERI#0 CAN FD ch.2	Accessing this region results to the access to CommonPERI#0 CAN FD ch.2 Message RAM area (0000_0000 to 0000_BFFF)	This area is covered by PPU #258 as well as the mirrored peripheral.
0000_C000 to 0000_FFFF	CommonPERI#0 CAN FD ch.3	Accessing this region results to the access to CommonPERI#0 CAN FD ch.3 Message RAM area (0000_0000 to 0000_BFFF)	This area is covered by PPU #259 as well as the mirrored peripheral.
0000_C000 to 0000_FFFF	CommonPERI#0 CAN FD ch.4	Accessing this region results to the access to CommonPERI#0 CAN FD ch.4 Message RAM area (0000_0000 to 0000_BFFF)	This area is covered by PPU #260 as well as the mirrored peripheral.

5.1. Core Release Register (MCG_CANFDx_CREL, CPG_CANFDx_CREL)

The Core Release Register displays the revision of the CAN FD Controller.

bit	31	30	29	28	27	26	25	24
Field	REL[3:0]				STEP[3:0]			
R/W Attribute	R,WX				R,WX			
Protection Attribute	-				-			
Initial value	0x3				0x2			

bit	23	22	21	20	19	18	17	16
Field	SUBSTEP[3:0]				YEAR[3:0]			
R/W Attribute	R,WX				R,WX			
Protection Attribute	-				-			
Initial value	0x0				0x4			

bit	15	14	13	12	11	10	9	8
Field	MON[7:0]							
R/W Attribute	R,WX							
Protection Attribute	-							
Initial value	0x12							

bit	7	6	5	4	3	2	1	0
Field	DAY[7:0]							
R/W Attribute	R,WX							
Protection Attribute	-							
Initial value	0x18							

[bit31:28] REL[3:0]: Core Release

One digit, BCD-coded.

[bit27:24] STEP[3:0]: Step of Core Release

One digit, BCD-coded.

[bit23:20] SUBSTEP[3:0]: Sub-Step of Core Release

One digit, BCD-coded.

[bit19:16] YEAR[3:0]: Time Stamp Year

One digit, BCD-coded.

[bit15:8] MON[7:0]: Time Stamp Month

Two digits, BCD-coded.

[bit7:0] DAY[7:0]: Time Stamp Day

Two digits, BCD-coded.

Table 5-5 Example for Coding of Revisions

Release	Step	SubStep	Year	Month	Day	Name
3	2	0	4	12	18	Revision 3.2.0, Date 2014/12/18

5.2. Endian Register (MCG_CANFDx_ENDN, CPG_CANFDx_ENDN)

The Endian Register can be used to check the endianness of the CAN FD Controller when accessed by the CPU.

bit	31	30	29	28	27	26	25	24
Field	ETV[31:24]							
R/W Attribute	R,WX							
Protection Attribute	-							
Initial value	0x87							

bit	23	22	21	20	19	18	17	16
Field	ETV[23:16]							
R/W Attribute	R,WX							
Protection Attribute	-							
Initial value	0x65							

bit	15	14	13	12	11	10	9	8
Field	ETV[15:8]							
R/W Attribute	R,WX							
Protection Attribute	-							
Initial value	0x43							

bit	7	6	5	4	3	2	1	0
Field	ETV[7:0]							
R/W Attribute	R,WX							
Protection Attribute	-							
Initial value	0x21							

[bit31:0] ETV[31:0]: Endianness Test Value

The endianness test value is 0x87654321.

5.3. Data Bit Timing & Prescaler Register (MCG_CANFDx_DBTP, CPG_CANFDx_DBTP)

The Data Bit Timing & Prescaler Register configures the data bit time and enables Transmitter Delay Compensation.

This register is only writable if bits Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) are set.

See "3.1.4.Transmitter Delay Compensation" for a description of TDC fields.

See "3.7.Configuring the CAN Bit Timing" for a description of all other fields.

Notes:

- *With a CAN clock (canfd_cclk) of 8 MHz, the reset value of 0x00000A33 configures the CAN FD Controller for a data phase bit rate of 500 kBit/s.*
- *The bit rate configured for the CAN FD data phase via the Data Bit Timing & Prescaler Register (DBTP) must be higher or equal to the bit rate configured for the arbitration phase via the Nominal Bit Timing & Prescaler Register (NBTP).*

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

bit	23	22	21	20	19	18	17	16
Field	TDC	Reserved		DBRP[4:0]				
R/W Attribute	R/W	R0,W0		R/W				
Protection Attribute	-	-		-				
Initial value	0	00		00000				

Bit	15	14	13	12	11	10	9	8
Field	Reserved			DTSEG1[4:0]				
R/W Attribute	R0,W0			R/W				
Protection Attribute	-			-				
Initial value	000			0x0A				

bit	7	6	5	4	3	2	1	0
Field	DTSEG2[3:0]				DSJW[3:0]			
R/W Attribute	R/W				R/W			
Protection Attribute	-				-			
Initial value	0x3				0x3			

[bit31:24] Reserved: Reserved Bits

When writing, always write "0". When reading, "0" is always read.

[bit23] TDC: Transmitter Delay Compensation

Bit	Description
0	Transmitter Delay Compensation disabled.
1	Transmitter Delay Compensation enabled.

[bit22:21] Reserved: Reserved Bits

When writing, always write "0". When reading, "0" is always read.

[bit20:16] DBRP[4:0]: Data Bit Rate Prescaler

DBRP[4:0]	Description
0x00-0x1F	The value by which the oscillator frequency is divided for generating the data bit time quanta. The data bit time is built up from a multiple of this quanta. Valid values for the Data Bit Rate Prescaler are 0 to 31. When DBTP.TDC = "1", the range is limited to 0, 1. This limitation is required by the ISO. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used.

[bit15:13] Reserved: Reserved Bits

When writing, always write "0". When reading, "0" is always read.

[bit12:8] DTSEG1[4:0]: Data Time Segment before sample Point

DTSEG1[4:0]	Description
0x00-0x1F	Valid values are 0 to 31. The actual interpretation by the hardware of this value is such that one more than the programmed value is used.

[bit7:4] DTSEG2[3:0]: Data Time Segment after sample Point

DTSEG2[3:0]	Description
0x0-0xF	Valid values are 0 to 15. The actual interpretation by the hardware of this value is such that one more than the programmed value is used.

[bit3:0] DSJW[3:0]: Data (Re) Synchronization Jump Width

DSJW[3:0]	Description
0x0-0xF	Valid values are 0 to 15. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used.

5.4. Test Register (MCG_CANFDx_TEST, CPG_CANFDx_TEST)

The Test Register monitors the canfd_rx/canfd_tx pins. It is also used to enable the Loop Back Modes.

Write access to the Test Register has to be enabled by setting the Test Mode Enable bit CCCR.TEST to "1".

All Test Register functions are set to their reset values when bit CCCR.TEST is reset.

Loop Back Mode and software control of pin canfd_tx are hardware test modes. Programming of TX ≠ "00" may disturb the message transfer on the CAN bus.

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

bit	23	22	21	20	19	18	17	16
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

bit	15	14	13	12	11	10	9	8
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

bit	7	6	5	4	3	2	1	0
Field	RX	TX[1:0]		LBCK	Reserved			
R/W Attribute	R,WX	R/W		R/W	R0,W0			
Protection Attribute	-	-		-	-			
Initial value	U	00		0	0000			

U = undefined. The RX bit will reflect the actual level at pin canfd_rx.

[bit31:8] Reserved: Reserved Bits

When writing, always write "0". When reading, "0" is always read.

[bit7] RX: Receive Pin

Monitors the actual value of pin canfd_rx

Bit	Description
0	The CAN bus is dominant (dominant level at pin canfd_rx).
1	The CAN bus is recessive (recessive level at pin canfd_rx).

[bit6:5] TX[1:0]: Control of Transmit Pin

TX[1:0]	Description
00	Reset value. canfd_tx controlled by the CAN Core, updated at the end of the bit time.
01	Sample Point can be monitored at pin canfd_tx.
10	Dominant level at pin canfd_tx.
11	Recessive level at pin canfd_tx.

[bit4] LBCK: Loop Back Mode

Bit	Description
0	Reset value, Loop Back Mode is disabled.
1	Loop Back Mode is enabled.

[bit3:0] Reserved: Reserved Bits

When writing, always write "0". When reading, "0" is always read.

5.5. RAM Watchdog (MCG_CANFDx_RWD, CPG_CANFDx_RWD)

The RAM Watchdog monitors the Message RAM to see if it is ready to be accessed or not.

A Message RAM access starts the RAM Watchdog Counter with the value configured by the Watchdog Configuration (RWD.WDC[7:0]). The counter is reloaded with RWD.WDC when a successful access to the Message RAM has been completed. In case there is no response from the Message RAM until the counter has counted down to zero, the counter stops and interrupt flag Watchdog Interrupt (IR.WDI) is set.

The RAM Watchdog Counter is clocked by the Bus clock (canfd_bclk).

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

bit	23	22	21	20	19	18	17	16
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

bit	15	14	13	12	11	10	9	8
Field	WDV[7:0]							
R/W Attribute	R,WX							
Protection Attribute	-							
Initial value	0x00							

bit	7	6	5	4	3	2	1	0
Field	WDC[7:0]							
R/W Attribute	R/W							
Protection Attribute	-							
Initial value	0x00							

[bit31:16] Reserved: Reserved Bits

When writing, always write "0". When reading, "0" is always read.

[bit15:8] WDV[7:0]: Watchdog Value

WDV[7:0]	Description
0x00-0xFF	Actual Message RAM Watchdog Counter Value

[bit7:0] WDC[7:0]: Watchdog Configuration

Start value of the Message RAM Watchdog Counter. With the reset value of 0x00 the counter is disabled.

Write access to this field is possible only when the Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) bits are both "1".

5.6. CC Control Register (MCG_CANFDx_CCCR, CPG_CANFDx_CCCR)

The CC Control Register configures various operating modes of the CAN FD Controller.

For details about setting and resetting of single bits see "3.1.1. Software Initialization".

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

bit	23	22	21	20	19	18	17	16
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

bit	15	14	13	12	11	10	9	8
Field	NISO	TXP	EFBI	PXHD	Reserved		BRSE	FDOE
R/W Attribute	R/W	R/W	R/W	R/W	R0,W0		R/W	R/W
Protection Attribute	-	-	-	-	-		-	-
Initial value	0	0	0	0	00		0	0

bit	7	6	5	4	3	2	1	0
Field	TEST	DAR	MON	CSR	CSA	ASM	CCE	INIT
R/W Attribute	R/W	R/W	R/W	R/W	R,WX	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	1

[bit31:16] Reserved: Reserved Bits

When writing, always write "0". When reading, "0" is always read.

[bit15] NISO: Non ISO Operation

If this bit is set, the CAN FD Controller uses the CAN FD frame format as specified by Bosch CAN FD Specification V1.0.

Write access to this bit is possible only when the Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) bits are both "1".

Bit	Description
0	CAN FD frame format according to ISO 11898-1.
1	CAN FD frame format according to Bosch CAN FD Specification V1.0.

[bit14] TXP: Transmit Pause

If this bit is set, the CAN FD Controller pauses for two nominal bit times before starting the next transmission after it has successfully transmitted a frame.

Write access to this bit is possible only when the Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) bits are both "1".

Bit	Description
0	Transmit pause disabled.
1	Transmit pause enabled.

[bit13] EFBI: Edge Filtering during Bus Integration

Write access to this bit is possible only when the Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) bits are both "1".

Bit	Description
0	Edge filtering disabled.
1	Two consecutive dominant tq required to detect an edge for hard synchronization.

Note:

- *In case that all of the following conditions are fulfilled, it may happen that the CAN FD Controller synchronizes itself wrongly. And it will rate the received CAN FD frame as faulty and error frame will be send. When that frame is retransmitted, Bus integration has finished and the frame is received correctly.*
  - *when edge filtering is activated and*
  - *when the end of Bus integration coincides with a falling edge (or dominant) at pin canfd_rx, and a CAN FD frame reception is started*

[bit12] PXHD: Protocol Exception Handling Disable

Write access to this bit is possible only when the Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) bits are both "1".

Bit	Description
0	Protocol exception handling enabled.
1	Protocol exception handling disabled.

Note:

- *When protocol exception handling is disabled, the CAN FD Controller will transmit an error frame when it detects a protocol exception condition.*

[bit11:10] Reserved: Reserved Bits

When writing, always write "0". When reading, "0" is always read.

[bit9] BRSE: Bit Rate Switch Enable

Write access to this bit is possible only when the Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) bits are both "1".

Bit	Description
0	Bit rate switching for transmissions disabled.
1	Bit rate switching for transmissions enabled.

Note:

- *When CAN FD operation is disabled CCCR.FDOE = "0", CCCR.BRSE is not evaluated.*

[bit8] FDOE: FD Operation Enable

Write access to this bit is possible only when the Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) bits are both "1".

Bit	Description
0	FD operation disabled.
1	FD operation enabled.

[bit7] TEST: Test Mode Enable

Bit TEST can only be set by the CPU when the Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) bits are both "1".

The bit can be reset by the CPU at any time.

Bit	Description
0	Normal operation, register TEST holds reset values.
1	Test Mode, write access to register TEST enabled.

Note:

- *When CCCR.DAR is set "1", always set "00" to first two identifier (Arbitration field) bits on transmission frame configuration.*

[bit6] DAR: Disable Automatic Retransmission

Write access to this bit is possible only when the Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) bits are both "1". For a description of DAR mode, see "3.1.7. Disabled Automatic Retransmission".

Bit	Description
0	Automatic retransmission of messages not transmitted successfully enabled.
1	Automatic retransmission disabled.

[bit5] MON: Bus Monitoring Mode

Bit MON can only be set by the CPU when the Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) bits are both "1".

The bit can be reset by the CPU at any time.

Bit	Description
0	Bus Monitoring Mode is disabled.
1	Bus Monitoring Mode is enabled.

[bit4] CSR: Clock Stop Request

For a description of Power Down, see "3.1.8. Power Down (Sleep Mode)".

Bit	Description
0	No clock stop is requested.
1	Clock stop requested. When clock stop is requested, first CCCR.INIT and then CCCR.CSA will be set after all pending transfer requests have been completed and the CAN bus reached idle.

[bit3] CSA: Clock Stop Acknowledge

For a description of Power Down, see "3.1.8. Power Down (Sleep Mode)".

Bit	Description
0	No clock stop acknowledged.
1	CAN FD Controller may be set in power down by stopping canfd_bclk and canfd_cclk.

[bit2] ASM: Restricted Operation Mode

Bit ASM can only be set by the CPU when the Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) bits are both "1"

The bit can be reset by the CPU at any time. For a description of the Restricted Operation Mode see "3.1.5. Restricted Operation Mode".

Bit	Description
0	Normal CAN operation.
1	Restricted Operation Mode active.

[bit1] CCE: Configuration Change Enable

Write access to this bit is possible only when the Initialization (CCCR.INIT) bit is "1". This bit (CCCR.CCE) is reset to "0" when CCCR.INIT is set to "0".

Bit	Description
0	The CPU has no write access to the protected configuration registers.
1	The CPU has write access to the protected configuration registers (while bit Initialization CCCR.INIT = "1").

[bit0] INIT: Initialization

Bit	Description
0	Normal Operation.
1	Initialization is started.

Note:

- *Due to the synchronization mechanism between the two clock domains, there may be a delay until the value written to Initialization CCCR.INIT can be read back (the maximum length of the delay is four Bus clocks plus five CAN clocks). Therefore the programmer has to assure that the previous value written to CCCR.INIT has been accepted by reading CCCR.INIT before setting CCCR.INIT to a new value.*

5.7. Nominal Bit Timing & Prescaler Register (MCG_CANFDx_NBTP, CPG_CANFDx_NBTP)

The Nominal Bit Timing & Prescaler Register configures the nominal bit time of the CAN FD Controller.

This register is only writable if bits Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) are set.

See "3.7.Configuring the CAN Bit Timing" for a description of the Bit Timing.

Note:

- *With a CAN clock (canfd_cclk) of 8 MHz, the reset value of 0x06000A03 configures the CAN FD Controller for a bit rate of 500 kBit/s.*

bit	31	30	29	28	27	26	25	24
Field	NSJW[6:0]							NBRP[8]
R/W Attribute	R/W							R/W
Protection Attribute	-							-
Initial value	0x03							0

bit	23	22	21	20	19	18	17	16
Field	NBRP[7:0]							
R/W Attribute	R/W							
Protection Attribute	-							
Initial value	0x00							

Bit	15	14	13	12	11	10	9	8
Field	NTSEG1[7:0]							
R/W Attribute	R/W							
Protection Attribute	-							
Initial value	0x0A							

Bit	7	6	5	4	3	2	1	0
Field	Reserved	NTSEG2[6:0]						
R/W Attribute	R0,W0	R/W						
Protection Attribute	-	-						
Initial value	0	0x03						

[bit31:25] NSJW[6:0]: Nominal (Re) Synchronization Jump Width

NSJW[6:0]	Description
0x00-0x7F	Valid values are 0 to 127. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used.

[bit24:16] NBRP[8:0]: Nominal Bit Rate Prescaler

NBRP[8:0]	Description
0x000-0x1FF	The value by which the oscillator frequency is divided for generating the bit time quanta. The bit time is built up from a multiple of this quanta. Valid values for the Nominal Bit Rate Prescaler are 0 to 511. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used.

[bit15:8] NTSEG1[7:0]: Nominal Time Segment before sample Point

NTSEG1[7:0]	Description
0x01-0xFF	Valid values are 1 to 255. The value 0 must not be used. The actual interpretation by the hardware of this value is such that one more than the programmed value is used.

[bit7] Reserved: Reserved Bit

When writing, always write "0". When reading, "0" is always read.

[bit6:0] NTSEG2[6:0]: Nominal Time Segment after sample Point

NTSEG2[6:0]	Description
0x01-0x7F	Valid values are 1 to 127. The value 0 must not be used. The actual interpretation by the hardware of this value is such that one more than the programmed value is used.

5.8. Timestamp Counter Configuration (MCG_CANFDx_TSCC, CPG_CANFDx_TSCC)

The Timestamp Counter Configuration holds the settings for Timestamp Generation.

Write access to the Timestamp Counter Configuration (TSCC) is possible only when the Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) bits are both "1".

For a description of the Timestamp Counter see "3.2. Timestamp Generation".

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

bit	23	22	21	20	19	18	17	16
Field	Reserved				TCP[3:0]			
R/W Attribute	R0,W0				R/W			
Protection Attribute	-				-			
Initial value	0x0				0x0			

bit	15	14	13	12	11	10	9	8
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

Bit	7	6	5	4	3	2	1	0
Field	Reserved						TSS[1:0]	
R/W Attribute	R0,W0						R/W	
Protection Attribute	-						-	
Initial value	000000						00	

[bit31:20] Reserved: Reserved Bits

When writing, always write "0". When reading, "0" is always read.

[bit19:16] TCP[3:0]: Timestamp Counter Prescaler

TCP[3:0]	Description
0x0-0xF	Configures the timestamp and timeout counters time unit in multiples of bit times [1 to 16]. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used.

Note:

- *For each timestamp and timeout counter time unit tick, the CAN FD Controller counts the number of bit times that have elapsed with an internal counter, up to the configured TSCC.TCP[3:0] value. This internal counter is not initialized by Initialization (CCCR.INIT), but only by a hardware reset.*

[bit15:2] Reserved: Reserved Bits

When writing, always write "0". When reading, "0" is always read.

[bit1:0] TSS[1:0]: Timestamp Select

TSS[1:0]	Description
00	Timestamp counter (TSCV.TSC[15:0]) value always 0x0000.
01	Timestamp counter value incremented according to Timestamp Counter Prescaler (TSCC.TCP[3:0]).
10	Timestamp counter value of a counter external to the CAN FD Controller used.
11	Same as "00".

Note:

- *When Timestamp Select TSCC.TSS[1:0] = "01", the clock signal for the Timeout Counter is derived from the CAN Core's sample point signal. Therefore, if the baud rate switch feature in CAN FD is used, the timeout counter is clocked differently in arbitration and data field.*

5.9. Timestamp Counter Value (MCG_CANFDx_TSCV, CPG_CANFDx_TSCV)

Holds the value of the Timestamp Counter.

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

bit	23	22	21	20	19	18	17	16
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

bit	15	14	13	12	11	10	9	8
Field	TSC[15:8]							
R/W Attribute	R,W							
Protection Attribute	-							
Initial value	0x00							

bit	7	6	5	4	3	2	1	0
Field	TSC[7:0]							
R/W Attribute	R,W							
Protection Attribute	-							
Initial value	0x00							

[bit31:16] Reserved: Reserved Bits

When writing, always write "0". When reading, "0" is always read.

[bit15:0] TSC[15:0]: Timestamp Counter

The internal/external Timestamp Counter value is captured on start of frame (both Rx and Tx).

- When Timestamp Select TSCC.TSS[1:0] = "01", the Timestamp Counter TSCV.TSC[15:0] is incremented in multiples of bit times [1 to 16] depending on the configuration of the Timestamp Counter Prescaler (TSCC.TCP[3:0]). A wrap around sets interrupt flag Timestamp Wraparound (IR.TSW). Write access to TSCV resets the counter to zero.
- When Timestamp Select TSCC.TSS[1:0] = "10", TSCV.TSC[15:0] reflects a 16-bit Timestamp Counter value obtained from a counter external to the CAN FD Controller. For this case, a write access has no impact and will not clear the counter value.

Note:

- *A "wrap around" is a change of the Timestamp Counter value from non-zero to zero not caused by write access to Timestamp Counter Value (TSCV). Such write access will not set IR.TSW to "1".*

5.10. Timeout Counter Configuration (MCG_CANFDx_TOCC, CPG_CANFDx_TOCC)

The Timeout Counter Configuration holds the settings for the Timeout Counter.

Write access to the Timeout Counter Configuration (TOCC) is possible only when the Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) bits are both "1".

For a description of the Timeout Counter see "3.3. Timeout Counter".

bit	31	30	29	28	27	26	25	24
Field	TOP[15:8]							
R/W Attribute	R/W							
Protection Attribute	-							
Initial value	0xFF							

bit	23	22	21	20	19	18	17	16
Field	TOP[7:0]							
R/W Attribute	R/W							
Protection Attribute	-							
Initial value	0xFF							

bit	15	14	13	12	11	10	9	8
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

Bit	7	6	5	4	3	2	1	0
Field	Reserved					TOS[1:0]		ETOC
R/W Attribute	R0,W0					R/W		R/W
Protection Attribute	-					-		-
Initial value	00000					00		0

[bit31:16] TOP[15:0]: Timeout Period

Start value of the Timeout Counter (down-counter). Configures the Timeout Period.

[bit15:3] Reserved: Reserved Bits

When writing, always write "0". When reading, "0" is always read.

[bit2:1] TOS[1:0]: Timeout Select

When operating in Continuous mode (TOS = "00"), a write to Timeout Counter Value (TOCV) presets the counter to the value configured by Timeout Period (TOCC.TOP[15:0]) and continues down-counting.

When the Timeout Counter is controlled by one of the FIFOs (TOS = "01"/"10"/"11"), an empty FIFO presets the counter to the value configured by Timeout Period (TOCC.TOP[15:0]). Down-counting is started when the first FIFO element is stored.

TOS[1:0]	Description
00	Continuous operation.
01	Timeout controlled by Tx Event FIFO.
10	Timeout controlled by Rx FIFO 0.
11	Timeout controlled by Rx FIFO 1.

[bit0] ETOC: Enable Timeout Counter

Bit	Description
0	Timeout Counter disabled.
1	Timeout Counter enabled.

5.11. Timeout Counter Value (MCG_CANFDx_TOCV, CPG_CANFDx_TOCV)

Holds the value of the Timeout Counter.

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

bit	23	22	21	20	19	18	17	16
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

Bit	15	14	13	12	11	10	9	8
Field	TOC[15:8]							
R/W Attribute	R,W							
Protection Attribute	-							
Initial value	0xFF							

bit	7	6	5	4	3	2	1	0
Field	TOC[7:0]							
R/W Attribute	R,W							
Protection Attribute	-							
Initial value	0xFF							

[bit31:16] Reserved: Reserved Bits

When writing, always write "0". When reading, "0" is always read.

[bit15:0] TOC[15:0]: Timeout Counter

The Timeout Counter is decremented in multiples of bit times [1 to 16] depending on the configuration of Timestamp Counter Prescaler (TSCC.TCP[3:0]). Setting Configuration Change Enable (CCCR.CCE) will preset the Timeout Counter (TOCV.TOC[15:0]) to the Timeout Period value (TOCC.TOP[15:0]).

- When Timeout Select TOCC.TOS[1:0] = "00" (Continuous Mode), the counter starts when Initialization CCCR.INIT is reset. A write to the Timeout Counter Value register TOCV presets the counter to the value configured by Timeout Period (TOCC.TOP[15:0]) and continues

down-counting. When the counter reaches zero, interrupt flag Timeout Occurred (IR.TOO) is set, and the counter is immediately restarted at Timeout Period (TOCC.TOP[15:0]).

- When Timeout Select TOCC.TOS ≠ "00" (counter is controlled by one of the FIFOs), an empty FIFO presets the counter to the value configured by Timeout Period (TOCC.TOP[15:0]). Down-counting is started when the first FIFO element is stored. In this setting, writing to Timeout Counter Value (TOCV) has no effect. When the counter reaches zero, interrupt flag Timeout Occurred (IR.TOO) is set, and the Timeout Counter is stopped.

5.12. Error Counter Register (MCG_CANFDx_ECR, CPG_CANFDx_ECR)

Holds the values of the Error Counters.

Note:

- *When Restricted Operation Mode (CCCR.ASM) is set, the CAN protocol controller does not increment the Transmit Error Counter (ECR.TEC[7:0]) and Receive Error Counter (ECR.REC[6:0]) when a CAN protocol error is detected, but CAN Error Logging (ECR.CEL[7:0]) is still incremented.*

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	RX,W0							
Protection Attribute	-							
Initial value	0xXX							

bit	23	22	21	20	19	18	17	16
Field	CEL[7:0]							
R/W Attribute	R,WX							
Protection Attribute	-							
Initial value	0x00							

bit	15	14	13	12	11	10	9	8
Field	RP	REC[6:0]						
R/W Attribute	R,WX	R,WX						
Protection Attribute	-	-						
Initial value	0	0000000						

bit	7	6	5	4	3	2	1	0
Field	TEC[7:0]							
R/W Attribute	R,WX							
Protection Attribute	-							
Initial value	0x00							

[bit31:24] Reserved: Reserved Bits

When writing, always write "0". The read value is undefined.

[bit23:16] CEL[7:0]: CAN Error Logging

The counter is incremented each time when a CAN protocol error causes the Transmit Error Counter or the Receive Error Counter to be incremented. It is reset by read access to CEL.

The counter stops at 0xFF; the next increment of Transmit Error Counter (ECR.TEC[7:0]) or Receive Error Counter (ECR.REC[6:0]) sets interrupt flag Error Logging Overflow (IR.ELO).

[bit15] RP: Receive Error Passive

Bit	Description
0	The Receive Error Counter is below the error passive level of 128.
1	The Receive Error Counter has reached the error passive level of 128.

[bit14:8] REC[6:0]: Receive Error Counter

REC[6:0]	Description
0-127	Actual state of the Receive Error Counter, values between 0 and 127.

[bit7:0] TEC[7:0]: Transmit Error Counter

TEC[7:0]	Description
0-255	Actual state of the Transmit Error Counter, values between 0 and 255.

5.13. Protocol Status Register (MCG_CANFDx_PSR, CPG_CANFDx_PSR)

The Protocol Status Register displays the CAN protocol status of the CAN FD Controller.

Note:

- *The PSR register accepts only 16 bit half-word and 32 bit word accesses. 8 bit byte accesses to this register are prohibited.*

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	RX,W0							
Protection Attribute	-							
Initial value	0xXX							

Bit	23	22	21	20	19	18	17	16
Field	Reserved	TDCV[6:0]						
R/W Attribute	RX,W0	R,WX						
Protection Attribute	-	-						
Initial value	X	0x00						

bit	15	14	13	12	11	10	9	8
Field	Reserved	PXE	RFDF	RBRS	RESI	DLEC[2:0]		
R/W Attribute	RX,W0	R,WX	R,WX	R,WX	R,WX	R,WX		
Protection Attribute	-	-	-	-	-	-		
Initial value	X	0	0	0	0	111		

Bit	7	6	5	4	3	2	1	0
Field	BO	EW	EP	ACT[1:0]		LEC[2:0]		
R/W Attribute	R,WX	R,WX	R,WX	R,WX		R,WX		
Protection Attribute	-	-	-	-		-		
Initial value	0	0	0	00		111		

[bit31:23] Reserved: Reserved Bits

When writing, always write "0". The read value is undefined.

[bit22:16] TDCV[6:0]: Transmitter Delay Compensation Value

TDCV[6:0]	Description
0x00-0x7F	Position of the secondary sample point, defined by the sum of the measured delay from canfd_tx to canfd_rx and Transmitter Delay Compensation Offset (TDCR.TDCO[6:0]). The SSP position is, in the data phase, the number of mtq between the start of the transmitted bit and the secondary sample point. Valid values are 0 to 127 mtq.

[bit15] Reserved: Reserved Bit

When writing, always write "0". The read value is undefined.

[bit14] PXE: Protocol Exception Event

A read access will reset this bit to "0".

Bit	Description
0	No protocol exception event occurred since last read access.
1	Protocol exception event occurred.

Note:

- *Once set to "1", PXE will be held at "1" until it is reset to "0" by the CPU.*

[bit13] RFDF: Received a CAN FD Message

This bit is set independent of acceptance filtering. A read access will reset this bit to "0".

Bit	Description
0	Since this bit was reset by the CPU, no CAN FD message has been received.
1	Message in CAN FD format with FDF flag set has been received.

Note:

- *Once set to "1", RFDF will be held at "1" until it is reset to "0" by the CPU.*

[bit12] RBRS: BRS Flag of last Received CAN FD Message

This bit is set together with "Received a CAN FD Message" (PSR.RFDF), independent of acceptance filtering. A read access will reset this bit to "0".

Bit	Description
0	Last received CAN FD message did not have its BRS flag set.
1	Last received CAN FD message had its BRS flag set.

[bit11] RESI: ESI Flag of last Received CAN FD Message

This bit is set together with "Received a CAN FD Message" (PSR.RFDF), independent of acceptance filtering. A read access will reset this bit to "0".

Bit	Description
0	Last received CAN FD message did not have its ESI flag set.
1	Last received CAN FD message had its ESI flag set.

[bit10:8] DLEC[2:0]: Data Phase Last Error Code

Type of last error that occurred in the data phase of a CAN FD format frame with its BRS flag set. Coding is the same as for Last Error Code (PSR.LEC[2:0]).

This field will be cleared to zero when a CAN FD format frame with its BRS flag set has been transferred (reception or transmission) without error.

A read access will set this field to 7.

[bit7] BO: Bus_Off Status

Bit	Description
0	The CAN FD Controller is not Bus_Off.
1	The CAN FD Controller is in Bus_Off state.

[bit6] EW: Warning Status

Bit	Description
0	Both error counters are below the Error_Warning limit of 96.
1	At least one of error counter has reached the Error_Warning limit of 96.

[bit5] EP: Error Passive

Bit	Description
0	The CAN FD Controller is in the Error_Active state. It normally takes part in bus communication and sends an active error flag when an error has been detected.
1	The CAN FD Controller is in the Error_Passive state.

[bit4:3] ACT[1:0]: Activity

Monitors the CAN communication state. This field is reset to "00" when Initialize (CCCR.INIT) is set to "1".

ACT[1:0]	Description
00	Synchronizing - node is synchronizing on CAN communication.
01	Idle - node is neither receiver nor transmitter.
10	Receiver - node is operating as receiver.
11	Transmitter - node is operating as transmitter.

Note:

- *PSR.ACT[1:0] is set to "00" by a Protocol Exception Event.*

[bit2:0] LEC[2:0]: Last Error Code

The LEC indicates the type of the last error to occur on the CAN bus.

This field will be cleared to 0 when a message has been transferred (reception or transmission) without error.

LEC[2:0]	Description
0	No Error: No error occurred since LEC has been reset by successful reception or transmission.
1	Stuff Error: More than 5 equal bits in a sequence have occurred in a part of a received message where this is not allowed.
2	Form Error: A fixed format part of a received frame has the wrong format.
3	AckError: The message transmitted by the CAN FD Controller was not acknowledged by another node.
4	Bit1Error: During the transmission of a message (with the exception of the arbitration field), the CAN FD Controller wanted to send a recessive level, but the monitored bus value was dominant.
5	Bit0Error: During the transmission of a message (or acknowledge bit, or active error flag, or overload flag), the CAN FD Controller wanted to send a dominant level, but the monitored bus value was recessive. During Bus_Off recovery this status is set each time a sequence of 11 recessive bits has been monitored. This enables the CPU to monitor the proceeding of the Bus_Off recovery sequence (indicating the bus is not stuck at dominant or continuously disturbed).
6	CRCError: The CRC check sum of a received message was incorrect. The CRC of an incoming message does not match with the CRC calculated from the received data.
7	NoChange: Any read access to the Protocol Status Register (PSR) re-initializes the Last Error Code field (LEC[2:0]) to 7. When the LEC[2:0] shows the value 7, no CAN bus event was detected since the last CPU read access to the Protocol Status Register.

Notes:

- *When a frame in CAN FD format has reached the data phase with BRS flag set, the next CAN event (error or valid frame) will be shown in the Data Phase Last Error Code field (PSR.DLEC[2:0]) instead of the Last Error Code field (PSR.LEC[2:0]). An error in a fixed stuff bit of a CAN FD CRC sequence will be shown as a Form Error, not Stuff Error.*
- *The Bus_Off recovery sequence (see CAN Specification Rev. 2.0 or ISO11898-1) cannot be shortened by setting or resetting Initialization (CCCR.INIT). If the CAN FD Controller goes Bus_Off, it will set CCCR.INIT of its own accord, stopping all bus activities. Once CCCR.INIT has been cleared by the CPU, the CAN FD Controller will then wait for 129 occurrences of Bus Idle (129 × 11 consecutive recessive bits) before resuming normal operation. At the end of the Bus_Off recovery sequence, the Receive and Transmit Error Counters (ECR.REC[6:0] and ECR.TEC[7:0]) will be reset. During the waiting time after the resetting of CCCR.INIT, each time a sequence of 11 recessive bits has been monitored, a Bit0Error code is written to the Last Error Code field (PSR.LEC[2:0]), enabling the CPU to readily check up whether the CAN bus is stuck at dominant or continuously disturbed, and to monitor the Bus_Off recovery sequence. The Receive Error Counter (ECR.REC[6:0]) is used to count these sequences*

5.14. Transmitter Delay Compensation Register (MCG_CANFDx_TDCR, CPG_CANFDx_TDCR)

The Transmitter Delay Compensation Register configures the offset value and the filter window length for Transmitter Delay Compensation.

This register is only writable if bits Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) are set.

See "3.1.4. Transmitter Delay Compensation" for a description of the TDCO[6:0] and TDCF[6:0].

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0, W0							
Protection Attribute	-							
Initial value	0x00							

bit	23	22	21	20	19	18	17	16
Field	Reserved							
R/W Attribute	R0, W0							
Protection Attribute	-							
Initial value	0x00							

bit	15	14	13	12	11	10	9	8
Field	Reserved	TDCO[6:0]						
R/W Attribute	R0, W0	R/W						
Protection Attribute	-	-						
Initial value	0	0x00						

bit	7	6	5	4	3	2	1	0
Field	Reserved	TDCF[6:0]						
R/W Attribute	R0, W0	R/W						
Protection Attribute	-	-						
Initial value	0	0x00						

[bit31:15] Reserved: Reserved Bits

When writing, always write "0". When reading, "0" is always read.

[bit14:8] TDCO[6:0]: Transmitter Delay Compensation Offset

TDCO[6:0]	Description
0x00-0x7F	Offset value defining the distance between the measured delay from canfd_tx to canfd_rx and the secondary sample point. Valid values are 0 to 127 mtq.

[bit7] Reserved: Reserved Bit

When writing, always write "0". When reading, "0" is always read.

[bit6:0] TDCF[6:0]: Transmitter Delay Compensation Filter Window Length

TDCF[6:0]	Description
0x00-0x7F	Defines the minimum value for the SSP position, dominant edges on canfd_rx that would result in an earlier SSP position are ignored for transmitter delay measurement. The feature is enabled when TDCF is configured to a value greater than TDCO[6:0]. Valid values are 0 to 127 mtq.

5.15. Interrupt Register (MCG_CANFDx_IR, CPG_CANFDx_IR)

The Interrupt Register holds the flags that are set when one of the listed conditions is detected (edge-sensitive). The flags remain set until the CPU clears them. A flag is cleared by writing a "1" to the corresponding bit position. Writing a "0" has no effect. A hard reset will clear the register.

The configuration of Interrupt Enable (IE) controls whether an interrupt is generated.

The configuration of Interrupt Line Select (ILS) controls on which interrupt line (canfd_int0/1) an interrupt is signalled.

Note:

- *The TEFW, RF1W, and RF0W interrupts are asserted only when their respective FIFO levels are equal to their configured watermarks. If at this point the interrupt is cleared, and the respective FIFO continues to be filled, the interrupt will not be re-asserted even if the respective FIFO level is greater than the configured watermark.*

bit	31	30	29	28	27	26	25	24
Field	Reserved		ARA	PED	PEA	WDI	BO	EW
R/W Attribute	R0,W0		R,W	R,W	R,W	R,W	R,W	R,W
Protection Attribute	-		-	-	-	-	-	-
Initial value	00		0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	EP	ELO	BEU	BEC	DRX	TOO	MRAF	TSW
R/W Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	TEFL	TEFF	TEFW	TEFN	TFE	TCF	TC	HPM
R/W Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	RF1L	RF1F	RF1W	RF1N	RF0L	RF0F	RF0W	RF0N
R/W Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

[bit31:30] Reserved: Reserved Bits

When writing, always write "0". When reading, "0" is always read.

[bit29] ARA: Access to Reserved Address

See Table 5-1 for the reserved addresses in the CAN FD Controller.

Bit	Description
0	No access to reserved address occurred.
1	Access to reserved address occurred.

[bit28] PED: Protocol Error in Data Phase (Data Bit Time is Used)

Bit	Description
0	No protocol error in data phase.
1	Protocol error in data phase detected (Data Phase Last Error Code PSR.DLEC[2:0] ≠ 0, 7).

[bit27] PEA: Protocol Error in Arbitration Phase (Nominal Bit Time is Used)

Bit	Description
0	No protocol error in arbitration phase.
1	Protocol error in arbitration phase detected (Last Error Code PSR.LEC[2:0] ≠ 0, 7).

[bit26] WDI: Watchdog Interrupt

Bit	Description
0	No Message RAM Watchdog event occurred.
1	Message RAM Watchdog event due to incomplete access of Message RAM.

[bit25] BO: Bus_Off Status

Bit	Description
0	Bus_Off status unchanged.
1	Bus_Off status changed.

[bit24] EW: Warning Status

Bit	Description
0	Error_Warning status unchanged.
1	Error_Warning status changed.

[bit23] EP: Error Passive

Bit	Description
0	Error_Passive status unchanged.
1	Error_Passive status changed.

[bit22] ELO: Error Logging Overflow

Bit	Description
0	CAN Error Logging Counter (ECR.CEL[7:0]) did not overflow.
1	Overflow of CAN Error Logging Counter (ECR.CEL[7:0]) occurred.

[bit21] BEU: Bit Error Uncorrected

Message RAM bit error during access from CAN FD Controller detected, but could not be corrected by the ECC (Error Correction Code) logic attached to the Message RAM.

An uncorrected Message RAM bit error sets Initialization (CCCR.INIT) to "1". This is done to avoid transmission of corrupted data.

Bit	Description
0	No bit error detected when reading from Message RAM.
1	Bit error detected, but could not be corrected.

Notes:

- *The corresponding bit in the ECC logic must also be cleared when clearing IR.BEU.*
- *In case Bit Error Uncorrected IR.BEU occurs, due to the synchronization mechanism between the Bus clock and the CAN clock, there may be a delay until Initialization CCCR.INIT set to "1". Therefore the programmer has to assure that CCCR.INIT has been set to "1" by reading CCCR.INIT before resetting CCCR.INIT to "0".*

[bit20] BEC: Bit Error Corrected

Message RAM bit error during access from CAN FD Controller detected and corrected by the ECC logic attached to the Message RAM.

Bit	Description
0	No bit error detected when reading from Message RAM.
1	Bit error detected and corrected by ECC logic.

Note:

- *The corresponding bit in the ECC logic must also be cleared when clearing IR.BEC.*

[bit19] DRX: Message Stored to Dedicated Rx Buffer

The flag is set whenever a received message has been stored into a dedicated Rx Buffer. This flag is not set by reception of a Debug message.

Bit	Description
0	No Rx Buffer updated.
1	At least one received message stored into an Rx Buffer.

[bit18] TOO: Timeout Occurred

Bit	Description
0	No timeout.
1	Timeout reached.

[bit17] MRAF: Message RAM Access Failure

The flag is set, when the Rx Handler

- has not completed acceptance filtering or storage of an accepted message until the arbitration field of the following message has been received. In this case acceptance filtering or message storage is aborted and the Rx Handler starts processing of the following message.
- completed message reception operation, but was not able to write the message to the Message RAM. In this case message storage is aborted.
- The data of a received frame may be incompletely stored to the Message RAM after the flag is set by the Rx Handler operation. In addition, Rx FIFO 0 Status (RXF0S), Rx FIFO 1 Status (RXF1S), or New Data 1/2 (NDAT1/2) is also updated according as the storage location (Rx FIFO0, Rx FIFO1, or an Rx Buffer). Therefore, in this case discard the received frame data.

The flag is also set when the Tx Handler was not able to read a message from the Message RAM in time. In this case message trasmission is aborted. In case of a Tx Handler access failure the CAN FD Controller is switched into Restricted Operation Mode (see "3.1.5. Restricted Operation Mode"). To leave Restricted Operation Mode, the CPU has to reset the Restricted Operation Mode bit (CCCR.ASM).

Bit	Description
0	No Message RAM access failure occurred.
1	Message RAM access failure occurred.

[bit16] TSW: Timestamp Wraparound

Bit	Description
0	No timestamp counter wrap-around.
1	Timestamp counter wrapped around.

[bit15] TEFL: Tx Event FIFO Element Lost

Bit	Description
0	No Tx Event FIFO element lost.
1	Tx Event FIFO element lost. Also set after write attempt to Tx Event FIFO of size zero.

[bit14] TEFF: Tx Event FIFO Full

Bit	Description
0	Tx Event FIFO not full.
1	Tx Event FIFO full.

[bit13] TEFW: Tx Event FIFO Watermark Reached

Bit	Description
0	Tx Event FIFO fill level below watermark.
1	Tx Event FIFO fill level reached watermark.

[bit12] TEFN: Tx Event FIFO New Entry

Bit	Description
0	Tx Event FIFO unchanged.
1	Tx Handler wrote Tx Event FIFO element.

[bit11] TFE: Tx FIFO Empty

Bit	Description
0	Tx FIFO non-empty.
1	Tx FIFO empty.

Note:

- *The TFE bit will not be asserted for an empty Tx Queue.*

[bit10] TCF: Transmission Cancellation Finished

Bit	Description
0	No transmission cancellation finished.
1	Transmission cancellation finished.

[bit9] TC: Transmission Completed

Bit	Description
0	No transmission completed.
1	Transmission completed.

[bit8] HPM: High Priority Message

Bit	Description
0	No high priority message received.
1	High priority message received.

[bit7] RF1L: Rx FIFO 1 Message Lost

Bit	Description
0	No Rx FIFO 1 message lost.
1	Rx FIFO 1 message lost. Also set after write attempt to Rx FIFO 1 of size zero.

[bit6] RF1F: Rx FIFO 1 Full

Bit	Description
0	Rx FIFO 1 not full.
1	Rx FIFO 1 full.

[bit5] RF1W: Rx FIFO 1 Watermark Reached

Bit	Description
0	Rx FIFO 1 fill level below watermark.
1	Rx FIFO 1 fill level reached watermark.

[bit4] RF1N: Rx FIFO 1 New Message

Bit	Description
0	No new message written to Rx FIFO 1.
1	New message written to Rx FIFO 1.

[bit3] RF0L: Rx FIFO 0 Message Lost

Bit	Description
0	No Rx FIFO 0 message lost.
1	Rx FIFO 0 message lost. Also set after write attempt to Rx FIFO 0 of size zero.

[bit2] RF0F: Rx FIFO 0 Full

Bit	Description
0	Rx FIFO 0 not full.
1	Rx FIFO 0 full.

[bit1] RF0W: Rx FIFO 0 Watermark Reached

Bit	Description
0	Rx FIFO 0 fill level below watermark.
1	Rx FIFO 0 fill level reached watermark.

[bit0] RF0N: Rx FIFO 0 New Message

Bit	Description
0	No new message written to Rx FIFO 0.
1	New message written to Rx FIFO 0.

5.16. Interrupt Enable (MCG_CANFDx_IE, CPG_CANFDx_IE)

The settings in the Interrupt Enable register determine which status changes in the Interrupt Register (IR) will be signalled on an interrupt line.

Bit	Description
0	Interrupt disabled.
1	Interrupt enabled.

bit	31	30	29	28	27	26	25	24
Field	Reserved		ARAE	PEDE	PEAE	WDIE	BOE	EWE
R/W Attribute	R0,W0		R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-		-	-	-	-	-	-
Initial value	00		0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	EPE	ELOE	BEUE	BECE	DRXE	TOOE	MRAFE	TSWE
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	TEFLE	TEFFE	TEFWE	TEFNE	TFEE	TCFE	TCE	HPME
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	RF1LE	RF1FE	RF1WE	RF1NE	RF0LE	RF0FE	RF0WE	RF0NE
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

[bit31:30] Reserved: Reserved Bits

When writing, always write "0". When reading, "0" is always read.

[bit29] ARAE: Access to Reserved Address Enable

[bit28] PEDE: Protocol Error in Data Phase Enable

[bit27] PEAE: Protocol Error in Arbitration Phase Enable

[bit26] WDIE: Watchdog Interrupt Enable

[bit25] BOE: Bus_Off Status Interrupt Enable

[bit24] EWE: Warning Status Interrupt Enable

[bit23] EPE: Error Passive Interrupt Enable

[bit22] ELOE: Error Logging Overflow Interrupt Enable

[bit21] BEUE: Bit Error Uncorrected Interrupt Enable

[bit20] BECE: Bit Error Corrected Interrupt Enable

[bit19] DRXE: Message Stored to Dedicated Rx Buffer Interrupt Enable

[bit18] TOOE: Timeout Occurred Interrupt Enable

[bit17] MRAFE: Message RAM Access Failure Interrupt Enable

[bit16] TSWE: Timestamp Wraparound Interrupt Enable

[bit15] TEFLE: Tx Event FIFO Event Lost Interrupt Enable

[bit14] TEFFE: Tx Event FIFO Full Interrupt Enable

[bit13] TEFWE: Tx Event FIFO Watermark Reached Interrupt Enable

[bit12] TEFNE: Tx Event FIFO New Entry Interrupt Enable

[bit11] TFEE: Tx FIFO Empty Interrupt Enable

[bit10] TCFE: Transmission Cancellation Finished Interrupt Enable

[bit9] TCE: Transmission Completed Interrupt Enable

[bit8] HPME: High Priority Message Interrupt Enable

[bit7] RF1LE: Rx FIFO 1 Message Lost Interrupt Enable

[bit6] RF1FE: Rx FIFO 1 Full Interrupt Enable

[bit5] RF1WE: Rx FIFO 1 Watermark Reached Interrupt Enable

[bit4] RF1NE: Rx FIFO 1 New Message Interrupt Enable

[bit3] RF0LE: Rx FIFO 0 Message Lost Interrupt Enable

[bit2] RF0FE: Rx FIFO 0 Full Interrupt Enable

[bit1] RF0WE: Rx FIFO 0 Watermark Reached Interrupt Enable

[bit0] RF0NE: Rx FIFO 0 New Message Interrupt Enable

5.17. Interrupt Line Select (MCG_CANFDx_ILS, CPG_CANFDx_ILS)

The Interrupt Line Select register assigns an interrupt generated by a specific interrupt flag from the Interrupt Register (IR) to one of the two CAN FD Controller interrupt lines (canfd_int0/1).

Bit	Description
0	Interrupt assigned to interrupt line canfd_int0.
1	Interrupt assigned to interrupt line canfd_int1.

bit	31	30	29	28	27	26	25	24
Field	Reserved		ARAL	PEDL	PEAL	WDIL	BOL	EWL
R/W Attribute	R0,W0		R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-		-	-	-	-	-	-
Initial value	00		0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	EPL	ELOL	BEUL	BECL	DRXL	TOOL	MRAFL	TSWL
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	TEFLL	TEFFL	TEFWL	TEFNL	TFEL	TCFL	TCL	HPML
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	RF1LL	RF1FL	RF1WL	RF1NL	RF0LL	RF0FL	RF0WL	RF0NL
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

[bit31:30] Reserved: Reserved Bits

When writing, always write "0". When reading, "0" is always read.

[bit29] ARAL: Access to Reserved Address Line

[bit28] PEDL: Protocol Error in Data Phase Line

[bit27] PEAL: Protocol Error in Arbitration Phase Line

[bit26] WDIL: Watchdog Interrupt Line

[bit25] BOL: Bus_Off Status Interrupt Line

[bit24] EWL: Warning Status Interrupt Line

[bit23] EPL: Error Passive Interrupt Line

[bit22] ELOL: Error Logging Overflow Interrupt Line

[bit21] BEUL: Bit Error Uncorrected Interrupt Line

[bit20] BECL: Bit Error Corrected Interrupt Line

[bit19] DRXL: Message Stored to Dedicated Rx Buffer Interrupt Line

[bit18] TOOL: Timeout Occurred Interrupt Line

[bit17] MRAFL: Message RAM Access Failure Interrupt Line

[bit16] TSWL: Timestamp Wraparound Interrupt Line

[bit15] TEFLL: Tx Event FIFO Event Lost Interrupt Line

[bit14] TEFFL: Tx Event FIFO Full Interrupt Line

[bit13] TEFWL: Tx Event FIFO Watermark Reached Interrupt Line

[bit12] TEFNL: Tx Event FIFO New Entry Interrupt Line

[bit11] TFEL: Tx FIFO Empty Interrupt Line

[bit10] TCFL: Transmission Cancellation Finished Interrupt Line

[bit9] TCL: Transmission Completed Interrupt Line

[bit8] HPML: High Priority Message Interrupt Line

[bit7] RF1LL: Rx FIFO 1 Message Lost Interrupt Line

[bit6] RF1FL: Rx FIFO 1 Full Interrupt Line

[bit5] RF1WL: Rx FIFO 1 Watermark Reached Interrupt Line

[bit4] RF1NL: Rx FIFO 1 New Message Interrupt Line

[bit3] RF0LL: Rx FIFO 0 Message Lost Interrupt Line

[bit2] RF0FL: Rx FIFO 0 Full Interrupt Line

[bit1] RF0WL: Rx FIFO 0 Watermark Reached Interrupt Line

[bit0] RF0NL: Rx FIFO 0 New Message Interrupt Line

5.18. Interrupt Line Enable (MCG_CANFDx_ILE, CPG_CANFDx_ILE)

Interrupt Line Enable can separately enable/disable each of the two interrupt lines to the CPU by the values set to Enable Interrupt Line 0 (ILE.EINT0) and Enable Interrupt Line 1 (ILE.EINT1).

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

Bit	23	22	21	20	19	18	17	16
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

Bit	15	14	13	12	11	10	9	8
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

Bit	7	6	5	4	3	2	1	0
Field	Reserved						EINT1	EINT0
R/W Attribute	R0,W0						R/W	R/W
Protection Attribute	-						-	-
Initial value	000000						0	0

[bit31:2] Reserved: Reserved Bits

When writing, always write "0". When reading, "0" is always read.

[bit1] EINT1: Enable Interrupt Line 1

Bit	Description
0	Interrupt line canfd_int1 disabled.
1	Interrupt line canfd_int1 enabled.

[bit0] EINT0: Enable Interrupt Line 0

Bit	Description
0	Interrupt line canfd_int0 disabled.
1	Interrupt line canfd_int0 enabled.

5.19. Global Filter Configuration (MCG_CANFDx_GFC, CPG_CANFDx_GFC)

Global settings for Message ID filtering. The Global Filter Configuration (GFC) controls the filter path for standard and extended messages as described in Figure 3-4 and Figure 3-5.

Write access to the Global Filter Configuration (GFC) is possible only when the Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) bits are both "1".

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

Bit	23	22	21	20	19	18	17	16
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

Bit	15	14	13	12	11	10	9	8
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

Bit	7	6	5	4	3	2	1	0
Field	Reserved		ANFS[1:0]		ANFE[1:0]		RRFS	RRFE
R/W Attribute	R0,W0		R/W		R/W		R/W	R/W
Protection Attribute	-		-		-		-	-
Initial value	00		00		00		0	0

[bit31:6] Reserved: Reserved Bits

When writing, always write "0". When reading, "0" is always read.

[bit5:4] ANFS[1:0]: Accept Non-matching Frames Standard

Defines how received messages with 11-bit IDs that do not match any element of the filter list are treated.

ANFS[1:0]	Description
00	Accept in Rx FIFO 0.
01	Accept in Rx FIFO 1.
10	Reject.
11	Reject.

[bit3:2] ANFE[1:0]: Accept Non-matching Frames Extended

Defines how received messages with 29-bit IDs that do not match any element of the filter list are treated.

ANFE[1:0]	Description
00	Accept in Rx FIFO 0.
01	Accept in Rx FIFO 1.
10	Reject.
11	Reject.

[bit1] RRFS: Reject Remote Frames Standard

Bit	Description
0	Filter remote frames with 11-bit standard IDs.
1	Reject all remote frames with 11-bit standard IDs.

[bit0] RRFE: Reject Remote Frames Extended

Bit	Description
0	Filter remote frames with 29-bit extended IDs.
1	Reject all remote frames with 29-bit extended IDs.

5.20. Standard ID Filter Configuration (MCG_CANFDx_SIDFC, CPG_CANFDx_SIDFC)

Settings for 11-bit standard Message ID filtering. The Standard ID Filter Configuration controls the filter path for standard messages as described in Figure 3-4.

Write access to the Standard ID Filter Configuration (SIDFC) is possible only when the Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) bits are both "1".

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

bit	23	22	21	20	19	18	17	16
Field	LSS[7:0]							
R/W Attribute	R/W							
Protection Attribute	-							
Initial value	0x00							

Bit	15	14	13	12	11	10	9	8
Field	FLSSA[15:8]							
R/W Attribute	R/W							
Protection Attribute	-							
Initial value	00000000							

bit	7	6	5	4	3	2	1	0
Field	FLSSA[7:2]						Reserved	
R/W Attribute	R/W						R0,W0	
Protection Attribute	-						-	
Initial value	000000						00	

[bit31:24] Reserved: Reserved Bits

When writing, always write "0". When reading, "0" is always read.

[bit23:16] LSS[7:0]: List Size Standard

LSS[7:0]	Description
0	No standard Message ID filter.
1-128	Number of standard Message ID filter elements.
>128	Values greater than 128 are interpreted as 128.

[bit15:2] FLSSA[15:2]: Filter List Standard Start Address

Start address of standard Message ID filter list (32-bit word address, see Figure 6-1).

[bit1:0] Reserved: Reserved Bits

When writing, always write "0". When reading, "0" is always read.

5.21. Extended ID Filter Configuration (MCG_CANFDx_XIDFC, CPG_CANFDx_XIDFC)

Settings for 29-bit extended Message ID filtering. The Extended ID Filter Configuration controls the filter path for extended messages as described in Figure 3-5.

Write access to the Extended ID Filter Configuration (XIDFC) is possible only when the Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) bits are both "1".

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

Bit	23	22	21	20	19	18	17	16
Field	Reserved	LSE[6:0]						
R/W Attribute	R0,W0	R/W						
Protection Attribute	-	-						
Initial value	0	0000000						

Bit	15	14	13	12	11	10	9	8
Field	FLESA[15:8]							
R/W Attribute	R/W							
Protection Attribute	-							
Initial value	00000000							

Bit	7	6	5	4	3	2	1	0
Field	FLESA[7:2]						Reserved	
R/W Attribute	R/W						R0,W0	
Protection Attribute	-						-	
Initial value	000000						00	

[bit31:23] Reserved: Reserved Bits

When writing, always write "0". When reading, "0" is always read.

[bit22:16] LSE[6:0]: List Size Extended

LSE[6:0]	Description
0	No extended Message ID filter.
1-64	Number of extended Message ID filter elements.
>64	Values greater than 64 are interpreted as 64.

[bit15:2] FLESA[15:2]: Filter List Extended Start Address

Start address of extended Message ID filter list (32-bit word address, see Figure 6-1).

[bit1:0] Reserved: Reserved Bits

When writing, always write "0". When reading, "0" is always read.

5.22. Extended ID AND Mask (MCG_CANFDx_XIDAM, CPG_CANFDx_XIDAM)

The Extended ID AND Mask defines the valid bits of a 29-bit ID for acceptance filtering.

Write access to the Extended ID AND Mask (XIDAM) is possible only when the Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) bits are both "1".

bit	31	30	29	28	27	26	25	24
Field	Reserved			EIDM[28:24]				
R/W Attribute	R0,W0			R/W				
Protection Attribute	-			-				
Initial value	000			11111				

Bit	23	22	21	20	19	18	17	16
Field	EIDM[23:16]							
R/W Attribute	R/W							
Protection Attribute	-							
Initial value	0xff							

Bit	15	14	13	12	11	10	9	8
Field	EIDM[15:8]							
R/W Attribute	R/W							
Protection Attribute	-							
Initial value	0xff							

bit	7	6	5	4	3	2	1	0
Field	EIDM[7:0]							
R/W Attribute	R/W							
Protection Attribute	-							
Initial value	0xff							

[bit31:29] Reserved: Reserved Bits

When writing, always write "0". When reading, "0" is always read.

[bit28:0] EIDM[28:0]: Extended ID Mask

For acceptance filtering of extended frames the Extended ID AND Mask is ANDed with the Message ID of a received frame.

With the reset value of all bits set to one the mask is not active.

5.23. High Priority Message Status (MCG_CANFDx_HPMS, CPG_CANFDx_HPMS)

This register is updated every time a Message ID filter element configured to generate a priority event matches. This can be used to monitor the status of incoming high priority messages and to enable fast access to these messages.

Note:

- *If all the following conditions are fulfilled, set at least 1 to both List Size Standard (SIDFC.LSS[7:0]) and List Size Extended (XIDFC.LSE[6:0]).*
  - *To use high priority events*
  - *To use both standard and extended frame format*

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	RX,W0							
Protection Attribute	-							
Initial value	0xXX							

Bit	23	22	21	20	19	18	17	16
Field	Reserved							
R/W Attribute	RX,W0							
Protection Attribute	-							
Initial value	0xXX							

Bit	15	14	13	12	11	10	9	8
Field	FLST	FIDX[6:0]						
R/W Attribute	R,WX	R,WX						
Protection Attribute	-	-						
Initial value	0	0000000						

Bit	7	6	5	4	3	2	1	0
Field	MSI[1:0]		BIDX[5:0]					
R/W Attribute	R,WX		R,WX					
Protection Attribute	-		-					
Initial value	00		000000					

[bit31:16] Reserved: Reserved Bits

When writing, always write "0". The read value is undefined.

[bit15] FLST: Filter List

Indicates the filter list of the matching filter element.

Bit	Description
0	Standard Filter List.
1	Extended Filter List.

[bit14:8] FIDX[6:0]: Filter Index

FIDX[6:0]	Description
0-127	Index of matching Rx acceptance filter element. Range is 0 to List Size Standard/Extended minus 1 (i.e. SIDFC.LSS[7:0] - 1 resp. XIDFC.LSE[6:0] - 1).

[bit7:6] MSI[1:0]: Message Storage Indicator

MSI[1:0]	Description
00	No Rx FIFO selected.
01	Rx FIFO message lost.
10	Message stored in Rx FIFO 0.
11	Message stored in Rx FIFO 1.

[bit5:0] BIDX[5:0]: Buffer Index

Index of Rx FIFO element to which the message was stored. Only valid when bit[1] of the Message Storage Indicator MSI[1] = "1".

5.24. New Data 1 (MCG_CANFDx_NDAT1, CPG_CANFDx_NDAT1)

New Data 1 holds flags that are set when the respective dedicated Rx Buffer receives a frame.

bit	31	30	29	28	27	26	25	24
Field	ND31	ND30	ND29	ND28	ND27	ND26	ND25	ND24
R/W Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	ND23	ND22	ND21	ND20	ND19	ND18	ND17	ND16
R/W Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	ND15	ND14	ND13	ND12	ND11	ND10	ND9	ND8
R/W Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	ND7	ND6	ND5	ND4	ND3	ND2	ND1	ND0
R/W Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

[bit31:0] ND[31:0]: New Data

The register holds the New Data flags of dedicated Rx Buffers 0 to 31.

The flags are set when the respective dedicated Rx Buffer has been updated from a received frame. The flags remain set until the CPU clears them.

A flag is cleared by writing a "1" to the corresponding bit position. Writing a "0" has no effect. A hard reset will clear the register.

Bit	Description
0	Rx Buffer not updated.
1	Rx Buffer updated from new message.

5.25. New Data 2 (MCG_CANFDx_NDAT 2, CPG_CANFDx_NDAT 2)

New Data 2 holds flags that are set when the respective dedicated Rx Buffer receives a frame.

bit	31	30	29	28	27	26	25	24
Field	ND63	ND62	ND61	ND60	ND59	ND58	ND57	ND56
R/W Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	ND55	ND54	ND53	ND52	ND51	ND50	ND49	ND48
R/W Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	ND47	ND46	ND45	ND44	ND43	ND42	ND41	ND40
R/W Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	ND39	ND38	ND37	ND36	ND35	ND34	ND33	ND32
R/W Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

[bit31:0] ND[63:32]: New Data

The register holds the New Data flags of dedicated Rx Buffers 32 to 63.

The flags are set when the respective dedicated Rx Buffer has been updated from a received frame. The flags remain set until the CPU clears them.

A flag is cleared by writing a "1" to the corresponding bit position. Writing a "0" has no effect. A hard reset will clear the register.

Bit	Description
0	Rx Buffer not updated.
1	Rx Buffer updated from new message.

5.26. Rx FIFO 0 Configuration (MCG_CANFDx_RXF0C, CPG_CANFDx_RXF0C)

Settings for the Rx FIFO 0.

Write access to the Rx FIFO 0 Configuration (RXF0C) is possible only when the Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) bits are both "1".

bit	31	30	29	28	27	26	25	24
Field	F0OM	F0WM[6:0]						
R/W Attribute	R/W	R/W						
Protection Attribute	-	-						
Initial value	0	0000000						

bit	23	22	21	20	19	18	17	16
Field	Reserved	F0S[6:0]						
R/W Attribute	R0,W0	R/W						
Protection Attribute	-	-						
Initial value	0	0000000						

Bit	15	14	13	12	11	10	9	8
Field	F0SA[15:8]							
R/W Attribute	R/W							
Protection Attribute	-							
Initial value	00000000							

Bit	7	6	5	4	3	2	1	0
Field	F0SA[7:2]						Reserved	
R/W Attribute	R/W						R0,W0	
Protection Attribute	-						-	
Initial value	000000						00	

[bit31] F0OM: FIFO 0 Operation Mode

Rx FIFO 0 can be operated in blocking or in overwrite mode (see "3.4.2. Rx FIFOs").

Bit	Description
0	Rx FIFO 0 blocking mode.
1	Rx FIFO 0 overwrite mode.

[bit30:24] F0WM[6:0]: Rx FIFO 0 Watermark

F0WM[6:0]	Description
0	Watermark interrupt disabled.
1-64	Level for Rx FIFO 0 Watermark Reached interrupt (IR.RF0W).
>64	Watermark interrupt disabled.

[bit23] Reserved: Reserved Bit

When writing, always write "0". When reading, "0" is always read.

[bit22:16] F0S[6:0]: Rx FIFO 0 Size

The Rx FIFO 0 elements are indexed from 0 to F0S[6:0] - 1.

F0S[6:0]	Description
0	No Rx FIFO 0.
1-64	Number of Rx FIFO 0 elements.
>64	Values greater than 64 are interpreted as 64.

[bit15:2] F0SA[15:2]: Rx FIFO 0 Start Address

Start address of Rx FIFO 0 in Message RAM (32-bit word address, see Figure 6-1).

[bit1:0] Reserved: Reserved Bits

When writing, always write "0". When reading, "0" is always read.

5.27. Rx FIFO 0 Status (MCG_CANFDx_RXF0S, CPG_CANFDx_RXF0S)

Status of the Rx FIFO 0.

bit	31	30	29	28	27	26	25	24
Field	Reserved						RF0L	F0F
R/W Attribute	RX,W0						R,WX	R,WX
Protection Attribute	-						-	-
Initial value	XXXXXX						0	0

Bit	23	22	21	20	19	18	17	16
Field	Reserved		F0PI[5:0]					
R/W Attribute	RX,W0		R,WX					
Protection Attribute	-		-					
Initial value	XX		000000					

Bit	15	14	13	12	11	10	9	8
Field	Reserved		F0GI[5:0]					
R/W Attribute	RX,W0		R,WX					
Protection Attribute	-		-					
Initial value	XX		000000					

Bit	7	6	5	4	3	2	1	0
Field	Reserved	F0FL[6:0]						
R/W Attribute	RX,W0	R,WX						
Protection Attribute	-	-						
Initial value	X	0000000						

[bit31:26] Reserved: Reserved Bits

When writing, always write "0". The read value is undefined.

[bit25] RF0L: Rx FIFO 0 Message Lost

This bit is a copy of interrupt flag IR.RF0L (Rx FIFO 0 Message Lost).

When IR.RF0L is reset, this bit is also reset.

Bit	Description
0	No Rx FIFO 0 message lost.
1	Rx FIFO 0 message lost. Also set after write attempt to Rx FIFO 0 of size zero.

Note:

- *Overwriting a message when the FIFO is in overwrite mode (RXF0C.F0OM = "1") will not set this flag.*

[bit24] F0F: Rx FIFO 0 Full

Bit	Description
0	Rx FIFO 0 not full.
1	Rx FIFO 0 full.

[bit23:22] Reserved: Reserved Bits

When writing, always write "0". The read value is undefined.

[bit21:16] F0PI[5:0]: Rx FIFO 0 Put Index

F0PI[5:0]	Description
0-63	Rx FIFO 0 write index pointer, range 0 to 63.

[bit15:14] Reserved: Reserved Bits

When writing, always write "0". The read value is undefined.

[bit13:8] F0GI[5:0]: Rx FIFO 0 Get Index

F0GI[5:0]	Description
0-63	Rx FIFO 0 read index pointer, range 0 to 63.

[bit7] Reserved: Reserved Bit

When writing, always write "0". The read value is undefined.

[bit6:0] F0FL[6:0]: Rx FIFO 0 Fill Level

F0FL[6:0]	Description
0-64	Number of elements stored in Rx FIFO 0, range 0 to 64.

5.28. Rx FIFO 0 Acknowledge (MCG_CANFDx_RXF0A, CPG_CANFDx_RXF0A)

The Rx FIFO 0 Acknowledge is used to acknowledge that the CPU has read a message or a sequence of messages from the Rx FIFO 0 to indicate to the CAN FD Controller that the corresponding Message RAM area may be released. See "3.6. FIFO Acknowledge Handling" for details.

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

Bit	23	22	21	20	19	18	17	16
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

Bit	15	14	13	12	11	10	9	8
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

Bit	7	6	5	4	3	2	1	0
Field	Reserved		F0AI[5:0]					
R/W Attribute	R0,W0		R/W					
Protection Attribute	-		-					
Initial value	00		000000					

[bit31:6] Reserved: Reserved Bits

When writing, always write "0". When reading, "0" is always read.

[bit5:0] F0AI[5:0]: Rx FIFO 0 Acknowledge Index

After the CPU has read a message or a sequence of messages from Rx FIFO 0, it has to write the buffer index of the last element read from Rx FIFO 0 to this field (F0AI[5:0]). This will set the Rx FIFO 0 Get Index RXF0S.F0GI[5:0] to F0AI[5:0] + 1 and update the FIFO 0 Fill Level RXF0S.F0FL[6:0].

5.29. Rx Buffer Configuration (MCG_CANFDx_RXBC, CPG_CANFDx_RXBC)

Defines the start address of the Rx Buffers section in the Message RAM.

Write access to the Rx Buffer Configuration (RXBC) is possible only when the Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) bits are both "1".

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

bit	23	22	21	20	19	18	17	16
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

Bit	15	14	13	12	11	10	9	8
Field	RBSA[15:8]							
R/W Attribute	R/W							
Protection Attribute	-							
Initial value	00000000							

bit	7	6	5	4	3	2	1	0
Field	RBSA[7:2]						Reserved	
R/W Attribute	R/W						R0,W0	
Protection Attribute	-						-	
Initial value	000000						00	

[bit31:16] Reserved: Reserved Bits

When writing, always write "0". When reading, "0" is always read.

[bit15:2] RBSA[15:2]: Rx Buffer Start Address

Configures the start address of the Rx Buffers section in the Message RAM (32-bit word address).

Also used to reference debug messages A, B, C.

[bit1:0] Reserved: Reserved Bits

When writing, always write "0". When reading, "0" is always read.

5.30. Rx FIFO 1 Configuration (MCG_CANFDx_RXF1C, CPG_CANFDx_RXF1C)

Settings for the Rx FIFO 1.

Write access to the Rx FIFO 1 Configuration (RXF1C) is possible only when the Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) bits are both "1".

bit	31	30	29	28	27	26	25	24
Field	F1OM	F1WM[6:0]						
R/W Attribute	R/W	R/W						
Protection Attribute	-	-						
Initial value	0	0000000						

bit	23	22	21	20	19	18	17	16
Field	Reserved	F1S[6:0]						
R/W Attribute	R0,W0	R/W						
Protection Attribute	-	-						
Initial value	0	0000000						

bit	15	14	13	12	11	10	9	8
Field	F1SA[15:8]							
R/W Attribute	R/W							
Protection Attribute	-							
Initial value	00000000							

bit	7	6	5	4	3	2	1	0
Field	F1SA[7:2]						Reserved	
R/W Attribute	R/W						R0,W0	
Protection Attribute	-						-	
Initial value	000000						00	

[bit31] F1OM: FIFO 1 Operation Mode

Rx FIFO 1 can be operated in blocking or in overwrite mode (see "3.4.2. Rx FIFOs").

Bit	Description
0	Rx FIFO 1 blocking mode.
1	Rx FIFO 1 overwrite mode.

[bit30:24] F1WM[6:0]: Rx FIFO 1 Watermark

F1WM[6:0]	Description
0	Watermark interrupt disabled.
1-64	Level for Rx FIFO 1 Watermark Reached interrupt (IR.RF1W).
>64	Watermark interrupt disabled.

[bit23] Reserved: Reserved Bit

When writing, always write "0". When reading, "0" is always read.

[bit22:16] F1S[6:0]: Rx FIFO 1 Size

The Rx FIFO 1 elements are indexed from 0 to F1S[6:0] - 1.

F1S[6:0]	Description
0	No Rx FIFO 1.
1-64	Number of Rx FIFO 1 elements.
>64	Values greater than 64 are interpreted as 64.

[bit15:2] F1SA[15:2]: Rx FIFO 1 Start Address

Start address of Rx FIFO 1 in Message RAM (32-bit word address, see Figure 6-1).

[bit1:0] Reserved: Reserved Bits

When writing, always write "0". When reading, "0" is always read.

5.31. Rx FIFO 1 Status (MCG_CANFDx_RXF1S, CPG_CANFDx_RXF1S)

Status of the Rx FIFO 1.

bit	31	30	29	28	27	26	25	24
Field	DMS[1:0]		Reserved				RF1L	F1F
R/W Attribute	R,WX		RX,W0				R,WX	R,WX
Protection Attribute	-		-				-	-
Initial value	00		XXXX				0	0

bit	23	22	21	20	19	18	17	16
Field	Reserved		F1PI[5:0]					
R/W Attribute	RX,W0		R,WX					
Protection Attribute	-		-					
Initial value	XX		000000					

Bit	15	14	13	12	11	10	9	8
Field	Reserved		F1GI[5:0]					
R/W Attribute	RX,W0		R,WX					
Protection Attribute	-		-					
Initial value	XX		000000					

Bit	7	6	5	4	3	2	1	0
Field	Reserved	F1FL[6:0]						
R/W Attribute	RX,W0	R,WX						
Protection Attribute	-	-						
Initial value	X	0000000						

[bit31:30] DMS[1:0]: Debug Message Status

DMS[1:0]	Description
00	Idle state, wait for reception of debug messages, DMA request is cleared.
01	Debug message A received.
10	Debug messages A, B received.
11	Debug messages A, B, C received, DMA request is set.

Notes:

- *Debug message is used for Debug on CAN feature.*
- *Debug on CAN feature is not supported for TRAVEO™ T1G Platform.*

[bit29:26] Reserved: Reserved Bits

When writing, always write "0". The read value is undefined.

[bit25] RF1L: Rx FIFO 1 Message Lost

This bit is a copy of interrupt flag IR.RF1L (Rx FIFO 1 Message Lost).

When IR.RF1L is reset, this bit is also reset.

Bit	Description
0	No Rx FIFO 1 message lost.
1	Rx FIFO 1 message lost. Also set after write attempt to Rx FIFO 1 of size zero.

Note:

- *Overwriting a message when the FIFO is in overwrite mode (RXF1C.F1OM = "1") will not set this flag.*

[bit24] F1F: Rx FIFO 1 Full

Bit	Description
0	Rx FIFO 1 not full.
1	Rx FIFO 1 full.

[bit23:22] Reserved: Reserved Bits

When writing, always write "0". The read value is undefined.

[bit21:16] F1PI[5:0]: Rx FIFO 1 Put Index

F1PI[5:0]	Description
0-63	Rx FIFO 1 write index pointer, range 0 to 63.

[bit15:14] Reserved: Reserved Bits

When writing, always write "0". The read value is undefined.

[bit13:8] F1GI[5:0]: Rx FIFO 1 Get Index

F1GI[5:0]	Description
0-63	Rx FIFO 1 read index pointer, range 0 to 63.

[bit7] Reserved: Reserved Bit

When writing, always write "0". The read value is undefined.

[bit6:0] F1FL[6:0]: Rx FIFO 1 Fill Level

F1FL[6:0]	Description
0-64	Number of elements stored in Rx FIFO 1, range 0 to 64.

5.32. Rx FIFO 1 Acknowledge (MCG_CANFDx_RXF1A, CPG_CANFDx_RXF1A)

The Rx FIFO 1 Acknowledge is used to acknowledge that the CPU has read a message or a sequence of messages from the Rx FIFO 1 to indicate to the CAN FD Controller that the corresponding Message RAM area may be released. See "3.6. FIFO Acknowledge Handling" for details.

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

Bit	23	22	21	20	19	18	17	16
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

Bit	15	14	13	12	11	10	9	8
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

Bit	7	6	5	4	3	2	1	0
Field	Reserved		F1AI[5:0]					
R/W Attribute	R0,W0		R/W					
Protection Attribute	-		-					
Initial value	00		000000					

[bit31:6] Reserved: Reserved Bits

When writing, always write "0". When reading, "0" is always read.

[bit5:0] F1AI[5:0]: Rx FIFO 1 Acknowledge Index

After the CPU has read a message or a sequence of messages from Rx FIFO 1, it has to write the buffer index of the last element read from Rx FIFO 1 to this field (F1AI[5:0]). This will set the Rx FIFO 1 Get Index RXF1S.F1GI[5:0] to F1AI[5:0] + 1 and update the FIFO 1 Fill Level RXF1S.F1FL[6:0].

5.33. Rx Buffer/FIFO Element Size Configuration (MCG_CANFDx_RXESC, CPG_CANFDx_RXESC)

Configures the number of data bytes belonging to an Rx Buffer and FIFO element. Data field sizes > 8 bytes are intended for CAN FD operation only.

Write access to the Rx Buffer/FIFO Element Size Configuration (RXESC) is possible only when the Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) bits are both "1".

Note:

- *In case the data field size of an accepted CAN frame exceeds the data field size configured for the matching Rx Buffer or Rx FIFO, only the number of bytes as configured by this register (i.e. RXESC) are stored to the Rx Buffer resp. Rx FIFO element. The rest of the frame's data field is ignored.*

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

Bit	23	22	21	20	19	18	17	16
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

Bit	15	14	13	12	11	10	9	8
Field	Reserved					RBDS[2:0]		
R/W Attribute	R0,W0					R/W		
Protection Attribute	-					-		
Initial value	00000					000		

Bit	7	6	5	4	3	2	1	0
Field	Reserved	F1DS[2:0]			Reserved	F0DS[2:0]		
R/W Attribute	R0,W0	R/W			R0,W0	R/W		
Protection Attribute	-	-			-	-		
Initial value	0	000			0	000		

[bit31:11] Reserved: Reserved Bits

When writing, always write "0". When reading, "0" is always read.

[bit10:8] RBDS[2:0]: Rx Buffer Data Field Size

RBDS[2:0]	Description
000	8 byte data field.
001	12 byte data field.
010	16 byte data field.
011	20 byte data field.
100	24 byte data field.
101	32 byte data field.
110	48 byte data field.
111	64 byte data field.

[bit7] Reserved: Reserved Bit

When writing, always write "0". When reading, "0" is always read.

[bit6:4] F1DS[2:0]: Rx FIFO 1 Data Field Size

F1DS[2:0]	Description
000	8 byte data field.
001	12 byte data field.
010	16 byte data field.
011	20 byte data field.
100	24 byte data field.
101	32 byte data field.
110	48 byte data field.
111	64 byte data field.

[bit3] Reserved: Reserved Bit

When writing, always write "0". When reading, "0" is always read.

[bit2:0] F0DS[2:0]: Rx FIFO 0 Data Field Size

F0DS[2:0]	Description
000	8 byte data field.
001	12 byte data field.
010	16 byte data field.
011	20 byte data field.
100	24 byte data field.
101	32 byte data field.
110	48 byte data field.
111	64 byte data field.

5.34. Tx Buffer Configuration (MCG_CANFDx_TXBC, CPG_CANFDx_TXBC)

Settings for Tx Buffers stored in the Message RAM.

Write access to the Tx Buffer Configuration (TXBC) is possible only when the Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) bits are both "1".

Notes:

- *Be aware that the sum of Transmit FIFO/Queue Size (TXBC.TFQS[5:0]) and Number of Dedicated Transmit Buffers (TXBC.NDTB[5:0]) may be not greater than 32. There is no check for erroneous configurations.*
- *The Tx Buffers section in the Message RAM starts with the dedicated Tx Buffers.*
- *Don't use combination Tx FIFO and Dedicated Tx Buffers. In case of setting "1" to Tx FIFO/Queue Mode (TXBC.TFQM), set 0 to TXBC.NDTB[5:0].*

bit	31	30	29	28	27	26	25	24
Field	Reserved	TFQM	TFQS[5:0]					
R/W Attribute	R0,W0	R/W	R/W					
Protection Attribute	-	-	-					
Initial value	0	0	000000					

bit	23	22	21	20	19	18	17	16
Field	Reserved		NDTB[5:0]					
R/W Attribute	R0,W0		R/W					
Protection Attribute	-		-					
Initial value	00		000000					

bit	15	14	13	12	11	10	9	8
Field	TBSA[15:8]							
R/W Attribute	R/W							
Protection Attribute	-							
Initial value	00000000							

bit	7	6	5	4	3	2	1	0
Field	TBSA[7:2]						Reserved	
R/W Attribute	R/W						R0,W0	
Protection Attribute	-						-	
Initial value	000000						00	

[bit31] Reserved: Reserved Bit

When writing, always write "0". When reading, "0" is always read.

[bit30] TFQM: Tx FIFO/Queue Mode

Bit	Description
0	Tx FIFO operation.
1	Tx Queue operation.

[bit29:24] TFQS[5:0]: Transmit FIFO/Queue Size

TFQS[5:0]	Description
0	No Tx FIFO/Queue.
1-32	Number of Tx Buffers used for Tx FIFO/Queue.
>32	Values greater than 32 are interpreted as 32.

[bit23:22] Reserved: Reserved Bits

When writing, always write "0". When reading, "0" is always read.

[bit21:16] NDTB[5:0]: Number of Dedicated Transmit Buffers

NDTB[5:0]	Description
0	No Dedicated Tx Buffers.
1-32	Number of Dedicated Tx Buffers.
>32	Values greater than 32 are interpreted as 32.

[bit15:2] TBSA[15:2]: Tx Buffers Start Address

Start address of Tx Buffers section in Message RAM (32-bit word address, see Figure 6-1).

[bit1:0] Reserved: Reserved Bits

When writing, always write "0". When reading, "0" is always read.

5.35. Tx FIFO/Queue Status (MCG_CANFDx_TXFQS, CPG_CANFDx_TXFQS)

The Tx FIFO/Queue status is related to the pending Tx requests listed in the Tx Buffer Request Pending register TXBRP. Therefore the effect of Add/Cancellation requests may be delayed due to a running Tx scan (TXBRP not yet updated).

Note:

- *In case of mixed configurations where dedicated Tx Buffers are combined with a Tx FIFO or a Tx Queue, the Put and Get Indices (TXFQS.TFQPI[4:0] resp. TXFQS.TFGI[4:0]) indicate the number of the Tx Buffer starting with the first dedicated Tx Buffers.*

Example:

- *For a configuration of 12 dedicated Tx Buffers and a Tx FIFO of 20 Buffers a Put Index of 15 points to the fourth buffer of the Tx FIFO.*

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	RX,W0							
Protection Attribute	-							
Initial value	0xXX							

bit	23	22	21	20	19	18	17	16
Field	Reserved		TFQF	TFQPI[4:0]				
R/W Attribute	RX,W0		R,WX	R,WX				
Protection Attribute	-		-	-				
Initial value	XX		0	00000				

Bit	15	14	13	12	11	10	9	8
Field	Reserved			TFGI[4:0]				
R/W Attribute	RX,W0			R,WX				
Protection Attribute	-			-				
Initial value	XXX			00000				

Bit	7	6	5	4	3	2	1	0
Field	Reserved		TFFL[5:0]					
R/W Attribute	RX,W0		R,WX					
Protection Attribute	-		-					
Initial value	XX		000000					

[bit31:22] Reserved: Reserved Bits

When writing, always write "0". The read value is undefined.

[bit21] TFQF: Tx FIFO/Queue Full

Bit	Description
0	Tx FIFO/Queue not full.
1	Tx FIFO/Queue full.

[bit20:16] TFQPI[4:0]: Tx FIFO/Queue Put Index

TFQPI[4:0]	Description
0-31	Tx FIFO/Queue write index pointer, range 0 to 31.

[bit15:13] Reserved: Reserved Bits

When writing, always write "0". The read value is undefined.

[bit12:8] TFGI[4:0]: Tx FIFO Get Index

TFGI[4:0]	Description
0-31	Tx FIFO read index pointer, range 0 to 31. Read as zero when Tx Queue operation is configured (Tx FIFO/Queue Mode TXBC.TFQM = "1").

[bit7:6] Reserved: Reserved Bits

When writing, always write "0". The read value is undefined.

[bit5:0] TFFL[5:0]: Tx FIFO Free Level

TFFL[5:0]	Description
0-32	Number of consecutive free Tx FIFO elements starting from the Tx FIFO Get Index (TXFQS.TFGI[4:0]), range 0 to 32. Read as zero when Tx Queue operation is configured (Tx FIFO/Queue Mode TXBC.TFQM = "1").

5.36. Tx Buffer Element Size Configuration (MCG_CANFDx_TXESC, CPG_CANFDx_TXESC)

Configures the number of data bytes belonging to a Tx Buffer element. Data field sizes > 8 bytes are intended for CAN FD operation only.

Write access to the Tx Buffer Element Size Configuration (TXESC) is possible only when the Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) bits are both "1".

Note:

- *In case the data length code DLC of a Tx Buffer element is configured to a value higher than the Tx Buffer data field size TXESC.TBDS[2:0], the bytes not defined by the Tx Buffer are transmitted as "0xCC" (padding bytes).*

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

Bit	23	22	21	20	19	18	17	16
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

Bit	15	14	13	12	11	10	9	8
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

Bit	7	6	5	4	3	2	1	0
Field	Reserved					TBDS[2:0]		
R/W Attribute	R0,W0					R/W		
Protection Attribute	-					-		
Initial value	00000					000		

[bit31:3] Reserved: Reserved Bits

When writing, always write "0". When reading, "0" is always read.

[bit2:0] TBDS[2:0]: Tx Buffer Data Field Size

TBDS[2:0]	Description
000	8 byte data field.
001	12 byte data field.
010	16 byte data field.
011	20 byte data field.
100	24 byte data field.
101	32 byte data field.
110	48 byte data field.
111	64 byte data field.

5.37. Tx Buffer Request Pending (MCG_CANFDx_TXBRP, CPG_CANFDx_TXBRP)

Tx Buffer Request Pending holds the status of the transmission requests of each corresponding Tx Buffer.

bit	31	30	29	28	27	26	25	24
Field	TRP31	TRP30	TRP29	TRP28	TRP27	TRP26	TRP25	TRP24
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	TRP23	TRP22	TRP21	TRP20	TRP19	TRP18	TRP17	TRP16
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	TRP15	TRP14	TRP13	TRP12	TRP11	TRP10	TRP9	TRP8
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	TRP7	TRP6	TRP5	TRP4	TRP3	TRP2	TRP1	TRP0
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

[bit31:0] TRP[31:0]: Transmission Request Pending

Each Tx Buffer has its own Transmission Request Pending bit. The bits are set via register TXBAR (Tx Buffer Add Request). The bits are reset after a requested transmission has completed or has been cancelled via register TXBCR (Tx Buffer Cancellation Request).

TXBRP bits are set only for those Tx Buffers that are configured by the TXBC.TFQS[5:0] (Transmit FIFO/Queue Size) and TXBC.NDTB[5:0] (Number of Dedicated Transmit Buffers) fields. After a TXBRP bit has been set, a Tx scan (see "3.5. Tx Handling") is started to check for the pending Tx request with the highest priority (Tx Buffer with lowest Message ID).

Bit	Description
0	No transmission request pending.
1	Transmission request pending.

Note:

– *TXBRP bits which are set while a Tx scan is in progress are not considered during this particular Tx scan. In case a cancellation is requested for such a Tx Buffer, this Add Request is cancelled immediately, the corresponding TXBRP bit is reset.*

5.38. Tx Buffer Add Request (MCG_CANFDx_TXBAR, CPG_CANFDx_TXBAR)

Tx Buffer Add Request is used to request the transmission of each corresponding Tx Buffer.

bit	31	30	29	28	27	26	25	24
Field	AR31	AR30	AR29	AR28	AR27	AR26	AR25	AR24
R/W Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	AR23	AR22	AR21	AR20	AR19	AR18	AR17	AR16
R/W Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	AR15	AR14	AR13	AR12	AR11	AR10	AR9	AR8
R/W Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	AR7	AR6	AR5	AR4	AR3	AR2	AR1	AR0
R/W Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

[bit31:0] AR[31:0]: Add Request

Each Tx Buffer has its own Add Request bit. Writing a "1" will set the corresponding Add Request bit; writing a "0" has no impact. This enables the CPU to set transmission requests for multiple Tx Buffers with one write to TXBAR.

TXBAR bits are set only for those Tx Buffers that are configured by the TXBC.TFQS[5:0] (Transmit FIFO/Queue Size) and TXBC.NDTB[5:0] (Number of Dedicated Transmit Buffers) fields.

When no Tx scan is running, the bits are reset immediately, else the bits remain set until the Tx scan process has completed.

Bit	Description
0	No transmission request added.
1	Transmission requested added.

Note:

- *If an add request is applied for a Tx Buffer with pending transmission request (corresponding TXBRP bit already set), this add request is ignored.*

5.39. Tx Buffer Cancellation Request (MCG_CANFDx_TXBCR, CPG_CANFDx_TXBCR)

Used to cancel transmission requests of each corresponding Tx Buffer.

bit	31	30	29	28	27	26	25	24
Field	CR31	CR30	CR29	CR28	CR27	CR26	CR25	CR24
R/W Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	CR23	CR22	CR21	CR20	CR19	CR18	CR17	CR16
R/W Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	CR15	CR14	CR13	CR12	CR11	CR10	CR9	CR8
R/W Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	CR7	CR6	CR5	CR4	CR3	CR2	CR1	CR0
R/W Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

[bit31:0] CR[31:0]: Cancellation Request

Each Tx Buffer has its own Cancellation Request bit. Writing a "1" will set the corresponding Cancellation Request bit; writing a "0" has no impact. This enables the CPU to set cancellation requests for multiple Tx Buffers with one write to TXBCR.

TXBCR bits are set only for those Tx Buffers that are configured by the TXBC.TFQS[5:0] (Transmit FIFO/Queue Size) and TXBC.NDTB[5:0] (Number of Dedicated Transmit Buffers) fields. The bits remain set until the corresponding bit of TXBRP (Tx Buffer Request Pending) is reset.

Bit	Description
0	No cancellation pending.
1	Cancellation pending.

5.40. Tx Buffer Transmission Occurred (MCG_CANFDx_TXBTO, CPG_CANFDx_TXBTO)

Displays the status of whether the corresponding Tx Buffer has been transmitted or not.

bit	31	30	29	28	27	26	25	24
Field	TO31	TO30	TO29	TO28	TO27	TO26	TO25	TO24
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	TO23	TO22	TO21	TO20	TO19	TO18	TO17	TO16
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	TO15	TO14	TO13	TO12	TO11	TO10	TO9	TO8
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	TO7	TO6	TO5	TO4	TO3	TO2	TO1	TO0
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

[bit31:0] TO[31:0]: Transmission Occurred

Each Tx Buffer has its own Transmission Occurred bit.

The bits are set when the corresponding TXBRP (Tx Buffer Request Pending) bit is cleared after a successful transmission.

The bits are reset when a new transmission is requested by writing a "1" to the corresponding bit of register TXBAR (Tx Buffer Add Request).

Bit	Description
0	No transmission occurred.
1	Transmission occurred.

5.41. Tx Buffer Cancellation Finished (MCG_CANFDx_TXBCF, CPG_CANFDx_TXBCF)

Signals whether the cancellation request of the corresponding Tx Buffer has been successful or not.

bit	31	30	29	28	27	26	25	24
Field	CF31	CF30	CF29	CF28	CF27	CF26	CF25	CF24
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	CF23	CF22	CF21	CF20	CF19	CF18	CF17	CF16
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	CF15	CF14	CF13	CF12	CF11	CF10	CF9	CF8
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	CF7	CF6	CF5	CF4	CF3	CF2	CF1	CF0
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

[bit31:0] CF[31:0]: Cancellation Finished

Each Tx Buffer has its own Cancellation Finished bit.

The bits are set when the corresponding TXBRP (Tx Buffer Request Pending) bit is cleared after a cancellation was requested via TXBCR (Tx Buffer Cancellation Request). In case the corresponding TXBRP bit was not set at the point of cancellation, CF is set immediately. CF is also set for an unsuccessful transmission when in DAR mode.

The bits are reset when a new transmission is requested by writing a "1" to the corresponding bit of register TXBAR (Tx Buffer Add Request).

Bit	Description
0	No transmit buffer cancellation.
1	Transmit buffer cancellation finished.

5.42. Tx Buffer Transmission Interrupt Enable (MCG_CANFDx_TXBTIE, CPG_CANFDx_TXBTIE)

The settings in the Tx Buffer Transmission Interrupt Enable determine which Tx Buffer will assert an interrupt upon transmission.

bit	31	30	29	28	27	26	25	24
Field	TIE31	TIE30	TIE29	TIE28	TIE27	TIE26	TIE25	TIE24
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	TIE23	TIE22	TIE21	TIE20	TIE19	TIE18	TIE17	TIE16
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	TIE15	TIE14	TIE13	TIE12	TIE11	TIE10	TIE9	TIE8
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	TIE7	TIE6	TIE5	TIE4	TIE3	TIE2	TIE1	TIE0
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

[bit31:0] TIE[31:0]: Transmission Interrupt Enable

Each Tx Buffer has its own Transmission Interrupt Enable bit.

Bit	Description
0	Transmission interrupt disabled.
1	Transmission interrupt enable.

5.43. Tx Buffer Cancellation Finished Interrupt Enable (MCG_CANFDx_TXBCIE, CPG_CANFDx_TXBCIE)

The settings in the Tx Buffer Cancellation Finished Interrupt Enable determine which Tx Buffer will assert an interrupt upon completion of a transmission cancellation request.

bit	31	30	29	28	27	26	25	24
Field	CFIE31	CFIE30	CFIE29	CFIE28	CFIE27	CFIE26	CFIE25	CFIE24
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	CFIE23	CFIE22	CFIE21	CFIE20	CFIE19	CFIE18	CFIE17	CFIE16
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	CFIE15	CFIE14	CFIE13	CFIE12	CFIE11	CFIE10	CFIE9	CFIE8
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	CFIE7	CFIE6	CFIE5	CFIE4	CFIE3	CFIE2	CFIE1	CFIE0
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

[bit31:0] CFIE[31:0]: Cancellation Finished Interrupt Enable

Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit.

Bit	Description
0	Cancellation finished interrupt disabled.
1	Cancellation finished interrupt enabled.

5.44. Tx Event FIFO Configuration (MCG_CANFDx_TXEFC, CPG_CANFDx_TXEFC)

Settings for the Tx Event FIFO.

Write access to the Tx Event FIFO Configuration (TXEFC) is possible only when the Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) bits are both "1".

bit	31	30	29	28	27	26	25	24
Field	Reserved		EFWM[5:0]					
R/W Attribute	R0,W0		R/W					
Protection Attribute	-		-					
Initial value	00		000000					

Bit	23	22	21	20	19	18	17	16
Field	Reserved		EFS[5:0]					
R/W Attribute	R0,W0		R/W					
Protection Attribute	-		-					
Initial value	00		000000					

Bit	15	14	13	12	11	10	9	8
Field	EFSA[15:8]							
R/W Attribute	R/W							
Protection Attribute	-							
Initial value	00000000							

Bit	7	6	5	4	3	2	1	0
Field	EFSA[7:2]						Reserved	
R/W Attribute	R/W						R0,W0	
Protection Attribute	-						-	
Initial value	000000						00	

[bit31:30] Reserved: Reserved Bits

When writing, always write "0". When reading, "0" is always read.

[bit29:24] EFWM[5:0]: Event FIFO Watermark

EFWM[5:0]	Description
0	Watermark interrupt disabled.
1-32	Level for Tx Event FIFO Watermark Reached interrupt (IR.TEFW).
>32	Watermark interrupt disabled.

[bit23:22] Reserved: Reserved Bits

When writing, always write "0". When reading, "0" is always read.

[bit21:16] EFS[5:0]: Event FIFO Size

The Tx Event FIFO elements are indexed from 0 to EFS[5:0] - 1.

EFS[5:0]	Description
0	Tx Event FIFO disabled.
1-32	Number of Tx Event FIFO elements.
>32	Values greater than 32 are interpreted as 32.

[bit15:2] EFSA[15:2]: Event FIFO Start Address

Start address of Tx Event FIFO in Message RAM (32-bit word address, see Figure 6-1).

[bit1:0] Reserved: Reserved Bits

When writing, always write "0". When reading, "0" is always read.

5.45. Tx Event FIFO Status (MCG_CANFDx_TXEFS, CPG_CANFDx_TXEFS)

Status of the Tx Event FIFO.

bit	31	30	29	28	27	26	25	24
Field	Reserved						TEFL	EFF
R/W Attribute	RX,W0						R,WX	R,WX
Protection Attribute	-						-	-
Initial value	XXXXXX						0	0

bit	23	22	21	20	19	18	17	16
Field	Reserved			EFPI[4:0]				
R/W Attribute	RX,W0			R,WX				
Protection Attribute	-			-				
Initial value	XXX			00000				

Bit	15	14	13	12	11	10	9	8
Field	Reserved			EFGI[4:0]				
R/W Attribute	RX,W0			R,WX				
Protection Attribute	-			-				
Initial value	XXX			00000				

bit	7	6	5	4	3	2	1	0
Field	Reserved		EFFL[5:0]					
R/W Attribute	RX,W0		R,WX					
Protection Attribute	-		-					
Initial value	XX		000000					

[bit31:26] Reserved: Reserved Bits

When writing, always write "0". The read value is undefined.

[bit25] TEFL: Tx Event FIFO Element Lost

This bit is a copy of interrupt flag IR.TEFL (Tx Event FIFO Element Lost). When IR.TEFL is reset, this bit is also reset.

Bit	Description
0	No Tx Event FIFO element lost.
1	Tx Event FIFO element lost. Also set after write attempt to Tx Event FIFO of size zero.

[bit24] EFF: Event FIFO Full

Bit	Description
0	Tx Event FIFO not full.
1	Tx Event FIFO full.

[bit23:21] Reserved: Reserved Bits

When writing, always write "0". The read value is undefined.

[bit20:16] EFPI[4:0]: Event FIFO Put Index

EFPI[4:0]	Description
0-31	Tx Event FIFO write index pointer, range 0 to 31.

[bit15:13] Reserved: Reserved Bits

When writing, always write "0". The read value is undefined.

[bit12:8] EFGI[4:0]: Event FIFO Get Index

EFGI[4:0]	Description
0-31	Tx Event FIFO read index pointer, range 0 to 31.

[bit7:6] Reserved: Reserved Bits

When writing, always write "0". The read value is undefined.

[bit5:0] EFFL[5:0]: Event FIFO Fill Level

EFFL[5:0]	Description
0-32	Number of elements stored in Tx Event FIFO, range 0 to 32.

5.46. Tx Event FIFO Acknowledge (MCG_CANFDx_TXEFA, CPG_CANFDx_TXEFA)

The Tx Event FIFO Acknowledge is used to acknowledge that the CPU has read an event from the Tx Event FIFO to indicate to the CAN FD Controller that the corresponding Message RAM area may be released. See "3.6. FIFO Acknowledge Handling" for details.

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

Bit	23	22	21	20	19	18	17	16
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

Bit	15	14	13	12	11	10	9	8
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial value	0x00							

Bit	7	6	5	4	3	2	1	0
Field	Reserved			EFAI[4:0]				
R/W Attribute	R0,W0			R/W				
Protection Attribute	-			-				
Initial value	000			00000				

[bit31:5] Reserved: Reserved Bits

When writing, always write "0". When reading, "0" is always read.

[bit4:0] EFAI[4:0]: Event FIFO Acknowledge Index

After the CPU has read an element or a sequence of elements from the Tx Event FIFO, it has to write the index of the last element read from Tx Event FIFO to EFAI[4:0]. This will set the Tx Event FIFO Get Index TXEFS.EFGI[4:0] to EFAI[4:0] + 1 and update the Event FIFO 0 Fill Level TXEFS.EFFL[5:0].

6. Message RAM

The Message RAM stores Rx/Tx messages and filter configurations.

Note:

- *The Message RAM should be zeroized before configuration of the CAN FD Controller in order to prevent Message RAM bit errors when reading uninitialized words, and also to avoid unexpected filter element configurations in the Message RAM.*

6.1.Message RAM Configuration

6.2.Rx Buffer and FIFO Element

6.3.Tx Buffer Element

6.4.Tx Event FIFO Element

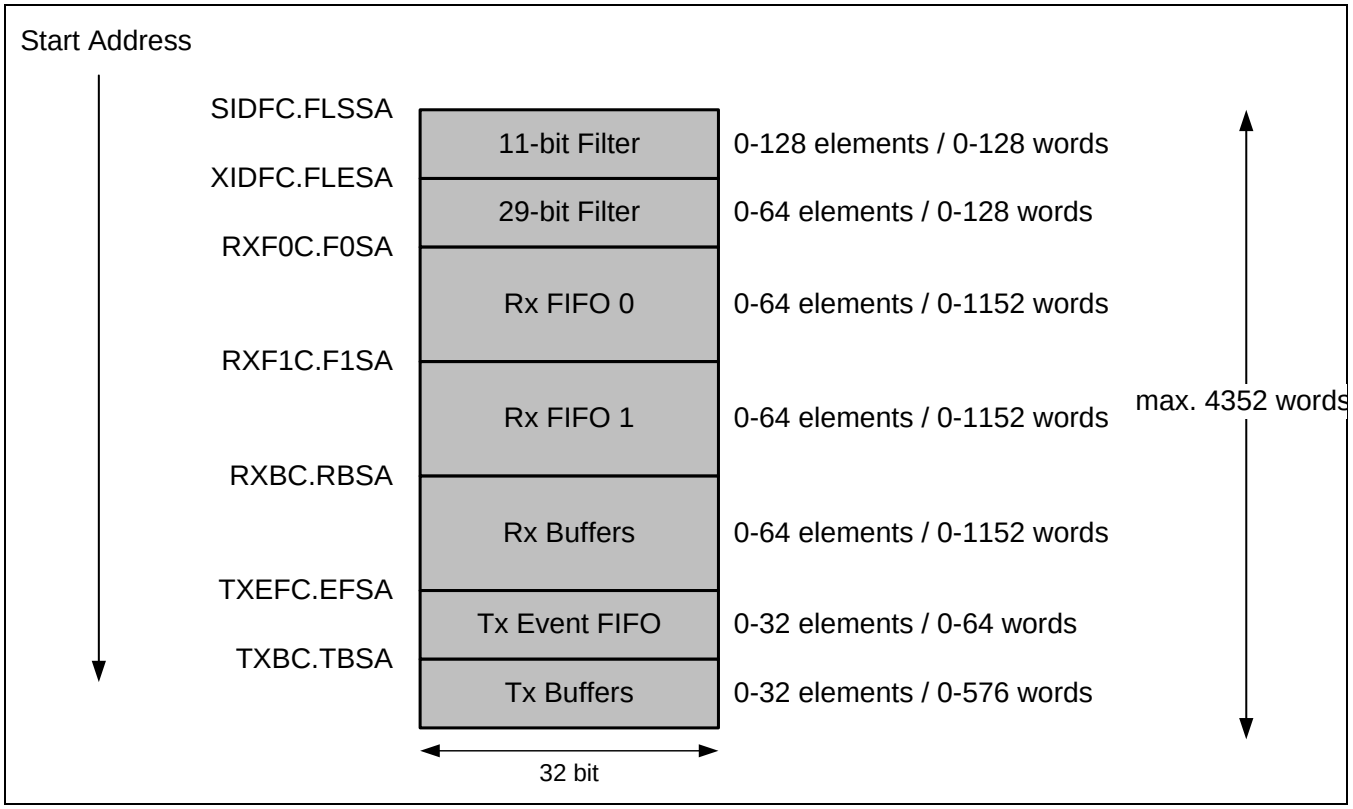
6.5.Standard Message ID Filter Element

6.6.Extended Message ID Filter Element

6.1. Message RAM Configuration

The Message RAM has a width of 32 bits. The CAN FD Controller can be configured to allocate up to 4352 words in the Message RAM (note that the number of words that can be used will be limited by the size of the actual Message RAM). It is not necessary to configure each of the sections listed in Figure 6-1, nor is there any restriction with respect to the sequence of the sections.

Figure 6-1 Message RAM Configuration



The CAN FD Controller addresses the Message RAM in 32-bit words, not single bytes. The configurable start addresses are 32-bit word addresses i.e. only bits 15 to 2 are evaluated, the two least significant bits are ignored.

Notes:

- *The CAN FD Controller does not check for erroneous configuration of the Message RAM. Especially the configuration of the start addresses of the different sections and the number of elements of each section has to be done carefully to avoid falsification or loss of data.*
- *Message RAM accesses by the CPU will take two to four Bus clock cycles.*

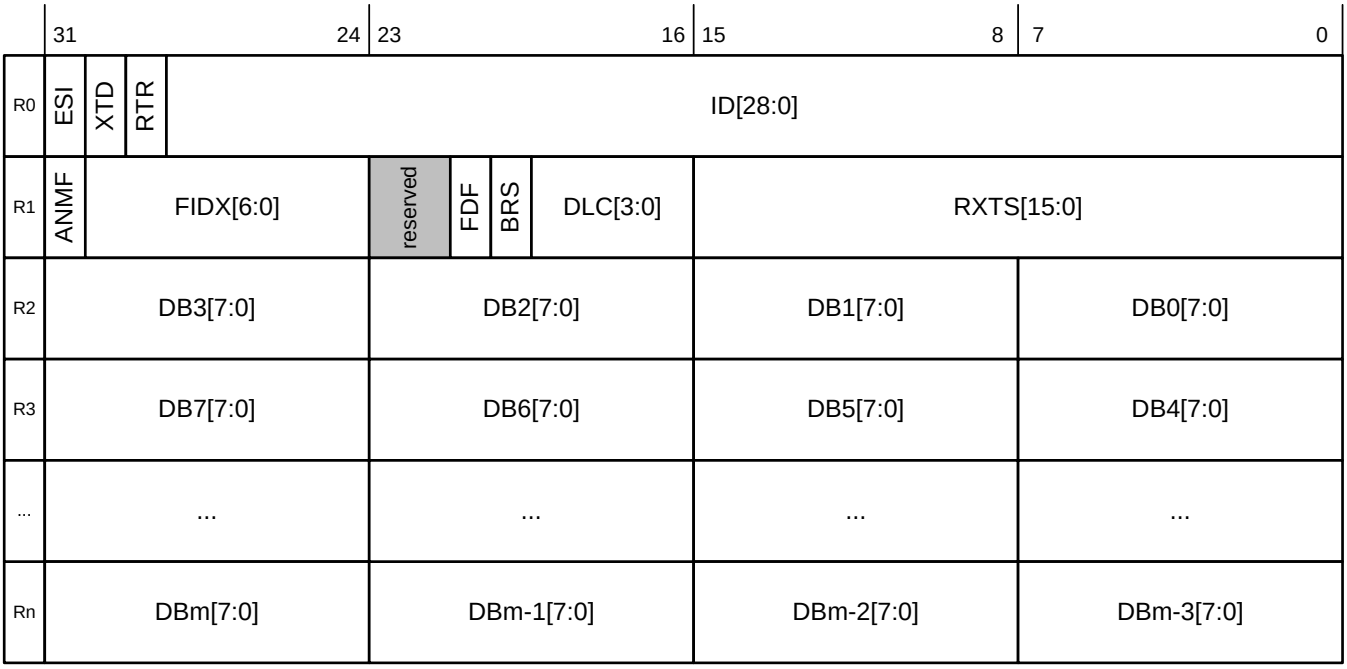
List of Message RAM Elements

Abbreviation	Reference
Rx Buffer and FIFO Element	6.2
Tx Buffer Element	6.3
Tx Event FIFO Element	6.4
Standard Message ID Filter Element	6.5
Extended Message ID Filter Element	6.6

6.2. Rx Buffer and FIFO Element

An Rx Buffer and FIFO Element is a block of 32-bit words that holds the data and status of a received frame that was stored in the Message RAM.

Up to 64 Rx Buffers and two Rx FIFOs can be configured in the Message RAM. Each Rx FIFO section can be configured to store up to 64 received messages. The structure of an Rx Buffer and FIFO element is shown in the figure below. The element size can be configured for storage of CAN FD messages with up to 64 bytes data field via register RXESC (Rx Buffer/FIFO Element Size Configuration).



R0 [bit31] ESI: Error State Indicator

Bit	Description
0	Transmitting node is error active.
1	Transmitting node is error passive.

R0 [bit30] XTD: Extended Identifier

Signals to the CPU whether the received frame has a standard or extended identifier.

Bit	Description
0	11-bit standard identifier.
1	29-bit extended identifier.

R0 [bit29] RTR: Remote Transmission Request

Signals to the CPU whether the received frame is a data frame or a remote frame.

Bit	Description
0	Received frame is a data frame.
1	Received frame is a remote frame.

Note:

- *There are no remote frames in CAN FD format. In CAN FD frames (FDF = "1"), the dominant RRS (Remote Request Substitution) bit replaces bit RTR (Remote Transmission Request).*

R0 [bit28:0] ID[28:0]: Identifier

Standard or extended identifier depending on bit XTD. A standard identifier is stored into ID[28:18].

R1 [bit31] ANMF: Accepted Non-matching Frame

Acceptance of non-matching frames may be enabled via GFC.ANFS[1:0] (Accept Non-matching Frames Standard) and GFC.ANFE[1:0] (Accept Non-matching Frames Extended).

Bit	Description
0	Received frame matching filter index FIDX.
1	Received frame did not match any Rx filter element.

R1 [bit30:24] FIDX[6:0]: Filter Index

FIDX[6:0]	Description
0-127	Index of matching Rx acceptance filter element (invalid if ANMF = "1"). Range is 0 to List Size Standard/Extended minus 1 (i.e. SIDFC.LSS - 1 resp. XIDFC.LSE - 1).

R1 [bit23:22] Reserved: Reserved Bits

When writing, always write "0". The read value is undefined.

R1 [bit21] FDF: Extended Data Length

Bit	Description
0	Classic CAN frame format.
1	CAN FD frame format.

R1 [bit20] BRS: Bit Rate Switch

Bit	Description
0	Frame received without bit rate switching.
1	Frame received with bit rate switching.

R1 [bit19:16] DLC[3:0]: Data Length Code

DLC[3:0]	Description
0-8	Classic CAN + CAN FD: received frame has 0-8 data bytes.
9-15	Classic CAN: received frame has 8 data bytes. CAN FD: received frame has 12/16/20/24/32/48/64 data bytes. See Table 3-1 for details.

R1 [bit15:0] RXTS[15:0]: Rx Timestamp

Timestamp Counter value captured on start of frame reception. Resolution depending on configuration of the Timestamp Counter Prescaler TSCC.TCP[3:0].

R2 [bit31:24] DB3[7:0] : Data Byte 3

R2 [bit23:16] DB2[7:0] : Data Byte 2

R2 [bit15:8] DB1[7:0] : Data Byte 1

R2 [bit7:0] DB0[7:0] : Data Byte 0

R3 [bit31:24] DB7[7:0] : Data Byte 7

R3 [bit23:16] DB6[7:0] : Data Byte 6

R3 [bit15:8] DB5[7:0] : Data Byte 5

R3 [bit7:0] DB4[7:0] : Data Byte 4

… … …

Rn [bit31:24] DBm[7:0]: Data Byte m

Rn [bit23:16] DBm-1[7:0]: Data Byte m-1

Rn [bit15:8] DBm-2[7:0]: Data Byte m-2

Rn [bit7:0] DBm-3[7:0]: Data Byte m-3

Notes:

- *Depending on the configuration of the element size (defined by Rx Buffer/FIFO Element Size Configuration (RXESC)), Rn will vary from n = 3 to 17.*
- *m is a function of n, m = (n - 1) × 4 - 1.*
- *The number of valid data bytes are defined by the Data Length Code.*

6.3. Tx Buffer Element

A Tx Buffer Element is a block of 32-bit words stored in the Message RAM that holds data and control information of a frame to be transmitted by the CAN FD Controller.

The Tx Buffers section can be configured to hold dedicated Tx Buffers as well as a Tx FIFO/Tx Queue. In case that the Tx Buffers section is shared by dedicated Tx buffers and a Tx FIFO/Tx Queue, the dedicated Tx Buffers start at the beginning of the Tx Buffers section followed by the buffers assigned to the Tx FIFO or Tx Queue. The Tx Handler distinguishes between dedicated Tx Buffers and Tx FIFO/Tx Queue by evaluating the Tx Buffer configuration TXBC.TFQS[5:0] (Transmit FIFO/Queue Size) and TXBC.NDTB[5:0] (Number of Dedicated Transmit Buffers). The element size can be configured for storage of CAN FD messages with up to 64 bytes data field via register TXESC (Tx Buffer Element Size Configuration).

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T0	ESI	XTD	RTR	ID[28:0]																												
T1	MM[7:0]								EFC	reserved	FDF	BRS	DLC[3:0]				reserved															
T2	DB3[7:0]								DB2[7:0]								DB1[7:0]								DB0[7:0]							
T3	DB7[7:0]								DB6[7:0]								DB5[7:0]								DB4[7:0]							
...	...								...								...								...							
Tn	DBm[7:0]								DBm-1[7:0]								DBm-2[7:0]								DBm-3[7:0]							

T0 [bit31] ESI: Error State Indicator

Bit	Description
0	ESI bit in CAN FD format depends only on error passive flag.
1	ESI bit in CAN FD format transmitted recessive.

Note:

- *The ESI bit of the transmit buffer is ORed with the error passive flag to decide the value of the ESI bit in the transmitted FD frame. As required by the CAN FD protocol specification, an error active node may optionally transmit the ESI bit recessive, but an error passive node will always transmit the ESI bit recessive.*

T0 [bit30] XTD: Extended Identifier

Bit	Description
0	11-bit standard identifier.
1	29-bit extended identifier.

T0 [bit29] RTR: Remote Transmission Request

Bit	Description
0	Transmit data frame.
1	Transmit remote frame.

Note:

- *When RTR = "1", the CAN FD Controller transmits a remote frame according to ISO11898-1, even if FD Operation Enable (CCCR.FDOE) enables the transmission in CAN FD format.*

T0 [bit28:0] ID[28:0]: Identifier

Standard or extended identifier depending on bit XTD. A standard identifier has to be written to ID[28:18].

T1 [bit31:24] MM[7:0]: Message Marker

Written by CPU during Tx Buffer configuration. Copied into Tx Event FIFO element for identification of Tx message status.

T1 [bit23] EFC: Event FIFO Control

Bit	Description
0	Don’t store Tx events.
1	Store Tx events.

T1 [bit22] Reserved: Reserved Bit

When writing, always write "0". The read value is undefined.

T1 [bit21] FDF: FD Format

Bit	Description
0	Frame transmitted in Classic CAN format.
1	Frame transmitted in CAN FD format.

T1 [bit20] BRS: Bit Rate Switching

Bit	Description
0	CAN FD frames transmitted without bit rate switching.
1	CAN FD frames transmitted with bit rate switching.

Note:

- *Bits ESI, FDF, and BRS are only evaluated when CAN FD operation is enabled CCCR.FDOE = "1". Bit BRS is only evaluated when in addition CCCR.BRSE = "1". See Table 3-5 for details of bits FDF and BRS.*

T1 [bit19:16] DLC[3:0]: Data Length Code

DLC[3:0]	Description
0-8	Classic CAN + CAN FD: transmit frame has 0-8 data bytes.
9-15	Classic CAN: transmit frame has 8 data bytes. CAN FD: transmit frame has 12/16/20/24/32/48/64 data bytes. See Table 3-1 for details.

T1 [bit15:0] Reserved: Reserved Bits

When writing, always write "0". The read value is undefined.

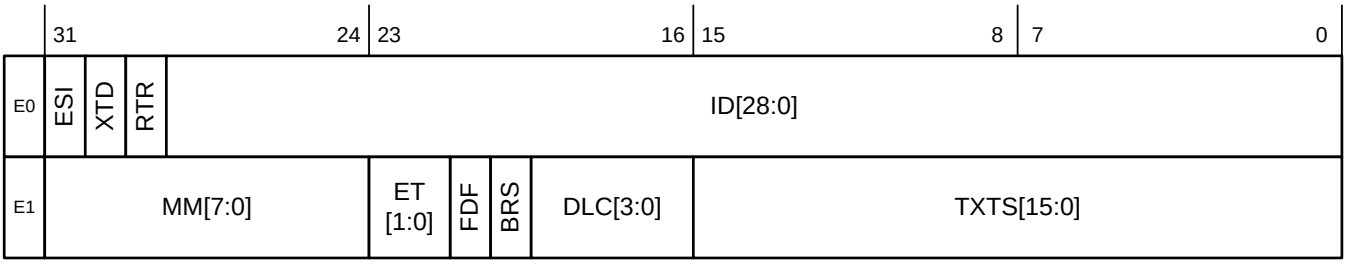
T2 [bit31:24]	**DB3[7:0] :**	**Data Byte 3**
T2 [bit23:16]	**DB2[7:0] :**	**Data Byte 2**
T2 [bit15:8]	**DB1[7:0] :**	**Data Byte 1**
T2 [bit7:0]	**DB0[7:0] :**	**Data Byte 0**
T3 [bit31:24]	**DB7[7:0] :**	**Data Byte 7**
T3 [bit23:16]	**DB6[7:0] :**	**Data Byte 6**
T3 [bit15:8]	**DB5[7:0] :**	**Data Byte 5**
T3 [bit7:0]	**DB4[7:0] :**	**Data Byte 4**
…	**…**	**…**
Tn [bit31:24]	**DBm[7:0]:**	**Data Byte m**
Tn [bit23:16]	**DBm-1[7:0]:**	**Data Byte m-1**
Tn [bit15:8]	**DBm-2[7:0]:**	**Data Byte m-2**
Tn [bit7:0]	**DBm-3[7:0]:**	**Data Byte m-3**

Notes:

- *Depending on the configuration of the element size (TXESC), Tn will vary from n = 3 to 17.*
- *m is a function of n: m = (n - 1) × 4 - 1.*

6.4. Tx Event FIFO Element

Each Tx Event FIFO Element stores information about transmitted messages. By reading the Tx Event FIFO the CPU gets this information in the order the messages were transmitted. Status information about the Tx Event FIFO can be obtained from register TXEFS (Tx Event FIFO Status).



E0 [bit31] ESI: Error State Indicator

Bit	Description
0	Transmitting node is error active.
1	Transmitting node is error passive.

E0 [bit30] XTD: Extended Identifier

Bit	Description
0	11-bit standard identifier.
1	29-bit extended identifier.

E0 [bit29] RTR: Remote Transmission Request

Bit	Description
0	Data frame transmitted.
1	Remote frame transmitted.

E0 [bit28:0] ID[28:0]: Identifier

Standard or extended identifier depending on bit XTD. A standard identifier is stored into ID[28:18].

E1 [bit31:24] MM[7:0]: Message Marker

Copied from Tx Buffer into Tx Event FIFO element for identification of Tx message status.

E1 [bit23:22] ET[1:0]: Event Type

ET[1:0]	Description
00	Reserved.
01	Tx event.
10	Transmission in spite of cancellation. Always set for transmissions in DAR mode (Disable Automatic Retransmission mode).
11	Reserved.

E1 [bit21] FDF: FD Format

Bit	Description
0	Classic CAN frame format.
1	CAN FD frame format (new DLC-coding and CRC).

E1 [bit20] BRS: Bit Rate Switching

Bit	Description
0	Frame transmitted without bit rate switching.
1	Frame transmitted with bit rate switching.

E1 [bit19:16] DLC[3:0]: Data Length Code

DLC[3:0]	Description
0-8	Classic CAN + CAN FD: frame with 0-8 data bytes transmitted.
9-15	Classic CAN: frame with 8 data bytes transmitted. CAN FD: frame with 12/16/20/24/32/48/64 data bytes transmitted. See Table 3-1 for details.

E1 [bit15:0] TXTS[15:0]: Tx Timestamp

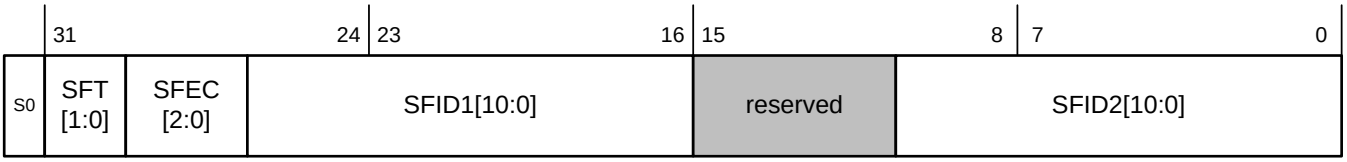
Timestamp Counter value captured on start of frame transmission. Resolution depending on configuration of the Timestamp Counter Prescaler (TSCC.TCP[3:0]).

6.5. Standard Message ID Filter Element

A Standard Message ID Filter Element consists of a single 32-bit word, and can be configured as a range filter, dual filter, classic bit mask filter, or filter for a single dedicated ID, for messages with 11-bit standard IDs.

Up to 128 filter elements can be configured for 11-bit standard IDs. When accessing a Standard Message ID Filter element, its address is

Filter List Standard Start Address (SIDFC.FLSSA[15:2]) + index of the filter element (0 to 127).



S0 [bit31:30] SFT[1:0]: Standard Filter Type

SFT[1:0]	Description
00	Range filter from SFID1[10:0] to SFID2[10:0] (SFID2[10:0] ≥ received ID ≥ SFID1[10:0]).
01	Dual ID filter for SFID1[10:0] or SFID2[10:0].
10	Classic filter: SFID1[10:0] = filter, SFID2[10:0] = mask. Only those bits of SFID1[10:0] where the corresponding SFID2[10:0] bits are "1" are relevant.
11	Filter element disabled.

Note:

- *With SFT = "11" the filter element is disabled and the acceptance filtering continues. (same behaviour as with SFEC = "000")*

S0 [bit29:27] SFEC[2:0]: Standard Filter Element Configuration

All enabled filter elements are used for acceptance filtering of standard frames. Acceptance filtering stops at the first matching enabled filter element or when the end of the filter list is reached.

If SFEC[2:0] = "100", "101", or "110" a match sets interrupt flag IR.HPM (High Priority Message) and, if enabled, an interrupt is generated. In this case register HPMS (High Priority Message Status) is updated with the status of the priority match.

SFEC[2:0]	Description
000	Disable filter element.
001	Store in Rx FIFO 0 if filter matches.
010	Store in Rx FIFO 1 if filter matches.
011	Reject ID if filter matches.
100	Set priority if filter matches.
101	Set priority and store in Rx FIFO 0 if filter matches.
110	Set priority and store in Rx FIFO 1 if filter matches.
111	Store into dedicated Rx Buffer or as debug message, configuration of SFT[1:0] ignored.

S0 [bit26:16] SFID1[10:0]: Standard Filter ID 1

This bit field has a different meaning depending on the configuration of SFEC[2:0]:

- SFEC[2:0] = "001" to "110"

  Set SFID1[10:0] according to the SFT[1:0] setting.

- SFEC[2:0] = "111"

  SFID1[10:0] defines the ID of a standard dedicated Rx Buffer or debug message to be stored. The received identifiers must match exactly, no masking mechanism is used.

S0 [bit15:11] Reserved: Reserved Bits

When writing, always write "0". The read value is undefined.

S0 [bit10:0] SFID2[10:0]: Standard Filter ID 2

This bit field has a different meaning depending on the configuration of SFEC[2:0]:

- SFEC[2:0] = "001" to "110"

  Set SFID2[10:0] according to the SFT[1:0] setting

- SFEC[2:0] = "111"

  Filter for dedicated Rx Buffers or for debug messages

  SFID2[10:9] decides whether the received message is stored into a dedicated Rx Buffer or treated as message A, B, or C of the debug message sequence.

SFID2[10:9]	Description
00	Store message into a dedicated Rx Buffer.
01	Debug Message A.
10	Debug Message B.
11	Debug Message C.

  SFID2[8:6] are reserved bits. When writing, always write "0". The read value is undefined.

  SFID2[5:0] defines the offset to the Rx Buffer Start Address RXBC.RBSA[15:2] for storage of a matching message.

Notes:

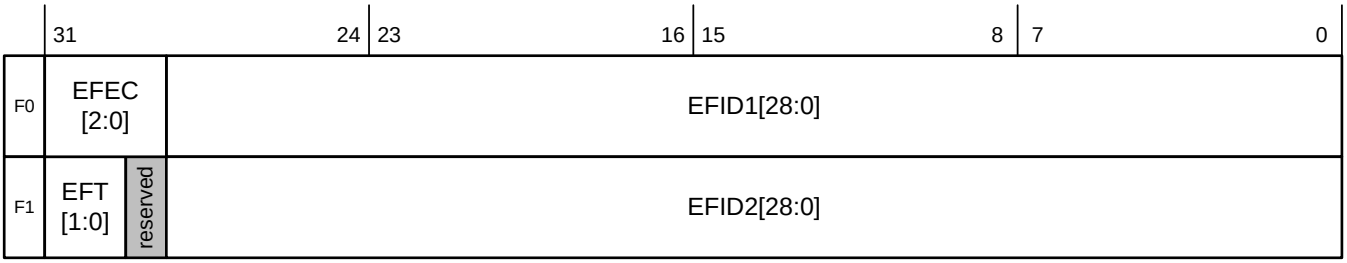
- *Debug message is used for Debug on CAN feature.*
- *Debug on CAN feature is not supported for TRAVEO™ T1G Platform.*

6.6. Extended Message ID Filter Element

An Extended Message ID Filter Element consists of two 32-bit words, and can be configured as a range filter, dual filter, classic bit mask filter, or filter for a single dedicated ID, for messages with 29-bit extended IDs.

Up to 64 filter elements can be configured for 29-bit extended IDs. When accessing an Extended Message ID Filter element, its address is

Filter List Extended Start Address (XIDFC.FLESA[15:2]) + 2 × index of the filter element (0 to 63).



F0 [bit31:29] EFEC[2:0]: Extended Filter Element Configuration

All enabled filter elements are used for acceptance filtering of extended frames. Acceptance filtering stops at the first matching enabled filter element or when the end of the filter list is reached.

If EFEC[2:0] = "100", "101", or "110" a match sets interrupt flag IR.HPM (High Priority Message) and, if enabled, an interrupt is generated. In this case register HPMS (High Priority Message Status) is updated with the status of the priority match.

EFEC[2:0]	Description
000	Disable filter element.
001	Store in Rx FIFO 0 if filter matches.
010	Store in Rx FIFO 1 if filter matches.
011	Reject ID if filter matches.
100	Set priority if filter matches.
101	Set priority and store in Rx FIFO 0 if filter matches.
110	Set priority and store in Rx FIFO 1 if filter matches.
111	Store into dedicated Rx Buffer or as debug message, configuration of EFT[1:0] ignored.

F0 [bit28:0] EFID1[28:0]: Extended Filter ID 1

This bit field has a different meaning depending on the configuration of EFEC[2:0].

- EFEC[2:0] = "001" to "110"

  Set EFID1[28:0] according to the EFT[1:0] setting.

- EFEC[2:0] = "111"

  EFID1[28:0] defines the ID of an extended dedicated Rx Buffer or debug message to be stored. The received identifiers must match exactly, only XIDAM masking mechanism (see "3.4.1.3. Extended Message ID Filtering") is used.

F1 [bit31:30] EFT[1:0]: Extended Filter Type

EFT[1:0]	Description
00	Range filter from EFID1[28:0] to EFID2[28:0] (EFID2[28:0] ≥ received ID ANDed with XIDAM ≥ EFID1[28:0]).
01	Dual ID filter Matches when EFID1[28:0] or EFID2[28:0] is equal to received ID ANDed with XIDAM.
10	Classic filter: EFID1[28:0] = filter, EFID2[28:0] = mask. Only those bits of EFID1[28:0] where the corresponding EFID2[28:0] bits are "1" are relevant. Matches when the received ID ANDed with XIDAM is equal to EFID1[28:0] masked by EFID2[28:0].
11	Range filter from EFID1[28:0] to EFID2[28:0] (EFID2[28:0] ≥ EFID1[28:0]), XIDAM mask not applied.

F1 [bit29] Reserved: Reserved Bit

When writing, always write "0". The read value is undefined.

F1 [bit28:0] EFID2[28:0]: Extended Filter ID 2

This bit field has a different meaning depending on the configuration of EFEC[2:0]:

- EFEC[2:0] = "001" to "110"

  Set EFID2[28:0] according to the EFT[1:0] setting

- EFEC[2:0] = "111"

  EFID2[28:0] is used to configure this filter for dedicated Rx Buffers or for debug messages

  EFID2[28:11] are reserved bits. When writing, always write "0". The read value is undefined.

  EFID2[10:9] decides whether the received message is stored into a dedicated Rx Buffer or treated as message A, B, or C of the debug message sequence.

EFID2[10:9]	Description
00	Store message into a dedicated Rx Buffer.
01	Debug Message A.
10	Debug Message B.
11	Debug Message C.

  EFID2[8:6] are reserved bits. When writing, always write "0". The read value is undefined.

  EFID2[5:0] defines the offset to the Rx Buffer Start Address RXBC.RBSA[15:2] for storage of a matching message.

Notes:

- *Debug message is used for Debug on CAN feature.*
- *Debug on CAN feature is not supported for TRAVEO™ T1G Platform.*

CHAPTER 32: External Time Stamp Counter For CAN FD

This chapter explains the functions and operations of the External Time Stamp Counter for CAN FD.

1. **Overview**
2. **Configuration**
3. **Operations**
4. **Example of the Operation**
5. **Register**

1. Overview

The Time Stamp Counter for CAN FD is the 16-bit counter of external time stamp function.

Time stamp Counter

- This counter is a 16-bit counter, and only counts up.
- If the counter is enable, it counts up from the current value.
- If the counter is disable, it holds current value.
- Setting enable/disable the counter is programmable by software.
- The counter counts up to the comparison value.
- When counter value comes to greater than or equal to the comparison value, the counter loads 0 by the next bus clock cycle and continues counting up.
- The limit value of the counter can be programmable by software.
- Time Stamp Counter Data Register (TSCDTR) indicates the count value.

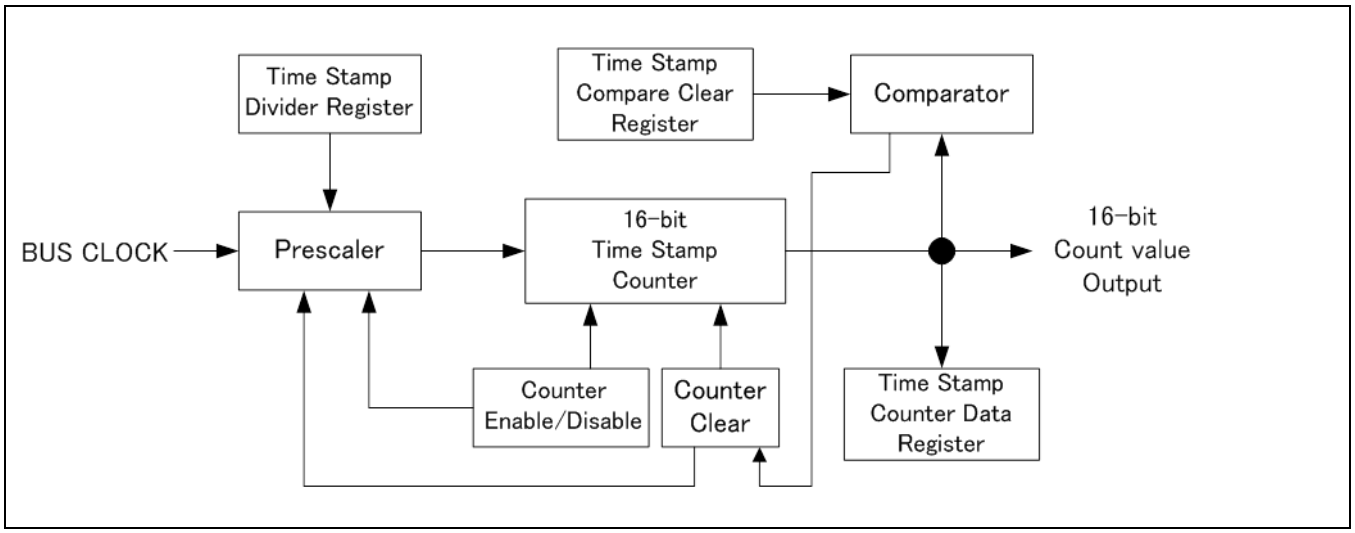
Count Up Cycle of the Time stamp Counter

- Enable to select a count up cycle (divided from bus clock ; 1 - 65536 division)
- Enable to set the division ratio any value (i.e. each step)
- Enable to set a count up cycle by software.

2. Configuration

This section explains configuration of the time stamp counter for CAN FD.

Figure 2 1 Block Diagram of Time Stamp Counter for CAN FD



3. Operations

3.1. Operation of Time Stamp Counter for CAN FD

This section explains the operation of Time Stamp Counter for CAN FD.

3.1.1. Count Up Cycle for Time Stamp Counter

Count up cycle for the Time Stamp Counter can be set for each step between 1 to 65536. Count up cycle is set by the Time Stamp Divider register (TSDIVR).

3.1.2. Time Stamp Counter

After setting the count enable (TSCNTR:CNTEN=1), the count up is started for the current counter value. After setting the counter disable (TSCNTR:CNTEN=0), the count up is stopped and the count value is held.

The counter counts up to the value that is set in the Time Stamp Compare Clear Register (TSCPCLR). When counter value comes to greater than or equal to the value that is set in the Time Stamp Compare Clear Register (TSCPCLR), the counter loads 0 by the next bus clock cycle and continues counting up.

The count value can be seen by reading Time Stamp Counter Data Register (TSCDTR).

Figure 3-1 Operation to Enable Counting (TSDIVR:CDIV15-CDIV0=0x0003)

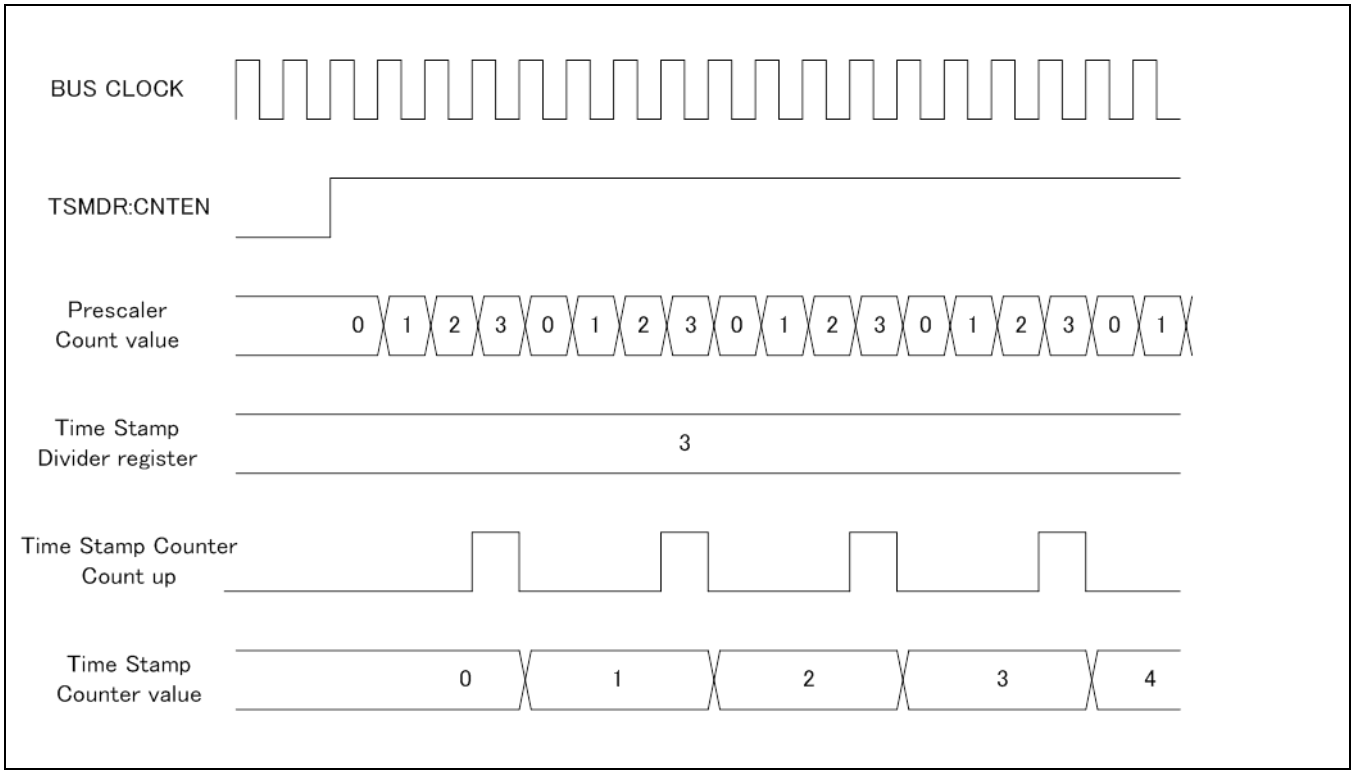


Figure 3-2 Operation to Disable Counting (TSDIVR:CDIV15-CDIV0=0x0003)

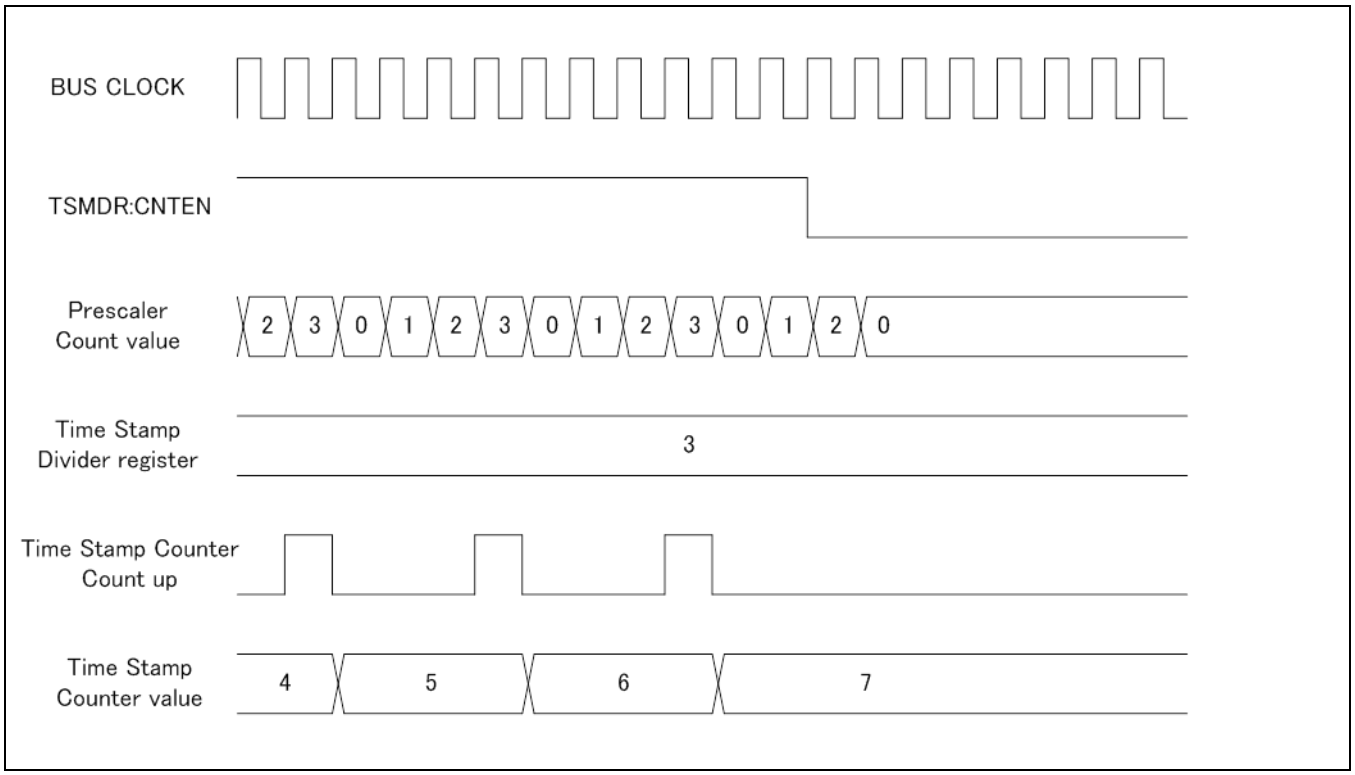
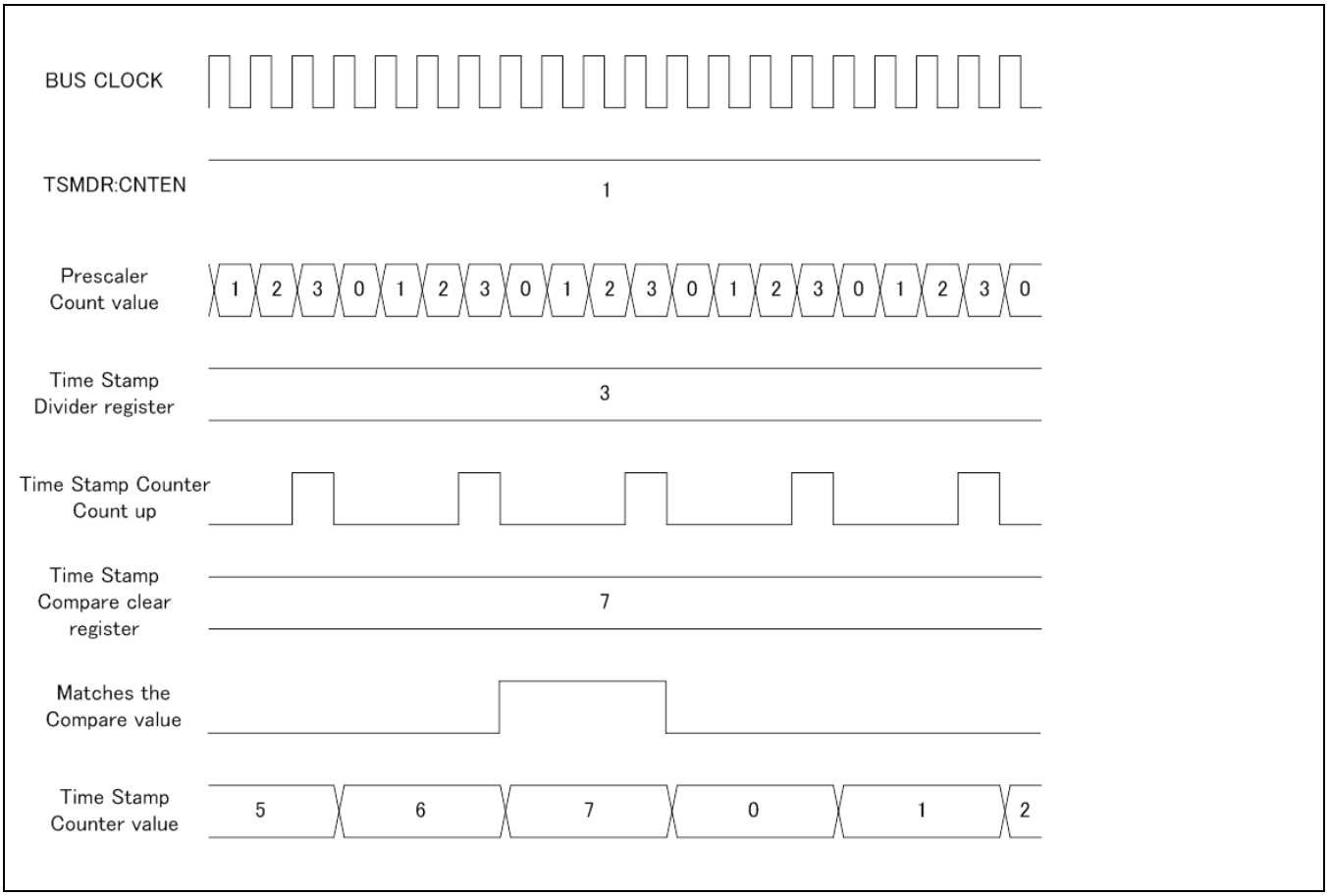


Figure 3-3 Operation to Clear the Time Stamp Counter by matching TSCPCLR (TSDIVR:CDIV15-CDIV0=0x0003, TSCPCLR:CMP15-CMP0=0x0007)



3.1.3. Clear Request

By setting the Counter Clear Request of the Time Stamp Control register (TSCNTR), the counters in the macro (Timestamp Counter, Prescaler) are cleared. Time Stamp Mode register (TSMDR), Time Stamp Divider register (TSDIVR), Time Stamp Compare Clear register (TSCPCLR) are not cleared by this request.

When the Counter Clear Request is set with the counter enabled, Time Stamp Counter and Prescaler are cleared, and the counter keeps counting up.

When the Counter Clear Request is set with the counter disabled, Time Stamp Counter and Prescaler are cleared.

Figure 3-4 Setting Counter Clear Request with the Counter Enabled (TSDIVR:CDIV15-CDIV0=0x0003, TSCPCLR:CMP15-CMP0=0x0007)

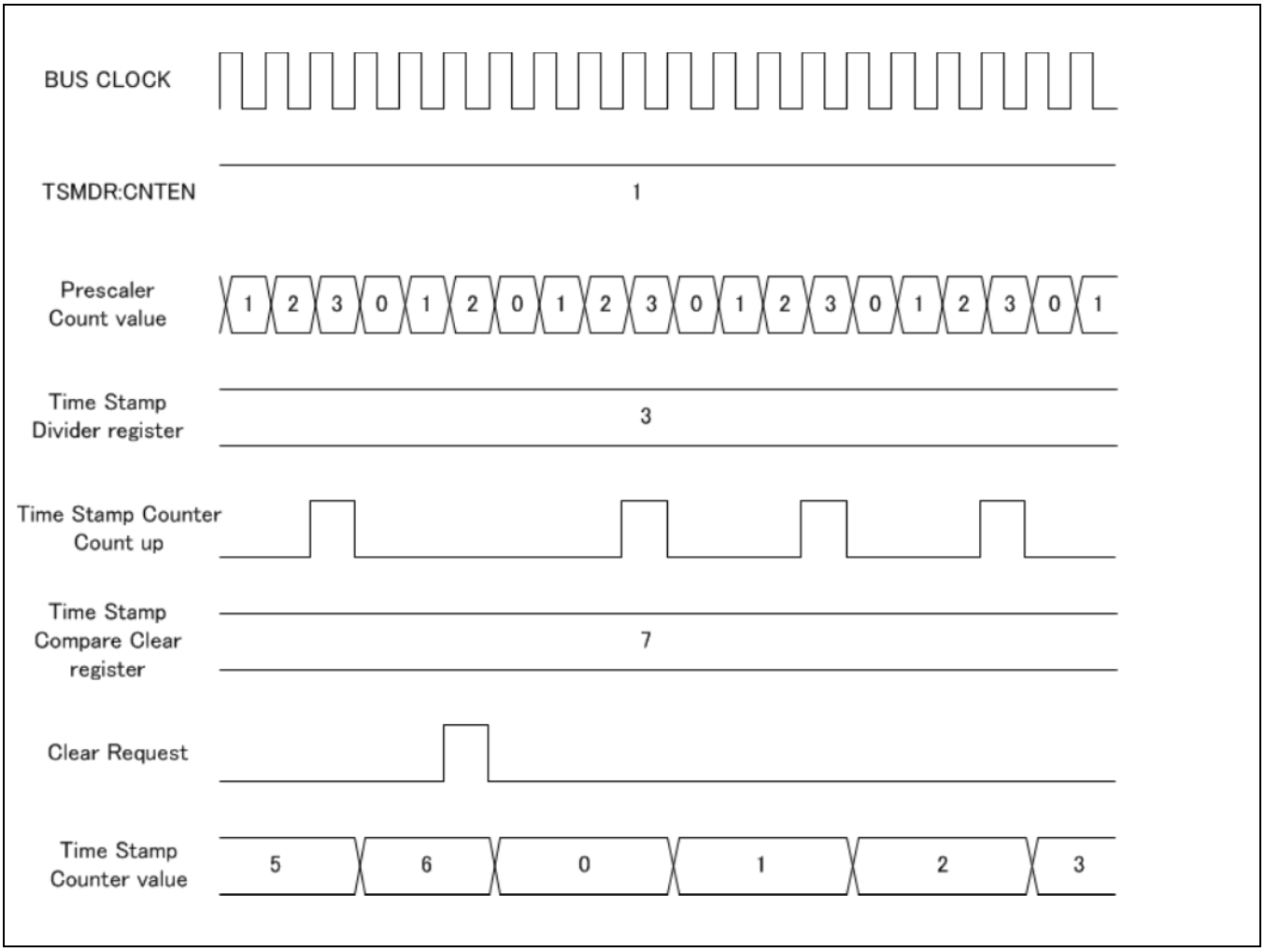
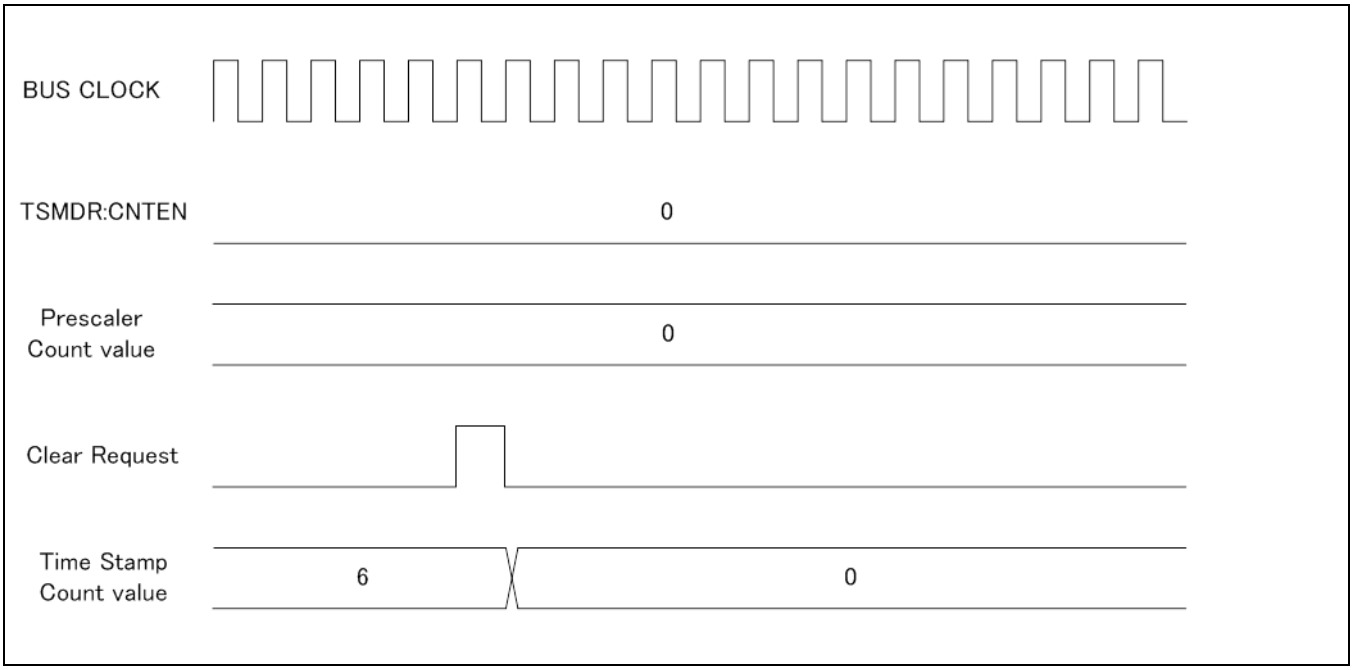


Figure 3 5 Setting Counter Clear Request with the Counter Disabled

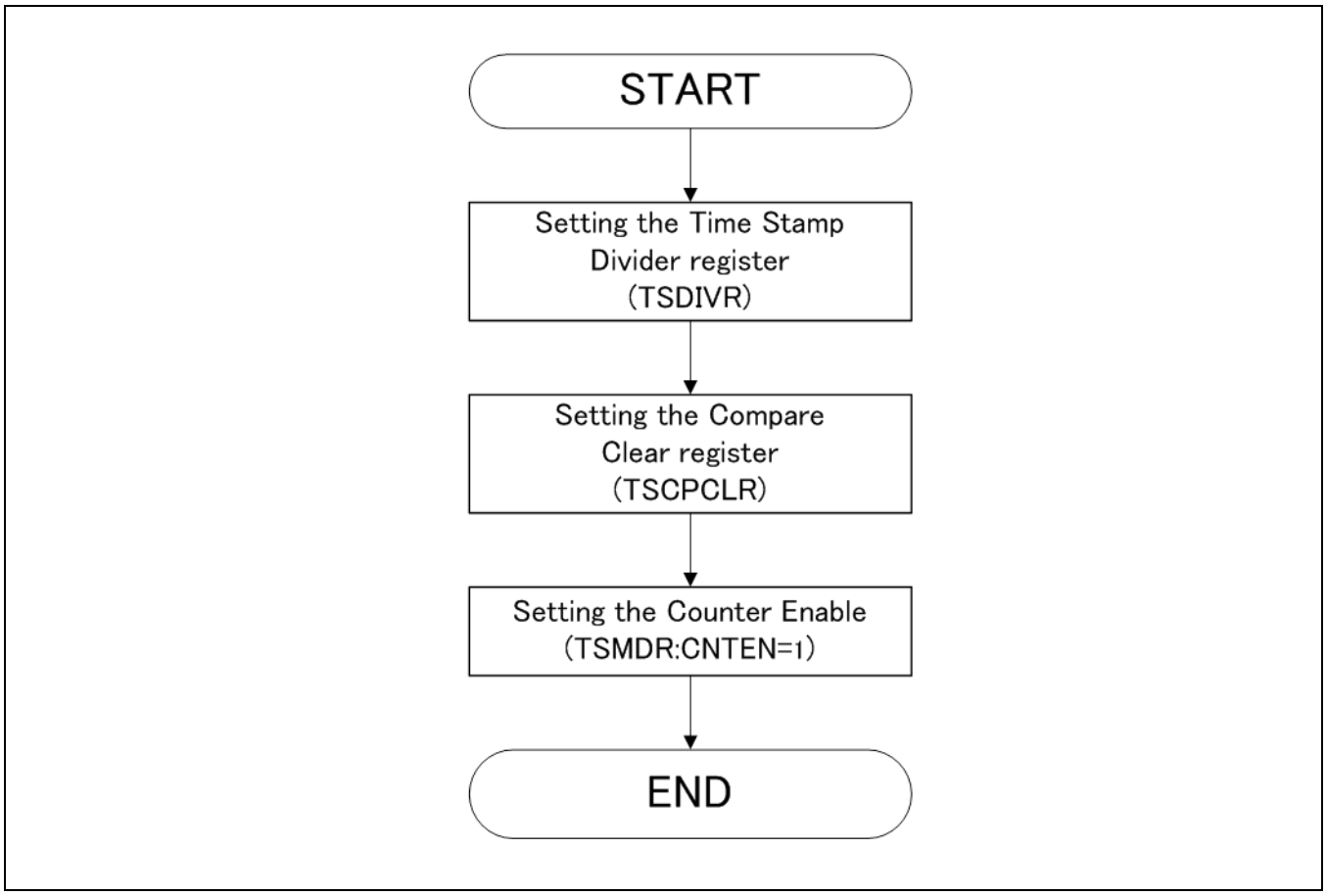


4. Example of the Operation

This section explains the example of operating Time Stamp Counter for CAN FD.

The procedure for setting Time Stamp Counter for CAN FD is shown in Figure 4-1.

Figure 4-1 The Setting Procedure for the Time Stamp Counter for CAN FD.



5. Register

This section explains the configuration and functions of the registers used for the CAN FD Time Stamp Counter.

- List of CAN FD Time Stamp Counter registers

Table 5-1 List of CAN FD Time Stamp Counter Registers

Abbreviation	Register name	See
MCG_CANFDx_TSCNTR	Time Stamp Control Register of MCU Config Group (channel 0, 1, 2)	5.1
CPG_CANFDx_TSCNTR	Time Stamp Control Register of Common Peripheral #0 Group (channel 0, 1, 2, 3, 4)	5.1
MCG_CANFDx_TSMDR	Time Stamp Mode Register of MCU Config Group (channel 0, 1, 2)	5.2
CPG_CANFDx_TSMDR	Time Stamp Mode Register of Common Peripheral #0 Group (channel 0, 1, 2, 3, 4)	5.2
MCG_CANFDx_TSDIVR	Time Stamp Divider Register of MCU Config Group (channel 0, 1, 2)	5.3
CPG_CANFDx_TSDIVR	Time Stamp Divider Register of Common Peripheral #0 Group (channel 0, 1, 2, 3, 4)	5.3
MCG_CANFDx_TSCDTR	Time Stamp Counter Data Register of MCU Config Group (channel 0, 1, 2)	5.4
CPG_CANFDx_TSCDTR	Time Stamp Counter Data Register of Common Peripheral #0 Group (channel 0, 1, 2, 3, 4)	5.4
MCG_CANFDx_TSCPCLR	Time Stamp Compare Clear Register of MCU Config Group (channel 0, 1, 2)	5.5
CPG_CANFDx_TSCPCLR	Time Stamp Compare Clear Register of Common Peripheral #0 Group (channel 0, 1, 2, 3, 4)	5.5

Table 5-2 Register Map of CAN FD Time Stamp Counter

MCU Config Group (Channel No.: 0, 1, and 2)

Offset	Register Name / Initial Value			
	+3	+2	+1	+0
0x300	MCG_CANFDx_TSMDR 00000000_00000000		MCG_CANFDx_TSCNTR 00000000_00000000	
0x304	MCG_CANFDx_TSDIVR 00000000_00000000_00000000_00000000			
0x308	MCG_CANFDx_TSCPCLR 00000000_00000000		MCG_CANFDx_TSCDTR 00000000_00000000	
0x30C-0x3FC	Reserved			

Note:

- *x is the channel number. (0 to 2)*

Common Peripheral #0 Group (Channel No.: 0, 1, 2, 3, and 4)

Offset	Register Name / Initial Value			
	+3	+2	+1	+0
0x300	CPG_CANFDx_TSMDR 00000000_00000000		CPG_CANFDx_TSCNTR 00000000_00000000	
0x304	CPG_CANFDx_TSDIVR 00000000_00000000_00000000_00000000			
0x308	CPG_CANFDx_TSCPCLR 00000000_00000000		CPG_CANFDx_TSCDTR 00000000_00000000	
0x30C-0x3FC	Reserved			

Note:

- *x is the channel number. (0 to 4)*

5.1. Time Stamp Control Register (MCG_CANFDx_TSCNTR, CPG_CANFDx_TSCNTR)

(MCG_CANFD0_TSCNTR, MCG_CANFD1_TSCNTR, MCG_CANFD2_TSCNTR, CPG_CANFD0_TSCNTR, CPG_CANFD1_TSCNTR, CPG_CANFD2_TSCNTR, CPG_CANFD3_TSCNTR, CPG_CANFD4_TSCNTR)

The Time Stamp Control Register (TSCNTR) is used to clear the counter in the macro.

bit	15	14	13	12	11	10	9	8
Field	Reserved							
R/W Attribute	R0, W0							
Protection attribute	-							
Initial value	00000000							

bit	7	6	5	4	3	2	1	0
Field	Reserved							CCLR
R/W Attribute	R0, W0							R0, W
Protection attribute	-							-
Initial value	0000000							0

[bit15:1] Reserved: Reserved bits

Always write "0" to these bits. The read value is "0".

[bit0] CCLR : Counter clear bit

The read value is "0".

bit	Description
	Write
0	No effect
1	Clear the counter(time stamp counter, prescaler).

Note:

- *Time Stamp Counter Mode Register, Time Stamp Divider Register and Time Stamp Compare Clear Register are not cleared.*

5.2. Time Stamp Mode Register(MCG_CANFDx_TSMDR, CPG_CANFDx_TSMDR)

(MCG_CANFD0_TSMDR, MCG_CANFD1_TSMDR, MCG_CANFD2_TSMDR,

CPG_CANFD0_TSMDR, CPG_CANFD1_TSMDR, CPG_CANFD2_TSMDR, CPG_CANFD3_TSMDR, CPG_CANFD4_TSMDR)

The Time Stamp Mode Register (TSMDR) is used to enable/disable the counter.

bit	15	14	13	12	11	10	9	8
Field	Reserved							
R/W Attribute	R0,W0							
Protection attribute	-							
Initial value	00000000							

bit	7	6	5	4	3	2	1	0
Field	Reserved							CNTEN
R/W Attribute	R0,W0							R/W
Protection attribute	-							-
Initial value	0000000							0

[bit15:1] Reserved: Reserved bits

Always write "0" to these bits. The read value is "0".

[bit0] CNTEN : Counter enable bit

CNTEN	Description
0	Count disabled Stop counting up and keep the counter value.
1	Count enabled Start counting up from the current value.

5.3. Time Stamp Divider Register(MCG_CANFDx_TSDIVR, CPG_CANFDx_TSDIVR)

(MCG_CANFD0_TSDIVR, MCG_CANFD1_TSDIVR, MCG_CANFD2_TSDIVR,

CPG_CANFD0_TSDIVR, CPG_CANFD1_TSDIVR, CPG_CANFD2_TSDIVR, CPG_CANFD3_TSDIVR, CPG_CANFD4_TSDIVR)

The Time Stamp Divider Register (TSDIVR) is used to set the clock division ratio for the counter.

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,W0							
Protection attribute	-							
Initial value	00000000							

bit	23	22	21	20	19	18	17	16
Field	Reserved							
R/W Attribute	R0,W0							
Protection attribute	-							
Initial value	00000000							

bit	15	14	13	12	11	10	9	8
Field	CDIV15	CDIV14	CDIV13	CDIV12	CDIV11	CDIV10	CDIV9	CDIV8
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	CDIV7	CDIV6	CDIV5	CDIV4	CDIV3	CDIV2	CDIV1	CDIV0
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

[bit31:16] Reserved: Reserved bits

Always write "0" to these bits. The read value is "0".

[bit15:0] CDIV15 to CDIV0 : Counter clock division ratio setting bit.

These bits set the counter clock division ratio.

The division ratio corresponds to the value added 1 to CDIV15-CDIV0.

Notes:

- *If counter is enabled (TSMDR:CNTEN=1), operating this bit is prohibited*
- *This register must be accessed in 16-bit mode.*

5.4. Time Stamp Counter Data Register(MCG_CANFDx_TSCDTR, CPG_CANFDx_TSCDTR)

(MCG_CANFD0_TSCDTR, MCG_CANFD1_TSCDTR, MCG_CANFD2_TSCDTR,

CPG_CANFD0_TSCDTR, CPG_CANFD1_TSCDTR, CPG_CANFD2_TSCDTR, CPG_CANFD3_TSCDTR, CPG_CANFD4_TSCDTR)

The Time Stamp Counter Data Register (TSCDTR) indicates the count value of the Time Stamp Counter.

bit	15	14	13	12	11	10	9	8
Field	CNT15	CNT14	CNT13	CNT12	CNT11	CNT10	CNT9	CNT8
R/W attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	CNT7	CNT6	CNT5	CNT4	CNT3	CNT2	CNT1	CNT0
R/W attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

[bit15:0] CNT15 to CNT0 : Counter value bits

The counter value of the Time Stamp Counter is indicated.

These bits are cleared to "0" by setting Counter Clear bit to "1" (TSCNTR:CCLR=1).

Note:

- *This register must be accessed in 16-bit mode.*

5.5. Time Stamp Compare Clear Register(MCG_CANFDx_TSCPCLR, CPG_CANFDx_TSCPCLR)

(MCG_CANFD0_TSCPCLR, MCG_CANFD1_TSCPCLR, MCG_CANFD2_TSCPCLR,

CPG_CANFD0_TSCPCLR, CPG_CANFD1_TSCPCLR, CPG_CANFD2_TSCPCLR, CPG_CANFD3_TSCPCLR, CPG_CANFD4_TSCPCLR)

The Time Stamp Compare Clear Register (TSCPCLR) is used to set the comparison value for clearing the counter.

bit	15	14	13	12	11	10	9	8
Field	CMP15	CMP14	CMP13	CMP12	CMP11	CMP10	CMP9	CMP8
R/W attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	CMP7	CMP6	CMP5	CMP4	CMP3	CMP2	CMP1	CMP0
R/W attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

[bit15:0] CMP15 to CMP0 : Compare clear setting bits

These bits are compared with the counter value, and the counter is cleared when the counter value is greater than or equal to these bits.

If these bits are "0x0000", the Time Stamp Counter is always "0".

Notes:

- *If counter is enabled (TSMDR:CNTEN=1), operating this bit is prohibited.*
- *This register must be accessed in 16-bit mode.*

CHAPTER 33: CAN FD Message RAM ECC Function

This chapter explains the functions and operations of the CAN FD Message RAM ECC Function.

1. Overview
2. Configuration
3. Interrupts
4. Explanation of Operation
5. Setup Procedure Examples
6. Registers
7. Precautions

This chapter explains only the block of the CAN FD Message RAM ECC Function.

For the functions and operations of the CAN FD Controller, see CAN FD Controller External Specification.

CANFDMRE-TXXPT03P01R02L03-E1-XX

1. Overview

In the CAN FD Message RAM, CAN message objects are stored. The ECC Function of the CAN FD Message RAM enables detection and correction of data errors in the Message RAM.

The CAN FD Message RAM ECC Function has the following features.

- 2-bit error detection and 1-bit error correction for 32-bit data in the Message RAM
- Stopping of the CAN FD Controller upon error detection
- Not support for the error generation function of the Message RAM

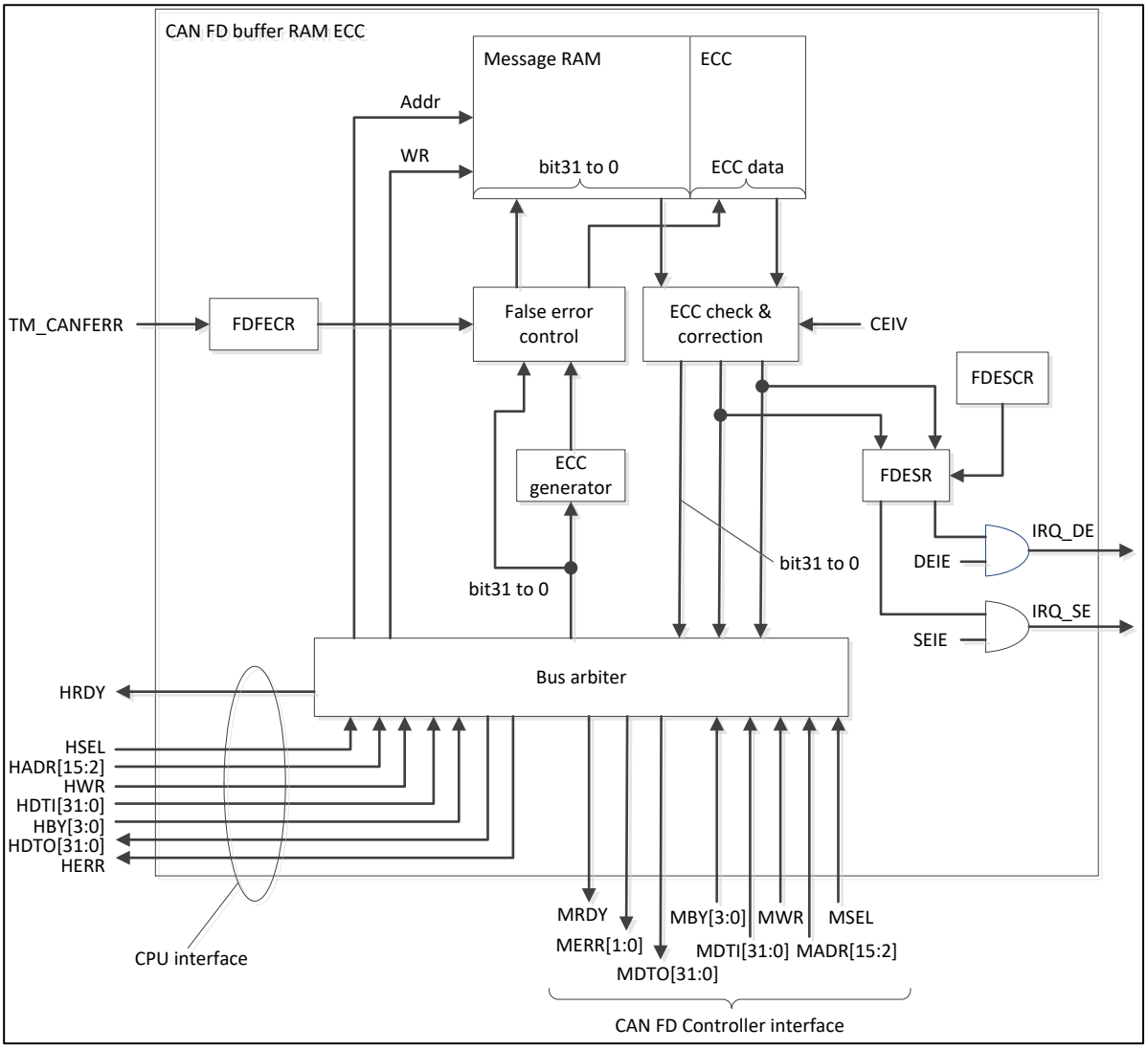
Note:

- *ECC error generation function of the message RAM is not supported for TRAVEO™ T1G Platform. Therefore CAN FD ECC False Error Control Register (FDFECR) is not writeable.*

2. Configuration

Figure 2-1 is a block diagram of the CAN FD Message RAM ECC Function.

Figure 2-1 Block Diagram of the CAN FD Message RAM ECC Function



- Message RAM
  Consists of 32-bit RAM, in which Tx/Rx messages is stored.
- ECC
  Consists of 7 bits, in which the ECC data for bits 31 to 0 of the Message RAM is stored.
- ECC generator
  Generates ECC data from data to be written in the Message RAM.
- False error control
  Controls whether to generate ECC data that causes an ECC error or to output normal ECC data.
- ECC check & correction
  Combines Message RAM data and ECC data together to check for a single- or double-bit error. For a single-bit error, this circuit corrects the data.
- Bus arbiter
  Arbitrates access by the CPU and CAN FD Controller, and reads data from or writes data to the Message RAM.
- FDESCR
  CAN FD ECC error status clear register
- FDESR
  CAN FD ECC error status register
- FDFECR
  CAN FD ECC false error control register

3. Interrupts

This section explains the interrupts of the CAN FD Message RAM ECC Function.

Interrupts of the CAN FD Message RAM ECC Function

The Message RAM ECC Function supports the following interrupts.

Table 3-1 Interrupt Controller Bits and Interrupt Factors of the CAN FD Message RAM ECC Function

Interrupt Type	Interrupt Request Flag Bit	Flag Register	Interrupt Factor	Interrupt Factor Enable Bit	Clearing of Interrupt Request Flag
Double-bit error interrupt	DEI	FDESR	2-bit data error	FDECR.DEIE	Write "1" to the double-bit error clear bit (FDESCR.DEIC).
Single-bit error interrupt	SEI	FDESR	1-bit data error	FDECR.SEIE	Write "1" to the single-bit error clear bit (FDESCR.SEIC).

4. Explanation of Operation

This section explains the operations and functions of the CAN FD Message RAM ECC.

This section explains the following functions.

- Message RAM ECC generation
- Message RAM ECC check
- ECC false error control function of the Message RAM

4.1. Message RAM ECC Generation

This section explains ECC generation of the Message RAM.

ECC Generation

The CAN FD Message RAM is made up of 32 bits and an additional 7 bits for ECC. When data is written to the CAN FD Message RAM, ECC data is generated from the write data based on the syndrome, and is written together with the write data.

4.2. Message RAM ECC Check

This section explains the ECC check of the Message RAM.

ECC Check

When data is read from the Message RAM, the ECC data is also read. In the ECC check, calculation is performed based on the syndrome to determine whether the data in the Message RAM is correct. If it is determined from the check result that the data in the Message RAM, the following operation is performed.

- In the case where only 1 bit in the data is incorrect (single-bit error):
  The single-bit error occurrence bit (FDESR.SEI) is set and the RAM address is set in the CAN FD ECC single-bit error address register (FDSEAR). The incorrect 1-bit data is corrected and then is read as normal data.
- In the case where 2 bits in the data are incorrect (double-bit error):
  The double-bit error occurrence bit (FDESR.DEI) is set and the RAM address is set in the CAN FD ECC double-bit error address register (FDDEAR). The incorrect data is not corrected and then is read as it is.

If the interrupt enable bit corresponding to each error bit (FDESR.SEI or FDESR.DEI) has been set to "1", an interrupt occurs when the error bit is set to "1".

Stopping of the CAN FD Controller upon Error Detection

If the CAN FD Controller detects a double-bit error when it reads the Message RAM, it sets the INIT bit in its CC control register (CCCR). Then, the CAN FD Controller stops.

Error Response to CPU upon Error Detection

When error response is enabled (FDECR.CEREN = "1"), if the CPU reads the Message RAM and a double-bit error is detected at this time, an error response is returned to the CPU.

4.3. ECC False Error Control Function of the Message RAM

This section explains the ECC false error control function. This function is used for testing program operation for when an ECC error occurs.

ECC False Error Control Function

The ECC false error control function allows you to write data that causes an ECC error in the Message RAM. By reading such data from the Message RAM, you can generate a single- or double-bit error. If the error interrupt has been enabled, the interrupt occurs. If ECC error processing has been programmed in the interrupt processing, you can verify whether the ECC error processing operates correctly.

Setting Method

Setting "H" to the TM_CANFERR signal allows you to set the CAN FD ECC false error control register, enabling the use of the ECC false error control function. The following is the setting method.

1. Set "H" to the TM_CANFERR.
   This allows you to set registers related to CAN FD ECC false error control.
2. Configure the CAN FD ECC false error control register (FDFECR).
   Set "1" to FERR, and set each bit of EY2 to EY0 and EI15 to EI0 for the bits which you want to make an error.
3. Write data to the Message RAM.
   The bits are inverted according to the CAN FD ECC false error control register (FDFECR), and then they are written to the Message RAM.

After this setting is made, reading the Message RAM generates an ECC error.

Notes:

- *Even if the TM_CANFERR signal is set to "L" after error data is written to the Message RAM by using the ECC false error control function, the error data remains in the Message RAM. Therefore, reading this error data generates an ECC error.*
- *When the TM_CANFERR signal is "L", the CAN FD ECC false error control register (FDFECR) cannot be written. When the TM_CANFERR signal becomes "L", the bits that have been set to "1" become "0".*

5. Setup Procedure Examples

This section provides examples of setup procedures for the CAN FD Message RAM ECC Function.

Setup Procedures for the CAN FD Message RAM ECC Function

This section explains a setting procedure example for the CAN FD Message RAM ECC Function. For details on the setting of the CAN FD Controller, see the chapter on the CAN FD Controller.

1) Set "1" in the INIT bit and CCE bit in the CC control register (CCCR) of the CAN FD Controller.

   This enables write operation to the control bit of registers.

2) Set the CAN FD Controller registers.

   Make the settings of the CAN FD Controller according to the use conditions.

3) Initialize the Message RAM.

   Initializing the RAM, which is in an undefined state, prevents an ECC error from being detected by mistake.

4) Set "1" to the SEIC bit and DEIC bit in the CAN FD ECC error status clear register (FDESCR).

   This clears the SEI bit and DEI bit in the CAN FD ECC error status register (FDESR).

5) Set each bit in the CAN FD ECC error control register.

   Make the settings of the CAN FD ECC Function according to the use conditions.

Performing the above setting procedure enables the use of the CAN FD Message RAM ECC Function. Since the INIT bit in the CC control register (CCCR) is "1", the CAN FD Controller is in the operation stop state. To enable operation of the CAN FD Controller, set "0" in the INIT bit in the CC control register (CCCR) after performing the above procedure.

6. Registers

This section explains the registers of the CAN FD Message RAM ECC Function.

Table 6-1 List of the Registers of the CAN FD Message RAM ECC Function

Abbreviated Register Name	Register Name	Reference
MCG_CANFDx_FDECR	CAN FD ECC error control register of MCU Config Group (channel 0,1,2)	6.1
CPG_CANFDx_FDECR	CAN FD ECC error control register of Common Peripheral #0 Group (channel 0,1,2,3,4)	6.1
MCG_CANFDx_FDESR	CAN FD ECC error status register of MCU Config Group (channel 0,1,2)	6.2
CPG_CANFDx_FDESR	CAN FD ECC error status register of Common Peripheral #0 Group (channel 0,1,2,3,4)	6.2
MCG_CANFDx_FDESCR	CAN FD ECC error status clear register of MCU Config Group (channel 0,1,2)	6.3
CPG_CANFDx_FDESCR	CAN FD ECC error status clear register of Common Peripheral #0 Group (channel 0,1,2,3,4)	6.3
MCG_CANFDx_FDDEAR	CAN FD ECC double-bit error address register of MCU Config Group (channel 0,1,2)	6.4
CPG_CANFDx_FDDEAR	CAN FD ECC double-bit error address register of Common Peripheral #0 Group (channel 0,1,2,3,4)	6.4
MCG_CANFDx_FDSEAR	CAN FD ECC single-bit error address register of MCU Config Group (channel 0,1,2)	6.5
CPG_CANFDx_FDSEAR	CAN FD ECC single-bit error address register of Common Peripheral #0 Group (channel 0,1,2,3,4)	6.5
MCG_CANFDx_FDFECR	CAN FD ECC false error control register of MCU Config Group (channel 0,1,2)	6.6
CPG_CANFDx_FDFECR	CAN FD ECC false error control register of Common Peripheral #0 Group (channel 0,1,2,3,4)	6.6

Register Map

Table 6-2 Register Map of CAN FD MESSAGE RAM ECC FUNCTION

MCU Config Group (Channel No.: 0, 1, and 2)

Offset	Register Name / Initial Value			
	+3	+2	+1	+0
0x200	MCG_CANFDx_FDSEAR 00000000_00000000		MCG_CANFDx_FDESR 00000000	MCG_CANFDx_FDECR 00000000
0x204	MCG_CANFDx_FDDEAR 00000000_00000000		MCG_CANFDx_FDESCR 00000000	Reserved
0x208	MCG_CANFDx_FDFECR 00000000_00000000_00000000_00000000			
0x20A	Reserved	Reserved	Reserved	
0x20C-0x2FC	Reserved			

Notes:

- *x is the channel number. (0 to 2)*
- *It depends on the product spec whether to use MCG_CANFDx _*** register group.*
- *Please refer to Hardware Manual or Data Sheet for more details.*

Common Peripheral #0 Group (Channel No.: 0, 1, 2, 3, and 4)

Offset	Register Name / Initial Value			
	+3	+2	+1	+0
0x200	CPG_CANFDx_FDSEAR 00000000_00000000		CPG_CANFDx_FDESR 00000000	CPG_CANFDx_FDECR 00000000
0x204	CPG_CANFDx_FDDEAR 00000000_00000000		CPG_CANFDx_FDESCR 00000000	Reserved
0x208	CPG_CANFDx_FDFECR 00000000_00000000_00000000_00000000			
0x20A	Reserved	Reserved	Reserved	
0x20C-0x2FC	Reserved			

Notes:

- *x is the channel number. (0 to 4)*
- *It depends on the product spec whether to use CPG_CANFDx _*** register group.*
- *Please refer to Hardware Manual or Data Sheet for more details.*

6.1. CAN FD ECC Error Control Register (MCG_CANFDx_FDECR, CPG_CANFDx_FDECR)

(MCG_CANFD0_FDECR, MCG_CANFD1_FDECR, MCG_CANFD2_FDECR,

CPG_CANFD0_FDECR, CPG_CANFD1_FDECR, CPG_CANFD2_FDECR, CPG_CANFD3_FDECR, CPG_CANFD4_FDECR)

The CAN FD ECC error control register (FDECR) is used to set whether to enable the interrupt when single-bit error correction or double-bit error detection occurs during the ECC check. It is also used to set ECC error detection stop and response to the CPU.

bit	7	6	5	4	3	2	1	0
Field	Reserved				CEIV	CEREN	DEIE	SEIE
R/W attribute	R0,W0				R/W	R/W	R/W	R/W
Protection attribute	-				-	-	-	-
Initial value	0000				0	0	0	0

[bit7:4] Reserved: Reserved Bits

When writing, always write "0". When reading, "0" is always read.

[bit3] CEIV: ECC Check Disable Bit

Bit	Description
0	Enable ECC check. [Initial value]
1	Disable ECC check.

- This bit can be changed when the CCE bit in the CC control register (CCCR) of the CAN FD Controller is set "1".
- When CEIV bit is "0", the ECC check is enabled.
- When CEIV bit is "1", ECC check is disabled.

[bit2] CEREN: ECC Error Response Enable Bit

Bit	Description
0	Disable error response for double-bit error detection. [Initial value]
1	Enable error response for double-bit error detection.

- This bit can be changed when the CCE bit in the CC control register (CCCR) of the CAN FD Controller is set "1".
- When CEREN bit is "0", even if a double-bit error is detected in the read operation of the Message RAM by the CPU, an error response is not made for this setting.
- When CEREN bit is "1", if a double-bit error is detected in the read operation of the Message RAM by the CPU, an error response is made for this setting.

Note:

- *The CEREN bit does not apply to the single-bit error.*

[bit1] DEIE: Double-Bit Error Factor Interrupt Enable Bit

Bit	Description
0	Disable the interrupt caused by the double-bit error (FDESR.DEI). [Initial value]
1	Enable the interrupt caused by the double-bit error (FDESR.DEI).

- When DEIE bit is "0", even if a double-bit error is detected (FDESR.DEI = "1"), the IRQ_DE signal remains "L".
- When DEIE bit is "1", if a double-bit error is detected (FDESR.DEI = "1"), the IRQ_DE signal becomes "H" to request an interrupt.

[bit0] SEIE: Single-Bit Error Factor Interrupt Enable Bit

Bit	Description
0	Disable the interrupt caused by the single-bit error (FDESR.SEI). [Initial value]
1	Enable the interrupt caused by the single-bit error (FDESR.SEI).

- When SEIE bit is "0", even if a single-bit error is detected (FDESR.SEI = "1"), the IRQ_SE signal remains "L".
- When SEIE bit is "1", if a single-bit error is detected (FDESR.SEI = "1"), the IRQ_SE signal becomes "H" to request an interrupt.

6.2. CAN FD ECC Error Status Register (MCG_CANFDx_FDESR, CPG_CANFDx_FDESR)

(MCG_CANFD0_FDESR, MCG_CANFD1_FDESR, MCG_CANFD2_FDESR,

CPG_CANFD0_FDESR, CPG_CANFD1_FDESR, CPG_CANFD2_FDESR, CPG_CANFD3_FDESR, CPG_CANFD4_FDESR)

The CAN FD ECC error status register (FDESR) displays whether a single-bit error has been corrected in the ECC check and whether a double-bit error has been detected. When any bit in this register becomes "1", it remains "1" unless it is cleared by using CAN FD ECC error status clear register (FDESCR).

bit	7	6	5	4	3	2	1	0
Field	Reserved						DEI	SEI
R/W attribute	R0,W0						R,WX	R,WX
Protection attribute	-						-	-
Initial value	000000						0	0

[bit7:2] Reserved: Reserved Bits

When writing, always write "0". When reading, "0" is always read.

[bit1] DEI: Double-Bit Error Occurrence Bit

Bit	Description
0	Indicate that a double-bit error has not been detected. [Initial value]
1	Indicate that a double-bit error has been detected.

- Any value written in the DEI bit is ignored.
- When the DEI bit is "1", to clear this bit to "0", write "1" to the DEIC bit in the CAN FD ECC error status clear register (FDESCR).

Notes:

- *When the DEI bit is "1", the CAN FD ECC double-bit error address register (FDDEAR) behaves as follows: If a double-bit error is detected in the Message RAM read operation at an address other than the one retained in this register, the register is not updated, and retains the previous address value.*
- *Even if the DEI bit is "1", a single-bit error can be detected.*
- *If a double-bit error is detected when the CAN FD Controller or CPU reads data from the Message RAM, this bit is set "1".*

[bit0] SEI: Single-Bit Error Occurrence Bit

Bit	Description
0	Indicate that a single-bit error has not been detected. [Initial value]
1	Indicate that a single-bit error has been detected.

- Any value written in the SEI bit is ignored.

- When the SEI bit is "1", to clear this bit to "0", write "1" to the SEIC bit in the CAN FD ECC error status clear register (FDESCR).

Notes:

- *When the SEI bit is "1", the CAN FD ECC single-bit error address register (FDSEAR) behaves as follows: if a single-bit error is detected in the Message RAM read operation at an address other than the one retained in this register, the register is not updated, and retains the previous address value.*
- *Even if the SEI bit is "1", a double-bit error can be detected.*
- *If a single-bit error is detected when the CAN FD Controller or CPU reads data from the Message RAM, this bit is set "1".*

6.3. CAN FD ECC Error Status Clear Register (MCG_CANFDx_FDESCR, CPG_CANFDx_FDESCR)

(MCG_CANFD0_FDESCR, MCG_CANFD1_FDESCR, MCG_CANFD2_FDESCR,

CPG_CANFD0_FDESCR, CPG_CANFD1_FDESCR, CPG_CANFD2_FDESCR, CPG_CANFD3_FDESCR, CPG_CANFD4_FDESCR)

The CAN FD ECC error status clear register (FDESCR) is used to clear bits in the CAN FD ECC error status register.

bit	7	6	5	4	3	2	1	0
Field	Reserved						DEIC	SEIC
R/W attribute	R0,W0						R0,W	R0,W
Protection attribute	-						-	-
Initial value	000000						0	0

[bit7:2] Reserved: Reserved Bits

When writing, always write "0". When reading, "0" is always read.

[bit1] DEIC: Double-Bit Error Clear Bit

Bit	Description
	During Write Operation
0	No effect
1	Change the double-bit error occurrence bit (FDESR.DEI) to "0".

■ "0" is always read.

[bit0] SEIC: Single-Bit Error Clear Bit

Bit	Description
	During Write Operation
0	No effect
1	Change the single-bit error occurrence bit (FDESR.SEI) to "0".

■ "0" is always read.

6.4. CAN FD ECC Double-Bit Error Address Register (MCG_CANFDx_FDDEAR, CPG_CANFDx_FDDEAR)

(MCG_CANFD0_FDDEAR, MCG_CANFD1_FDDEAR, MCG_CANFD2_FDDEAR,

CPG_CANFD0_FDDEAR, CPG_CANFD1_FDDEAR, CPG_CANFD2_FDDEAR,
CPG_CANFD3_FDDEAR, CPG_CANFD4_FDDEAR)

The CAN FD ECC double-bit error address register (FDDEAR) indicates the address which a double-bit error has occurred at the Message RAM during the ECC check. This register is valid when the DEI bit in the CAN FD ECC error status register (FDESR) is "1". While the DEI bit in the CAN FD ECC error status register (FDESR) is "1", the value of this register is retained.

bit	15	14	13	12	11	10	9	8
Field	DRA15	DRA14	DRA13	DRA12	DRA11	DRA10	DRA9	DRA8
R/W attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	DRA7	DRA6	DRA5	DRA4	DRA3	DRA2	DRA1*	DRA0*
R/W attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R0,WX	R0,WX
Protection attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

*: DRA1 and DRA0 are fixed to "0".

[bit15:0] DRA15 to DRA0: Double-Bit Error Message RAM Address Bits

DRA15 to DRA0	Description
0x0000	Indicates that a double-bit error has occurred at Message RAM address 0.
0x0004	Indicates that a double-bit error has occurred at Message RAM address 4.
0x0008	Indicates that a double-bit error has occurred at Message RAM address 8.
...	.....

- Writing access to DRA15 to DRA0 bits is ignored.
- DRA15 to DRA0 bits are valid when the DEI bit in the CAN FD ECC error status register (FDESR) is "1".
- When the DEI bit in the CAN FD ECC error status register (FDESR) changes from "0" to "1", the Message RAM address is set to the CAN FD ECC double-bit error address register (FDDEAR). Then, this value is retained as long as the DEI bit in the CAN FD ECC error status register (FDESR) is "1". Therefore, even if a double-bit error is detected more than once, this register retains the address value for the first detected error until the DEI bit in the CAN FD ECC error status register (FDESR) is cleared to "0".

6.5. CAN FD ECC Single-Bit Error Address Register (MCG_CANFDx_FDSEAR, CPG_CANFDx_FDSEAR)

(MCG_CANFD0_FDSEAR, MCG_CANFD1_FDSEAR, MCG_CANFD2_FDSEAR,

CPG_CANFD0_FDSEAR, CPG_CANFD1_FDSEAR, CPG_CANFD2_FDSEAR, CPG_CANFD3_FDSEAR, CPG_CANFD4_FDSEAR)

The CAN FD ECC single-bit error address register (FDSEAR) indicates the address which a single-bit error has occurred at the Message RAM during the ECC check. This register is valid when the SEI bit in the CAN FD ECC error status register (FDESR) is "1". While the SEI bit in the CAN FD ECC error status register (FDESR) is "1", the value of this register is retained.

bit	15	14	13	12	11	10	9	8
Field	SRA15	SRA14	SRA13	SRA12	SRA11	SRA10	SRA9	SRA8
R/W attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	SRA7	SRA6	SRA5	SRA4	SRA3	SRA2	SRA1*	SRA0*
R/W attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R0,WX	R0,WX
Protection attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

*: SRA1 and SRA0 are fixed to "0".

[bit15:0] SRA15 to SRA0: Single-Bit Error Message RAM Address Bits

SRA15 to SRA0	Description
0x0000	Indicates that a single-bit error has occurred at Message RAM address 0.
0x0004	Indicates that a single-bit error has occurred at Message RAM address 4.
0x0008	Indicates that a single-bit error has occurred at Message RAM address 8.
...	.....

- Writing access to SRA15 to SRA0 bits is ignored.
- SRA15 to SRA0 bits are valid when the SEI bit in the CAN FD ECC error status register (FDESR) is "1".
- When the SEI bit in the CAN FD ECC error status register (FDESR) changes from "0" to "1", the Message RAM address is set to the CAN FD ECC single-bit error address register (FDSEAR). Then, this value is retained as long as the SEI bit in the CAN FD ECC error status register (FDESR) is "1". Therefore, even if a single-bit error is detected more than once, this register retains the address value for the first detected error until the SEI bit in the CAN FD ECC error status register (FDESR) is cleared to "0".

6.6. CAN FD ECC False Error Control Register (MCG_CANFDx_FDFECR, CPG_CANFDx_FDFECR)

(MCG_CANFD0_FDFECR, MCG_CANFD1_FDFECR, MCG_CANFD2_FDFECR,

CPG_CANFD0_FDFECR, CPG_CANFD1_FDFECR, CPG_CANFD2_FDFECR,
CPG_CANFD3_FDFECR, CPG_CANFD4_FDFECR)

The CAN FD ECC false error control register (FDFECR) is used to specify the bytes and bits in which an error is to be generated. This register can be set when the TM_CANFERR signal is "H".

bit	31	30	29	28	27	26	25	24
Field	FERR	Reserved						
R/W attribute	R/W	R0,W0						
Protection attribute	-	-						
Initial value	0	0000000						

bit	23	22	21	20	19	18	17	16
Field	Reserved					EY2	EY1	EY0
R/W attribute	R0,W0					R/W	R/W	R/W
Protection attribute	-					-	-	-
Initial value	00000					0	0	0

bit	15	14	13	12	11	10	9	8
Field	EI15	EI4	EI3	EI2	EI1	EI10	EI9	EI8
R/W attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	EI7	EI6	EI5	EI4	EI3	EI2	EI1	EI0
R/W attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

[bit31] FERR: Error insertion Enable Bit

Bit	Description
0	Disable writing of error data. [Initial value]
1	Enable writing of error data.

- When the TM_CANFERR signal is "L", the FERR bit is fixed to "0" and writing to this bit is ignored.
- When the TM_CANFERR signal is "H", the FERR bit is writable.

- When the FERR bit is "1", error data is written to the Message RAM according to EY2 to EY0 bits and EI15 to EI0 bits.

[bit30:19] Reserved: Reserved Bits

When writing, always write "0". When reading, "0" is always read.

[bit18:16] EY2 to EY0: Error Byte Specification Bits

bit	Description
EY0	Sets bit15 to 0 in the Message RAM as error targets.
EY1	Sets bit31 to 16 in the Message RAM as error targets.
EY2	Sets bit38 to 32 in the Message RAM (bits for ECC) error targets.

- When "1" is set, the specified 2-byte range of the Message RAM is the error target.
  Example 1) In the case of EY0 = "1", and EY1 and EY2 = "0", bit15 to 0 in the Message RAM are the error targets.

  Example 2) In the case of EY0 = EY2 = "1", and EY1 = "0", bit15 to 0 and bit38 to 32 in the Message RAM are the error targets.
- When EY2 to EY0 bits are all "0", error data is not written.
- EY2 sets the 7-bit range in the Message RAM as the error target. Use EI6 to EI0 to specify bits.
- When the TM_CANFERR signal is "L", the EY2 to EY0 bits are all fixed to "0" and writing to these bits is ignored.
- When the TM_CANFERR signal is "H", the EY2 to EY0 bits are writable.

[bit15:0] EI15 to EI0: Error Bit Specification Bits

bit	Description
EI0	In the 2 bytes specified by EY2 to EY0, bit0 is an error target.
EI1	In the 2 bytes specified by EY2 to EY0, bit1 is an error target.
EI2	In the 2 bytes specified by EY2 to EY0, bit2 is an error target.
EI3	In the 2 bytes specified by EY2 to EY0, bit3 is an error target.
EI4	In the 2 bytes specified by EY2 to EY0, bit4 is an error target.
EI5	In the 2 bytes specified by EY2 to EY0, bit5 is an error target.
EI6	In the 2 bytes specified by EY2 to EY0, bit6 is an error target.
EI7	In the 2 bytes specified by EY1 to EY0, bit7 is an error target.
EI8	In the 2 bytes specified by EY1 to EY0, bit8 is an error target.
EI9	In the 2 bytes specified by EY1 to EY0, bit9 is an error target.
EI10	In the 2 bytes specified by EY1 to EY0, bit10 is an error target.
EI11	In the 2 bytes specified by EY1 to EY0, bit11 is an error target.
EI12	In the 2 bytes specified by EY1 to EY0, bit12 is an error target.
EI13	In the 2 bytes specified by EY1 to EY0, bit13 is an error target.
EI14	In the 2 bytes specified by EY1 to EY0, bit14 is an error target.
EI15	In the 2 bytes specified by EY1 to EY0, bit15 is an error target.

- When data is written to the Message RAM, if "1" has been set, the specified bits in the specified 2 bytes are inverted before they are written.

Example 1) In the case of EY1 = "1", EY0 = EY2 = "0", EI3 = "1", and EI0 to EI2 = EI4 to EI15 = "0", the value of bit19 in the Message RAM is inverted before it is written. The other bits are written as they are.

Example 2) In the case of EY0 = EY1 = "1", EY2 = "0", EI3 = "1", and EI0 to EI2 = EI4 to EI15 = "0", the values of bit3 and bit19 in the Message RAM is inverted before it is written. The other bits are written as they are.

- When EI15 to EI0 bits are all "0", error data is not written.
- EY2 sets the 7-bit range in the Message RAM as the error target. Use EI6 to EI0 to specify bits.
- When the TM_CANFERR signal is "L", the EI15 to EI0 bits are all fixed to "0" and writing to these bits is ignored.
- When the TM_CANFERR signal is "H", the EI15 to EI0 bits are writable.

7. Precautions

This section explains precautions on the use of the CAN FD Message RAM ECC.

Access to the Message RAM

- To access the Message RAM, perform 32-bit access. Using another number of bits as the unit of access may cause an ECC error or the writing of wrong data to the Message RAM because the data in the Message RAM is tested as 32-bit long data.

ECC Check

- The data in the Message RAM is undefined after power-on. Therefore, a double- or single-bit error may occur if the Message RAM is read before data is written to the Message RAM area to be used. After power-on, write data to the Message RAM area to be used before reading the Message RAM.
- The following bits can be changed when the configuration change enable bit in the CC control register of the CAN FD Controller (CCCR.CCE bit) is "1".
  - ECC error response enable bit (CEREN) in the CAN FD ECC error control register (FDECR)
  - ECC check disable bit (CEIV) in the CAN FD ECC error control register (FDECR)
- If an ECC error is detected in read access to the Message RAM from the CAN FD Controller, the following bits are set: the error flags regarding the ECC of the CAN FD Controller (IR.BEU and IR.BEC) and the error flags regarding the ECC of the Message RAM (FDESR.DEI and FDESR.SEI).
- If an ECC error is detected in read access to the Message RAM from the CPU, the error flags regarding the ECC of the Message RAM (FDESR.DEI and FDESR.SEI) are set.

ECC False Error Control Function

- Even if the TM_CANFERR signal is set to "L" after error data is written to the Message RAM by using the ECC false error control function, the error data remains in the Message RAM. Therefore, reading this error data detects an ECC error.
- When the TM_CANFERR signal is "L", the CAN FD ECC false error control register (FDFECR) cannot be written. When the TM_CANFERR signal becomes "L", the bits that have been set to "1" become "0".
- With the ECC false error control function, error data can be written to the Message RAM by the CAN FD Controller or CPU.

CHAPTER 34: CAN Prescaler

This chapter describes the CAN prescaler.

1. **Overview**
2. **Configuration**
3. **Explanation of Operation**
4. **Setting Procedure**
5. **Registers**
6. **Restriction**

CANP-TXXPT03P01R01L05-E1-XX

1. Overview

This section provides an overview of CAN prescaler.

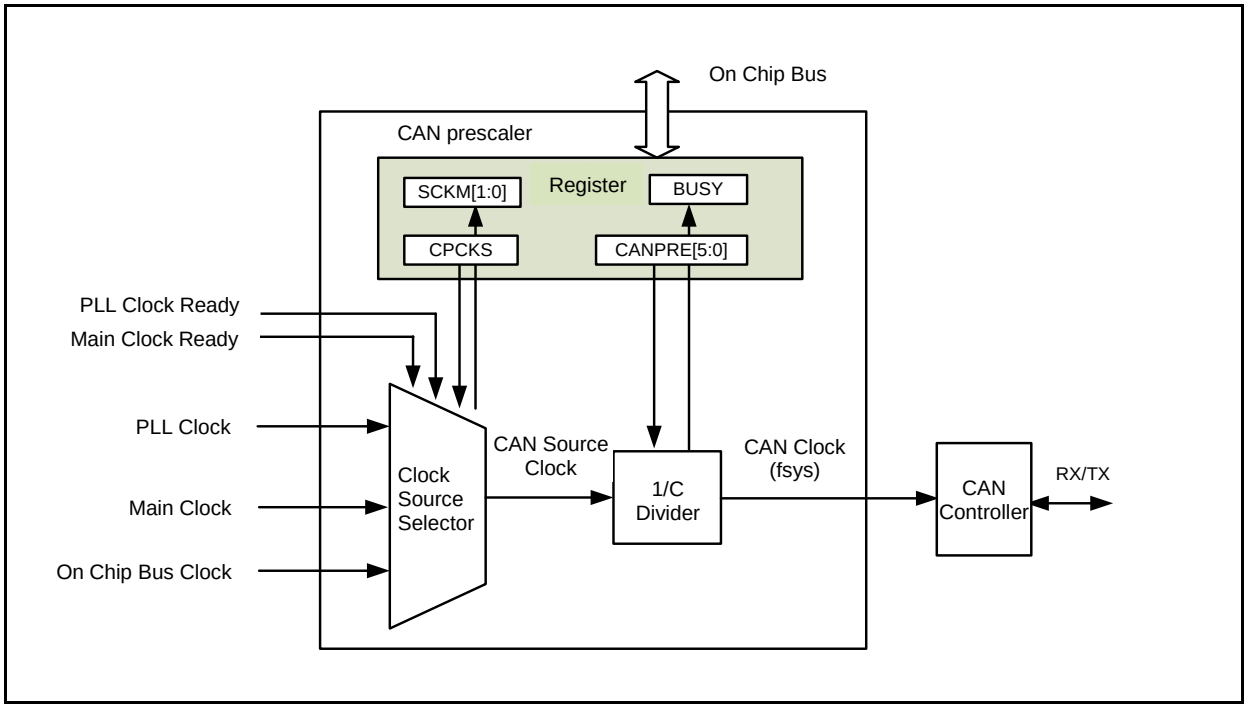
This module generates clocks (fsys) to be supplied from each clock source to the CAN macro.

2. Configuration

This section provides a block diagram of CAN prescaler.

The following block diagram shows the CAN controller and the CAN prescaler (register, clock source selector circuit, and division circuit).

Figure 2-1 Block Diagram of CAN Prescaler



3. Explanation of Operation

This section describes the operation of CAN prescaler.

Source Clock Selection

PLL(product specification) clock and main clock are selectable for a source clock of the CAN prescaler.

If the PLL(product specification) clock is selected but PLL(product specification) clock ready is not valid (oscillation is stopped or oscillation stabilization wait operation is in progress), a bus clock is selected. Similarly, if the main clock is selected but main clock ready is not valid (oscillation is stopped or oscillation stabilization wait operation is in progress), a bus clock is selected.

The currently selected clock source can be read from the status register.

Division Circuit

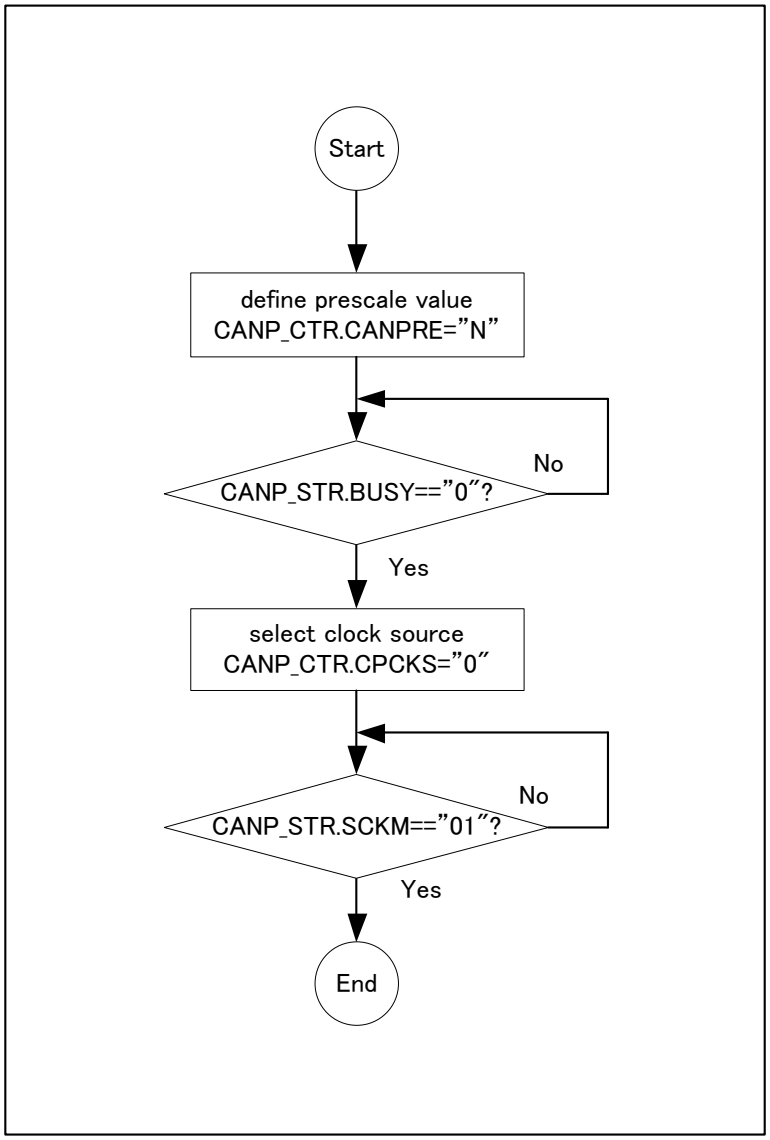
This module has the C division circuit that can change the CAN system clock (fsys) cycle. A division ratio from C = 1 to 64 can be set.

While the division setting is being changed, the status is busy. The next division setting change cannot be accepted until the current division setting is completed. The busy status can be read from the status register. If you make a new change in the busy status, a bus error response is returned.

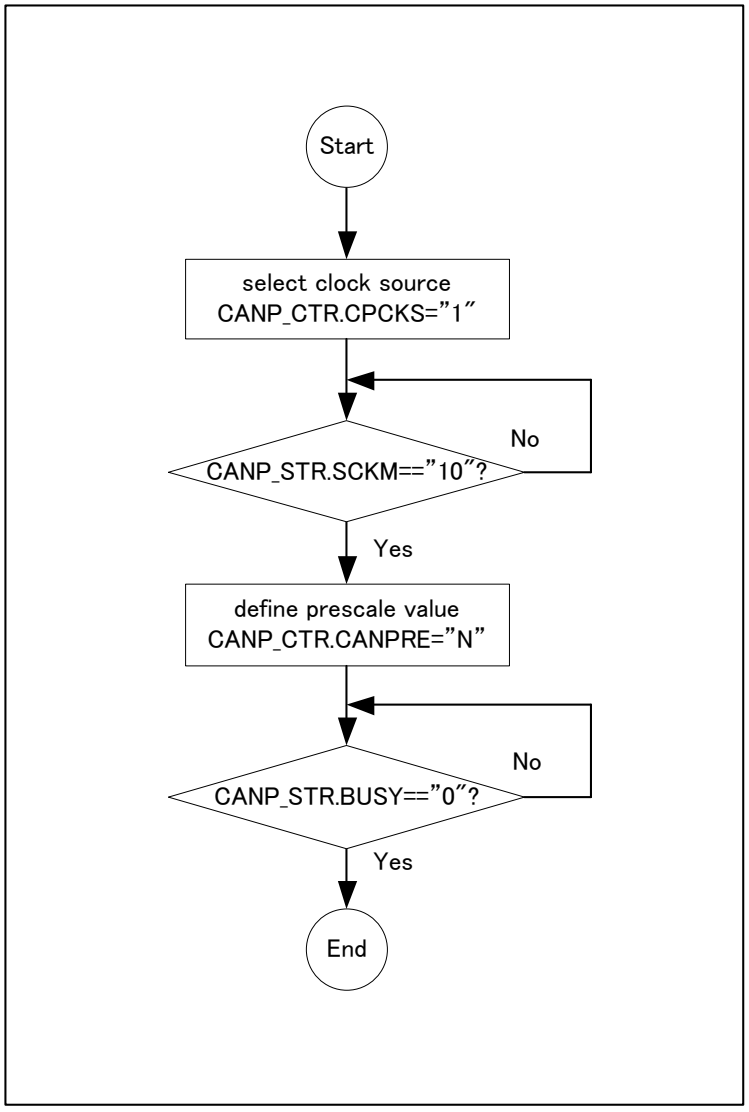
4. Setting Procedure

This section describes the procedure for setting the registers of CAN prescaler.

The following shows the procedure for switching from the main clock to the PLL(product specification) clock.



The following shows the procedure for switching from the PLL(product specification) clock to the main clock.



5. Registers

This section describes the registers of CAN prescaler.

Register List

Table 5-1 Register List

Abbreviated Register Name	Register Name	Reference
CANP_CTR	CAN prescaler control register	5.1
CANP_STR	CAN prescaler status register	5.2

Register Map

Table 5-2 Register Map

Offset	Register Name
0x0000_0000	CANP_CTR 00000000_00000000_00000001_00000000
0x0000_0004	CANP_STR 00000000_00000000_00000000_00000000

5.1. CAN Prescaler Control Register (CANP_CTR)

This register sets the division ratio of the CAN system clock and selects a clock source.

Bit	31															16
Field	Reserved															
R/W Attribute	R0,WX															
Protection Attribute	-															
Initial Value	00000000_00000000															

Bit	15	14	13	12	11	10	9	8
Field	Reserved							CPCKS
R/W Attribute	R0,WX							R/W
Protection Attribute	-							
Initial Value	0000000							1

Bit	7	6	5	4	3	2	1	0
Field	Reserved		CANPRE[5:0]					
R/W Attribute	R0,WX		R/W					
Protection Attribute	--							
Initial Value	00		000000					

[bit31:9] Reserved: Reserved Bits

[bit8] CPCKS: CAN Prescaler Source Clock Selection Bit

This bit selects the source clock to be used for the CAN system clock.

Bit	Description
0	Select the PLL(product specification) clock.
1	Select the main clock [initial value].

Note:

- *Before changing the CAN prescaler source clock selection bit, set the initial bit (CCCR.Init) of the control register of the CAN controller to "0x1" to stop the CAN controller.*

[bit7:6] Reserved: Reserved Bits

[bit5:0] CANPRE: CAN Prescaler Division Setting Bits

These bits set the division ratio from the CAN source clock of the CAN system clock.

Bit5:0						Description
0	0	0	0	0	0	No division [initial value]
0	0	0	0	0	1	Divided by 2
0	0	0	0	1	0	Divided by 3
...						
1	1	1	1	1	0	Divided by 63
1	1	1	1	1	1	Divided by 64

Notes:

- *Before changing the CAN prescaler division setting bits, set the initial bit (CCCR.Init) of the control register of the CAN controller to "0x1" to stop the CAN controller.*
- *While the clock division setting is being changed, the CANP_STR.BUSY bit in the CAN prescaler status register is "1". If a new division ratio is set during the setting change, a bus error response is returned.*
  The new division ratio setting becomes invalid.
- *If you select a clock for which oscillation is stopped or oscillation stabilization wait operation is in progress, the on-chip bus clock is selected. Confirm that a clock is selected based on the setting of the bit by reading CANP_STR.SCKM in the CAN prescaler status register.*
- *If you use the PLL(product specification) clock, confirm that the PLL(product specification) clock is selected by reading CANP_STR.SCKM in the CAN prescaler status register before starting communication with CAN.*
- *The CAN prescaler division setting bits have to be set before PLL(product specification) clock is selected.*

5.2. CAN Prescaler Status Register (CANP_STR)

This register displays the status of CAN prescaler.

Bit	31	8
Field	Reserved	
R/W Attribute	R0,WX	
Protection Attribute	-	
Initial Value	00000000_00000000_00000000	

Bit	7	6	5	4	3	2	1	0
Field	Reserved					SCKM[1:0]		BUSY
R/W Attribute	R0,WX					R,WX		R,WX
Protection Attribute	-							
Initial Value	00000					00		0

[bit31:3] Reserved: Reserved Bits

[bit2:1] SCKM: Source Clock Display

These bits indicate the currently selected source clock.

Bit1:0		Description
0	0	Clock stop
0	1	PLL(product specification) clock
1	0	Main clock
1	1	State in the middle of the source clock selection

Note:

- *In case of the middle of the source clock selection, On-chip bus clock is selected.*

[bit0] BUSY:

This bit indicates the status of switching by the CAN prescaler division setting register.

Bit	Description
0	Clock division setting is completed.
1	Clock division setting is being changed.

Notes:

- *While the clock division setting is being changed, the CANP_STR.BUSY bit in the CAN prescaler status register is "1". If a division ratio is changed to a new one during the setting change, a bus error response is returned and the setting becomes invalid.*

6. Restriction

This section describes the restriction of CAN clock(fsys).

Frequency of Bus clock and CAN clock for CAN macro(CAN FD) have to meet conditions as below.

Bus clock frequency > CAN clock frequency

Therefore CAN macro of MCU Config Group and Common Peripheral Group have to meet conditions as below.

Bus clock(CLK_COMH) frequency for MCU Config Group > CAN clock frequency

Bus clock(CLK_LCP) frequency for Common Peripheral Group > CAN clock frequency

CHAPTER 35: Overview of Multi-function Serial Interface

This chapter provides an overview of the Multi-function Serial Interface.

1. **Overview of the UART (Asynchronous Serial Interface)**
2. **Overview of the CSIO (Clock Synchronous Serial Interface)**
3. **Overview of the LIN Interface (v2.1) (LIN Communication Control Interface (v2.1))**
4. **Overview of the I^2C Interface (I^2C Communication Control Interface)**
5. **Block Diagram**
6. **Registers of the Multi-Function Serial Interface**

MFS-TXXPT03P01R01L08-E1-XX

1. Overview of the UART (Asynchronous Serial Interface)

The UART (asynchronous serial interface) is a general-purpose serial data communication interface for asynchronous communication with external devices. This interface supports the bidirectional communication function (normal mode) and master/slave-type communication function (multi-processor mode: both master and slave roles are supported). In addition, this interface has FIFOs for transmission and reception.

Functions of the UART (Asynchronous Serial Interface)

	Function	Description
1	Data	− Full duplex, double buffering (when FIFOs are not used) − Transmission and reception FIFOs (64 bytes each) (when FIFOs are used)
2	Serial input	− Oversampling is performed 3 times by the bus clock. The reception value is determined by majority decision.
3	Transfer format	− Asynchronous
4	Baud rate	− Dedicated baud rate generator (15-bit reload counter configuration) − The external clock input can be adjusted by the reload counter.
5	Data length	− 5 to 9 bits (for normal mode), or 7 to 8 bits (for multi-processor mode)
6	Signaling method	− NRZ (Non Return to Zero) and inverted NRZ
7	Start bit detection	− Synchronized to falling edges of the start bit (for NRZ method) − Synchronized to rising edges of the start bit (for inverted NRZ method)
8	Reception error detection	− Framing error − Overrun error − Parity error*1
9	Hardware flow control	− Transmission/reception auto control by CTS and RTS
10	Synchronous transmission function	− Can automatically transmit data periodically in synchronization with the serial timer. − Can transmit in synchronization with an external trigger.
11	Timer function	− A 16-bit serial timer is provided. − A division value can be selected for the operation clock (division by 1 to 256). − Activation by external trigger is available.
12	Interrupt request	− Reception interrupt (reception completion, framing error, overrun error, parity error *1, and reception block transfer error) − Reception FIFO interrupt (reception FIFO under run) − Transmission interrupt (transmission data empty, transmission bus idle, and transmission block transfer error) − Transmission FIFO interrupt (when the transmission FIFO is not higher than the interrupt trigger level or when the transmission FIFO is empty, transmission FIFO overrun) − DMA transfer is supported for both transmission and reception. − Status interrupt (serial timer interrupt)
13	Master/slave-type communication function (multi-processor mode)	− 1 (master) -to-n (slave) communication is possible. (Both master and slave systems are supported.)
14	FIFO option	− Transmission and reception FIFOs are provided (64 bytes for the transmission FIFO and 64 bytes for the reception FIFO). − Transmission FIFO and reception FIFO can be selected. − Transmission data can be retransmitted. − The timing of the reception FIFO interrupt can be changed by software. − Independent FIFO reset is supported.

*1 Parity errors are for normal mode only.

2. Overview of the CSIO (Clock Synchronous Serial Interface)

The CSIO (clock synchronous serial interface) is a general-purpose serial data communication interface for synchronous communication with external devices (supports SPI). In addition, this interface has FIFOs for transmission and reception.

Functions of the CSIO (Clock Synchronous Serial Interface)

	Function	Description
1	Data buffer	− Full duplex, double buffering (when FIFOs are not used) − Transmission and reception FIFOs (64 bytes each) (when FIFOs are used)
2	Transfer format	− Clock synchronization (no start bits/stop bits) − Master/slave function − Supports SPI (supports both master and slave)
3	Baud rate	− Dedicated baud rate generator is provided (configured from a 15-bit reload counter during master operation). − External clock input is enabled (during slave operation).
4	Data length	− Variable to 5 to 16, 20, 24, and 32 bits.
5	Reception error detection	− Overrun error
6	Interrupt request	− Reception interrupt (reception completion, overrun error, and reception block transfer error) − Reception FIFO interrupt (reception FIFO under run) − Transmission interrupt (transmission data empty, transmission bus idle, chip error interrupt, and transmission block transfer error) − Transmission FIFO interrupt (when the transmission FIFO is not higher than the interrupt trigger level or when the transmission FIFO is empty, transmission FIFO overrun) − DMA transfer is supported for both transmission and reception. − Status interrupt (serial timer interrupt)
7	Serial chip select	− 4-channel control (independent control, round control) − The setup/hold/deselect times can be set to variable. − The active level can be selected for each channel.
8	Synchronous transmission function	− Can automatically transmit data periodically in synchronization with the serial timer. − Can transmit in synchronization with an external trigger.
9	Timer function	− A 16-bit serial timer is provided. ➢ A division value can be selected for the operation clock (division by 1 to 256). ➢ Activation by external trigger is available.
10	Synchronous mode	− Master or slave function
11	Pin access	− The serial data output pin can be set to "1".
12	FIFO option	− Transmission and reception FIFOs are provided (64 bytes for the transmission FIFO and 64 bytes for the reception FIFO). − Transmission FIFO and reception FIFO can be selected. − Transmission data can be retransmitted. − The timing of the reception FIFO interrupt can be changed by software. − Independent FIFO reset is supported.

3. Overview of the LIN Interface (v2.1) (LIN Communication Control Interface (v2.1))

3.1. Manual Mode

The LIN interface (v2.1) (LIN communication control interface (v2.1)) provides the functions to support the LIN bus. In addition, this interface has FIFOs (64 bytes each) for transmission and reception.

Functions of the LIN Interface (v2.1) (LIN Communication Control Interface (v2.1)) (Manual Mode)

	Function	Description
1	Data buffer	− Full duplex, double buffering (when FIFOs are not used) − Transmission and reception FIFOs (64 bytes each) (when FIFOs are used)
2	Serial input	− Oversampling is performed 3 times by the bus clock. The reception value is determined by majority decision.
3	Transfer mode	− Asynchronous
4	Baud rate	− Dedicated baud rate generator is provided (configured from a 15-bit reload counter). − The external clock can be adjusted by the reload counter. − Auto baud rate adjustment with Sync Field reception.
5	Data length	− 8 bits
6	Signaling method	− NRZ (Non Return to Zero)
7	Start bit detection	− Synchronized to falling edges of the start bit
8	Reception error detection	− Framing error − Overrun error
9	Interrupt request	− Reception interrupt (reception completion, framing error, overrun error, and reception block transfer error) − Reception FIFO interrupt (reception FIFO under run) − Transmission interrupt (transmission data empty, transmission bus idle, and transmission block transfer error) − Status interrupt (LIN Break field detection and serial timer interrupt) − Interrupt request to ICU (LIN Sync field detection: LSYN) − Transmission FIFO interrupt (when the transmission FIFO is not higher than the interrupt trigger level or when the transmission FIFO is empty, transmission FIFO overrun) − DMA transfer is supported for both transmission and reception.
10	Timer function	− A 16-bit serial timer is provided. − A division value can be selected for the operation clock (division by 1 to 256). − Activation by external trigger is available.
11	LIN bus option	− Support for LIN Protocol Revision 2.1 − Master device operation − Slave device operation − LIN Break field generation (changeable from 13 to 16 bits) − LIN Break delimiter generation (changeable from 1 to 4 bits) − LIN Break field detection − Detection of the start/stop edge of the LIN Sync field connected to an input capture
12	FIFO option	− Transmission and reception FIFOs are provided (64 bytes for the transmission FIFO and 64 bytes for the reception FIFO). − Transmission FIFO and reception FIFO can be selected. − Transmission data can be retransmitted. − The timing of the reception FIFO interrupt can be changed by software. − Independent FIFO reset is supported.

Note:

− *The Wake Up function of the LIN is not supported.*

3.2. Assist Mode

The LIN interface (v2.1) (LIN communication control interface (v2.1)) provides the functions to support the LIN bus. The header section can be automatically transmitted/detected in LIN communication. In addition, this interface has FIFOs (64 bytes each) for transmission and reception.

Functions of the LIN Interface (v2.1) (LIN Communication Control Interface (v2.1)) (Assist Mode)

	Function	Description
1	Data buffer	− Full duplex, double buffering (when FIFOs are not used) − Transmission and reception FIFOs (64 bytes each) (when FIFOs are used)
2	Serial input	− Oversampling is performed 3 times by the bus clock. The reception value is determined by majority decision.
3	Transfer mode	− Asynchronous
4	Baud rate	− Dedicated baud rate generator is provided (configured from a 15-bit reload counter). − The external clock can be adjusted by the reload counter. − Auto baud rate adjustment with Sync Field reception.
5	Data length	− 8 bits
6	Signaling method	− NRZ (Non Return to Zero)
7	Start bit detection	− Synchronized to falling edges of the start bit
8	Reception error detection	− Framing error − Overrun error − LIN bus error − LIN ID parity error − LIN checksum error
9	Interrupt request	− Transmission interrupt ➢ Data transmission interrupt (transmission data empty, transmission bus idle, and transmission block transfer error) ➢ Transmission FIFO interrupt (when the transmission FIFO is not higher than the interrupt trigger level or when the transmission FIFO is empty, transmission FIFO overrun) − Reception interrupt ➢ Data reception interrupt (reception completion and reception block transfer error) ➢ Reception FIFO interrupt (when the reception FIFO is higher than the interrupt threshold, reception FIFO under run) ➢ Various types of error interrupts (LIN bus error, LIN Sync Data error, LIN ID parity error, framing error, overrun error, and LIN checksum error) − Status interrupt ➢ Auto header completion interrupt ➢ Sync Field detection interrupt ➢ Checksum calculation completion interrupt − DMA transfer is supported for both transmission and reception.
10	Timer function	− A 16-bit serial timer is provided. − A division value can be selected for the operation clock (division by 1 to 256). − Activation by external trigger is available.

	Function	Description
11	LIN bus option	− Support for LIN Protocol Revision 2.1 − Automatic transmission/reception of the header of the master/slave device ➢ LIN Break field generation (changeable from 13 to 20 bits) ➢ LIN Break delimiter generation (changeable from 1 to 4 bits) ➢ Automatic generation of the Sync Field and automatic check of data value (0x55) ➢ Automatic generation/check of the parity value of the ID Field ➢ Automatic generation/check of a checksum (support for standard/extended checksum)
12	FIFO option	− Transmission and reception FIFOs are provided (64 bytes for the transmission FIFO and 64 bytes for the reception FIFO). − Transmission FIFO and reception FIFO can be selected. − Transmission data can be retransmitted. − The timing of the transmission/reception FIFO interrupt can be changed by software. − Independent FIFO reset is supported.
13	LIN communication test function	− Serial communication test function − Pseudo error generation functions (LIN bus error, LIN ID parity error, LIN checksum error, LIN Sync Field error, and framing error)

Note:

− *The Wake Up function of the LIN is not supported.*

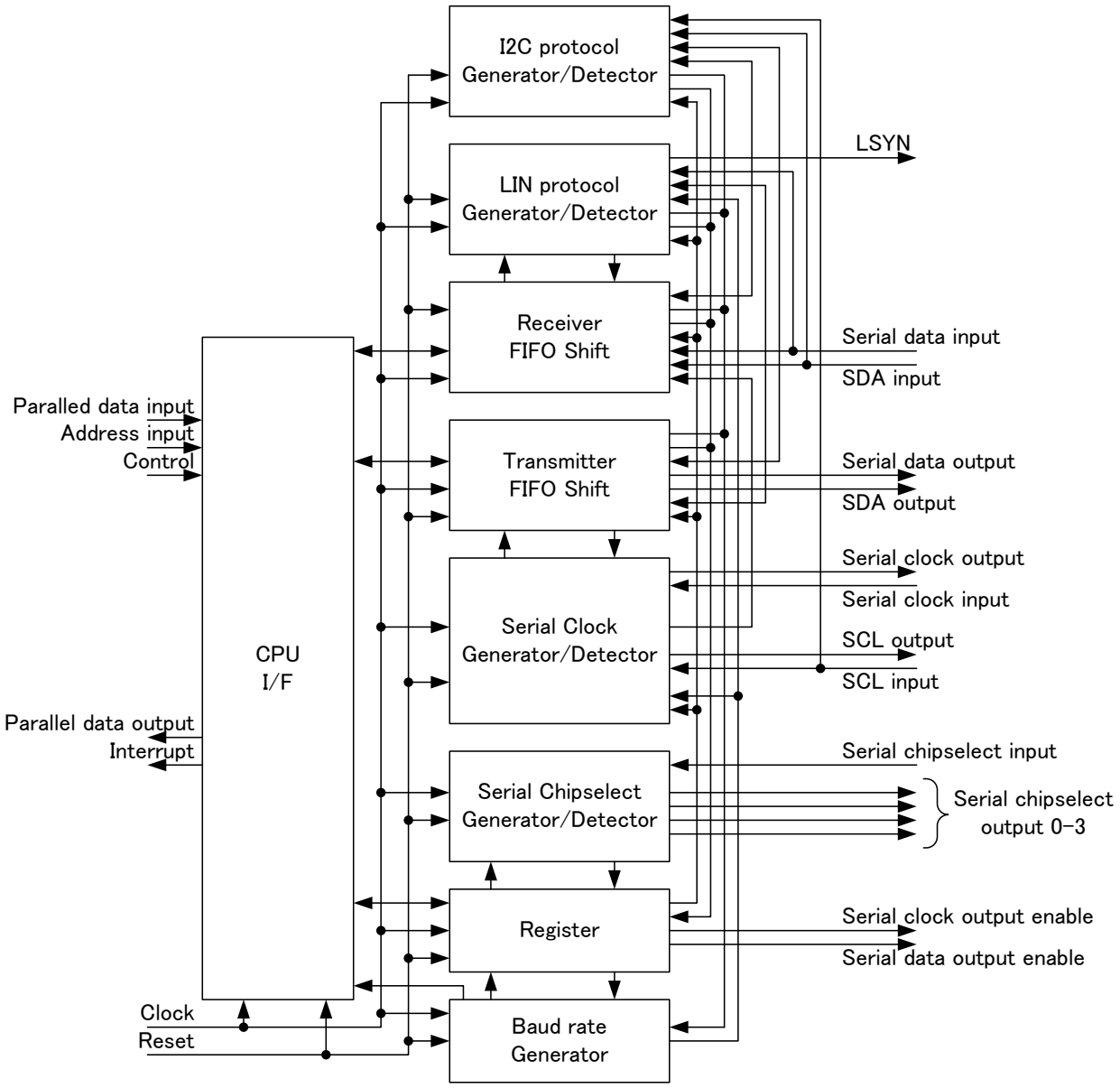
4. Overview of the I²C Interface (I²C Communication Control Interface)

The I²C interface (I²C communication control interface) supports the I²C bus and operates as a master/slave device on the I²C bus. In addition, this interface has FIFOs (64 bytes each) for transmission and reception.

Functions of the I²C Interface (I²C Communication Control Interface)

	Function	Description
1	Data buffer	− Full duplex, double buffering (when FIFOs are not used) − Transmission and reception FIFOs (64 bytes each) (when FIFOs are used)
2	Serial input	− For serial clock and data input, noise from 2 to 38 bus clocks is filtered out.
3	Transfer mode	− Synchronous
4	Baud rate	− Dedicated baud rate generator is provided (configured from a 15-bit reload counter).
5	Data length	− 8 bits
6	Signaling method	− NRZ (Non Return to Zero)
7	Interrupt request	− Reception interrupt − Transmission interrupt − Status interrupt (INT interrupt, stop condition interrupt, repeated start detection interrupt, and serial timer interrupt) − Transmission FIFO interrupt (when the transmission FIFO is not higher than the interrupt trigger level or when the transmission FIFO is empty, reception FIFO under run) − Reception FIFO interrupt (reception FIFO under run) − DMA transfer is supported for both transmission and reception.
8	I²C	− Master/slave transmission and reception function − Arbitration function − Clock synchronization function − Transfer direction detection function − Function to generate and detect the repeated start condition − Bus error detection function − General call addressing function − 7-bit addressing as master and slave − An interrupt can be generated upon transfer and bus error occurrence. − 10-bit addressing function can be supported by program. − Standard-mode and Fast-mode are supported. Fast-mode Plus, High-speed mode, and Ultra Fast-mode are not supported.
9	Timer function	− A 16-bit serial timer is provided. − A division value can be selected for the operation clock (division by 1 to 256). − Activation by external trigger is available.
10	FIFO	− Transmission and reception FIFOs are provided (64 bytes for the transmission FIFO and 64 bytes for the reception FIFO). − Transmission FIFO and reception FIFO can be selected. − Transmission data can be retransmitted. − The timing of the reception FIFO interrupt can be changed by software. − Independent FIFO reset is supported.

5. Block Diagram



Explanation of Each Block

Baud Rate Generator

This 15-bit reload counter functions as a dedicated baud rate generator. This block consists of 15-bit registers for reload values, and it generates reception and transmission clocks and LIN Break field detection clocks from external or internal clocks. Also, the count value of the transmission reload counter is read from BGR1 or BGR0.

Receiver/FIFO Shift

This block consists of a reception shift register, reception FIFO, reception bit counter, start bit detection circuit, and reception parity counter. When the reception bit counter counts the reception data bit and completes the reception of 1 piece of data corresponding to the set data length, the reception data full flag bit (SSR:RDRF) is set to "1". At this time, if reception interrupts are enabled, it generates a reception interrupt request. The start bit detection circuit detects the start bit from a serial input signal, synchronizes the falling edge of the start bit, and sends a signal to the reload counter. The reception parity counter calculates the reception data parity.

The reception shift register fetches the received data input from the SIN (serial data input) pin while bits are shifted. After the completion of reception, the register transfers the received data to the RDR register, or to the reception FIFO when using the reception FIFO.

Transmitter/FIFO Shift

This block consists of a transmission shift register, transmission bit counter, transmission start circuit, and transmission parity counter. The transmission bit counter counts the transmission data bits and transmits 1 piece of data corresponding to the set data length. When the transmission bit counter indicates the start of write data transmission, a flag is set in the serial status register. At this time, if transmission interrupts are enabled, it generates a transmission interrupt request. The transmission start circuit starts a transmission operation with data writing of the TDR register. If the transmission parity counter uses parity bits, it generates a parity bit for the transmission data.

The transmission shift register transfers the written data in the TDR register, or in the transmission FIFO when using the transmission FIFO, to the transmission shift register and outputs it to SOUT (serial data output) while bits are shifted.

Serial Clock Generator/Detector

The serial clock generator generates a serial clock during master operations in CSIO or I^2C mode.

The serial clock detector detects the serial clock during slave operations in CSIO or I^2C mode. There is synchronization with the detected serial clock to transmit and receive serial data.

LIN Protocol Generator/Detector

The LIN protocol Detector detects the LIN Break field when the LIN master node transmits the message header. Upon detecting the LIN Break field, the detector sets "1" in the LIN Break field detection flag bit (SSR:LBD). It outputs an internal signal (LSYN) to a capture unit to detect the first and the fifth falling edges of the LIN Sync field and to measure the synchronization of the actual serial clock transmitted by the LIN master node.

The LIN protocol Generator generates the LIN break field corresponding to the length selected with the LIN break field length selection bit in the extended status control register.

I²C Protocol Generator/Detector

The I²C protocol detector detects start and stop conditions and monitors the status of the I²C bus.

The I²C protocol generator generates start and stop conditions and controls wait for the I²C bus.

Serial Chip Select Generator/Detector

The serial chip select generator controls each chip select pin during master operations in CSIO mode.

The serial chip select detector monitors the chip select input status during slave operations in CSIO mode.

Register

This is the setting part of each register.

CPU I/F

This is the interface part between the bus and a register.

6. Registers of the Multi-Function Serial Interface

This section lists the multi-function serial interface registers.

operation mode	register select signal	DATAO[31:24]/ DATAI[31:24]	DATAO[23:16]/ DATAI[23:16]	DATAO[15:8]/ DATAI[15:8]	DATAO[7:0]/ DATAI[7:0]
mode0 to 3	SSRSEL, ESCRSEL SCRSEL, SMRSEL	SSR[7:0]	ESCR[7:0]	SCR[7:0]	SMR[7:0]
mode4			IBSR[7:0]	IBCR[7:0]	
mode0, 1	RDRSEL[1:0]	-	-	RDR[9]	RDR[7:0]
mode2		RDR[31:24]	RDR[23:16]	RDR[15:8]	
mode3, 4		-	-	-	
mode0 to 4	STMRSEL[1:0] SACSRSEL[1:0]	STMR[15:8]	STMR[7:0]	SACSR[15:8]	SACSR[7:0]
mode0, 1	SCSCRSEL [1:0] STMCRSEL [1:0]	-	-	STMCR[15:8]	STMCR[7:0]
mode2		SCSCR[15:8]	SCSCR[7:0]		
mode3		SFUR[15:8]	SFUR[7:0]		
mode4		-	-		
mode0, 1	SCSTRSEL [3:0]	-	-	-	-
mode2		SCSTR[31:24]	SCSTR[23:16]	SCSTR[15:8]	SCSTR[7:0]
mode3		LAMSR[7:0]	LAMCR[7:0]	SFLR[15:8]	SFLR[7:0]
mode4		-	-	EIBCR[7:0]	NFCR[7:0]
mode0, 1	SCSFRSEL [2:0]	-	-	-	-
mode2			SCSFR[23:16]	SCSFR[15:8]	SCSFR[7:0]
mode3, 4			-	-	-
mode0, 1	TBYTESEL [3:0]	-	-	-	TBYTE[7:0]
mode2		TBYTE[13:24]	TBYTE[23:16]	TBYTE[15:8]	
mode3		LAMESR[7:0]	LAMERT[7:0]	LAMIER[7:0]	LAMRID[7:0]/ LAMTID[7:0]
mode4		-	-	-	-
mode0 to 3	ISMKSEL ISBASEL BGR1SEL	-	-	BGR1[7:0]	BGR0[7:0]
mode4	BGR0SEL	ISMK[7:0]	ISBA[7:0]		
mode0 to 4	FBYTE2SEL FBYTE1SEL FCR1SEL FCR0SEL	FBYTE2[7:0]	FBYTE1[7:0]	FCR1[7:0]	FCR0[7:0]
mode0 to 4	FTICR2SEL FTICR1SEL	-	-	FTICR2[7:0]	FTICR1[7:0]

3050462701

List of clear registers

operation mode	register select signal	DATAO[31:24]/ DATAI[31:24]	DATAO[23:16]/ DATAI[23:16]	DATAO[15:8]/ DATAI[15:8]	DATAO[7:0]/ DATAI[7:0]
mode3	SSRSEL, ESCRSEL SCRSEL, SMRSEL	SSRC[7:0]	ESCRC[7:0]	SCRC[7:0]	SMRC[7:0]
mode4		SSRC[7:0]	IBSRC[7:0]	IBCRC[7:0]	SMRC[7:0]
mode0 to 4	STMRSEL[1:0] SACSRSEL[1:0]	-	-	SACSRC[15:8]	SACSRC[7:0]
mode3	SCSTRSEL[3:0]	LAMSRC[7:0]	LAMCRC[7:0]	-	-
mode3	TBYTESEL[3:0]	LAMESRC[7:0]	-	LAMIERC[7:0]	-
mode0 to 4	FBYTE2SEL FBYTE1SEL FCR1SEL FCR0SEL	-	-	FCR1C[7:0]	FCR0C[7:0]

3050462711

List of set registers

operation mode	register select signal	DATAO[31:24]/ DATAI[31:24]	DATAO[23:16]/ DATAI[23:16]	DATAO[15:8]/ DATAI[15:8]	DATAO[7:0]/ DATAI[7:0]
mode3	SSRSEL, ESCRSEL SCRSEL, SMRSEL	SSRS[7:0]	ESCRS[7:0]	SCRS[7:0]	SMRS[7:0]
mode4		SSRS[7:0]	-	IBCRS[7:0]	SMRS[7:0]
mode0 to 4	STMRSEL[1:0] SACSRSEL[1:0]	-	-	SACSRS[15:8]	SACSRS[7:0]
mode3	SCSTRSEL[3:0]	-	LAMCRS[7:0]	-	-
mode3	TBYTESEL[3:0]	-	-	LAMIERS[7:0]	-
mode0 to 4	FBYTE2SEL FBYTE1SEL FCR1SEL FCR0SEL	-	-	FCR1S[7:0]	FCR0S[7:0]

3050462721

Register Map

Table 6-1 Register Map of UART/CSIO/LIN/I2C in Each Modes

UART

MCU Config Group (Channel No.:0, 1, and2)

Offset	Register Name			
	+3	+2	+1	+0
0x000	MCG_MFSxx_SSR	MCG_MFSxx_ESCR	MCG_MFSxx_SCR	MCG_MFSxx_SMR
0x004	-		MCG_MFSxx_RDR[8] (MCG_MFSxx_TDR[8])	MCG_MFSxx_RDR[7:0] (MCG_MFSxx_TDR[7:0])
0x008	MCG_MFSxx_STMR[15:0]		MCG_MFSxx_SACSR[15:0]	
0x00C	-		MCG_MFSxx_STMCR[15:0]	
0x010	-			
0x014	-			
0x018	-			MCG_MFSxx_TBYTE
0x01C	-		MCG_MFSxx_BGR1	MCG_MFSxx_BGR0
0x020	MCG_MFSxx_FBYTE2	MCG_MFSxx_FBYTE1	MCG_MFSxx_FCR1	MCG_MFSxx_FCR0
0x024	MCG_MFSxx_ESR	MCG_MFSxx_ECR	MCG_MFSxx_FTICR2	MCG_MFSxx_FTICR1
0x028	-			MCG_MFSxx_TBSIZE
0x02C	-			
0x030	-		MCG_MFSxx_SACSRC	
0x034	-			
0x038	-			
0x03C	-		MCG_MFSxx_FCR1C	MCG_MFSxx_FCR0C
0x040	MCG_MFSxx_ESRC	-	-	
0x044	-			
0x048	-		MCG_MFSxx_SACSRS	
0x04C	-			
0x050	--			
0x054	-		MCG_MFSxx_FCR1S	MCG_MFSxx_FCR0S
0x058 to 0x3FC	-			

Notes:

- *xx is the channel number. (0 to2)*
- *It depends on the product spec whether to use MCG_MFSxx _ *** register group. Please refer to Hardware Manual or Data Sheet for more details.*

Common Peripheral #0 Group (Channel No.: 0, 1, 2, 3, 4, 5, 6, and 7)

Common Peripheral #1 Group (Channel No.: 8, 9, 10, 11, 12, 13, 14, and 15)

Offset	Register Name			
	+3	+2	+1	+0
0x000	CPG_MFSxx_SSR	CPG_MFSxx_ESCR	CPG_MFSxx_SCR	CPG_MFSxx_SMR
0x004	-		CPG_MFSxx_RDR[8] (CPG_MFSxx_TDR[8])	CPG_MFSxx_RDR[7:0] (CPG_MFSxx_TDR[7:0])
0x008	CPG_MFSxx_STMR[15:0]		CPG_MFSxx_SACSR[15:0]	
0x00C	-		CPG_MFSxx_STMCR[15:0]	
0x010	-			
0x014	-			
0x018	-			CPG_MFSxx_TBYTE
0x01C	-		CPG_MFSxx_BGR1	CPG_MFSxx_BGR0
0x020	CPG_MFSxx_FBYTE2	CPG_MFSxx_FBYTE1	CPG_MFSxx_FCR1	CPG_MFSxx_FCR0
0x024	CPG_MFSxx_ESR	CPG_MFSxx_ECR	CPG_MFSxx_FTICR2	CPG_MFSxx_FTICR1
0x028	-			CPG_MFSxx_TBSIZE
0x02C	-			
0x030	-		CPG_MFSxx_SACSRC	
0x034	-			
0x038	-			
0x03C	-		CPG_MFSxx_FCR1C	CPG_MFSxx_FCR0C
0x040	CPG_MFSxx_ESRC	-	-	
0x044	-			
0x048	-		CPG_MFSxx_SACSRS	
0x04C	-			
0x050	--			
0x054	-		CPG_MFSxx_FCR1S	CPG_MFSxx_FCR0S
0x058 to 0x3FC	-			

Notes:

- *xx is the channel number. (0 to 15)*
- *It depends on the product spec whether to use CPG_MFSxx _ *** register group. Please refer to Hardware Manual or Data Sheet for more details.*

CSIO

MCU Config Group (Channel No.:0, 1, and2)

Offset	Register Name			
	+3	+2	+1	+0
0x000	MCG_MFSxx_SSR	MCG_MFSxx_ESCR	MCG_MFSxx_SCR	MCG_MFSxx_SMR
0x004	MCG_MFSxx_RDR(CPG_MFSxx_TDR)			
0x008	MCG_MFSxx_STMR		MCG_MFSxx_SACSR	
0x00C	MCG_MFSxx_SCSCR		MCG_MFSxx_STMCR	
0x010	MCG_MFSxx_SCSTR			
0x014	-	MCG_MFSxx_SCSFR2	MCG_MFSxx_SCSFR1	MCG_MFSxx_SCSFR0
0x018	MCG_MFSxx_TBYTE3	MCG_MFSxx_TBYTE2	MCG_MFSxx_TBYTE1	MCG_MFSxx_TBYTE0
0x01C	-		MCG_MFSxx_BGR1	MCG_MFSxx_BGR0
0x020	MCG_MFSxx_FBYTE2	MCG_MFSxx_FBYTE1	MCG_MFSxx_FCR1	MCG_MFSxx_FCR0
0x024	MCG_MFSxx_ESR	MCG_MFSxx_ECR	MCG_MFSxx_FTICR2	MCG_MFSxx_FTICR1
0x028				MCG_MFSxx_TBSIZE
0x02C	-			
0x030			MCG_MFSxx_SACSRC	
0x034	-			
0x038	-			
0x03C			MCG_MFSxx_FCR1C	MCG_MFSxx_FCR0C
0x040	MCG_MFSxx_ESRC	-		
0x044	-			
0x048	-		MCG_MFSxx_SACSRS	
0x04C	-			
0x050	-			
0x054	-		MCG_MFSxx_FCR1S	MCG_MFSxx_FCR0S
0x058 to 0x3FC	-			

Notes:

- *xx is the channel number. (0 to2)*
- *It depends on the product spec whether to use MCG_MFSxx _ *** register group.*
- *Please refer to Hardware Manual or Data Sheet for more details.*

Common Peripheral #0 Group (Channel No.: 0, 1, 2, 3, 4, 5, 6, and 7)

Common Peripheral #1 Group (Channel No.: 8, 9, 10, 11, 12, 13, 14, and 15)

Offset	Register Name			
	+3	+2	+1	+0
0x000	CPG_MFSxx_SSR	CPG_MFSxx_ESCR	CPG_MFSxx_SCR	CPG_MFSxx_SMR
0x004	CPG_MFSxx_RDR(CPG_MFSxx_TDR)			
0x008	CPG_MFSxx_STMR		CPG_MFSxx_SACSR	
0x00C	CPG_MFSxx_SCSCR		CPG_MFSxx_STMCR	
0x010	CPG_MFSxx_SCSTR			
0x014	-	CPG_MFSxx_SCSFR2	CPG_MFSxx_SCSFR1	CPG_MFSxx_SCSFR0
0x018	CPG_MFSxx_TBYTE3	CPG_MFSxx_TBYTE2	CPG_MFSxx_TBYTE1	CPG_MFSxx_TBYTE0
0x01C	-		CPG_MFSxx_BGR1	CPG_MFSxx_BGR0
0x020	CPG_MFSxx_FBYTE2	CPG_MFSxx_FBYTE1	CPG_MFSxx_FCR1	CPG_MFSxx_FCR0
0x024	CPG_MFSxx_ESR	CPG_MFSxx_ECR	CPG_MFSxx_FTICR2	CPG_MFSxx_FTICR1
0x028				CPG_MFSxx_TBSIZE
0x02C	-			
0x030			CPG_MFSxx_SACSRC	
0x034	-			
0x038	-			
0x03C			CPG_MFSxx_FCR1C	CPG_MFSxx_FCR0C
0x040	CPG_MFSxx_ESRC	-		
0x044	-			
0x048	-		CPG_MFSxx_SACSRS	
0x04C	-			
0x050	-			
0x054	-		CPG_MFSxx_FCR1S	CPG_MFSxx_FCR0S
0x058 to 0x3FC	-			

Notes:

- *xx is the channel number. (0 to 15)*
- *It depends on the product spec whether to use CPG_MFSxx _ *** register group.*
- *Please refer to Hardware Manual or Data Sheet for more details.*

LIN

MCU Config Group (Channel No.:0, 1, and2)

Offset	Register Name			
	+3	+2	+1	+0
0x000	MCG_MFSxx_SSR	MCG_MFSxx_ESCR	MCG_MFSxx_SCR	MCG_MFSxx_SMR
0x004	-			MCG_MFSxx_RDR (MCG_MFSxx_TDR)
0x008	MCG_MFSxx_STMR		MCG_MFSxx_SACSR	
0x00C	MCG_MFSxx_SFUR		MCG_MFSxx_STMCR	
0x010	MCG_MFSxx_LAMSR	MCG_MFSxx_LAMCR	MCG_MFSxx_SFLR	
0x014	-			
0x018	MCG_MFSxx_LAMESR	MCG_MFSxx_LAMERT	MCG_MFSxx_LAMIER	MCG_MFSxx_LAMRID (MCG_MFSxx_LAMTID)
0x01C	-		MCG_MFSxx_BGR1	MCG_MFSxx_BGR0
0x020	MCG_MFSxx_FBYTE2	MCG_MFSxx_FBYTE1	MCG_MFSxx_FCR1	MCG_MFSxx_FCR0
0x024	MCG_MFSxx_ESR	MCG_MFSxx_ECR	MCG_MFSxx_FTICR2	MCG_MFSxx_FTICR1
0x028	-	-	-	MCG_MFSxx_TBSIZE
0x02C	MCG_MFSxx_SSRC	MCG_MFSxx_ESCRC	MCG_MFSxx_SCRC	MCG_MFSxx_SMRC
0x030	-		MCG_MFSxx_SACSRC	
0x034	MCG_MFSxx_LAMSRC	MCG_MFSxx_LAMCRC	-	
0x038	MCG_MFSxx_LAMESRC	-	MCG_MFSxx_LAMIERC	-
0x03C	-		MCG_MFSxx_FCR1C	MCG_MFSxx_FCR0C
0x040	MCG_MFSxx_ESRC	-		
0x044	MCG_MFSxx_SSRS	MCG_MFSxx_ESCRS	MCG_MFSxx_SCRS	MCG_MFSxx_SMRS
0x048	-		MCG_MFSxx_SACSRS	
0x04C	-	MCG_MFSxx_LAMCRS	-	
0x050	-		MCG_MFSxx_LAMIERS	-
0x054	-		MCG_MFSxx_FCR1S	MCG_MFSxx_FCR0S
0x058 to 0x3FC	-			

Notes:

- *xx is the channel number. (0 to2)*
- *It depends on the product spec whether to use MCG_MFSxx _ *** register group. Please refer to Hardware Manual or Data Sheet for more details.*

Common Peripheral #0 Group (Channel No.: 0, 1, 2, 3, 4, 5, 6, and 7)

Common Peripheral #1 Group (Channel No.: 8, 9, 10, 11, 12, 13, 14, and 15)

Offset	Register Name			
	+3	+2	+1	+0
0x000	CPG_MFSxx_SSR	CPG_MFSxx_ESCR	CPG_MFSxx_SCR	CPG_MFSxx_SMR
0x004	-			CPG_MFSxx_RDR (CPG_MFSxx_TDR)
0x008	CPG_MFSxx_STMR		CPG_MFSxx_SACSR	
0x00C	CPG_MFSxx_SFUR		CPG_MFSxx_STMCR	
0x010	CPG_MFSxx_LAMSR	CPG_MFSxx_LAMCR	CPG_MFSxx_SFLR	
0x014	-			
0x018	CPG_MFSxx_LAMESR	CPG_MFSxx_LAMERT	CPG_MFSxx_LAMIER	CPG_MFSxx_LAMRID (CPG_MFSxx_LAMTID)
0x01C	-		CPG_MFSxx_BGR1	CPG_MFSxx_BGR0
0x020	CPG_MFSxx_FBYTE2	CPG_MFSxx_FBYTE1	CPG_MFSxx_FCR1	CPG_MFSxx_FCR0
0x024	CPG_MFSxx_ESR	CPG_MFSxx_ECR	CPG_MFSxx_FTICR2	CPG_MFSxx_FTICR1
0x028	-	-	-	CPG_MFSxx_TBSIZE
0x02C	CPG_MFSxx_SSRC	CPG_MFSxx_ESCRC	CPG_MFSxx_SCRC	CPG_MFSxx_SMRC
0x030	-		CPG_MFSxx_SACSRC	
0x034	CPG_MFSxx_LAMSRC	CPG_MFSxx_LAMCRC	-	
0x038	CPG_MFSxx_LAMESRC	-	CPG_MFSxx_LAMIERC	-
0x03C	-		CPG_MFSxx_FCR1C	CPG_MFSxx_FCR0C
0x040	CPG_MFSxx_ESRC	-		
0x044	CPG_MFSxx_SSRS	CPG_MFSxx_ESCRS	CPG_MFSxx_SCRS	CPG_MFSxx_SMRS
0x048	-		CPG_MFSxx_SACSRS	
0x04C	-	CPG_MFSxx_LAMCRS	-	
0x050	-		CPG_MFSxx_LAMIERS	-
0x054	-		CPG_MFSxx_FCR1S	CPG_MFSxx_FCR0S
0x058 to 0x3FC	-			

Notes:

- *xx is the channel number. (0 to 15)*
- *It depends on the product spec whether to use CPG_MFSxx _ *** register group.*
- *Please refer to Hardware Manual or Data Sheet for more details.*

I2C

MCU Config Group (Channel No.:0, 1, and2)

Offset	Register Name			
	+3	+2	+1	+0
0x000	MCG_MFSxx_SSR	MCG_MFSxx_IBSR	MCG_MFSxx_IBCR	MCG_MFSxx_SMR
0x004	-			MCG_MFSxx_RDR (MCG_MFSxx_TDR)
0x008	MCG_MFSxx_STMR		MCG_MFSxx_SACSR	
0x00C	-		MCG_MFSxx_STMCR	
0x010	-		MCG_MFSxx_EIBCR	MCG_MFSxx_NFCR
0x014	-			
0x018	-			
0x01C	MCG_MFSxx_ISMK	MCG_MFSxx_ISBA	MCG_MFSxx_BGR1	MCG_MFSxx_BGR0
0x020	MCG_MFSxx_FBYTE2	MCG_MFSxx_FBYTE1	MCG_MFSxx_FCR1	MCG_MFSxx_FCR0
0x024	MCG_MFSxx_ESR	MCG_MFSxx_ECR	MCG_MFSxx_FTICR2	MCG_MFSxx_FTICR1
0x028	-			MCG_MFSxx_TBSIZE
0x02C	MCG_MFSxx_SSRC	MCG_MFSxx_IBSRC	MCG_MFSxx_IBCRC	MCG_MFSxx_SMRC
0x030	-		MCG_MFSxx_SACSRC	
0x034	-			
0x038	-			
0x03C	-		MCG_MFSxx_FCR1C	MCG_MFSxx_FCR0C
0x040	MCG_MFSxx_ESRC	-		
0x044	MCG_MFSxx_SSRS	-	MCG_MFSxx_IBCRS	MCG_MFSxx_SMRS
0x048	-		MCG_MFSxx_SACSRS	
0x04C	-			
0x050	-			
0x054	-		MCG_MFSxx_FCR1S	MCG_MFSxx_FCR0S
0x058 to 0x3FC	-			

Notes:

- *xx is the channel number. (0 to2)*
- *It depends on the product spec whether to use MCG_MFSxx _ *** register group. Please refer to Hardware Manual or Data Sheet for more details.*

Common Peripheral #0 Group (Channel No.: 0, 1, 2, 3, 4, 5, 6, and 7)

Common Peripheral #1 Group (Channel No.: 8, 9, 10, 11, 12, 13, 14, and 15)

Offset	Register Name			
	+3	+2	+1	+0
0x000	CPG_MFSxx_SSR	CPG_MFSxx_IBSR	CPG_MFSxx_IBCR	CPG_MFSxx_SMR
0x004	-			CPG_MFSxx_RDR (CPG_MFSxx_TDR)
0x008	CPG_MFSxx_STMR		CPG_MFSxx_SACSR	
0x00C	-		CPG_MFSxx_STMCR	
0x010	-		CPG_MFSxx_EIBCR	CPG_MFSxx_NFCR
0x014	-			
0x018	-			
0x01C	CPG_MFSxx_ISMK	CPG_MFSxx_ISBA	CPG_MFSxx_BGR1	CPG_MFSxx_BGR0
0x020	CPG_MFSxx_FBYTE2	CPG_MFSxx_FBYTE1	CPG_MFSxx_FCR1	CPG_MFSxx_FCR0
0x024	CPG_MFSxx_ESR	CPG_MFSxx_ECR	CPG_MFSxx_FTICR2	CPG_MFSxx_FTICR1
0x028	-			CPG_MFSxx_TBSIZE
0x02C	CPG_MFSxx_SSRC	CPG_MFSxx_IBSRC	CPG_MFSxx_IBCRC	CPG_MFSxx_SMRC
0x030	-		CPG_MFSxx_SACSRC	
0x034	-			
0x038	-			
0x03C	-		CPG_MFSxx_FCR1C	CPG_MFSxx_FCR0C
0x040	CPG_MFSxx_ESRC	-		
0x044	CPG_MFSxx_SSRS	-	CPG_MFSxx_IBCRS	CPG_MFSxx_SMRS
0x048	-		CPG_MFSxx_SACSRS	
0x04C	-			
0x050	-			
0x054	-		CPG_MFSxx_FCR1S	CPG_MFSxx_FCR0S
0x058 to 0x3FC	-			

Notes:

- *xx is the channel number. (0 to 15)*
- *It depends on the product spec whether to use CPG_MFSxx _ *** register group.*
- *Please refer to Hardware Manual or Data Sheet for more details.*

Registers	Mode	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SCR /SMR	0/1	UPCL	-	-	RIE	TIE	TBIE	RXE	TXE	MD2	MD1	MD0	WUCR	SBL	BDS	-	SOE
	2	UPCL	MS	SPI	RIE	TIE	TBIE	RXE	TXE	MD2	MD1	MD0	WUCR	SCINV	BDS	SCKE	SOE
	3	UPCL	MS	LBR	RIE	TIE	TBIE	RXE	TXE	MD2	MD1	MD0	WUCR	SBL	-	-	SOE
IBCR /SMR	4	MSS	ACT/ SCC	SCKE	WSEL	CNDE	INTE	BER	INT	MD2	MD1	MD0	WUCR	RIE	TIE	-	-
SSR /ESCR	0/1	REC	-	PE	FRE	ORE	RDRF	TDRE	TBI	FLWEN	ESBL	INV	PEN	P	L2	L1	L0
	2	REC	-	-	AWC	ORE	RDRF	TDRE	TBI	SOP	L3	CSFE	WT1	WT0	L2	L1	L0
	3	REC	-	LBD	FRE	ORE	RDRF	TDRE	TBI	-	ESBL	LBL2	LBIE	LBL1	LBL0	DEL1	DEL0
SSR/IBSR	4	REC	TEST	DMA	TBIE	ORE	RDRF	TDRE	TBI	FBT	RACK	RSA	TRX	AL	RSC	SPC	BB
TDR1/0 (RDR1/0)	0/1	-	-	-	-	-	-	-	D8	D7	D6	D5	D4	D3	D2	D1	D0
	2	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
	3	-	-	-	-	-	-	-	-	D7	D6	D5	D4	D3	D2	D1	D0
	4	-	-	-	-	-	-	-	-	D7	D6	D5	D4	D3	D2	D1	D0
-	0/1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
TDR3/2 (RDR3/2)	2	D31	D30	D29	D28	D27	D26	D25	D24	D23	D22	D21	D20	D19	D18	D17	D16
-	3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SACSR 1/0	0/1	STST	-	-	-	-	TRG1	TRG0	TINT	TINTE	TSYNE	TRGE	TDIV3	TDIV2	TDIV1	TDIV0	TMRE
	2	STST	-	TBEEN	CSEIE	CSE	TRG1	TRG0	TINT	TINTE	TSYNE	TRGE	TDIV3	TDIV2	TDIV1	TDIV0	TMRE
	3	STST	BST	SFD	SFDE	AUTE	TRG1	TRG0	TINT	TINTE	-	TRGE	TDIV3	TDIV2	TDIV1	TDIV0	TMRE
	4	-	-	-	-	-	TRG1	TRG0	TINT	TINTE	-	TRGE	TDIV3	TDIV2	TDIV1	TDIV0	TMRE
STMR 1/0	0/1	TM15	TM14	TM13	TM12	TM11	TM10	TM9	TM8	TM7	TM6	TM5	TM4	TM3	TM2	TM1	TM0
	2	TM15	TM14	TM13	TM12	TM11	TM10	TM9	TM8	TM7	TM6	TM5	TM4	TM3	TM2	TM1	TM0
	3	TM15	TM14	TM13	TM12	TM11	TM10	TM9	TM8	TM7	TM6	TM5	TM4	TM3	TM2	TM1	TM0
	4	TM15	TM14	TM13	TM12	TM11	TM10	TM9	TM8	TM7	TM6	TM5	TM4	TM3	TM2	TM1	TM0
STMCR 1/0	0/1	TC15	TC14	TC13	TC12	TC11	TC10	TC9	TC8	TC7	TC6	TC5	TC4	TC3	TC2	TC1	TC0
	2	TC15	TC14	TC13	TC12	TC11	TC10	TC9	TC8	TC7	TC6	TC5	TC4	TC3	TC2	TC1	TC0
	3	TC15	TC14	TC13	TC12	TC11	TC10	TC9	TC8	TC7	TC6	TC5	TC4	TC3	TC2	TC1	TC0
	4	TC15	TC14	TC13	TC12	TC11	TC10	TC9	TC8	TC7	TC6	TC5	TC4	TC3	TC2	TC1	TC0
-	0/1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SCSCR 1/0	2	SST1	SST0	SED1	SED0	SCD1	SCD0	SCAM	CDIV2	CDIV1	CDIV0	CSLVL	CSEN3	CSEN2	CSEN1	CSEN0	CSOE
SFUR 1/0	3	-	TU14	TU13	TU12	TU11	TU10	TU9	TU8	TU7	TU6	TU5	TU4	TU3	TU2	TU1	TU0
-	4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	0/1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SCSTR 1/0	2	CSSU7	CSSU6	CSSU5	CSSU4	CSSU3	CSSU2	CSSU1	CSSU0	CSHD 7	CSHD 6	CSHD 5	CSHD 4	CSHD 3	CSHD 2	CSHD 1	CSHD 0
SFLR 1/0	3	-	TL14	TL13	TL12	TL11	TL10	TL9	TL8	TL7	TL6	TL5	TL4	TL3	TL2	TL1	TL0
-	4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Registers	Mode	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
-	0/1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SCSTR 2	2	CSDS 15	CSDS 14	CSDS 13	CSDS 12	CSDS 11	CSDS 10	CSDS9	CSDS8	CSDS7	CSDS6	CSDS5	CSDS4	CSDS3	CSDS2	CSDS1	CSDS 0
LAMSR/ LAMCR	3	LER	SER	RDRF	TDRE	TBI	LCSC	-	LAHC	LDL3	LDL2	LDL1	LDL0	LTDR CL	LCS TYP	LIDEN	LAME N
EIBCR/ NFCR	4	-	-	SDAS	SCLS	SDAC	SCLC	SOCE	BEC	-	-	-	NFT4	NFT3	NFT2	NFT1	NFT0
-	0/1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SCSFR 0/1	2	CS2 CSLVL	CS2 SCINV	CS2 SPI	CS2 BDS	CS2 L3	CS2 L2	CS2 L1	CS2 L0	CS1 CSLVL	CS1 SCINV	CS1 SPI	CS1 BDS	CS1 L3	CS1 L2	CS1 L1	CS1 L0
-	3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	0/1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SCSFR 2	2	-	-	-	-	-	-	-	-	CS3 CSLVL	CS3 SCINV	CS3 SPI	CS3 BDS	CS3 L3	CS3 L2	CS3 L1	CS3 L0
-	3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
TBYTE0	0/1	-	-	-	-	-	-	-	-	TD7	TD6	TD5	TD4	TD3	TD2	TD1	TD0
TBYTE 1/0	2	CS1 TD7	CS1 TD6	CS1 TD5	CS1 TD4	CS1 TD3	CS1 TD2	CS1 TD1	CS1 TD0	CS0 TD7	CS0 TD6	CS0 TD5	CS0 TD4	CS0 TD3	CS0 TD2	CS0 TD1	CS0 TD0
LAMIER/ LAMTID (LAMRID)	3	-	LCSER IE	LPTER IE	LSFER IE	LBSER IE	LCSC IE	-	LAHC IE	P1	P0	LID5	LID4	LID3	LID2	LID1	LID0
-	4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	0/1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
TBYTE 3/2	2	CS3 TD7	CS3 TD6	CS3 TD5	CS3 TD4	CS3 TD3	CS3 TD2	CS3 TD1	CS3 TD0	CS2 TD7	CS2 TD6	CS2 TD5	CS2 TD4	CS2 TD3	CS2 TD2	CS2 TD1	CS2 TD0
LAMESR/ LAMERT	3	-	LCSER	LPTER	LSFER	LBSER	-	-	-	KEY1	KEY0	-	LCSER T	LPTER T	LSFER T	LBSER T	FRE T
-	4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
BGR 1/0	0/1	EXT	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
	2	-	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
	3	EXT	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
	4	-	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
-	0/1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ISMK /ISBA	4	EN	SM6	SM5	SM4	SM3	SM2	SM1	SM0	SAEN	SA6	SA5	SA4	SA3	SA2	SA1	SA0
FCR1/0	0 to 4	-	-	-	FLSTE	FRIIE	FDRQ	FTIE	FSEL	-	FLST	FLD	FSET	FCL2	FCL1	FE2	FE1
FBYTE 2/1	0 to 4	FD15	FD14	FD13	FD12	FD11	FD10	FD9	FD8	FD7	FD6	FD5	FD4	FD3	FD2	FD1	FD0
FTICR 2/1	0 to 4	TXFTL 15	TXFTL 14	TXFTL 13	TXFTL 12	TXFTL 11	TXFTL 10	TXFTL 9	TXFTL 8	TXFTL 7	TXFTL 6	TXFTL 5	TXFTL 4	TXFTL 3	TXFTL 2	TXFTL 1	TXFTL 0
ESR/ECR	0 to 4	-	-	-	-	RXUD R	TXOV R	RBER R	TBER R	-	-	-	EISEL	REIE	TEIE	RXBLK EN	TXBLK EN
TBSIZE	0 to 4	-	-	-	-	-	-	-	-	TBSIZ E7	TBSIZ E6	TBSIZ E5	TBSIZ E4	TBSIZ E3	TBSIZ E2	TBSIZ E1	TBSIZ E0

Registers	Mode	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SCRC/ SMRC	3	-	MSC	-	RIEC	TIEC	TBIEC	RXEC	TXEC	-	-	-	WUCRC	SBLC	-	-	SOEC
IBCRC/ SMRC	4	MSSC	-	ACKEC	WSELC	CNDEC	INTEC	-	INTC	-	-	-	WUCRC	RIEC	TIEC	-	-
SSRC/ ESCRC	3	-	-	LBDC	-	-	-	-	-	-	ESBLC	-	LBIEC	-	-	-	-
SSRC/ IBSRC	4	-	-	DMAC	TBIEC	-	-	-	-	-	-	-	-	-	RSCC	SPCC	-
SACSR1C / SACSR0C	0/1	STSTC	-	-	-	-	-	-	TINTC	TINTEC	TSYNEC	TGREC	-	-	-	-	TMREC
	2	STSTC	-	TBEENC	CSIEC	CSEC	-	-	TINTC	TINTEC	TSYNEC	TGREC	-	-	-	-	TMREC
	3	STSTC	-	SFDC	SFDEC	AUTEC	-	-	TINTC	TINTEC	-	TGREC	-	-	-	-	TMREC
	4	-	-	-	-	-	-	-	TINTC	TINTEC	-	TGREC	-	-	-	-	TMREC
LAMSRC/ LAMCRC	3	-	-	-	-	-	LCSCC	-	LAHCC	-	-	-	-	-	LCSTYPC	LIDENC	LAMENC
LAMIERC/ -	3	-	LCSERIEC	LPTERIEC	LSFERIEC	LBSERIEC	LCSCIEC	-	LAHCIEC	-	-	-	-	-	-	-	-
LAMESRC / -	3	-	LCSERC	LPTERC	LSFERC	LBSERC	-	-	-	-	-	-	-	-	-	-	-
FCR1C/ FCR0C	0 to 4	-	-	-	FLSTEC	FRIIEC	FDRQC	FTIEC	FSELC	-	-	-	-	-	-	FE2C	FE1C
ESRC/-	0 to 4	-	-	-	-	RXUDRC	TXOVRC	RBERRC	TBERRC	-	-	-	-	-	-	-	-

Registers	Mode	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SCRS/ SMRS	3	UPCLS	MSS	LBRS	RIES	TIES	TBIES	RXES	TXES	-	-	-	WUCRS	SBLS	-	-	SOES
IBCRS/ SMRS	4	MSSS	ACTS	ACKES	WSELS	CNDES	INTES	-	INTS	-	-	-	WUCRS	RIES	TIES	-	-
SSRS/ ESCRS	3	RECS	-	-	-	-	-	-	-	-	ESBLS	-	LBIES	-	-	-	-
SSRS/ -	4	RECS	TSETS	DMAS	TBIES	-	-	-	-	-	-	-	-	-	-	-	-
SACSR1S/ SACSR0S	0/1	STSTS	-	-	-	-	-	-	-	TINTES	TSYNES	TGRES	-	-	-	-	TMRES
	2	STSTS	-	TBEENS	CSEIES	-	-	-	-	TINTES	TSYNES	TGRES	-	-	-	-	TMRES
	3	STSTS	-	-	SFDES	AUTES	-	-	-	TINTES	-	TGRES	-	-	-	-	TMRES
	4	-	-	-	-	-	-	-	-	TINTES	-	TGRES	-	-	-	-	TMRES
-/ LAMCRS	3	-	-	-	-	-	-	-	-	-	-	-	-	LTDRCLS	LCSTYPS	LIDENS	LAMENS
LAMIERS/ -	3	-	LCSERIES	LPTERIES	LSFERIES	LBSERIES	LCSCIES	-	LAHCIES	-	-	-	-	-	-	-	-
FCR1S/ FCR0S	0 to 4	-	-	-	FLSTES	FRIIES	-	FTIES	FSELS	-	-	FLDS	FSETS	FCL2S	FCL1S	FE2S	FE1S

CHAPTER 36: UART (Asynchronous Serial Interface)

This chapter describes the UART (Asynchronous Serial Interface) function supported by operation modes 0 and 1 of the multi-function serial interface.

1. Overview of the UART (Asynchronous Serial Interface)
2. Interrupts from the UART
3. UART Operation
4. Serial Timer Operation
5. Test Mode
6. Dedicated Baud Rate Generator
7. Block Transfer
8. Setup Procedure and Program Flow for Operation Mode 0 (Asynchronous Normal Mode)
9. Setup Procedure and Program Flow for Operation Mode 1 (Asynchronous Multi-processor Mode)
10. Registers of UART (Asynchronous Serial Interface)

UART-TXXPT03P01R01L11-E1-XX

1. Overview of the UART (Asynchronous Serial Interface)

UART (Asynchronous Serial Interface) is a general-purpose serial data communication interface for asynchronous communication with external devices. This interface supports bidirectional communication (normal mode) and master/slave-type communication (multi-processor mode: both master and slave roles are supported). In addition, this interface has FIFOs for transmission and reception.

UART (Asynchronous Serial Interface) Functions

	Function	Description
1	Data	– Full-duplex double buffer (when FIFO is not used) – Transmission and reception FIFOs (64 bytes for each) (when FIFO is used)
2	Serial input	– Oversampling is performed by the bus clock 3 times. The received value is determined by the rule of majority.
3	Transfer format	– Asynchronous
4	Baud rate	– Dedicated baud rate generator (15-bit reload counter configuration) – The external clock input can be adjusted by the reload counter.
5	Data length	– 5 to 9 bits (for normal mode), or 7 to 8 bits (multi-processor mode)
6	Signaling method	– NRZ (Non Return to Zero) and inverted NRZ
7	Start bit detection	– Synchronized to falling edges of the start bit (for NRZ method) – Synchronized to rising edges of the start bit (for inverted NRZ method)
8	Reception error detection	– Framing error – Overrun error – Parity error *1
9	Hardware flow control	– Transmission/reception auto control by CTS and RTS
10	Synchronous transmission function	– Can automatically transmit data periodically in synchronization with the serial timer. – Can transmit in synchronization with an external trigger.
11	Timer function	– A 16-bit serial timer is available. – A division value can be selected for the operation clock (1 to 256 divisions). – Activation by an external trigger is available.
12	Interrupt request	– Reception interrupt (reception completion, framing error, overrun error, parity error *1, reception block transfer error) – Reception FIFO interrupt (reception FIFO under run) – Transmission interrupt (transmission data empty, transmission bus idle, transmission block transfer error) – Transmission FIFO interrupt (when the transmission FIFO is not higher than the interrupt trigger level or when the transmission FIFO is empty, transmission FIFO overrun) – DMA Transfer is supported for both transmission and reception. – Status interrupt (serial timer interrupt)
13	Master/slave-type communication function (Multi-processor mode)	– 1 (master) -to-n (slave) communication is possible. (Both master and slave systems are supported.)
14	FIFO option	– Transmission and reception FIFOs are available (transmission FIFO: 64 bytes, reception FIFO: 64 bytes). – Transmission FIFO and reception FIFO can be selected. – Transmission data can be retransmitted. – The timing of the reception FIFO interrupt can be changed by software. – Independent FIFO reset is supported.

*1: Parity errors occur only in normal mode.

Interrupt request and DMA request

- Interrupt request of this module can be used for purposes below:
  1. Interrupt request to CPU (via Interrupt Controller)
  2. DMA request to DMA Controller
- To use the interrupt as interrupt request to CPU, following configuration is necessary:
  - Enable the interrupt request in this module
  - Enable the interrupt channel of the interrupt in Interrupt Controller
- To use the interrupt as DMA request to DMA Controller, following configuration is necessary:
  - Enable the interrupt request in this module
  - Select the DMA channel from this module as DMA client in DMA Controller

Note:

- *Interrupt request that supports for DMA request is product specification. Please refer to the product's HWM for detail.*

2. Interrupts from the UART

UART has transmission/reception interrupts and status interrupts. Interrupt requests can be generated with the factors described below.

- When reception data is set in the reception data register (RDR), or when a reception error occurs
- When transmission data is transferred from the transmission data register (TDR) to the transmission shift register and transmission starts
- Transmission bus idle (no transmission)
- Transmission FIFO data request
- When the serial timer comparison value (STMCR) and the serial timer value (STMR) coincide

Interrupts from the UART

Table 2-1 lists the interrupt control bits and interrupt factors of UART.

Table 2-1 UART Interrupt Control Bits and Interrupt Factors

Interrupt Types	Interrupt Request Flag Bit	Flag Register	Operation Mode		Interrupt Factor	Interrupt Factor Enable Bit	Interrupt Request Flag Clearing
			0	1			
Reception	RDRF	SSR	Appl	Appl	1-byte reception	SCR:RIE	- Reading of reception data (RDR) - Software reset (SCR:UPCL=1)
					Reception of as much data as the amount set in FBYTE		- Reading of reception data (RDR) until the reception FIFO is empty - Software reset (SCR:UPCL=1)
					Detection of reception idle for the 8-bit time or longer while the FRIIE bit is "1" and the reception FIFO contains valid data		
	ORE	SSR	Appl	Appl	Overrun error	SCR:RIE ECR:REIE	- Writing "1" to the reception error flag clear bit (SSR:REC) - Software reset (SCR:UPCL=1)
	FRE	SSR	Appl	Appl	Framing error		
	PE	SSR	Appl	NA	Parity error		
	RXUDR	ESR	Appl	Appl	Reception FIFO under run	ECR:REIE	- Writing "1" to the reception FIFO under run flag clear bit (ESRC:RXUDRC) - Software reset (SCR:UPCL=1) - Disable reception FIFO (1) When FCR1:FSEL=0, set FCR0.FE2=0 (2) When FCR1:FSEL=1, set FCR0.FE1=0
	RBERR	ESR	Appl	Appl	Reception block transfer error	ECR:REIE	- Writing "1" to the reception block transfer error clear bit (ESRC:RBERRC) - Software reset(SCR:UPCL=1) - Disable reception block transfer mode (ECR:RXBLKEN=0)

Interrupt Types	Interrupt Request Flag Bit	Flag Register	Operation Mode		Interrupt Factor	Interrupt Factor Enable Bit	Interrupt Request Flag Clearing
			0	1			
Transmission	TDRE	SSR	Appl	Appl	The transmission register is empty.	SCR:TIE	Writing to the transmission data register (TDR), or writing "1" to the transmission FIFO operation enable bit (retransmission) when the bit value is "0" and the transmission FIFO contains valid data*1
	TBI	SSR	Appl	Appl	No transmission	SCR:TBIE	Writing to the transmission data register (TDR), or writing "1" to the transmission FIFO operation enable bit (retransmission) when the bit value is "0" and the transmission FIFO contains valid data*1
	FDRQ	FCR1	Appl	Appl	The transmission FIFO stores a data amount equal to or smaller than the FTICR setting value or is empty.	FCR1:FTIE	Writing "1" to the FIFO transmission data request clear bit (FCR1C:FDRQC), or the transmission FIFO is full.
	TXOVR	ESR	Appl	Appl	Transmission FIFO overrun	ECR:TEIE	- Writing "1" to the transmission FIFO overrun flag clear bit (ESRC:TXOVRC) - Software reset(SCR:UPCL=1) - Disable transmission FIFO (1) When FCR1:FSEL=0, set FCR0:FE1=0 (2) When FCR1.FSEL=1, set FCR0:FE2=0
	TBERR	ESR	Appl	Appl	Transmission block transfer error	ECR:TEIE	- Writing "1" to the transmission block transfer error clear bit (ESRC:TBERRC) - Software reset(SCR:UPCL=1) - Disable transmission block transfer (ECR.TXBLKEN=0)
Status	TINT	SACSR	Appl	Appl	Coincidence of the serial timer register (STMR) and the serial timer comparison register (STMCR) values	SACSR: TINTE	- Writing "1" to the timer interrupt flag clear bit (SACSRC:TINTC) - Software reset(SCR:UPCL=1)

*1: Set the TIE bit to "1" after the TDRE bit becomes "0".

Error Interrupt Request Output switching

Depending on the settings of the error interrupt request output switch bit (ECR:EISEL), transmission and reception error interrupt enable bits (ECR:REIE, TEIE), and transmission and reception interrupt enable bits (SCR:RIE, TIE, TBIE, FCR1:FTIE), the error interrupt request output method differs.

Table 2-2 Transmission Interrupt Request Output Method

ECR:EISEL	Interrupt Output	Target Interrupt Enable Bit	Interrupt Factor
0	Transmission interrupt request output	SCR:TIE	No transmission data (SSR:TDRE)
		SCR:TBIE	No transmission operation (SSR:TBI)
		FCR1:FTIE	Transmission FIFO empty (FCR1:FDRQ)
		ECR:TEIE	Transmission FIFO overrun (ESR:TXOVR) Transmission block transfer error (ESR:TBERR)
	Transmission error interrupt request output	-	No interrupt factor
1	Transmission interrupt request output	SCR:TIE	No transmission data (SSR:TDRE)
		SCR:TBIE	No transmission operation (SSR:TBI)
		FCR1:FTIE	Transmission FIFO empty (FCR1:FDRQ)
	Transmission error interrupt request output	ECR:TEIE	Transmission FIFO overrun (ESR:TXOVR) Transmission block transfer error (ESR:TBERR)

Table 2-3 Reception Interrupt Request Output Method

ECR:EISEL	Interrupt Output	Target Interrupt Enable Bit	Interrupt Factor
0	Reception interrupt request output	SCR:RIE	Reception full (SSR:RDRF) Framing error (SSR:FRE) Overrun error (SSR:ORE) Parity error (SSR:PE)
		ECR:REIE	Reception FIFO under run (ESR:RXUDR) Reception block transfer error (ESR:RBERR)
	Reception error interrupt request output	-	No interrupt factor
1	Reception interrupt request output	SCR:RIE	Reception full (SSR:RDRF)
	Reception error interrupt request output	ECR:REIE	Framing error (SSR:FRE) Overrun error (SSR:ORE) Parity error (SSR:PE) Reception FIFO under run (ESR:RXUDR) Reception block transfer error (ESR:RBERR)

2.1. Occurrence of Reception Interrupts and Flag Set Timing

There are 2 reception interrupt types: Reception completion (SSR:RDRF) and reception error (SSR:PE, ORE, FRE).

Occurrence of Reception Interrupts and Flag Set Timing

Upon the detection of the first stop bit, the reception data is stored in the reception data register (RDR). Related flags are set upon the completion of reception (SSR:RDRF = 1) or upon a reception error (SSR:PE, ORE, FRE = 1). At that time, a reception interrupt occurs if the reception interrupt is enabled (SCR:RIE = 1).

Note:

- *If a reception error occurs, the data in the reception data register (RDR) will be invalid.*

Figure 2-1 Set Timing of RDRF (Reception Data Full) Flag Bit

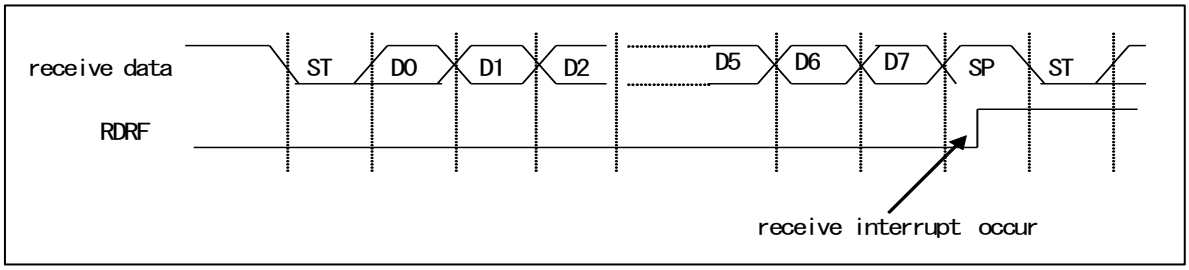
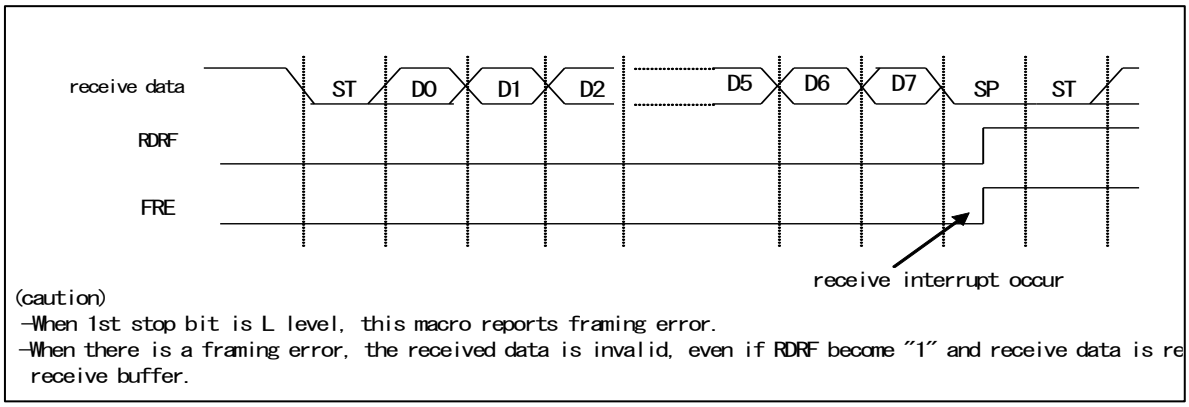


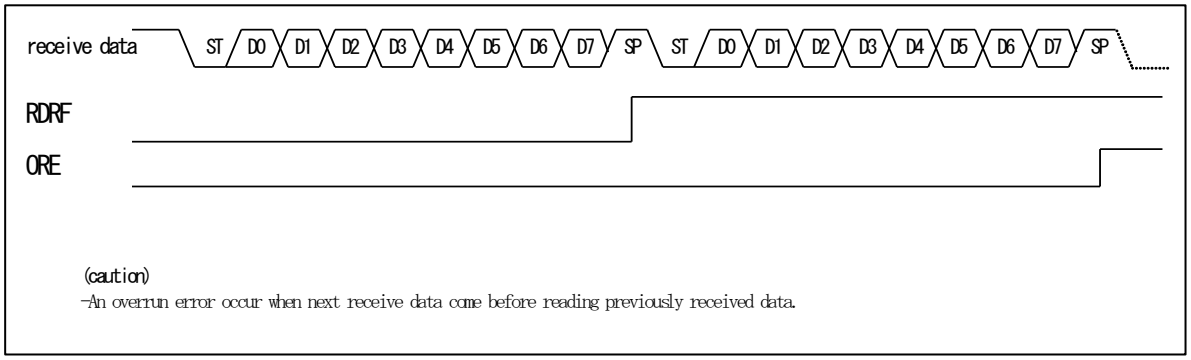
Figure 2-2 Set Timing of FRE (Framing Error) Flag Bit



Note:

- *During data reception, the following will happen if one of the following edges is detected at the same time as, or 1 or 2 bus clocks earlier than, the stop bit sampling point: The edge becomes invalid and the data that follows may not be received normally. Consecutive frames must have intervals between them.*
  - Falling edge of serial data (when ESCR:INV = 0)
  - Rising edge of serial data (when ESCR:INV = 1)

Figure 2-3 Set Timing of ORE (Overrun Error) Flag Bit



2.2. Occurrence of Interrupts and Flag Set Timing When the Reception FIFO Is Used

When the reception FIFO is used, an interrupt occurs if an amount of data equal to the amount set in the FBYTE register (FBYTE) is received.

Occurrence of Reception Interrupts and Flag Set Timing When the Reception FIFO is Used

When the reception FIFO is used, occurrence of interrupts is determined by the value set in the FBYTE register.

- If as much transfer data as the amount set in the FBYTE register is received, the reception data full flag (SSR:RDRF) in the serial status register is set to "1". At this time, a reception interrupt occurs if the reception interrupt is enabled (SCR:RIE).
- When both of the following conditions are satisfied, the continuation of reception idle status for 8 baud rate clocks or longer sets the interrupt flag (SSR:RDRF) to "1".
  - The reception FIFO idle detection enable bit (FCR1:FRIIE) is "1".
  - The number of data items in the reception FIFO does not reach the transfer count.

During an 8-clock count, the counter is reset to 0 when RDR is read, then the system starts counting the 8 clocks again. The counter is reset to 0 when the reception FIFO is disabled. If the reception FIFO is enabled when there is data remaining in the reception FIFO, counting restarts.

- When the reception FIFO becomes empty as a result of reading the reception data (RDR), the reception data full flag (SSR:RDRF) is cleared.
- If the reception valid data count becomes equal to the value of the FIFO capacity, reception of any subsequent data triggers an overrun error (SSR:ORE = 1).

Figure 2-4 Occurrence Timing of Reception Interrupt When Reception FIFO Is Used

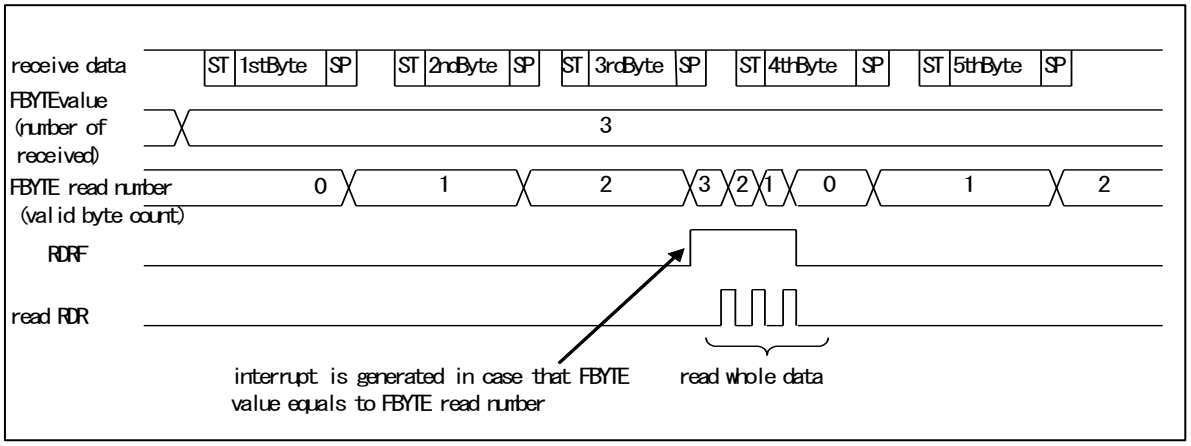
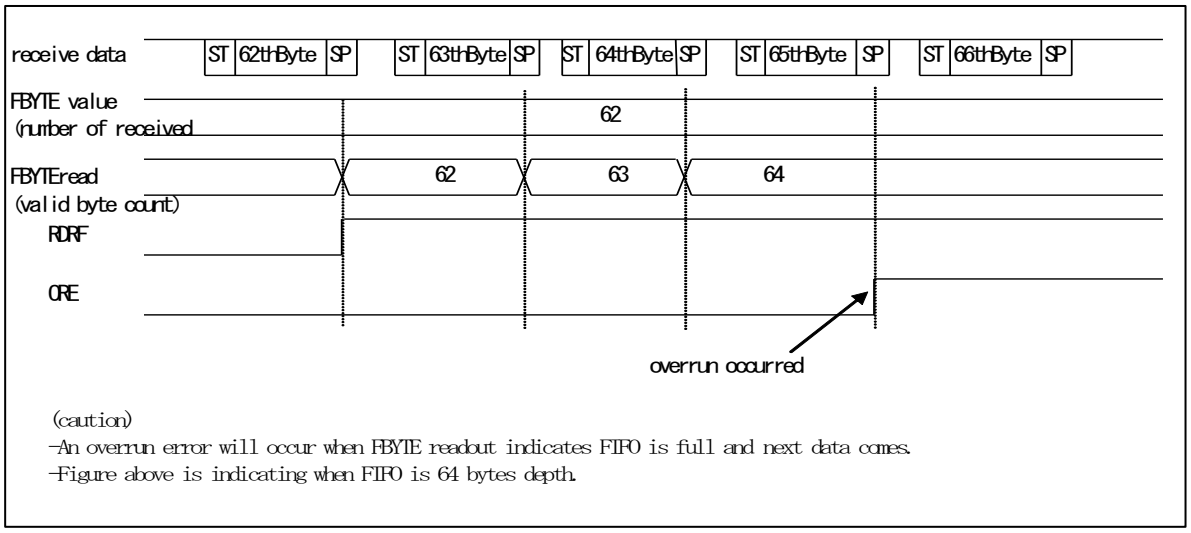


Figure 2-5 Set Timing of ORE (Overrun Error) Flag Bit



2.3. Occurrence of Transmission Interrupts and Flag Set Timing

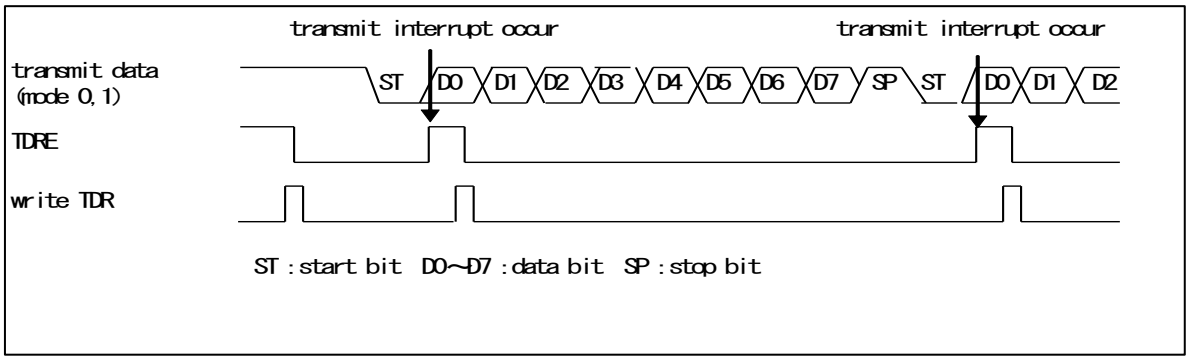
A data transmission interrupt occurs in the following cases: 1. The transmission data is transferred from the transmission data register (TDR) to the transmission shift register (SSR:TDRE = 1) to start data transmission. 2. The transmission operation is not performed (SSR:TBI = 1).

Occurrence of Transmission Interrupts and Flag Set Timing

Set Timing of Transmission Data Empty Flag (SSR:TDRE)

Transfer of data from the transmission data register (TDR) to the transmission shift register allows the next data to be written (SSR:TDRE = 1). At that time, a transmission interrupt occurs if the transmission interrupt is enabled (SCR:TIE = 1). The SSR:TDRE bit is a read-only bit, and it is cleared to "0" when data is written to the transmission data register (TDR).

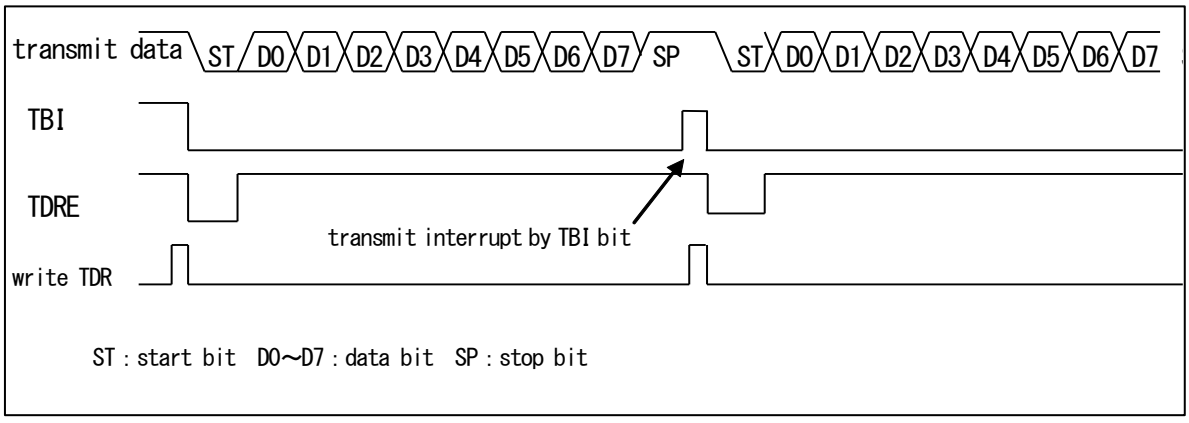
Figure 2-6 Set Timing of Transmission Data Empty Flag (SSR:TDRE)



Set Timing of Transmission Bus Idle Flag (SSR:TBI)

The SSR:TBI bit is set to "1" when the transmission data register is empty (SSR:TDRE = 1) and transmission is not performed. At that time, a transmission interrupt occurs if the transmission bus idle interrupt is enabled (SCR:TBIE = 1). Writing transmission data to the transmission data register (TDR) clears the SSR:TBI bit and the transmission interrupt request.

Figure 2-7 Set Timing of Transmission Bus Idle Flag (TBI) (Synchronous Transmission Disabled SACSR:TSYNE = 0)



Note:

- *In the case of synchronous transmission (SACSR:TSYNE = 1), the SSR:TBI bit is set to "1" if the transmission data register is empty (SSR:TDRE = 0) after transmitting data for the byte count set in the transfer byte count (TBYTE0).*

2.4. Occurrence of Interrupts and Flag Set Timing When the Transmission FIFO Is Used

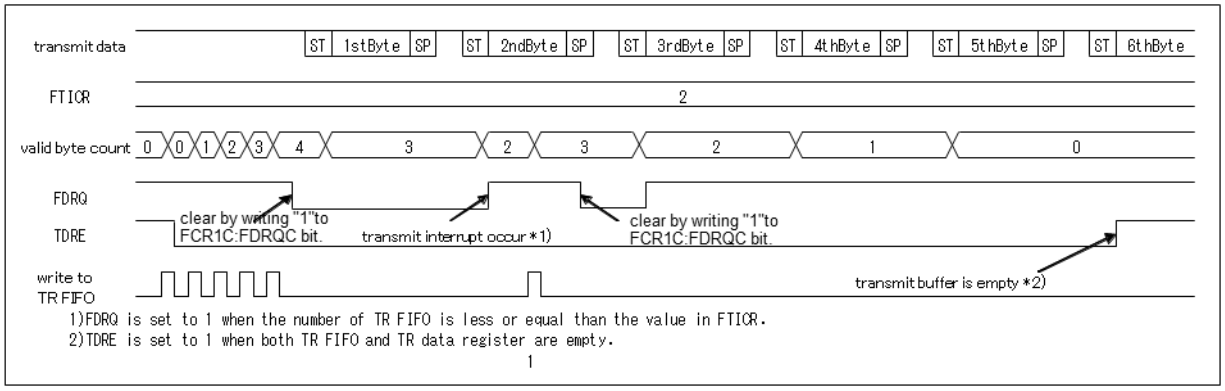
When the transmission FIFO is used, an interrupt occurs if the number of data items stored in the transmission FIFO is equal to or less than the setting of the FTICR register (FTICR).

Occurrence of Transmission Interrupts and Flag Set Timing When the Transmission FIFO is Used

When the transmission FIFO is used, occurrence of interrupts is determined by the value set in the FTICR register.

- When the amount of data stored in the transmission FIFO is equal to or less than the value set for the FTICR register, the FIFO transmission data request bit (FCR1:FDRQ) is set to "1".
  At that time, a transmission interrupt occurs if the FIFO transmission interrupt is enabled (FCR1:FTIE = 1).
- When you write the necessary data to the transmission FIFO after a transmission interrupt occurs, write "1" to the FIFO transmission data request clear bit (FCR1C:FDRQC) to clear the interrupt request.
- The FIFO transmission data request bit (FCR1:FDRQ) is set to "0" once the transmission FIFO is full.
- The existence of data in the transmission FIFO can be verified by reading the FIFO byte register (FBYTE) or the transmission FIFO interrupt control register (FTICR).
  FBYTE = 0x00 and FTICR = 0x00 indicate that the transmission FIFO does not contain data.

Figure 2-8 Occurrence Timing of Transmission Interrupt When Transmission FIFO is Used



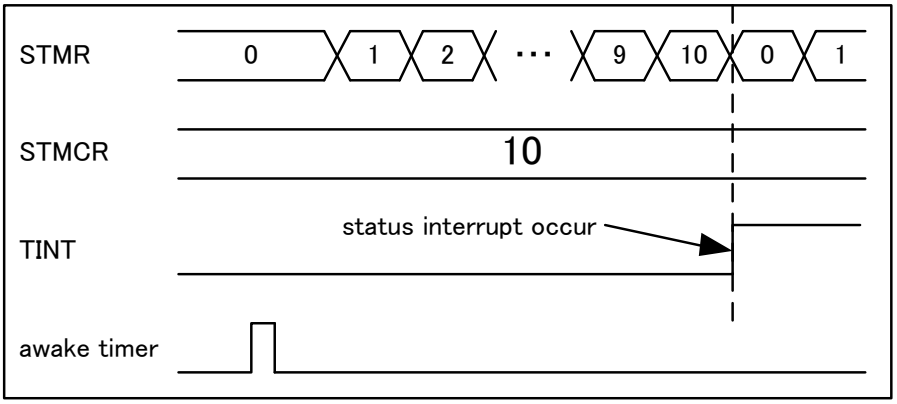
2.5. Occurrence of Timer Interrupts and Flag Set Timing

A timer interrupt occurs when the serial timer register (STMR) coincides with the serial timer comparison register (STMCR).

Timer Interrupt Generation and Flag Set Timing

The timer interrupt flag (SACSR:TINT) is set to "1" when the serial timer register (STMR) coincides with the serial timer comparison register.
At that time, a status interrupt occurs if the timer interrupt is enabled (SACSR:TINTE = 1).

Figure 2-9 Occurrence Timing of Timer Interrupt



3. UART Operation

UART operates in mode 0, which corresponds to bidirectional serial asynchronous communication, and also in mode 1, which corresponds to master/slave multi-processor communication.

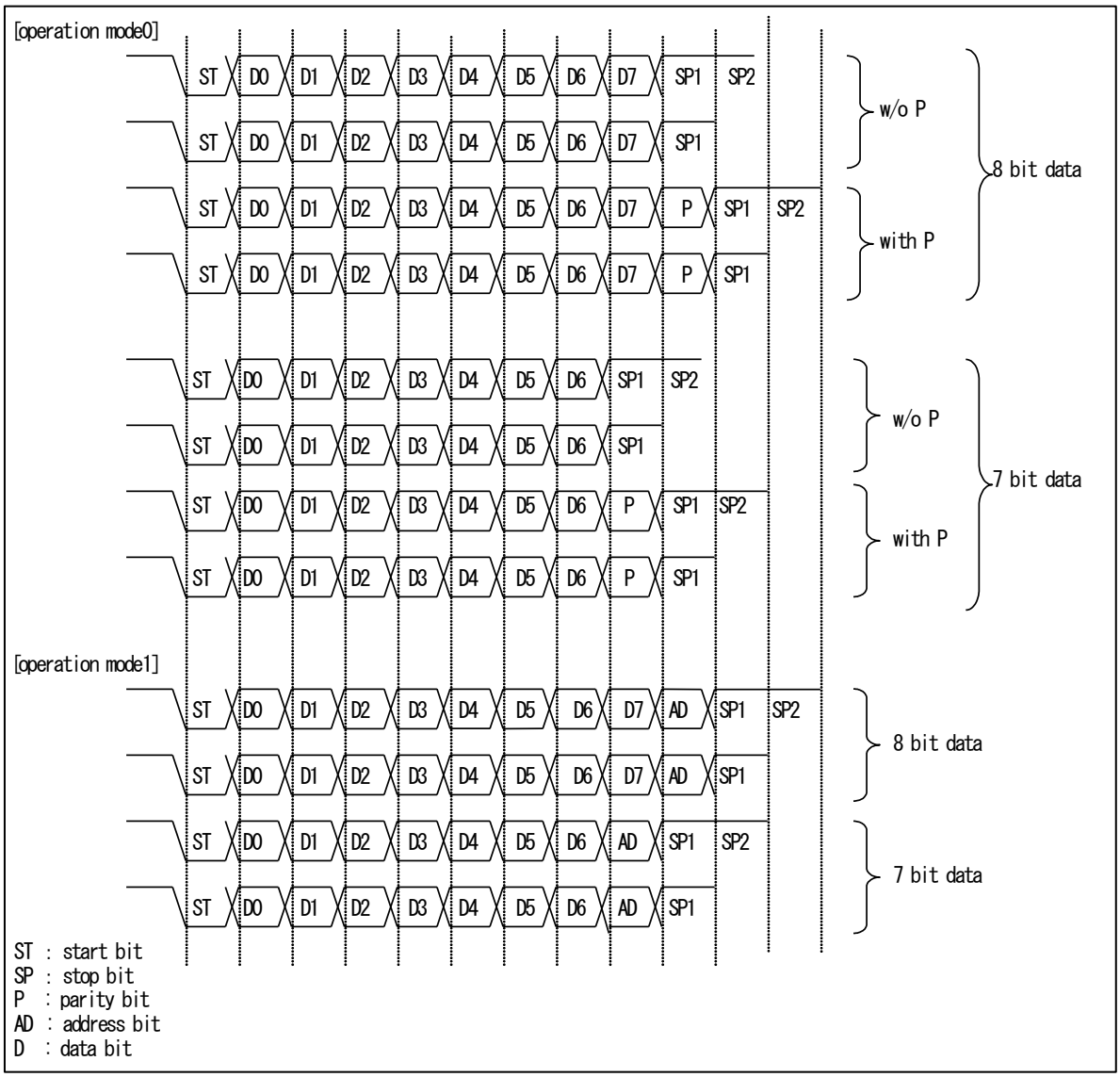
UART Operation

Transmission and Reception Data Format

- Transmission or reception data always starts with a start bit followed by transmission/reception of a specified data bit length and ending with at least one stop bit.
- The data transfer direction (LSB first or MSB first) is determined by the BDS bit in the serial mode register (SMR). If parity check is enabled, the parity bit is always placed between the last data bit and the first stop bit.
- Parity check can be enabled or disabled for operation mode 0 (normal mode).
- In operation mode 1 (multi-processor mode), AD bits are appended instead of parity bits.

Figure 3-1 shows the transmission and reception data formats of operation modes 0 and 1.

Figure 3-1 Example of Transmission and Reception Data Formats (Operation Modes 0 and 1)



Notes:

- *The above figure show cases for data lengths of 7 bits and 8 bits. (Data length can be set to 5 bits to 9 bits for operation mode 0.)*
- *When the BDS bit in the serial mode register (SMR) is set to "1" (MSB first), bits are processed in the order of D7, D6, D5, ..., D1, D0 (P).*
- *If the data length is set to X bits, the lowest X bits of the transmission and reception data register (RDR/TDR) are valid.*

Transmission Operation

- When the transmission data empty flag bit (TDRE) in the serial status register (SSR) is "1", transmission data can be written to the transmission data register (TDR). (When the transmission FIFO is enabled, transmission data can be written even if TDRE = 0.)
- Writing transmission data to the transmission data register (TDR) sets the transmission data empty flag bit (SSR:TDRE) to "0".
- Setting the transmission operation enable bit (SCR:TXE) in the serial control register to "1" loads the transmission data to the transmission shift register and starts transmission from the start bit.
- The transmission data empty flag bit (SSR:TDRE) is set to "1" again when transmission starts. At this time, a transmission interrupt occurs if transmission interrupts are enabled (SCR:TIE = 1). The interrupt processing can write the next transmission data to the transmission data register.

Notes:

- *Because the initial value of the transmission data empty flag bit (SSR:TDRE) is "1", a transmission interrupt occurs as soon as the transmission interrupt is enabled (SCR:TIE).*
- *Because the initial value of the FIFO transmission data request bit (FCR1:FDRQ) is "1", a transmission interrupt occurs as soon as the FIFO transmission interrupt is enabled (FCR1:FTIE = 1).*

Reception Operation

- When the reception operation is enabled (SCR:RXE = 1), the reception operation starts.
- After a start bit is detected, 1-frame data is received according to the data format set in the extended communication control register (ESCR:PEN, P, L2, L1, and L0) and the serial mode register (SMR:BDS). A start bit is detected when both of the following are satisfied: 1. The noise-filtered result (the result of applying the rule of majority to the 3-time bus clock sampling of the serial data input) shows falling (when ESCR:INV = 0) or rising (when ESCR:INV = 1). 2. The filtered data shows "L" at the sampling point.
- The reception data full flag bit (SSR:RDRF) is set to "1" when 1-frame data has been received. At that time, a reception interrupt occurs if the reception interrupt is enabled (SCR:RIE = 1).
- When reading the reception data, read the reception data after all of the 1-frame data has been received, and check the error flag in the serial status register (SSR). If a reception error occurs, perform error processing.
- Reading the reception data clears the reception data full flag bit (SSR:RDRF) to "0".
- If the reception FIFO is enabled, the reception data full flag bit (SSR:RDRF) is set to "1" upon the reception of as many frames of data as the value set in the reception FBYTE.
- When both of the following conditions are satisfied, continuation of reception idle status for 8 baud rate clocks or longer sets the interrupt flag (RDRF) to "1".
  - The reception FIFO idle detection enable bit (FRIIE) is "1".
  - The number of data items in the reception FIFO does not reach the transfer count.

  During an 8-clock count, the counter is reset to 0 when RDR is read, and the system starts counting the 8 clocks again. The counter is reset to 0 when the reception FIFO is disabled. If the reception FIFO is enabled while data remains in the reception FIFO, counting restarts.
- When the error flag in the serial status register (SSR) is set to "1" while the reception FIFO is enabled, the data related to the error is not stored in the reception FIFO. At that time, the reception data full flag bit (SSR:RDRF) is not set to "1". (However, an overrun error sets the RDRF flag to "1".) The reception FBYTE indicates the data count that had been received normally before the error occurred. The reception FIFO is not enabled unless the error flag in the serial status register (SSR) is cleared to "0".
- If the reception FIFO is enabled, the reception data full flag bit (SSR:RDRF) is cleared to "0" when the reception FIFO becomes empty.

Notes:

- *The reception data register (RDR) is effective when the reception data register full flag bit (SSR:RDRF) is set to "1" and there has been no reception error (SSR:PE, ORE, FRE = 0).*
- *There is a built-in noise filter (where the rule of majority is applied to the 3-time bus clock sampling of the serial data input), but erroneous data is received if noise has passed through the filter. We suggest that you design the board in such a way that noise does not pass through the filter. Alternatively, we suggest that you implement a communication method whereby noise passing through the filter does not cause a problem (by, for example, appending a checksum to the data at the end of transmission for retransmission in the event of an error).*
- *During data reception, the following will happen if one of the following edges is detected at the same time as, or 1 or 2 bus clocks earlier than, the stop bit sampling point: The edge becomes invalid and the data that follows may not be received normally. Consecutive frames must have intervals between them.*
  - Falling edge of serial data (when ESCR:INV = 0)
  - Rising edge of serial data (when ESCR:INV = 1)

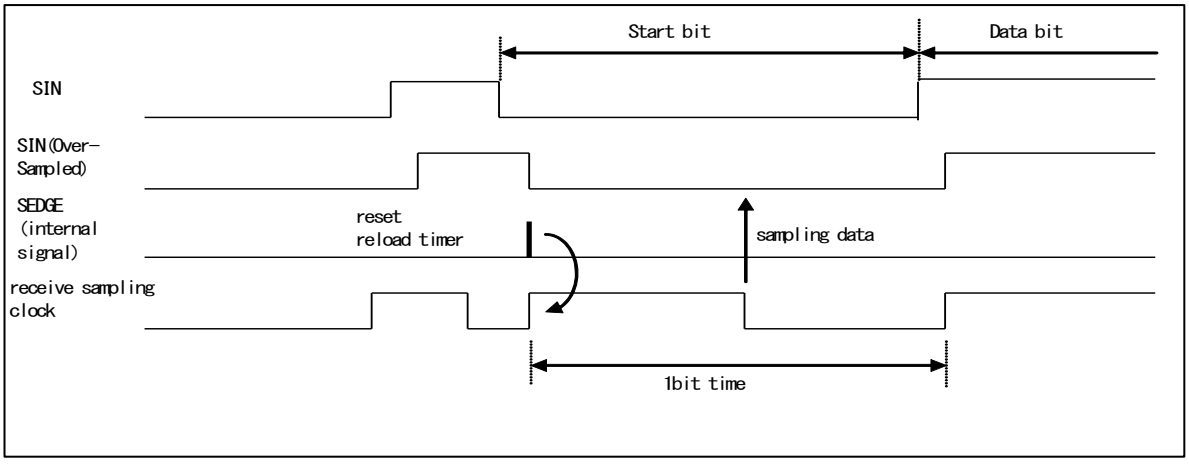
Clock Selection

- An internal clock or an external clock can be used.
- Set BGR1:EXT = 1 when using an external clock. In this case, the external clock is divided by a baud rate generator.

Start Bit Detection

- In the case of asynchronous mode, a start bit is identified by a falling edge of the SIN signal. Therefore, enabling reception operation (SCR:RXE = 1) does not start the reception operation unless there is a falling edge input of the SIN signal.
- When a falling edge of the start bit is detected, the reception reload counter of the baud rate generator is reset, reloaded again, and then starts counting down. This ensures that sampling is always performed in the center of the data.

Figure 3-2 Start Bit Detection



Stop Bit

- A bit length from 1 to 4 can be selected.
- The reception data full flag bit (SSR:RDRF) is set to "1" when the first stop bit is detected.

Error Detection

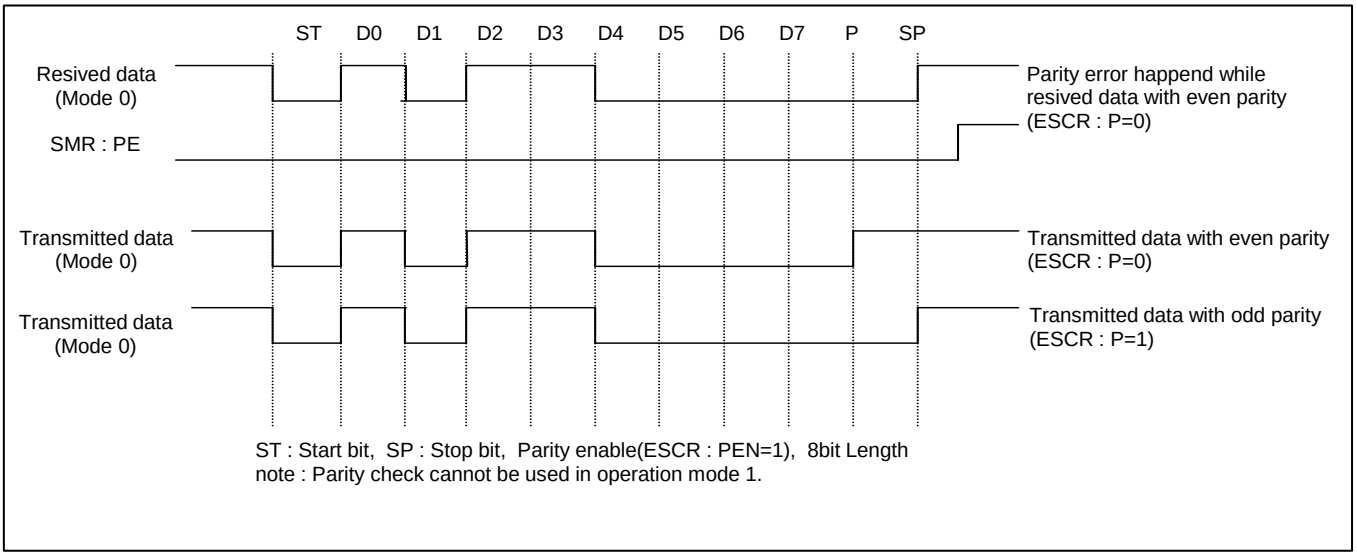
- Operation mode 0 allows the detection of parity errors, overrun errors, and framing errors.
- Operation mode 1 allows the detection of overrun errors and framing errors. The detection of parity errors is not possible.

Parity Bit

- A parity bit can be appended only in operation mode 0. The parity enable bit (ESCR:PEN) sets the presence or absence of parity check, while the parity selection bit (ESCR:P) sets even or odd parity.
- Parity check cannot be used in operation mode 1.

Figure 3-3 shows the transmission and reception data when parity check is enabled.

Figure 3-3 Operation with Parity Check Enabled

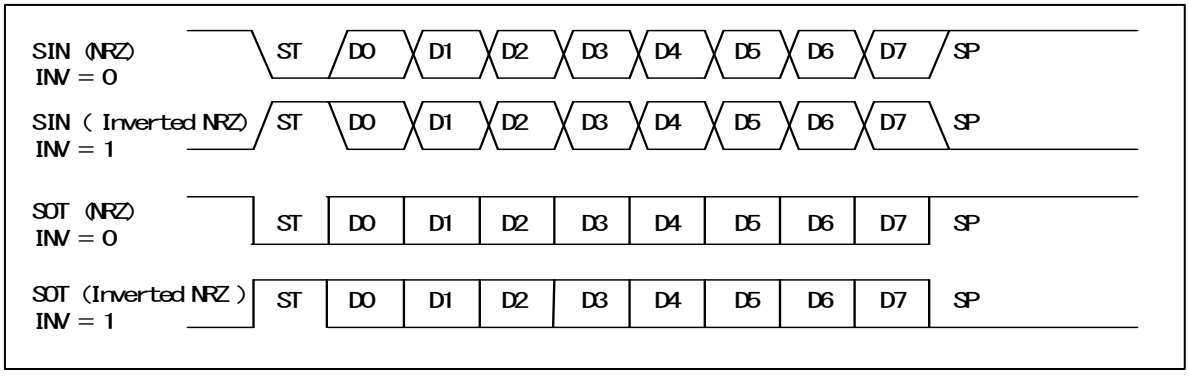


Data Signaling Method

- Setting of the INV bit in the extended communication control register selects the NRZ (Non Return to Zero) signaling method (ESCR:INV = 0) or the inverted NRZ signaling method (ESCR:INV = 1).

Figure 3-4 shows the NRZ signaling method and the inverted NRZ signaling method.

Figure 3-4 NRZ (Non Return to Zero) Signaling Method and Inverted NRZ Signaling Method



Data Transfer Method

- Either LSB first or MSB first can be selected as the data bit transfer method.

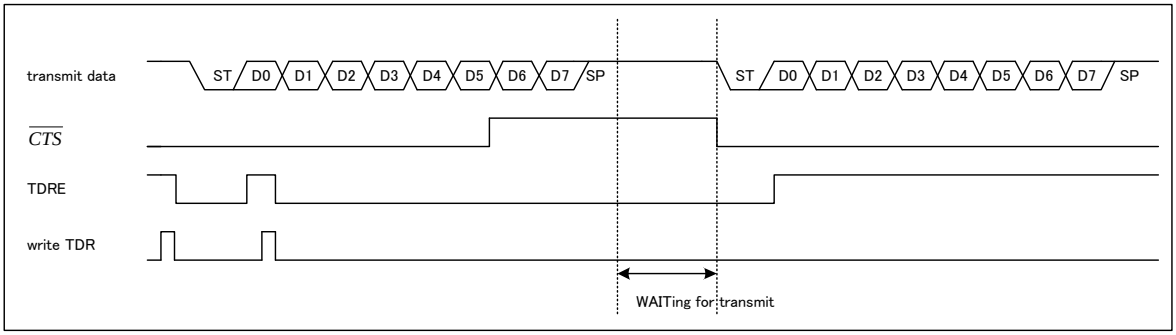
Hardware Flow Control

When flow control is enabled (ESCR:FLWEN = 1), UART performs hardware flow control.

- During data transmission

  If $\overline{CTS}$ is "H" after data transmission, the next data is not transmitted even if data exists in the transmission buffer (TDRE = 0). Transmission of the data is suspended until $\overline{CTS}$ becomes "L". To suspend transmission, set $\overline{CTS}$ to "H" before the completion of the stop bit transmission. However, to suspend transmission when synchronous transmission is enabled (SACSR:TSYNE = 1, TMRE = 1), set $\overline{CTS}$ to "H" before the start of transmission of the last stop bit. If $\overline{CTS}$ is set to "H" during transmission, the transmission continues until after the stop bit.

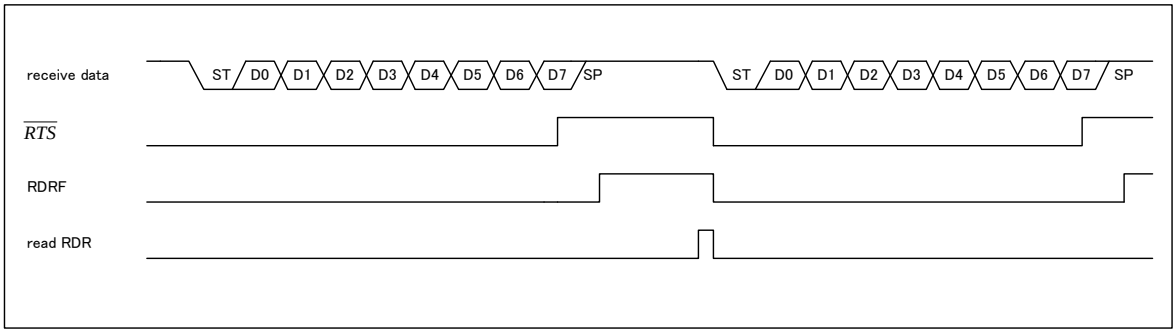
Figure 3-5 Hardware Flow Control for Data Transmission (SMR:SBL = 0, ESCR:ESBL = INV = PEN = L2 = L1 = L0 = 0)



- During data reception
  - FIFO not used

    $\overline{RTS}$ outputs "H" upon the reception of the bit preceding the stop bit. After reading the reception data, $\overline{RTS}$ outputs "L".

Figure 3-6 Hardware Flow Control for Data Reception (FIFO Not Used) (SMR:SBL = 0, ESCR:ESBL = INV = PEN = L2 = L1 = L0 = 0)

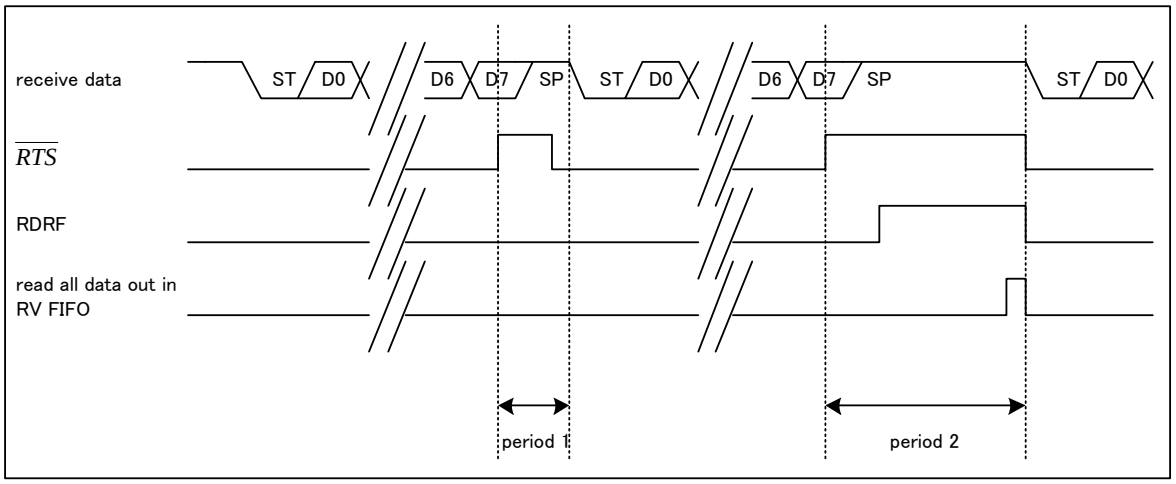


- FIFO used

  If SSR:RDRF is not set (if the reception FIFO has not received the predefined data count), $\overline{RTS}$ outputs "H" upon the reception of the bit preceding the stop bit. However, $\overline{RTS}$ is set to "L" upon the detection of a stop bit. (period 1)

  If SSR:RDRF is set (if the reception FIFO has received the predefined data count), $\overline{RTS}$ outputs "H" upon the reception of the bit preceding the stop bit. $\overline{RTS}$ outputs "L" once all the reception FIFO data has been read. (period 2)

Figure 3-7 Hardware Flow Control for Data Reception (FIFO Used)
(SMR:SBL = 0, ESCR:ESBL = INV = PEN = L2 = L1 = L0 = 0)



Notes:

- *When reception operation is disabled (RXE = 0), the $\overline{RTS}$ signal is fixed to "L".*
- *During the use of the reception FIFO, if both of the following conditions are satisfied, RDRF is set to "1" when the reception idle condition continues for 8 baud rate clocks, but the $\overline{RTS}$ signal remains at "L".*
  - The reception FIFO idle detection enable bit (FCR1:FRIIE) is "1".
  - The reception FIFO contains data but the data count has not reached the predefined data count.
- *Programmable reset (SCR:UPCL = 1) clears the $\overline{RTS}$ signal to "L".*

4. Serial Timer Operation

The serial timer can be used as either a timer function or a synchronous transmission function.

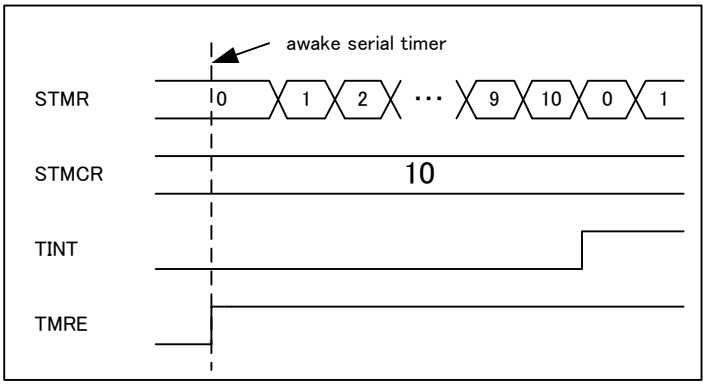
Serial Timer Operation

Activating the Serial Timer

The serial timer can be activated either by setting the serial timer enable bit (SACSR:TMRE) to "1" or by an external trigger.

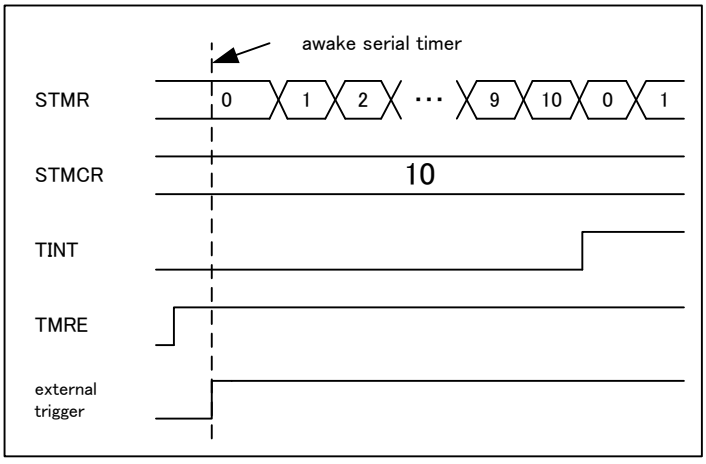
- Activation by the serial timer enable bit (SACSR:TMRE)
  When the external trigger enable bit (SACSR:TRGE) is set to "0", setting the serial timer enable bit (SACSR:TMRE) to "1" activates the serial timer and causes the serial timer register (STMR) to start counting from 0.

Figure 4-1 Activation by Serial Timer Enable Bit (STMCR = 10, SACSR:TRGE = 0, TSYNE = 0)



- Activation by an external trigger
  The serial timer is activated and the serial timer register (STMR) starts counting from 0 when all of the following are satisfied: 1. The external trigger enable bit (SACSR:TRGE) is set to "1". 2. An external trigger edge specified by the trigger selection bit (TRG1, 0) is detected. 3. The serial timer enable bit (SACSR:TMRE) is set to "1".

Figure 4-2 Activation by an External Trigger (STMCR = 10, SACSR:TRGE = 1, TSYNE = 0, TRG1 = 0, TRG0 = 0)



Note:

- *Detection of an external trigger edge specified by the trigger selection bit (SACSR:TRG1, 0) does not start the serial timer if both of the following cases are satisfied: 1. The external trigger enable bit (SACSR:TRGE) is "1". 2. The serial timer enable bit (SACSR:TMRE) is "0".*

Stopping the Serial Timer

The serial timer stops when the serial timer enable bit (SACSR:TMRE) is set to "0". At that time, the value of the serial timer register (STMR) is retained.

Timer Operation

When the synchronous transmission enable bit (SACSR:TSYNE) is "0", the serial timer operates as a timer.

The timer interrupt flag (SACSR:TINT) is set to "1" and the serial timer register (STMR) is reset to 0 when the serial timer register (STMR) coincides with the serial timer comparison register (STMCR).

Figure 4-3 Timer Operation (STMCR = 10, SACSR:TSYNE = 0)

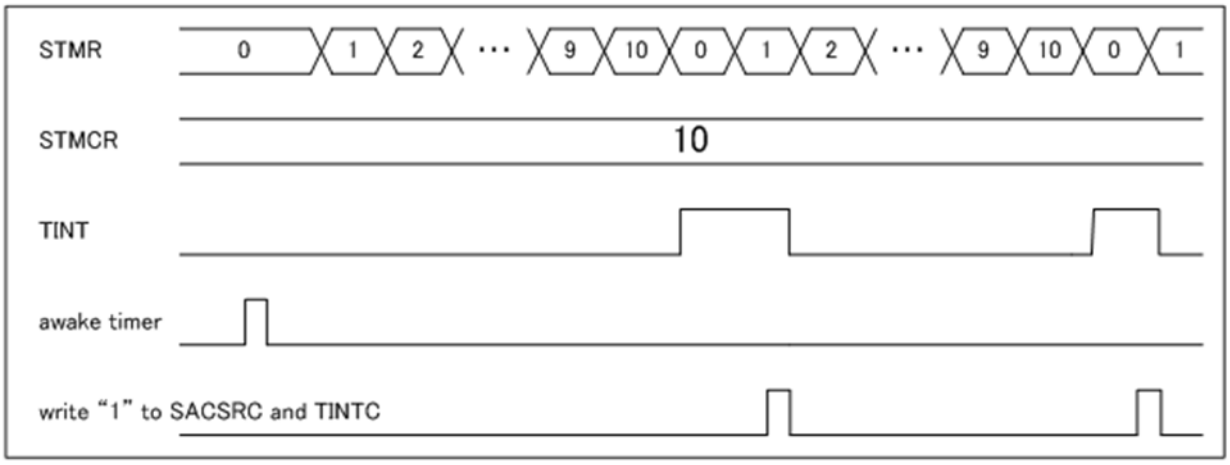


Figure 4-4 Flow Chart of Serial Timer Initialization

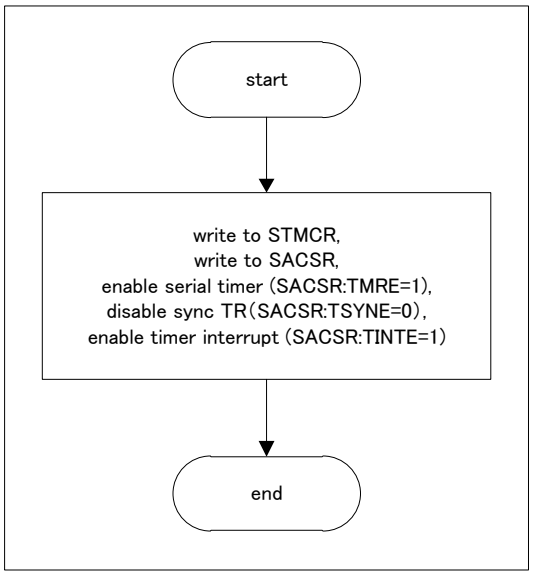
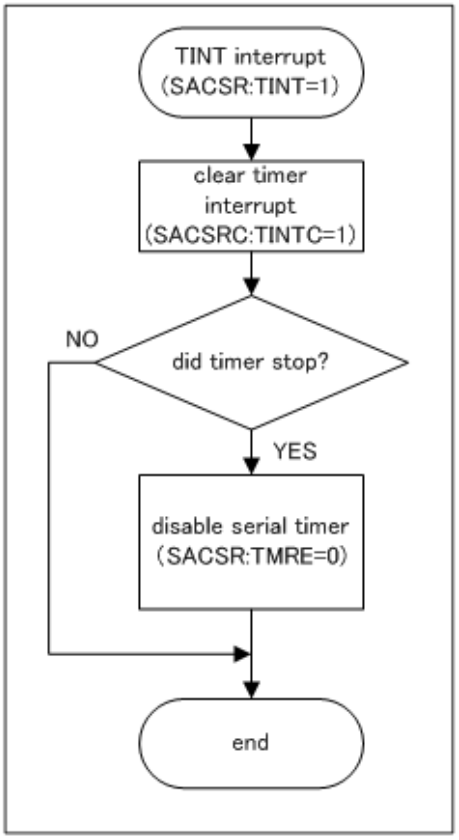


Figure 4-5 Flow Chart of Serial Timer Interrupt Processing



Note:

- *In the state in which synchronous transmission is disabled (SACSR:TSYNE = 0) and 0x0000 is set in the timer comparison register (STMCR), if the timer is operating and the division value*

(SACSR:TDIV3-0) of the timer operation clock is set to 0b0000, the timer interrupt flag (SACSR:TINT) is fixed to "1".

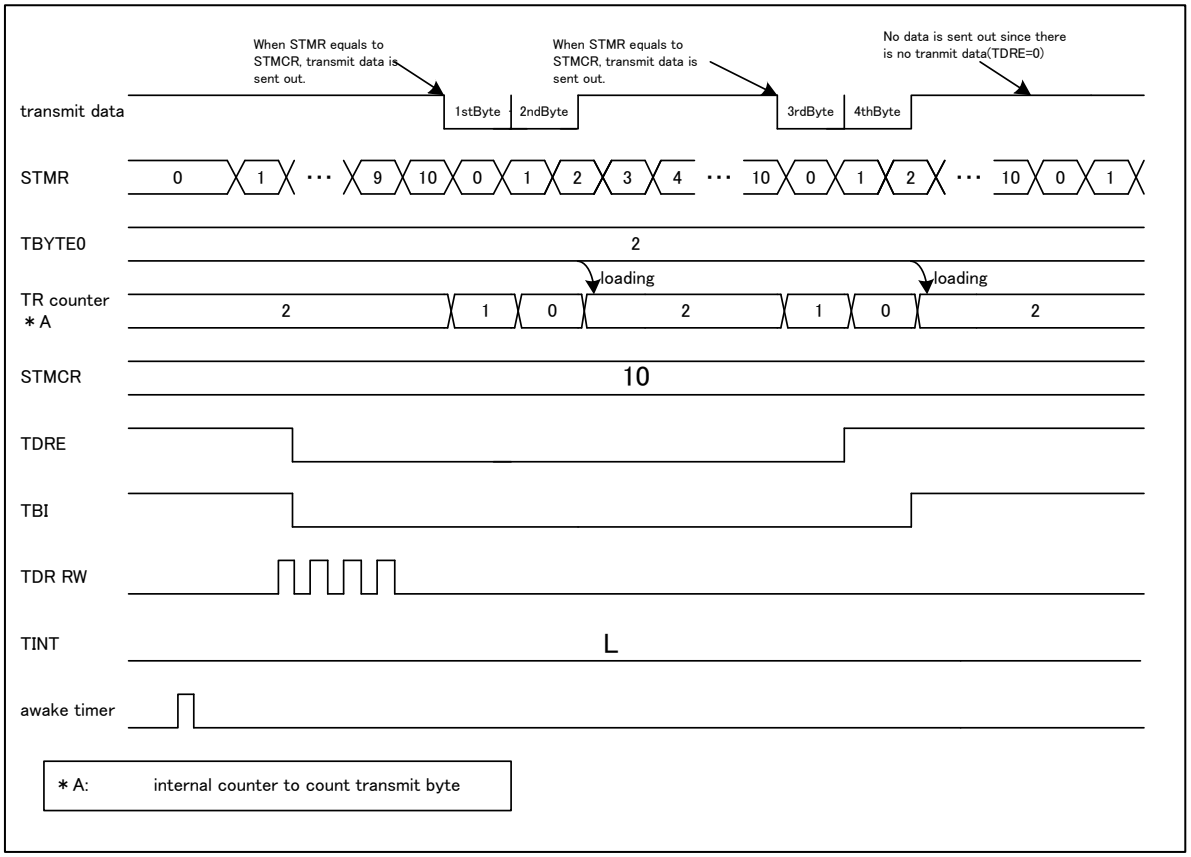
Transmission Operation Synchronized with the Timer

The serial timer is used for synchronous transmission when the synchronous transmission enable bit (SACSR:TSYNE) is "1".

Timer-synchronized transmission operates in the following manner:

1. If the transmission data register contains data (SSR:TDRE = 0) and if the serial timer register (STMR) and the serial timer comparison register (STMCR) coincide, transmission operation starts, and the serial timer register (STMR) is reset to 0. Data transmission is performed for the data count specified in TBYTE0.
2. After transmitting the data count as specified by TBYTE0, transmission is suspended until the next time the serial timer register (STMR) and serial timer comparison register (STMCR) coincide.

Figure 4-6 Timer-Synchronized Transmission (SACSR:TSYNE = 1, STMR = 10, TBYTE0 = 2)



When synchronous transmission is enabled (SACSR:TSYNE = 1) and the serial timer register (STMR) and the serial timer comparison register (STMCR) coincide, transmission is not activated under the following conditions:

- Transmission is disabled (SCR:TXE = 0).
- The transmission data register does not contain valid data (SSR:TDRE = 1).

However, writing transmission data to the transmission data register instantly starts transmission when all of the following are satisfied: 1. The transmission data register does not contain valid data

(SSR:TDRE = 1). 2. Synchronous transmission is enabled (SACSR:TSYNE = 1). 3. The serial timer register (STMR) and serial timer comparison register (STMCR) coincide.

If the transmission data register (TDR) contains valid data (SSR:TDRE = 0) after transmission of the data count specified by TBYTE0, the data is not transmitted until the next time the serial timer register (STMR) and the serial timer comparison register (STMCR) coincide.

If, however, synchronous transmission is enabled (SACSR:TSYNE = 1) and transmission operation is in progress (SSR:TBI = 0), and the serial timer register (STMR) and the serial timer comparison register (STMCR) coincide, a transmission reservation is made. If transmission reservation has been made, transmission does not stop after transmission of the data count specified by TBYTE0, and the subsequent transmission is performed.

Transmission reservation is released under one of the conditions below.

- Programmable reset (SCR:UPCL = 1)
- Transmission disabled (SCR:TXE = 0)

Figure 4-7 Flow Chart of Initialization of Transmission Synchronized with Timer

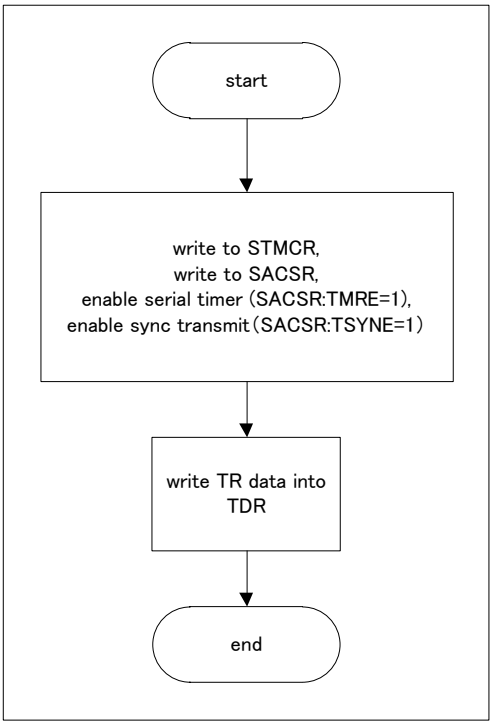
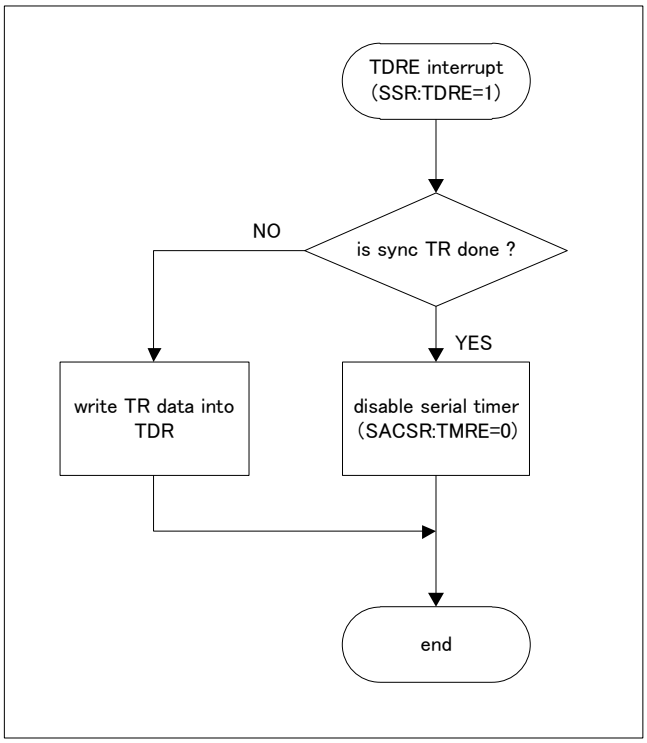


Figure 4-8 Flow Chart of Interrupt Processing in Transmission Synchronized with Timer



Notes:

- *If synchronous transmission is enabled (SACSR:TSYNE = 1), external triggers are enabled (SACSR:TRGE = 1), and (0000) H is set in the timer comparison register (STMCR), transmission synchronized with the timer is not performed, but transmission operation synchronized with an external trigger is performed.*
- *If the transmission data register (TDR) does not contain valid transmission data (SSR:TDRE = 1) before transmitting a data frame of the size specified by TBYTE0, transmission is suspended until the transmission data is written to the transmission data register (TDR). Writing transmission data to the transmission data register (TDR) restarts transmission.*

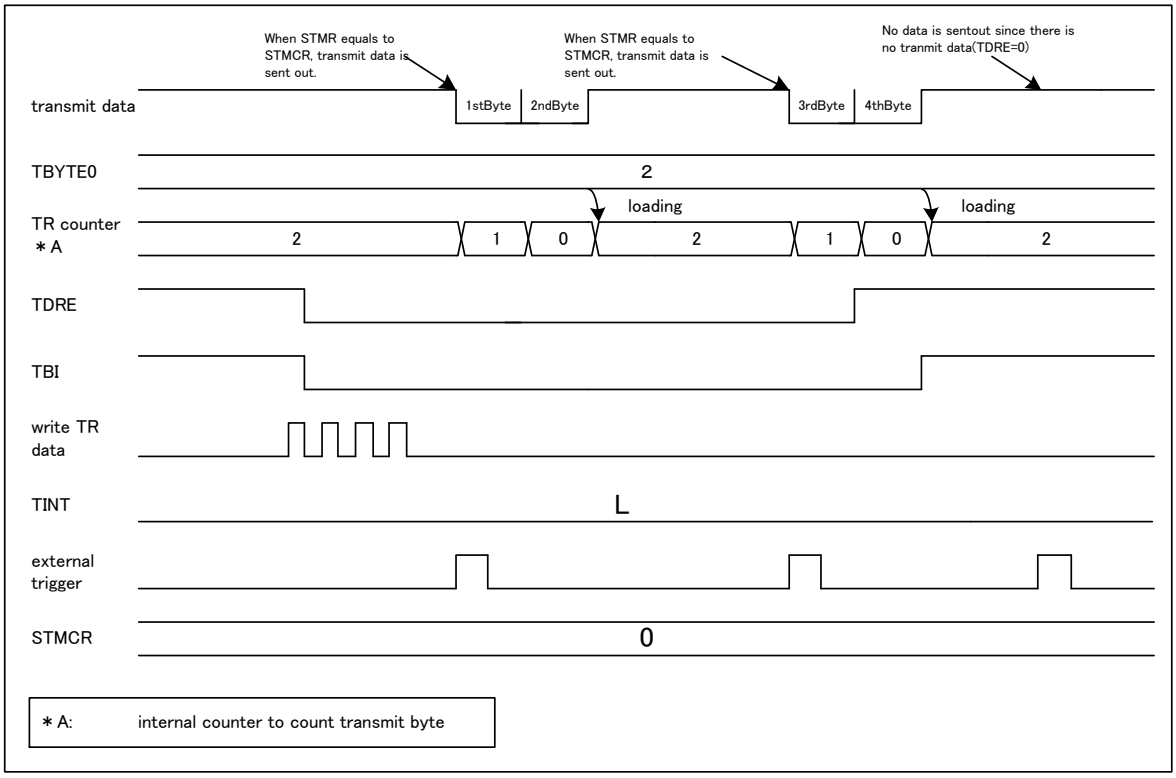
Transmission Operation Synchronized with an External Trigger

If synchronous transmission is enabled (SACSR:TSYNE = 1), external triggers are enabled (SACSR:TRGE = 1), (0000) H is set in the timer comparison register (STMCR), transmission is enabled (SCR:TXE = 1), and there is valid transmission data (SSR:TDRE = 0), transmission operation synchronized with an external trigger is started.

The start of transmission synchronized with an external trigger is performed as described below.

1. The operation starts immediately upon the detection of an external trigger edge specified by the trigger selection bit (SACSR:TRG1, 0). Data transmission is performed for the data count specified in TBYTE0.
2. After the completion of the transmission of the data count specified by TBYTE0, transmission is suspended until an external trigger edge specified by the trigger selection bit (SACSR:TRG1, 0) is next detected.

Figure 4-9 External-Trigger-Synchronized Transmission (SACSR:TSYNE = 1, TRG1 = 0, TRG0 = 0)



During external-trigger-synchronized transmission, the detection of an external trigger edge specified by the trigger selection bit (SACSR:TRG1, 0) does not start transmission under the following conditions:

- Transmission is disabled (SCR:TXE = 0).
- The transmission data register does not contain valid data (SSR:TDRE = 1).

However, writing transmission data to the transmission data register starts transmission immediately provided all of the following are satisfied: 1. The external trigger edge specified by the trigger selection bit (SACSR:TRG1, 0) is detected. 2. The transmission data register does not contain valid data (SSR:TDRE = 1).

If the transmission data register (TDR) contains valid data (SSR:TDRE = 0) after transmission of the data count specified by TBYTE0, the following occurs: The remaining data is not transmitted until an external trigger edge specified by the trigger selection bit (SACSR:TRG1, 0) is next detected.

However, when an external trigger edge specified by the trigger selection bit (SACSR:TRG1, 0) is detected during transmission (SSR:TBI = 0) with synchronous transmission enabled (SACSR:TSYNE = 1), transmission reservation is performed. If transmission reservation has been performed, transmission does not stop after transmission of the data count specified by TBYTE0, and the subsequent transmission occurs.

Transmission reservation is released under one of the conditions below.

- Programmable reset (SCR:UPCL = 1)
- Transmission disabled (SCR:TXE = 0)

Figure 4-10 Flow Chart of Initialization of Transmission Synchronized with External Trigger

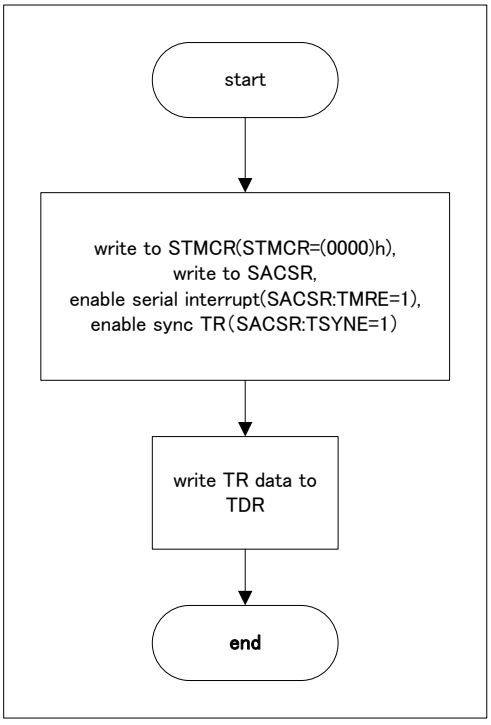
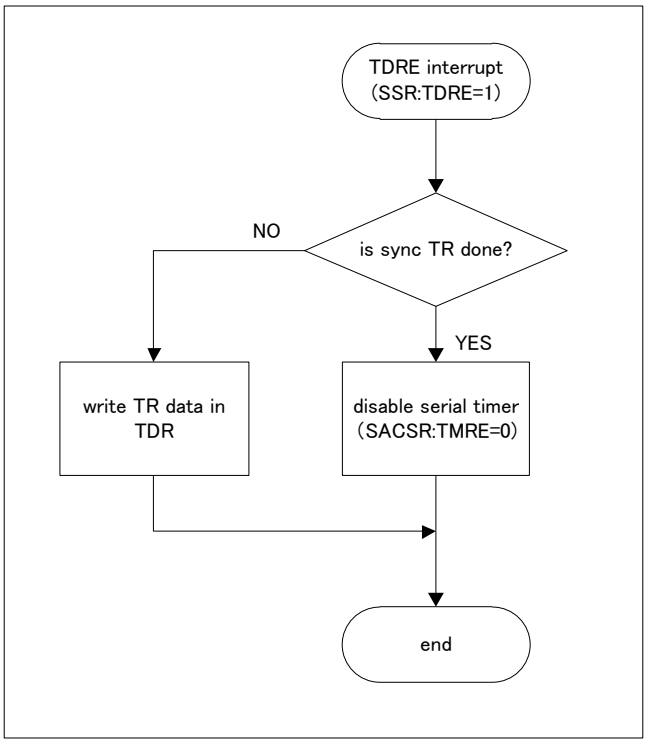


Figure 4-11 Flow Chart of Interrupt Processing in Transmission Synchronized with External Trigger



Note:

- *If the transmission data register (TDR) does not contain valid transmission data (SSR:TDRE = 1) before transmitting a data frame of the size specified by TBYTE0, transmission is suspended until the transmission data is written to the transmission data register (TDR). Writing transmission data to the transmission data register (TDR) restarts transmission.*

5. Test Mode

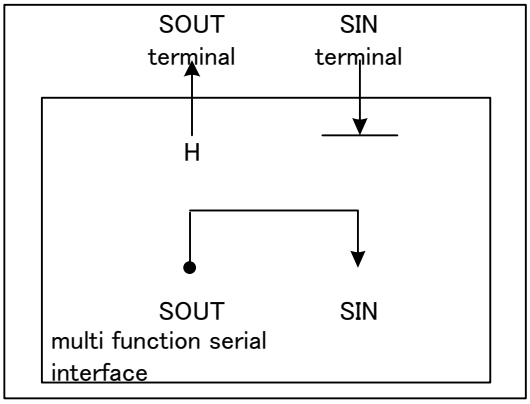
This section describes operation in a test mode.

Serial Test Mode

When serial test mode is enabled (SACSR:STST = 1), SOUT and SIN are connected together inside the multi-function serial interface, so that the data transmitted from SOUT can be received from SIN directly.

When serial test mode is enabled (SACSR:STST = 1), the SOUT pin is fixed to "H", and the data input to the SIN pin is ignored.

Figure 5-1 Serial Test Mode



Note:

- *The value of the serial test mode enable bit (SACSR:STST) can be changed only when transmission and reception are prohibited (SCR:TXE = RXE = 0).*

6. Dedicated Baud Rate Generator

A transmission and reception clock source for the UART can be selected from the following:

- Dedicated baud rate generator (reload counter)
- Input of an external clock to the baud rate generator (reload counter)

UART Baud Rate Selection

A baud rate can be selected from the following 2 types:

Baud Rate Obtained by Dividing the Internal Clock by the Dedicated Baud Rate Generator (Reload Counter)

There are 2 internal reload counters, each of which corresponds to the transmission or reception serial clock. A baud rate can be selected by setting a 15-bit reload value in baud rate generator register 1 or 0 (BGR1, BGR0).

The reload counters divide the internal clock according to the setting value.

Set the clock source by selecting an internal clock (BGR1:EXT = 0).

Baud Rate Obtained by Dividing an External Clock by the Dedicated Baud Rate Generator (Reload Counter)

An external clock is used as the clock source of the reload counter.

A baud rate can be selected by setting a 15-bit reload value in baud rate generator register 1 or 0 (BGR1, BGR0).

The reload counters divide the external clock according to the setting value.

Set a clock source by selecting an external clock and using a baud rate generator clock (BGR1:EXT = 1).

This mode is provided for dividing and using a special-frequency oscillator.

Notes:

- *Set the external clock (BGR1:EXT = 1) under the condition whereby the reload counter is stopped (BGR1/0 = 0x0000).*
- *When the external clock setting is specified (BGR1:EXT = 1), the "H" width and "L" width of the external clock must be 2 bus clocks or more.*

6.1. Setting the Baud Rate

This section describes baud rate setting. This section also describes the result of calculating the serial clock frequency.

Calculation of Baud Rate

The 2 15-bit reload counters are set by baud rate generator register 1, 0 (BGR1, BGR0).
The baud rate calculation formulas are shown below.

(1) Reload value:

```
V = Φ / b - 1

V : reload value  b : baud rate   Φ :  frequency of bus clocl or that of external clock
```

(2) Calculation example

```
When bus clock is 16MHz, using internal clock, baud rate is 19200 bps, the reload value can be
calculated as follows.
                reload value :
                    V = (16 x 1000000)/19200 - 1 = 832
                          then baud rate will be
                    b = (16 x 1000000)/(832 + 1) = 19208bps
```

(3) Baud rate error

The baud rate error is obtained using the following formula:

```
error rate(%) = (calculated value - target value)/target value * 100
  (ex. When bus clock is 20MHz, target baud rate is 153600 bps,
       the result will be as follows.
               reload value    = (20 x 1000000)/153600 - 1 = 129
          calculated baud rate =  20 x 1000000)/(129 + 1) = 153846 (bps)
               error rate (%) = (153846 - 153600)/153600 x 100 = 0.16 (%)
```

Notes:

- *Setting the reload value to "0" stops the reload counter.*
- *When the reload value is even, the "L" width of the reception serial clock is 1-bus-clock-cycle longer than the "H" width. When the reload value is odd, the "L" width and the "H" width of the serial clock are equal.*
- *Set the reload value to 4 or larger. However, data may not be received normally depending on baud rate errors and the reload value setting.*
- *When considering the tolerable baud rate range, also consider the impact of the jitter of the clock input to the macro.*

Example of Reload Values and Baud Rate Settings for Individual Bus Clock Frequencies

The following table lists example reload values and baud rate settings.

Table 6-1 Example Reload Values and Baud Rate Settings

Baud Rate	8 MHz		10 MHz		16 MHz		20 MHz		24 MHz		32 MHz	
(bps)	Value	ERR	Value	ERR	Value	ERR	Value	ERR	Value	ERR	Value	ERR
4 M	-	-	-	-	-	0	4	0	5	0	7	0
2.5 M	-	-	-	-	-	-	7	0	-	-	-	-
2 M	-	0	4	0	7	0	9	0	11	0	15	0
1 M	7	0	9	0	15	0	19	0	23	0	31	0
500000	15	0	19	0	31	0	39	0	47	0	63	0
460800	-	-	-	-	-	-	-	-	51	0.16	-	-
250000	31	0	39	0	63	0	79	0	95	0	127	0
230400	-	-	-	-	-	-	86	-0.22	103	0.16	138	-0.08
153600	51	0.16	64	0.16	103	0.16	129	0.16	155	0.16	207	0.16
125000	63	0	79	0	127	0	159	0	191	0	255	0
115200	-	-	86	-0.22	138	-0.08	173	-0.22	207	0.16	277	-0.08
76800	103	0.16	129	0.16	207	0.16	259	0.16	311	-0.16	416	-0.08
57600	138	-0.08	173	-0.22	277	-0.08	346	0.06	416	-0.08	555	-0.08
38400	207	0.16	259	0.16	416	-0.08	520	-0.03	624	0	832	0.04
28800	277	-0.08	346	<0.01	554	-0.01	693	0.06	832	0.03	1110	0.01
19200	416	-0.08	520	-0.03	832	0.03	1041	-0.03	1249	0	1666	-0.02
10417	767	<0.01	959	<0.01	1535	<0.01	1919	<0.01	2303	<0.01	3071	<0.01
9600	832	0.04	1041	-0.03	1666	-0.02	2083	0.03	2499	0	3332	0.01
7200	1110	<0.01	1388	<0.01	2221	<0.01	2777	<0.01	3332	<0.01	4443	0.01
4800	1666	-0.02	2082	0.02	3332	<0.01	4166	<0.01	4999	0	6666	<0.01
2400	3332	<0.01	4166	<0.01	6666	<0.01	8332	<0.01	9999	0	13332	<-0.01
1200	6666	<0.01	8334	0.02	13332	<0.01	16666	<0.01	19999	0	26666	<0.01
600	13332	<0.01	16666	<0.01	26666	<0.01	-	-	-	-	-	-
300	26666	<0.01	-	-	-	-	-	-	-	-	-	-

- Value: Setting value of the BGR1/0 registers (decimal)
- ERR: Baud rate error (%)

Table 6-2 Example Reload Values and Baud Rate Settings (Continued)

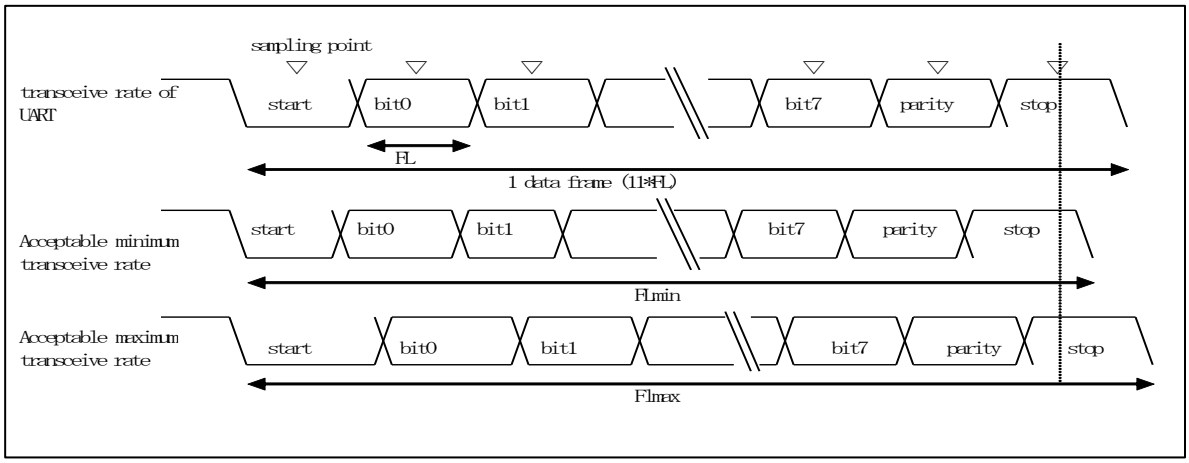
Baud Rate (bps)	40 MHz		48 MHz		72 MHz		80 MHz	
	Value	ERR	Value	ERR	Value	ERR	Value	ERR
4 M	9	0	11	0	17	0	19	0
2.5 M	15	0	-	-	-	-	31	0
2 M	19	2	23	0	35	0	39	0
1 M	39	0	47	0	71	0	79	0
500000	79	0	95	0	143	0	159	0
460800	86	-0.22	103	0.16	155	0.16	173	-0.22
250000	159	0	191	0	287	0	319	0
230400	173	-0.22	207	0.16	312	-0.16	346	0.06
153600	259	0.16	312	-0.16	468	-0.05	520	-0.03
125000	319	0	383	0	575	0	639	0
115200	346	0.06	416	-0.08	624	0	693	0.06
76800	520	-0.03	624	0	937	-0.05	1041	-0.03
57600	693	0.06	832	0.04	1249	0	1388	<0.01
38400	1041	-0.03	1249	0	1874	0	2082	0.01
28800	1388	<0.01	1666	-0.02	2499	0	2777	<0.01
19200	2082	0.01	2499	0	3749	0	4166	-0.01
10417	3839	<0.01	4607	<0.01	6911	<0.01	7679	<0.01
9600	4166	<0.08	4999	0	7499	0	8332	0
7200	5555	<0.01	6666	<0.01	9999	0	11110	0
4800	8332	<0.01	9999	0.02	14999	0	16666	0
2400	16666	<0.01	19999	0	29999	0	-	-
1200	-	-	-	-	-	-	-	-
600	-	-	-	-	-	-	-	-
300	-	-	-	-	-	-	-	-

Reception Baud Rate Tolerance

The following describes the degree of error in the receiver baud rate that is tolerable during reception.

Use the following formula to ensure that the baud rate errors during data reception fall within the tolerable range.

Figure 6-1 Reception Baud Rate Tolerance



After the detection of a start bit, the sampling timing of the reception data is determined by the counters set in registers BGR1/0, as shown in the figure. The data can be received normally provided all the data up to the last data (stop bit) is included within this sampling timing.

This theoretically gives the following for 11-bit data reception.

For a sampling timing margin of 1 bus clock cycle (Φ), the tolerable minimum transfer rate (FLmin) will be as follows:

FLmin = (11 bits * (V + 1) - (V + 1) / 2 + 2) / Φ = (21 V + 25) / 2 Φ (s) V: Reload value, Φ: Bus clock

Therefore, the maximum receiver baud rate (BGmax) receivable will be as follows:

BGmax = 11 / FLmin = 22 Φ / (21 V + 25) (bps) V: Reload value, Φ: Bus clock

When receiving data at the tolerable maximum transfer rate (FLmax), sampling is performed at the first point of the 11th-bit in the reception data.

Therefore, the tolerable maximum transfer rate (FLmax) will be as follows:

10 / 11 * FImax = (11 bits * (V + 1) - (V + 1) / 2) / Φ V: Reload value, Φ: Bus clock

FImax = (21 / 20 * 11 * (V + 1)) / Φ

For a sampling timing margin (Φ) of 2 clock cycles, the tolerable maximum transfer rate (FLmax) will be as follows:

FImax = (21 / 20 * 11 * (V + 1) - 2) / Φ = (231 V + 191) / 20 Φ (s) V: Reload value, Φ: Bus clock

Therefore, the minimum receiver baud rate (BGmin) receivable will be as follows:

BGmin = 11 / FLmax = 220 Φ / (231 V + 191) (bps) V: Reload value, Φ: Bus clock

The tolerable baud rate errors for the UART and the receiver will be as follows, given the calculation formulas for the minimum and maximum baud rate values.

Reload Value (V)	Tolerable Maximum Baud Rate Error	Tolerable Minimum Baud Rate Error
3	0%	0
10	+2.98%	-3.24%
50	+4.37%	-4.44%
100	+4.56%	-4.60%
200	+4.66%	-4.68%
32767	+4.76%	-4.76%

Note:

- *The receiving accuracy depends on the number of bits in a frame, the bus clock, and the reload value. A higher accuracy is achieved with a higher bus clock and a higher division ratio.*

External Clock

Writing "1" to the EXT bit in the baud rate generator register (BGR1) causes the baud rate generator to divide the external clock.

Note:

- *The external clock synchronizes with the internal clock of the UART. Therefore, the existence of an external clock that cannot be synchronized makes this operation unstable.*

Reload Counter Function

There is both a transmission reload counter and a reception reload counter, which function as dedicated baud rate generators. These feature 15-bit registers for the reload values. They generate a transmission clock and a reception clock from either an external or internal clock.

Start of Counting

When a reload value is written to the baud rate generator registers (BGR1 and BGR0), the reload counters start counting.

Restarting

A reload counter restarts under the following conditions:

For both the Transmission and Reception Reload Counters

- Programmable reset (SCR:UPCL bit)

For the Reception Reload Counter

- Detection of the falling edge of the start bit in the asynchronous mode

7. Block Transfer

This section describes the operation of block transfer.

Setting Block Transfer

Block transfer can be performed for reception by setting RXBLKEN in the extended control register (ECR) to 1 and for transmission by setting TXBLKEN to 1.

Block transfer can be performed in that mode in which it is operated with the READ/WRITE access standard (RW access mode).

The block size can be set with the transmission block size register and the FBYTE register (reception side).

The threshold at which to output a block transfer request can be set with the FBYTE register for reception and with the FTICR register for transmission. When the amount of data in the reception FIFO exceeds the setting made for the FBYTE register, a reception block transfer request is output. When the amount of free space in the transmission FIFO exceeds the setting made for the FTICR register, a transmission block transfer request is output.

7.1. RW Access Mode

This section describes block transfer in RW access mode.

Reception Block Transfer

Reception block transfer is performed as described below.

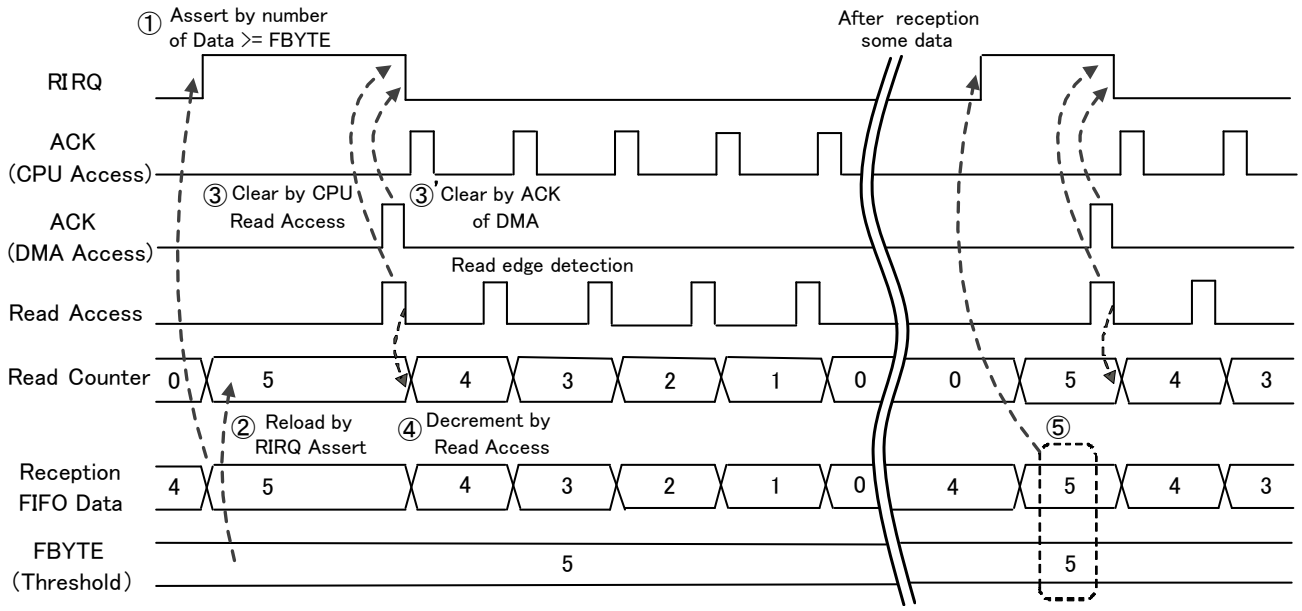
If the amount of data in the reception FIFO is same or more than the threshold (FBYTE), a reception block transfer request (RIRQ) is asserted. At this time, the value of the reception block size is reloaded from the FBYTE register to the Read counter. When the reception FIFO is not used, the value of the reception block size is "1".

After the reception block transfer request, if a read to the reception data occurs, the reception block transfer request (RIRQ) is cleared. With a read to the reception data, the Read counter is decremented.

When the Read counter becomes "0", if the amount of data in the reception FIFO is same or more than the threshold (FBYTE), the reception block transfer request (RIRQ) is asserted again. With the assertion of the reception block transfer request (RIRQ), the Read counter is reloaded.

During application operation, if the next block transfer request occurs before the completion of the previous block transfer, this state might be reported to the system by a block transfer error. This state does not occur if you configure FBYTE=1. Completion of block transfer by DMA depends on occupation of bus transaction by application. You need to fully evaluate the application integrally, design the system, and choose the optimal number of block transfers.

Figure 7-1 Reception Block Transfer in RW Access Mode



Transmission Block Transfer

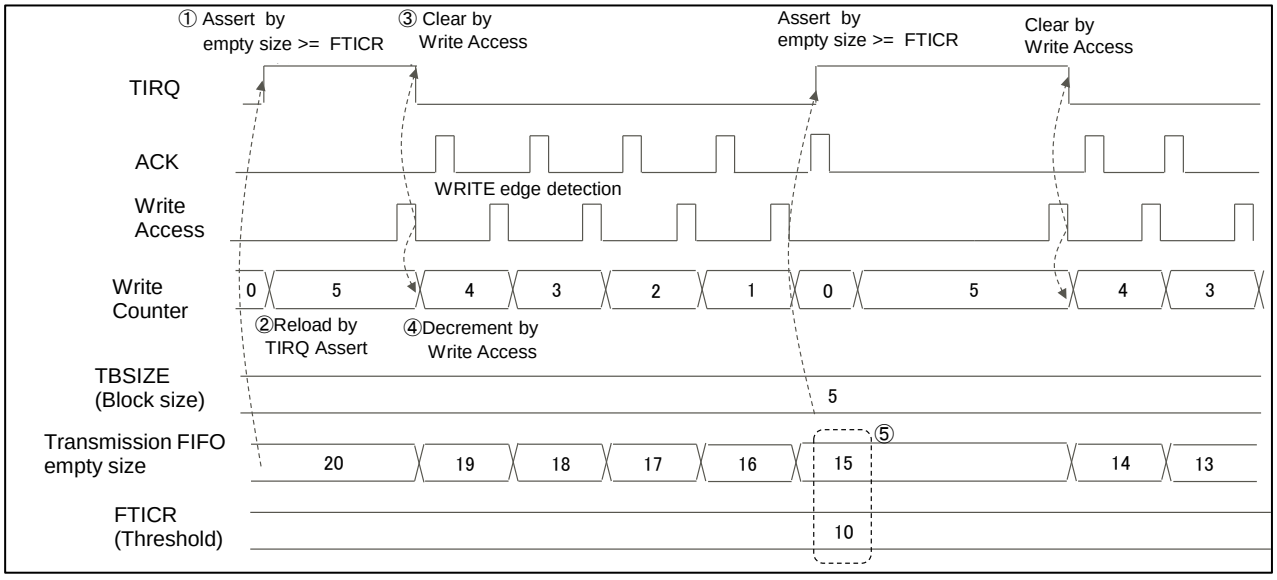
Transmission block transfer is performed as described below.

If the amount of free space in the transmission FIFO exceeds the threshold (FTICR), a transmission block transfer request (TIRQ) is asserted. At this time, the value of the transmission block size is reloaded from the TBSIZE register to the Write counter. When the transmission FIFO is not used, the value of the transmission block size is "1".

After the transmission block transfer request, if a write to the transmission data occurs, the transmission block transfer request (TIRQ) is cleared. With a write to the transmission data, the Write counter is decremented.

When the Write counter becomes "0", if the amount of free space in the transmission FIFO exceeds the threshold (FTICR), the transmission block transfer request (TIRQ) is asserted again. With the assertion of the transmission block transfer request (TIRQ), the Write counter is reloaded.

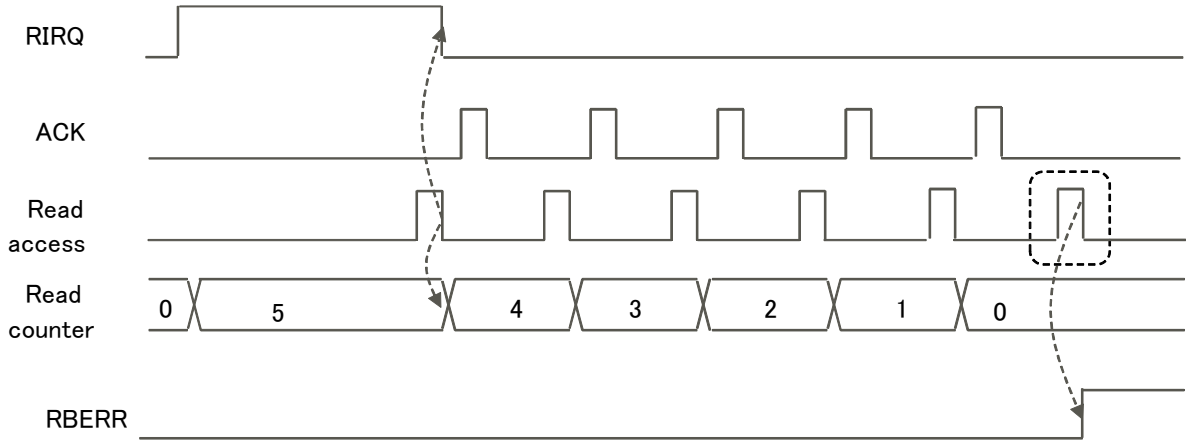
Figure 7-2 Transmission Block Transfer in RW Access Mode



Reception Block Transfer Error

If, while the Read counter is "0", a read access occurs before the next reception block transfer request (RIRQ) is asserted, a reception block transfer error occurs, and the ESR:RBERR bit is asserted.

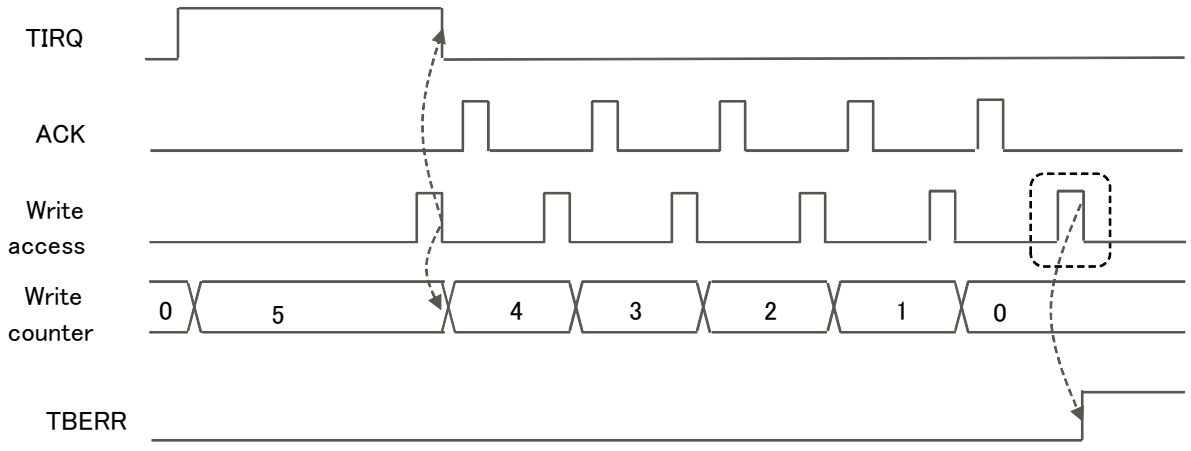
Figure 7-3 Reception Block Transfer Error in RW Access Mode



Transmission Block Transfer Error

If, while the Write counter is "0", a write access occurs before the next transmission block transfer request (TIRQ) is asserted, a transmission block transfer error occurs, and the ESR:TBERR bit is asserted.

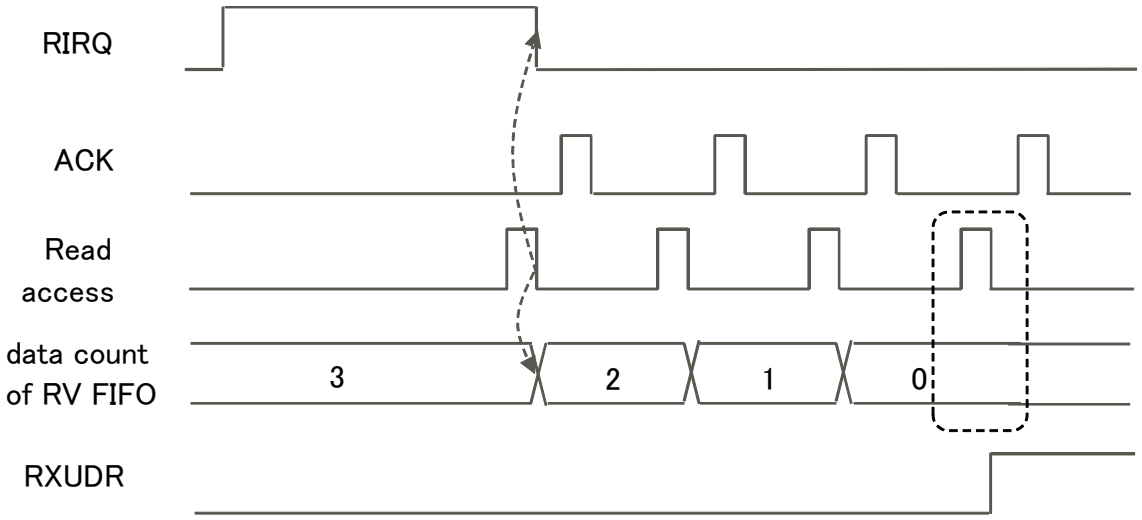
Figure 7-4 Transmission Block Transfer Error in RW Access Mode



Reception FIFO under Run

If, while the amount of data in the reception FIFO is "0", reception data is read from the reception FIFO, a reception FIFO under run occurs, and the ESR:RXUDR bit is asserted.

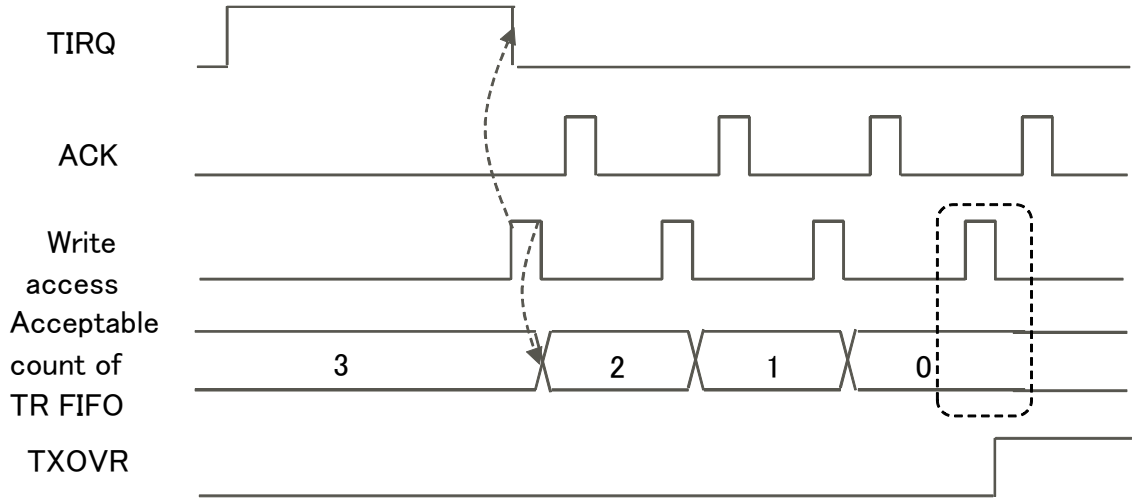
Figure 7-5 Reception FIFO Under Run in RW Access Mode



Transmission FIFO Overrun

If, while the amount of free space in the transmission FIFO is "0", transmission data is written to the transmission FIFO, a transmission FIFO overrun occurs, and the ESR:TXOVR bit is asserted.

Figure 7-6 Transmission FIFO Overrun in RW Access Mode



8. Setup Procedure and Program Flow for Operation Mode 0 (Asynchronous Normal Mode)

Operation mode 0 allows asynchronous serial bidirectional communication.

Connection between MCUs

Select bidirectional communication for operation mode 0 (normal mode). As shown in Figure 8-1, interconnect two MCUs.

Figure 8-1 Example of Bidirectional Communication Connection for UART Operation Mode 0 (Flow Control Prohibited)

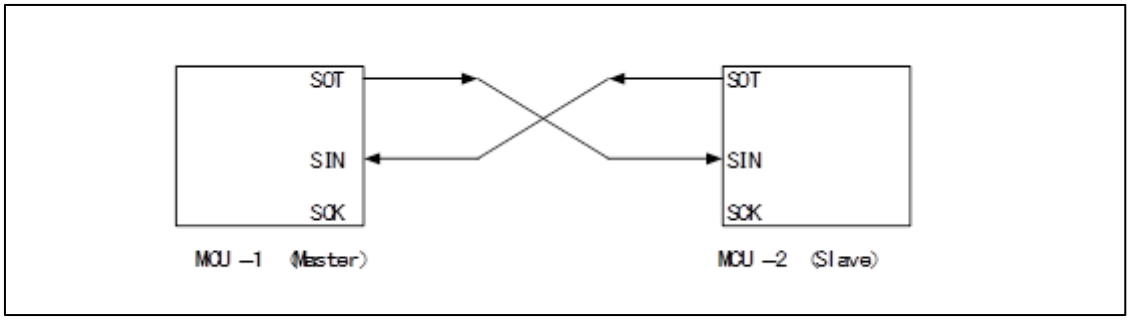
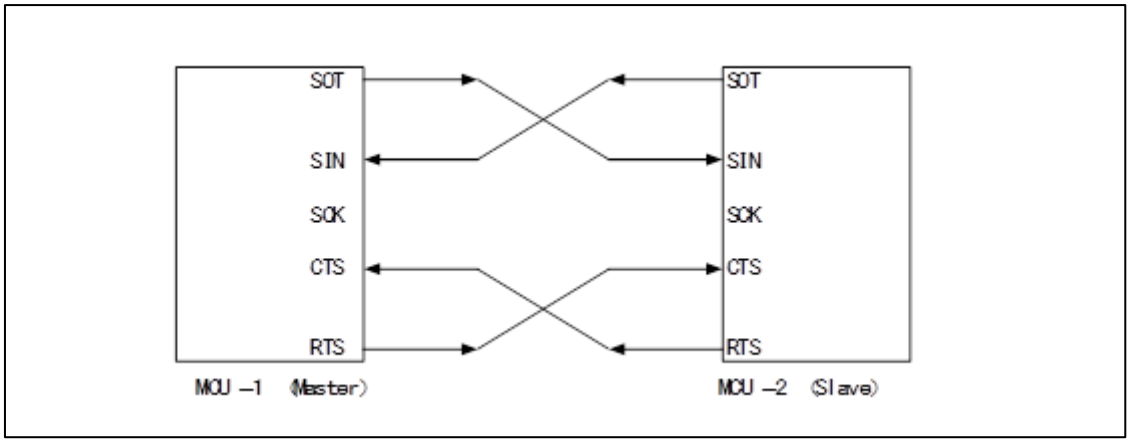


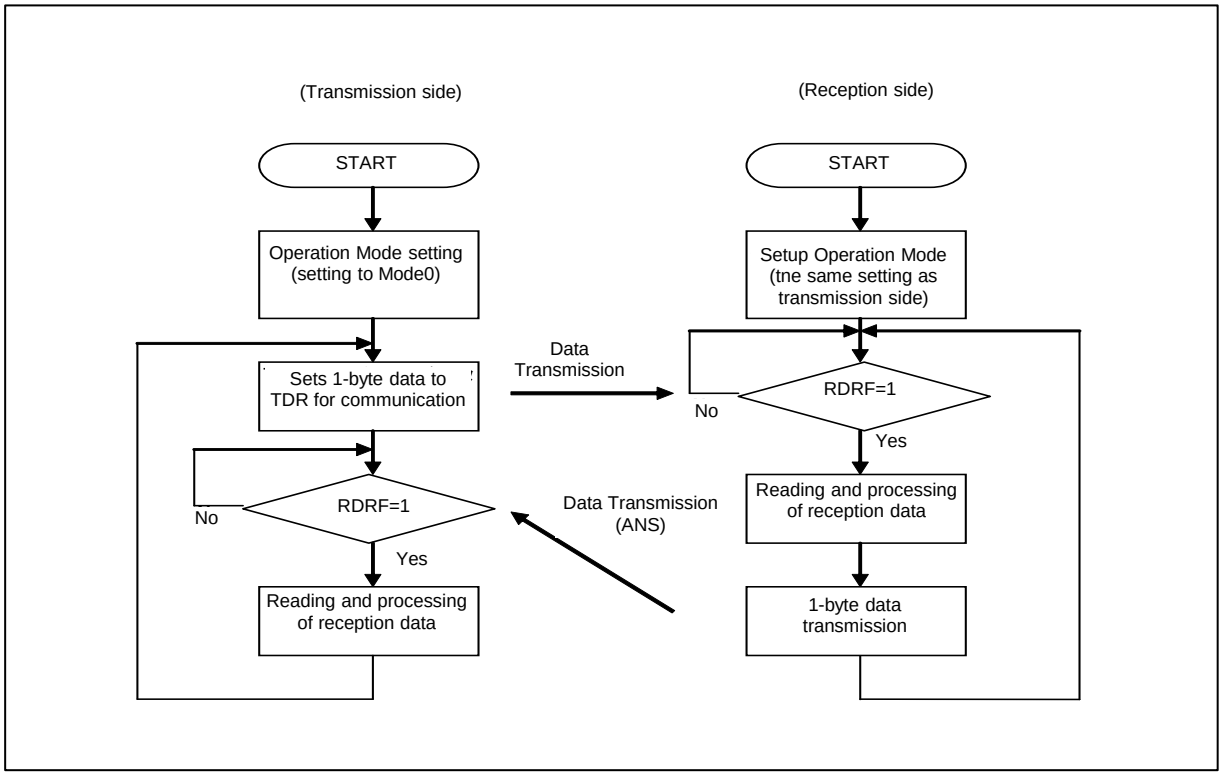
Figure 8-2 Example of Bidirectional Communication Connection for UART Operation Mode 0 (Flow Control)



Flow Chart

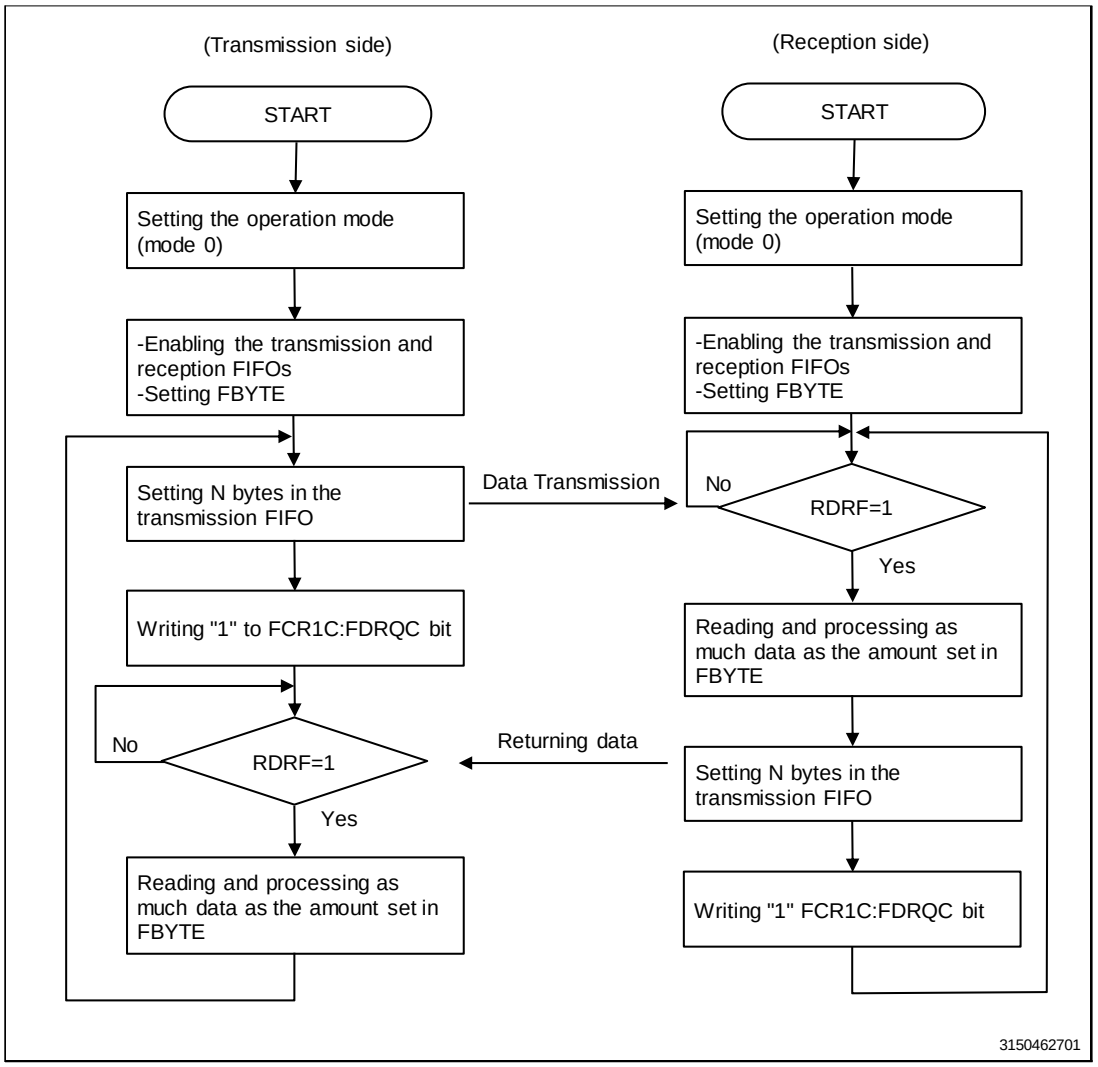
FIFO Not Used

Figure 8-3 Example of Bidirectional Communication Flow Chart (FIFO Not Used)



FIFO Used

Figure 8-4 Example of Bidirectional Communication Flow Chart (FIFO Used)



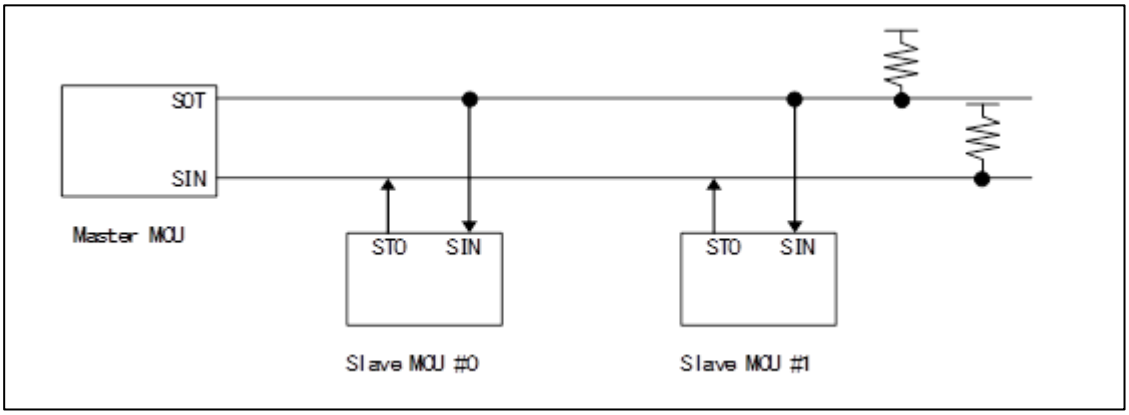
9. Setup Procedure and Program Flow for Operation Mode 1 (Asynchronous Multi-processor Mode)

Operation mode 1 (multi-processor mode) allows communication in master/slave connections of multiple MCUs. This mode can be used for master/slave-type communication.

Connection between MCUs

Master/slave-type communication enables a communication system that has 2 shared communication lines connecting 1 master MCU and multiple slave MCUs, as shown in the figure. The UART can be used either as a master or a slave.

Figure 9-1 Connection Example for UART Master/Slave-Type Communication



Function Selection

Select an operation mode and a data transfer method for master/slave-type communication as shown in Table 9-1.

Table 9-1 Selection of Master/Slave-Type Communication Function

	Operation Mode		Data	Parity	Stop Bit	Bit Direction
	Master MCU	Slave MCU				
Address Transmission/Reception	Mode 1 (A/D bit transmission)	Mode 1 (A/D bit reception)	AD = 1 + 7-bit or 8-bit address	None	1 bit or 2 bits	LSB or MSB first
Data Transmission/Reception			AD = 0 + 7-bit or 8-bit data			

Note:

- *Access the transmission/reception data (TDR/RDR) in word units for operation mode 1.*

Communication Procedure

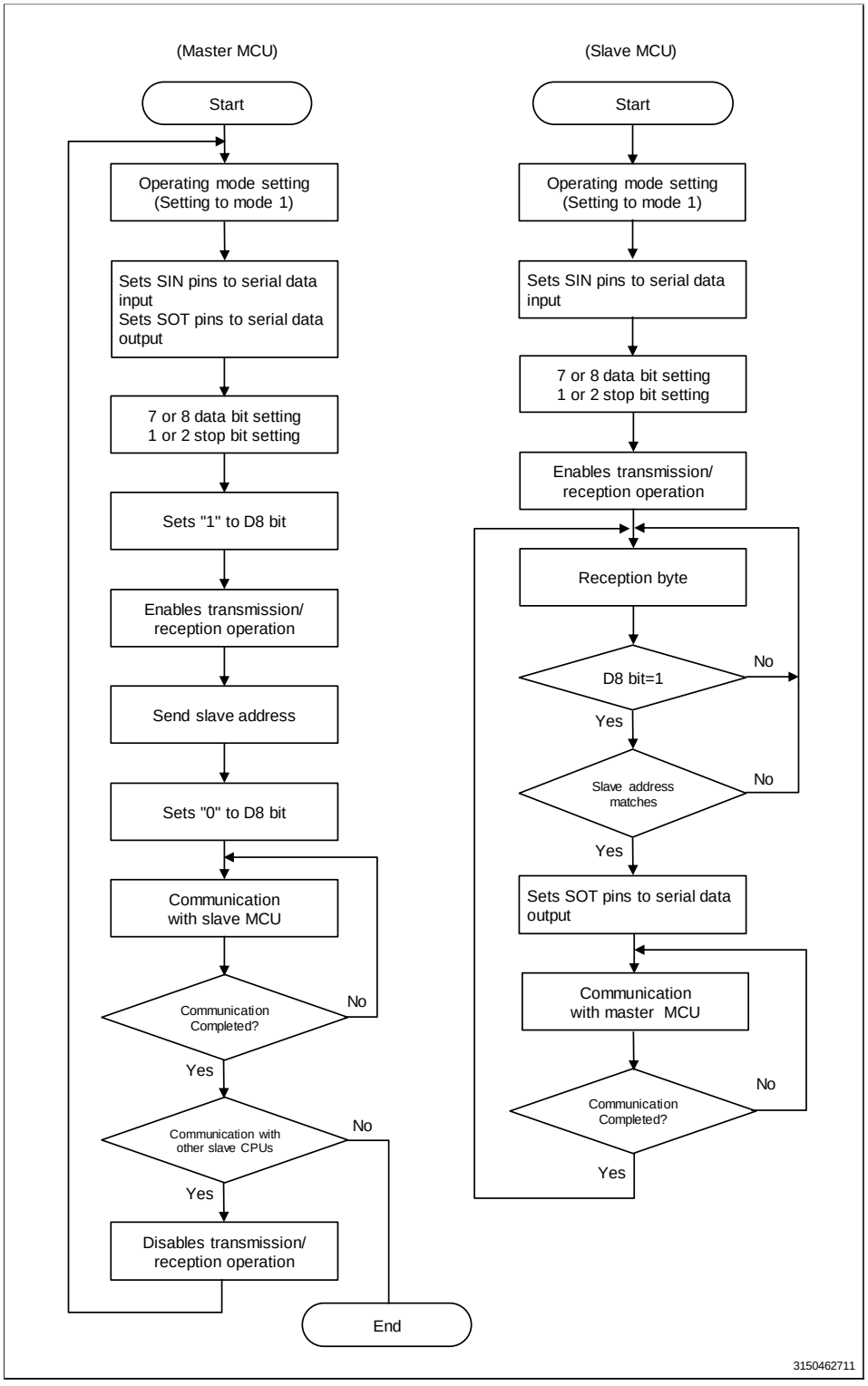
Communication starts when the master MCU transmits address data. Address data, for which the D8 bit is set to "1", selects the destination slave MCU. Each slave CPU checks the address data by its program, and if it coincides with the allocated address, the MCU communicates (by normal data) with the master MCU.

Figure 9-2 and Figure 9-3 provide flow charts of master/slave-type communication (multi-processor mode).

Flow Chart

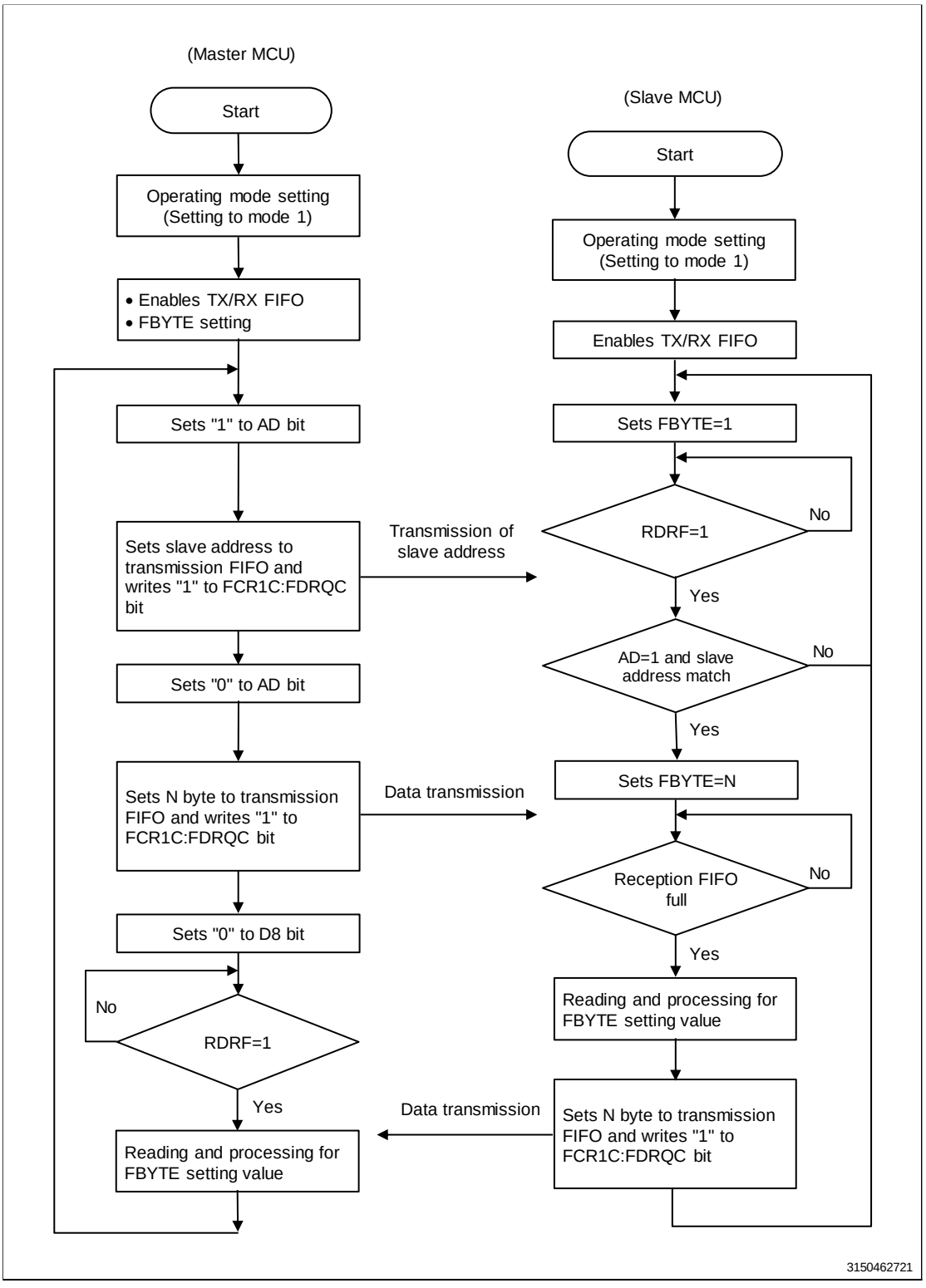
FIFO Not Used

Figure 9-2 Example of Flow Chart of Master/Slave-Type Communication (FIFO Not Used)



FIFO Used

Figure 9-3 Example of Flow Chart of Master/Slave-Type Communication (FIFO Used)



10. Registers of UART (Asynchronous Serial Interface)

This section provides a list of the registers of the UART (Asynchronous Serial Interface).

List of Registers of UART (Asynchronous Serial Interface)

Table 10-1 List of Registers of UART (Asynchronous Serial Interface)

	Bit15 Bit8	Bit7 Bit0
UART	SCR (serial control register)	SMR (serial mode register)
	SSR (serial status register)	ESCR (extended communication control register)
	RDR1/TDR1 (transmission and reception data register 1)	RDR0/TDR0 (transmission and reception data register)
	-	-
	SACSR1 (serial auxiliary control status register 1)	SACSR0 (serial auxiliary control status register 0)
	STMR1 (serial timer register 1)	STMR0 (serial timer register 0)
	STMCR1 (serial timer comparison register 1)	STMCR0 (serial timer comparison register 0)
	-	-
	-	-
	-	-
	-	-
	-	-
	-	TBYTE0 (transfer byte register 0)
	-	-
	BGR1 (baud rate generator register 1)	BGR0 (baud rate generator register 0)
	-	-
FIFO	FCR1 (FIFO control register 1)	FCR0 (FIFO control register 0)
	FBYTE2 (FIFO2 byte register)	FBYTE1 (FIFO1 byte register)
	FTICR2 (transmission FIFO interrupt control register 2)	FTICR1 (transmission FIFO interrupt control register 1)
Auxiliary Register	ESR (extended status register)	ECR (extended control register)
	-	TBSIZE (transmission block size register)
	-	-
Clear Register	-	-
	-	-
	SACSR1C (serial auxiliary control status clear register 1)	SACSR0C (serial auxiliary control status clear register 0)
	-	-
	-	-
	-	-
	-	-
	-	-
	FCR1C (FIFO control clear register 1)	FCR0C (FIFO control clear register 0)
	-	-
	-	-
	ESRC (extended status clear register)	-

	Bit15 Bit8	Bit7 Bit0
Set Register	-	-
	-	-
	SACSR1S (serial auxiliary control status set register 1)	SACSR0S (serial auxiliary control status set register 0)
	-	-
	-	-
	-	-
	-	-
	-	-
	FCR1S (FIFO control set register 1)	FCR0S (FIFO control set register 0)
	-	-
Reception/Transmission Register	RDR1/TDR1 (transmission and reception data register 1)*1	RDR0/TDR0 (transmission and reception data register)*1
	-	-
	RDR1/TDR1 (transmission and reception data register 1)*1	RDR0/TDR0 (transmission and reception data register)*1
	-	-
	RDR1/TDR1 (transmission and reception data register 1)*1	RDR0/TDR0 (transmission and reception data register)*1
	-	-
	RDR1/TDR1 (transmission and reception data register 1)*1	RDR0/TDR0 (transmission and reception data register)*1
	-	-
	RDR1/TDR1 (transmission and reception data register 1)*1	RDR0/TDR0 (transmission and reception data register)*1
	-	-
	RDR1/TDR1 (transmission and reception data register 1)*1	RDR0/TDR0 (transmission and reception data register)*1
	-	-
	RDR1/TDR1 (transmission and reception data register 1)*1	RDR0/TDR0 (transmission and reception data register)*1
	-	-
	RDR1/TDR1 (transmission and reception data register 1)*1	RDR0/TDR0 (transmission and reception data register)*1
	-	-
	RDR1/TDR1 (transmission and reception data register 1)*1	RDR0/TDR0 (transmission and reception data register)*1
	-	-
	RDR1/TDR1 (transmission and reception data register 1)*1	RDR0/TDR0 (transmission and reception data register)*1
	-	-
	RDR1/TDR1 (transmission and reception data register 1)*1	RDR0/TDR0 (transmission and reception data register)*1
	-	-
	RDR1/TDR1 (transmission and reception data register 1)*1	RDR0/TDR0 (transmission and reception data register)*1
	-	-
	RDR1/TDR1 (transmission and reception data register 1)*1	RDR0/TDR0 (transmission and reception data register)*1
	-	-
	RDR1/TDR1 (transmission and reception data register 1)*1	RDR0/TDR0 (transmission and reception data register)*1
	-	-

	Bit15 Bit8	Bit7 Bit0
	RDR1/TDR1 (transmission and reception data register 1)*1	RDR0/TDR0 (transmission and reception data register)*1
	-	-
	RDR1/TDR1 (transmission and reception data register 1)*1	RDR0/TDR0 (transmission and reception data register)*1
	-	-
	RDR1/TDR1 (transmission and reception data register 1)*1	RDR0/TDR0 (transmission and reception data register)*1
	-	-

*1: Access with a mirror address area

Operation Mode

The UART (Asynchronous Serial Interface) can operate in either of 2 modes. The operation mode is determined by MD2, MD1, and MD0 in the serial mode register (SMR).

Table 10-2 Operation Modes of UART (Asynchronous Serial Interface)

Operation Mode	MD2	MD1	MD0	Type
0	0	0	0	UART0 (asynchronous normal mode)
1	0	0	1	UART1 (asynchronous multi-processor mode)

10.1. Serial Control Register (SCR)

The serial control register (SCR) can be used for permission/prohibition of transmission and reception, permission/prohibition of transmission and reception interrupts, permission/prohibition of transmission bus idle interrupts, and UART reset.

Serial Control Register (SCR)

Figure 10-1 shows the bit configuration of the serial control register (SCR).

Figure 10-1 Bit Configuration of Serial Control Register (SCR)

Bit	15	14	13	12	11	10	9	8
Field	UPCL	Reserved		RIE	TIE	TBIE	RXE	TXE
R/W Attribute	R0,W	R0,W0		R/W	R/W	R/W	R/W	R/W
Protection attribute	-							
Initial Value	0	00		0	0	0	0	0

Lower byte [bit7:0] of this register is Serial Mode Register (SMR)

[bit15] UPCL: Programmable Clear Bit

This bit initializes the UART's internal status.

When "1" is set:

- UART is directly reset (software reset). However, the register settings are retained. At this time, transmission and reception communications are immediately disconnected.
- The baud rate generator reloads the setting values for the BGR1/0 registers and restarts.
- All transmission and reception communications and status interrupt factors (SSR:PE, FRE, ORE, RDRF, TDRE, TBI, SACSR:TINT) are initialized to 0b0000110.
- The $\overline{RTS}$ signal is cleared to "L".

When "0" is set:

There is no effect.

When it is read, "0" is always read.

Bit	Description	
	Write	Read
0	No effect	"0" is always read.
1	Programmable clear	

Notes:

- *Execute programmable clear immediately after setting interrupt prohibition.*
- *When FIFO is used, prohibit FIFO (FCR0:FE2, FE1 = 0) before executing programmable clear.*
- *Executing programmable clear (SCR:UPCL = 1) does not clear the value of the serial timer register (STMR).*

[bit14:13] Reserved: Reserved Bits

[bit12] RIE: Reception Interrupt Enable Bit

- This bit enables/disables reception interrupt request output to the CPU.
- When ECR:EISEL = 0, and the RIE bit and the reception data flag bit (SSR:RDRF) are "1" or one of the error flag bits (SSR:PE, ORE, FRE, ESR:RXUDR, and RBERR) is "1", a reception interrupt request is issued.
- When ECR:EISEL = 1, and the RIE bit and the reception data flag bit (SSR:RDRF) are "1", a reception interrupt request is issued.

Bit	Description
0	Disable reception interrupts.
1	Enable reception interrupts.

[bit11] TIE: Transmission Interrupt Enable Bit

- This bit enables/disables transmission interrupt request output to the CPU.
- When ECR:EISEL = 0, and the TIE bit and the SSR:TDRE bit are "1" or one of the error flag bits (ESR:TXOVR and TBERR) is "1", a transmission interrupt request is issued.
- When ECR:EISEL = 1, and the TIE bit and the SSR:TDRE bit are "1", a transmission interrupt request is issued.

Bit	Description
0	Disable transmission interrupts.
1	Enable transmission interrupts.

[bit10] TBIE: Transmission Bus Idle Interrupt Enable Bit

- This bit enables/disables transmission bus idle interrupt request output to the CPU.
- A transmission bus idle interrupt request is issued when the TBIE bit and the TBI bit are "1".

Bit	Description
0	Disable transmission bus idle interrupts.
1	Enable transmission bus idle interrupts.

[bit9] RXE: Reception Operation Enable Bit

This bit enables/disables UART reception operation.

- "0": Disable reception operation.
- "1": Enable reception operation.

Bit	Description
0	Disable reception.
1	Enable reception.

Note:

- *Even if reception operation is enabled (RXE = 1), reception does not start unless there is a falling edge of a start bit (for NRZ format (ESCR:INV = 0)). (For inverted NRZ format (ESCR:INV = 1), reception does not start unless there is a rising edge.)*

- If reception operation is disabled (RXE = 0) during reception, the operation is stopped immediately.
- When reception operation is disabled (RXE = 0), the $\overline{RTS}$ signal is fixed to "L".

[bit8] TXE: Transmission Operation Enable Bit

This bit enables/disables UART transmission operation.

- "0": Disable transmission operation.
- "1": Enable transmission operation.

Bit	Description
0	Disable transmission.
1	Enable transmission.

Note:

- *If transmission operation is disabled (TXE = 0) during transmission, the operation is stopped immediately.*

10.2. Serial Mode Register (SMR)

The serial mode register (SMR) is responsible for the following: 1. Operation mode setting, 2. Selection of the transfer direction, the data length, and the stop bit length, and 3. Settings of enabling/disabling of output on the serial data and clock pins.

Serial Mode Register (SMR)

Figure 10-2 shows the bit configuration of the serial mode register (SMR).

Figure 10-2 Bit Configuration of Serial Mode Register (SMR)

Bit	7	6	5	4	3	2	1	0
Field	MD2	MD1	MD0	WUCR	SBL	BDS	Reserved	SOE
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R0,W0	R/W
Protection attribute	-							
Initial Value	0	0	0	0	0	0	0	0

[bit7:5] MD[2:0]: Operation Mode Setting Bits

These bits set the operation mode of the Asynchronous Serial Interface.

Bits			Description
0	0	0	Operation mode 0 (asynchronous normal mode)
0	0	1	Operation mode 1 (asynchronous multi-processor mode)
0	1	0	Operation mode 2 (clock synchronous mode)
0	1	1	Operation mode 3 (LIN communication mode)
1	0	0	Operation mode 4 (I2C mode)

* This section describes the registers and their operations in operation mode 0 and operation mode 1.

Notes:

- *Settings other than those described above are prohibited.*
- *To switch the operation mode, execute programmable clear (SCR:UPCL = 1) and then switch the operation mode.*
- *Set the registers after setting the operation mode.*

[bit4] WUCR: WAKE UP Control Bit

WUCR is not supported

This bit selects the pin to be used for external interrupts.

"0": Set the INT pin for external interrupts.

"1": Set the SIN pin for external interrupts.

Bit	Description
0	Disable the WAKE UP function.
1	Enable the WAKE UP function.

Note:

- *The WAKE UP function is not supported.*

[bit3] SBL: Stop Bit Length Selection Bit

This bit sets the length of the stop bits (the frame end mark of the transmission data).

Settings for which SBL is "0" and ESCR:ESBL is "0": The stop bit length is set to 1 bit.
Settings for which SBL is "1" and ESCR:ESBL is "0": The stop bit length is set to 2 bits.
Settings for which SBL is "0" and ESCR:ESBL is "1": The stop bit length is set to 3 bits.
Settings for which SBL is "1" and ESCR:ESBL is "1": The stop bit length is set to 4 bits.

Bit	Description	
0	ESCR.ESBL = 0	1 bit
	ESCR.ESBL = 1	3 bits
1	ESCR.ESBL = 0	2 bits
	ESCR.ESBL = 1	4 bits

Notes:

- *Reception always detects only the first stop bit.*
- *Set this bit when transmission is disabled (SCR:TXE = 0).*

[bit2] BDS: Transfer Direction Selection Bit

This bit selects between whether the lowest bit is transferred first (LSB first, BDS = 0) or the highest bit is transferred first (MSB first, BDS = 1), for transfer serial data.

Bit	Description
0	LSB first (lowest bit transferred first)
1	MSB first (highest bit transferred first)

Note:

- *Set this bit when transmission and reception are disabled (SCR:TXE = SCR:RXE = 0).*

[bit1] Reserved: Reserved Bit

[bit0] SOE: Serial Data Output Enable Bit

This bit enables/stops the output of serial data.

Bit	Description
0	Stop the serial data output.
1	Enable the serial data output.

10.3. Serial Status Register (SSR)

The serial status register (SSR) checks the transmission and reception status, checks the reception error flag, and clears the reception error flag.

Serial Status Register (SSR)

Figure 10-3 shows the bit configuration of the serial status register (SSR).

Figure 10-3 Bit Configuration of Serial Status Register (SSR)

Bit	15	14	13	12	11	10	9	8
Field	REC	Reserved	PE	FRE	ORE	RDRF	TDRE	TBI
R/W Attribute	R0,W	R0,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection attribute	-							
Initial Value	0	0	0	0	0	0	1	1

* Lower byte [bit7:0] of this register is Extended Communication Control Register (ESCR)

[bit15] REC: Reception Error Flag Clear Bit

This bit clears the PE, FRE, and ORE flags in the serial status register (SSR).

- Writing "1" clears the error flags.
- Writing "0" does not have any effect.

If it is read, "0" is always read.

Bit	Description	
	Write	Read
0	No effect	"0" is always read.
1	Clear the reception error flags (PE, FRE, and ORE).	

[bit14] Reserved: Reserved Bit

[bit13] PE: Parity Error Flag Bit (Effective Only in Operation Mode 0)

- This bit is set to "1" when a parity error occurs in data reception with ESCR:PEN = 1, and is cleared when writing "1" to the REC bit in the serial status register (SSR).
- A reception interrupt request is issued when the PE bit and the SCR:RIE bit are both "1".
- For details on the output of interrupt requests, see Table 2-3.
- If this flag is set, the reception data register (RDR) data will be invalid.
- If this flag is set when the reception FIFO is used, the reception FIFO enable bits will be cleared, and no reception data will be stored in the reception FIFO.

Bit	Description
0	No parity error
1	Parity error found

Note:

- *This bit is reset by the Software reset (SCR:UPCL=1).*

[bit12] FRE: Framing Error Flag Bit

- This bit is set to "1" when a framing error occurs in data reception, and is cleared when writing "1" to the REC bit in the serial status register (SSR).
- A reception interrupt request is issued when the FRE bit and the SCR:RIE bit are both "1".
- For details on the output of interrupt requests, see Table 2-3.
- If this flag is set, the reception data register (RDR) data will be invalid.
- If this flag is set when the reception FIFO is used, the reception FIFO enable bits will be cleared, and no reception data will be stored in the reception FIFO.

Bit	Description
0	No framing error
1	Framing error found

Note:

- *This bit is reset by the Software reset (SCR:UPCL=1).*

[bit11] ORE: Overrun Error Flag Bit

- This bit is set to "1" when an overrun error occurs in data reception, and is cleared when writing "1" to the REC bit in the serial status register (SSR).
- A reception interrupt request is issued when the ORE bit and the SCR:RIE bit are both "1".
- For details on the output of interrupt requests, see Table 2-3.
- If this flag is set, the reception data register (RDR) data will be invalid.
- If this flag is set when the reception FIFO is used, the reception FIFO enable bits will be cleared, and no reception data will be stored in the reception FIFO.

Bit	Description
0	No overrun error
1	Overrun error found

Note:

- *This bit is reset by the Software reset (SCR:UPCL=1).*

[bit10] RDRF: Reception Data Full Flag Bit

- This flag indicates the status of the reception data register (RDR).
- This flag is set to "1" when reception data is loaded to RDR, and is cleared to "0" when the data in the reception data register (RDR) is read.
- A reception interrupt request is issued when the RDRF bit and the SCR:RIE bit are both "1".
- When the reception FIFO is used, if the reception FIFO receives a prescribed data count, RDRF is set to "1".
- When the reception FIFO is used, if both of the conditions below are satisfied, and the reception idle state continues for at least 8 clock pulses of the baud rate clock, RDRF is set to "1".
  - The reception FIFO idle detection enable bit (FCR1:FRIIE) is "1".
  - The reception FIFO has not received a prescribed data count, and data remains in the reception FIFO.

  During an 8-clock count, the counter is reset to 0 when RDR is read, and the system starts counting the 8 clocks again.

- When ECR:RXBLKEN = 0, and the reception FIFO is used, this bit is cleared to "0" when the reception FIFO becomes empty by reading data from it.
- When ECR:RXBLKEN = 1, and the reception FIFO is used, this bit is cleared to "0" when the data count in the reception FIFO becomes equal to or less than the setting of the FBYTE register.

Bit	Description
0	The reception data register (RDR) is empty.
1	The reception data register RDR contains data.

Notes:

- *When the reception FIFO is used and RDRF has become "1", resetting the reception FIFO (FCR0:FCL2, FCL1 = 1) does not result in RDRF being set to "0". Therefore, in order to set RDRF to "0" after resetting the reception FIFO, perform a dummy read of the reception data register during the reception disable status (SCR:RXE = 0).*
- *This bit is reset by the Software reset (SCR:UPCL=1).*

[bit9] TDRE: Transmission Data Empty Flag Bit

- This flag indicates the status of the transmission data register (TDR).
- Writing transmission data to TDR sets this bit to "0", indicating that TDR contains valid data. Loading the data to the transmission shift register to start transmission sets this bit to "1", indicating that TDR does not contain valid data.
- A transmission interrupt request is issued when the TDRE bit and the SCR:TIE bit are both "1".
- The TDRE bit is set to "1" by the programmable clear (SCR:UPCL=1) is executed.
- For set/reset timing of the TDRE bit when the transmission FIFO is used, see "CHAPTER 36:2.4 Occurrence of Interrupts and Flag Set Timing When the Transmission FIFO Is Used."

Bit	Description
0	The transmission data register (TDR) contains data.
1	The transmission data register is empty.

[bit8] TBI: Transmission Bus Idle Flag Bit

- This bit indicates that the UART is not performing transmission.
- This bit is set to "0" when transmission data is written to the transmission data register (TDR).
- This bit is set to "1" during periods in which the transmission data register is empty (TDRE = 1) and transmission is not being performed.
- The TBI bit is set to "1" by the programmable clear (SCR:UPCL=1) is executed.
- A transmission interrupt request is issued if this bit is set to "1" and a transmission bus idle interrupt is enabled (SCR:TBIE = 1).

Bit	Description
0	Transmission in progress
1	No transmission operation

10.4. Extended Communication Control Register (ESCR)

The extended communication control register (ESCR) sets the transmission and reception data length, enables/disables the parity bits, selects a parity bit, inverts the serial data format, and selects a stop bit length.

Bit Configuration of the Extended Communication Control Register (ESCR)

Figure 10-4 shows the bit configuration of the extended communication control register (ESCR).

Figure 10-4 Bit Configuration of Extended Communication Control Register (ESCR)

Bit	7	6	5	4	3	2	1	0
Field	FLWEN	ESBL	INV	PEN	P	L2	L1	L0
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection attribute	-							
Initial Value	0	0	0	0	0	0	0	0

[bit7] FLWEN: Flow Control Enable Bit

This bit enables or disables hardware flow control.

- "0": Disable hardware flow control.
- "1": Enable hardware flow control.

Bit	Description
0	Disable hardware flow control.
1	Enable hardware flow control.

Notes:

- *Set this bit when transmission and reception are disabled (SCR:TXE = 0, RXE = 0).*
- *Set this bit to "1" only when performing hardware flow control.*

[bit6] ESBL: Extended Stop Bit Length Selection Bit

This bit sets the length of the stop bits (the frame end mark of the transmission data).

Settings for which SBL is "0" and ESCR:ESBL is "0": The stop bit length is set to 1 bit.
Settings for which SBL is "1" and ESCR:ESBL is "0": The stop bit length is set to 2 bits.
Settings for which SBL is "0" and ESCR:ESBL is "1": The stop bit length is set to 3 bits.
Settings for which SBL is "1" and ESCR:ESBL is "1": The stop bit length is set to 4 bits.

Bit	Extended Stop Bit Length	
0	SMR.SBL = 0	1 bit
	SMR.SBL = 1	2 bits
1	SMR.SBL = 0	3 bits
	SMR.SBL = 1	4 bits

Notes:

- *Reception always detects only the first stop bit.*
- *Set this bit when transmission is disabled (SCR:TXE = 0).*

[bit5] INV: Inverted Serial Data Format Bit

This bit selects the NRZ format or the inverted NRZ format as the serial data format.

Bit	Description
0	NRZ format
1	Inverted NRZ format

[bit4] PEN: Parity Enable Bit (Effective Only in Operation Mode 0)

This bit specifies whether to append (in transmission) and detect (in reception) parity bits.

- "0": Parity bits are not appended.
- "1": Parity bits are appended.

Bit	Description
0	Disable parity.
1	Enable parity.

Note:

- *This bit is internally fixed to "0" in operation mode 1.*

[bit3] P: Parity Selection Bit (Effective Only in Operation Mode 0)

This bit selects the odd parity "1" or the even parity "0" when parity check is enabled (ESCR:PEN = 1).

- "0": Set even parity.
- "1": Set odd parity.

Bit	Description
0	Even parity
1	Odd parity

[bit2:0] L[2:0]: Data Length Selection Bits

These bits specify the data length of the transmission/reception data.

- 0b000: The data length is 8 bits.
- 0b001: The data length is 5 bits.
- 0b010: The data length is 6 bits.
- 0b011: The data length is 7 bits.
- 0b100: The data length is 9 bits.

Bits			Data Length
0	0	0	8-bit length
0	0	1	5-bit length
0	1	0	6-bit length
0	1	1	7-bit length
1	0	0	9-bit length

Notes:

- *Values other than the above are prohibited.*
- *Change these bits when transmission and reception are disabled (SCR:TXE = RXE = 0).*
- *Set the data length to 7 or 8 bits for operation mode 1. Other values are prohibited.*

10.5. Reception Data Register/Transmission Data Register (RDR/TDR)

The reception data register and the transmission data register are placed at the same address. If read, the register functions as a reception data register, and if written to, it functions as a transmission register. When FIFO operation is enabled, the RDR/TDR address is the FIFO read/write address.

Reception Data Register (RDR)

Figure 10-5 shows the bit configuration of the serial reception register (RDR).

Figure 10-5 Bit Configuration of Reception Data Register (RDR)

Bit	15	14	13	12	11	10	9	8
Field	Reserved							D8
R/W Attribute	R0							R
Protection attribute	-							
Initial Value	0000000							0

Bit	7	6	5	4	3	2	1	0
Field	D7	D6	D5	D4	D3	D2	D1	D0
R/W Attribute	R	R	R	R	R	R	R	R
Protection attribute	-							
Initial Value	0	0	0	0	0	0	0	0

The reception data register (RDR) is a 9-bit data buffer register for serial data reception.

- Serial data signals transmitted to the serial input pin (SIN pin) are converted by the shift register, and are stored in the reception data register (RDR).
- Higher bits are padded with "0s" depending on the data length, as follows:

Data Length	D8	D7	D6	D5	D4	D3	D2	D1	D0
9 bits	X	X	X	X	X	X	X	X	X
8 bits	0	X	X	X	X	X	X	X	X
7 bits	0	0	X	X	X	X	X	X	X
6 bits	0	0	0	X	X	X	X	X	X
5 bits	0	0	0	0	X	X	X	X	X

(X represents a reception data bit.)

- The reception data full flag bit (SSR:RDRF) is set to "1" when reception data is stored in the reception data register (RDR). A reception interrupt request occurs if the reception interrupt is enabled (SCR:RIE = 1).
- Read the reception data register (RDR) when the reception data full flag bit (SSR:RDRF) is "1". The reception data full flag bit (SSR:RDRF) is automatically cleared to "0" when reading data from the reception data register (RDR).
- If a reception error occurs (when SSR:PE, ORE, or FRE becomes "1"), the data in the reception data register (RDR) will be invalid.
- In operation mode 1 (multi-processor mode), the data length is 7 or 8 bits, and the received AD bit is stored in the D8 bit.
- In the case of 9-bit transfer or in operation mode 1, RDR is read in 16-bit access.

Notes:

- *When the reception FIFO is used, SSR:RDRF is set to "1" when a predefined amount of data is received by the reception FIFO.*
- *When the reception FIFO is used, SSR:RDRF is cleared to "0" when the reception FIFO becomes empty.*
- *When a reception error occurs (when SSR:PE, ORE, or FRE becomes "1") during use of the reception FIFO, the enable bit in the reception FIFO is cleared and the reception data is not stored in the reception FIFO.*

Transmission Data Register (TDR)

Figure 10-6 shows the bit configuration of the transmission data register.

Figure 10-6 Bit Configuration of Transmission Data Register (TDR)

Bit	15	14	13	12	11	10	9	8
Field	Reserved							D8
R/W Attribute	WX							W
Protection attribute	-							
Initial Value	0000000							1

Bit	7	6	5	4	3	2	1	0
Field	D7	D6	D5	D4	D3	D2	D1	D0
R/W Attribute	W	W	W	W	W	W	W	W
Protection attribute	-							
Initial Value	1	1	1	1	1	1	1	1

The transmission data register (TDR) is a 9-bit data buffer register for serial data transmission.

- When data to be transmitted is written to the transmission data register (TDR) while transmission is enabled (SCR:TXE = 1), the transmission data is transferred to the transmission shift register. The transmission data is converted to serial data and sent out from the serial data output pin (SOUT pin).
- Bits are invalidated in descending order depending on the data length, as follows:

Data Length	D8	D7	D6	D5	D4	D3	D2	D1	D0
9 bits	X	X	X	X	X	X	X	X	X
8 bits	Invalid	X	X	X	X	X	X	X	X
7 bits	Invalid	Invalid	X	X	X	X	X	X	X
6 bits	Invalid	Invalid	Invalid	X	X	X	X	X	X
5 bits	Invalid	Invalid	Invalid	Invalid	X	X	X	X	X

- The transmission data empty flag (SSR:TDRE) is cleared to "0" when transmission data is written to the transmission data register (TDR).
- If the transmission FIFO is disabled or empty, the transmission data empty flag (SSR:TDRE) is set to "1" when transmission data is transferred to the transmission shift register to start transmission.
- When the transmission data empty flag (SSR:TDRE) is "1", transmission data can be written. A transmission interrupt occurs if the transmission interrupt is enabled. Write transmission data after the occurrence of a transmission interrupt or when the transmission data empty flag (SSR:TDRE) is "1".
- When the transmission data empty flag (SSR:TDRE) is "0" and the transmission FIFO is disabled or full, transmission data cannot be written.
- In operation mode 1 (multi-processor mode), the data length is 7 or 8 bits, and the AD bit is transmitted by writing it to the D8 bit.
- In the case of 9-bit transfer or in operation mode 1, write data to TDR by 16-bit access.

Notes:

- *The transmission data register is a write-only register, and the reception data register is a read-only register. Because the transmission and reception registers are allocated at the same address, the write value and the read value are different.*

- *For the set timing for the transmission data empty flag (SSR:TDRE) when the transmission FIFO is used, see "CHAPTER 36:2.4 Occurrence of Interrupts and Flag Set Timing When the Transmission FIFO Is Used."*

10.6. Serial Auxiliary Control Status Register (SACSR)

The serial auxiliary control status register (SACSR) performs the following: 1. Controlling serial test operation, 2. selecting a serial timer activation method, 3. enabling/disabling timer interrupts, 4. enabling/disabling synchronous transmission, 5. setting division value of the serial timer operation clock, and 6. enabling/disabling the serial timer.

Bit Configuration of the Serial Auxiliary Control Status Register (SACSR)

Figure 10-7 shows the bit configuration of the serial auxiliary control status register (SACSR).

Figure 10-7 Bit Configuration of Serial Auxiliary Control Status Register (SACSR)

Bit	15	14	13	12	11	10	9	8
Field	STST	Reserved				TRG1	TRG0	TINT
R/W Attribute	R/W	RX,WX				R/W	R/W	R
Protection attribute	-							
Initial Value	0	0000				0	0	0

Bit	7	6	5	4	3	2	1	0
Field	TINTE	TSYNE	TRGE	TDIV3	TDIV2	TDIV1	TDIV0	TMRE
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection attribute	-							
Initial Value	0	0	0	0	0	0	0	0

[bit15] STST: Serial Test Bit

This bit enables or disables serial test mode.

When serial test mode is enabled, SOUT and SIN are connected together inside the multi-function serial interface, so that the data transmitted from SOUT can be received from SIN directly.

When serial test mode is enabled, the SOUT pin is fixed to "H", and any data input to the SIN pin is ignored.

- This bit is reset when the SACSRC:STSTC bit in the clear register is set to "1".
- This bit is set when the SACSRS:STSTS bit in the set register is set to "1".

Bit	Serial Test Bit
0	Disable the serial test mode.
1	Enable the serial test mode.

Note:

- *This bit can be changed only when transmission/reception are disabled (SCR:TXE = 0, SCR:RXE = 0).*

[bit14:11] Reserved: Reserved Bits

[bit10:9] TRG[1:0]: Trigger Selection Bits

These bits select the method for detecting external trigger edges for activating the serial timer.

Bits		Edge Detection Method for External Trigger
0	0	Rising edge detection
0	1	Falling edge detection
1	0	Both edge detection
1	1	Setting prohibited

Notes:

- *These bits are invalid if the external trigger enable bit (TRGE) is "0".*
- *These bits can be changed only when the serial timer enable bit (TMRE) is "0".*

[bit8] TINT: Timer Interrupt Flag

If the serial timer register (STMR) and the serial timer comparison register (STMCR) match, the serial timer register (STMR) is set to "0", and this bit is set to "1".

When this bit is "1" and the timer interrupt enable bit (TINTE) is "1", a status interrupt request is issued.

Writing to this bit is invalid.

- This bit is reset when the SACSRC:TINTC bit in the clear register is set to "1".

Bit	Description
0	No timer interrupt request
1	Timer interrupt request issued

Notes:

- *Software reset (SCR:UPCL = 1) resets this bit to "0".*
- *This bit is not set to "1" when the synchronous transmission enable bit (TSYNE) is "1".*

[bit7] TINTE: Timer Interrupt Enable Bit

This bit enables/disables timer interrupts to the CPU.

When this bit is "1" and the timer interrupt flag (TINT) is "1", a status interrupt request is issued.

- This bit is reset when the SACSRC:TINTEC bit in the clear register is set to "1".
- This bit is set when the SACSRS:TINTES bit in the set register is set to "1".

Bit	Description
0	Disable interrupts triggered by the serial timer.
1	Enable interrupts triggered by the serial timer.

[bit6] TSYNE: Synchronous Transmission Enable Bit

This bit enables or disables synchronous transmission.

Transmission is activated if this bit is "1" and in any of the cases below.

- The serial timer register (STMR) and the serial timer comparison register (STMCR) coincide at the time of timer-synchronized transmission.
- An external trigger edge specified by the trigger selection bits (SACSR:TRG1, 0) is detected at the time of external-trigger-synchronized transmission.
- This bit is reset when the SACSRC:TSYNEC bit in the clear register is set to "1".
- This bit is set when the SACSRC:TSYNEC bit in the set register is set to "1".

Bit	Description
0	Disable synchronous transmission. The serial timer is used as a timer.
1	Enable synchronous transmission. The serial timer is not used as a timer.

Notes:

- *This bit can be changed only when the serial timer enable bit (TMRE) is "0".*
- *When synchronous transmission is enabled (TSYNE = 1), if transmission is disabled (SCR:TXE = 0), transmission is not activated even in any of the following cases.*
  - The serial timer register (STMR) and the serial timer comparison register (STMCR) coincide at the time of timer-synchronized transmission.
  - An external trigger edge specified by the trigger selection bits (SACSR:TRG1, 0) is detected at the time of external-trigger-synchronized transmission.

[bit5] TRGE: External Trigger Enable Bit

This bit selects a way to activate the serial timer.

- This bit is reset when the SACSRC:TRGEC bit in the clear register is set to "1".
- This bit is set when the SACSRS:TRGES bit in the set register is set to "1".

Bit	Description
0	The serial timer starts when the serial timer enable bit (TMRE) is changed from "0" to "1".
1	If the serial timer enable bit (TMRE) is set to "1", the detection of an external trigger edge specified by the trigger selection bits (TRG1, 0) starts the serial timer.

Notes:

- *This bit can be changed only when the serial timer enable bit (TMRE) is "0".*
- *If the serial timer enable bit (TMRE) is set to "0", the detection of an external trigger edge specified by the trigger selection bits (TRG1, 0) does not start the serial timer.*
- *When synchronous transmission is enabled (SACSR:TSYNE = 1), external triggers are enabled (SACSR:TRGE = 1), the timer comparison register (STMCR) is set to 0x0000, and the serial timer is enabled (SACSR:TMRE = 1), transmission is activated in synchronization with an external trigger.*

[bit4:1] TDIV3, TDIV2, TDIV1, TDIV0: Timer Operation Clock Division Bits

These bits set the division ratio of the serial timer.

TDIV3	TDIV2	TDIV1	TDIV0	Timer Operation Clock						
				Division Ratio	Φ= 8 MHz	Φ= 10 MHz	Φ= 16 MHz	Φ= 20 MHz	Φ= 24 MHz	Φ= 32 MHz
0	0	0	0	Φ	125 ns	100 ns	62.5 ns	50 ns	41.67 ns	31.25 ns
0	0	0	1	Φ/2	250 ns	200 ns	125 ns	100 ns	83.33 ns	62.5 ns
0	0	1	0	Φ/4	500 ns	400 ns	250 ns	200 ns	166.67 ns	125 ns
0	0	1	1	Φ/8	1 us	800 ns	500 ns	400 ns	333.33 ns	250 ns
0	1	0	0	Φ/16	2 us	1.6 us	1 us	800 ns	666.67 ns	500 ns
0	1	0	1	Φ/32	4 us	3.2 us	2 us	1.6 us	1.33 us	1 us
0	1	1	0	Φ/64	8 us	6.4 us	4 us	3.2 us	2.67 us	2 us
0	1	1	1	Φ/128	16 us	12.8 us	8 us	6.4 us	5.33 us	4 us
1	0	0	0	Φ/256	32 us	25.6 us	16 us	12.8 us	10.67 us	8 us

Φ: Bus clock

Notes:

- *These bits can be changed only when the serial timer enable bit (TMRE) is "0".*
- *Settings other than the above are prohibited.*

[bit0] TMRE: Serial Timer Enable Bit

This bit enables or disables the serial timer operation.

- This bit is reset when the SACSRC:TMREC bit in the clear register is set to "1".
- This bit is set when the SACSRS:TMRES bit in the set register is set to "1".

Bit	Description
0	Stop the serial timer operation. When stopped, the value of the serial timer register (STMR) is retained.
1	Changing this bit from "0" to "1" initializes the serial timer register (STMR) to "0" and starts the operation of the serial timer.

Notes:

- *If the external trigger is enabled (TRGE = 1), setting this bit to "1" does not start the serial timer until the detection of an external trigger edge as specified by the trigger selection bits (SACSR:TRG1, 0).*
- *For serial timer synchronous transmission or external trigger transmission, change this bit from "0" to "1" if transmission is disabled.*

10.7. Serial Timer Register (STMR)

The serial timer register (STMR) represents the timer value of the serial timer.

Bit Configuration of the Serial Timer Register (STMR)

Figure 10-8 shows the bit configuration of the serial timer register (STMR).

Figure 10-8 Bit Configuration of Serial Timer Register (STMR)

Bit	15	14	13	12	11	10	9	8
Field	TM15	TM14	TM13	TM12	TM11	TM10	TM9	TM8
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	TM7	TM6	TM5	TM4	TM3	TM2	TM1	TM0
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection attribute	-							
Initial Value	0	0	0	0	0	0	0	0

[bit15:0] TM15-0: Timer Data Bits

These bits indicate the timer value of the serial timer.

During timer operation, the timer value of the serial timer is incremented by 1 for each timer operation clock (specified by SACSR:TDIV3 to TDIV0).

Note:

- *These bits are initialized to "0" when the timer operation starts.*

10.8. Serial Timer Comparison Register (STMCR)

The serial timer comparison register (STMCR) sets the timer comparison value for the serial timer.

Bit Configuration of the Serial Timer Comparison Register (STMCR)

Figure 10-9 shows the bit configuration of the serial timer comparison register (STMCR).

Figure 10-9 Bit Configuration of Serial Timer Comparison Register (STMCR)

Bit	15	14	13	12	11	10	9	8
Field	TC15	TC14	TC13	TC12	TC11	TC10	TC9	TC8
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	TC7	TC6	TC5	TC4	TC3	TC2	TC1	TC0
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection attribute	-							
Initial Value	0	0	0	0	0	0	0	0

[bit15:0] TC15-0: Compare Bits

These bits set a comparison value for the serial timer.

These bits are compared with the serial timer register (STMR). If the serial timer register (STMR) coincides with these bits when the serial timer register is updated, the serial timer register is set to "0". At that time, if synchronous transmission is disabled (SACSR:TSYNE = 0), the timer interrupt flag (SACSR:TINT) is set to "1". If synchronous transmission is enabled (SACSR:TSYNE = 1), transmission is activated.

The interval of the operations below is (STMCR:TC + 1) * timer operation clock (specified by SACSR:TDIV3 to TDIV0).

- SACSR:TINT is set to "1".
- Serial-timer-synchronized transmission is activated.

Notes:

- *The timer interrupt flag (SACSR:TINT) is fixed to "1" when all of the following conditions are satisfied.*
  - Synchronous transmission is disabled (SACSR:TSYNE = 0).
  - 0x0000 is set in this register.
  - Timer operating
  - Set the timer operating clock division value (SACSR:TDIV[3:0]) to 0b0000.
- *When all of the following conditions are satisfied, transmission starts immediately upon the detection of an edge of the external trigger specified by the trigger selection bits (SACSR:TRG1, 0).*

- Synchronous transmission is enabled (SACSR:TSYNE = 1).
- External trigger is enabled (SACSR:TRGE = 1).
- 0x0000 is set in this register.
- Transmission is enabled (SCR:TXE = 1), and transmission data exists (SSR:TDRE = 0).

- *This register can be changed only when the serial timer is disabled (SACSR:TMRE = 0).*

10.9. Transfer Byte Register (TBYTE0)

The transfer byte (TBYTE0) represents the transfer data amount when synchronous transmission or external trigger transmission starts.

Bit Configuration of the Transfer Byte Register (TBYTE0)

Figure 10-10 shows the bit configuration of the transfer byte register (TBYTE0).

Figure 10-10 Bit Configuration of Transfer Byte Register (TBYTE0)

Bit	7	6	5	4	3	2	1	0
Field	TBYTE0							
R/W Attribute	R/W							
Protection attribute	-							
Initial Value	00000000							

[bit7:0] TBYTE0: Transfer Byte Register

The transfer byte register 0 (TBYTE0) is used for synchronous transmission or external trigger transmission. The data amount set in TBYTE0 is transferred when synchronous transmission or external trigger transmission starts.

TBYTE0	Description
Write	Writing to TBYTE0
Read	Setting of TBYTE0

Note:

- *If one of these bits is set to 0x00, the transfer count is 8.*

10.10. Baud Rate Generator Register 1, 0 (BGR1, BGR0)

The baud rate generator register 1, 0 (BGR1, BGR0) set the division ratio of the serial clock. An external clock can also be selected as the clock source of the reload counter.

Bit Configuration of the Baud Rate Generator Register 1, 0 (BGR1, BGR0)

Figure 10-11 shows the bit configuration of baud rate generator register 1, 0 (BGR1, BGR0).

Figure 10-11 Bit Configuration of Baud Rate Generator Register 1, 0 (BGR1, BGR0)

Bit	15	14	13	12	11	10	9	8
Field	EXT	BGR1						
R/W Attribute	R/W	R/W						
Protection attribute	-							
Initial Value	0	0000000						

Bit	7	6	5	4	3	2	1	0
Field	BGR0							
R/W Attribute	R/W							
Protection attribute	-							
Initial Value	00000000							

- The baud rate generator registers set the division ratio of the serial clock.
- BGR1 represents the higher bits while BGR0 represents the lower bits. The reload value for counting can be written to these registers, while the currently set reload values can be read from these registers.
- The reload counter starts counting when a reload value is written to baud rate generator register 0, 1 (BGR1, BGR0).
- The EXT bit, bit15, selects the internal clock or an external clock as the clock source of the reload counter. EXT = 0 selects the internal clock. EXT = 1 selects an external clock.

[bit15] EXT: External Clock Selection Bit

Bit	Description
0	Use the internal clock.
1	Use an external clock.

[bit14:8] BGR1: Baud Rate Generator Register 1

BGR1	Description
Write	Write to reload counter bits 8 to 14.
Read	Read the setting value of BGR1.

[bit7:0] BGR0: Baud Rate Generator Register 0

BGR0	Description
Write	Write to reload counter bits 0 to 7.
Read	Read the setting value of BGR0.

Notes:

- *Use 16-bit access for writing a value to the baud rate generator register (BGR1, BGR0).*
- *If the setting values of the baud rate generator registers (BGR1, BGR0) are changed, the new values are not reloaded until the counter value becomes 0x0000. Therefore, in order to enable the new value immediately, execute programmable clear (UPCL) after changing the values of BGR1/0.*
- *When the reload value is even, the "L" width of the reception serial clock is 1-bus-clock-cycle longer than the "H" width. When the reload value is odd, the "L" width and the "H" width of the serial clock are equal.*
- *Set a reload value of 4 or larger in BGR1/0. However, data may not be received normally depending on baud rate errors and the reload value setting.*
- *To change the clock setting to that of the external clock (EXT = 1) when the baud rate generator is operating, perform the following: 1. Write "0" to baud rate generators 1, 0 (BGR1, BGR0). 2. Execute programmable clear (UPCL). 3. Set the external clock mode (EXT = 1).*

10.11. FIFO Control Register 1 (FCR1)

FIFO control register 1 (FCR1) sets the FIFO test, selects transmission and reception FIFOs, enables transmission FIFO interrupts, and controls the interrupt flag.

Bit Configuration of FIFO Control Register 1 (FCR1)

Figure 10-12 shows the bit configuration of FIFO control register 1 (FCR1).

Figure 10-12 Bit Configuration of FIFO Control Register 1 (FCR1)

Bit	15	14	13	12	11	10	9	8
Field	Reserved			FLSTE	FRIIE	FDRQ	FTIE	FSEL
R/W Attribute	R0,W0			R/W	R/W	R	R/W	R/W
Protection attribute	-							
Initial Value	000			0	0	1	0	0

*: Lower byte of this register [bit7:0] is FIFO Control Register 0 (FCR0)

[bit15:13] Reserved: Reserved Bits

[bit12] FLSTE: Retransmission Data Lost Detection Enable Bit

This bit enables the detection of the FIFO retransmission data lost flag (FLST).

- This bit is reset when the FCR1C:FLSTEC bit in the clear register is set to "1".
- This bit is set when the FCR1S:FLSTES bit in the set register is set to "1".

"0": Disable FLST bit detection.
"1": Enable FLST bit detection.

Bit	Description
0	Disable data lost detection.
1	Enable data lost detection.

Note:

- *To set this bit to "1", set the FSET bit to "1" first and then set this bit to "1".*

[bit11] FRIIE: Reception FIFO Idle Detection Enable Bit

When the reception FIFO contains valid data, this bit enables or disables the detection of continuation of reception idle status for the 8-bit time or longer. If the reception interrupt is enabled (SCR:RIE = 1), detection of reception idle status triggers a reception interrupt.

- This bit is reset when the FCR1C:FRIIEC bit in the clear register is set to "1".
- This bit is set when the FCR1S:FRIIES bit in the set register is set to "1".

"0": Disable detection of reception idle status.
"1": Enable detection of reception idle status.

Bit	Description
0	Disable detection of reception FIFO idle.
1	Enable detection of reception FIFO idle.

Note:

- *To use the reception FIFO, set this bit to "1".*

[bit10] FDRQ: Transmission FIFO Data Request Bit

This bit requests transmission FIFO data.

Value "1" of this bit indicates that transmission data is requested. At that time, a FIFO transmission interrupt request is issued if the transmission FIFO interrupt is enabled (FTIE = 1).

FDRQ set conditions

- When transmission FIFO interrupt control is not used
  - FBYTE (for transmission) = 0 (Transmission FIFO is empty.)
  - Reset of the transmission FIFO
- When transmission FIFO interrupt control is used
  - When ECR:TXBLKEN = 0
    - FTICR setting value ≥ FTICR read value (The amount of data in the transmission FIFO at the interrupt trigger level or lower.)
  - When ECR:TXBLKEN = 1
    - FTICR setting value ≤ Free space in the transmission FIFO
  - Reset of the transmission FIFO

FDRQ reset conditions

- "1" is written to FCR1C:FDRQC.
- When ECR:TXBLKEN = 0
  - The transmission FIFO is full.
- When ECR:TXBLKEN = 1
    - FTICR setting value ≥ Free space in the transmission FIFO

Bit	Description
0	No transmission FIFO data request
1	Transmission FIFO data request issued

Notes:

- *Writing to this bit is invalid.*
- *The FSEL bit cannot be changed when this bit is "0".*
- *When you write the necessary data to the transmission FIFO after a transmission interrupt occurs, write "1" to the FIFO transmission data request clear bit (FCR1C:FDRQC) to clear the interrupt request.*

[bit9] FTIE: Transmission FIFO Interrupt Enable Bit

This bit enables a transmission FIFO interrupt. If this bit is set to "1", an interrupt occurs when the FDRQ bit is "1".

- This bit is reset when the FCR1C:FTIEC bit in the clear register is set to "1".
- This bit is set when the FCR1S:FTIES bit in the set register is set to "1".

Bit	Description
0	Disable transmission FIFO interrupts.
1	Enable transmission FIFO interrupts.

Note:

- *Set FTIE=0 if FIFO is not used when block transfer.*

[bit8] FSEL: FIFO Selection Bit

This bit selects the transmission and reception FIFOs.

- This bit is reset when the FCR1C:FSELC bit in the clear register is set to "1".
- This bit is set when the FCR1S:FSELS bit in the set register is set to "1".

"0": Allocation is made as a transmission FIFO: FIFO1 and reception FIFO: FIFO2.
"1": Allocation is made as a transmission FIFO: FIFO2 and reception FIFO: FIFO1.

Bit	Description
0	Transmission FIFO: FIFO1, reception FIFO: FIFO2
1	Transmission FIFO: FIFO2, reception FIFO: FIFO1

Notes:

- *This bit is not cleared by a FIFO reset (FCR0:FCL2, FCL1 = 1).*
- *To change the value of this bit, disable the FIFO operation (FCR0:FE2, FE1 = 0) first.*
- *This bit cannot be changed when FDRQ = 0.*
- *Set this bit before setting the FIFO byte register(FBYTE) and the transmission FIFO interrupt control register(FTICR).*
- *This bit and the FIFO byte register(FBYTE) cannot be accessed at the same time.*

10.12. FIFO Control Register 0 (FCR0)

FIFO control register 0 (FCR0) enables/disables the FIFO operation, resets FIFO, saves the read pointer, and makes the retransmission setting.

Bit Configuration of FIFO Control Register 0 (FCR0)

Figure 10-13 shows the bit configuration of FIFO control register 0 (FCR0).

Figure 10-13 Bit Configuration of FIFO Control Register 0 (FCR0)

Bit	7	6	5	4	3	2	1	0
Field	Reserved	FLST	FLD	FSET	FCL2	FCL1	FE2	FE1
R/W Attribute	R0,W0	R,WX	R,W	R0,W	R0,W	R0,W	R/W	R/W
Protection attribute	-							
Initial Value	0	0	0	0	0	0	0	0

[bit7] Reserved: Reserved Bit

[bit6] FLST: FIFO Retransmission Data Lost Flag Bit

This bit indicates that the retransmission data of the transmission FIFO has been lost.

FLST set condition

- Data is written in the FIFO (overwriting the contents) when the FLSTE bit in the FIFO control register 1 (FCR1) is "1" and the write pointer of the transmission FIFO and the read pointer saved by the FSET bit coincide.

FLST reset conditions

- The transmision FIFO reset
  - set "1" to FCR0:FCL1 when FCR1:FSEL=0
  - set "1" to FCR0:FCL2 when FCR1:FSEL=1
- Writing "1" to the FSET bit

If "1" is set in this bit, the data pointed to by the read pointer saved with the FSET bit is overwritten. For this reason, retransmission with the FLD bit cannot be set even if an error occurs. When performing retransmission with this bit set to "1", resets the FIFO and then writes the data to the FIFO again.

Bit	Description
0	Data has not been lost.
1	Data has been lost.

[bit5] FLD: FIFO Pointer Reload Bit

This bit reloads, to the read pointer, the data saved to the transmission FIFO by the FSET bit. This bit is used for retransmission in cases such as communication errors.
This bit is set to "0" when the retransmission setting has been completed.

■ This bit is set when the FCR0S:FLDS bit in the set register is set to "1".

Bit	Description
0	Do not execute reload.
1	Execute reload.

Notes:

- *While this bit is set to "1", reloading to the read pointer is in progress, so writing is prohibited except in the case of a FIFO reset.*
- *Setting this bit to "1" is prohibited when FIFO is enabled or transmission is in progress.*
- *To set the TIE and TBIE bits to "1", first set them to "0" and write "1" to this bit, and then enable the transmission FIFO and set the SCR:TIE and SCR:TBIE bits to "1".*

[bit4] FSET: FIFO Pointer Saving Bit

This bit stores the read pointer for the transmission FIFO.
If the read pointer is saved before transmission starts and then a communication error occurs, retransmission is possible if the FLST bit is "0".

■ This bit is set when the FCR0S:FSETS bit in the set register is set to "1".

"1": Save the current read pointer value.
"0": No effect.

Bit	Description	
	Write	Read
0	Do not save.	"0" is always read.
1	Save the read pointer value.	

Note:

- *Set this bit to "1" when the transmission byte count (FBYTE) is 0.*

[bit3] FCL2: FIFO2 Reset Bit

This bit resets FIFO2.

If this bit is set to "1", the internal status of FIFO2 is initialized.

Only the FCR0:FLST bit is initialized and the values of the other bits in the FCR1/0 registers are retained.

- This bit is set when the FCR0S:FCL2S bit in the set register is set to "1".

Bit	Description	
	Write	Read
0	No effect	"0" is always read.
1	Reset FIFO2.	

Notes:

- *Disable transmission/reception first and then execute a FIFO2 reset.*
- *Execute the reset after setting the transmission FIFO interrupt enable bit to "0".*
- *The valid data count of the FBYTE2 register is set to 0.*
- *The TDR register and the RDR register are not initialized.*

[bit2] FCL1: FIFO1 Reset Bit

This bit resets FIFO1.

If this bit is set to "1", the internal status of FIFO1 is initialized.

Only the FCR0:FLST bit is initialized and the values of the other bits in the FCR1/0 registers are retained.

- This bit is set when the FCR0S:FCL1S bit in the set register is set to "1".

Bit	Description	
	Write	Read
0	No effect	"0" is always read.
1	Reset FIFO1.	

Notes:

- *Disable transmission/reception first and then execute a FIFO1 reset.*
- *Execute the reset after setting the transmission FIFO interrupt enable bit to "0".*
- *The valid data count of the FBYTE1 register is set to 0.*
- *The TDR register and the RDR register are not initialized.*

[bit1] FE2: FIFO2 Operation Enable Bit

This bit enables or disables the operation of FIFO2.

- Set this bit to "1" when using FIFO2.
- When FIFO2 is specified as the transmission FIFO (FCR1:FSEL = 1) and "1" is written to this bit, transmission starts immediately if FIFO2 contains data and UART transmission is enabled (SCR:TXE = 1). At that time, to set the SCR:TIE and SCR:TBIE bits to "1", first set them to "0", write "1" to this bit, and then set the SCR:TIE and SCR:TBIE bits to "1".

- A reception error clears this bit to "0" if the FIFO is selected as the reception FIFO by the FSEL bit. After that, this bit cannot be set to "1" unless the reception error is cleared.
- To use FIFO2 as the transmission or reception FIFO, set this bit to "1" or "0" when the transmission or reception buffer is empty ((SSR:TDRE = 1) or (SSR:RDRF = 0)), respectively.
- To use FIFO2 as the reception FIFO, first disable reception (SCR:RXE = 0) and then change the setting of this bit when the reception buffer is empty (SSR:RDRF = 0) and the reception FIFO does not contain valid data (FBYTE2 = 0).
- To use FIFO2 as the reception FIFO, first disable reception (SCR:RXE = 0) and then set this bit to "1" when the reception buffer is empty (SSR:RDRF = 0).
- Disabling FIFO2 does not change the state of FIFO2.
- This bit is reset when the FCR0C:FE2C bit in the clear register is set to "1".
- This bit is set when the FCR0S:FE2S bit in the set register is set to "1".

Bit	Description
0	Disable FIFO2 operation.
1	Enable FIFO2 operation.

[bit0] FE1: FIFO1 Operation Enable Bit

This bit enables or disables the operation of FIFO1.

- Set this bit to "1" when using FIFO1.
- When FIFO1 is specified as the transmission FIFO (FCR1:FSEL = 0) and "1" is written to this bit, transmission starts immediately if FIFO1 contains data and UART transmission is enabled (SCR:TXE = 1). At that time, to set the SCR:TIE and SCR:TBIE bits to "1", first set them to "0", write "1" to this bit, and then set the TIE and SCR:TBIE bits to "1".
- A reception error clears this bit to "0" if the FIFO is selected as the reception FIFO by the FSEL bit. After that, this bit cannot be set to "1" unless the reception error is cleared.
- To use FIFO1 as the transmission or reception FIFO, set this bit to "1" or "0" when the transmission or reception buffer is empty ((SSR:TDRE = 1) or (SSR:RDRF = 0)), respectively.
- To use FIFO1 as the reception FIFO, first disable reception (SCR:RXE = 0) and then change the setting of this bit when the reception buffer is empty (SSR:RDRF = 0).
- Disabling FIFO1 does not change the state of FIFO1.
- This bit is reset when the FCR0C:FE1C bit in the clear register is set to "1".
- This bit is set when the FCR0S:FE1S bit in the set register is set to "1".

Bit	Description
0	Disable FIFO1 operation.
1	Enable FIFO1 operation.

10.13. FIFO Byte Register (FBYTE)

The FIFO byte register (FBYTE) indicates the valid data count of the FIFO. This register also specifies whether a reception interrupt is generated when the predefined amount of data is received by the reception FIFO.

Bit Configuration of the FIFO Byte Register (FBYTE)

Figure 10-14 shows the bit configuration of the FIFO byte register (FBYTE).

Figure 10-14 Bit Configuration of FIFO Byte Register (FBYTE)

Bit	15	14	13	12	11	10	9	8
Field	FBYTE2							
R/W Attribute	R,W							
Protection attribute	-							
Initial Value	00000000							

Bit	7	6	5	4	3	2	1	0
Field	FBYTE1							
R/W Attribute	R,W							
Protection attribute	-							
Initial Value	00000000							

The FBYTE register indicates the valid count for the data that has been written or received by the FIFO. Settings with the FCR1:FSEL bit are listed below.

Table 10-3 Data Count Indication

FSEL	FIFO Selection	Data Count Indication
0	FIFO2: Reception FIFO, FIFO1: Transmission FIFO	FIFO2:FBYTE2, FIFO1:FBYTE1
1	FIFO2: Transmission FIFO, FIFO1: Reception FIFO	FIFO2:FBYTE2, FIFO1:FBYTE1

- The initial transfer count value of the FBYTE register is 0x08.
- The FBYTE for the reception FIFO contains the data count for setting the reception interrupt flag. When the transferred data count and the data count indication in the FBYTE register coincide, the interrupt flag (SSR:RDRF) is set to "1".
- When both of the following conditions are satisfied, the continuation of reception idle status for 8 baud rate clocks or longer sets the interrupt flag (RDRF) to "1".
  - The reception FIFO idle detection enable bit (FRIIE) is "1".
  - The number of data items in the reception FIFO does not reach the transfer count.

  During an 8-clock count, the counter is reset to 0 when RDR is read, and the system then starts counting the 8 clocks again. The counter is reset to 0 when the reception FIFO is disabled. If the reception FIFO is enabled when data remains in the reception FIFO, the counting restarts.
- Whenever transmission data is written to TDR, the FBYTE of the transmission FIFO is incremented by 1.
- Whenever reception data is read from RDR, the FBYTE of the reception FIFO is decremented by 1.
- During block transfer, the value in this register is the block transfer count.

[bit15:8] FBYTE2: FIFO2 Data Count Indication byte
[bit7:0] FBYTE1: FIFO1 Data Count Indication Byte

FBYTE2, FBYTE1	Description
Write	Set the transfer count.
Read	Read valid data count.

Read (valid data count)

Transmission: Data count written in the FIFO and not yet transmitted
Reception: Data count that has been received but not read by the reception FIFO

Write (transfer count)

Transmission: Set 0x00.
Reception: Set the data count that triggers reception interrupts.

Table 10-4 Data Count Stored in FIFO

FIFO Capacity	Operation Mode	Data Length	Max FBYTE Count	Data Count That Can Be Stored
16 bytes	Mode 0	5 to 8 bits	16	16
	Mode 0	9 bits	8	8
	Mode 1	All		
32 bytes	Mode 0	5 to 8 bits	32	32
	Mode 0	9 bits	16	16
	Mode 1	All		
64 bytes	Mode 0	5 to 8 bits	64	64
	Mode 0	9 bits	32	32
	Mode 1	All		

Notes:

- *Set 0x00 in the FBYTE register for the transmission FIFO.*
- *When the reception FIFO is not used, the reception block size during block transfer will be "1" regardless of the setting of this register.*
- *Change the value of the FBYTE of the reception FIFO after disabling the reception operation.*
- *Any setting that exceeds the FIFO capacity is prohibited.*
- *Set the FIFO byte register (FBYTE) after setting the FIFO selection bit (FCR1:FSEL).*
- *The FIFO selection bit (FCR1:FSEL) and the FIFO byte register (FBYTE) cannot be set at the same time.*
- *As the FIFO data count for transmission, the transmission data count that has been written, minus 1, is displayed as the valid data count. This is because an attempt to write transmission data to the TDR register stores the data in the transmission FIFO if the TDR register contains data that has not been transmitted yet. When the data in the TDR register is transmitted, the data in the transmission FIFO that has not been transmitted is transferred to the TDR register.*
- *The FIFO data count for reception represents the data count received by the reception FIFO but which has not yet been read. The data count does not include the data being received by the RDR register.*

10.14. Transmission FIFO Interrupt Control Register (FTICR)

The transmission FIFO interrupt control register (FTICR) sets the condition for interrupts triggered by the valid data count of the FIFO transmission.

Transmission FIFO Interrupt Control Register (FTICR)

Figure 10-15 shows the bit configuration of the transmission FIFO interrupt control register (FTICR).

Figure 10-15 Bit Configuration of Transmission FIFO Interrupt Control Register (FTICR)

Bit	15	14	13	12	11	10	9	8
Field	FTICR2							
R/W Attribute	R,W							
Protection attribute	-							
Initial Value	00000000							

Bit	7	6	5	4	3	2	1	0
Field	FTICR1							
R/W Attribute	R,W							
Protection attribute	-							
Initial Value	00000000							

These bits have different functions depending on the setting of the ECR:TXBLKEN bit. The details are as shown below.

- ECR:TXBLKEN = 0
  The FTICR register sets the trigger level for interrupts by the valid transmission data count (remaining amount) in the transmission FIFO.
- ECR:TXBLKEN = 1
  The FTICR register sets the trigger level for interrupts by the amount of free space in the transmission FIFO.

The table below shows the setting conditions for different FCR1:FSEL bit values.

Table 10-5 Transmission FIFO Setting by FCR1:FSEL

FSEL	Selection of Transmission FIFO	Transmission FIFO Interrupt Control Register
0	FIFO1	FTICR1
1	FIFO2	FTICR2

- The initial value of the valid data count for triggering interrupts of the FTICR register is 0x00.
- Set the number of data items that generate a transmission interrupt to FTICR of the transmission FIFO.
  - When ECR:TXBLKEN = 0, if the set number of data items matches, or becomes smaller than, the indication of the number of valid data items in the transmission FIFO (FTICR or FBYTE), the interrupt flag (FDRQ) is set to "1".
  - When ECR:TXBLKEN = 1, if the set number of data items matches, or becomes smaller than, the amount of free space in the transmission FIFO, the interrupt flag (FDRQ) is set to "1". If the

number of data items that is set in this bit becomes larger than the amount of free space in the transmission FIFO, the interrupt flag (FDRQ) is set to "0".

- Set FTICR so that it satisfies: FTICR =< FIFO capacity - 2.
- The read value indicates the valid data count of the FIFO.
- Transmission FIFO: Data count written to the transmission FIFO that has not yet been transmitted.
- Reception FIFO: Data count received by the reception FIFO that has not yet been read.

[bit15:8] FTICR2: FIFO2 Data Count Indication byte
[bit7:0] FTICR1:FIFO1 Data Count Indication Byte

FTICR2, FTICR1	Description
Write	Set the number of valid data items that generates an interrupt.
Read	Read valid data count.

Notes:

- *Any setting that exceeds the FIFO capacity is prohibited.*
- *The setting value cannot be read.*
- *As the FIFO data count for transmission, the transmission data count that has been written, minus 1, is displayed as the valid data count. This is because an attempt to write data to the TDR register stores the data in the transmission FIFO if the TDR register contains data that is yet to be transmitted. When the data in the TDR register is transmitted, the data in the transmission FIFO that has not yet been transmitted is transferred to the TDR register.*
- *The FIFO data count for reception represents the data count received by the reception FIFO but which has not yet been read. The data count does not include the data that has been received by the RDR register.*
- *For ECR:TXBLKEN = 0, if block transfer is performed during DMA transfer, only 1 can be set as the block size.*
- *To perform block transfer with ECR:TXBLKEN = 1, set the value of the block size to this register.*

10.15. Extended Control Register (ECR)

The extended control register (ECR) can set block transfer.

Bit Configuration of the Extended Control Register (ECR)

Figure 10-16 shows the bit configuration of the extended control register (ECR).

Figure 10-16 Bit Configuration of Extended Control Register (ECR)

Bit	7	6	5	4	3	2	1	0
Field	Reserved			EISEL	REIE	TEIE	RX BLKEN	TX BLKEN
R/W Attribute	R0,W0			R/W	R/W	R/W	R/W	R/W
Protection Attribute	-			-	-	-	-	-
Initial Value	000			0	0	0	0	0

[bit7:5] Reserved: Reserved Bits

[bit4] EISEL: Error Interrupt Request Output Selection Bit

This bit selects between the output of an error interrupt from an interrupt request pin and output from an error interrupt request pin. For details on interrupt output selection, see Table 2-2 and Table 2-3.

Bit	Description
0	Output a reception error interrupt request from a reception interrupt request pin. Output a transmission error interrupt request from a transmission interrupt request pin.
1	Output a reception error interrupt request from a reception error interrupt request pin. Output a transmission error interrupt request from a transmission error interrupt request pin.

[bit3] REIE: Reception Error Interrupt Enable Bit

This bit enables/disables reception error interrupt request output. For details on the target interrupt factors, see Table 2-3.

Bit	Description
0	Disable reception error interrupts.
1	Enable reception error interrupts.

[bit2] TEIE: Transmission Error Interrupt Enable Bit

This bit enables/disables transmission error interrupt request output. For details on the target interrupt factors, see Table 2-2.

Bit	Description
0	Disable transmission error interrupts.
1	Enable transmission error interrupts.

[bit1] RXBLKEN: Reception Block Transfer Setting Bit

This bit sets the reception DMA transfer mode.

- "0": Perform DMA transfer in demand transfer mode.
- "1": Perform DMA transfer in block transfer mode.

Bit	Description
0	Demand transfer mode
1	Block transfer mode

[bit0] TXBLKEN: Transmission Block Transfer Setting Bit

This bit sets the transmission DMA transfer mode.

- "0": Perform DMA transfer in demand transfer mode.
- "1": Perform DMA transfer in block transfer mode.

Bit	Description
0	Demand transfer mode
1	Block transfer mode

10.16. Extended Status Register (ESR)

The extended status register (ESR) verifies the reception block transfer error flag and the transmission block transfer error flag.

Extended Status Register (ESR)

Figure 10-17 shows the bit configuration of the extended status register (ESR).

Figure 10-17 Bit Configuration of Extended Status Register (ESR)

Bit	15	14	13	12	11	10	9	8
Field	Reserved				RXUDR	TXOVR	RBERR	TBERR
R/W Attribute	R0,W0				R,WX	R,WX	R,WX	R,WX
Protection Attribute	-				-	-	-	-
Initial Value	0000				0	0	0	0

[bit15:12] Reserved: Reserved Bits

[bit11] RXUDR: Reception FIFO under Run Flag Bit

This bit is set to "1" if the reception FIFO becomes empty due to block transfer, and then a read occurs.

- This bit is reset when the ESRC:RXUDRC bit in the clear register is set to "1".
- For details on the output of interrupt requests, see Table 2-3.

Bit	Description
0	No reception FIFO under run has occurred.
1	A reception FIFO under run has occurred.

Notes:

- *Software reset (SCR:UPCL = 1) resets this bit to "0".*
- *This bit is "0" if reception block transfer enable (ECR:RXBLKEN) is "0".*
- *This bit is set only when the reception FIFO is used.*
- *If Reception FIFO is disabled, this bit is set to "0".*

[bit10] TXOVR: Transmission FIFO Overrun Flag Bit

This bit is set to "1" if the transmission FIFO becomes full due to block transfer, and then a write occurs.

- This bit is reset when the ESRC:TXOVRC bit in the clear register is set to "1".
- For details on the output of interrupt requests, see Table 2-2.

Bit	Description
0	No transmission FIFO overrun has occurred.
1	A transmission FIFO overrun has occurred.

Notes:

- *Software reset (SCR:UPCL = 1) resets this bit to "0".*
- *This bit is "0" if transmission block transfer enable (ECR:TXBLKEN) is "0".*
- *This bit is set only when the transmission FIFO is used.*

– *If Transmission FIFO is disabled, this bit is set to "0".*

[bit9] RBERR: Reception Block Transfer Error Bit

This bit indicates that a block transfer error has occurred during reception.
If block transfer is executed with a value greater than the threshold set in the FBYTE register, "1" is set in this bit as a block transfer error. A reception interrupt occurs if reception interrupts are enabled (SCR:RIE = 1).

- This bit is reset when the ESRC:RBERRC bit in the clear register is set to "1".
- For details on the output of interrupt requests, see Table 2-3.

Bit	Description
0	No reception block transfer error has occurred.
1	A reception block transfer error has occurred.

Notes:

– *Software reset (SCR:UPCL = 1) resets this bit to "0".*
– *This bit is "0" if reception block transfer enable (ECR:RXBLKEN) is "0" or reception interrupt enable (SCR:RIE) is "0".*

[bit8] TBERR: Transmission Block Transfer Error Bit

This bit indicates that a block transfer error has occurred during transmission.
If block transfer is executed with a value greater than the threshold set in the FTICR register, "1" is set in this bit as a block transfer error. A transmission interrupt occurs if transmission interrupts are enabled (SSR:TIE=1).

- This bit is reset when the ESRC:TBERRC bit in the clear register is set to "1".
- For details on the output of interrupt requests, see Table 2-2.

Bit	Description
0	No transmission block transfer error has occurred.
1	A transmission block transfer error has occurred.

Notes:

– *Software reset (SCR:UPCL = 1) resets this bit to "0".*
– *This bit is "0" if transmission block transfer enable (ECR:TXBLKEN) is "0".*
– *If the transmission FIFO is OFF, this bit is set to "1" as a block transfer error if block transfer is executed under one of the conditions below.*
  – Transmission interrupts are enabled (SCR:TIE = 1) and data exists in the transmission data register (SSR:TDRE = 0).
  – Transmission interrupts are disabled (SCR:TIE = 0), transmission bus idle interrupts are enabled (SCR:TBIE = 1), and transmission is in progress (SSR:TBI = 0).
– *If transmission interrupts are enabled (SCR:TIE = 1) and transmission bus idle interrupts are enabled (SCR:TBIE = 1), the transmission interrupt enable (SCR:TIE = 1) condition is given priority.*

10.17. Transmission Block Size Register (TBSIZE)

This register sets the block size for transmission block transfer.

Bit Configuration of the Transmission Block Size Register (TBSIZE)

Figure 10-18 shows the bit configuration of the transmission block size register (TBSIZE).

Figure 10-18 Bit Configuration of Transmission Block Size Register (TBSIZE)

Bit	7	6	5	4	3	2	1	0
Field	TBSIZE							
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

The TBSIZE register sets the block size for transmission block transfer.

[bit7:0] TBSIZE: Transmission Block Size byte

TBSIZE	Description
Write	Set the number of transmission blocks.
Read	Read the setting.

Notes:

- *Any setting that exceeds the FIFO capacity is prohibited.*
- *It is prohibited to set "0" in these bits.*
- *It is prohibited to make a setting that exceeds the setting of the transmission FIFO interrupt control register (FTICR).*
- *In this register, set the same value as the transmission block size set in the DMA controller.*
- *When the transmission FIFO is not used, the transmission block size during block transfer will be "1" regardless of the setting of this register.*

10.18. Serial Auxiliary Control Status Clear Register (SACSRC)

The serial auxiliary control status clear register (SACSRC) can clear the bits in the serial auxiliary control status register (SACSR).

Note:

- *For operations on this register, also see the description of target register SACSR.*

Bit Configuration of Serial Auxiliary Control Status Clear Register (SACSRC)

Figure 10-19 shows the bit configuration of the serial auxiliary control status clear register (SACSRC).

Figure 10-19 Bit Configuration of Serial Auxiliary Control Status Clear Register (SACSRC)

Bit	15	14	13	12	11	10	9	8
Field	STSTC	Reserved						TINTC
R/W Attribute	R0,W	R0,W0						R0,W
Protection Attribute	-							
Initial Value	0	000000						0

Bit	7	6	5	4	3	2	1	0
Field	TINTEC	TSYNEC	TRGEC	Reserved				TMREC
R/W Attribute	R0,W	R0,W	R0,W	R0,W0				R0,W
Protection Attribute	-							
Initial Value	0	0	0	0000				0

[bit15] STSTC: Clearing the Serial Test Bit

Writing "1" to this bit resets SACSR:STST to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit14:9] Reserved: Reserved Bits

[bit8] TINTC: Clearing the Timer Interrupt Flag

Writing "1" to this bit resets SACSR:TINT to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit7] TINTEC: Clearing the Timer Interrupt Enable Bit

Writing "1" to this bit resets SACSR:TINTE to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit6] TSYNEC: Clearing the Synchronous Transmission Enable Bit

Writing "1" to this bit resets SACSR:TSYNE to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit5] TRGEC: Clearing the External Trigger Enable Bit

Writing "1" to this bit resets SACSR:TRGE to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit4:1] Reserved: Reserved Bits

[bit0] TMREC: Clearing the Serial Timer Enable Bit

Writing "1" to this bit resets SACSR:TMRE to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

10.19. FIFO Control Clear Register 1 (FCR1C)

FIFO control clear register 1 (FCR1C) can clear a bit in FIFO control register 1 (FCR1).

Note:

- *For operations on this register, also see the description of target register FCR1.*

Bit Configuration of FIFO Control Clear Register 1 (FCR1C)

Figure 10-20 shows the bit configuration of FIFO control clear register 1 (FCR1C).

Figure 10-20 Bit Configuration of FIFO Control Clear Register 1 (FCR1C)

Bit	15	14	13	12	11	10	9	8
Field	Reserved			FLSTEC	FRIIEC	FDRQC	FTIEC	FSELC
R/W Attribute	R0,W0			R0,W	R0,W	R0,W	R0,W	R0,W
Protection Attribute	-							
Initial Value	000			0	0	0	0	0

* Lower byte [bit7:0] of this register is FIFO Control Clear Register 0 (FCR0C)

[bit15:13] Reserved: Reserved Bits

[bit12] FLSTEC: Clearing the Retransmission Data Lost Detection Enable Bit

Writing "1" to this bit resets FCR1:FLSTE to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit11] FRIIEC: Clearing the Reception FIFO Idle Detection Enable Bit

Writing "1" to this bit resets FCR1:FRIIE to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit10] FDRQC: Clearing the Transmission FIFO Data Request Bit

Writing "1" to this bit resets FCR1:FDRQ to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit9] FTIEC: Clearing the Transmission FIFO Interrupt Enable Bit

Writing "1" to this bit resets FCR1:FTIE to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit8] FSELC: Clearing the FIFO Selection Bit

Writing "1" to this bit resets FCR1:FSEL to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

10.20. FIFO Control Clear Register 0 (FCR0C)

FIFO control clear register 0 (FCR0C) can clear a bit in FIFO control register 0 (FCR0).

Note:

- *For operations on this register, also see the description of target register FCR0.*

Bit Configuration of FIFO Control Clear Register 0 (FCR0C)

Figure 10-21 shows the bit configuration of FIFO control clear register 0 (FCR0C).

Figure 10-21 Bit Configuration of FIFO Control Clear Register 0 (FCR0C)

Bit	7	6	5	4	3	2	1	0
Field	Reserved						FE2C	FE1C
R/W Attribute	R0,W0						R0,W	R0,W
Protection Attribute	-							
Initial Value	00000						0	0

[bit7:2] Reserved: Reserved Bits

[bit1] FE2C: Clearing the FIFO2 Operation Enable Bit

Writing "1" to this bit resets FCR0:FE2 to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit0] FE1C: Clearing the FIFO1 Operation Enable Bit

Writing "1" to this bit resets FCR0:FE1 to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

10.21. Extended Status Clear Register (ESRC)

The extended status clear register (ESRC) can clear a bit in the extended status register (ESR).

Extended Status Clear Register (ESRC)

Figure 10-22 shows the bit configuration of the extended status clear register (ESRC).

Figure 10-22 Bit Configuration of Extended Status Clear Register (ESRC)

Bit	15	14	13	12	11	10	9	8
Field	Reserved				RXUDRC	TXOVRC	RBERRC	TBERRC
R/W Attribute	R0,W0				R0,W	R0,W	R0,W	R0,W
Protection Attribute	-							
Initial Value	0000				0	0	0	0

[bit15:12] Reserved: Reserved Bits

[bit11] RXUDRC: Reception FIFO under Run Flag Clear Bit

Writing "1" to this bit resets ESR:RXUDR to "0".
Writing "0" to this bit is invalid.
"0" is always read from this bit.

[bit10] TXOVRC: Transmission FIFO Overrun Flag Clear Bit

Writing "1" to this bit resets the ESR:TXOVR to "0".
Writing "0" to this bit is invalid.
"0" is always read from this bit.

[bit9] RBERRC: Reception Block Transfer Error Clear Bit

Writing "1" to this bit resets the ESR:RBERR to "0".
Writing "0" to this bit is invalid.
"0" is always read from this bit.

[bit8] TBERRC: Transmission Block Transfer Error Clear Bit

Writing "1" to this bit resets the ESR:TBERR to "0".
Writing "0" to this bit is invalid.
"0" is always read from this bit.

10.22. Serial Auxiliary Control Status Set Register (SACSRS)

The serial auxiliary control status set register (SACSRS) can set the bits in the serial auxiliary control status register (SACSR).

Note:

- *For operations on this register, also see the description of target register SACSR.*

Bit Configuration of the Serial Auxiliary Control Status Set Register (SACSRS)

Figure 10-23 shows the bit configuration of the serial auxiliary control status register (SACSRS).

Figure 10-23 Bit Configuration of Serial Auxiliary Control Status Set Register (SACSRS)

Bit	15	14	13	12	11	10	9	8
Field	STSTS	Reserved						
R/W Attribute	R0,W	R0,W0						
Protection Attribute	-							
Initial Value	0	0000000						

Bit	7	6	5	4	3	2	1	0
Field	TINTES	TSYNES	TRGES	Reserved				TMRES
R/W Attribute	R0,W	R0,W	R0,W	R0,W0				R0,W
Protection Attribute	-							
Initial Value	0	0	0	0000				0

[bit15] STSTS: Setting the Serial Test Bit

Writing "1" to this bit sets the SACSR:STST to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit14:8] Reserved: Reserved Bits

[bit7] TINTES: Setting the Timer Interrupt Enable Bit

Writing "1" to this bit sets SACSR:TINTE to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit6] TSYNES: Setting the Synchronous Transmission Enable Bit

Writing "1" to this bit sets SACSR:TSYNE to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit5]TRGES: Setting the External Trigger Enable Bit

Writing "1" to this bit sets SACSR:TRGE to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit4:1] Reserved: Reserved Bits

[bit0] TMRES: Setting the Serial Timer Enable Bit

Writing "1" to this bit sets SACSR:TMRE to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

10.23. FIFO Control Set Register 1 (FCR1S)

FIFO control set register 1 (FCR1S) can set a bit in FIFO control register 1 (FCR1).

Note:

- *For operations on this register, also see the description of target register FCR1.*

Bit Configuration of FIFO Control Set Register 1 (FCR1S)

Figure 10-24 shows the bit configuration of FIFO control set register 1 (FCR1S).

Figure 10-24 Bit Configuration of FIFO Control Set Register 1 (FCR1S)

Bit	15	14	13	12	11	10	9	8
Field	Reserved			FLSTES	FRIIES	Reserved	FTIES	FSELS
R/W Attribute	R0,W0			R0,W	R0,W	R0,W0	R0,W	R0,W
Protection Attribute	-							
Initial Value	000			0	0	0	0	0

* Lower byte [bit7:0] of this register is FIFO Control Set Register 0 (FCR0S)

[bit15:13] Reserved: Reserved Bits

[bit12] FLSTES: Setting the Retransmission Data Lost Detection Enable Bit

Writing "1" to this bit sets FCR1:FLSTE to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit11] FRIIES: Setting the Reception FIFO Idle Detection Enable Bit

Writing "1" to this bit sets FCR1:FRIIE to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit10] Reserved: Reserved Bit

[bit9] FTIES: Setting the Transmission FIFO Interrupt Enable Bit

Writing "1" to this bit sets FCR1:FTIE to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit8] FSELS: Setting the FIFO Selection Bit

Writing "1" to this bit sets FCR1:FSEL to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

10.24. FIFO Control Set Register 0 (FCR0S)

FIFO control set register 0 (FCR0S) can set a bit in FIFO control register 0 (FCR0).

Note:

- *For operations on this register, also see the description of target register FCR0.*

Bit Configuration of FIFO Control Set Register 0 (FCR0S)

Figure 10-25 shows the bit configuration of FIFO control set register 0 (FCR0S).

Figure 10-25 Bit Configuration of FIFO Control Set Register 0 (FCR0S)

Bit	7	6	5	4	3	2	1	0
Field	Reserved		FLDS	FSETS	FCL2S	FCL1S	FE2S	FE1S
R/W Attribute	R0,W0		R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
Protection Attribute	-							
Initial Value	00		0	0	0	0	0	0

[bit7:6] Reserved: Reserved Bits

[bit5] FLDS: Setting the FIFO Pointer Reload Bit

Writing "1" to this bit sets FCR0:FLD to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit4] FSETS: Setting the FIFO Pointer Saving Bit

Writing "1" to this bit sets FCR0:FSET to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit3] FCL2S: Setting the FIFO2 Reset Bit

Writing "1" to this bit sets FCR0:FCL2 to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit2] FCL1S: Setting the FIFO1 Reset Bit

Writing "1" to this bit sets FCR0:FCL1 to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit1] FE2S: Setting the FIFO2 Operation Enable Bit

Writing "1" to this bit sets FCR0:FE2 to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit0] FE1S: Setting the FIFO1 Operation Enable Bit

Writing "1" to this bit sets FCR0:FE1 to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

CHAPTER 37: CSIO (Clock Synchronous Serial Interface)

This chapter explains the CSIO functions of the multi-function serial interface, supported by operating mode 2.

1. Overview of the CSIO (Clock Synchronous Serial Interface)
2. Interrupts of the CSIO (Clock Synchronous Serial Interface)
3. Operation of the CSIO (Clock Synchronous Serial Interface)
4. Serial Timer Operation
5. Operation of Serial Chip Select
6. Test Mode
7. Dedicated Baud Rate Generator
8. Block Transfer
9. Registers of the CSIO (Clock Synchronous Serial Interface)

CSIO-TXXPT03P01R01L13-E1-XX

1. Overview of the CSIO (Clock Synchronous Serial Interface)

The CSIO (Clock Synchronous Serial Interface) is a general-purpose serial data communication interface for performing synchronous communication with an external machine (supports SPI). In addition, this interface incorporates transmission/reception FIFOs (64 bytes each).

Functions of the CSIO (Clock Synchronous Serial Interface)

	Function	Description
1	Data buffer	– Full duplex, double buffering (when FIFOs are not used) – Transmission and reception FIFOs (64 bytes each) (when FIFOs are used)
2	Transfer format	– Clock synchronization (no start bits/stop bits) – Master/slave function – Supports SPI (supports both master and slave)
3	Baud rate	– Dedicated baud rate generator is provided (configured from a 15-bit reload counter during master operation). – External clock input is enabled (during slave operation).
4	Data length	– Variable to 5 to 16, 20, 24, and 32 bits.
5	Reception error detection	– Overrun error
6	Interrupt request	– Reception interrupt (reception completion, overrun error, and reception block transfer error) – Reception FIFO interrupt (reception FIFO under run) – Transmission interrupt (transmission data empty, transmission bus idle, chip error interrupt, and transmission block transfer error) – Transmission FIFO interrupt (when the transmission FIFO is not higher than the interrupt trigger level or when the transmission FIFO is empty, transmission FIFO overrun) – DMA transfer is supported for both transmission and reception. – Status interrupt (serial timer interrupt)
7	Serial chip select	– 4-channel control (independent control, round control) – The setup/hold/deselect times can be set to variable. – The active level can be selected for each channel.
8	Synchronous transmission function	– Can automatically transmit data periodically in synchronization with the serial timer. – Can transmit in synchronization with an external trigger.
9	Timer function	– A 16-bit serial timer is provided. ➢ A division value can be selected for the operation clock (division by 1 to 256). ➢ Activation by external trigger is available.
10	Synchronous mode	– Master or slave function
11	Pin access	– The serial data output pin can be set to "1".
12	FIFO option	– Transmission and reception FIFOs are provided (64 bytes for the transmission FIFO and 64 bytes for the reception FIFO). – Transmission FIFO and reception FIFO can be selected. – Transmission data can be retransmitted. – The timing of the reception FIFO interrupt can be changed by software. – Independent FIFO reset is supported.

2. Interrupts of the CSIO (Clock Synchronous Serial Interface)

The CSIO (Clock Synchronous Serial Interface) interrupts include reception interrupts, transmission interrupts, and status interrupts. Interrupt requests can be generated according to the factors described below.

- When reception data is set in the reception data register (RDR), or when a reception error occurs
- When transmission data is transferred from the transmission data register (TDR) to the transmission shift register, causing transmission to start
- Transmission bus idle (no transmission)
- Transmission FIFO data request
- When the serial timer comparison value (STMCR) and the serial timer value (STMR) coincide.
- A chip select error occurs.

CSIO Interrupts

Table 2-1 lists the CSIO interrupt control bits and interrupt factors.

Table 2-1 CSIO Interrupt Control Bits and Interrupt Factors

Interrupt Type	Interrupt Request Flag Bit	Flag Register	Interrupt Factor	Interrupt Factor Enable Bit	Interrupt Request Flag Clearing
Reception	RDRF	SSR	1-byte reception	SCR:RIE	- Reading of reception data (RDR) - Software reset (SCR:UPCL=1)
			Reception of as much data as the value set in FBYTE		- Reading of reception data until the reception FIFO is empty - Software reset (SCR:UPCL=1)
			Detection of reception idle for the 8-bit time or longer while the FRIIE bit is "1" and the reception FIFO contains valid data		
	ORE	SSR	Overrun error	SCR:RIE ECR:REIE	- Writing "1" to the reception error flag clear bit (SSR:REC) - Software reset (SCR:UPCL=1)
	RXUDR	ESR	Reception FIFO under run	ECR:REIE	- Writing "1" to the transmission FIFO under run flag clear bit (ESRC:RXUDRC) - Software reset (SCR:UPCL=1) -Disable reception FIFO (1) When FCR1:FSEL=0, set FCR0:FE2=0 (2) When FCR1:FSEL=1, set FCR0:FE1=0
	RBERR	ESR	Reception block transfer error	ECR:REIE	- Writing "1" to the reception block transfer error clear bit (ESRC:RBERRC) - Software reset(SCR:UPCL=1) - Disable reception block transfer mode (ECR:RXBLKEN=0)

Interrupt Type	Interrupt Request Flag Bit	Flag Register	Interrupt Factor	Interrupt Factor Enable Bit	Interrupt Request Flag Clearing
Transmission	TDRE	SSR	The transmission register is empty.	SCR:TIE	Writing to the transmission data register (TDR) or writing "1" to the transmission FIFO operation enable bit (retransmission) when the transmission FIFO operation enable bit is "0" and the transmission FIFO contains valid data*1
	TBI	SSR	No transmission operation	SCR:TBIE	Writing to the transmission data register (TDR) or writing "1" to the transmission FIFO operation enable bit (retransmission) when the transmission FIFO operation enable bit is "0" and the transmission FIFO contains valid data*1
	FDRQ	FCR1	The amount of data stored in the transmission FIFO is equal to or smaller than the FTICR setting value or it is empty.	FCR1:FTIE	Writing "1" to the FIFO transmission data request clear bit (FCR1C:FDRQC), or the transmission FIFO full condition
	CSE	SACSR	In slave mode (SCR:MS = 1), the serial chip select pin is inactive during transmission. In master mode (SCR:MS = 0), the transmission count is equal to or less than the TBYTE setting and the next transmission data is not written to TDR (SSR:TDRE = 1).	SACSR:CSEIE	- Writing "1" to the serial chip select flag clear bit (SACSRC:CSEC) - Software reset (SCR:UPCL=1)
	TXOVR	ESR	Transmission FIFO overrun	ECR:TEIE	- Writing "1" to the transmission FIFO overrun flag clear bit (ESRC:TXOVRC) - Software reset(SCR:UPCL=1) - Disable transmission FIFO (1) When FCR1:FSEL=0, set FCR0:FE1=0 (2) When FCR1:FSEL=1, set FCR0:FE2=0
	TBERR	ESR	Transmission block transfer error	ECR:TEIE	- Writing "1" to the transmission block transfer error clear bit (ESRC:TBERRC) - Software reset(SCR:UPCL=1) - Disable transmission block transfer(ECR:TXBLKEN=0)

Interrupt Type	Interrupt Request Flag Bit	Flag Register	Interrupt Factor	Interrupt Factor Enable Bit	Interrupt Request Flag Clearing
Status	TINT	SACSR	Coincidence of the serial timer register (STMR) and the serial timer comparison register (STMCR) values	SACSR:TINTE	- Writing "1" to the timer interrupt flag clear bit (SACSRC:TINTC) - Software reset(SCR:UPCL=1)

*1: Set the TIE bit to "1" after the TDRE bit becomes "0".

Switching of Error Interrupt Output

The error interrupt request output method depends on the setting of the error interrupt request output switching bit (ECR:EISEL), the transmission and reception error interrupt enable bits (SACSR:CSEIE, ECR:REIE, TEIE) or the transmission and reception interrupt enable bits (SCR:RIE, TIE, TBIE, FCR1:FTIE).

Table 2-2 How to Output Transmission Interrupt Requests

ECR:EISEL	Interrupt Output	Target Interrupt Enable Bit	Interrupt Factor
0	Transmission interrupt request output	SCR:TIE	Transmission data empty (SSR:TDRE)
		SCR:TBIE	No transmission operation (SSR:TBI)
		FCR1:FTIE	Transmission FIFO empty (FCR1:FDRQ)
		SACSR:CSEIE	Chip select error (SACSR:CSE)
		ECR:TEIE	Transmission FIFO overrun (ESR:TXOVR) Transmission block transfer error (ESR:TBERR)
	Transmission error interrupt request output	-	No interrupt factor
1	Transmission interrupt request output	SCR:TIE	Transmission data empty (SSR:TDRE)
		SCR:TBIE	No transmission operation (SSR:TBI)
		FCR1:FTIE	Transmission FIFO empty (FCR1:FDRQ)
	Transmission error interrupt request output	SACSR:CSEIE	Chip select error (SACSR:CSE)
		ECR:TEIE	Transmission FIFO overrun (ESR:TXOVR) Transmission block transfer error (ESR:TBERR)

Table 2-3 How to Output Reception Interrupt Requests

ECR:EISEL	Interrupt Output	Target Interrupt Enable Bit	Interrupt Factor
0	Reception interrupt request output	SCR:RIE	Reception full (SSR:RDRF) Overrun error (SSR:ORE)
		ECR:REIE	Reception FIFO under run (ESR:RXUDR) Reception block transfer error (ESR:RBERR)
	Reception error interrupt request output	-	No interrupt factor
1	Reception interrupt request output	SCR:RIE	Reception full (SSR:RDRF)
	Reception error interrupt request output	ECR:REIE	Overrun error (SSR:ORE) Reception FIFO under run (ESR:RXUDR) Reception block transfer error (ESR:RBERR)

2.1. Reception Interrupt Generation and Flag Set Timing

Reception-time interrupts include reception completion (SSR:RDRF) and reception error occurrence (SSR:ORE).

Reception Interrupt Generation and Flag Set Timing

Upon the detection of the last data bit, reception data is stored in the reception data register (RDR). Once reception is completed (SSR:RDRF = 1) or when a reception error occurs (SSR:ORE = 1), each flag is set. At this time, if reception interrupts are enabled (SCR:RIE = 1), a reception interrupt is generated.

Note:

- *If a reception error occurs, the data in the reception data register (RDR) will be invalid.*

Figure 2-1 Reception Operation and Flag Set Timing

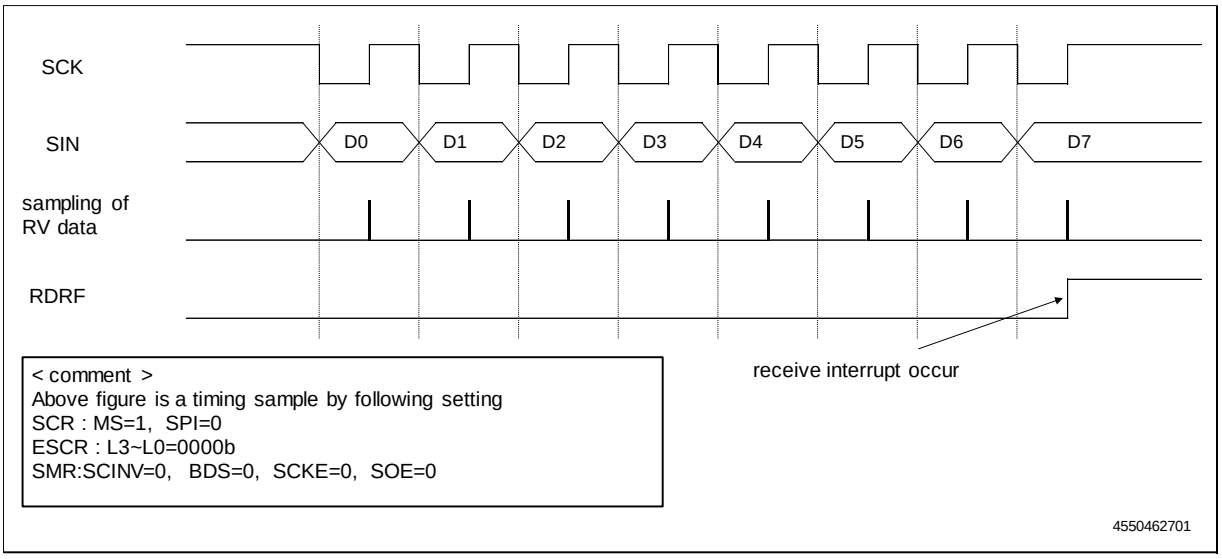
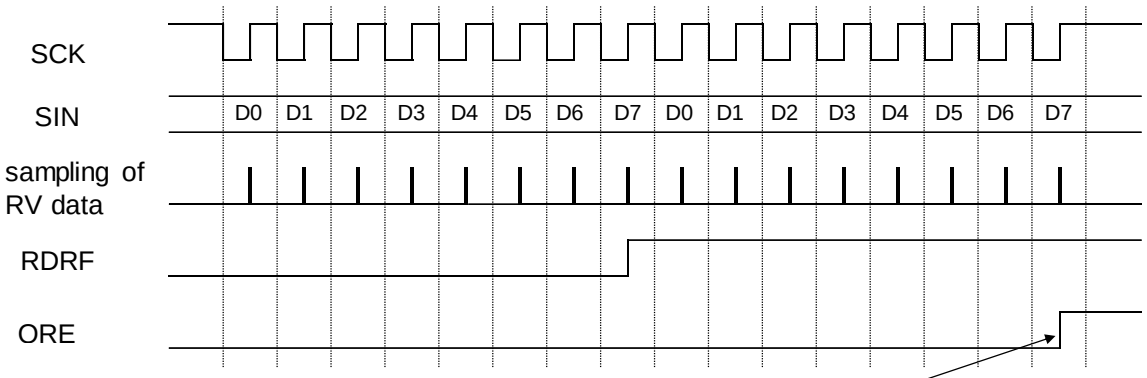


Figure 2-2 ORE (Overrun Error) Flag Set Timing



< comment >
Above figure is a timing sample by following setting
SCR : MS=1, SPI=0
ESCR : L3~L0=0000b
SMR:SCINV=0, BDS=0, SCKE=0, SOE=0

(caution)
An overrun error will occur when next receive data come before reading previous received data.

4550462711

2.2. Interrupt Generation and Flag Set Timing When the Reception FIFO Is Used

When the reception FIFO is used, an interrupt is generated if as many data items as the setting in the FBYTE register (FBYTE) are received.

Reception Interrupt Generation and Flag Set Timing When the Reception FIFO is Used

The occurrence of interrupts during the use of the reception FIFO is determined by the value set in the FBYTE register.

- If as much transfer data as the amount set in the FBYTE register is received, the reception data full flag (SSR:RDRF) in the serial status register will be set to "1". At this time, if reception interrupts are enabled (SCR:RIE), a reception interrupt will be generated.
- When both of the following conditions are satisfied, continuation of reception idle status for 8 baud rate clocks or longer will set the interrupt flag (RDRF) to "1".
  - The reception FIFO idle detection enable bit (FRIIE) is "1".
  - The number of data items in the reception FIFO does not reach the transfer count.

  During an 8-clock count, the counter is reset to 0 when RDR is read, so that the system starts counting the 8 clocks again. The counter is reset to 0 when the reception FIFO is disabled. If the reception FIFO is enabled when there is remaining data in the reception FIFO, counting restarts.
- Once the reception FIFO becomes empty through reading of the reception data (RDR), the reception data full flag (SSR:RDRF) is cleared.
- If the reception valid data count becomes equal to the value of the FIFO capacity, reception of the next data triggers an overrun error (SSR:ORE = 1).

Figure 2-3 Reception Interrupt Generation Timing When Reception FIFO Is Used

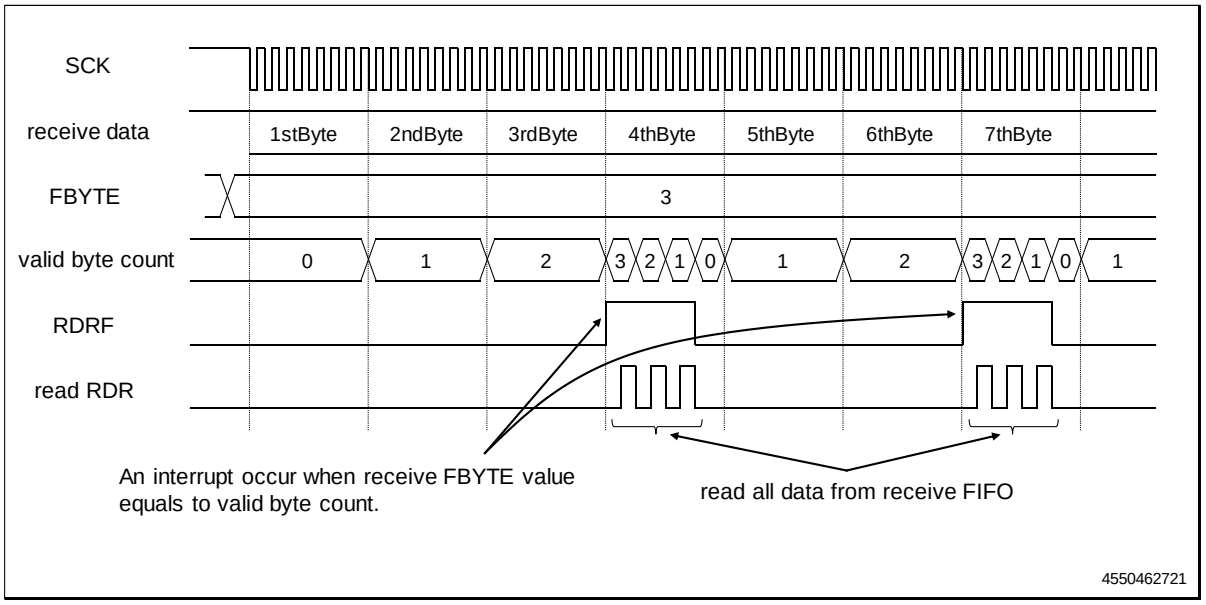
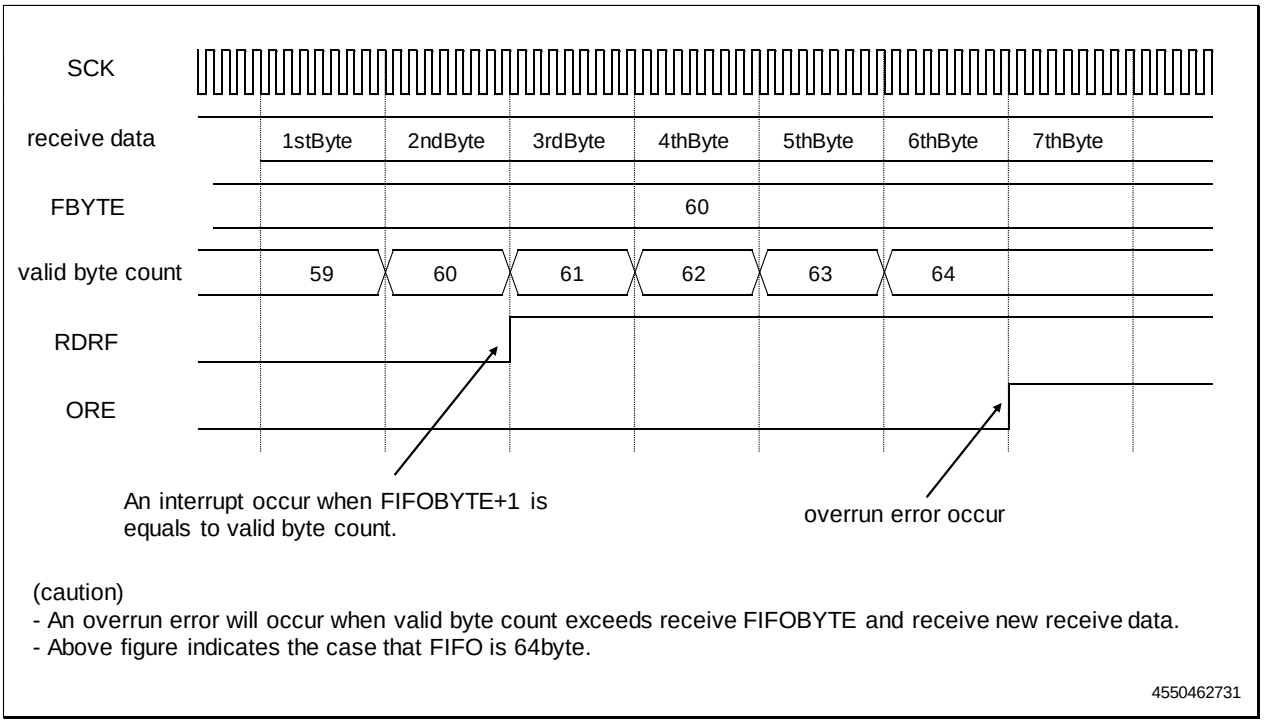


Figure 2-4 Set Timing of ORE (Overrun Error) Flag Bit



2.3. Transmission Interrupt Generation and Flag Set Timing

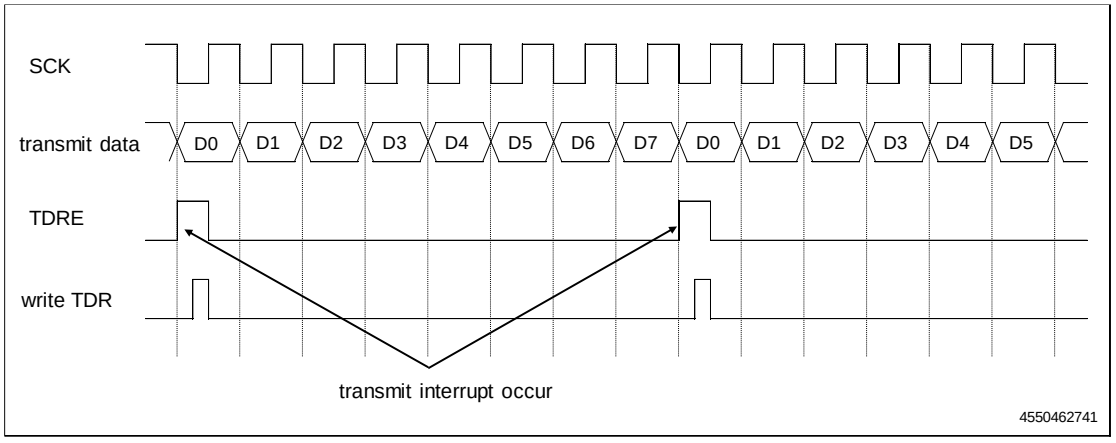
An interrupt during transmission is generated if transmission data is transferred from the transmission data register (TDR) to the transmission shift register (SSR:TDRE = 1) and transmission is started, or if transmission is not being performed (SSR:TBI = 1).

Transmission Interrupt Generation and Flag Set Timing

Set Timing of the Transmission Data Empty Flag (SSR:TDRE)

When the data written to the transmission data register (TDR) is transferred to the transmission shift register, the system enters the state in which the next data can be written (SSR:TDRE = 1). At this time, if transmission interrupts are enabled (SCR:TIE = 1), a transmission interrupt is generated. The SSR:TDRE bit is a read only bit, so the SSR:TDRE bit is cleared to "0" with the writing of data to the transmission data register (TDR).

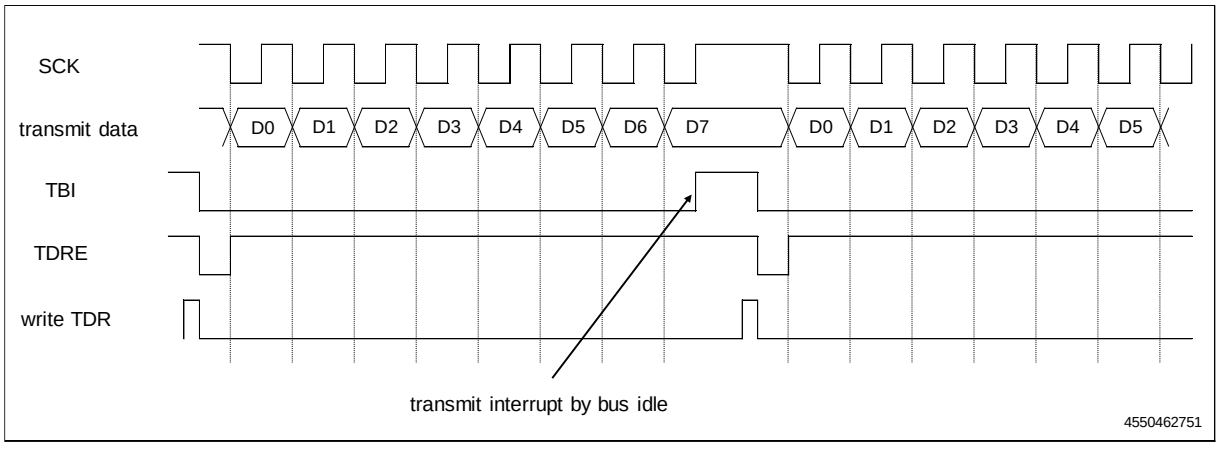
Figure 2-5 Set Timing of Transmission Data Empty Flag (SSR:TDRE) (SCR:MS = 0, SCR:SPI = 0)



Set Timing of the Transmission Bus Idle Flag (SSR:TBI)

The SSR:TBI bit is set to "1" during those periods in which the transmission data register is empty (SSR:TDRE = 1) and transmission is not being performed. At that time, a transmission interrupt occurs if the transmission bus idle interrupt is enabled (SCR:TBIE = 1). If transmission data is set in the transmission data register (TDR), the SSR:TBI bit and the transmission interrupt request are cleared.

Figure 2-6 Set Timing of Transmission Bus Idle Flag (TBI)
(SCSCR:CSEN3 to CSEN0 = 0b0000, SACSR:TSYNE = 0)



2.4. Interrupt Generation and Flag Set Timing When the Transmission FIFO Is Used

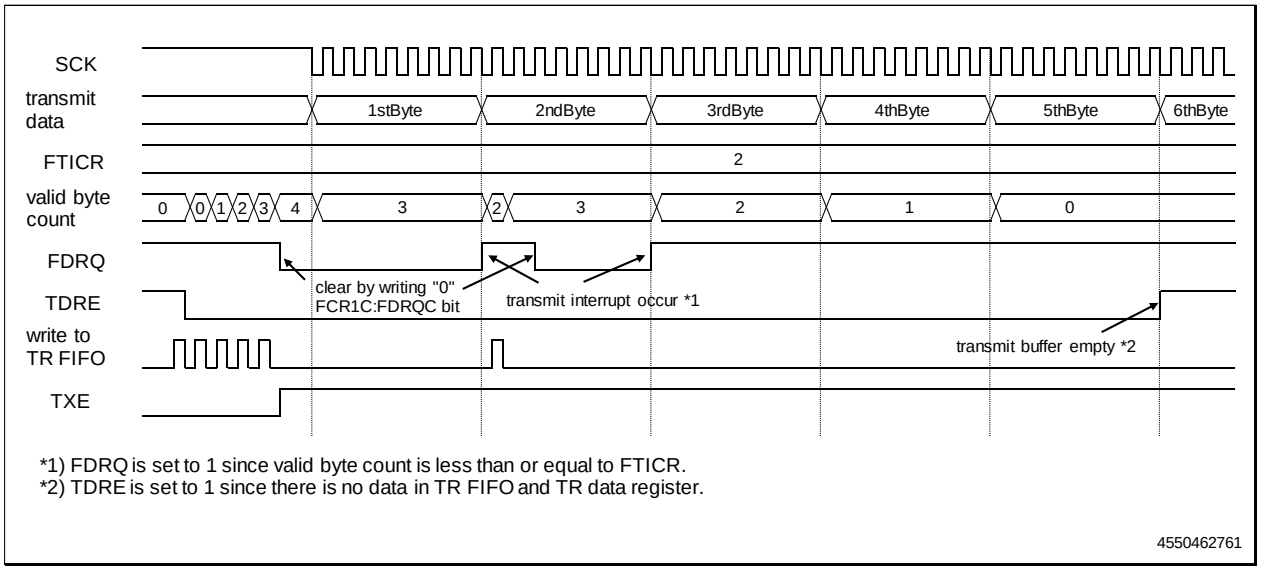
When the transmission FIFO is used, an interrupt is generated if the number of data items stored in the transmission FIFO is equal to or less than the setting made for the FTICR register (FTICR).

Transmission Interrupt Generation and Flag Set Timing When the Transmission FIFO is Used

The occurrence of interrupts during the use of the transmission FIFO is determined by the value set in the FTICR register.

- When the data count in the transmission FIFO is equal to or less than the set value in the FTICR register, the FIFO transmission data request bit (FCR1:FDRQ) is set to "1".
  At this time, a transmission interrupt occurs if the FIFO transmission interrupt is enabled (FCR1:FTIE = 1).
- After a transmission interrupt is generated, write the necessary data to the transmission FIFO, and then clear the interrupt request by writing "1" to the FIFO transmission data request clear bit (FCR1C:FDRQC).
- The FIFO transmission data request bit (FCR1:FDRQ) is set to "0" once the transmission FIFO becomes full.
- The existence of data in the transmission FIFO can be verified by reading the FIFO byte register (FBYTE) or the transmission FIFO interrupt control register (FTICR).
  FBYTE = 0x00 and FTICR = 0x00 indicate that the transmission FIFO does not contain data.

Figure 2-7 Occurrence Timing of Transmission Interrupt When Transmission FIFO Is Used



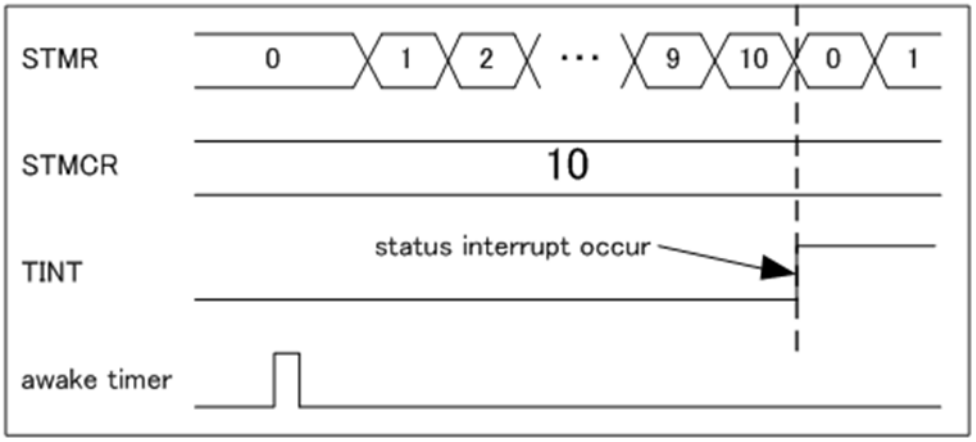
2.5. Timer Interrupt Generation and Flag Set Timing

A timer interrupt occurs when the serial timer register (STMR) coincides with the serial timer comparison register (STMCR).

Timer Interrupt Generation and Flag Set Timing

- The timer interrupt flag (SACSR:TINT) is set to "1" when the serial timer register (STMR) coincides with the serial timer comparison register.
  At this time, a status interrupt occurs if the timer interrupt is enabled (SACSR:TINTE = 1).

Figure 2-8 Occurrence Timing of Timer Interrupt



2.6. Chip Select Error Occurrence and Flag Set Timing

A chip select error occurs in master mode (SCR:MS = 0) if fewer frames than the value set in TBYTE have been transmitted, and the transmission data register (TDR) contains no valid data (SSR:TDRE = 1) . A chip select error also occurs during transmission in slave mode (SCR:MS = 1) if the serial chip select pin becomes inactive.

Chip Select Error Occurrence and Flag Set Timing

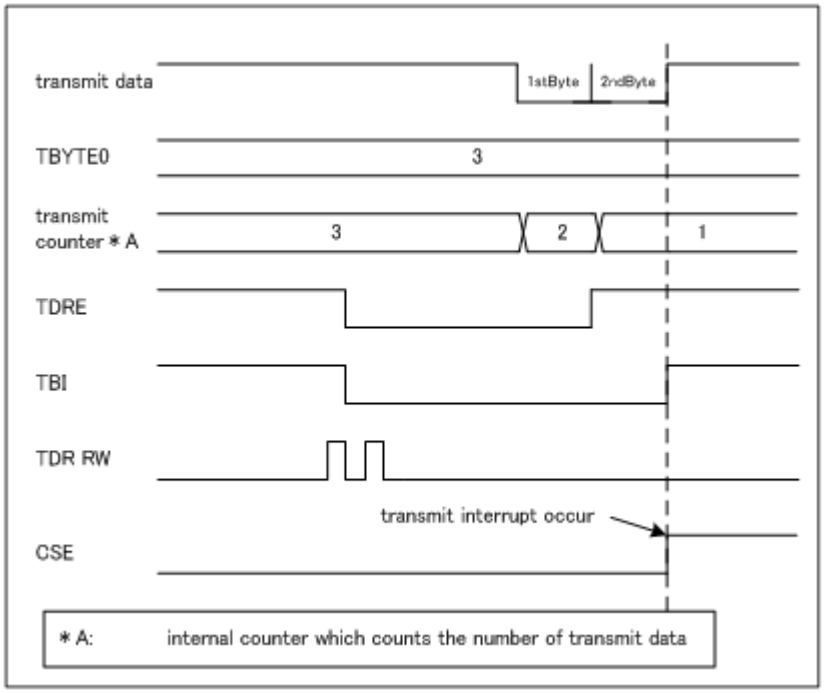
Master Mode (SCR:MS = 0)

A chip select error occurs when transfer byte errors are enabled (TBEEN = 1) in any of the following cases, if the transmission data register (TDR) contains no valid transmission data (SSR:TDRE = 1) before as many data frames as the setting of TBYTE are transmitted.

- When chip select is used
- When synchronous transmission with the serial timer is used
- When transmission activation with an external trigger is used

At this time, if chip select error interrupts are enabled (SACSR:CSEIE = 1), a transmission interrupt is generated.

Figure 2-9 Chip Select Error Occurrence Timing (SCSCR:CSEN3 to CSEN0 = 0b0000, SACSR:TSYNE = 1)



Notes:

- *When serial chip select is used, the chip select error flag (SACSR:CSE) is set to "1" after the elapse of the deselect time from the occurrence of a chip select error. Even if transmission data is written to the transmission data register (TDR) within the hold delay time, transmission operation does not start, and the chip select error flag (SACSR:CSE) is set to "1" after the elapse of the deselect time.*

- *If "1" is set in the chip select error flag (SACSR:CSE), transmission does not start even if transmission data is written to the transmission data register (TDR).*
- *If "1" is set in the chip select error flag (SACSR:CSE) when synchronous transmission is used, transmission operation does not start under the conditions described below.*
  - The serial timer register (STMR) and the serial timer comparison register match during transmission in synchronization with the serial timer.
  - During transmission in synchronization with an external trigger, an edge of the external trigger that is set with the trigger select bits (TRG1, TRG0) is detected.

Slave Mode (SCR:MS = 1)

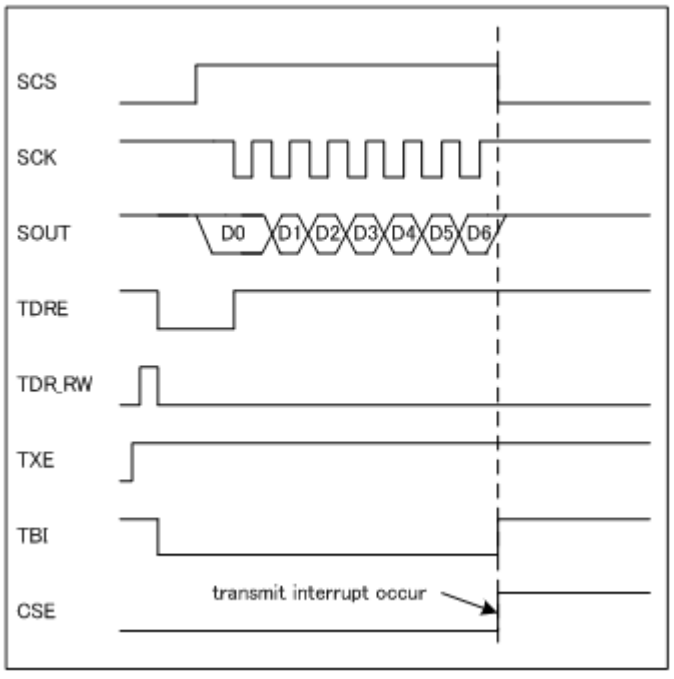
A chip select error occurs if chip select becomes inactive in any of the following cases.

- Serial clock in operation
- The transmission module is not idle and the serial clock has changed.

*If the transmission module is not idle, this means that the transmission data has been prepared and that transmission starts when the serial clock is entered.

At this time, if chip select error interrupts are enabled (SACSR:CSEIE = 1), a transmission interrupt is generated.

Figure 2-10 Chip Select Error Occurrence Timing (CSLVL = 0, SCR:SPI = 0)



Note:

- *When the transmission data register (SSR:TDR) is empty (TDRE = 1), if a serial chip select error (SACSR:CSE = 1) occurs, this bit is set to "1" within the baud rate period.*

3. Operation of the CSIO (Clock Synchronous Serial Interface)

Clock synchronization is used as the transfer method.

3.1. Normal Transfer (I)

Features

	Item	Description
1	Serial clock (SCK) mark level	"H"
2	Transmission data output timing	SCK falling edge
3	Reception data sampling	SCK rising edge
4	Data length	5 to 16, 20, 24 and 32 bits

Register Setting

The register settings needed for normal transfer (I) are listed below.

SCR:SPI = 0*1, SMR:MD2 = 0, MD1 = 1, MD0 = 0, SCINV = 0*1

During master operation: SCR:MS = 0, SMR:SCKE = 1

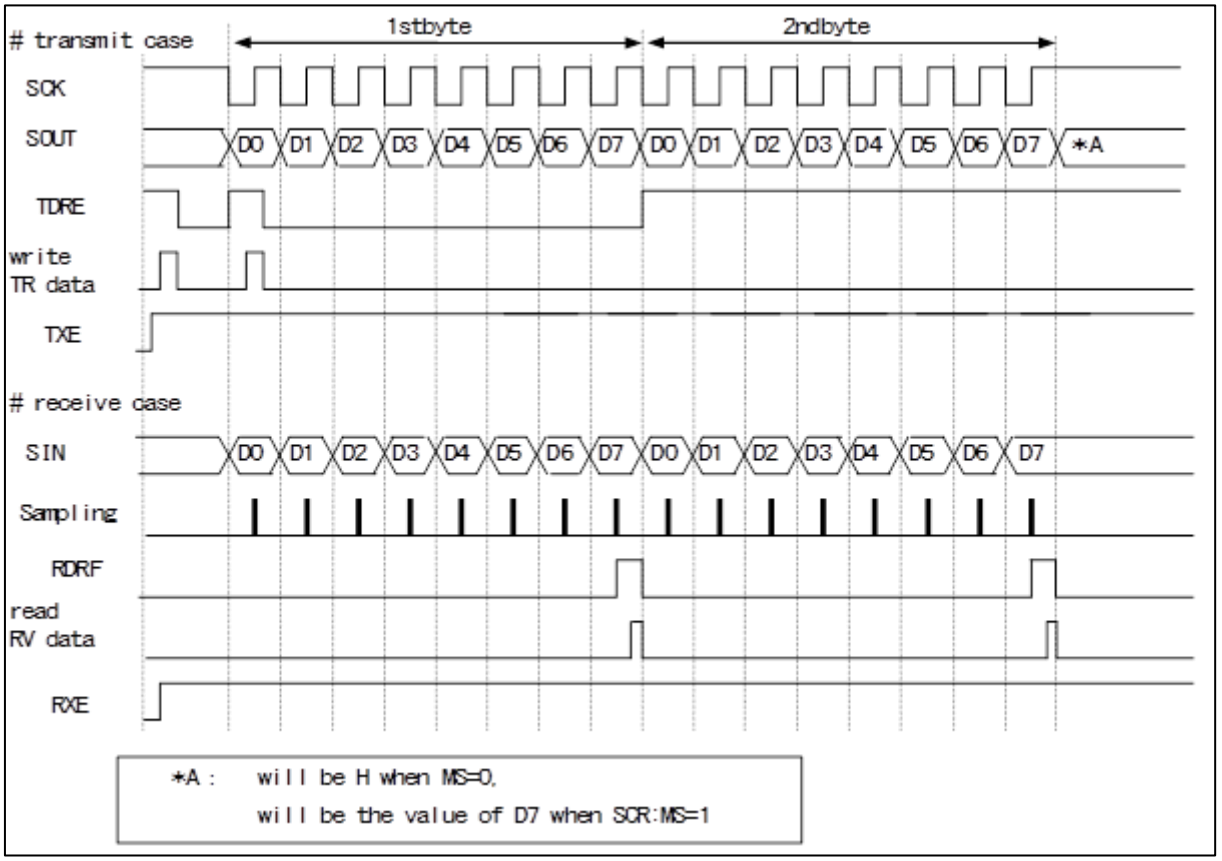
During slave operation: SCR:MS = 1, SMR:SCKE = 0

*1) The bit to be set differs depending on the condition. See Table 5-2.

Note:

- *Except for the bits described above, set the registers according to their use.*

Normal Transfer (I) Timing Chart (When the Serial Chip Select Pin is Not Used)



Master Operation (SCR:MS = 0, SMR:SCKE = 1, SCSCR:CSEN3 to CSEN0 = 0b0000)

Transmission Operation

1. After serial data output is enabled (SMR:SOE = 1), transmission operation is enabled (SCR:TXE = 1), and reception operation is disabled (SCR:RXE = 0), the writing of transmission data to TDR causes SSR:TDRE to be set to 0. In this way, transmission data is output in synchronization with a falling edge of the serial clock (SCK) output.
2. When the first 1 bit of the transmission data is output, SSR:TDRE is set to 1. Thus, if the transmission interrupts are enabled (SCR:TIE = 1), a transmission interrupt request is output. At this time, the second byte of transmission data can be written.

Reception Operation

1. If serial data output is disabled (SMR:SOE = 0) and transmission operation is enabled (SCR:TXE = 1) while reception operation is enabled (SCR:RXE = 1), writing dummy data to TDR causes the reception data to be sampled at a rising edge of the serial clock output (SCK).
2. When the last bit is received, SSR:RDRF is set to 1. At this time, if reception interrupts are enabled (SCR:RIE = 1), a reception interrupt request is output.
   At this time, reception data (RDR) can be read.
3. When reception data (RDR) is read, SSR:RDRF is cleared to "0".

Notes:

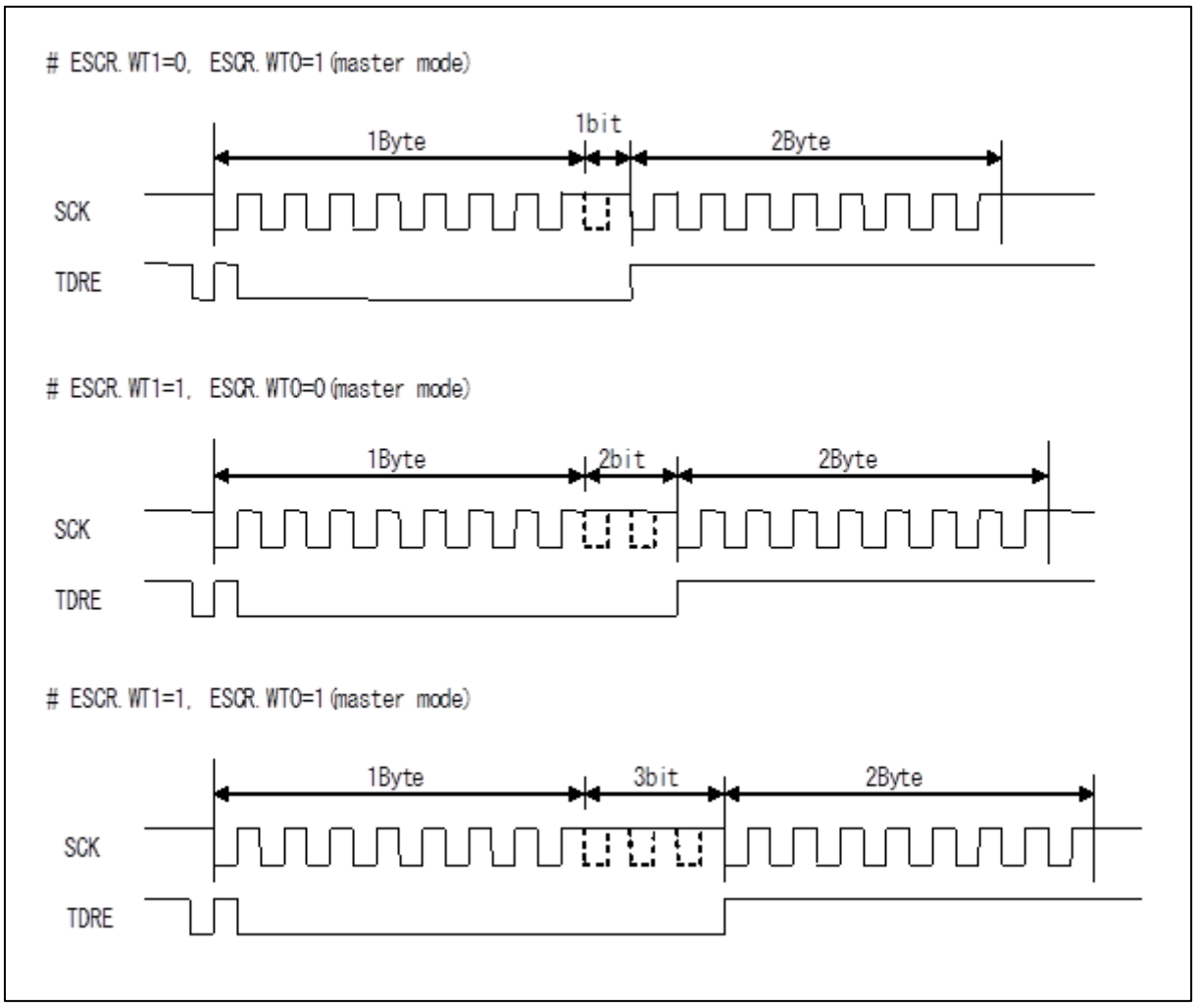
- *If performing reception operation only, write dummy data to TDR to output the serial clock (SCK).*
- *When the transmission/reception FIFOs are enabled, setting the number of frames to transfer in the FBYTE register causes as many serial clock (SCK) pulses as the number of frames to be output.*

Transmission/Reception Operations

1. To perform transmission/reception at the same time, enable serial data output (SMR:SOE = 1) and enable transmission/reception (SCR:TXE, RXE = 1).
2. When transmission data is written to TDR, SSR:TDRE is set to 0, and transmission data is output in synchronization with a falling edge of the serial clock (SCK) output. When the first 1 bit of the transmission data is output, SSR:TDRE is set to 1, and if transmission interrupts are enabled (SCR:TIE = 1), a transmission interrupt request is output. At this time, the second byte of the transmission data can be written.
3. Reception data is sampled at a rising edge of the serial clock (SCK) output. When the last bit of the reception data is received, SSR:RDRF is set to 1. If reception interrupts are enabled (SCR:RIE = 1), a reception interrupt request is output. At this time, reception data (RDR) can be read. When reception data is read, SSR:RDRF is cleared to "0".

Continuous Data Transmission or Reception Wait Operation

If settings other than (ESCR:WT1, ESCR.WT0) = (0, 0) are set for continuous data transmission or reception, a wait is inserted between the frames.



Slave Operation (SCR:MS = 1, SMR:SCKE = 0, SCSCR:CSEN0 = 0)

Transmission Operation

1. Enabling serial data output (SMR:SOE = 1) as well as transmission (SCR:TXE = 1) and then writing transmission data to TDR causes SSR:TDRE to be set to 0. For this reason, transmission data is output in synchronization with a falling edge of the serial clock (SCK) input.
2. When the first 1 bit of the transmission data is output, SSR:TDRE is set to 1. If transmission interrupts are enabled (SCR:TIE = 1), a transmission interrupt request is output. At this time, the second byte of the transmission data can be written.

Note:

- *After transmission operation is enabled (SCR:TXE = 1), if the first writing of transmission data to TDR is performed while the serial clock (SCK) is not at the mark level, the first 1 bit of data is not output, and transmission operation is not performed normally. After transmission is enabled (SCR:TXE = 1), perform the first writing of transmission data to TDR with SSR:TBI = 1 when the serial clock (SCK) is at the mark level.*

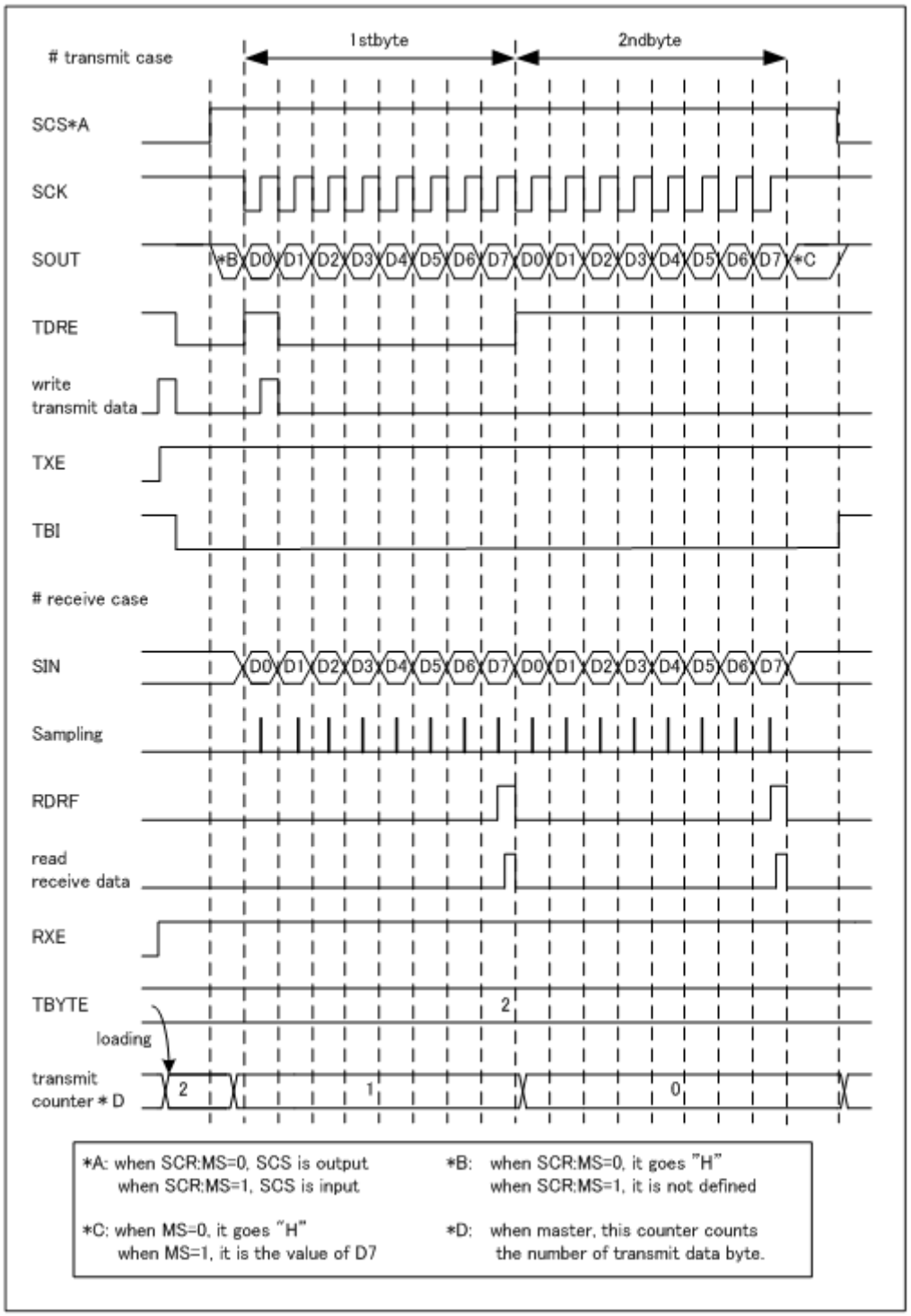
Reception Operation

1. If serial data output is disabled (SMR:SOE = 0) and reception operation is enabled (SCR:RXE = 1), reception data is sampled at a rising edge of the serial clock input (SCK).
2. If the last bit is received, SSR:RDRF is set to 1. If reception interrupts are enabled (SCR:RIE = 1), a reception interrupt request is output.
   At this time, reception data (RDR) can be read.
3. When reception data (RDR) is read, SSR:RDRF is cleared to "0".

Transmission/Reception Operations

1. To perform transmission/reception at the same time, enable serial data output (SMR:SOE = 1) and enable transmission/reception (SCR:TXE, RXE = 1).
2. When transmission data is written to TDR, SSR:TDRE is set to 0, and transmission data is output in synchronization with a falling edge of the serial clock (SCK) input. When the first 1 bit of transmission data is output, SSR:TDRE is set to 1, and if transmission interrupts are enabled (SCR:TIE = 1), a transmission interrupt request is output. At this time, the second byte of transmission data can be written.
3. Reception data is sampled at a rising edge of the serial clock (SCK) input. If the last bit of the reception data is received, SSR:RDRF is set to 1, and if reception interrupts are enabled (SCR:RIE = 1), a reception interrupt request is output. At this time, reception data (RDR) can be read. When reception data is read, SSR:RDRF is cleared to "0".

Normal Transfer (I) Timing Chart (When the Serial Chip Select Pin is Used)



Master Operation (SCR:MS = 0, SMR:SCKE = 1, SCSCR:CSOE = 1, SCSCR:CSENn* = 1)

*: n is replaced by the serial chip select pin number used.

Transmission Operation

1. Enabling serial data output (SMR:SOE = 1) and transmission operation (SCR:TXE = 1), while disabling reception operation (SCR:RXE = 0), and then writing transmission data to TDR causes SSR:TDRE to be set to 0. Later, the serial chip select pin (SCS) becomes active, and after the elapse of the serial chip select pin setup time, serial clock output is started. After serial clock output is started, transmission data is output in synchronization with a falling edge of the serial clock (SCK) output.
2. When the first 1 bit of transmission data is output, SSR:TDRE is set to 1, and if transmission interrupts are enabled (SCR:TIE = 1), a transmission interrupt request is output. At this time, the second byte of transmission data can be written.
3. After the end of data transmission for which the count is set in TBYTE, the serial clock stops.
4. After the serial clock stops, the serial chip select pin (SCS) becomes inactive after the elapse of the serial chip select pin hold time. At this time, however, if the serial chip select active level (SCSCR:SCAM = 1) is retained, the serial chip select pin (SCS) retains the active state.

Reception Operation

1. Disabling serial data output (SMR:SOE = 0), enabling transmission operation (SCR:TXE = 1), and enabling reception operation (SCR:RXE = 1), and then writing dummy data to TDR, causes the serial chip select pin (SCS) to become active, and after the elapse of the serial chip select pin setup time, serial clock output is started. After serial clock output is started, reception data is sampled at a rising edge of the serial clock output (SCK).
2. When the last bit is received, SSR:RDRF is set to 1, and if reception interrupts are enabled (SCR:RIE = 1), a reception interrupt request is output.

   At this time, reception data (RDR) can be read.
3. When reception data (RDR) is read, SSR:RDRF is cleared to "0".
4. After the end of data reception for which the count is set in TBYTE, the serial clock stops.
5. After the serial clock stops, the serial chip select pin (SCS) becomes inactive after the elapse of the serial chip select pin hold time. At this time, however, if the serial chip select active level (SCSCR:SCAM = 1) is retained, the serial chip select pin (SCS) retains the active state.

Notes:

- *When performing reception only, write dummy data to TDR to output the serial clock (SCK).*
- *When the transmission/reception FIFOs are enabled, setting the number of frames to transfer in the FBYTE register causes as many serial clock (SCK) pulses as the number of frames to be output.*

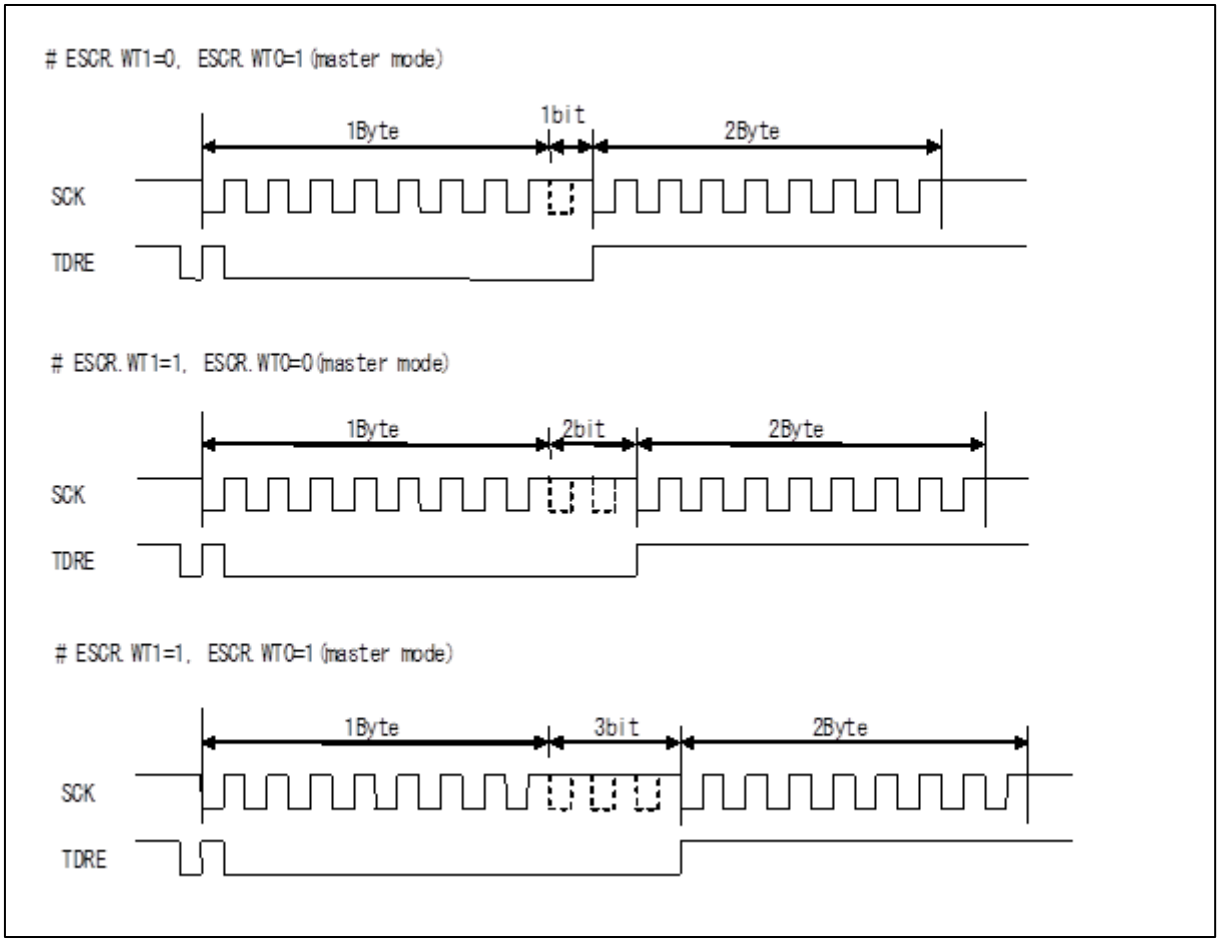
Transmission/Reception Operations

1. To perform transmission/reception at the same time, enable serial data output (SMR:SOE = 1) as well as transmission/reception (SCR:TXE, RXE = 1).
2. When transmission data is written to TDR, SSR:TDRE is set to 0. Later, the serial chip select pin (SCS) becomes active, and after the elapse of the serial chip select pin setup time, serial clock output is started. After serial clock output is started, transmission data is output in synchronization with a falling edge of the serial clock (SCK) output. When the first 1 bit of transmission data is output, SSR:TDRE is set to 1, and if transmission interrupts are enabled (SCR:TIE = 1), a transmission interrupt request is output. At this time, the second byte of transmission data can be written.
3. During transmission/reception, reception data is sampled at a rising edge of the serial clock (SCK) output. If the last bit of the reception data is received, SSR:RDRF is set to 1, and if reception interrupts are enabled (SCR:RIE = 1), a reception interrupt request is output. At this time, reception data (RDR) can be read. When reception data is read, SSR:RDRF is cleared to "0".
4. After the end of data transmission/reception for which the count is set in TBYTE, the serial clock stops.

5. After the serial clock stops, the serial chip select pin (SCS) becomes inactive after the elapse of the serial chip select pin hold time. At this time, however, if the serial chip select active level (SCSCR:SCAM = 1) is retained, the serial chip select pin (SCS) retains the active state.

Continuous Data Transmission or Reception Wait Operation

If settings other than (ESCR:WT1, ESCR.WT0) = (0, 0) are set for continuous data transmission or reception, a wait is inserted between the frames.



Slave Operation (SCR:MS = 1, SMR:SCKE = 0, SCSCR:CSEN0 = 1, SCSCR:CSOE = 0, SCSCR:SCAM = 0)

Transmission Operation

1. Enabling serial data output (SMR:SOE = 1) and enabling transmission (SCR:TXE = 1) and then writing transmission data to TDR causes SSR:TDRE to be set to 0.
2. If the serial chip select pin (SCS) becomes active, transmission is started, and transmission data is output in synchronization with a falling edge of the serial clock (SCK) input.
3. When the first 1 bit of transmission data is output, SSR:TDRE is set to 1, and if transmission interrupts are enabled (SCR:TIE = 1), a transmission interrupt request is output. At this time, the second byte of transmission data can be written.
4. If the serial chip select pin (SCS) becomes inactive, transmission operation is ended, and the serial output pin (SOUT) becomes "H".

Note:

- *After transmission operation is enabled (SCR:TXE = 1), if the first writing of transmission data to TDR is performed when the serial clock (SCK) is not at the mark level, the first 1 bit of data is not output, and transmission operation is not performed normally. After transmission is enabled (SCR:TXE = 1), perform the first writing of transmission data to TDR with SSR:TBI = 1 when the serial clock (SCK) is at the mark level.*

Reception Operation

1. If serial data output is disabled (SMR:SOE = 0) and reception is enabled (SCR:RXE = 1), and if the serial chip select pin (SCS) becomes active, reception starts, and reception data is sampled at a rising edge of the serial clock input (SCK).
2. When the last bit is received, SSR:RDRF is set to 1, and if reception interrupts are enabled (SCR:RIE = 1), a reception interrupt request is output.
3. At this time, reception data (RDR) can be read.
4. When reception data (RDR) is read, SSR:RDRF is cleared to "0".
5. If the serial chip select pin (SCS) becomes inactive, reception operation is ended.

Transmission/Reception Operations

1. To perform transmission/reception operations at the same time, enable serial data output (SMR:SOE = 1) and enable transmission/reception operations (SCR:TXE, RXE = 1).
2. When transmission data is written to TDR, SSR:TDRE is set to 0. Later, if the serial chip select pin (SCS) becomes active, transmission/reception operations are started, and transmission data is output in synchronization with a falling edge of the serial clock (SCK) input. When the first 1 bit of transmission data is output, SSR:TDRE is set to 1, and if transmission interrupts are enabled (SCR:TIE = 1), a transmission interrupt request is output. At this time, the second byte of transmission data can be written.
3. During transmission/reception, reception data is sampled at a rising edge of the serial clock (SCK) input. If the last bit of the reception data is received, SSR:RDRF is set to 1, and if reception interrupts are enabled (SCR:RIE = 1), a reception interrupt request is output. At this time, reception data (RDR) can be read. When reception data is read, SSR:RDRF is cleared to "0".
4. If the serial chip select pin (SCS) becomes inactive, transmission/reception operations stop, and the serial output pin (SOUT) becomes "H".

3.2. Normal Transfer (II)

Features

	Item	Description
1	Serial clock (SCK) mark level	"L"
2	Transmission data output timing	SCK rising edge
3	Reception data sampling	SCK falling edge
4	Data length	5 to 16, 20, 24 and 32 bits

Register Setting

The register settings necessary for normal transfer (II) are listed below.

SCR:SPI*1 = 0, SMR:MD2 = 0, MD1 = 1, MD0 = 0, SCINV*1 = 1

During master operation: SCR:MS = 0, SMR:SCKE = 1

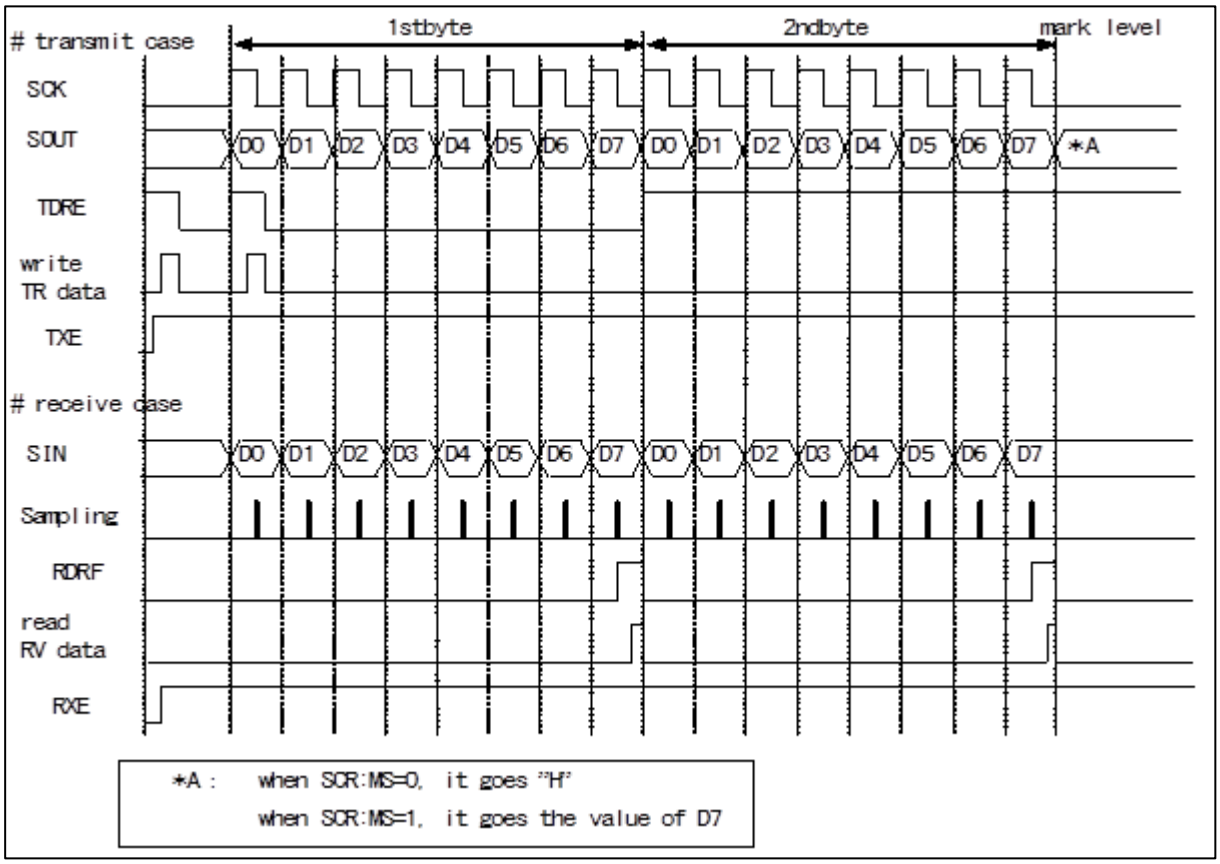
During slave operation: SCR:MS = 1, SMR:SCKE = 0

*1) The bit to be set differs depending on the condition. See Table 5-2.

Note:

- *Except for the above, set registers according to their use.*

Normal Transfer (II) Timing Chart (When the Serial Chip Select Pin is Not Used)



Master Operation (SCR:MS = 0, SMR:SCKE = 1, SCSCR:CSEN3 to CSEN0 = 0b0000)

Transmission Operation

1. Enabling serial data output (SMR:SOE = 1) and transmission operation (SCR:TXE = 1), while disabling reception operation (SCR:RXE = 0) and then writing transmission data to TDR causes SSR:TDRE to be set to 0. With this, transmission data is output in synchronization with a rising edge of the serial clock (SCK) output.
2. When the first 1 bit of transmission data is output, SSR:TDRE is set to 1. Thus, if transmission interrupts are enabled (SCR:TIE = 1), a transmission interrupt request is output. At this time, the second byte of transmission data can be written.

Reception Operation

1. If serial data output is disabled (SMR:SOE = 0), transmission is enabled (SCR:TXE = 1), and reception is enabled (SCR:RXE = 1), writing dummy data to TDR causes reception data to be sampled at a falling edge of the serial clock output (SCK).
2. If the last bit is received, SSR:RDRF is set to 1. At this time, if reception interrupts are enabled (SCR:RIE = 1), a reception interrupt request is output.
   At this time, reception data (RDR) can be read.
3. When reception data (RDR) is read, SSR:RDRF is cleared to "0".

Notes:

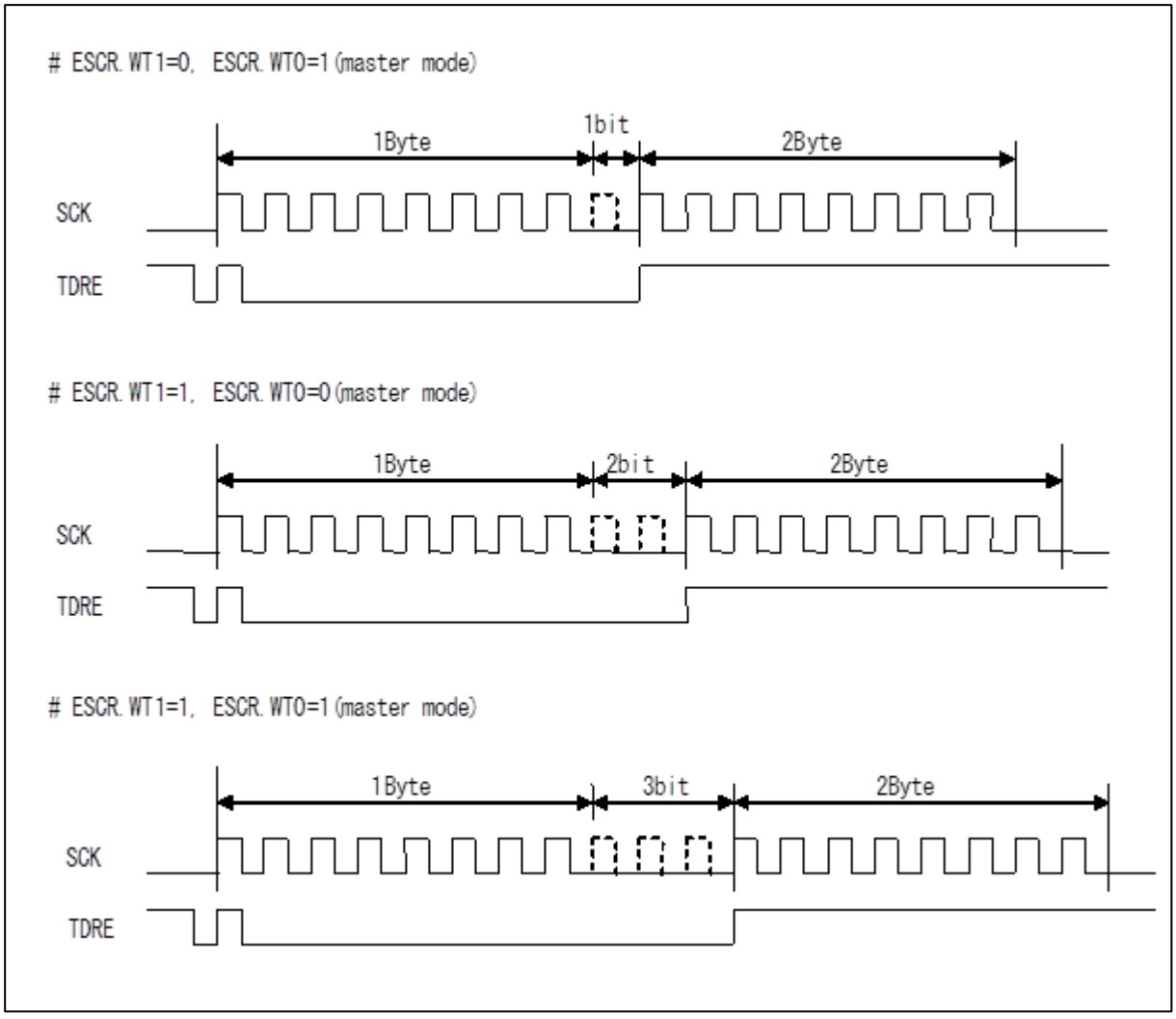
- *When performing reception only, write dummy data to TDR to output the serial clock (SCK).*
- *When the transmission/reception FIFOs are enabled, setting the number of frames to transfer in the FBYTE register causes as many serial clock (SCK) pulses as the number of frames to be output.*

Transmission/Reception Operations

1. To perform transmission/reception at the same time, enable serial data output (SMR:SOE = 1) and also transmission/reception (SCR:TXE, RXE = 1).
2. When transmission data is written to TDR, SSR:TDRE is set to 0, and transmission data is output in synchronization with a rising edge of the serial clock (SCK) output. When the first 1 bit of transmission data is output, SSR:TDRE is set to 1, and if transmission interrupts are enabled (SCR:TIE = 1), a transmission interrupt request is output. At this time, the second byte of transmission data can be written.
3. Reception data is sampled at a falling edge of the serial clock (SCK) output. If the last bit of reception data is received, SSR:RDRF is set to 1. If reception interrupts are enabled (SCR:RIE = 1), a reception interrupt request is output. At this time, reception data (RDR) can be read. When reception data is read, SSR:RDRF is cleared to "0".

Continuous Data Transmission or Reception Wait Operation

If settings other than (ESCR:WT1, ESCR.WT0) = (0, 0) are set for continuous data transmission or reception, a wait is inserted between the frames.



Slave Operation (SCR:MS = 1, SMR:SCKE = 0, SCSCR:CSEN0 = 0)

Transmission Operation

1. Enabling serial data output (SMR:SOE = 1) and also transmission (SCR:TXE = 1) and then writing transmission data to TDR causes SSR:TDRE to be set to 0. For this reason, transmission data is output in synchronization with a rising edge of the serial clock (SCK) input.
2. When the first 1 bit of transmission data is output, SSR:TDRE is set to 1. If transmission interrupts are enabled (SCR:TIE = 1), a transmission interrupt request is output. At this time, the second byte of transmission data can be written.

Note:

- *After transmission is enabled (SCR:TXE = 1), if the first writing of transmission data to TDR is performed when the serial clock (SCK) is not at the mark level, the first 1 bit of data is not output, and transmission is not performed normally. After transmission is enabled (SCR:TXE = 1), perform the first writing of transmission data to TDR with SSR:TBI = 1 when the serial clock (SCK) is at the mark level.*

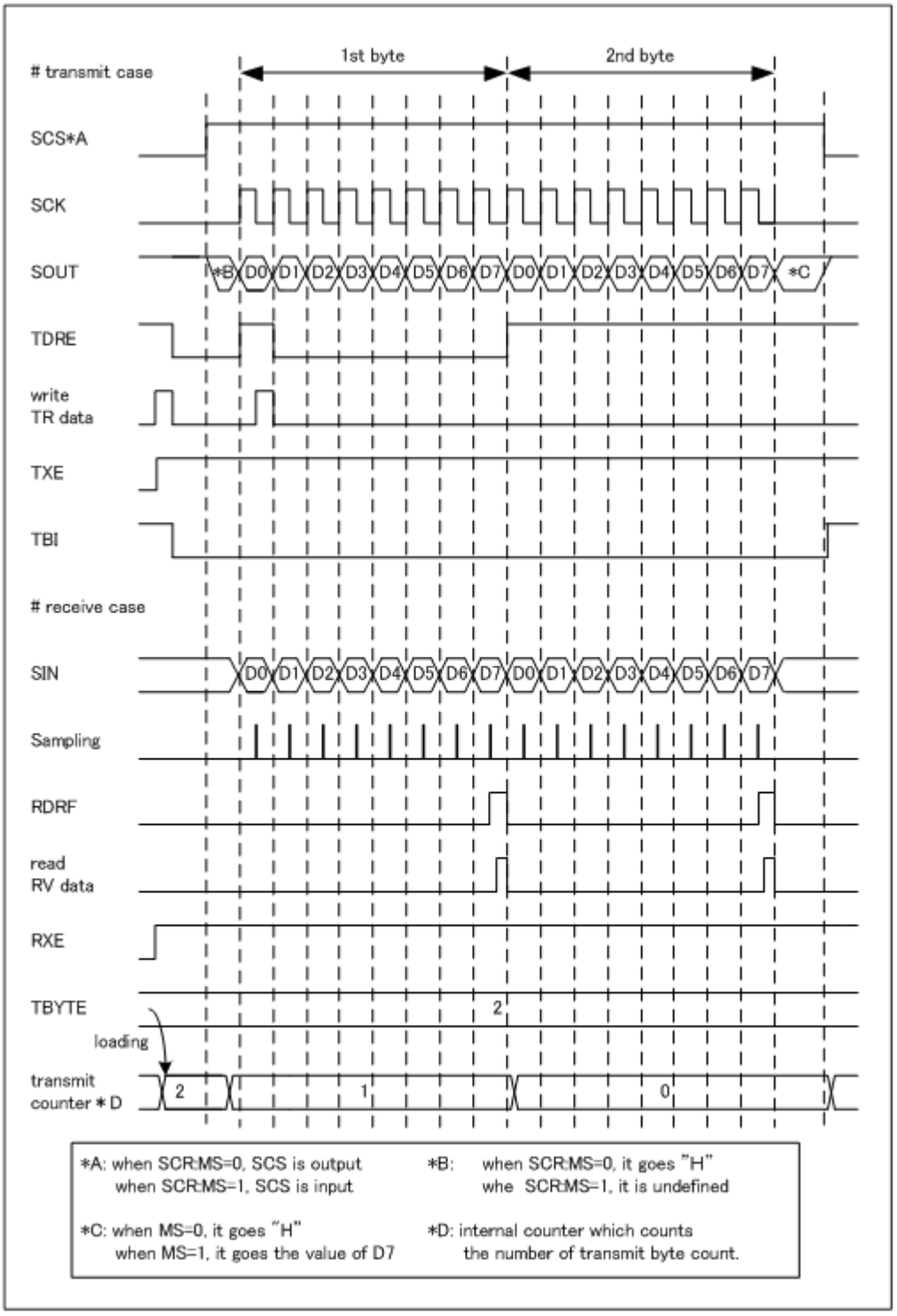
Reception Operation

1. If serial data output is disabled (SMR:SOE = 0) but reception is enabled (SCR:RXE = 1), reception data is sampled at a falling edge of the serial clock input (SCK).
2. If the last bit is received, SSR:RDRF is set to 1. If reception interrupts are enabled (SCR:RIE = 1), a reception interrupt request is output.
   At this time, reception data (RDR) can be read.
3. When reception data (RDR) is read, SSR:RDRF is cleared to "0".

Transmission/Reception Operations

1. To perform transmission/reception at the same time, enable serial data output (SMR:SOE = 1) and also transmission/reception (SCR:TXE, RXE = 1).
2. When transmission data is written to TDR, SSR:TDRE is set to 0, and transmission data is output in synchronization with a rising edge of the serial clock (SCK) input. When the first 1 bit of transmission data is output, SSR:TDRE is set to 1, and if transmission interrupts are enabled (SCR:TIE = 1), a transmission interrupt request is output. At this time, the second byte of transmission data can be written.
3. Reception data is sampled at a falling edge of the serial clock (SCK) input. If the last bit of reception data is received, SSR:RDRF is set to 1, and if reception interrupts are enabled (SCR:RIE = 1), a reception interrupt request is output. At this time, reception data (RDR) can be read. When reception data is read, SSR:RDRF is cleared to "0".

Normal Transfer (II) Timing Chart (When the Serial Chip Select Pin is Used)



Master Operation (SCR:MS = 0, SMR:SCKE = 1, SCSCR:CSOE = 1, SCSCR:CSENn = 1)

*: n is replaced by the serial chip select pin number used.

Transmission Operation

1. Enabling serial data output (SMR:SOE = 1) and transmission (SCR:TXE = 1), while disabling reception (SCR:RXE = 0) and then writing transmission data to TDR causes SSR:TDRE to be set to 0. Later, the serial chip select pin (SCS) becomes active, and after the elapse of the serial chip select pin setup time, serial clock output is started. After serial clock output is started, transmission data is output in synchronization with a rising edge of the serial clock (SCK) output.
2. When the first 1 bit of transmission data is output, SSR:TDRE is set to 1, and if transmission interrupts are enabled (SCR:TIE = 1), a transmission interrupt request is output. At this time, the second byte of transmission data can be written.
3. After the end of data transmission for which the count is set in TBYTE, the serial clock is stopped.
4. After the serial clock stops, the serial chip select pin (SCS) becomes inactive after the elapse of the serial chip select pin hold time. At this time, however, if the serial chip select active level (SCSCR:SCAM = 1) is retained, the serial chip select pin (SCS) retains the active state.

Reception Operation

1. Disabling serial data output (SMR:SOE = 0) while enabling transmission (SCR:TXE = 1) and also reception (SCR:RXE = 1), and then writing dummy data to TDR, causes the serial chip select pin (SCS) to become active, and after the elapse of the serial chip select pin setup time, serial clock output is started. After serial clock output starts, reception data is sampled at a falling edge of the serial clock output (SCK).
2. When the last bit is received, SSR:RDRF is set to 1, and if reception interrupts are enabled (SCR:RIE = 1), a reception interrupt request is output.

   At this time, reception data (RDR) can be read.
3. When reception data (RDR) is read, SSR:RDRF is cleared to "0".
4. After the end of data reception for which the count is set in TBYTE, the serial clock is stopped.
5. After the serial clock stops, the serial chip select pin (SCS) becomes inactive after the elapse of the serial chip select pin hold time. At this time, however, if the serial chip select active level (SCSCR:SCAM = 1) is retained, the serial chip select pin (SCS) retains the active state.

Notes:

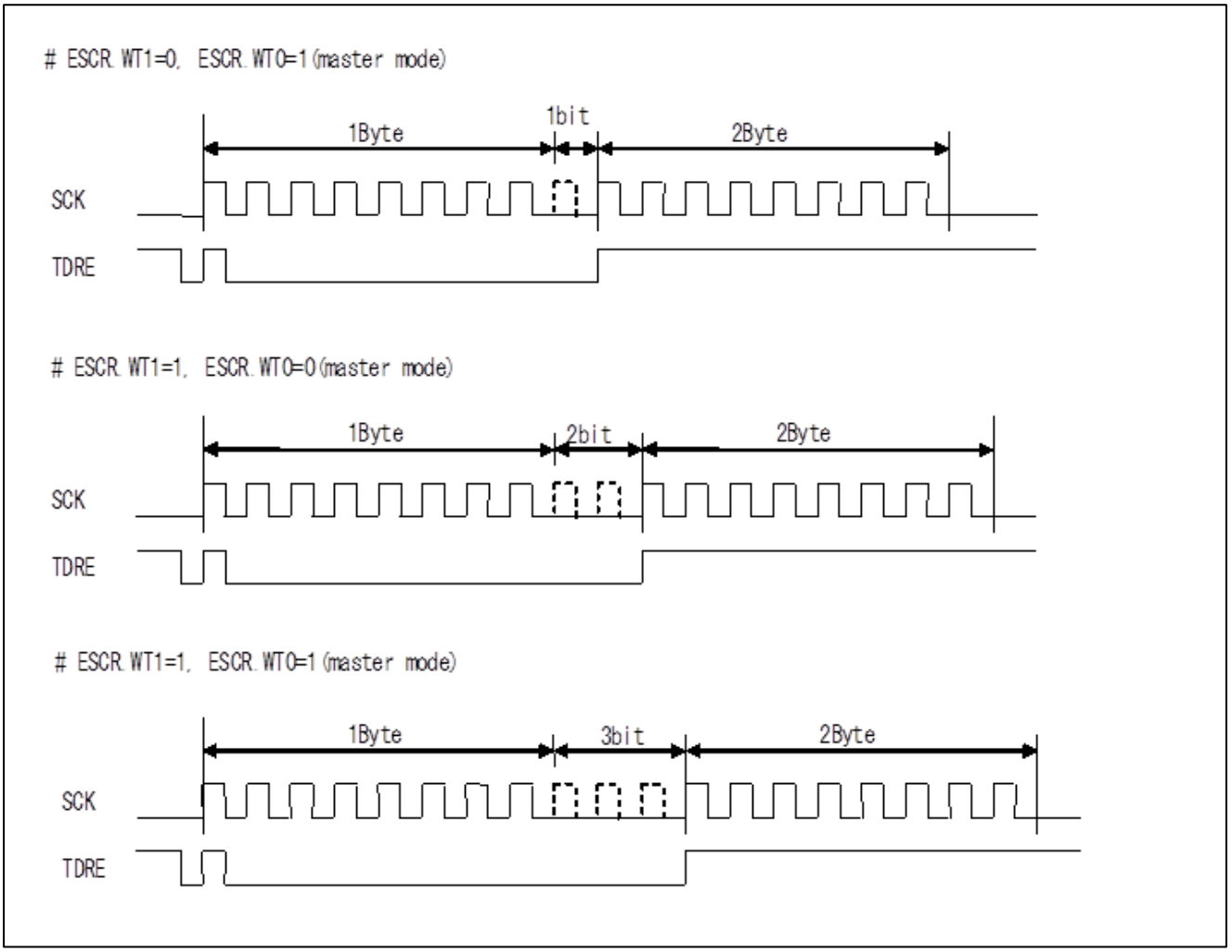
- *When performing reception only, write dummy data to TDR to output the serial clock (SCK).*
- *When the transmission/reception FIFOs are enabled, setting the number of frames to transfer in the FBYTE register causes as many serial clock (SCK) pulses as the number of frames to be output.*

Transmission/Reception Operations

1. To perform transmission/reception at the same time, enable serial data output (SMR:SOE = 1) and also transmission/reception (SCR:TXE, RXE = 1).
2. When transmission data is written to TDR, SSR:TDRE is set to 0. Later, the serial chip select pin (SCS) becomes active, and after the elapse of the serial chip select pin setup time, serial clock output is started. After serial clock output is started, transmission data is output in synchronization with a rising edge of the serial clock (SCK) output. When the first 1 bit of transmission data is output, SSR:TDRE is set to 1, and if transmission interrupts are enabled (SCR:TIE = 1), a transmission interrupt request is output. At this time, the second byte of transmission data can be written.
3. During transmission/reception, reception data is sampled at a falling edge of the serial clock (SCK) output. If the last bit of the reception data is received, SSR:RDRF is set to 1, and if reception interrupts are enabled (SCR:RIE = 1), a reception interrupt request is output. At this time, reception data (RDR) can be read. When reception data is read, SSR:RDRF is cleared to "0".
4. After the end of data transmission/reception for which the count is set in TBYTE, serial clock output is stopped.

5. After serial clock output is stopped, the serial chip select pin (SCS) becomes inactive after the elapse of the serial chip select pin hold time. At this time, however, if the serial chip select active level (SCSCR:SCAM = 1) is retained, the serial chip select pin (SCS) retains the active state.

Continuous Data Transmission or Reception Wait Operation



If settings other than (ESCR:WT1, ESCR.WT0) = (0, 0) are set for continuous data transmission or reception, a wait is inserted between the frames.

Slave Operation (SCR:MS = 1, SMR:SCKE = 0, SCSCR:CSEN0 = 1, SCSCR:CSOE = 0, SCSCR:SCAM = 0)

Transmission Operation

1. Enabling serial data output (SMR:SOE = 1) and also transmission (SCR:TXE = 1) and then writing transmission data to TDR causes SSR:TDRE to be set to 0.
2. If the serial chip select pin (SCS) becomes active, transmission is started, and transmission data is output in synchronization with a rising edge of the serial clock (SCK) input.
3. When the first 1 bit of transmission data is output, SSR:TDRE is set to 1, and if transmission interrupts are enabled (SCR:TIE = 1), a transmission interrupt request is output. At this time, the second byte of transmission data can be written.
4. If the serial chip select pin (SCS) becomes inactive, transmission operation is stopped, and the serial output pin (SOUT) becomes "H".

Note:

- *After transmission is enabled (SCR:TXE = 1), if the first writing of transmission data to TDR is performed when the serial clock (SCK) is not at the mark level, the first 1 bit of data is not output, and transmission is not performed normally. After transmission is enabled (SCR:TXE = 1), perform the first writing of transmission data to TDR with SSR:TBI = 1 when the serial clock (SCK) is at the mark level.*

Reception Operation

1. If serial data output is disabled (SMR:SOE = 0) but reception is enabled (SCR:RXE = 1), and the serial chip select pin (SCS) becomes active, reception starts, and reception data is sampled at a falling edge of the serial clock input (SCK).
2. When the last bit is received, SSR:RDRF is set to 1, and if reception interrupts are enabled (SCR:RIE = 1), a reception interrupt request is output.
3. At this time, reception data (RDR) can be read.
4. When reception data (RDR) is read, SSR:RDRF is cleared to "0".
5. If the serial chip select pin (SCS) becomes inactive, reception is stopped.

Transmission/Reception Operations

1. To perform transmission/reception at the same time, enable serial data output (SMR:SOE = 1) and also transmission/reception (SCR:TXE, RXE = 1).
2. When transmission data is written to TDR, SSR:TDRE is set to 0. Later, if the serial chip select pin (SCS) becomes active, transmission/reception operations are started, and transmission data is output in synchronization with a rising edge of the serial clock (SCK) input. When the first 1 bit of transmission data is output, SSR:TDRE is set to 1, and if transmission interrupts are enabled (SCR:TIE = 1), a transmission interrupt request is output. At this time, the second byte of transmission data can be written.
3. During transmission/reception, reception data is sampled at a falling edge of the serial clock (SCK) input. When the last bit of the reception data is received, SSR:RDRF is set to 1, and if reception interrupts are enabled (SCR:RIE = 1), a reception interrupt request is output. At this time, reception data (RDR) can be read. When reception data is read, SSR:RDRF is cleared to "0".
4. If the serial chip select pin (SCS) becomes inactive, transmission/reception operations stop, and the serial output pin (SOUT) becomes "H".

3.3. SPI Transfer (I)

Features

	Item	Description
1	Serial clock (SCK) mark level	"H"
2	Transmission data output timing	SCK rising edge
3	Reception data sampling	SCK falling edge
4	Data length	5 to 16, 20, 24 and 32 bits

Register Setting

The register settings necessary for SPI transfer (I) are listed below.

SCR:SPI*1 = 1, SMR:MD2 = 0, MD1 = 1, MD0 = 0, SCINV*1 = 0

During master operation: SCR:MS = 0, SMR:SCKE = 1

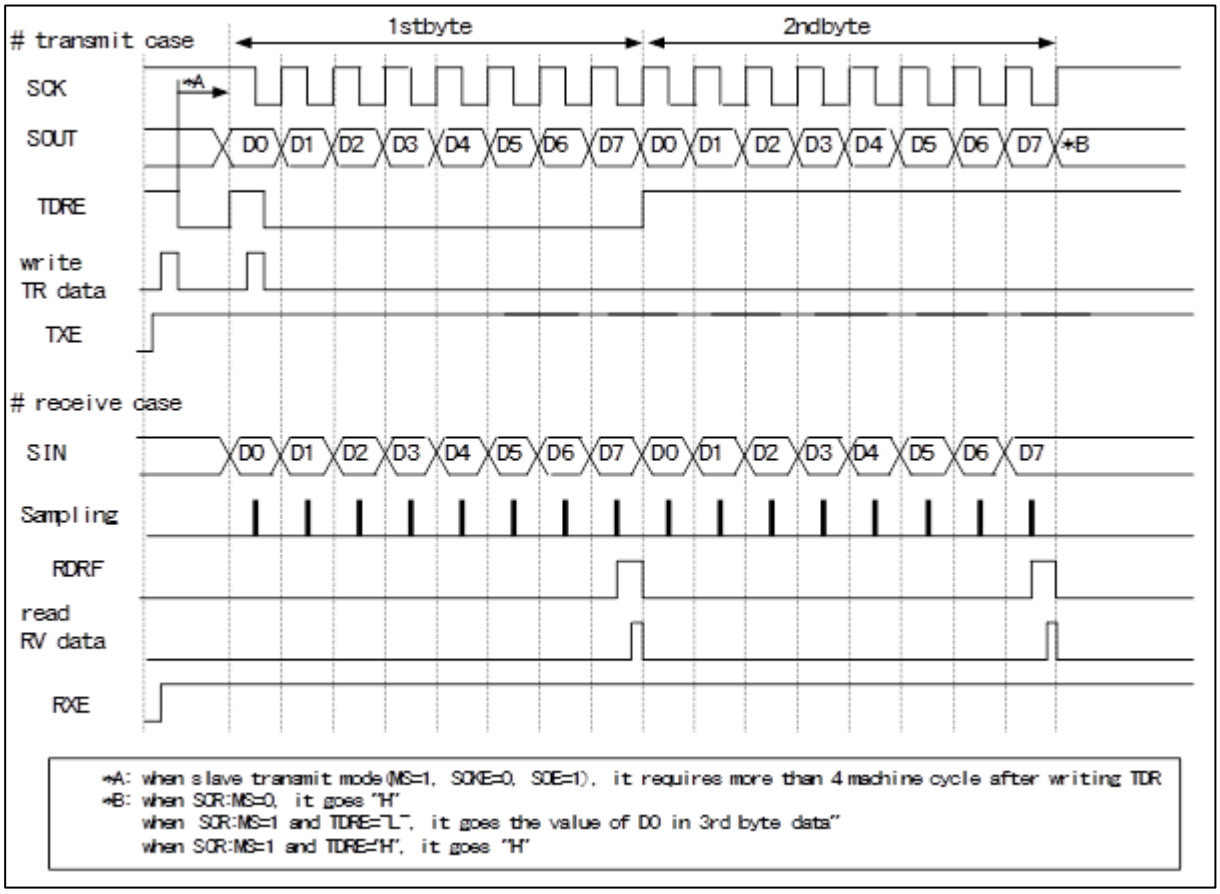
During slave operation: SCR:MS = 1, SMR:SCKE = 0

*1) The bit to be set differs depending on the condition. See Table 5-2.

Note:

- *Except for those above, set registers according to their use.*

SPI Transfer (I) Timing Chart (When the Serial Chip Select Pin is Not Used)



Master Operation (SCR:MS = 0, SMR:SCKE = 1, SCSCR:CSEN3 to CSEN0 = 0x0000)

Transmission Operation

1. Enabling serial data output (SMR:SOE = 1), and also transmission operation (SCR:TXE = 1), while disabling reception operation (SCR:RXE = 0) and then writing transmission data to TDR causes SSR:TDRE to be set to 0. This causes the first bit to be output. Later, transmission data is output in synchronization with a rising edge of the serial clock (SCK) output.
2. Half a cycle before a falling edge of the first serial clock (SCK) output, SSR:TDRE is set to 1. Thus, if transmission interrupts are enabled (SCR:TIE = 1), a transmission interrupt request is output. At this time, the second byte of transmission data can be written.

Reception Operation

1. If serial data output is disabled (SMR:SOE = 0), and transmission is enabled (SCR:TXE = 1), and also reception (SCR:RXE = 1), writing dummy data to TDR causes reception data to be sampled at a falling edge of the serial clock (SCK) output.
2. When the last bit is received, SSR:RDRF is set to 1. At this time, if reception interrupts are enabled (SCR:RIE = 1), a reception interrupt request is output.
   At this time, reception data (RDR) can be read.
3. When reception data (RDR) is read, SSR:RDRF is cleared to "0".

Notes:

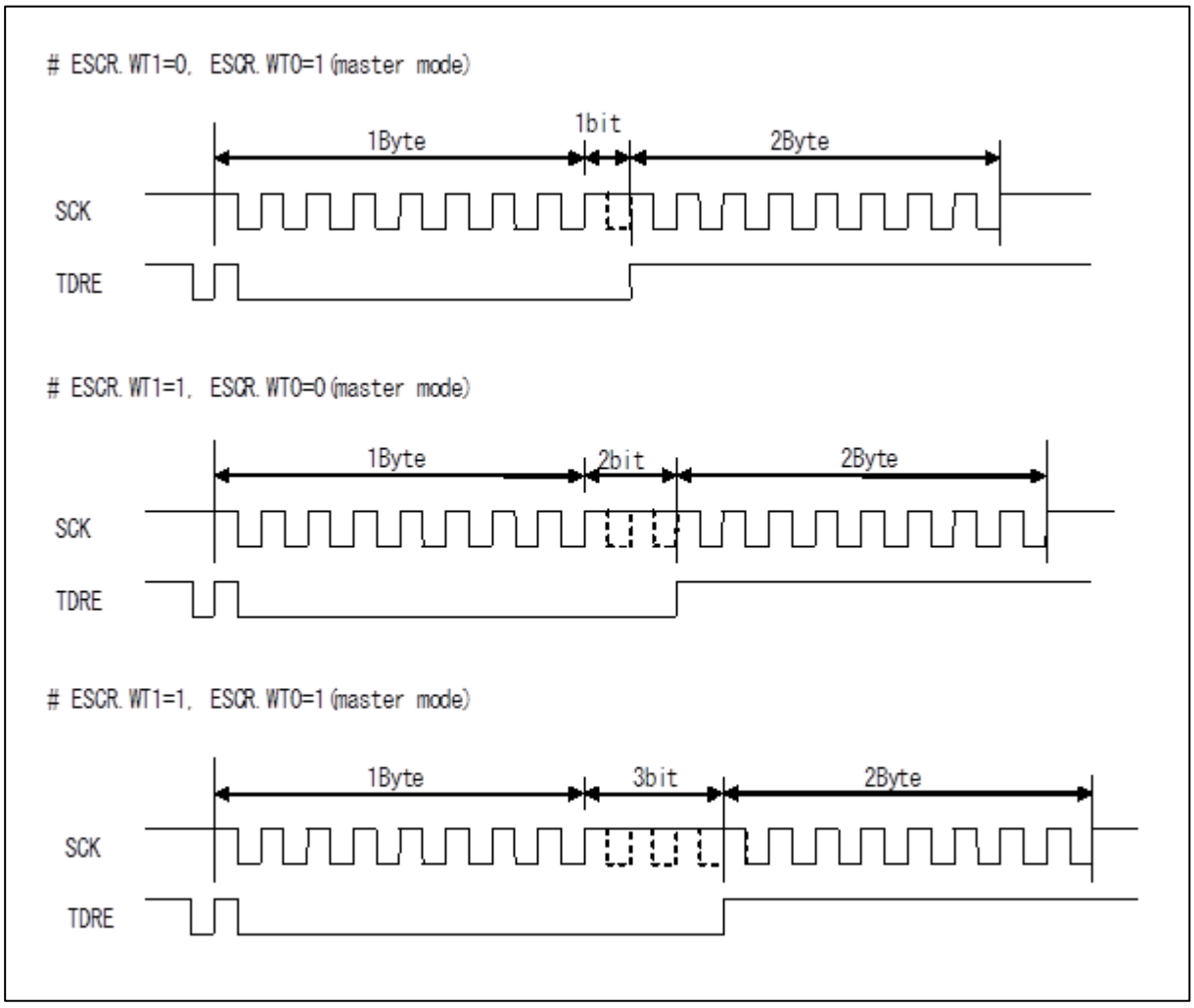
- *When performing reception only, write dummy data to TDR to output the serial clock (SCK).*
- *When the transmission/reception FIFOs are enabled, setting the number of frames to transfer in the FBYTE register causes as many serial clock (SCK) pulses as the number of frames to be output.*

Transmission/Reception Operations

1. To perform transmission/reception at the same time, enable serial data output (SMR:SOE = 1) and also transmission/reception (SCR:TXE, RXE = 1).
2. When transmission data is written to TDR, SSR:TDRE is set to 0, and the first bit is output. Later, transmission data is output in synchronization with a rising edge of the serial clock (SCK) output. Half a cycle before a falling edge of the first serial clock, SSR:TDRE is set to 1, and if transmission interrupts are enabled (SCR:TIE = 1), a transmission interrupt request is output. At this time, the second byte of transmission data can be written.
3. Reception data is sampled at a falling edge of the serial clock (SCK) output. When the last bit of reception data is received, SSR:RDRF is set to 1. If reception interrupts are enabled (SCR:RIE = 1), a reception interrupt request is output. At this time, reception data (RDR) can be read. When reception data is read, SSR:RDRF is cleared to "0".

Continuous Data Transmission or Reception Wait Operation

If settings other than (ESCR:WT1, ESCR.WT0) = (0, 0) are set for continuous data transmission or reception, a wait is inserted between the frames.



Slave Operation (SCR:MS = 1, SMR:SCKE = 0, SCSCR:CSEN0 = 0)

Transmission Operation

1. Enabling serial data output (SMR:SOE = 1) and also transmission (SCR:TXE = 1) and then writing transmission data to TDR causes SSR:TDRE to be set to 0. Thus, the first bit is output. Later, transmission data is output in synchronization with a rising edge of the serial clock (SCK) output.
2. When the first 1 bit of transmission data is output, SSR:TDRE is set to 1. If transmission interrupts are enabled (SCR:TIE = 1), a transmission interrupt request is output. At this time, the second byte of transmission data can be written.

Note:

- *After transmission is enabled (SCR:TXE = 1), if the first writing of transmission data to TDR is performed when the serial clock (SCK) is not at the mark level, the first 1 bit of data is not output, and transmission is not performed normally. After transmission is enabled (SCR:TXE = 1), perform the first writing of transmission data to TDR with SSR:TBI = 1 when the serial clock (SCK) is at the mark level.*

Reception Operation

1. If serial data output is disabled (SMR:SOE = 0) but reception is enabled (SCR:RXE = 1), reception data is sampled at a falling edge of the serial clock input (SCK).
2. When the last bit is received, SSR:RDRF is set to 1. If reception interrupts are enabled (SCR:RIE = 1), a reception interrupt request is output.
   At this time, reception data (RDR) can be read.
3. When reception data (RDR) is read, SSR:RDRF is cleared to "0".

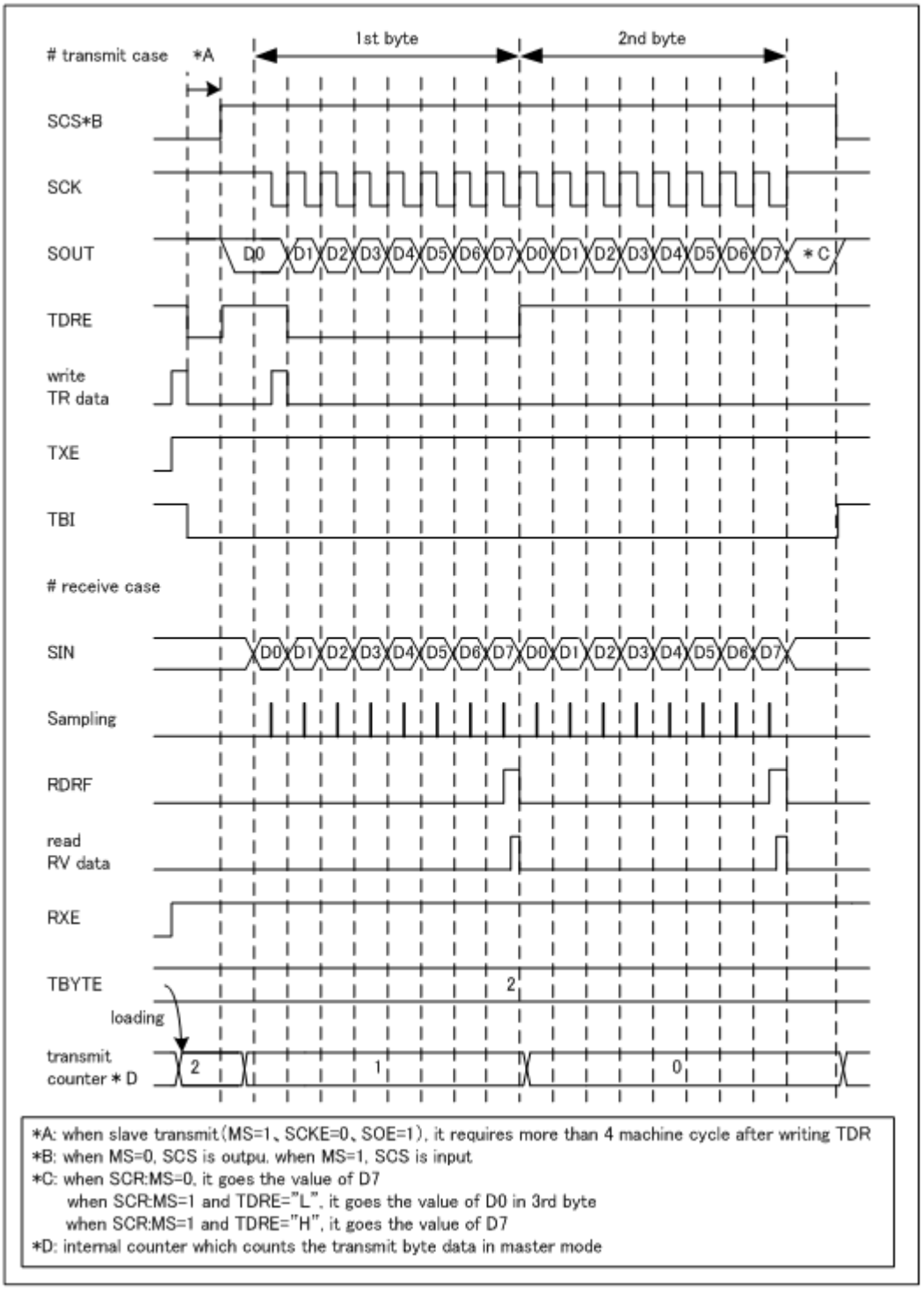
Transmission/Reception Operations

1. To perform transmission/reception at the same time, enable serial data output (SMR:SOE = 1) and also transmission/reception (SCR:TXE, RXE = 1).
2. When transmission data is written to TDR, SSR:TDRE is set to 0, and the first bit is output. Later, transmission data is output in synchronization with a rising edge of the serial clock (SCK) input. When the first 1 bit of transmission data is output, SSR:TDRE is set to 1, and if transmission interrupts are enabled (SCR:TIE = 1), a transmission interrupt request is output. At this time, the second byte of transmission data can be written.
3. Reception data is sampled at a falling edge of the serial clock (SCK) input. When the last bit of the reception data is received, SSR:RDRF is set to 1, and if reception interrupts are enabled (SCR:RIE = 1), a reception interrupt request is output. At this time, reception data (RDR) can be read. When reception data is read, SSR:RDRF is cleared to "0".

Continuous Switching from Reception Operation to Transmission Operation

1. Disable serial data output (SMR:SOE = 0), enable reception interrupts (SCR:RIE = 1) and reception operation (SCR:RXE = 1), and also transmission operation (SCR:TXE = 1). If dummy data is written to TDR when the serial clock (SCK) is at the mark level, reception data is sampled at a falling edge of the serial clock input (SCK).
2. To continue reception, write dummy data to TDR between a reception interrupt request and the next rising edge of the serial clock (SCK).
3. To switch from reception to transmission, enable serial data output (SMR:SOE = 1), disable reception interrupts (SCR:RIE = 0), and also reception (SCR:RXE = 0) between a reception interrupt request and the next rising edge of the serial clock (SCK), and write transmission data to TDR; after the end of reception, transmission data is output in synchronization with a rising edge of the serial clock.

SPI Transfer (I) Timing Chart (When the Serial Chip Select Pin is Used)


transmit case
*A
1st byte
2nd byte
SCS*B
SCK
SOUT
D0 D1 D2 D3 D4 D5 D6 D7 D0 D1 D2 D3 D4 D5 D6 D7 *C
TDRE
write
TR data
TXE
TBI
receive case
SIN
D0 D1 D2 D3 D4 D5 D6 D7 D0 D1 D2 D3 D4 D5 D6 D7
Sampling
RDRF
read
RV data
RXE
TBYTE
2
loading
transmit
counter * D
2
1
0
*A: when slave transmit (MS=1、SCKE=0、SOE=1), it requires more than 4 machine cycle after writing TDR
*B: when MS=0, SCS is outpu. when MS=1, SCS is input
*C: when SCR:MS=0, it goes the value of D7
when SCR:MS=1 and TDRE="L", it goes the value of D0 in 3rd byte
when SCR:MS=1 and TDRE="H", it goes the value of D7
*D: internal counter which counts the transmit byte data in master mode

Master Operation (SCR:MS = 0, SMR:SCKE = 1, SCSCR:CSOE = 1, SCSCR:CSENn* = 1)

*: n is replaced by the serial chip select pin number used.

Transmission Operation

1. Enabling serial data output (SMR:SOE = 1) and also transmission operation (SCR:TXE = 1), while disabling reception (SCR:RXE = 0) and then writing transmission data to TDR causes SSR:TDRE to be set to 0. Later, at the same time as the first bit is output, the serial chip select pin (SCS) becomes active, and after the elapse of the serial chip select pin setup time, serial clock output is started. After the start of serial clock output, transmission data is output in synchronization with a rising edge of the serial clock (SCK) output.
2. Half a cycle before a falling edge of the first serial clock (SCK) output, SSR:TDRE is set to 1, and if transmission interrupts are enabled (SCR:TIE = 1), a transmission interrupt request is output. At this time, the second byte of transmission data can be written.
3. After the end of data transmission for which the count is set in TBYTE, serial clock output is stopped.
4. After the serial clock output is stopped, the serial chip select pin (SCS) becomes inactive after the elapse of the serial chip select pin hold time. At this time, however, if the serial chip select active level (SCSCR:SCAM = 1) is retained, the serial chip select pin (SCS) retains the active state.

Reception Operation

1. Disabling serial data output (SMR:SOE = 0), while enabling transmission (SCR:TXE = 1) and also reception (SCR:RXE = 1), and then writing dummy data to TDR causes the serial chip select pin (SCS) to become active, and after the elapse of the serial chip select pin setup time, serial clock output is started. After serial clock output starts, reception data is sampled at a falling edge of the serial clock (SCK) output.
2. When the last bit is received, SSR:RDRF is set to 1, and if reception interrupts are enabled (SCR:RIE = 1), a reception interrupt request is output.

   At this time, reception data (RDR) can be read.
3. When reception data (RDR) is read, SSR:RDRF is cleared to "0".
4. After the end of data reception for which the count is set in TBYTE, serial clock output is stopped.
5. After serial clock output is stopped, the serial chip select pin (SCS) becomes inactive after the elapse of the serial chip select pin hold time. At this time, however, if the serial chip select active level (SCSCR:SCAM = 1) is retained, the serial chip select pin (SCS) retains the active state.

Notes:

- *When performing reception only, write dummy data to TDR to output the serial clock (SCK).*
- *When the transmission/reception FIFOs are enabled, setting the number of frames to transfer in the FBYTE register causes as many serial clock (SCK) pulses as the number of frames to be output.*

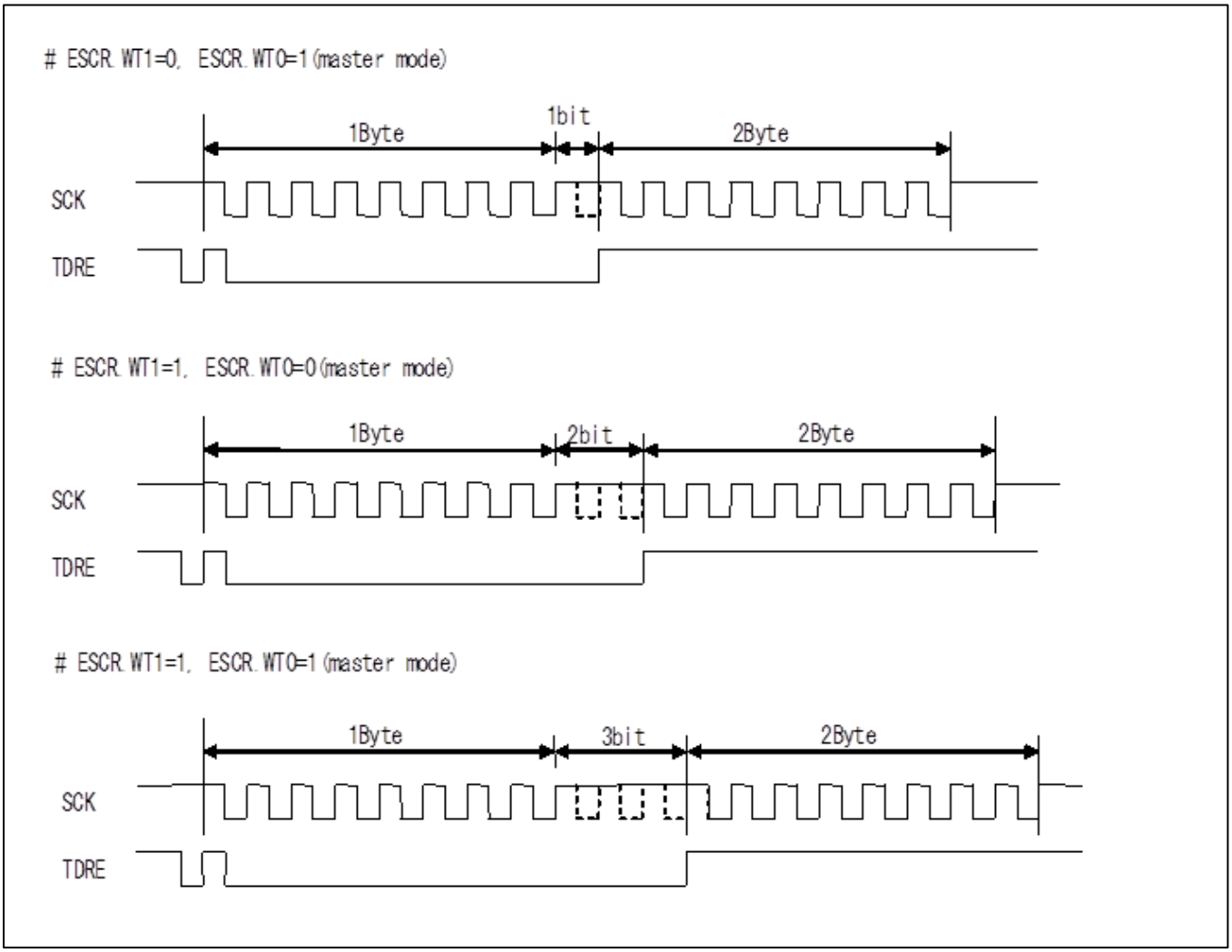
Transmission/Reception Operations

1. To perform transmission/reception at the same time, enable serial data output (SMR:SOE = 1) and also transmission/reception (SCR:TXE, RXE = 1).
2. When transmission data is written to TDR, SSR:TDRE is set to 0. Later, at the same time as the first bit is output, the serial chip select pin (SCS) becomes active, and after the elapse of the serial chip select pin setup time, serial clock output is started. After the start of serial clock output, transmission data is output in synchronization with a rising edge of the serial clock (SCK) output. Half a cycle before a falling edge of the first serial clock, SSR:TDRE is set to 1, and if transmission interrupts are enabled (SCR:TIE = 1), a transmission interrupt request is output. At this time, the second byte of transmission data can be written.
3. Reception data is sampled at a falling edge of the serial clock (SCK) output. When the last bit of reception data is received, SSR:RDRF is set to 1, and if reception interrupts are enabled (SCR:RIE = 1), a reception interrupt request is output. At this time, reception data (RDR) can be read. When reception data is read, SSR:RDRF is cleared to "0".

4. After the end of data transmission/reception for which the count is set in TBYTE, serial clock output is stopped.
5. After serial clock output is stopped, the serial chip select pin (SCS) becomes inactive after the elapse of the serial chip select pin hold time. At this time, however, if the serial chip select active level (SCSCR:SCAM = 1) is retained, the serial chip select pin (SCS) retains the active state.

Continuous Data Transmission or Reception Wait Operation

If settings other than (ESCR:WT1, ESCR.WT0) = (0, 0) are set for continuous data transmission or reception, a wait is inserted between the frames.



Slave Operation (SCR:MS = 1, SMR:SCKE = 0, SCSCR:CSEN = 1, SCSCR:SCAM = 0)

Transmission Operation

1. Enabling serial data output (SMR:SOE = 1) and also transmission (SCR:TXE = 1) and then writing transmission data to TDR causes SSR:TDRE to be set to 0.
2. If the serial chip select pin (SCS) becomes active, transmission is started, and the first bit is output. After the start of transmission, transmission data is output in synchronization with a rising edge of the serial clock (SCK) output.

3. When the first 1 bit of transmission data is output, SSR:TDRE is set to 1, and if transmission interrupts are enabled (SCR:TIE = 1), a transmission interrupt request is output. At this time, the second byte of transmission data can be written.
4. If the serial chip select pin (SCS) becomes inactive, transmission is stopped, and the serial output pin (SOUT) becomes "H".

Note:

- *After transmission is enabled (SCR:TXE = 1), if the first writing of transmission data to TDR is performed when the serial clock (SCK) is not at the mark level, the first 1 bit of data is not output, and transmission is not performed normally. After transmission is enabled (SCR:TXE = 1), perform the first writing of transmission data to TDR when the serial clock (SCK) is at the mark level.*

Reception Operation

1. If serial data output is disabled (SMR:SOE = 0) and reception is enabled (SCR:RXE = 1), if the serial chip select pin (SCS) becomes active, reception starts, and reception data is sampled at a falling edge of the serial clock input (SCK).
2. If the last bit is received, SSR:RDRF is set to 1, and if reception interrupts are enabled (SCR:RIE = 1), a reception interrupt request is output.
3. At this time, reception data (RDR) can be read.
4. When reception data (RDR) is read, SSR:RDRF is cleared to "0".
5. If the serial chip select pin (SCS) becomes inactive, reception is stopped.

Transmission/Reception Operations

1. To perform transmission/reception at the same time, enable serial data output (SMR:SOE = 1) and also transmission/reception (SCR:TXE, RXE = 1).
2. When transmission data is written to TDR, SSR:TDRE is set to 0. If the serial chip select pin (SCS) becomes active, transmission/reception starts, and the first bit is output. After the start of transmission/reception, transmission data is output in synchronization with a rising edge of the serial clock (SCK) input. When the first 1 bit of transmission data is output, SSR:TDRE is set to 1, and if transmission interrupts are enabled (SCR:TIE = 1), a transmission interrupt request is output. At this time, the second byte of transmission data can be written.
3. Reception data is sampled at a falling edge of the serial clock (SCK) input. When the last bit of reception data is received, SSR:RDRF is set to 1, and if reception interrupts are enabled (SCR:RIE = 1), a reception interrupt request is output. At this time, reception data (RDR) can be read. When reception data is read, SSR:RDRF is cleared to "0".
4. If the serial chip select pin (SCS) becomes inactive, transmission/reception stops, and the serial output pin (SOUT) becomes "H".

3.4. SPI Transfer (II)

Features

	Item	Description
1	Serial clock (SCK) mark level	"L"
2	Transmission data output timing	SCK falling edge
3	Reception data sampling	SCK rising edge
4	Data length	5 to 16, 20, 24 and 32 bits

Register Setting

The register settings necessary for SPI transfer (II) are listed below.

SCR:SPI*1 = 1, SMR:MD2 = 0, MD1 = 1, MD0 = 0, SCINV*1 = 1

During master operation: SCR:MS = 0, SMR:SCKE = 1

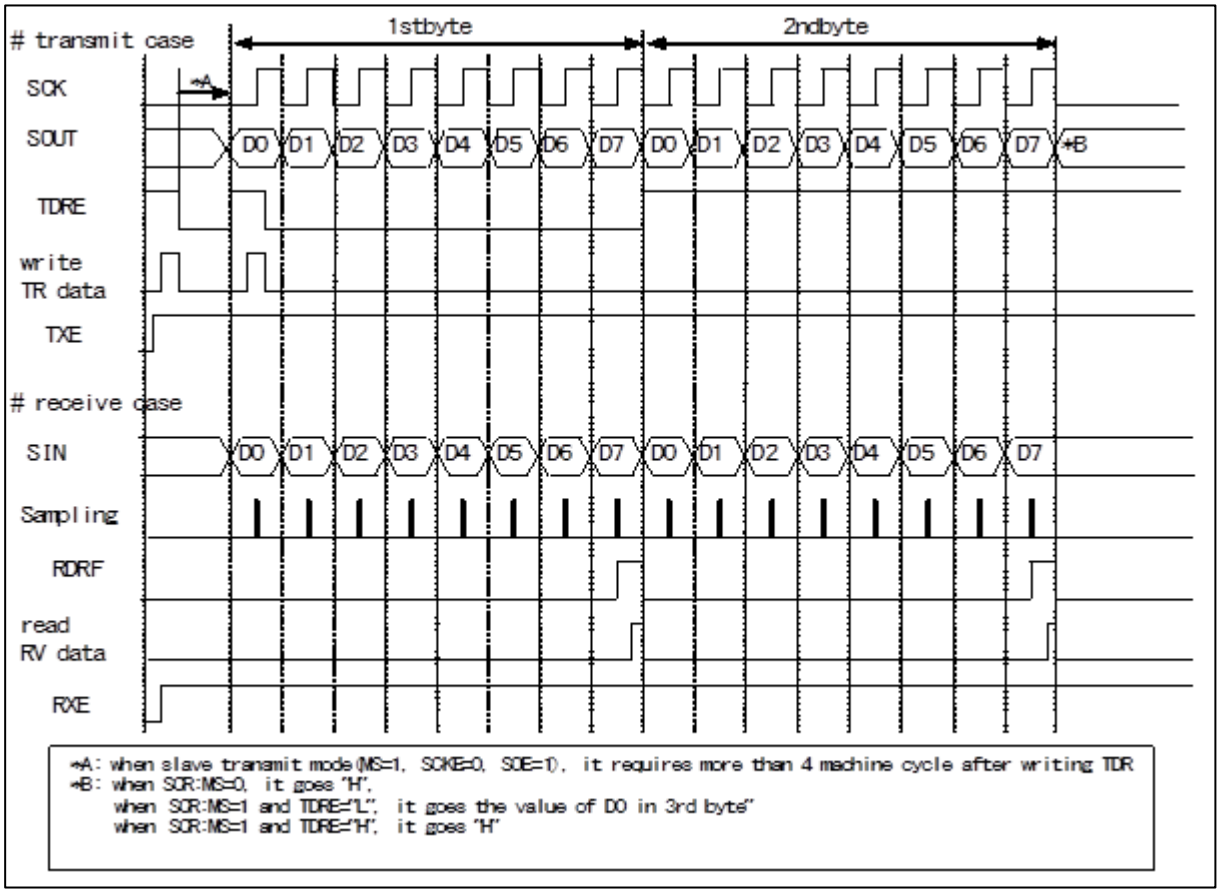
During slave operation: SCR:MS = 1, SMR:SCKE = 0

*1) The bit to be set differs depending on the condition. See Table 5-2.

Note:

- *Except for those described above, set the registers according to their use.*

SPI Transfer (II) Timing Chart (When the Serial Chip Select Pin is Not Used)



Master Operation (Setting to SCR:MS = 0, SMR:SCKE = 1)

Transmission Operation

1. Enabling serial data output (SMR:SOE = 1) and transmission (SCR:TXE = 1), while disabling reception (SCR:RXE = 0) and then writing transmission data to TDR causes SSR:TDRE to be set to 0. With this, transmission data is output in synchronization with a falling edge of the serial clock (SCK) output.
2. Half a cycle before a rising edge of the first serial clock (SCK) output, SSR:TDRE is set to 1. Thus, if transmission interrupts are enabled (SCR:TIE = 1), a transmission interrupt request is output. At this time, the second byte of transmission data can be written.

Reception Operation

1. If serial data output is disabled (SMR:SOE = 0), while transmission (SCR:TXE = 1) and reception (SCR:RXE = 1) are enabled, writing dummy data to TDR causes reception data to be sampled at a rising edge of the serial clock output (SCK).
2. When the last bit is received, SSR:RDRF is set to 1. At this time, if reception interrupts are enabled (SCR:RIE = 1), a reception interrupt request is output.
   At this time, reception data (RDR) can be read.
3. When reception data (RDR) is read, SSR:RDRF is cleared to "0".

Notes:

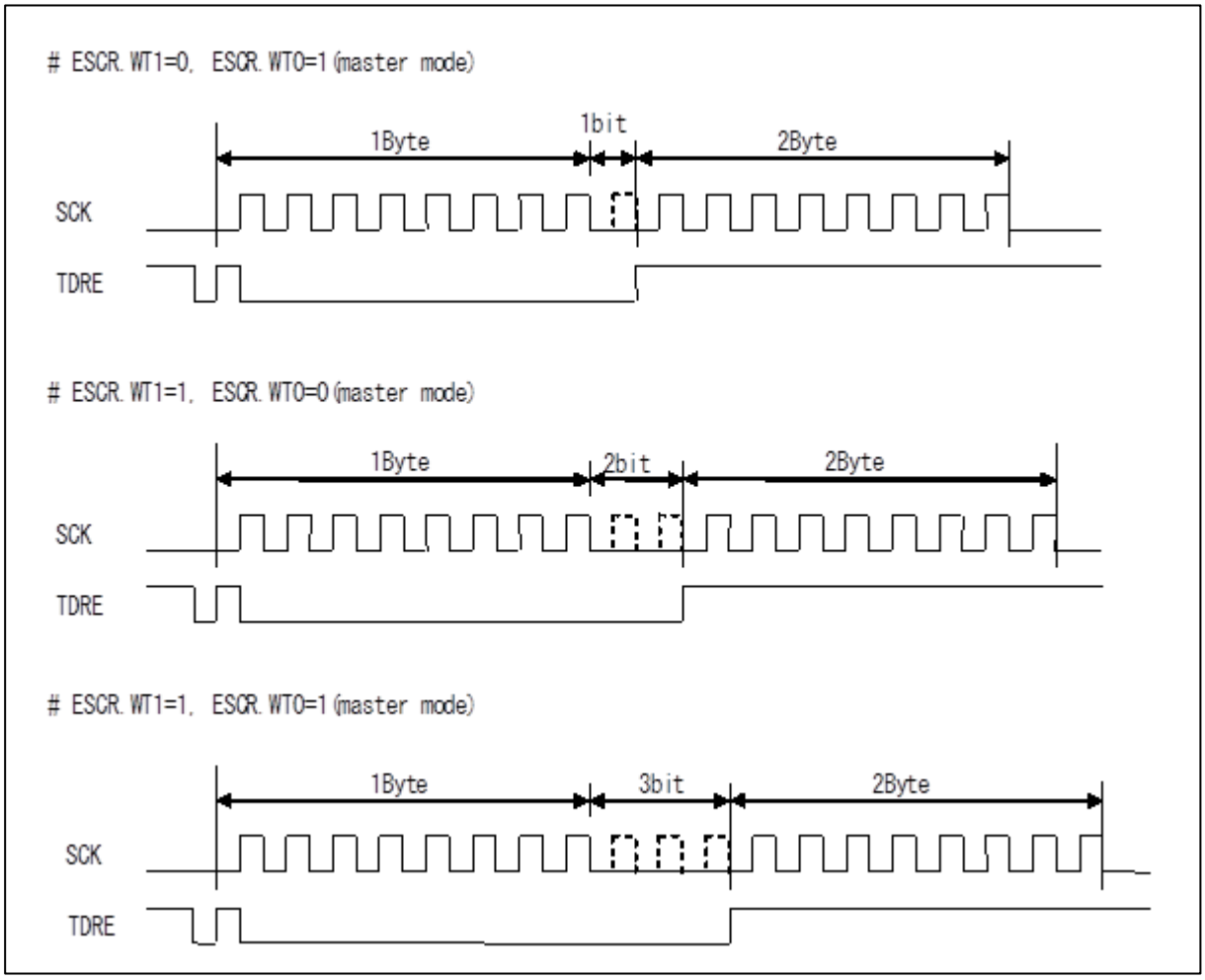
- *When performing reception only, write dummy data to TDR to output the serial clock (SCK).*
- *When the transmission/reception FIFOs are enabled, setting the number of frames to transfer in the FBYTE register causes as many serial clock (SCK) pulses as the number of frames to be output.*

Transmission/Reception Operations

1. To perform transmission/reception at the same time, enable serial data output (SMR:SOE = 1) and also transmission/reception (SCR:TXE, RXE = 1).
2. When transmission data is written to TDR, SSR:TDRE is set to 0, and the first bit is output. Later, transmission data is output in synchronization with a falling edge of the serial clock (SCK) output. Half a cycle before a rising edge of the first serial clock, SSR:TDRE is set to 1, and if transmission interrupts are enabled (SCR:TIE = 1), a transmission interrupt request is output. At this time, the second byte of transmission data can be written.
3. Reception data is sampled at a rising edge of the serial clock (SCK) output. When the last bit of reception data is received, SSR:RDRF is set to 1. If reception interrupts are enabled (SCR:RIE = 1), a reception interrupt request is output. At this time, reception data (RDR) can be read. When reception data is read, SSR:RDRF is cleared to "0".

Continuous Data Transmission or Reception Wait Operation

If settings other than (ESCR:WT1, ESCR.WT0) = (0, 0) are set for continuous data transmission or reception, a wait is inserted between the frames.



Slave Operation (Setting to SCR:MS = 1, SMR:SCKE = 0)

Transmission Operation

1. Enabling serial data output (SMR:SOE = 1) and transmission (SCR:TXE = 1) and then writing transmission data to TDR causes SSR:TDRE to be set to 0. Thus, the first bit is output. Later, transmission data is output in synchronization with a falling edge of the serial clock (SCK) input.
2. When the first 1 bit of transmission data is output, SSR:TDRE is set to 1. If transmission interrupts are enabled (SCR:TIE = 1), a transmission interrupt request is output. At this time, the second byte of transmission data can be written.

Note:

- *After transmission is enabled (SCR:TXE = 1), if the first writing of transmission data to TDR is performed when the serial clock (SCK) is not at the mark level, the first 1 bit of data is not output, and transmission is not performed normally. After transmission is enabled (SCR:TXE = 1), perform the first writing of transmission data to TDR with SSR:TBI = 1 when the serial clock (SCK) is at the mark level.*

Reception Operation

1. If serial data output is disabled (SMR:SOE = 0) and reception operation is enabled (SCR:RXE = 1), reception data is sampled at a rising edge of the serial clock input (SCK).
2. When the last bit is received, SSR:RDRF is set to 1. If reception interrupts are enabled (SCR:RIE = 1), a reception interrupt request is output.
   At this time, reception data (RDR) can be read.
3. When reception data (RDR) is read, SSR:RDRF is cleared to "0".

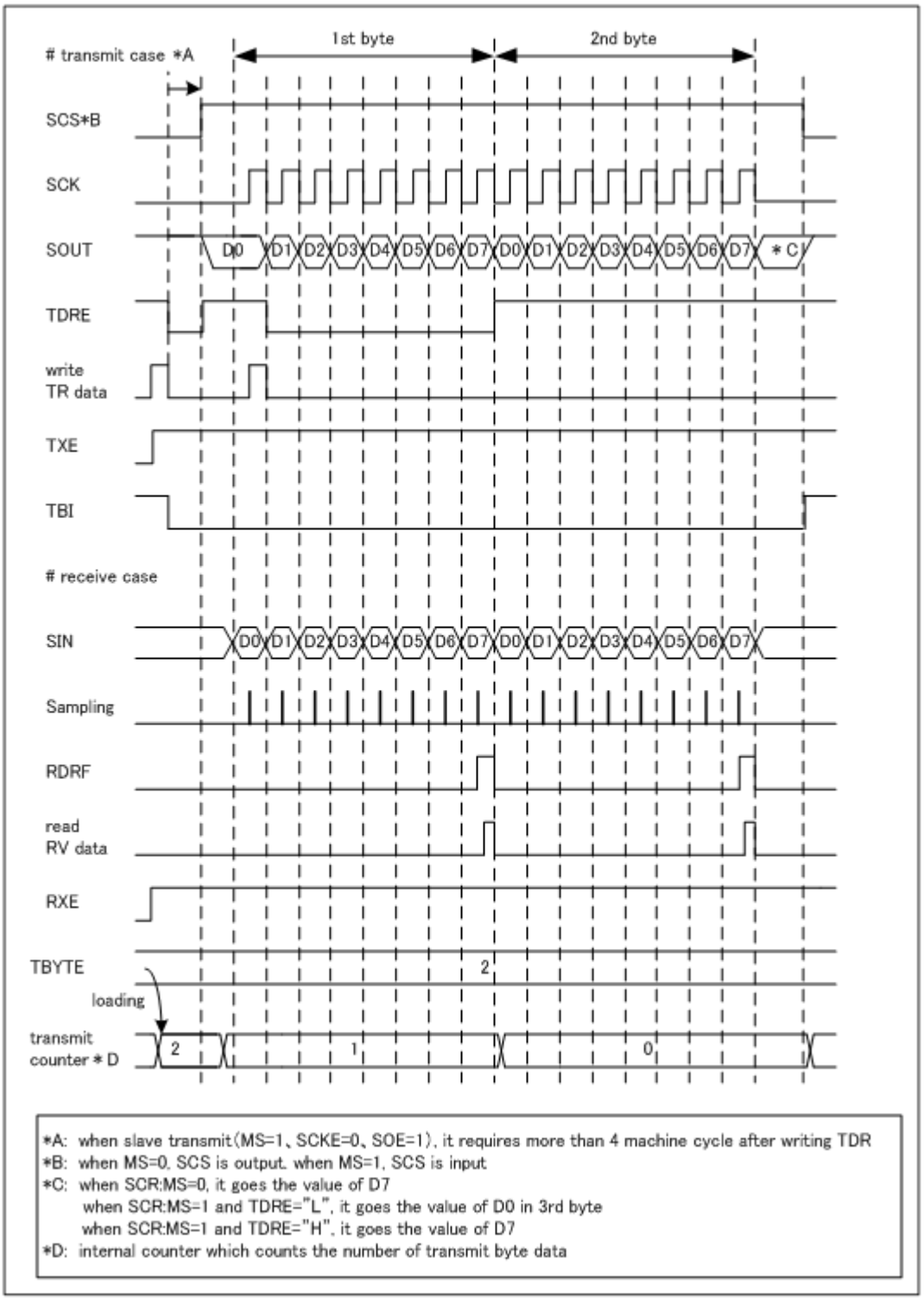
Transmission/Reception Operations

1. To perform transmission/reception at the same time, enable serial data output (SMR:SOE = 1) and also transmission/reception (SCR:TXE, RXE = 1).
2. When transmission data is written to TDR, SSR:TDRE is set to 0, and the first bit is output. Later, transmission data is output in synchronization with a falling edge of the serial clock (SCK) input. When the first 1 bit of transmission data is output, SSR:TDRE is set to 1, and if transmission interrupts are enabled (SCR:TIE = 1), a transmission interrupt request is output. At this time, the second byte of transmission data can be written.
3. Reception data is sampled at a rising edge of the serial clock (SCK) input. If the last bit of the reception data is received, SSR:RDRF is set to 1, and if reception interrupts are enabled (SCR:RIE = 1), a reception interrupt request is output. At this time, reception data (RDR) can be read. When reception data is read, SSR:RDRF is cleared to "0".

Continuous Switching from Reception Operation to Transmission Operation

1. Disable serial data output (SMR:SOE = 0), enable reception interrupts (SCR:RIE = 1), enable reception (SCR:RXE = 1) and also transmission (SCR:TXE = 1). If dummy data is written to TDR when the serial clock (SCK) is at the mark level, reception data is sampled at a falling edge of the serial clock input (SCK).
2. To continue reception, write dummy data to TDR between a reception interrupt request and the next rising edge of the serial clock (SCK).
3. To switch from reception to transmission, enable serial data output (SMR:SOE = 1), disable reception interrupts (SCR:RIE = 0), and disable reception (SCR:RXE = 0) between a reception interrupt request and the next rising edge of the serial clock (SCK), and write transmission data to TDR; after the end of reception, transmission data is output in synchronization with a rising edge of the serial clock.

SPI Transfer (II) Timing Chart (When the Serial Chip Select Pin is Used)


transmit case *A
1st byte
2nd byte
SCS*B
SCK
SOUT
D0 D1 D2 D3 D4 D5 D6 D7 D0 D1 D2 D3 D4 D5 D6 D7 * C
TDRE
write
TR data
TXE
TBI
receive case
SIN
D0 D1 D2 D3 D4 D5 D6 D7 D0 D1 D2 D3 D4 D5 D6 D7
Sampling
RDRF
read
RV data
RXE
TBYTE
2
loading
transmit
counter * D
2
1
0
*A: when slave transmit(MS=1、SCKE=0、SOE=1), it requires more than 4 machine cycle after writing TDR
*B: when MS=0, SCS is output. when MS=1, SCS is input
*C: when SCR:MS=0, it goes the value of D7
when SCR:MS=1 and TDRE="L", it goes the value of D0 in 3rd byte
when SCR:MS=1 and TDRE="H", it goes the value of D7
*D: internal counter which counts the number of transmit byte data

Master Operation (SCR:MS = 0, SMR:SCKE = 1, SCSCR:CSOE = 1, SCSCR:CSENn* = 1)

*: n is replaced by the serial chip select pin number used.

Transmission Operation

1. Enabling serial data output (SMR:SOE = 1) and transmission (SCR:TXE = 1), while disabling reception (SCR:RXE = 0) and then writing transmission data to TDR causes SSR:TDRE to be set to 0. Later, at the same time as the output of the first bit, the serial chip select pin (SCS) becomes active, and after the elapse of the serial chip select pin setup time, serial clock output is started. After the start of serial clock output, transmission data is output in synchronization with a falling edge of the serial clock (SCK) output.
2. Half a cycle before a falling edge of the first serial clock (SCK) output, SSR:TDRE is set to 1, and if transmission interrupts are enabled (SCR:TIE = 1), a transmission interrupt request is output. At this time, the second byte of transmission data can be written.
3. After the end of data transmission for which the count is set in TBYTE, serial clock output is stopped.
4. After the serial clock output is stopped, the serial chip select pin (SCS) becomes inactive after the elapse of the serial chip select pin hold time. At this time, however, if the serial chip select active level (SCSCR:SCAM = 1) is retained, the serial chip select pin (SCS) retains the active state.

Reception Operation

1. Disabling serial data output (SMR:SOE = 0), while enabling transmission (SCR:TXE = 1) and also reception (SCR:RXE = 1) and then writing dummy data to TDR causes the serial chip select pin (SCS) to become active, and after the elapse of the serial chip select pin setup time, serial clock output is started. After serial clock output is started, reception data is sampled at a rising edge of the serial clock (SCK) output.
2. When the last bit is received, SSR:RDRF is set to 1, and if reception interrupts are enabled (SCR:RIE = 1), a reception interrupt request is output.

   At this time, reception data (RDR) can be read.
3. When reception data (RDR) is read, SSR:RDRF is cleared to "0".
4. After the end of data reception for which the count is set in TBYTE, the serial clock output is stopped.
5. After the serial clock output is stopped, the serial chip select pin (SCS) becomes inactive after the elapse of the serial chip select pin hold time. At this time, however, if the serial chip select active level (SCSCR:SCAM = 1) is retained, the serial chip select pin (SCS) retains the active state.

Notes:

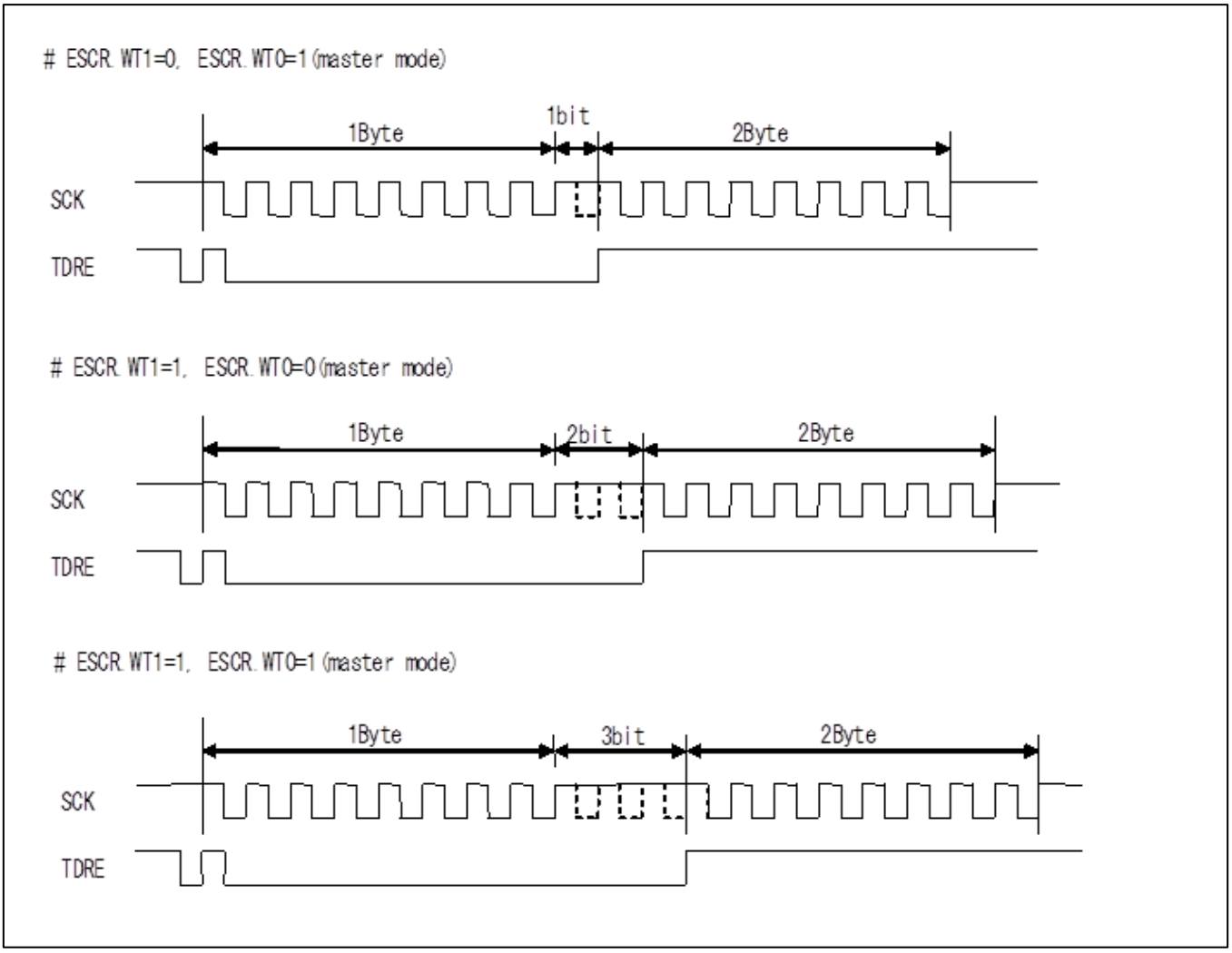
- *When performing only reception, write dummy data to TDR to cause the serial clock (SCK) to be output.*
- *When the transmission/reception FIFOs are enabled, setting the number of frames to transfer in the FBYTE register causes as many serial clock (SCK) pulses as the number of frames to be output.*

Transmission/Reception Operations

1. To perform transmission/reception at the same time, enable both serial data output (SMR:SOE = 1) and also transmission/reception (SCR:TXE, RXE = 1).
2. When transmission data is written to TDR, SSR:TDRE is set to 0. Later, at the same time as the output of the first bit, the serial chip select pin (SCS) becomes active, and after the elapse of the serial chip select pin setup time, serial clock output is started. After the start of serial clock output, transmission data is output in synchronization with a falling edge of the serial clock (SCK) output. Half a cycle before a rising edge of the first serial clock, SSR:TDRE is set to 1, and if transmission interrupts are enabled (SCR:TIE = 1), a transmission interrupt request is output. At this time, the second byte of transmission data can be written.
3. Reception data is sampled at a rising edge of the serial clock (SCK) output. When the last bit of the reception data is received, SSR:RDRF is set to 1, and if reception interrupts are enabled (SCR:RIE = 1), a reception interrupt request is output. At this time, reception data (RDR) can be read. When the reception data is read, SSR:RDRF is cleared to "0".
4. After the end of data transmission/reception for which the count is set in TBYTE, the serial clock output is stopped.
5. After serial clock output is stopped, the serial chip select pin (SCS) becomes inactive after the elapse of the serial chip select pin hold time. At this time, however, if the serial chip select active level (SCSCR:SCAM = 1) is retained, the serial chip select pin (SCS) retains the active state.

Continuous Data Transmission or Reception Wait Operation

If settings other than (ESCR:WT1, ESCR.WT0) = (0, 0) are set for continuous data transmission or reception, a wait is inserted between the frames.



Slave Operation (SCR:MS = 1, SMR:SCKE = 0, SCSCR:CSEN = 1, SCSCR:SCAM = 0)

Transmission Operation

1. Enabling both serial data output (SMR:SOE = 1) and transmission (SCR:TXE = 1) and then writing transmission data to TDR causes SSR:TDRE to be set to 0.
2. If the serial chip select pin (SCS) becomes active, transmission operation is started, and the first bit is output. After the start of transmission, transmission data is output in synchronization with a falling edge of the serial clock (SCK) output.
3. When the first 1 bit of transmission data is output, SSR:TDRE is set to 1, and if transmission interrupts are enabled (SCR:TIE = 1), a transmission interrupt request is output. At this time, the second byte of transmission data can be written.
4. If the serial chip select pin (SCS) becomes inactive, transmission operation is ended, and the serial output pin (SOUT) becomes "H".

Note:

- *After transmission is enabled (SCR:TXE = 1), if the first writing of transmission data to TDR is performed when the serial clock (SCK) is not at the mark level, the first 1 bit of data is not output, and transmission is not performed normally. After transmission is enabled (SCR:TXE = 1), perform the first writing of transmission data to TDR when the serial clock (SCK) is at the mark level.*

Reception Operation

1. If serial data output is disabled (SMR:SOE = 0) but reception is enabled (SCR:RXE = 1), and if the serial chip select pin (SCS) becomes active, reception operation starts, and reception data is sampled at a rising edge of the serial clock input (SCK).
2. When the last bit is received, SSR:RDRF is set to 1, and if reception interrupts are enabled (SCR:RIE = 1), a reception interrupt request is output.

   At this time, reception data (RDR) can be read.
3. When reception data (RDR) is read, SSR:RDRF is cleared to "0".
4. If the serial chip select pin (SCS) becomes inactive, reception is stopped.

Transmission/Reception Operations

1. To perform transmission/reception at the same time, enable both serial data output (SMR:SOE = 1) and also transmission/reception (SCR:TXE, RXE = 1).
2. When transmission data is written to TDR, SSR:TDRE is set to 0. If the serial chip select pin (SCS) becomes active, transmission/reception starts, and the first bit is output. After the start of transmission, transmission data is output in synchronization with a falling edge of the serial clock (SCK) input. When the first 1 bit of transmission data is output, SSR:TDRE is set to 1, and if transmission interrupts are enabled (SCR:TIE = 1), a transmission interrupt request is output. At this time, the second byte of transmission data can be written.
3. Reception data is sampled at a rising edge of the serial clock (SCK) input. If the last bit of the reception data is received, SSR:RDRF is set to 1, and if reception interrupts are enabled (SCR:RIE = 1), a reception interrupt request is output. At this time, reception data (RDR) can be read. When reception data is read, SSR:RDRF is cleared to "0".
4. If the serial chip select pin (SCS) becomes inactive, transmission/reception operations are stopped, and the serial output pin (SOUT) becomes "H".

4. Serial Timer Operation

The serial timer can be used as either a timer function or as a synchronous transmission function.

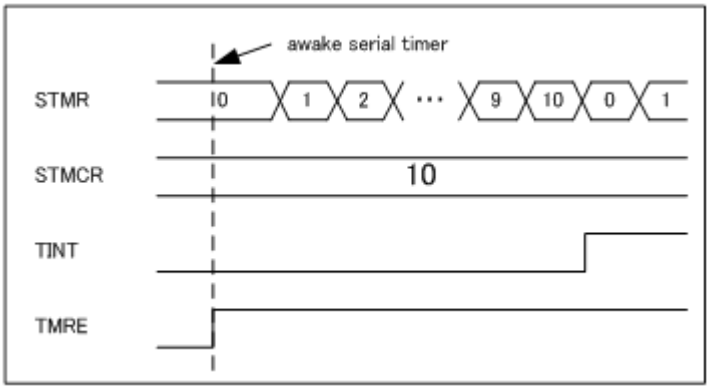
Activating the Serial Timer

The serial timer can be activated either by setting the serial timer enable bit (SACSR:TMRE) to "1" or by using an external trigger.

- Activation by the serial timer enable bit (SACSR:TMRE)

  Provided that the external trigger enable bit (SACSR:TRGE) is set to "0", if the serial timer enable bit (SACSR:TMRE) is set to "1", the serial timer is activated, and the serial timer register (STMR) starts counting from 0.

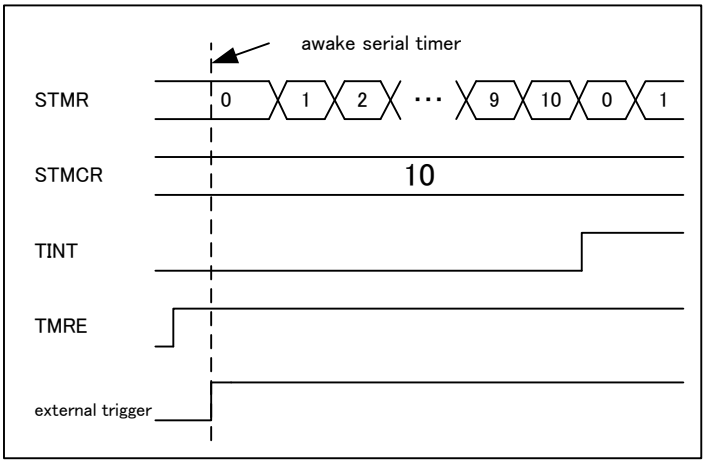
Figure 4-1 Activation by Serial Timer Enable Bit (STMCR = 10, SACSR:TRGE = 0, TSYNE = 0)



- Activation by an external trigger

  The serial timer is activated and counting of the serial timer register (STMR) starts from 0 when all of the following are satisfied: 1. The external trigger enable bit (SACSR:TRGE) is set to "1". 2. An external trigger edge specified by the trigger selection bit (TRG1, 0) is detected. 3. The serial timer enable bit (SACSR:TMRE) is set to "1".

Figure 4-2 Activation by an External Trigger (STMCR = 10, SACSR:TRGE = 1, TSYNE = 0, TRG1 = 0, TRG0 = 0)



Note:

- *Detection of an external trigger edge specified by the trigger selection bit (SACSR:TRG1, 0) does not start the serial timer if both of the following are satisfied: 1. The external trigger enable bit (SACSR:TRGE) is "1". 2. The serial timer enable bit (SACSR:TMRE) is "0".*

Stopping the Serial Timer

The serial timer stops when the serial timer enable bit (SACSR:TMRE) is set to "0". At this time, the value of the serial timer register (STMR) is retained.

Timer Operation

When the synchronous transmission enable bit (SACSR:TSYNE) is "0", the serial timer operates as a timer.

The timer interrupt flag (SACSR:TINT) is set to "1" and the serial timer register (STMR) is reset to 0 when the serial timer register (STMR) coincides with the serial timer comparison register (STMCR).

Figure 4-3 Timer Operation (STMCR = 10, SACSR:TSYNE = 0)

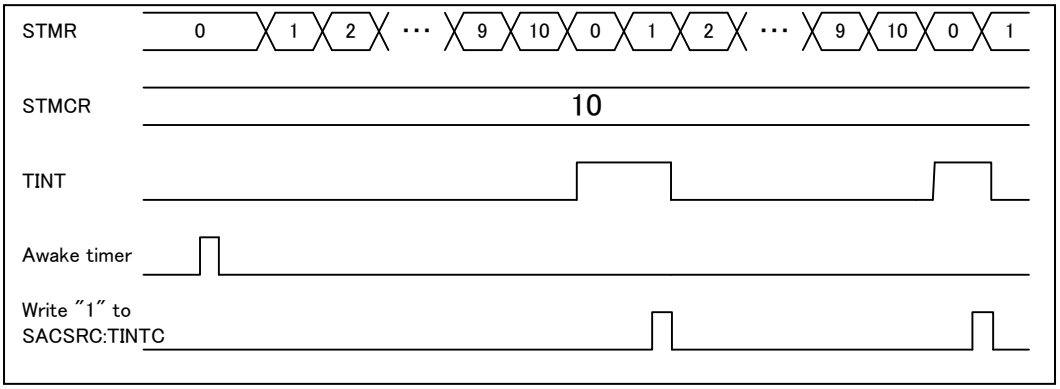


Figure 4-4 Flow Chart of Serial Timer Initialization

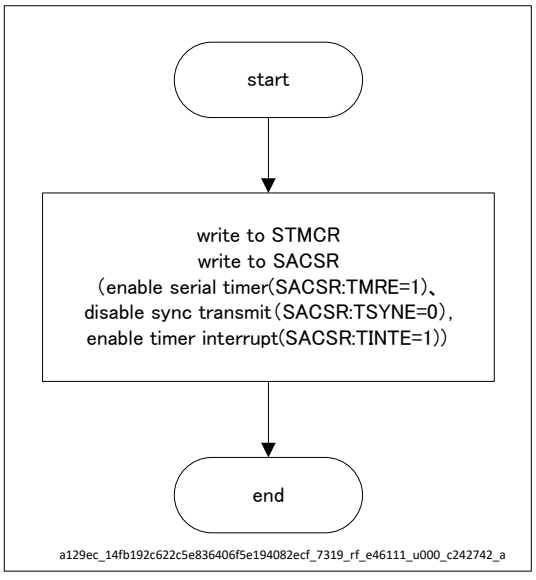
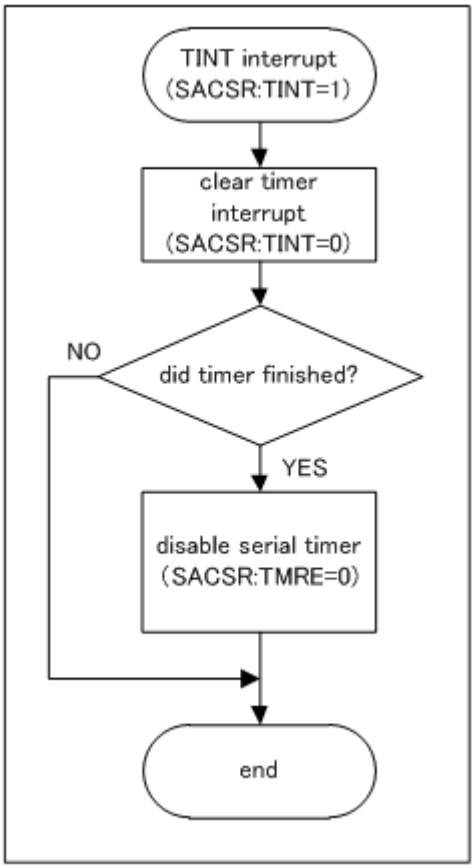


Figure 4-5 Flow Chart of Serial Timer Interrupt Processing



Note:

- *In the state in which synchronous transmission is disabled (SACSR:TSYNE = 0) while 0x0000 is set in the timer comparison register (STMCR), if the timer is operating and the division value (SACSR:TDIV3-0) of the timer operation clock is set to 0b0000, the timer interrupt flag (SACSR:TINT) is fixed to "1".*

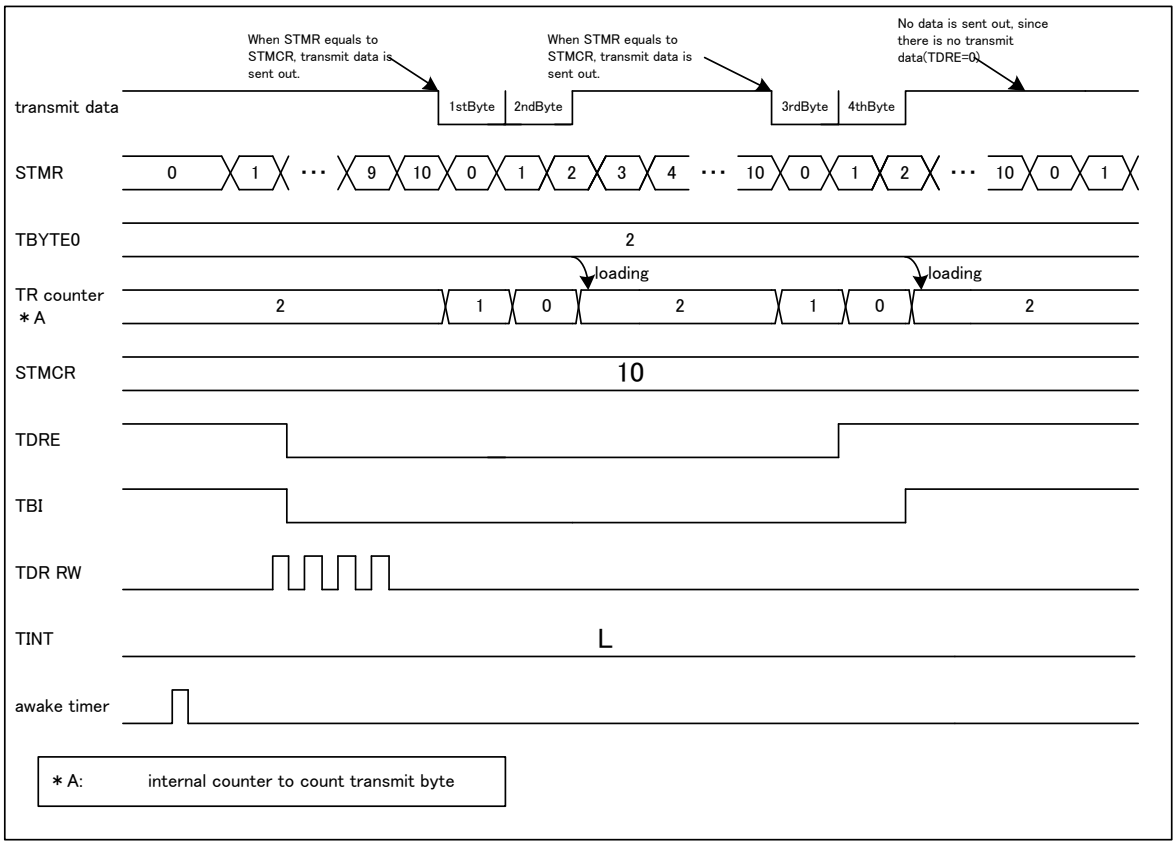
Transmission Operation Synchronized with the Timer

The serial timer is used for synchronous transmission when the synchronous transmission enable bit (SACSR:TSYNE) is set to "1".

Timer-synchronized transmission operates in the following manner:

1. If the transmission data register contains data (SSR:TDRE = 0) and if the contents of the serial timer register (STMR) and the serial timer comparison register (STMCR) match, transmission operation starts, and the serial timer register (STMR) is reset to 0. Data transmission is performed for the data count specified in TBYTE0.
2. After transmitting the data count as specified by TBYTE0, transmission is suspended until the next time the serial timer register (STMR) and serial timer comparison register (STMCR) coincide.

Figure 4-6 Transmission Operation Synchronized with the Timer (STMR = 10, TBYTE0 = 2, SACSR:TSYNE = 1)



When synchronous transmission is enabled (SACSR:TSYNE = 1) and the serial timer register (STMR) and the serial timer comparison register (STMCR) coincide, transmission is not activated under the following conditions:

- Transmission is disabled (SCR:TXE = 0).
- In slave mode (SCR:MS = 1)
- When a chip select error (SACSR:CSE = 1) occurs
- The transmission data register does not contain valid data (SSR:TDRE = 1).

However, writing transmission data to the transmission data register instantly starts transmission when all of the following are satisfied: 1. The transmission data register does not contain valid data (SSR:TDRE = 1). 2. Synchronous transmission is enabled (SACSR:TSYNE = 1). 3. The serial timer register (STMR) and serial timer comparison register (STMCR) coincide.

If the transmission data register (TDR) contains valid data (SSR:TDRE = 0) after transmission of the data count specified by TBYTE, the data is not transmitted until the next time the serial timer register (STMR) and the serial timer comparison register (STMCR) coincide.

If, however, synchronous transmission is enabled (SACSR:TSYNE = 1), transmission is in progress (SSR:TBI = 0), and the serial timer register (STMR) and the serial timer comparison register (STMCR) match, transmission is reserved. When transmission reservation has been made, transmission does not stop after transmission of the data count specified by TBYTE0, and any subsequent transmission is performed.

Transmission reservation is released under one of the conditions below.

- Programmable reset (SCR:UPCL = 1)
- Transmission disabled (SCR:TXE = 0)
- Chip select error (SACSR:CSE = 1)

To perform synchronous reception, disable serial data output (SMR:SOE = 0), enable transmission (SCR:TXE = 1) and also reception (SCR:RXE = 1) and then write dummy data to TDR for the reception count.

Figure 4-7 Flow Chart of Initialization of Transmission Synchronized with Timer

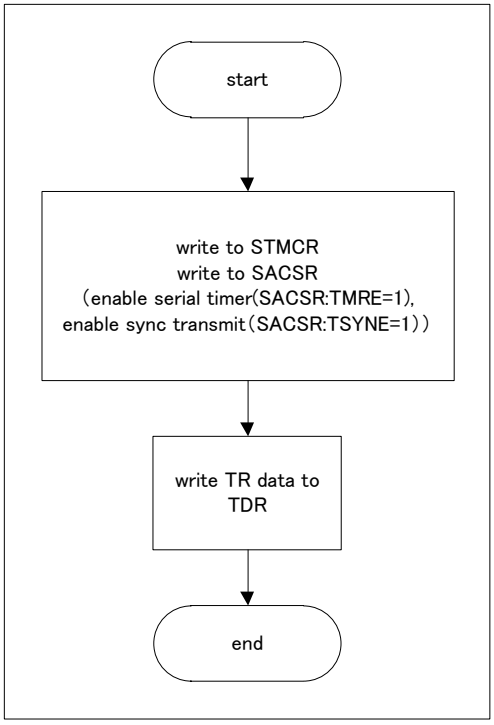
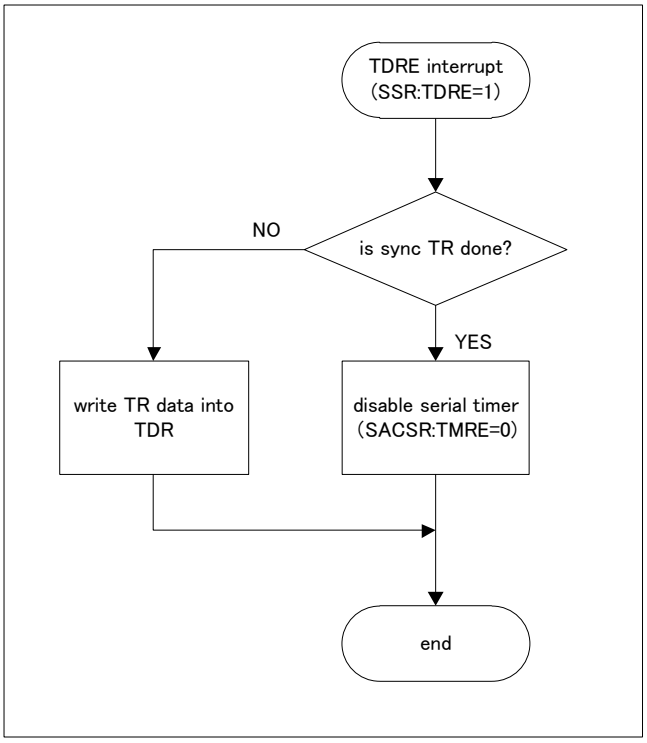


Figure 4-8 Flow Chart of Interrupt Processing in Transmission Synchronized with Timer



Notes:

- *If synchronous transmission is enabled (SACSR:TSYNE = 1), external triggers are enabled (SACSR:TRGE = 1), and 0x0000 is set in the timer comparison register (STMCR), transmission synchronized with the timer is not performed, but transmission synchronized with an external trigger is performed.*
- *If the transmission data register (TDR) contains no valid transmission data (SSR:TDRE = 1) before the transmission of as many data frames as the setting made for TBYTE, the following operation is performed.*
  - If transfer byte errors are enabled (TBEEN = 1), a chip select error (SACSR:CSE = 1) occurs. If "1" is set in the chip select error flag (SACSR:CSE), transmission is not started even if transmission data is written to the transmission data register (TDR).
  - If transfer byte errors are disabled (TBEEN = 0), transmission is stopped until transmission data is written to the transmission data register (TDR). Writing transmission data to the transmission data register (TDR) restarts the transmission operation.

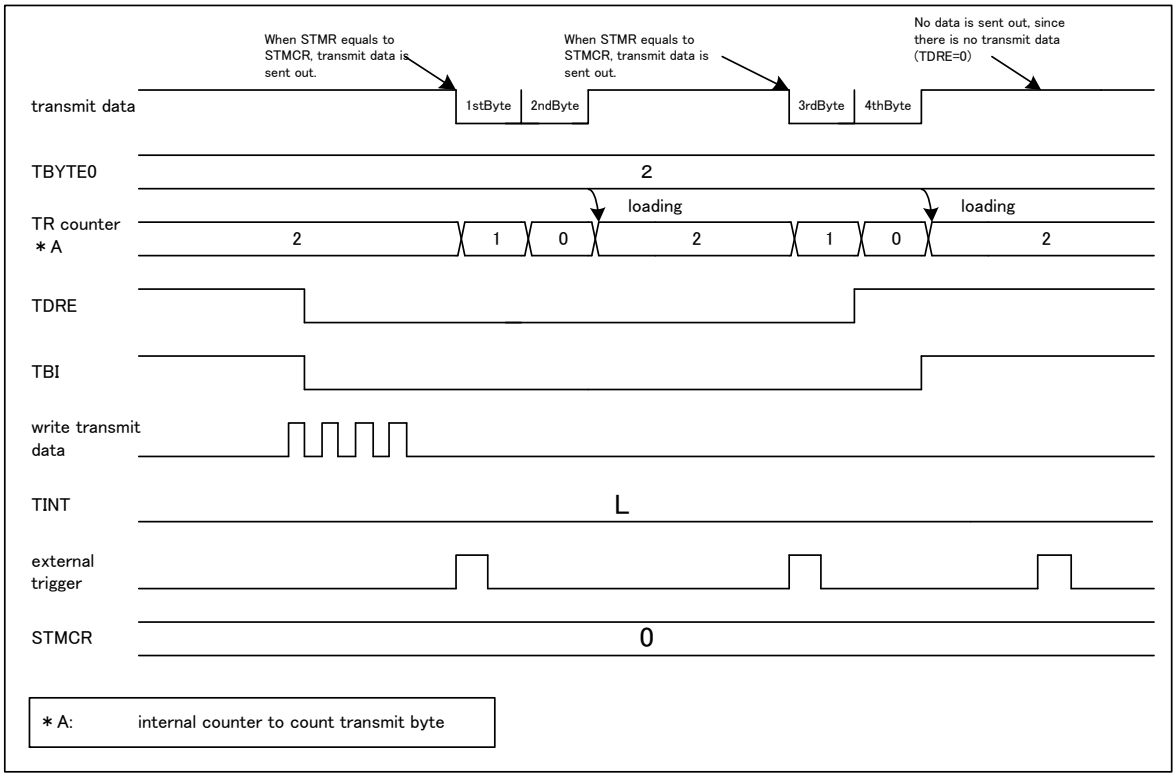
Transmission Operation Synchronized with an External Trigger

If synchronous transmission is enabled (SACSR:TSYNE = 1), external triggers are enabled (SACSR:TRGE = 1), 0x0000 is set in the timer comparison register (STMCR), transmission is enabled (SCR:TXE = 1), and there is valid transmission data (SSR:TDRE = 0), transmission synchronized with an external trigger is started.

The start of transmission synchronized with an external trigger is performed as described below.

1. The operation starts immediately when an external trigger edge specified by the trigger selection bits (SACSR:TRG1, 0) is detected. Data transmission is performed for the data count specified in TBYTE0.
2. After the completion of data transmission for the number of data items that is set in TBYTE0, transmission stops until the next time an external trigger edge that is set in the trigger select bits (SACSR:TRG1, 0) is detected.

Figure 4-9 Transmission Operation Synchronized with External Trigger (TRG1 = 0, TRG0 = 0, SACSR:TSYNE = 1)



During external-trigger-synchronized transmission, the detection of an external trigger edge specified by the trigger selection bits (SACSR:TRG1, 0) does not start transmission under the following conditions:

- Transmission is disabled (SCR:TXE = 0).
- In slave mode (SCR:MS = 1)
- When a chip select error (SACSR:CSE = 1) occurs
- The transmission data register does not contain valid data (SSR:TDRE = 1).

If, however, the transmission data register contains no valid data (SSR:TDRE = 1), and an external trigger edge that is set in the trigger selection bits (SACSR:TRG1, 0) is detected, transmission starts as soon as transmission data is written to the transmission data register.

If the transmission data register (TDR) contains valid transmission data (SSR:TDRE = 0) after transmission of the data count specified by TBYTE, the following occurs: The remaining data is not transmitted until an external trigger edge specified by the trigger selection bits (SACSR:TRG1, 0) is next detected.

However, when an external trigger edge specified by the trigger selection bits (SACSR:TRG1, 0) is detected during a transmission operation (SSR:TBI = 0) with synchronous transmission enabled (SACSR:TSYNE = 1), a transmission reservation takes place. If transmission reservation has been performed, transmission does not stop after transmission of the data count specified by TBYTE0, and the subsequent transmission is performed.

Transmission reservation is released under one of the conditions below.

- Programmable reset (SCR:UPCL = 1)
- Transmission disabled (SCR:TXE = 0)
- Chip select error (SACSR:CSE = 1)

To perform reception operation synchronized with an external trigger, disable serial data output (SMR:SOE = 0), enable both transmission (SCR:TXE = 1) and reception (SCR:RXE = 1) and then write dummy data to TDR for the reception count.

Figure 4-10 Flow Chart of Initialization of Transmission Synchronized with External Trigger

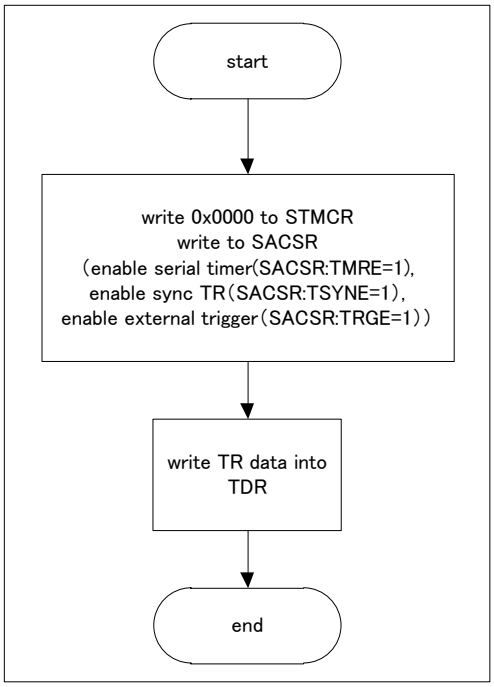
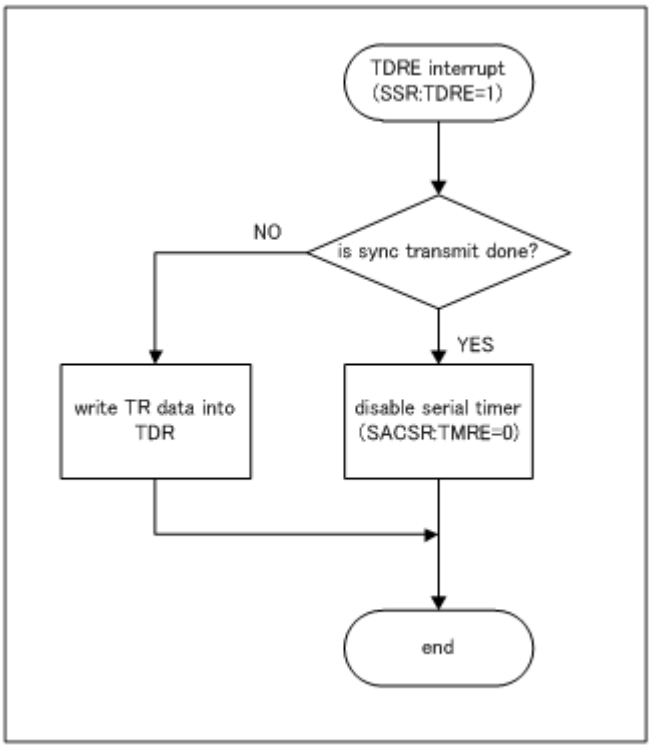


Figure 4-11 Flow Chart of Interrupt Processing in Transmission Synchronized with External Trigger



Notes:

- *If the transmission data register (TDR) contains no valid transmission data (SSR:TDRE = 1) before the transmission of as many data frames as the setting made for TBYTE, the operation described below is performed.*

- If transfer byte errors are enabled (TBEEN = 1), a chip select error (SACSR:CSE = 1) occurs. If "1" is set in the chip select error flag (SACSR:CSE), transmission is not started even if transmission data is written to the transmission data register (TDR).
- If transfer byte errors are disabled (TBEEN = 0), transmission is stopped until transmission data is written to the transmission data register (TDR). Writing transmission data to the transmission data register (TDR) restarts transmission.

5. Operation of Serial Chip Select

This section explains serial chip select operation.

Operation in Master Mode (SCR:MS = 0)

In master mode (SCR:MS = 0), the serial chip select pin operates as described below.

1. If you enable both serial chip select (SCSCR:CSENn = 1) and transmission (SCR:TXE = 1), the serial chip select pin becomes active when transmission data is written.
2. After the elapse of the serial chip select pin setup time, transmission/reception operations are started.
3. After data transmission/reception for which the count is set in TBYTE, the serial clock stops.
4. After the serial clock is stopped, the serial chip select pin becomes inactive after the elapse of the serial chip select pin hold time.

Figure 5-1 Serial Chip Select Operation (Master Transmission (MS = 0), Normal Transfer (SPI = 0), SCINV = 0)

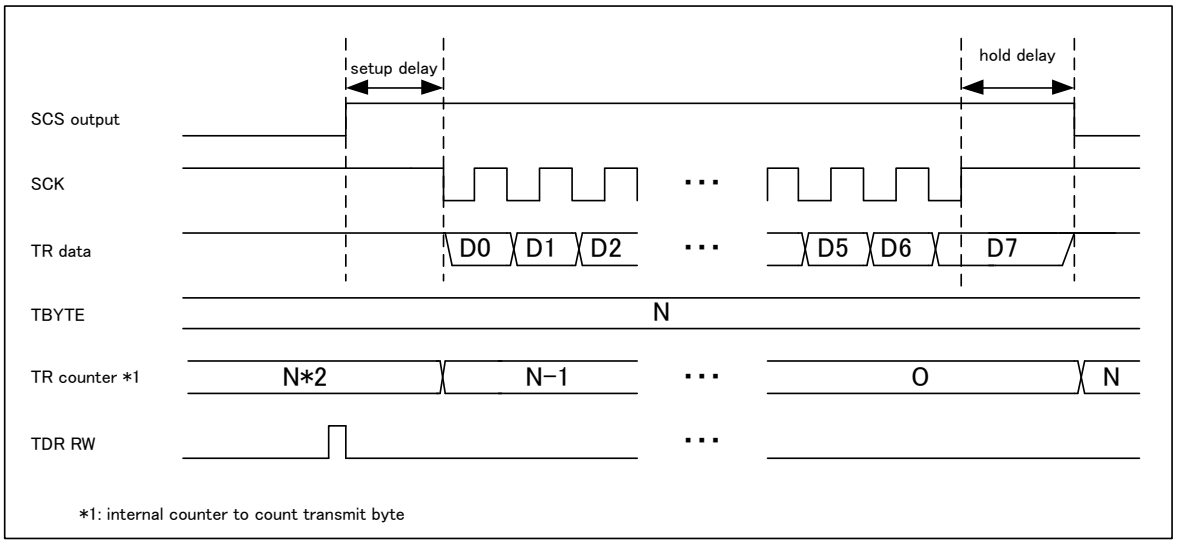
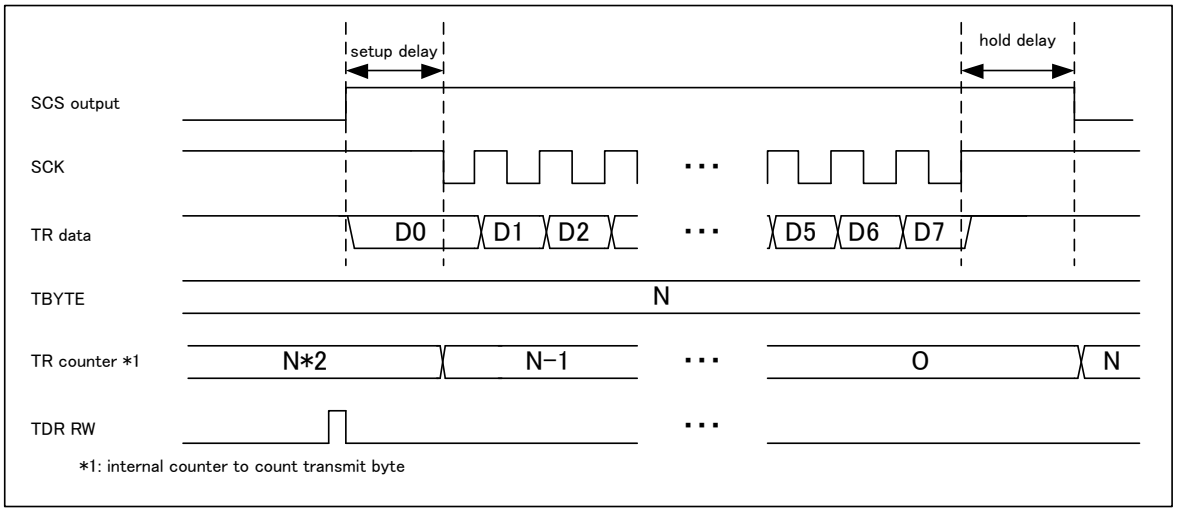


Figure 5-2 Serial Chip Select Operation (Master Transmission (MS = 0), SPI Transfer (SPI = 1), SCINV = 0)



Notes:

- *While the serial chip select pin is active, if transmission is disabled (SCR:TXE = 0) and software is reset (SCR:UPCL = 1), the serial chip select pin becomes inactive.*
- *If the active state of the serial chip select pin is not retained (SCSCR:SCAM = 0), the serial chip select pin becomes inactive, and if there is no transmission data (SSR:TDRE = 1) after the elapse of the deselect time, the transmission bus idle state (SSR:TBI = 1) is assumed.*
- *In master mode (SCR:MS = 0), if SCSCR:CSEN3 to CSEN0 is set to 0b0000, transmission/reception operations are performed regardless of the status of the serial chip select pin.*
- *When fewer frames than the number set in TBYTE have been transmitted, if the transmission data register (TDR) contains no valid transmission data at the time of completion of 1-frame transmission (SSR:TDRE = 1), the operation described below is performed.*
  - If transfer byte errors are enabled (TBEEN = 1), a chip select error (SACSR:CSE = 1) occurs. Once the hold delay time elapses after the occurrence of a chip select error (SACSR:CSE = 1), the serial chip select pin becomes inactive. If "1" is set in the chip select error flag (SACSR:CSE), transmission is not started even if transmission data is written to the transmission data register (TDR).
  - If transfer byte errors are disabled (TBEEN = 0), transmission is stopped until transmission data is written to the transmission data register (TDR). At this time, the serial chip select pin is active. Writing transmission data to the transmission data register (TDR) restarts the transmission operation.

Serial Chip Select Timing Adjustment

If, in master mode (SCR:MS = 0), serial chip select operation is enabled (SCSCR:CSENn = 1), by adjusting the serial chip select timing registers (SCSTR3 to SCSTR0), setup delay, hold delay, and deselect time can be adjusted.

- Setup delay time

  Time from the instant at which the serial chip select pin becomes active until the serial clock is output. For details on the regulations governing the setup delay time, see Figure 5-3 and Figure 5-4.

  This can be adjusted with the chip select setup delay bits (SCSTR1:CSSU7 to CSSU0).

- Hold delay time

  Time from the instant at which the output of the serial clock ends until the serial chip select pin becomes inactive. For details on the regulations governing the hold delay time, see Figure 5-3 and Figure 5-4.

  This can be adjusted with the chip select hold delay bits (SCSTR0:CSHD7 to CSHD0).

- Deselect time

  Minimum time from the instant at which the serial chip select pin becomes inactive until the next time that the serial chip select pin becomes active. Even if transmission data is written to the transmission data register (TDR) within the deselect time, the serial chip select pin does not become active until the end of the deselect time. For details on the regulations governing the deselect time, see Figure 5-3 and Figure 5-4.

  This can be adjusted with the chip select/deselect bits (SCSTR3 to SCSTR2:CSDS15 to CSDS0).

Figure 5-3 Timing Adjustment (Normal Transfer (SPI = 0), SCINV = 0)

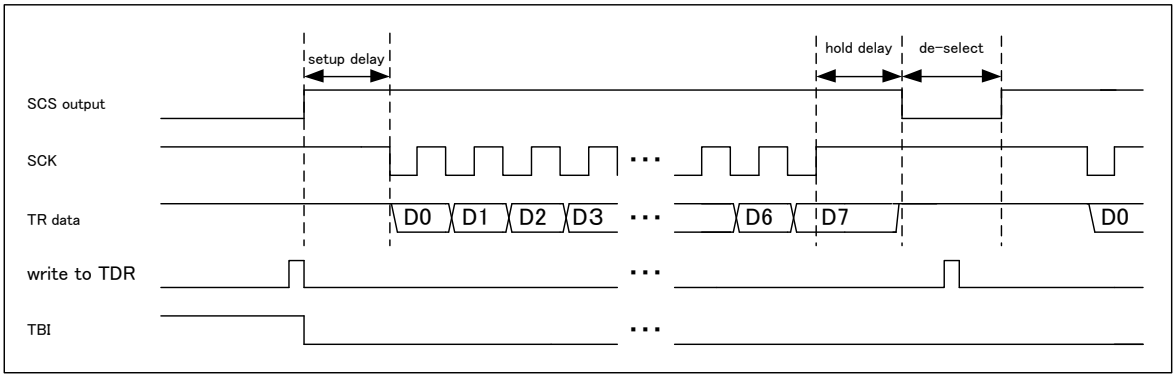
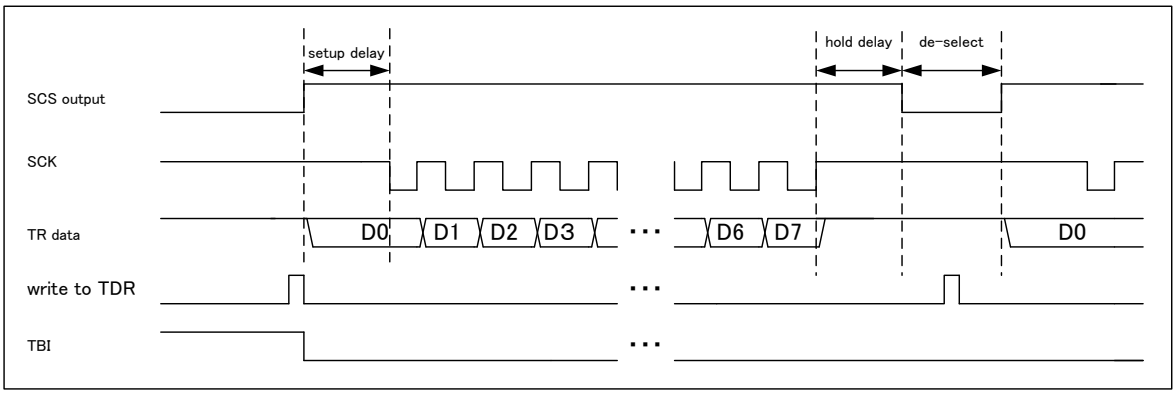


Figure 5-4 Timing Adjustment (Normal Transfer (SPI = 1), SCINV = 0)



Notes:

- *During normal transfer (SCR:SPI = 0) with no hold delay time(SCSTR0:CSHD7 to CSHD0 = 0x00), the chip select pin may become inactive before the sampling of the last bit. In this case, perform adjustments by increasing the values of SCSTR0:CSHD7 to CSHD0.*
- *During SPI transfer (SCR:SPI = 1) with no setup delay time (SCSTR1:CSSU7 to CSSU0 = 0x00), the chip select pin may become active after the sampling of the first bit. In this case, perform adjustments by increasing the values of SCSTR1:CSSU7 to CSSU0.*

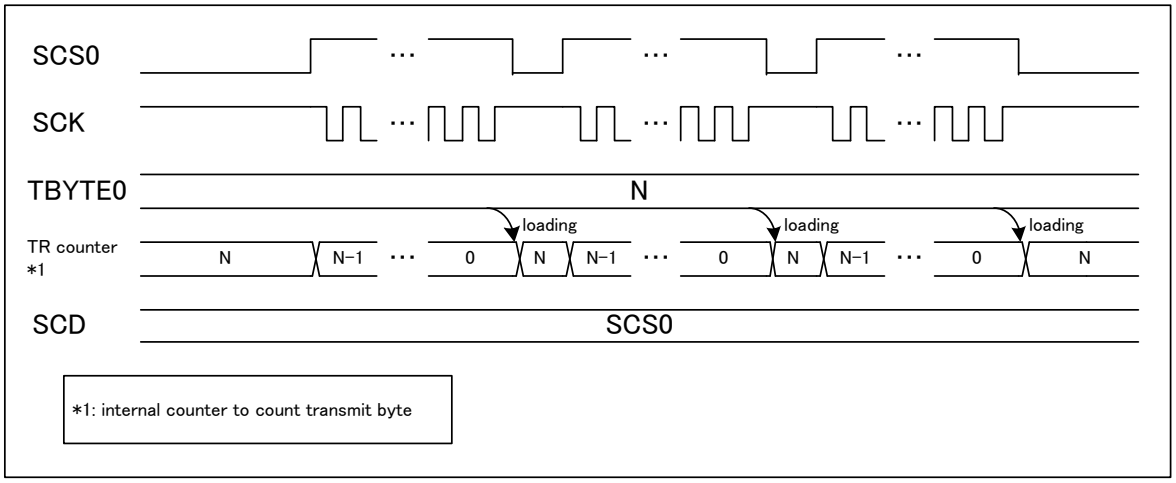
Independent Operation of the Chip Select Pin (Valid Only in Master Mode (SCR:MS = 0))

If the serial chip select start bits (SCSCR:SST1 and SST0) and the serial chip select end bits (SCSCR:SED1 and SED0) are equal, operation is performed based on only the set serial chip select pin.

When the active level of serial chip select is not retained (SCSCR:SCAM = 0), the serial chip select pin becomes inactive for each data transmission/reception for which the count is set in TBYTE.

For details on the serial chip select pin operation performed if the active level of serial chip select is retained (SCSCR:SCAM = 1), see "Serial chip select active level retention operation."

Figure 5-5 Independent Operation of Chip Select (SST1 and SST0 = 0, SED1 and SED0 = 0, CSEN0 = 1, SCAM = 0)



Note:

- *During independent operation, serial chip select pin timing adjustment (setup time, hold time, and deselect time) is valid.*

Rounding Operation of the Chip Select Pin (Valid Only in Master Mode (SCR:MS = 0))

If the serial chip select start bits (SCSCR:SST1 and SST0) and the serial chip select end bits (SCSCR:SED1 and SED0) differ, multiple serial chip select pins become active sequentially.

1. If transmission data is written when you enable both serial chip select output (SCSCR:CSOE = 1) and transmission (SCR:TXE = 1), the serial chip select pins become active, starting with that specified with the serial chip select start bits (SCSCR:SST1 and SST0).
2. When the active level of serial chip select is not retained (SCSCR:SCAM = 0), the serial chip select pin becomes inactive after the end of data transmission/reception for which the count is set in TBYTE. Later, the serial chip select pin with the pin number equal to the number of the last active serial chip select pin + 1 becomes active.* 1

   If the next serial chip select pin to become active is disabled (SCSCR:CSENn = 0), the serial chip select pin does not become active but is instead skipped.
3. If the active serial chip select pin number matches the serial chip select pin specified with the serial chip select end bits (SCSCR:SED1 and SED0), the serial chip select pin specified with the serial chip select start bits (SCSCR:SST1 and SST0) becomes active next.

*1: If the last active serial chip select pin is pin 0, pin 1 becomes active; if it is pin 1, pin 2 becomes active; if it is pin 2, pin 3 becomes active; and if it is pin 3, pin 0 becomes active.

For details on the serial chip select pin operation performed if the active level of serial chip select is retained (SCSCR:SCAM = 1), see "Serial chip select active level retention operation."

Figure 5-6 shows the timing chart that is assumed if the serial chip select start pin is SCS0 (SST1 and SST0 = 0) and the end pin is SCS3 (SED1 and SED0 = 3).

Figure 5-6 Rounding Operation of Chip Select (SST1-0 = 0, SED1-0 = 3, CSEN3 = 1, CSEN2 = 1, CSEN1 = 1, CSEN0 = 1, SCAM = 0)

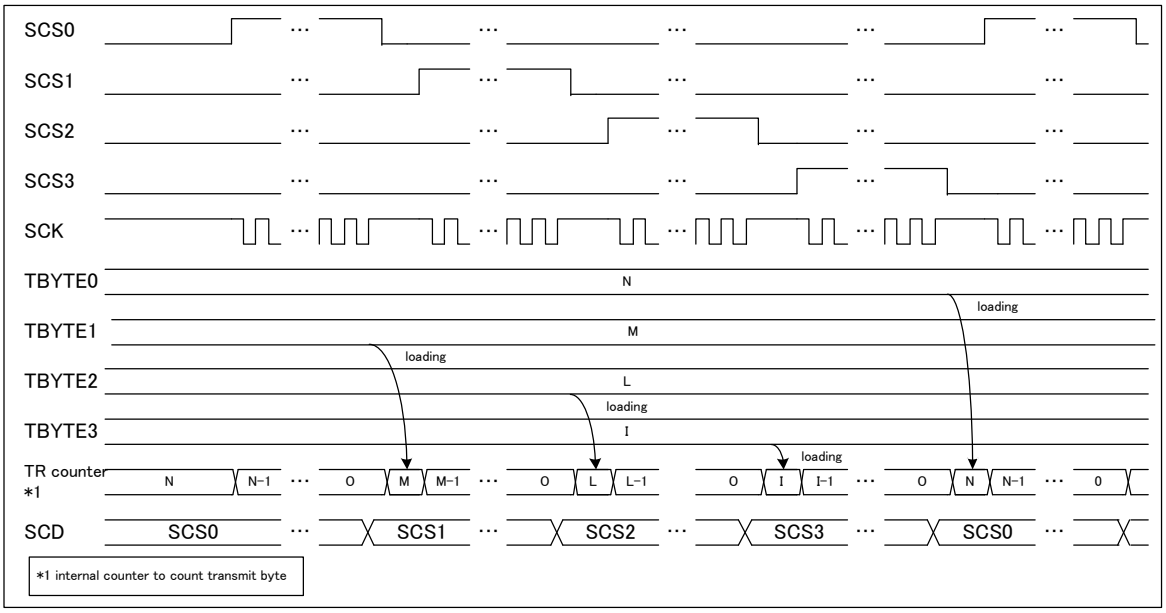


Figure 5-7 shows the timing chart that is assumed if the serial chip select start pin is SCS1 (SST1 and SST0 = 1) and the end pin is SCS2 (SED1 and SED0 = 2).

Figure 5-7 Rounding Operation of Chip Select (SST1-0 = 1, SED1-0 = 2, CSEN3 = 0, CSEN2 = 1, CSEN1 = 1, CSEN0 = 0, SCAM = 0)

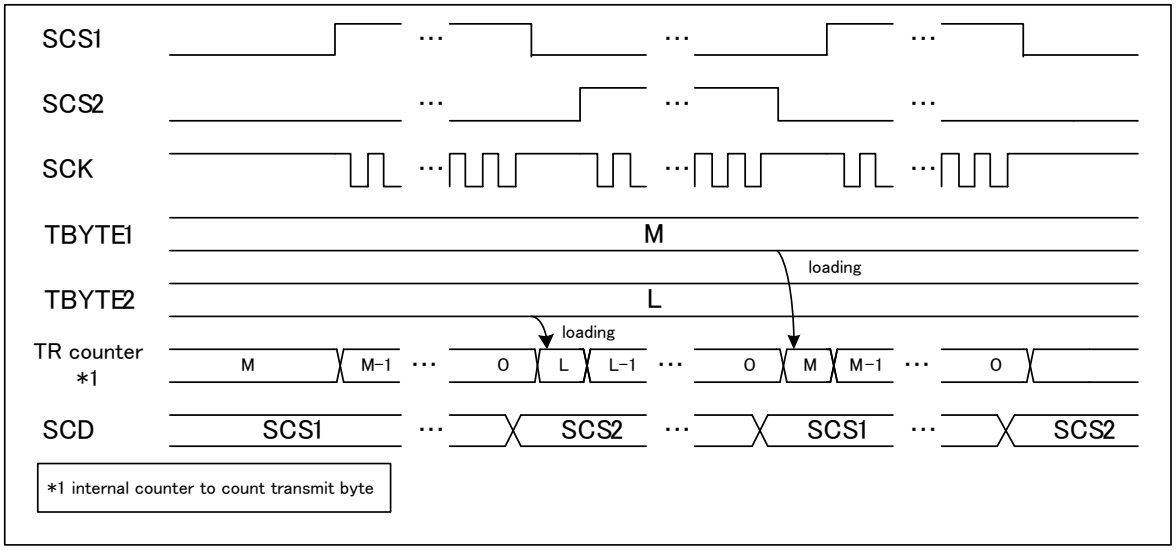
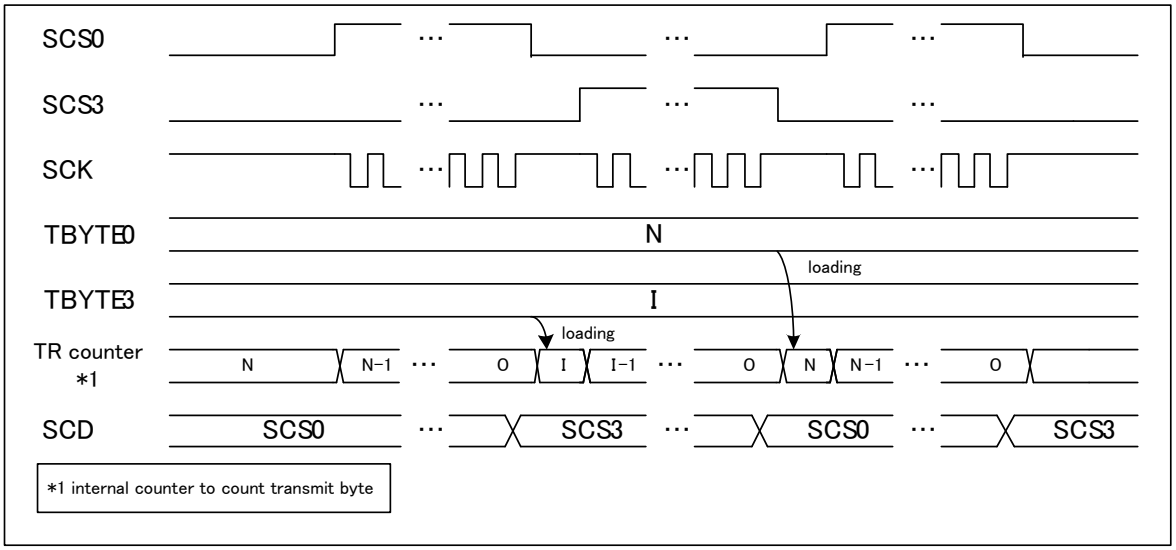


Figure 5-8 shows the timing chart that is assumed if the serial chip select start pin is SCS0 (SST1 and SST0 = 0), the end pin is SCS3 (SED1 and SED0 = 3), and chip select pins 1 and 2 are disabled (CSEN1 and CSEN2 = 0b00). After serial chip select pin 0 becomes active, pins 1 and 2 are skipped, and pin 3 becomes active.

Figure 5-8 Rounding Operation of Chip Select (SST1-0 = 0, SED1-0 = 3, CSEN3 = 1, CSEN2 = 0, CSEN1 = 0, CSEN0 = 1, SCAM = 0)



Notes:

- *Serial chip select pins become active starting with that specified with the serial chip select start bits (SCSCR:SST1 and SST0) in any of the following cases:*
  - If a change is made from transmission operation disable (SCR:TXE = 0) to transmission operation enable (SCR:TXE = 1)
  - If a software reset (SCR:UPCL = 1) is performed
- *During a rounding operation, serial chip select pin timing adjustment (setup time, hold time, and deselect time) is valid.*

Serial Chip Select Active Level Retention Operation (SCSCR:SCAM = 1) (Valid Only in Master Mode (SCR:MS = 0))

If the serial chip select active level retention bit (SCSCR:SCAM) is set to "1" and transmission is started, the serial chip select pin is retained in the active state.

Table 5-1 Serial Chip Select Active Level Retention Bit (SCSCR:SCAM)

Current State	Current SCSCR:SCAM Bit	Current SSR:TDRE Bit	Next State
Transmitting (Transmission count < TBYTE)	0	—	Until as many frames as the value set for TBYTE have been transmitted, the serial chip select pin remains active.
	1		
End of transmission of as many frames as the setting of TBYTE	0	0	After the hold delay time, the serial chip select pin is deactivated. After the elapse of the deselect time, the next transmission is started.
		1	After the hold delay time, the serial chip select pin is deactivated. After the elapse of the deselect time, transmission is stopped until the next transmission is written.
	1	1	The active level of the serial chip select is retained.
		0	With serial chip select being in the active level, transmission operation continues. Until as many frames as the value set for TBYTE have been transmitted, the serial chip select pin remains active.
A chip select error (SACSR:CSE = 1) occurs.	—	—	Regardless of the setting of SCAM, the serial chip select pin is deactivated after the hold delay time.
A software reset is executed (SCR:UPCL = 1).	—	—	Regardless of the setting of SCAM, the serial chip select pin is deactivated immediately.
Transmission disabled (SCR:TXE = 0)			

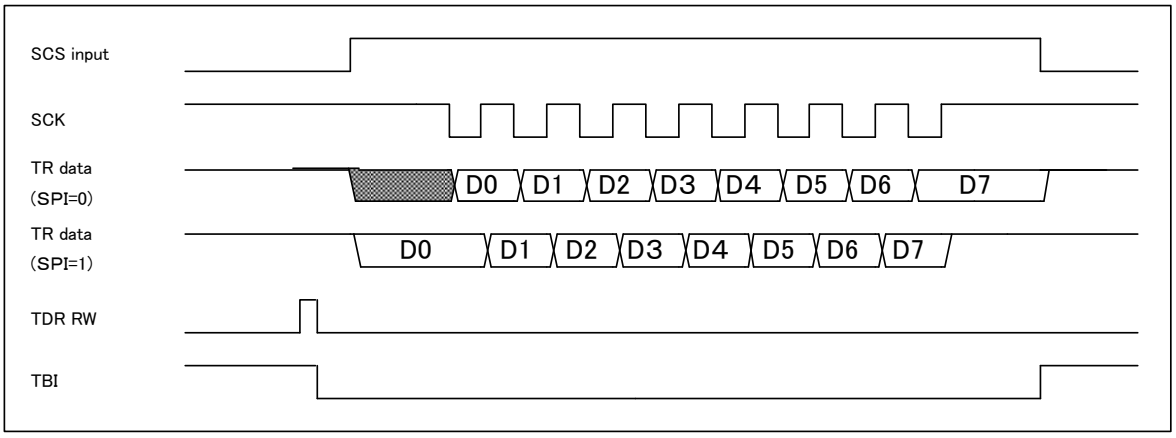
Notes:

- *If the conditions described below are all satisfied, the serial chip select pin is not retained, and after the elapse of the hold delay time, the serial chip select pin becomes inactive, causing a chip select error (SACSR:CSE = 1) to occur.*
  - Transfer byte errors are enabled (SACSR:TBEEN = 1).
  - If data transmission/reception for which the count is set in TBYTE has not ended.
  - If the transmission data register (TDR) is empty (SSR:TDRE = 1).

Operation in Slave Mode (SCR:MS = 1)

If serial chip select pin 0 (SCS0) is enabled (SCSCR:CSEN0 = 1) and the serial chip select pin input becomes active, transmission or reception is performed in synchronization with the serial clock (SCK). Later, if the serial chip select pin input becomes inactive, transmission or reception is ended.

Figure 5-9 Serial Chip Select Operation in Slave Mode (Slave Transmission, SCINV = 0)



Notes:

- *When the serial chip select pin input is inactive, operation is not performed even if the serial clock is input.*
- *If the serial chip select input becomes inactive before bits are last sampled during reception, the data being received is erased.*
- *If the serial chip select input becomes inactive during transmission, the data being transmitted is erased and a chip select error occurs (SACSR:CSE).*
- *If TDR is empty (SSR:TDRE = 1) and the serial chip select pin input becomes inactive, transmission bus idle state (SSR:TBI = 1) is assumed.*
- *In slave mode (SCR:MS = 1), if SCSCR:CSEN0 is set to "0", transmission/reception operations are performed regardless of the status of the serial chip select pin.*

Serial Chip Select Pin Format Setting

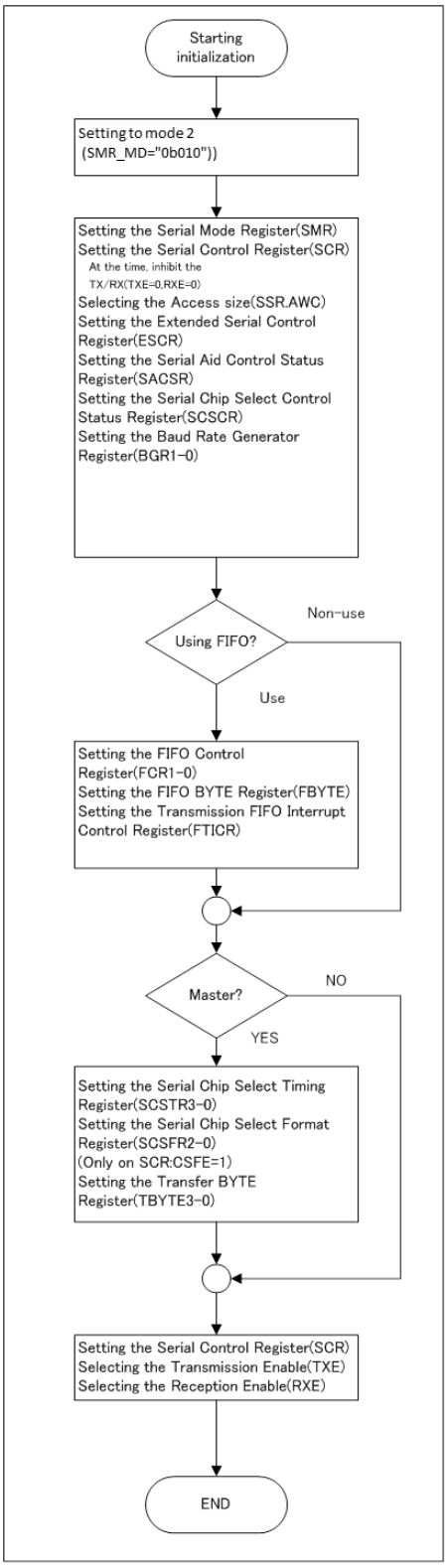
The active level of chip select of each serial chip select pin, mark level of the serial clock, SPI mode enabling and disabling, and data direction and data length of serial data output can be set with the bits listed in Table 5-2.

Table 5-2 Serial Chip Select Pin Format Setting

Conditions		Chip Select Active Level	Serial Clock Inversion	SPI Setting	Data Direction	Data Length
Chip select format is enabled (SCR:CSFE = 1) and master mode (SCR:MS = 0)	Serial chip select pin 0 output	SCSCR0:CSLVL	SMR:SCINV	SCR:SPI	SMR:BDS	ESCR:L3 to L0
	Serial chip select pin 1 output	SCSFR0: CS1CSLVL	SCSFR0: CS1SCINV	SCSFR0: CS1SPI	SCSFR0: CS1BDS	SCSFR0: CS1L3 to CS1L0
	Serial chip select pin 2 output	SCSFR1: CS2CSLVL	SCSFR1: CS2SCINV	SCSFR1: CS2SPI	SCSFR1: CS2BDS	SCSFR1: CS2L3 to CS2L0
	Serial chip select pin 3 output	SCSFR2: CS3CSLVL	SCSFR2: CS3SCINV	SCSFR2: CS3SPI	SCSFR2: CS3BDS	SCSFR2: CS3L3 to CS3L0
Chip select format disable (SCR:CSFE = 0)		SCSCR0:CSLVL	SMR:SCINV	SCR:SPI	SMR:BDS	ESCR:L3 to L0
Slave mode (MS = 1)						
When chip select is not used (CSEN3 to CSEN0 = 0b0000)						

Initialization Flow

Figure 5-10 Chip Select Initialization Flow



6. Test Mode

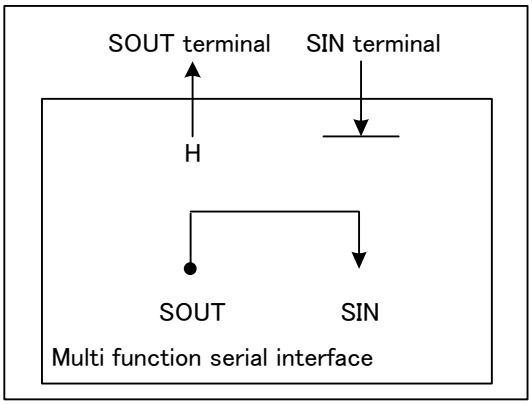
This section describes operation in test mode.

Serial Test Mode

When serial test mode is enabled (SACSR:STST = 1), SOUT and SIN are connected together inside the multi-function serial interface, so that the data transmitted from SOUT can be received from SIN directly.

When serial test mode is enabled (SACSR:STST = 1), the SOUT pin is fixed to "H", and the data input to the SIN pin is ignored.

Figure 6-1 Serial Test Mode



Note:

- *The value of the serial test mode enable bit (SACSR:STST) can be changed only when transmission and reception are prohibited (SCR:TXE = RXE = 0).*

7. Dedicated Baud Rate Generator

The dedicated baud rate generator functions during master operation only. When using the reception FIFO, however, set the dedicated baud rate generator even during slave operation.

CSIO (Clock Synchronous Serial Interface) Baud Rate Selection

The dedicated baud rate generator must be set differently for master operation and slave operation.

[1] During Master Operation

Use the Dedicated Baud Rate Generator to Divide the Internal Clock and Select a Baud Rate.

- There are 2 internal reload counters, each of which corresponds to the transmission or reception serial clock. A baud rate can be selected by setting a 15-bit reload value in baud rate generator register 1 or 0 (BGR1, BGR0).
- The reload counters divide the internal clock by the value that is set.

[2] During Slave Operation

- During slave operation (SCR:MS = 1), the dedicated baud rate generator does not function. (Instead, the external clock, input from clock input pin SCK, is input directly.)

Note:

- *When using the reception FIFO, set the dedicated baud rate generator even during slave operation.*

7.1. Setting the Baud Rate

This section describes the baud rate setting. It also describes the results of calculating serial clock frequencies.

Baud Rate Calculation

The 2 15-bit reload counters are set by baud rate generator registers 1, 0 (BGR1, BGR0). The baud rate is calculated using the formula given below.

(1) Reload value:

```
V = Phy / b - 1

V : reload value  b : baud rate  Phy : bus clock frequency
```

(2) Calculation example

```
When bus clock is 16MHz, using internal clock, baud rate is 19200 bps, the reload value can be
 calculated as follows.
               reload value :
                   V = (16 * 1000000)/19200 - 1 = 832
                   and then baud rate will be
                   b = (16 * 1000000)/(832 + 1) = 19208 bps
```

6e9f8f_ef8b95fa570a507b8b91fc169c501b19_7320_rf_96ff61_u000_c242742_a

(3) Baud rate error

The baud rate error is obtained by using the following formula.

```
error rate(%) = (calculated value - target value)/target value * 100
  (ex. When bus clock is 20MHz, target baud rate is 153600 bps,
       the result will be as follows.
            reload value    = (20 x 1000000)/153600 - 1 = 129
       calculated baud rate =  20 x 1000000)/(129 + 1) = 153846 (bps)
            error rate (%) = (153846 - 153600)/153600 x 100 = 0.16 (%)
```

Notes:

- *Setting the reload value to "0" stops the reload counter.*
- *If the reload value is even-numbered, the "H" width and "L" width of the serial clock will be as described below, depending on the settings made for the SMR:SCINV bit and SCR:SPI bit. If the reload value is odd-numbered, the "H" width and the "L" width of the serial clock will be the same.*
  - For normal transfer (SCR:SPI = 0) when the mark level of the serial clock is "H" (SMR:SCINV = 0) or for SPI transfer (SCR:SPI = 1) when the mark level of the serial clock is "L" (SMR:SCINV = 1), the "H" width of the serial clock will be longer by 1 bus clock cycle.
  - For normal transfer (SCR:SPI = 0) when the mark level of the serial clock is "L" (SMR:SCINV = 1) or for SPI transfer (SCR:SPI = 1) when the mark level of the serial clock is "H" (SMR:SCINV = 0), the "L" width of the serial clock will be longer by 1 bus clock cycle.
- *Set the reload value to 2 or more in master mode, or to 3 or more in slave mode.*

– *When considering the tolerable baud rate range, also consider the impact of the jitter in the clock input on the macro.*

Example Reload Values and Baud Rate Settings for Individual Bus Clock Frequencies

The following table contains example reload values and baud rate settings.

Table 7-1 Example Reload Values and Baud Rate Settings

Baud Rate	8 MHz		10 MHz		16 MHz		20 MHz		24 MHz		32 MHz	
(bps)	Value	ERR	Value	ERR	Value	ERR	Value	ERR	Value	ERR	Value	ERR
8 M	-	-	-	-	-	-	-	-	-	-	3	0
6 M	-	-	-	-	-	-	-	-	3	0	-	-
5 M	-	-	-	-	-	-	3	0	-	-	-	-
4 M	-	-	-	-	3	0	4	0	5	0	7	0
2.5 M	-	-	3	0	-	-	7	0	-	-	-	-
2 M	3	0	4	0	7	0	9	0	11	0	15	0
1 M	7	0	9	0	15	0	19	0	23	0	31	0
500000	15	0	19	0	31	0	39	0	47	0	63	0
460800	-	-	-	-	-	-	-	-	51	0.16	-	-
250000	31	0	39	0	63	0	79	0	95	0	127	0
230400	-	-	-	-	-	-	86	-0.22	103	0.16	-	-
153600	51	0.16	64	0.16	103	0.16	129	0.16	155	0.16	207	0.16
125000	63	0	79	0	127	0	159	0	191	0	255	0
115200	-	-	86	-0.22	138	-0.08	173	-0.22	207	0.16	277	-0.08
76800	103	0.16	129	0.16	207	0.16	259	0.16	311	-0.16	416	-0.08
57600	138	-0.08	173	-0.22	277	-0.08	346	0.06	416	-0.08	555	-0.08
38400	207	0.16	259	0.16	416	-0.08	520	-0.03	624	0	832	0.04
28800	277	-0.08	346	0.06	554	-0.01	693	0.06	832	0.03	1110	0.01
19200	416	-0.08	520	-0.03	832	-0.03	1041	-0.03	1249	0	1666	-0.02
10417	767	<0.01	959	<0.01	1535	<0.01	1919	<0.01	2303	<0.01	3071	<0.01
9600	832	0.04	1041	-0.03	1666	-0.02	208	0.01	2499	0	3332	0.01
7200	1110	<0.01	1388	<0.01	2221	<0.01	2777	<0.01	3332	<0.01	4443	0.01
4800	1666	-0.02	2082	-0.02	3332	<0.01	4166	<0.01	4999	0	6666	<0.01
2400	3332	<0.01	4166	<0.01	6666	<0.01	8332	<0.01	9999	0	13332	<-0.01
1200	6666	<0.01	8332	<0.01	13332	<0.01	16666	<0.01	19999	0	26666	<0.01
600	13332	<0.01	16666	<0.01	26666	<0.01	-	-	-	-	-	-
300	26666	<0.01	-	-	-	-	-	-	-	-	-	-

- Value: Setting of the BGR1/0 register
- ERR: Baud rate error (%)

Table 7-2 Example Reload Values and Baud Rate Settings (Continued from the Previous Page)

Baud Rate (bps)	40 MHz		48 MHz		72 MHz		80 MHz	
	Value	ERR	Value	ERR	Value	ERR	Value	ERR
8 M	4	0	5	0	8	0	9	0
6 M	-	-	7	0	11	0	-	-
5 M	7	0	-	-	-	-	15	0
4 M	9	0	11	0	17	0	19	0
2.5 M	15	0	-	-	-	-	31	0
2 M	19	0	23	0	35	0	39	0
1 M	39	0	47	0	71	0	79	0
500000	79	0	95	0	143	0	159	0
460800	86	-0.22	103	0.16	155	0.16	173	-0.22
250000	159	0	191	0	287	0	319	0
230400	173	-0.22	207	0.16	312	-0.16	346	0.06
153600	259	0.16	312	-0.16	468	-0.05	520	-0.03
125000	319	0	383	0	575	0	639	0
115200	346	0.06	416	-0.08	624	0	693	0.06
76800	520	-0.03	624	0	937	-0.05	1041	-0.03
57600	693	0.06	832	0.04	1249	0	1388	<0.01
38400	1041	-0.03	1249	0	1874	0	2082	0.01
28800	1388	<0.01	1666	-0.02	2499	0	2777	<0.01
19200	2082	0.01	2499	0	3749	0	4166	-0.01
10417	3839	<0.01	4607	<0.01	6911	<0.01	7679	0
9600	4166	<0.01	4999	0	7499	0	8332	0
7200	5555	<0.01	6666	<0.01	9999	0	11110	0
4800	8332	<0.01	9999	0	14999	0	16666	0
2400	16666	<0.01	19999	0	29999	0	-	-
1200	-	-	-	-	-	-	-	-
600	-	-	-	-	-	-	-	-
300	-	-	-	-	-	-	-	-

- Value: Setting of the BGR1/0 register
- ERR: Baud rate error (%)

Functions of the Reload Counters

The reload counters include the transmission reload counter and the reception reload counter. They function as a dedicated baud rate generator. They consist of 15-bit registers for reload values, and generate transmission/reception clocks from their internal clocks.

Start of Counting

When a reload value is written to the baud rate generator registers (BGR1 and BGR0), the reload counters start counting.

Restarting

The reload counter restarts under the conditions described below.

For both Transmission and Reception Reload Counters

Programmable reset (SCR:UPCL bit)

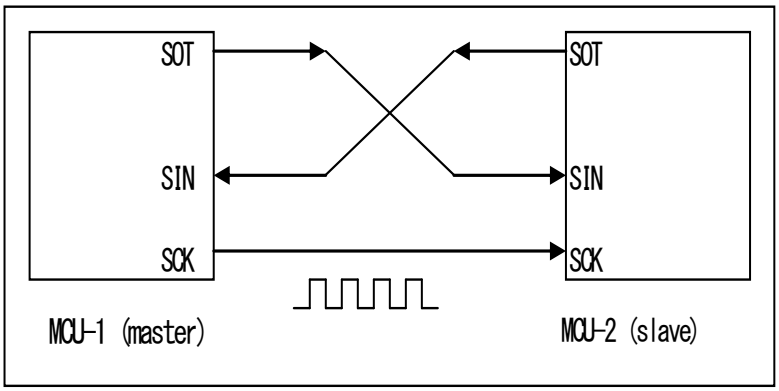
7.2. Setup Procedure and Program Flow for the CSIO (Clock Synchronous Serial Interface)

The CSIO (Clock Synchronous Serial Interface) can perform serial bidirectional transmission.

Connection Between MCUs

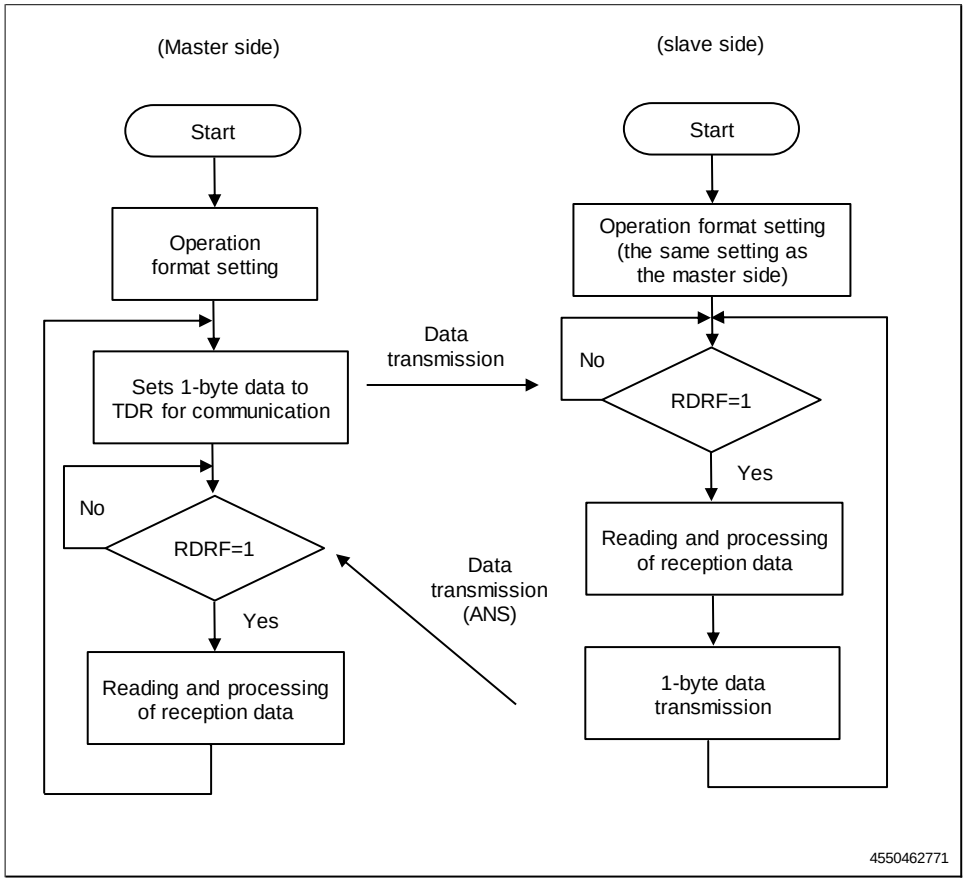
For the CSIO (Clock Synchronous Serial Interface), select bidirectional communication. As shown in Figure 7-1, 2 MCUs are interconnected.

Figure 7-1 Connection Example for CSIO (Clock Synchronous Serial Interface) Bidirectional Communication



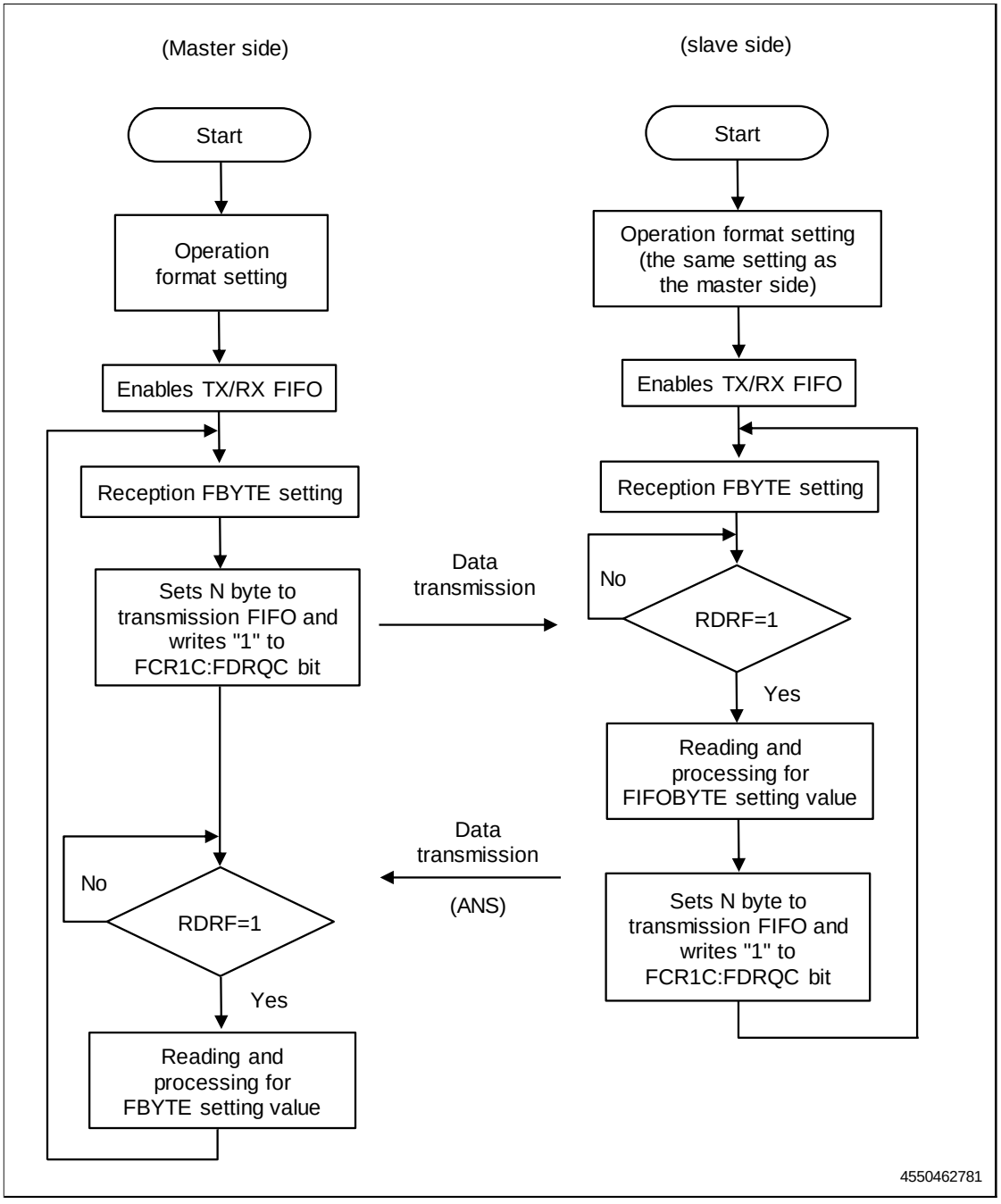
■ FIFO not used

Figure 7-2 Example of Bidirectional Communication Flow Chart (FIFO Not Used)



■ FIFO used

Figure 7-3 Example of Bidirectional Communication Flow Chart (FIFO Used)



8. Block Transfer

This section describes the operation of block transfer.

Setting Block Transfer

Block transfer can be performed for reception by setting RXBLKEN in the extended control register (ECR) to 1 and for transmission by setting TXBLKEN to 1.

Block transfer can be performed in that mode in which it is operated with the READ/WRITE access standard (RW access mode).

The block size can be set with the transmission block size register and the FBYTE register (for reception).

The threshold at which to output a block transfer request can be set with the FBYTE register for reception and with the FTICR register for transmission. When the amount of data in the reception FIFO exceeds the setting made for the FBYTE register, a reception block transfer request is output. When the amount of free space in the transmission FIFO exceeds the setting made for the FTICR register, a transmission block transfer request is output.

8.1. RW Access Mode

This section describes block transfer in RW access mode.

Reception Block Transfer

Reception block transfer is performed as described below.

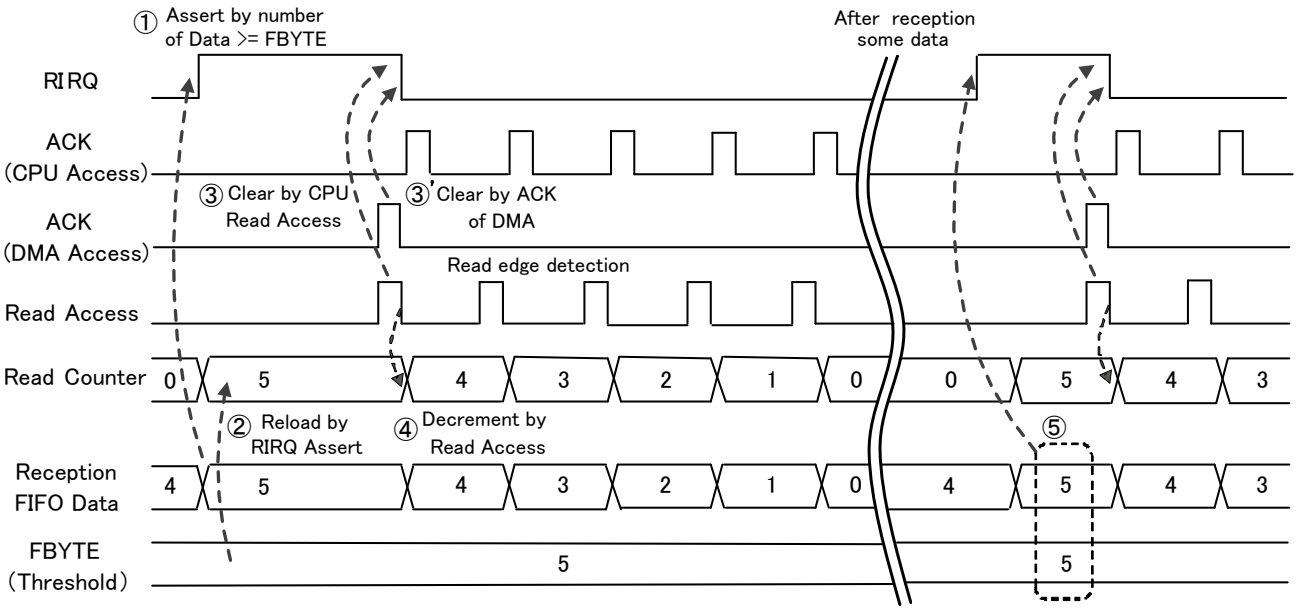
If the amount of data in the reception FIFO is same or more than the threshold (FBYTE), a reception block transfer request (RIRQ) is asserted. At this time, the value of the reception block size is reloaded from the FBYTE register to the Read counter. When the reception FIFO is not used, the value of the reception block size is "1".

After the reception block transfer request, if a reading a reception data occurs, the reception block transfer request (RIRQ) is cleared. With a reading a reception data, the Read counter is decremented.

When the Read counter becomes "0", if the amount of data in the reception FIFO is same or more than the threshold (FBYTE), the reception block transfer request (RIRQ) is asserted again. With the assertion of the reception block transfer request (RIRQ), the Read counter is reloaded.

During application operation, if the next block transfer request occurs before the completion of the previous block transfer, this state might be reported to the system by a block transfer error. This state does not occur if you configure FBYTE=1. Completion of block transfer by DMA depends on occupation of bus transaction by application. You need to fully evaluate the application integrally, design the system, and choose the optimal number of block transfers.

Figure 8-1 Reception Block Transfer in RW Access Mode



Transmission Block Transfer

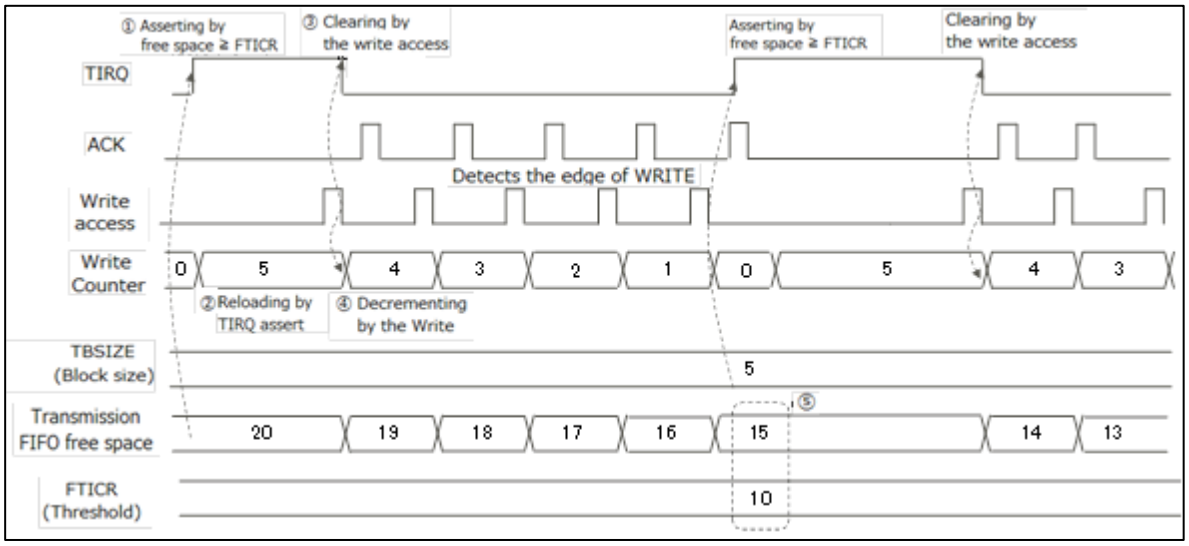
Transmission block transfer is performed as described below.

If the amount of free space in the transmission FIFO exceeds the threshold (FTICR), a transmission block transfer request (TIRQ) is asserted. At this time, the value of the transmission block size is reloaded from the TBSIZE register to the Write counter. When the transmission FIFO is not used, the value of the transmission block size is "1".

After the transmission block transfer request, if a writing a transmission data occurs, the transmission block transfer request (TIRQ) is cleared. With a writing a transmission data, the Write counter is decremented.

When the Write counter becomes "0", if the amount of free space in the transmission FIFO exceeds the threshold (FTICR), the transmission block transfer request (TIRQ) is asserted again. With the assertion of the transmission block transfer request (TIRQ), the Write counter is reloaded.

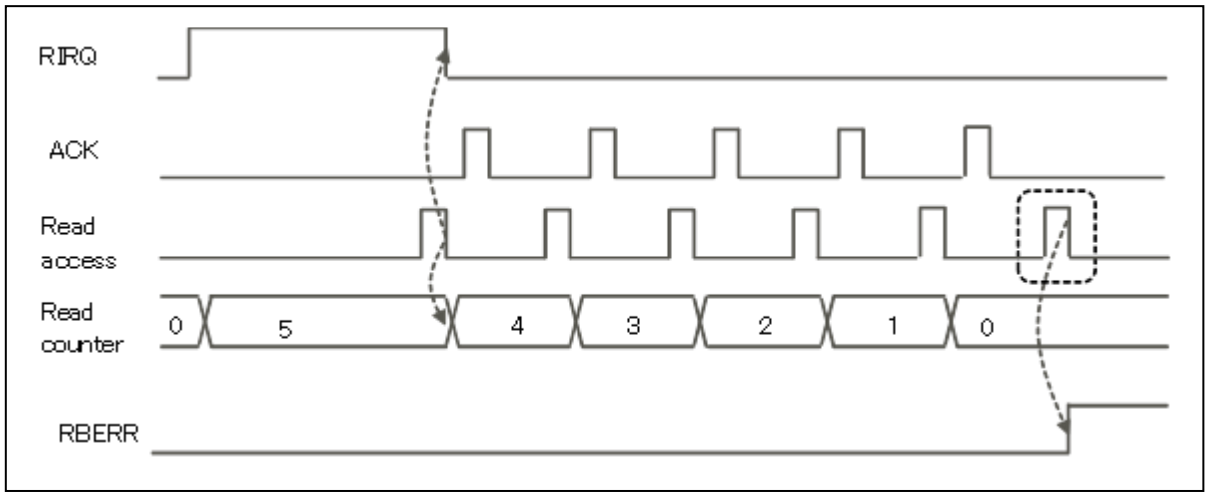
Figure 8-2 Transmission Block Transfer in RW Access Mode



Reception Block Transfer Error

If, while the Read counter is "0", a read access occurs before the next reception block transfer request (RIRQ) is asserted, a reception block transfer error occurs, and the ESR:RBERR bit is asserted.

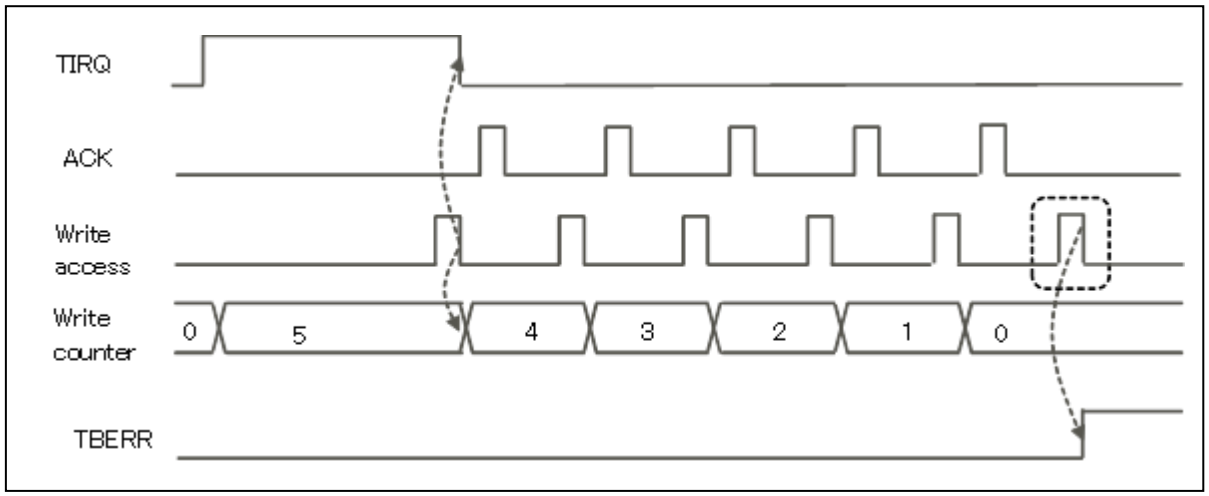
Figure 8-3 Reception Block Transfer Error in RW Access Mode



Transmission Block Transfer Error

If, while the Write counter is "0", a write access occurs before the next transmission block transfer request (TIRQ) is asserted, a transmission block transfer error occurs, and the ESR:TBERR bit is asserted.

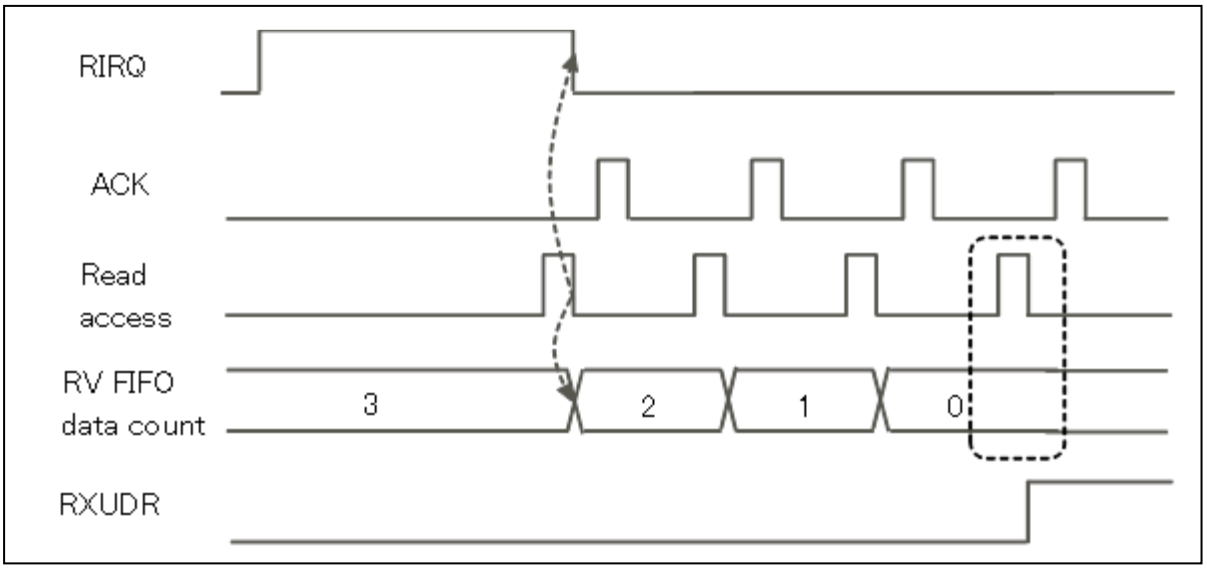
Figure 8-4 Transmission Block Transfer Error in RW Access Mode



Reception FIFO under Run

If, while the amount of data in the reception FIFO is "0", reception data is read from the reception FIFO, a reception FIFO under run occurs, and the ESR:RXUDR bit is asserted.

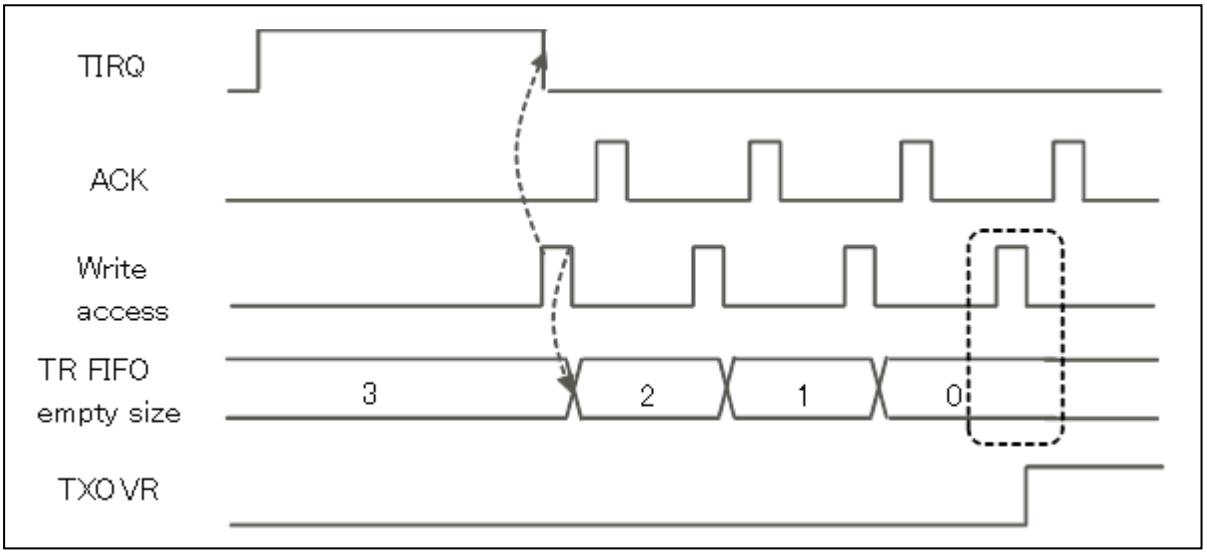
Figure 8-5 Reception FIFO Under Run in RW Access Mode



Transmission FIFO Overrun

If, while the amount of free space in the transmission FIFO is "0", transmission data is written to the transmission FIFO, a transmission FIFO overrun occurs, and the ESR:TXOVR bit is asserted.

Figure 8-6 Transmission FIFO Overrun in RW Access Mode



9. Registers of the CSIO (Clock Synchronous Serial Interface)

This section lists the registers of the CSIO (Clock Synchronous Serial Interface).

CSIO (Clock Synchronous Serial Interface) Register List

Table 9-1 CSIO (Clock Synchronous Serial Interface) Register List

	bit15 bit8	bit7 bit0
CSIO	SCR (serial control register)	SMR (serial mode register)
	SSR (serial status register)	ESCR (extended communication control register)
	RDR1/TDR1 (reception data register / transmission data register)	RDR0/TDR0 (reception data register / transmission data register)
	RDR3/TDR3 (reception data register / transmission data register)	RDR2/TDR2 (reception data register / transmission data register)
	SACSR1 (serial auxiliary control status register 1)	SACSR0 (serial auxiliary control status register 0)
	STMR1 (serial timer register 1)	STMR0 (serial timer register 0)
	STMCR1 (serial timer comparison register 1)	STMCR0 (serial timer comparison register 0)
	SCSCR1 (serial chip select control status register 1)	SCSCR0 (serial chip select control status register 0)
	SCSTR1 (serial chip select timing register 1)	SCSTR0 (serial chip select timing register 0)
	SCSTR3 (serial chip select timing register 3)	SCSTR2 (serial chip select timing register 2)
	SCSFR1 (serial chip select format register 1)	SCSFR0 (serial chip select format register 0)
	-	SCSFR2 (serial chip select format register 2)
	TBYTE1 (transfer byte register 1)	TBYTE0 (transfer byte register 0)
	TBYTE3 (transfer byte register 3)	TBYTE2 (transfer byte register 2)
	BGR1 (baud rate generator register 1)	BGR0 (baud rate generator register 0)
	-	-
FIFO	FCR1 (timer control register 1)	FCR0 (timer control register 0)
	FBYTE2 (FIFO2 byte register)	FBYTE1 (FIFO1 byte register)
	FTICR2 (transmission FIFO interrupt control register 2)	FTICR1 (transmission FIFO interrupt control register 1)
Auxiliary Register	ESR (extended status register)	ECR (extended control register)
	-	TBSIZE (transmission block size register)
	-	-
Clear Register	-	-
	-	-
	SACSR1C (serial auxiliary control status clear register 1)	SACSR0C (serial auxiliary control status clear register 0)
	-	-
	-	-
	-	-
	-	-
	-	-
	FCR1C (FIFO control clear register 1)	FCR0C (FIFO control clear register 0)
	-	-
	-	-
	ESRC (extended status clear register)	-

	bit15 bit8	bit7 bit0
Set Register	-	-
	-	-
	SACSR1S (serial auxiliary control status set register 1)	SACSR0S (serial auxiliary control status set register 0)
	-	-
	-	-
	-	-
	-	-
	-	-
	FCR1S (FIFO control set register 1)	FCR0S (FIFO control set register 0)
	-	-
Reception/Transmission Register	RDR1/TDR1 (transmission/reception data register 1)*1	RDR0/TDR0 (transmission/reception data register)*1
	RDR3/TDR3 (transmission/reception data register 1)*1	RDR2/TDR2 (transmission/reception data register)*1
	RDR1/TDR1 (transmission/reception data register 1)*1	RDR0/TDR0 (transmission/reception data register)*1
	RDR3/TDR3 (transmission/reception data register 1)*1	RDR2/TDR2 (transmission/reception data register)*1
	RDR1/TDR1 (transmission/reception data register 1)*1	RDR0/TDR0 (transmission/reception data register)*1
	RDR3/TDR3 (transmission/reception data register 1)*1	RDR2/TDR2 (transmission/reception data register)*1
	RDR1/TDR1 (transmission/reception data register 1)*1	RDR0/TDR0 (transmission/reception data register)*1
	RDR3/TDR3 (transmission/reception data register 1)*1	RDR2/TDR2 (transmission/reception data register)*1
	RDR1/TDR1 (transmission/reception data register 1)*1	RDR0/TDR0 (transmission/reception data register)*1
	RDR3/TDR3 (transmission/reception data register 1)*1	RDR2/TDR2 (transmission/reception data register)*1
	RDR1/TDR1 (transmission/reception data register 1)*1	RDR0/TDR0 (transmission/reception data register)*1
	RDR3/TDR3 (transmission/reception data register 1)*1	RDR2/TDR2 (transmission/reception data register)*1
	RDR1/TDR1 (transmission/reception data register 1)*1	RDR0/TDR0 (transmission/reception data register)*1
	RDR3/TDR3 (transmission/reception data register 1)*1	RDR2/TDR2 (transmission/reception data register)*1
	RDR1/TDR1 (transmission/reception data register 1)*1	RDR0/TDR0 (transmission/reception data register)*1
	RDR3/TDR3 (transmission/reception data register 1)*1	RDR2/TDR2 (transmission/reception data register)*1
	RDR1/TDR1 (transmission/reception data register 1)*1	RDR0/TDR0 (transmission/reception data register)*1
	RDR3/TDR3 (transmission/reception data register 1)*1	RDR2/TDR2 (transmission/reception data register)*1
	RDR1/TDR1 (transmission/reception data register 1)*1	RDR0/TDR0 (transmission/reception data register)*1
	RDR3/TDR3 (transmission/reception data register 1)*1	RDR2/TDR2 (transmission/reception data register)*1
	RDR1/TDR1 (transmission/reception data register 1)*1	RDR0/TDR0 (transmission/reception data register)*1
	RDR3/TDR3 (transmission/reception data register 1)*1	RDR2/TDR2 (transmission/reception data register)*1
	RDR1/TDR1 (transmission/reception data register 1)*1	RDR0/TDR0 (transmission/reception data register)*1
	RDR3/TDR3 (transmission/reception data register 1)*1	RDR2/TDR2 (transmission/reception data register)*1
	RDR1/TDR1 (transmission/reception data register 1)*1	RDR0/TDR0 (transmission/reception data register)*1
	RDR3/TDR3 (transmission/reception data register 1)*1	RDR2/TDR2 (transmission/reception data register)*1
	RDR1/TDR1 (transmission/reception data register 1)*1	RDR0/TDR0 (transmission/reception data register)*1
	RDR3/TDR3 (transmission/reception data register 1)*1	RDR2/TDR2 (transmission/reception data register)*1
	RDR1/TDR1 (transmission/reception data register 1)*1	RDR0/TDR0 (transmission/reception data register)*1
	RDR3/TDR3 (transmission/reception data register 1)*1	RDR2/TDR2 (transmission/reception data register)*1
	RDR1/TDR1 (transmission/reception data register 1)*1	RDR0/TDR0 (transmission/reception data register)*1
	RDR3/TDR3 (transmission/reception data register 1)*1	RDR2/TDR2 (transmission/reception data register)*1
	RDR1/TDR1 (transmission/reception data register 1)*1	RDR0/TDR0 (transmission/reception data register)*1
	RDR3/TDR3 (transmission/reception data register 1)*1	RDR2/TDR2 (transmission/reception data register)*1
	RDR1/TDR1 (transmission/reception data register 1)*1	RDR0/TDR0 (transmission/reception data register)*1
	RDR3/TDR3 (transmission/reception data register 1)*1	RDR2/TDR2 (transmission/reception data register)*1

*1: Access by the mirror address area

9.1. Serial Control Register (SCR)

The serial control register (SCR) is used to enable/disable transmission/reception interrupts, enable/disable transmission idle interrupts, and enable/disable transmission/reception operations. It can also be used to make settings for connection to SPI and to reset the CSIO.

Serial Control Register (SCR)

Figure 9-1 shows the bit configuration of the serial control register (SCR).

Figure 9-1 Bit Configuration of Serial Control Register (SCR)

Bit	15	14	13	12	11	10	9	8	7 ... 0
Field	UPCL	MS	SPI	RIE	TIE	TBIE	RXE	TXE	(SMR)
R/W Attribute	R0,W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Protection Attribute	-	-	-	-	-	-	-	-	
Initial Value	0	0	0	0	0	0	0	0	

[bit15] UPCL: Programmable Clear Bit

This bit initializes the internal state of the CSIO.

When "1" is set:

- Resets the CSIO directly (software reset). However, the register setting is retained. At this time, anything in the transmission/reception state is disconnected immediately.
- The baud rate generator reloads the setting of the BGR1/0 register and then restarts.
- All transmission/reception and status interrupt factors (SSR:TDRE, TBI, RDRF, ORE, SACSR:TINT, CSE[*]) are initialized.
- All serial chip select pins become inactive.

When "0" is set:

Does not have any effect on the operation.

At reading, "0" is always read.

Bit	Description	
	Write	Read
0	No effect	"0" is always read.
1	Programmable clear	

Notes:

- *Execute programmable clear (SCR:UPCL = 1) after disabling interrupts.*
- *When FIFOs are used, execute programmable clear after disabling the FIFOs (FCR0:FE2, FE1 = 0).*
- *Executing programmable clear (SCR:UPCL = 1) does not clear the value of the serial timer register (STMR).*

[bit14] MS: Master/Slave Function Selection Bit

This bit selects either master or slave mode.

- 0: Set master mode.
- 1: Set slave mode.

Bit	Description
0	Master mode
1	Slave mode

Notes:

- *If slave mode is selected, an external clock is input directly, provided that SMR:SCKE = 0.*
- *Change this bit when transmission and reception are disabled (SCR:TXE = RXE = 0).*
- *After setting the MS bit, enable reception (RXE = 1).*

[bit13] SPI: SPI-Supporting Bit

This bit is for performing communication that supports SPI. When chip select is used in master mode (SCR:MS = 0), the bit is used for communication with serial chip select pin 0.

- 0: Perform normal synchronous communication.
- 1: Support SPI.

Bit	Description
0	Normal synchronous transfer
1	SPI support

Notes:

- *Set this bit when transmission and reception are disabled (TXE = RXE = 0).*
- *This bit is used in any of the following cases:*
  - The chip select pins are disabled (SCSCR:CSEN3 to CSEN0 = 0b0000).
  - In slave mode (SCR:MS = 1).
  - When the chip select data format is disabled (ESCR:CSFE = 0).
  - When the chip select data format is enabled (ESCR:CSFE = 1) and serial chip select pin 0 is active.

[bit12] RIE: Reception Interrupt Enable Bit

- This bit enables/disables reception interrupt request output to the CPU.
- For ECR:EISEL = 0, a reception interrupt request is issued when the RIE bit and the reception data flag bit (SSR:RDRF) are "1" or when one of the error flag bits (SSR:ORE, ESR:RXUDR, RBERR) is "1".
- For ECR:EISEL = 1, a reception interrupt request is issued when the SMR:RIE bit and the reception data flag bit (SSR:RDRF) are both "1".

Bit	Description
0	Disable reception interrupts.
1	Enable reception interrupts.

[bit11] TIE: Transmission Interrupt Enable Bit

- This bit enables/disables transmission interrupt request output to the CPU.
- For ECR:EISEL = 0, a transmission interrupt request is issued when the TIE bit and the SSR:TDRE bit are set to "1" or when one of the error flag bits (ESR:TXOVR, TBERR) is set to "1".
- For ECR:EISEL = 1, a transmission interrupt request is issued when the TIE bit and the SSR:TDRE bit are both "1" or when one of the error flag bits (ESR:TXOVR, TBERR) is "1".

Bit	Description
0	Disable transmission interrupts.
1	Enable transmission interrupts.

[bit10] TBIE: Transmission Bus Idle Interrupt Enable Bit

- This bit enables/disables transmission bus idle interrupt request output to the CPU.
- A transmission bus idle interrupt request is output when the TBIE bit and the SSR:TBI bit are both "1".

Bit	Description
0	Disable transmission bus idle interrupts.
1	Enable transmission bus idle interrupts.

[bit9] RXE: Reception Operation Enable Bit

This bit enables/disables CSIO reception.

- 0: Disable data frame reception.
- 1: Enable data frame reception.

Bit	Description
0	Disable reception.
1	Enable reception.

Notes:

- *If reception is disabled (RXE = 0) during reception, reception is stopped immediately.*
- *After setting the MS bit and the SMR:SCINV bit, enable reception (RXE = 1).*

[bit8] TXE: Transmission Operation Enable Bit

This bit enables/disables CSIO transmission.

- 0: Disable data frame transmission.
- 1: Enable data frame transmission.

Bit	Description
0	Disable transmission.
1	Enable transmission.

Notes:

- *If transmission is disabled (RXE = 0) during transmission, transmission is stopped immediately.*
- *When serial chip select is used (SCSCR:CSEN = 1) in master mode (SCR:MS = 0), perform a programmable reset (SCR:UPCL = 1) after disabling transmission.*

9.2. Serial Mode Register (SMR)

The serial mode register (SMR) is used to set the operating mode, transfer direction, and data length, invert the serial clock, and enable/disable the output of serial data and clocks to pins.

Serial Mode Register (SMR)

Figure 9-2 shows the bit configuration of the serial mode register (SMR).

Figure 9-2 Bit Configuration of Serial Mode Register (SMR)

Bit	15 ... 8	7	6	5	4	3	2	1	0
Field	(SCR)	MD2	MD1	MD0	WUCR	SCINV	BDS	SCKE	SOE
R/W Attribute		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute		-	-	-	-	-	-	-	-
Initial Value		0	0	0	0	0	0	0	0

[bit7:5] MD[2:0]: Operation Mode Setting Bits

These bits set the operation modes.

Bits			Description
0	0	0	Operation mode 0 (asynchronous normal mode)
0	0	1	Operation mode 1 (asynchronous multi-processor mode)
0	1	0	Operation mode 2 (clock synchronous mode)
0	1	1	Operation mode 3 (LIN communication mode)
1	0	0	Operation mode 4 (I2C mode)

* This section describes the registers and their operations in operation mode 2.

Notes:

- *Settings other than above are prohibited.*
- *To switch the operation mode, execute programmable clear (SCR:UPCL = 1) and then switch the operation mode.*
- *Set the registers after setting an operation mode.*

[bit4] WUCR: WAKE UP Control Bit

WUCR is not supported

This bit selects the pin used for external interrupts.

0: The pin used for external interrupts is the INT pin.

1: The pin used for external interrupts is the SCK pin.

Bit	Description
0	Disable the WAKE UP function.
1	Enable the WAKE UP function.

Note:

- *The WAKE UP function is not supported.*

[bit3] SCINV: Serial Clock Invert Bit

This bit inverts the serial clock format. When chip select is used in master mode (SCR:MS = 0), this bit is used for communication with serial chip select pin 0.

0:

- The mark level of serial clock output is set to "H".
- During normal transfer, transmission data is output in synchronization with a falling edge of the serial clock. In SPI transfer, it is output in synchronization with a rising edge of the serial clock.
- During normal transfer, reception data is sampled at a rising edge of the serial clock. In SPI transfer, it is sampled at a falling edge of the serial clock.

1:

- The mark level of serial clock output is set to "L".
- During normal transfer, transmission data is output in synchronization with a rising edge of the serial clock. During SPI transfer, it is output in synchronization with a falling edge of the serial clock.
- During normal transfer, reception data is sampled at a falling edge of the serial clock. In SPI transfer, it is sampled at a rising edge of the serial clock.

Bit	Description
0	Mark level "H" format
1	Mark level "L" format

Notes:

- *Set this bit when the serial clock output is disabled (SCKE = 0).*
- *After setting the SCINV bit, enable reception (SCR:RXE = 1).*
- *Set this bit when transmission and reception are disabled (SCR:TXE = RXE = 0).*
- *This bit is used in any of the following cases:*
  - The chip select pins are disabled (SCSCR:CSEN3 to CSEN0 = 0b0000).
  - In slave mode (SCR:MS = 1).
  - When the chip select data format is disabled (ESCR:CSFE = 0).
  - When the chip select data format is enabled (ESCR:CSFE = 1) and serial chip select pin 0 is active.
- *Set this bit to "0" when serial test mode is enabled (SACSR:STST = 1).*

[bit2] BDS: Transfer Direction Selection Bit

This bit selects whether the lowest bit is transferred first (LSB first, BDS = 0) or the highest bit is transferred first (MSB first, BDS = 1), when transferring serial data. When chip select is used in master mode (SCR:MS = 0), the bit is used for communication with serial chip select pin 0.

Bit	Description
0	LSB first (lowest bit is transferred first)
1	MSB first (highest bit is transferred first)

Notes:

- *Set this bit when transmission and reception are disabled (SCR:TXE = RXE = 0).*
- *This bit is used in any of the following cases:*
  - The chip select pins are disabled (SCSCR:CSEN3 to CSEN0 = 0b0000).
  - In slave mode (SCR:MS = 1).
  - When the chip select data format is disabled (ESCR:CSFE = 0).
  - When the chip select data format is enabled (ESCR:CSFE = 1) and serial chip select pin 0 is active.

[bit1] SCKE: Serial Clock Input/Output Enable Bit

This bit enables the input or output of the serial clock.

Bit	Description
0	Enable the Serial clock input.
1	Enable the Serial clock output.

Notes:

- *Set this bit to "0" in slave mode (SCR:MS = 1) and set this bit to "1" in master mode (SCR:MS = 0).*
- *Enable serial clock output (SCKE = 1) after setting the SCINV bit.*

[bit0] SOE: Serial Data Output Enable Bit

This bit enables/stops the output of serial data.

Bit	Description
0	Stop the serial data output.
1	Enable the serial data output.

Note:

- *In the case of serial data output (SOE = 1), the SOUT pin operates as the SOUT pin regardless of the setting of the general-purpose I/O port (DDR).*

9.3. Serial Status Register (SSR)

The serial status register (SSR) is used to verify the transmission/reception state, verify the reception error flag, or clear the reception error flag.

Serial Status Register (SSR)

Figure 9-3 shows the bit configuration of the serial status register (SSR).

Figure 9-3 Bit Configuration of Serial Status Register (SSR)

Bit	15	14	13	12	11	10	9	8	7 ... 0
Field	REC	Reserved	Reserved	AWC	ORE	RDRF	TDRE	TBI	(ESCR)
R/W Attribute	R0,W	R0, W0	R0, W0	R/W	R, WX	R, WX	R, WX	R, WX	
Protection Attribute	-	-	-	-	-	-	-	-	
Initial Value	0	0	0	0	0	0	1	1	

[bit15] REC: Reception Error Flag Clear Bit

This bit clears the ORE flag of the serial status register (SSR).

- Writing "1" clears the error flag.
- Writing "0" does not have any effect.

Upon reading, "0" is always read.

Bit	Description	
	Write	Read
0	No effect	"0" is always read.
1	Clear the reception error flag (ORE).	

[bit14:13] Reserved: Reserved Bits

[bit12] Access width Control Bit

This bit selects between 16- and 32-bit access when accessing the transmission data register (TDR) and the reception data register (RDR).

Bit	Description
0	16-bit access
1	32-bit access

Notes:

- *Change this bit only when transmission and reception are disabled (SCR:TXE = RXE = 0) and TDR and RDR are empty (SSR:TDRE = 1, SSR:RDRF = 0).*
- *If the data length is 20, 24, or 32 bits, set "1" in this bit.*
- *If the data length is 8 bits or less, the TDR and RDR registers can be byte-accessed.*

[bit11] ORE: Overrun Error Flag Bit

- This bit is set to "1" when an overrun error occurs during data reception, and is cleared when "1" is written to the REC bit of the serial status register (SSR).
- A reception interrupt request is issued when the ORE bit and the SCR:RIE bit are "1".
- Please refer to "CHAPTER: INTERRUPT CONTROLLER" for interrupt request output
- If this flag is set, the reception data register (RDR) data will be invalid.
- If this flag is set when the reception FIFO is used, the reception FIFO enable bits are cleared, and no reception data is stored in the reception FIFO.

Bit	Description
0	No overrun error
1	Overrun error found

Note:

- *This bit is cleared by the Software reset (SCR:UPCL=1).*

[bit10] RDRF: Reception Data Full Flag Bit

- This flag indicates the status of the reception data register (RDR).
- This flag is set to "1" when reception data is loaded to the RDR, and is cleared to "0" when the data in the reception data register (RDR) is read.
- A reception interrupt request is issued when the RDRF bit and the SCR:RIE bit are both "1".
- When the reception FIFO is used, if the reception FIFO receives a prescribed number of data items, RDRF is set to "1".
- When the reception FIFO is used, both of the following conditions are satisfied, and the reception idle state continues for at least 8 clock pulses of the baud rate clock, RDRF is set to"1".
  - The reception FIFO idle detection enable bit (FCR1:FRIIE) is set to "1".
  - The reception FIFO has not received the prescribed number of data items, and data remains in the reception FIFO.

  During an 8-clock count, the counter is reset to 0 when RDR is read, and the system starts counting the 8 clocks again.
- For ECR:RXBLKEN = 0, when the reception FIFO is used, this bit is cleared to "0" when the reception FIFO empties as a result of data being read from it.
- For ECR:RXBLKEN = 1, when the reception FIFO is used, this bit is cleared to "0" when the number of data items in the reception FIFO becomes equal to or less than the value set for the FBYTE register.

Bit	Description
0	The reception data register RDR is empty.
1	The reception data register RDR contains data.

Note:

- *When the reception FIFO is used and RDRF has become "1", resetting the reception FIFO (FCR0:FCL2, FCL1 = 1) does not cause RDRF to be set to "0". Therefore, in order to set RDRF to "0" after resetting the reception FIFO, perform a dummy reading of the reception data register during the reception disable status (SCR:RXE = 0).*

– *This bit is cleared by the Software reset (SCR:UPCL=1).*

[bit9] TDRE: Transmission Data Empty Flag Bit

- This bit indicates the status of the transmission data register (TDR).
- Writing transmission data to the TDR sets this bit to "0", indicating that the TDR contains valid data. Loading the data into the transmission shift register to start transmission sets this bit to "1", indicating that the TDR does not contain valid data. A transmission interrupt request is issued when the TDRE bit and the SCR:TIE bit are both "1".
- The TDRE bit is set to "1" when programmable reset is executed (SCR:UPCL=1).
- For the set/reset timing of the TDRE bit when using the transmission FIFO, see "2.4 Interrupt Generation and Flag Set Timing When the Transmission FIFO Is Used."

Bit	Description
0	The transmission data register (TDR) contains data.
1	The transmission data register is empty.

[bit8] TBI: Transmission Bus Idle Flag Bit

- This bit indicates that the CSIO is not performing transmission.
- If data is written to the transmission data register (TDR), this bit is set to "0".
- When the transmission data register (TDR) is empty (TDRE = 1), if the serial chip select pin is deselected and transmission is not being performed, this bit is set to "1".
- The TBI bit is set to "1" when programmable reset is executed (SCR:UPCL=1).
- When this bit is "1", if transmission bus idle interrupts are enabled (SCR:TBIE = 1), a transmission interrupt request is output.

Bit	Description
0	Transmission in progress
1	No transmission

Note:

– *When the transmission data register (TDR) is empty (TDRE = 1), if a serial chip select error (CSE = 1) occurs, this bit is set to "1" within the baud rate period.*

9.4. Extended Communication Control Register (ESCR)

The extended communication control register (ESCR) can be used to set the transmission/reception data length and fix the serial output to "H".

Bit Configuration of the Extended Communication Register (ESCR)

Figure 9-4 shows the bit configuration of the extended communication control register (ESCR).

Figure 9-4 Bit Configuration of Extended Communication Control Register (ESCR)

Bit	15 ... 8	7	6	5	4	3	2	1	0
Field	-	SOP	L3	CSFE	WT1	WT0	L2	L1	L0
R/W Attribute		R0,W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute		-	-	-	-	-	-	-	-
Initial Value		0	0	0	0	0	0	0	0

[bit7] SOP: Serial Output Pin Set Bit

- This bit sets the serial output pin to "H". When "1" is written to this bit, the SOUT pin is set to "H". Later, "0" need not be written to this bit.
- When this bit is read, "0" is always read.

Bit	Description	
	Write	Read
0	No effect	"0" is always read.
1	Set the SOUT pin to "H".	

Note:

- *Do not set this bit while transmitting serial data.*
- *To set Serial output (SOT) pin to “High”, follow one of the below mentioned methods.*
  1. Execute programmable clear. (Set SCR.UPCL=”1”)
  - After disabling interrupt, execute programmable clear. (Set SCR.UPCL=”1”)
  - When FIFOs are used, execute programmable clear after disabling the FIFOs (FCR0:FE2, FE1 = 0).
  2. Disable transmission. (Set SCR.TXE=”0”)
  - When serial chip select is used (SCSCR:CSEN = 1) in master mode (SCR:MS = 0), perform a programmable reset (SCR:UPCL = 1) after disabling transmission.
  3. Use Bit Band Unit to set ESCR.SOP bit.

[bit5] CSFE: Serial Chip Select Format Enable Bit

This bit enables or disables the setting of a format for each serial chip select pin.

When this bit is set to "1", make the settings described below for each serial chip select pin.

- Inactive level of serial chip select
- Mark level of the serial clock
- Selection between SPI transfer and normal transfer
- Serial data transfer direction
- Data length of serial data

Bit	Description
0	Set the same data format and clock format for all of the serial chip select pins.
1	Set a data format and clock format for each serial chip select pin.

Notes:

- *The setting of this bit is invalid in any of the cases described below.*
  - The chip select pins are disabled (SCSCR0:CSEN3 to CSEN0 = 0b0000).
  - In slave mode (SCR:MS = 1).
- *Set this bit when transmission is disabled (SCR:TXE = 0).*

[bit4:3] WT[1:0]: Data Transmission/Reception Wait Select Bits

These bits specify the number of waits for the transmission or reception of continuous data in master mode. In slave mode, the operation is that for "00".

- "00": Output SCK continuously.
- "01": Output SCK after a 1-bit time wait.
- "10": Output SCK after a 2-bit time wait.
- "11": Output SCK after a 3-bit time wait.

Bits		Description
0	0	0 bit
0	1	1 bit
1	0	2 bits
1	1	3 bits

[bit6, bit2:0] L3, L2, L1, L0: Data Length Selection Bits

These bits specify the data length of the transmission/reception data. When chip select is used in master mode (SCR:MS = 0), this bit is used for communication with serial chip select pin 0.

L3	L2	L1	L0	Data Length
0	0	0	0	8-bit length
0	0	0	1	5-bit length
0	0	1	0	6-bit length
0	0	1	1	7-bit length
0	1	0	0	9-bit length
0	1	0	1	10-bit length
0	1	1	0	11-bit length
0	1	1	1	12-bit length
1	0	0	0	13-bit length
1	0	0	1	14-bit length
1	0	1	0	15-bit length
1	0	1	1	16-bit length
1	1	0	0	20-bit length
1	1	0	1	24-bit length
1	1	1	0	32-bit length

Notes:

- *Settings other than those described above are prohibited.*
- *Change these bits when transmission and reception are disabled (SCR:TXE = RXE = 0).*
- *These bits are used for any of the following:*
  - The chip select pins are disabled (SCSCR:CSEN3 to CSEN0 = 0b0000).
  - In slave mode (SCR:MS = 1).
  - When the chip select data format is disabled (ESCR:CSFE = 0).
  - When the chip select data format is enabled (ESCR:CSFE = 1) and serial chip select pin 0 is active.

9.5. Reception Data Register/Transmission Data Register (RDR/TDR)

The reception data register and transmission data register are placed at the same address. If read, the register functions as a reception data register, while if written to, it functions as a transmission data register. When FIFO operation is enabled, the RDR/TDR address is the FIFO read/write address.

Reception data register (RDR)

Figure 9-5 shows the bit configuration of the serial reception register (RDR).

Figure 9-5 Bit Configuration of Reception Data Register (RDR)

Bit	31	30	29	28	27	26	25	24
Field	D31	D30	D29	D28	D27	D26	D25	D24
R/W Attribute	R	R	R	R	R	R	R	R
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	D23	D22	D21	D20	D19	D18	D17	D16
R/W Attribute	R	R	R	R	R	R	R	R
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	D15	D14	D13	D12	D11	D10	D9	D8
R/W Attribute	R	R	R	R	R	R	R	R
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	D7	D6	D5	D4	D3	D2	D1	D0
R/W Attribute	R	R	R	R	R	R	R	R
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

The reception data register (RDR) is a 32-bit data buffer register that is used for serial data reception.

- Serial data signals transmitted to the serial input pin (SIN pin) is converted by the shift register, and the result is stored in the reception data register (RDR).
  According to the data length, the reception data is stored, starting with the lower bits, while the other bits are set to "0". Example: D7 to D0 = 0x45, D31 to D8 = 0x000000 if the data length is 8 bits, and 0x45 is received.
- The reception data full flag bit (SSR:RDRF) is set to "1" when reception data is stored to the reception data register (RDR). If reception interrupts are enabled (SCR:RIE = 1), a reception interrupt request is generated.
- Read the reception data register (RDR) when the reception data full flag bit (SSR:RDRF) is "1". The reception data full flag bit (SSR:RDRF) is such that if the reception data register (RDR) is read, it is automatically cleared to "0".
- If a reception error occurs (SSR:ORE), the data in the reception data register (RDR) will be invalid.
- To read the RDR, access it as described below.
  - If SSR:AWC = 0, perform 16-bit access to the lower 16 bits of the RDR.
  - If SSR:AWC = 1, perform 32-bit access.

Notes:

- *When the reception FIFO is used, RDRF is set to "1" when a predefined amount of data has been received by the reception FIFO.*
- *When the reception FIFO is used, RDRF is cleared to "0" once the reception FIFO is empty.*
- *If, while the reception FIFO is being used, a reception error occurs (SSR:ORE), the reception FIFO enable bit is cleared, and reception data is not stored in the reception FIFO.*
- *When AWC = 0, do not access D31 to D16.*

Transmission Data Register (TDR)

Figure 9-6 shows the bit configuration of the transmission data register.

Figure 9-6 Bit Configuration of Transmission Data Register (TDR)

Bit	31	30	29	28	27	26	25	24
Field	D31	D30	D29	D28	D27	D26	D25	D24
R/W Attribute	W	W	W	W	W	W	W	W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	1	1	1	1	1	1	1	1

Bit	23	22	21	20	19	18	17	16
Field	D23	D22	D21	D20	D19	D18	D17	D16
R/W Attribute	W	W	W	W	W	W	W	W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8
Field	D15	D14	D13	D12	D11	D10	D9	D8
R/W Attribute	W	W	W	W	W	W	W	W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	1	1	1	1	1	1	1	1

Bit	7	6	5	4	3	2	1	0
Field	D7	D6	D5	D4	D3	D2	D1	D0
R/W Attribute	W	W	W	W	W	W	W	W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	1	1	1	1	1	1	1	1

The transmission data register (TDR) is a 32-bit data buffer register that is used for serial data transmission.

- If transmission is enabled (SCR:TXE = 1), and the data to be transmitted is written to the transmission data register (TDR), the transmission data is transferred to the transmission shift register, converted into serial data, and is sent out from the serial data output pin (SOUT pin).
- According to the data length, the transmission data is stored, starting with the lower bits, while the other bits are "invalid". Example: If the data length is 8 bits, and 0x45 is transmitted, D7 to D0 = 0x45 and D31 to D8 will be invalid.
- The transmission data empty flag (SSR:TDRE) is cleared to "0" when transmission data is written to the transmission data register (TDR).
- If the transmission FIFO is disabled or empty, the transmission data empty flag (SSR:TDRE) is set to "1" when transmission data is transferred to the transmission shift register to start transmission.
- When the transmission data empty flag (SSR:TDRE) is set to "1", the next transmission data can be written. A transmission interrupt occurs if the transmission interrupt is enabled. Write the next transmission data after the occurrence of a transmission interrupt or when the transmission data empty flag (SSR:TDRE) is set to "1".
- When the transmission data empty flag (SSR:TDRE) is set to "0", and the transmission FIFO is disabled or the transmission FIFO is full, transmission data cannot be written to the transmission data register (TDR).

- To write to the TDR, perform access as described below.
  - If SSR:AWC = 0, perform 16-bit access to the lower 16 bits of the TDR.
  - If SSR:AWC = 1, perform 32-bit access.

Notes:

- *The transmission data register is a write-only register, while the reception data register is a read-only register. The two registers are placed at the same address, so the written value differs from that which is read.*
- *For details on the set timing of the transmission data empty flag (SSR:TDRE) when the transmission FIFO is used, see "2.4 Interrupt Generation and Flag Set Timing When the Transmission FIFO Is Used."*
- *When AWC = 0, do not access D31 to D16.*

Relationship between the Transmission Data Register (TDR) and the Transmission Data Empty Flag

For 16-bit access (SSR:AWC = 0), the TDR register has a 16-bit boundary, so that, with a single write, transmission data is stored in blocks of 16 bits. If the TDR register contains 32-bit transmission data, the transmission data empty flag (SSR:TDRE) is set to "0".

For 32-bit access (SSR:AWC = 1), the TDR register has a 32-bit boundary, so that, with a single write, transmission data is stored in blocks of 32 bits.

Table 9-2 Relationship between the Transmission Data Register (TDR) and the Transmission Data Empty Flag

Data Access Width	Number of Data Items Stored in the TDR Register	TBI Flag	TDRE Flag	Transmission
16-bit access (SSR:AWC = 0)	0 bit	1	1 *1	Transmission not possible
	16 bits	0		Transmission possible
	32 bits		0	
32-bit access (SSR:AWC = 1)	0 bit	1	1	Transmission not possible
	32 bits	0	0	Transmission possible

*1: For SSR:AWC = 0, when the number of storage data items in the TDR register is 0, if 16-bit data is written, TDRE is first set to 0 and then to 1.

9.6. Serial Auxiliary Control Status Register (SACSR)

The serial auxiliary control status register (SACSR) performs the following: 1. Controls serial test operation, 2. Selects a serial timer activation method, 3. Enables/disables timer interrupts, 4. Enables/disables synchronous transmission, 5. Sets the division value for the serial timer operation clock, and 6. Enables/disables the serial timer.

Bit Configuration of the Serial Auxiliary Control Status Register (SACSR)

Figure 9-7 shows the bit configuration of the serial auxiliary control status register (SACSR).

Figure 9-7 Bit Configuration of Serial Auxiliary Control Status Register (SACSR)

Bit	15	14	13	12	11	10	9	8
Field	STST	Reserved	TBEEN	CSEIE	CSE	TRG1	TRG0	TINT
R/W Attribute	R/W	R0,W0	R/W	R/W	R,WX	R/W	R/W	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	TINTE	TSYNE	TRGE	TDIV3	TDIV2	TDIV1	TDIV0	TMRE
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

[bit15] STST : Serial test bit

This bit enables or disables serial test mode.

When serial test mode is enabled, SOUT and SIN are connected together within the multi-function serial interface, so that the data transmitted from SOUT can be received from SIN directly.

When serial test mode is enabled, the SOUT pin is fixed to "H", and any data input to the SIN pin is ignored.

- This bit is reset when the SACSRC:STSTC bit in the clear register is set to "1".
- This bit is set when the SACSRS:STSTS bit in the set register is set to "1".

Bit	Description
0	Disable serial test mode.
1	Enable serial test mode.

Notes:

- *This bit can be changed only when transmission and reception are disabled (SCR:TXE = 0, SCR:RXE = 0).*
- *Set this bit to "0" in slave mode (SCR:MS = 1).*

[bit14] Reserved : Reserved Bit

[bit13] TBEEN: Transfer Byte Error Enable Bit

In master mode (SCR:MS = 0), this bit enables/disables the occurrence of serial chip select errors.

For details, see "2.6 Chip Select Error Occurrence and Flag Set Timing".

- This bit is reset when the SACSRC:TBEENC bit in the clear register is set to "1".
- This bit is set when the SACSRS:TBEENS bit in the set register is set to "1".

Bit	Description
0	In master mode (SCR:MS = 0), disable the occurrence of chip select errors.
1	In master mode (SCR:MS = 0), enable the occurrence of chip select errors.

Note:

– *Change this bit when transmission and reception are disabled (SCR:TXE = RXE = 0).*

[bit12] CSEIE: Chip Select Error Interrupt Enable Bit

- This bit enables/disables the output of chip select error interrupt requests.
- If the CSEIE bit and the chip select error flag bit (CSE) are "1", a transmission interrupt request is output.
- This bit is reset when the SACSRC:CSEIEC bit in the clear register is set to "1".
- This bit is set when the SACSRS:CSEIES bit in the set register is set to "1".

Bit	Description
0	Disable chip select error interrupts.
1	Enable chip select error interrupts.

[bit11] CSE: Chip Select Error Flag Bit

This bit indicates whether a chip select error has occurred.

For details, see "2.6 Chip Select Error Occurrence and Flag Set Timing".

When this bit is "1" and the chip select error interrupt enable bit (CSEIE) is also "1", a transmission interrupt request is output.

Writing to this bit is invalid.

- This bit is reset when the SACSRC:CSEC bit in the clear register is set to "1".

Bit	Description
0	No chip select error has occurred.
1	A chip select error has occurred.

Notes:

- *Software reset (SCR:UPCL = 1) resets this bit to "0".*
- *When serial chip select is not used (SCSCR:CSEN0 = 0) in slave mode (SCR:MS = 1), this bit is not set to "1".*
- *When a chip select error occurs (CSE = 1), disable transmission (SCR:TXE = 0), and then write "1" to SACSRC:CSEC. To restart transmission, write "1" to SACSRC:CSEC, enable transmission (SCR:TXE = 1), and then write transmission data to the transmission data buffer (TDR).*
- *If noise of 1 bus clock or greater is generated in the serial chip select input during slave transmission, this bit may be set to "1". Should this occur, restart transmission after the end of master transfer.*

[bit10:9] TRG[1:0] : Trigger selection bits

These bits select the method for detecting the external trigger edges for activating the serial timer.

Bits		Edge Detection Method for External Trigger
0	0	Rising edge detection
0	1	Falling edge detection
1	0	Both edge detection
1	1	Setting prohibited

Notes:

- *These bits are invalid if the external trigger enable bit (TRGE) is "0".*
- *These bits can be changed only when the serial timer enable bit (TMRE) is "0".*

[bit8] TINT : Timer interrupt flag

If the serial timer register (STMR) and the serial timer comparison register (STMCR) match, the serial timer register (STMR) is set to "0", and this bit is set to "1".

When this bit is "1" and the timer interrupt enable bit (TINTE) is "1", a status interrupt request is issued.

Writing to this bit is invalid.

■ This bit is reset when the SACSRC:TINTC bit in the clear register is set to "1".

Bit	Description
0	No timer interrupt request
1	Timer interrupt request issued

Notes:

- *Software reset (SCR:UPCL = 1) resets this bit to "0".*
- *This bit is not set to "1" when the synchronous transmission enable bit (TSYNE) is "1".*

[bit7] TINTE : Timer interrupt enable bit

This bit enables/disables timer interrupts to the CPU.

When this bit is "1" and the timer interrupt flag (TINT) is "1", a status interrupt request is issued.

- This bit is reset when the SACSRC:TINTEC bit in the clear register is set to "1".
- This bit is set when the SACSRS:TINTES bit in the set register is set to "1".

Bit	Description
0	Disable interrupts triggered by the serial timer.
1	Enable interrupts triggered by the serial timer.

[bit6] TSYNE : Synchronous transmission enable bit

This bit enables or disables synchronous transmission.

Transmission is activated if this bit is set to "1", as well as in any of the cases described below.

- The serial timer register (STMR) and the serial timer comparison register (STMCR) values coincide at the time of timer-synchronized transmission.
- An external trigger edge specified by the trigger selection bits (SACSR:TRG1, TRG0) is detected at the time of external-trigger-synchronized transmission.
- This bit is reset when the SACSRC:TSYNEC bit in the clear register is set to "1".
- This bit is set when the SACSRS:TSYNES bit in the set register is set to "1".

Bit	Description
0	Disable synchronous transmission. The serial timer is used as a timer.
1	Enable synchronous transmission. The serial timer is not used as a timer.

Notes:

- *This bit can be changed only when the serial timer enable bit (TMRE) is set to "0".*
- *When synchronous transmission is enabled (TSYNE = 1) and transmission is disabled (SCR:TXE = 0), transmission is not activated, even in any of the cases below.*
  - The serial timer register (STMR) and the serial timer comparison register (STMCR) values coincide at the time of timer-synchronized transmission.
  - An external trigger edge specified by the trigger selection bits (SACSR:TRG1, TRG0) is detected at the time of external-trigger-synchronized transmission.
- *In slave mode (SCR:MS = 1), this bit is internally fixed to "0".*

[bit5] TRGE: External trigger enable bit

This bit selects the method used to activate the serial timer.

- This bit is reset when the SACSRC:TRGEC bit in the clear register is set to "1".
- This bit is set when the SACSRS:TRGES bit in the set register is set to "1".

Bit	Description
0	The serial timer starts when the setting of the serial timer enable bit (TMRE) is changed from "0" to "1".
1	If the serial timer enable bit (TMRE) is set to "1", the detection of an external trigger edge specified by the trigger selection bits (TRG1, TRG0) starts the serial timer.

Notes:

- *This bit can be changed only when the serial timer enable bit (TMRE) is set to "0".*
- *If the serial timer enable bit (TMRE) is set to "0", the detection of an external trigger edge specified by the trigger selection bits (TRG1,TRG 0) does not start the serial timer.*
- *When synchronous transmission is enabled (SACSR:TSYNE = 1), external triggers are enabled (SACSR:TRGE = 1), the timer comparison register (STMCR) is set to 0x0000, and the serial timer is enabled (SACSR:TMRE = 1), transmission is activated in synchronization with an external trigger.*

[bit4:1] TDIV[3:0]: Timer operation clock division bits

These bits set the division ratio for the serial timer.

TDIV3	TDIV2	TDIV1	TDIV0	Timer Operation Clock						
				Division Ratio	Φ= 8 MHz	Φ= 10 MHz	Φ= 16 MHz	Φ= 20 MHz	Φ= 24 MHz	Φ= 32 MHz
0	0	0	0	Φ	125 ns	100 ns	62.5 ns	50 ns	41.67 ns	31.25 ns
0	0	0	1	Φ/2	250 ns	200 ns	125 ns	100 ns	83.33 ns	62.5 ns
0	0	1	0	Φ/4	500 ns	400 ns	250 ns	200 ns	166.67 ns	125 ns
0	0	1	1	Φ/8	1 us	800 ns	500 ns	400 ns	333.33 ns	250 ns
0	1	0	0	Φ/16	2 us	1.6 us	1 us	800 ns	666.67 ns	500 ns
0	1	0	1	Φ/32	4 us	3.2 us	2 us	1.6 us	1.33 us	1 us
0	1	1	0	Φ/64	8 us	6.4 us	4 us	3.2 us	2.67 us	2 us
0	1	1	1	Φ/128	16 us	12.8 us	8 us	6.4 us	5.33 us	4 us
1	0	0	0	Φ/256	32 us	25.6 us	16 us	12.8 us	10.67 us	8 us

Φ: Bus clock

Notes:

- *These bits can be changed only when the serial timer enable bit (TMRE) is set to "0".*
- *Settings other than the above are prohibited.*

[bit0] TMRE: Serial timer enable bit

This bit enables or disables serial timer operation.

- This bit is reset when the SACSRC:TMREC bit in the clear register is set to "1".
- This bit is set when the SACSRS:TMRES bit in the set register is set to "1".

Bit	Description
0	Stop the serial timer operation. When stopped, the value of the serial timer register (STMR) is retained.
1	Changing the setting of this bit from "0" to "1" initializes the serial timer register (STMR) to "0" and starts the serial timer operation.

Notes:

- *If the external trigger is enabled (TRGE = 1), setting this bit to "1" does not start the serial timer until detection of an external trigger edge specified by the trigger selection bits (SACSR:TRG1, TRG0).*
- *For serial timer synchronous transmission or external trigger transmission, change the setting of this bit from "0" to "1" if transmission is disabled.*

9.7. Serial Timer Register (STMR)

The serial timer register (STMR) represents the timer value of the serial timer.

Bit Configuration of the Serial Timer Register (STMR)

Figure 9-8 shows the bit configuration of the serial timer register (STMR).

Figure 9-8 Bit Configuration of Serial Timer Register (STMR)

Bit	15	14	13	12	11	10	9	8
Field	TM15	TM14	TM13	TM12	TM11	TM10	TM9	TM8
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	TM7	TM6	TM5	TM4	TM3	TM2	TM1	TM0
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

[bit15-0] TM[15:0]: Timer Data Bits

These bits indicate the timer value of the serial timer.

During timer operation, the timer value of the serial timer is incremented by 1 for each timer operation clock (specified by SACSR:TDIV3 to TDIV0).

Note:

- *These bits are initialized to "0" when the timer operation starts.*

9.8. Serial Timer Comparison Register (STMCR)

The serial timer comparison register (STMCR) sets the timer comparison value for the serial timer.

Bit Configuration of the Serial Timer Comparison Register (STMCR)

Figure 9-9 shows the bit configuration of the serial timer comparison register (STMCR).

Figure 9-9 Bit Configuration of Serial Timer Comparison Register (STMCR)

Bit	15	14	13	12	11	10	9	8
Field	TC15	TC14	TC13	TC12	TC11	TC10	TC9	TC8
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	TC7	TC6	TC5	TC4	TC3	TC2	TC1	TC0
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

[bit15:0] TC[15:0]: Compare bits

These bits set a comparison value for the serial timer.

These bits are compared with the serial timer register (STMR). If the serial timer register (STMR) coincides with these bits when the serial timer register is updated, the serial timer register is set to "0". At this time, if synchronous transmission is disabled (SACSR:TSYNE = 0), the timer interrupt flag (SACSR:TINT) is set to "1". If synchronous transmission is enabled (SACSR:TSYNE = 1), the transmission is activated.

The interval for the operations described below is (STMCR:TC + 1) * timer operation clock (specified by SACSR:TDIV3 to TDIV0).

- SACSR:TINT is set to "1".
- Serial-timer-synchronized transmission is activated.

Notes:

- *The timer interrupt flag (SACSR:TINT) is fixed to "1" when all of the following conditions are satisfied.*
  - Synchronous transmission is disabled (SACSR:TSYNE = 0).
  - 0x0000 is set in this register.
  - The timer is operating.
  - 0b0000 is set as the timer operating clock division value (SACSR:TDIV3-0).
- *When all of the following conditions are satisfied, transmission starts immediately upon the detection of an external trigger edge specified by the trigger selection bits (SACSR:TRG1, TRG0).*
  - Synchronous transmission is enabled (SACSR:TSYNE = 1).

- External trigger is enabled (SACSR:TRGE = 1).
- 0x0000 is set in this register.
- Transmission is enabled (SCR:TXE = 1), and transmission data exists (SSR:TDRE = 0).

- *This register can be changed only when the serial timer is disabled (SACSR:TMRE = 0).*

9.9. Serial Chip Select Control Status Register (SCSCR)

The serial chip select control register (SCSCR) is used to select the serial chip select start and end pins, display a serial chip select output pin, retain the active level of serial chip select, invert serial chip select, and enable/disable the output of the serial chip select pins.

Bit Configuration of the Serial Chip Select Control Status Register (SCSCR)

Figure 9-10 shows the bit configuration of the serial chip select control status register (SCSCR).

Figure 9-10 Bit Configuration of Serial Chip Select Control Status Register (SCSCR)

Bit	15	14	13	12	11	10	9	8
Field	SST1	SST0	SED1	SED0	SCD1	SCD0	SCAM	CDIV2
R/W Attribute	R/W	R/W	R/W	R/W	R,WX	R,WX	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	CDIV1	CDIV0	CSLVL	CSEN3	CSEN2	CSEN1	CSEN0	CSOE
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	1	0	0	0	0	0

[bit15:14] SST[1:0]: Serial chip select start bits

These bits select the pin at which serial chip select is to start.

If transmission is changed from the disabled state (SCR:TXE = 0) to the enabled state (SCR:TXE = 1), and transmission data is written to TDR, the serial chip select pins become active sequentially, starting with the one set with these bits.

Bits		Start Pin
0	0	SCS0
0	1	SCS1
1	0	SCS2
1	1	SCS3

Notes:

- *These bits can be changed only when transmission and reception are disabled (SCR:TXE = RXE = 0).*
- *If the serial chip select start bits (SST1, SST0) and the serial chip select end bits (SED1, SED0) are set to the same values, only the set serial chip select pin becomes active.*
- *In slave mode (SCR:MS = 1), the settings of these bits are invalid.*
- *Only those serial chip select pins for which serial chip select is enabled (SCSCR:CSEN3, CSEN2, CSEN1, CSEN0=1) become active.*

- *When using serial chip select in master mode (SCR:MS = 0), enable serial chip select (SCSCR:CSEN3, CSEN2, CSEN1, CSEN0="1") for the serial chip select pin that is set with these bits.*

[bit13:12] SED[1:0]: Serial chip select end bits

These bits select the pin at which serial chip select is to end.

If the serial chip select pins become active, up to the serial chip select pin that is set with these bits, the next serial chip select pin to become active is that specified with the serial chip select start bits (SST1, SST0).

Bits		End Pin
0	0	SCS0
0	1	SCS1
1	0	SCS2
1	1	SCS3

Notes:

- *These bits can be changed only when transmission and reception are disabled (SCR:TXE = RXE = 0).*
- *If the serial chip select start bits (SST1, SST0) and the serial chip select end bits (SED1, SED0) are set to the same values, only the set serial chip select pin becomes active.*
- *Only those serial chip select pins for which serial chip select is enabled (SCSCR:CSEN3, CSEN2, CSEN1, CSEN0="1") become active.*
- *In slave mode (SCR:MS = 1), the settings of these bits are invalid.*
- *When serial chip select is used in master mode (SCR:MS = 0), enable serial chip select (SCSCR:CSEN3, CSEN2, CSEN1, CSEN0="1") for the serial chip select pin that is set with these bits.*

[bit11:10] SCD[1:0]: Serial chip select display bits

These bits display the active serial chip select pin.

Bits		Display Pin
0	0	SCS0
0	1	SCS1
1	0	SCS2
1	1	SCS3

Notes:

- *If the serial chip select pins are inactive, these bits display the next serial chip select pin to become active.*
- *These bits are set to 0b00 in slave mode (SCR:MS = 1), at a software reset (SCR:UPCL = 1), or when transmission is disabled (SCR:TXE = 0).*

[bit9] SCAM: Serial chip select Active retention bit

This bit selects whether to retain the active state of the serial chip select pin.

For details, see "5 Operation of Serial Chip Select," "Serial Chip Select Active Level Retention Operation (SCSCR:SCAM = 1) (Valid Only in Master Mode (SCR:MS = 0))."

Bit	Description
0	Do not retain the active state of the serial chip select pin.
1	Retain the active state of the serial chip select pin.

Notes:

- *If transmission is disabled (SCR:TXE = 0) and a software reset is performed (SCR:UPCL = 1), the serial chip select pin will be inactive regardless of the value of this bit.*
- *When a serial chip select error occurs (SACSR:CSE = 1), the serial chip select pin will be inactive regardless of the value of this bit.*

[bit8:6] CDIV[2:0]: Serial chip select timing operating clock division bits

These bits set the division ratio of the serial chip select timing operating clock.

Bits			Serial Chip Select Timing Operating Clock						
			Division Ratio	Φ= 8 MHz	Φ= 10 MHz	Φ= 16 MHz	Φ= 20 MHz	Φ= 24 MHz	Φ= 32 MHz
0	0	0	Φ	125 ns	100 ns	62.5 ns	50 ns	41.67 ns	31.25 ns
0	0	1	Φ/2	250 ns	200 ns	125 ns	100 ns	83.33 ns	62.5 ns
0	1	0	Φ/4	500 ns	400 ns	250 ns	200 ns	166.67 ns	125 ns
0	1	1	Φ/8	1 us	800 ns	500 ns	400 ns	333.33 ns	250 ns
1	0	0	Φ/16	2 us	1.6 us	1 us	800 ns	666.67 ns	500 ns
1	0	1	Φ/32	4 us	3.2 us	2 us	1.6 us	1.33 us	1 us
1	1	0	Φ/64	8 us	6.4 us	4 us	3.2 us	2.67 us	2 us

Φ: Bus clock

Notes:

- *These bits can be changed only when transmission and reception are disabled (SCR:TXE = RXE = 0).*
- *In slave mode (SCR:MS = 1), the settings of these bits are invalid.*
- *Settings other than those above are prohibited.*

[bit5] CSLVL: Serial chip select level set bit

This bit switches the inactive-time level of the serial chip select pin between "H" and "L".

This bit is used for chip select pin 0.

Bit	Description
0	Set the inactive level to "L".
1	Set the inactive level to "H".

Notes:

- *This bit can be changed only when transmission and reception are disabled (SCR:TXE = RXE = 0).*
- *This bit setting is used in any of the following:*
  - In slave mode (SCR:MS = 1).
  - The chip select data format is disabled (ESCR:CSFE = 0).
  - The chip select data format is enabled (ESCR:CSFE = 1) and serial chip select pin 0 is active.

[bit4:1] CSEN3, CSEN2, CSEN1, CSEN0: Serial chip select enable bits

These bits enable or disable each serial chip select pin.

The CSEN3 bit corresponds to the SCS3 pin, the CSEN2 bit to the SCS2 pin, the CSEN1 bit to the SCS1 pin, and the CSEN0 bit to the SCS0 pin.

In slave mode (SCR:MS = 1), only the CSEN0 bit is used to enable or disable serial chip pins.

Bit	Description
0	Disable the operation of the serial chip select pin.
1	Enable the operation of the serial chip select pin.

Notes:

- *These bits can be changed only when transmission and reception are disabled (SCR:TXE = RXE = 0).*
- *In master mode (SCR:MS = 0), if CSEN3 to CSEN0 are set to 0b0000, transmission/reception operations are performed regardless of the status of the serial chip select pin.*
- *In slave mode (SCR:MS = 1), if CSEN0 is set to "0", transmission/reception operations are performed regardless of the status of the serial chip select pin.*
- *Disable any serial chip select pins that are not used.*

[bit0] CSOE: Serial chip select output enable bit

This bit enables or disables the output of serial chip select pins.

Bit	Description
0	Disable the output of all serial chip select pins.
1	Enable the output of all serial chip select pins.

Notes:

- *This bit can be changed only when transmission and reception are disabled (SCR:TXE = RXE = 0).*
- *In slave mode (SCR:MS = 1), set this bit to "0".*

9.10. Serial Chip Select Timing Registers (SCSTR3 to SCSTR0)

The serial chip select timing registers (SCSTR3 to SCSTR0) are used to set the serial chip select setup delay time, serial chip select hold delay time, and serial chip select deselect time.

Bit Configuration of the Serial Chip Select Timing Registers (SCSTR3 to SCSTR0)

Figure 9-11 and

Figure 9-12 show the bit configuration of the serial chip select timing registers (SCSTR3 to SCSTR0).

Figure 9-11 Bit Configuration of Serial Chip Select Timing Registers (SCSTR1, SCSTR0)

Bit	15	14	13	12	11	10	9	8
Field	CSSU7	CSSU6	CSSU5	CSSU4	CSSU3	CSSU2	CSSU1	CSSU0
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	CSHD7	CSHD6	CSHD5	CSHD4	CSHD3	CSHD2	CSHD1	CSHD0
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

[bit15:8] CSSU[7:0]: Serial chip select setup delay bits

These bits set the time from the instant that the serial chip select pin becomes active until the serial clock is output. If 0x00 is set in these bits, the serial clock is output at the same time as the serial chip select pin becomes active.

Bits								Setup Delay Time
0	0	0	0	0	0	0	0	The output of the serial clock is started at the same time as the serial chip select pin becomes active.
0	0	0	0	0	0	0	1	1x Serial chip select timing operating clock
0	0	0	0	0	0	1	0	2x Serial chip select timing operating clock
-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-
1	1	1	1	1	1	1	0	254x Serial chip select timing operating clock
1	1	1	1	1	1	1	1	255x Serial chip select timing operating clock

Notes:

- *These bits can be changed only when transmission and reception are disabled (SCR:TXE = RXE = 0).*
- *In slave mode (SCR:MS = 1), the settings of these bits are invalid.*

[bit7:0] CSHD[7:0]: Serial chip select hold delay bits

These bits set the time from the instant that the output of the serial clock ends until the serial chip select pin becomes inactive.

If these bits are set to 0x00, the output of the serial clock ends at the same time as the serial chip select pin becomes inactive.

Bits								Hold Delay Time
0	0	0	0	0	0	0	0	The output of the serial clock ends at the same time as the serial chip select pin becomes inactive.
0	0	0	0	0	0	0	1	1x Serial chip select timing operating clock
0	0	0	0	0	0	1	0	2x Serial chip select timing operating clock
-	-	-	-	-	-	-	-	-
1	1	1	1	1	1	1	0	254x Serial chip select timing operating clock
1	1	1	1	1	1	1	1	255x Serial chip select timing operating clock

Notes:

- *These bits can be changed only when transmission and reception are disabled (SCR:TXE = RXE = 0).*
- *In slave mode (SCR:MS = 1), the settings of these bits are invalid.*

Figure 9-12 Bit Configuration of Serial Chip Select Timing Registers (SCSTR3, SCSTR2)

Bit	15	14	13	12	11	10	9	8
Field	CSDS15	CSDS14	CSDS13	CSDS12	CSDS11	CSDS10	CSDS9	CSDS8
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	CSDS7	CSDS6	CSDS5	CSDS4	CSDS3	CSDS2	CSDS1	CSDS0
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

[bit15:0] CSDS[15:0]: Serial chip deselect bits

These bits set the minimum time from the instant the serial chip select pin becomes inactive until the next time that the serial chip select pin becomes active.

CSDS15	CSDS14	CSDS13	...	CSDS2	CSDS1	CSDS0	Minimum Deselect Time
0	0	0	...	0	0	0	No minimum deselect time specified (5 bus clock time)
0	0	0	...	0	0	1	1x Serial chip select timing operating clock
0	0	0	...	0	1	0	2x Serial chip select timing operating clock
- -	- -	- -	- -	- -	- -	- -	- -
1	1	1	...	1	1	0	65534x Serial chip select timing operating clock
1	1	1	...	1	1	1	65535x Serial chip select timing operating clock

Notes:

- *These bits can be changed only when transmission and reception are disabled (SCR:TXE = RXE = 0).*
- *In slave mode (SCR:MS = 1), the settings of these bits are invalid.*
- *Regardless of the deselect time setting, a minimum time of at least 5 bus clocks is required from the time the serial chip select pin becomes inactive until the next time it becomes active.*
- *Do not make the settings: SCSTR3-2:CSDS = 0x0001 and SCSCR:CDIV2-0 = "000".*

9.11. Serial Chip Select Format Registers (SCSFR2 to SCSFR0)

The serial chip select format registers (SCSFR2 to SCSFR0) are used to select the chip select active level of each serial chip select, invert the serial clock, make the settings for connecting to SPI, and set the data direction and data length of serial data output.

Bit Configuration of the Serial Chip Select Format Registers (SCSFR2 to SCSFR0)

Figure 9-13 and

Figure 9-14 show the bit configuration of the serial chip select format registers (SCSFR2 to SCSFR0).

Figure 9-13 Bit Configuration of Serial Chip Select Format Registers 1/0 (SCSFR1, SCSFR0)

Bit	15	14	13	12	11	10	9	8
Field	CS2 CSLVL	CS2 SCINV	CS2 SPI	CS2 BDS	CS2 L3	CS2 L2	CS2 L1	CS2 L0
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	1	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	CS1 CSLVL	CS1 SCINV	CS1 SPI	CS1 BDS	CS1 L3	CS1 L2	CS1 L1	CS1 L0
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	1	0	0	0	0	0	0	0

[bit15] CS2CSLVL: Bit for Setting the Serial Chip Select Level of Chip Select 2

When the chip select data format is enabled (ESCR:CSFE = 1), this bit selects the inactive-time level of serial chip select pin 2.

Bit	Description
0	Set the inactive level to "L".
1	Set the inactive level to "H".

Notes:

- *This bit can be changed only when transmission and reception are disabled (SCR:TXE = RXE = 0).*
- *In slave mode (SCR:MS = 1), the setting of this bit is invalid.*
- *When the chip select data format is disabled (ESCR:CSFE = 0), the setting of this bit is invalid.*

[bit14] CS2SCINV: Bit for Inverting the Serial Clock of Chip Select 2

When the chip select data format is enabled (ESCR:CSFE = 1), this bit sets the serial clock format to be assumed when serial chip select pin 2 is active.

0:

- Set the mark level of serial clock output to "H".
- In normal transfer, transmission data is output in synchronization with a falling edge of the serial clock. In SPI transfer, it is output in synchronization with a rising edge of the serial clock.
- In normal transfer, reception data is sampled at a rising edge of the serial clock. In SPI transfer, it is sampled at a falling edge of the serial clock.

1:

- Set the mark level of serial clock output to "L".
- In normal transfer, transmission data is output in synchronization with a rising edge of the serial clock. In SPI transfer, it is output in synchronization with a falling edge of the serial clock.
- In normal transfer, reception data is sampled at a falling edge of the serial clock. In SPI transfer, it is sampled at a rising edge of the serial clock.

Bit	Description
0	Mark level "H" format
1	Mark level "L" format

Notes:

- *This bit can be changed only when transmission and reception are disabled (SCR:TXE = RXE = 0).*
- *In slave mode (SCR:MS = 1), the setting of this bit is invalid.*
- *When the chip select data format is disabled (ESCR:CSFE = 0), the setting of this bit is invalid.*

[bit13] CS2SPI: Bit for Making Serial Chip Select Pin 2 Support SPI

When the chip select data format is enabled (ESCR:CSFE = 1), this bit is used for performing communication supporting SPI when serial chip select pin 2 is active.

- 0: Perform normal synchronous communication.
- 1: Support SPI.

Bit	Description
0	Normal synchronous transfer
1	SPI support

Notes:

- *This bit can be changed only when transmission and reception are disabled (SCR:TXE = RXE = 0).*
- *In slave mode (SCR:MS = 1), the setting of this bit is invalid.*
- *When the chip select data format is disabled (ESCR:CSFE = 0), the setting of this bit is invalid.*

[bit12] CS2BDS: Bit for Selecting the Transfer Direction of Chip Select Pin 2

When the chip select data format is enabled (ESCR:CSFE = 1), this bit selects the transfer of transfer serial data when serial chip select pin 2 is active, setting either transfer of the lowest bit first (LSB first, SCSFR1:CS2BDS = 0) or transfer of the highest bit first (MSB first, SCSFR1:CS2BDS = 1).

Bit	Description
0	LSB first (lowest bit transferred first)
1	MSB first (highest bit transferred first)

Notes:

- *This bit can be changed only when transmission and reception are disabled (SCR:TXE = RXE = 0).*
- *In slave mode (SCR:MS = 1), the setting of this bit is invalid.*
- *When the chip select data format is disabled (ESCR:CSFE = 0), the setting of this bit is invalid.*

[bit11:8] CS2L[3:0]: Bits for Selecting the Data Length of Serial Chip Select Pin 2

When the chip select data format is enabled (ESCR:CSFE = 1), these bits specify the data length of the transmission/reception data to be assumed when serial chip select pin 2 is active.

Bits				Data Length
0	0	0	0	8-bit length
0	0	0	1	5-bit length
0	0	1	0	6-bit length
0	0	1	1	7-bit length
0	1	0	0	9-bit length
0	1	0	1	10-bit length
0	1	1	0	11-bit length
0	1	1	1	12-bit length
1	0	0	0	13-bit length
1	0	0	1	14-bit length
1	0	1	0	15-bit length
1	0	1	1	16-bit length
1	1	0	0	20-bit length
1	1	0	1	24-bit length
1	1	1	0	32-bit length

Notes:

- *Settings other than those above are prohibited.*
- *These bits can be changed only when transmission and reception are disabled (SCR:TXE = RXE = 0).*
- *In slave mode (SCR:MS = 1), the settings of these bits are invalid.*
- *When the chip select data format is disabled (ESCR:CSFE = 0), the settings of these bits are invalid.*

[bit7] CS1CSLVL: Bit for Setting the Serial Chip Select Level of Chip Select 1

When the chip select data format is enabled (ESCR:CSFE = 1), this bit selects the inactive-time level of serial chip select pin 1.

Bit	Description
0	Set the inactive level to "L".
1	Set the inactive level to "H".

Notes:

- *This bit can be changed only when transmission and reception are disabled (SCR:TXE = RXE = 0).*
- *In slave mode (SCR:MS = 1), the setting of this bit is invalid.*
- *When the chip select data format is disabled (ESCR:CSFE = 0), the setting of this bit is invalid.*

[bit6] CS1SCINV: Bit for Inverting the Serial Clock of Chip Select 1

When the chip select data format is enabled (ESCR:CSFE = 1), this bit sets the serial clock format to be assumed when serial chip select pin 1 is active.

0:

- Set the mark level of serial clock output to "H".
- In normal transfer, transmission data is output in synchronization with a falling edge of the serial clock. In SPI transfer, it is output in synchronization with a rising edge of the serial clock.
- In normal transfer, reception data is sampled at a rising edge of the serial clock. In SPI transfer, it is sampled at a falling edge of the serial clock.

1:

- Set the mark level of serial clock output to "L".
- In normal transfer, transmission data is output in synchronization with a rising edge of the serial clock. In SPI transfer, it is output in synchronization with a falling edge of the serial clock.
- In normal transfer, reception data is sampled at a falling edge of the serial clock. In SPI transfer, it is sampled at a rising edge of the serial clock.

Bit	Description
0	Mark level "H" format
1	Mark level "L" format

Notes:

- *This bit can be changed only when transmission and reception are disabled (SCR:TXE = RXE = 0).*
- *In slave mode (SCR:MS = 1), the setting of this bit is invalid.*
- *When the chip select data format is disabled (ESCR:CSFE = 0), the setting of this bit is invalid.*

[bit5] CS1SPI: Bit for Making Serial Chip Select Pin 1 Support SPI

When the chip select data format is enabled (ESCR:CSFE = 1), this bit is used for performing communication supporting SPI when serial chip select pin 1 is active.

- 0: Perform normal synchronous communication.
- 1: Support SPI.

Bit	Description
0	Normal synchronous transfer
1	SPI support

Notes:

- *This bit can be changed only when transmission and reception are disabled (SCR:TXE = RXE = 0).*
- *In slave mode (SCR:MS = 1), the setting of this bit is invalid.*
- *When the chip select data format is disabled (ESCR:CSFE = 0), the setting of this bit is invalid.*

[bit4] CS1BDS: Bit for Selecting the Transfer Direction of Chip Select Pin 1

When the chip select data format is enabled (ESCR:CSFE = 1), this bit selects the transfer of transfer serial data when serial chip select pin 1 is active, setting either transfer of the lowest bit first (LSB first, SCSFR0:CS1BDS = 0) or transfer of the highest bit first (MSB first, SCSFR0:CS1BDS = 1).

Bit	Description
0	LSB first (lowest bit transferred first)
1	MSB first (highest bit transferred first)

Notes:

- *This bit can be changed only when transmission and reception are disabled (SCR:TXE = RXE = 0).*
- *In slave mode (SCR:MS = 1), the setting of this bit is invalid.*
- *When the chip select data format is disabled (ESCR:CSFE = 0), the setting of this bit is invalid.*

[bit3:0] CS1L[3:0]: Bits for Selecting the Data Length of Serial Chip Select Pin 1

When the chip select data format is enabled (ESCR:CSFE = 1), these bits specify the data length of the transmission/reception data to be assumed when serial chip select pin 1 is active.

Bits				Data Length
0	0	0	0	8-bit length
0	0	0	1	5-bit length
0	0	1	0	6-bit length
0	0	1	1	7-bit length
0	1	0	0	9-bit length
0	1	0	1	10-bit length
0	1	1	0	11-bit length
0	1	1	1	12-bit length
1	0	0	0	13-bit length
1	0	0	1	14-bit length
1	0	1	0	15-bit length
1	0	1	1	16-bit length
1	1	0	0	20-bit length
1	1	0	1	24-bit length
1	1	1	0	32-bit length

Notes:

- *Settings other than above are prohibited.*
- *These bits can be changed only when transmission and reception are disabled (SCR:TXE = RXE = 0).*
- *In slave mode (SCR:MS = 1), the settings of these bits are invalid.*
- *When the chip select data format is disabled (ESCR:CSFE = 0), the settings of these bits are invalid.*

Figure 9-14 Bit Configuration of Serial Chip Select Format Register 2 (SCSFR2)

Bit	15 ... 8	7	6	5	4	3	2	1	0
Field	-	CS3 CSLVL	CS3 SCINV	CS3 SPI	CS3 BDS	CS3 L3	CS3 L2	CS3 L1	CS3 L0
R/W Attribute		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute		-	-	-	-	-	-	-	-
Initial Value		1	0	0	0	0	0	0	0

[bit7] CS3CSLVL: Bit for Setting the Serial Chip Select Level of Chip Select 3

When the chip select data format is enabled (ESCR:CSFE = 1), this bit selects the inactive-time level of serial chip select pin 3.

Bit	Description
0	Set the inactive level to "L".
1	Set the inactive level to "H".

Notes:

- *This bit can be changed only when transmission and reception are disabled (SCR:TXE = RXE = 0).*
- *In slave mode (SCR:MS = 1), the setting of this bit is invalid.*
- *When the chip select data format is disabled (ESCR:CSFE = 0), the setting of this bit is invalid.*

[bit6] CS3SCINV: Bit for Inverting the Serial Clock of Chip Select 3

When the chip select data format is enabled (ESCR:CSFE = 1), this bit sets the serial clock format to be assumed when serial chip select pin 3 is active.

0:

- Set the mark level of the serial clock output to "H".
- In normal transfer, transmission data is output in synchronization with a falling edge of the serial clock. In SPI transfer, it is output in synchronization with a rising edge of the serial clock.
- In normal transfer, reception data is sampled at a rising edge of the serial clock. In SPI transfer, it is sampled at a falling edge of the serial clock.

1:

- Set the mark level of the serial clock output to "L".
- In normal transfer, transmission data is output in synchronization with a rising edge of the serial clock. In SPI transfer, it is output in synchronization with a falling edge of the serial clock.
- In normal transfer, reception data is sampled at a falling edge of the serial clock. In SPI transfer, it is sampled at a rising edge of the serial clock.

Bit	Description
0	Mark level "H" format
1	Mark level "L" format

Notes:

- *This bit can be changed only when transmission and reception are disabled (SCR:TXE = RXE = 0).*
- *In slave mode (SCR:MS = 1), the setting of this bit is invalid.*
- *When the chip select data format is disabled (ESCR:CSFE = 0), the setting of this bit is invalid.*

[bit5] CS3SPI: Bit for Making Serial Chip Select Pin 3 Support SPI

When the chip select data format is enabled (ESCR:CSFE = 1), this bit is used for performing communication supporting SPI when serial chip select pin 3 is active.

- 0: Perform normal synchronous communication.
- 1: Support SPI.

Bit	Description
0	Normal synchronous transfer
1	SPI support

Notes:

- *This bit can be changed only when transmission and reception are disabled (SCR:TXE = RXE = 0).*
- *In slave mode (SCR:MS = 1), the setting of this bit is invalid.*
- *When the chip select data format is disabled (ESCR:CSFE = 0), the setting of this bit is invalid.*

[bit4] CS3BDS: Bit for Selecting the Transfer Direction of Chip Select Pin 3

When the chip select data format is enabled (ESCR:CSFE = 1), this bit selects the transfer of transfer serial data when serial chip select pin 3 is active, setting either transfer of the lowest bit first (LSB first, SCSFR2.CS3BDS = 0) or transfer of the highest bit first (MSB first, SCSFR2:CS3BDS = 1).

Bit	Description
0	LSB first (lowest bit transferred first)
1	MSB first (highest bit transferred first)

Notes:

- *This bit can be changed only when transmission and reception are disabled (SCR:TXE = RXE = 0).*
- *In slave mode (SCR:MS = 1), the setting of this bit is invalid.*
- *When the chip select data format is disabled (ESCR:CSFE = 0), the setting of this bit is invalid.*

[bit3:0] CS3L[3:0]: Bits for Selecting the Data Length of Serial Chip Select Pin 3

When the chip select data format is enabled (ESCR:CSFE = 1), these bits specify the data length of the transmission/reception data to be assumed when serial chip select pin 3 is active.

Bits				Data Length
0	0	0	0	8-bit length
0	0	0	1	5-bit length
0	0	1	0	6-bit length
0	0	1	1	7-bit length
0	1	0	0	9-bit length
0	1	0	1	10-bit length
0	1	1	0	11-bit length
0	1	1	1	12-bit length
1	0	0	0	13-bit length
1	0	0	1	14-bit length
1	0	1	0	15-bit length
1	0	1	1	16-bit length
1	1	0	0	20-bit length
1	1	0	1	24-bit length
1	1	1	0	32-bit length

Notes:

- *Settings other than above are prohibited.*
- *These bits can be changed only when transmission and reception are disabled (SCR:TXE=RXE = 0).*
- *In slave mode (SCR:MS = 1), the settings of these bits are invalid.*
- *When the chip select data format is disabled (ESCR:CSFE = 0), the settings of these bits are invalid.*

9.12. Transfer Byte Registers (TBYTE3 to TBYTE0)

The transfer byte registers (TBYTE3 to TBYTE0) are used to set the number of transfer data items when the respective serial chip select pins are active.

Bit Configuration of the Transfer Bytes (TBYTE3 to TBYTE0)

Figure 9-15 shows the bit configuration of the transfer byte register (TBYTE).

Figure 9-15 Bit Configuration of Transfer Byte Register (TBYTE)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	TBYTE1								TBYTE0							
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	TBYTE3								TBYTE2							
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The transfer byte registers can set the number of transfer data items when the respective serial chip select pins are active. After a serial chip select pin becomes active, if the transfer of data items for which a number is set in the corresponding one of these bits is completed, the serial chip select pin becomes inactive.

Serial chip select pin 0 (SCS0) corresponds to TBYTE0, serial chip select pin 1 (SCS1) to TBYTE1, serial chip select pin 2 (SCS2) to TBYTE2, and serial chip select pin 3 (SCS3) to TBYTE3.

When serial chip select is disabled (SCSCR:CSEN3-0 = 0b0000), transfer byte register 0 (TBYTE0) is used for transmission synchronized with the timer or transmission synchronized with an external trigger. After the start of transmission due to transmission being synchronized with the timer, or transmission synchronized with an external trigger, data items for which a number is set in TBYTE0 are transferred.

If the value of one of these bits is changed during transmission (SSR:TBI = 0), the new number of transfer data items becomes valid after the end of transmission with the old number of transfer data items.

[bit15] TBYTE: Transfer Byte

TBYTE	Description
Write	Writing to TBYTE
Read	Setting of TBYTE

Notes:

- *If one of these bits is set to 0x00, the transfer count will be 8.*
- *In slave mode (SCR:MS = 1), the setting of this bit is invalid.*

9.13. Baud Rate Generator Registers 1, 0 (BGR1, BGR0)

The baud rate generator registers 1, 0 (BGR1 and BGR0) are used to set the division ratio of the serial clock.

Bit Configuration of Baud Rate Generator Registers 1, 0 (BGR1, BGR0)

Figure 9-16 shows the bit configuration of baud rate generator registers 1, 0 (BGR1, BGR0).

Figure 9-16 Bit Configuration of Baud Rate Generator Registers 1, 0 (BGR1, BGR0)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved	BGR1							BGR0							
R/W Attribute	R0,W0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

- These registers are set the value of the division ratio of the serial clock.
- BGR1 represents the upper bits while BGR0 represents the lower bits. The reload value for counting can be written to these registers, and the setting values can be read from these registers.
- The reload counter starts counting when a reload value is written to baud rate generator registers 1, 0 (BGR1, BGR0).

[bit15] Reserved: Reserved Bit

[bit14:8] BGR1: Baud Rate Generator Register 1

BGR1	Description
Write	Write to reload counter bits 8 to 14.
Read	Read the setting value of BGR1.

[bit7:0] BGR0: Baud Rate Generator Register 0

BGR0	Description
Write	Write to reload counter bits 0 to 7.
Read	Read the setting value of BGR0.

Notes:

- *Use 16-bit access for writing a value to baud rate generator register (BGR1, BGR0).*
- *If the reload value is even-numbered, the "H" width and the "L" width of the serial clock will be as described below, depending on the setting of the SCINV bit. If it is odd-numbered, the "H" width and the "L" width of the serial clock will be the same.*
  When SMR:SCINV = 0, the "H" width of the serial clock will be longer by 1 cycle of the bus clock.
  When SMR:SCINV = 1, the "L" width of the serial clock will be longer by 1 cycle of the bus clock.
- *Set the reload value to 2 or more in master mode and to 3 or more in slave mode.*
- *If the set values of the baud rate generator registers (BGR1, BGR0) are changed, the new values are reloaded when the counter value becomes 0x00000. Thus, to have the new settings take effect immediately, execute a CSIO reset (SCR:UPCL) after changing the settings of BGR1/0.*
- *When the reception FIFO is used, set the reception FIFO idle detection enable bit (FCR1:FRIIE) to "1", and if performing operation in slave mode, set baud rates in BGR1/0.*

9.14. FIFO Control Register 1 (FCR1)

FIFO control register 1 (FCR1) sets the FIFO test, selects the transmission and reception FIFOs, enables the transmission FIFO interrupts, and controls the interrupt flag.

Bit Configuration of FIFO Control Register 1 (FCR1)

Figure 9-17 shows the bit configuration of FIFO control register 1 (FCR1).

Figure 9-17 Bit Configuration of FIFO Control Register 1 (FCR1)

Bit	15	14	13	12	11	10	9	8	7 ... 0
Field	Reserved	Reserved	Reserved	FLSTE	FRIIE	FDRQ	FTIE	FSEL	(FCR0)
R/W Attribute	R0,W0	R0,W0	R0,W0	R/W	R/W	R,WX	R/W	R/W	
Protection Attribute	-	-	-	-	-	-	-	-	
Initial Value	0	0	0	0	0	1	0	0	

[bit15:13] Reserved: Reserved Bits

[bit12] FLSTE: Retransmission Data Lost Detection Enable Bit

This bit enables FIFO Retransmission Data Lost Flag Bit (FCR0:FLST) detection.

- This bit is reset when the FCR1C:FLSTEC bit in the clear register is set to "1".
- This bit is set when the FCR1S:FLSTES bit in the set register is set to "1".

"0": Disable FLST bit detection.

"1": Enable FLST bit detection.

Bit	Description
0	Disable data lost detection.
1	Enable data lost detection.

Note:

- *To set this bit to "1", first set the FSET bit to "1" and then set this bit to "1".*

[bit11] FRIIE: Reception FIFO Idle Detection Enable Bit

When the reception FIFO contains valid data, this bit enables or disables the detection of the continuation of reception idle status for the 8-bit time or longer. If the reception interrupt is enabled (SCR:RIE = 1), detection of the reception idle status triggers a reception interrupt.

- This bit is reset when the FCR1C:FRIIEC bit in the clear register is set to "1".
- This bit is set when the FCR1S:FRIIES bit in the set register is set to "1".

"0": Disable the detection of reception idle status.

"1": Enable the detection of reception idle status.

Bit	Description
0	Disable the detection of reception FIFO idle.
1	Enable the detection of reception FIFO idle.

Note:

- *To use the reception FIFO, set this bit to "1".*

[bit10] FDRQ: Transmission FIFO Data Request Bit

This bit requests transmission FIFO data.

Value "1" of this bit indicates that transmission data is requested. At this time, if transmission FIFO interrupts are enabled (FTIE = 1), a transmission FIFO interrupt request is output.

ECR.TXBLKEN=0

	FTICR=0	FTICR>0
Setting FDRQ	If transmission FIFO is empty	FTICR ≧ the number of valid data of transmitting FIFO
Clearing FDRQ	If transmission FIFO is full	If transmitting FIFO is full.

ECR.TXBLKEN=1

	FTICR=0	FTICR>0
Setting FDRQ	If transmission FIFO is empty	FTICR ≦ empty data of transmitting FIFO
Clearing FDRQ	If transmission FIFO is full	FTICR > empty data of transmitting FIFO

Regardless of the setting value of ECR.TXBLKEN, the FDRQ bit is cleared by writing "1" to the FCR1C.FDRQC bit.

Regardless of the setting value of ECR.TXBLKEN, the FDRQ bit is set by resetting the transmission FIFO.

Bit	Description
0	No transmission FIFO data request
1	Transmission FIFO data request issued

Notes:

- *The FSEL bit cannot be changed when this bit is "0".*
- *Writing to this bit is invalid.*
- *If a transmission interrupt is generated, and the necessary data is written to the transmission FIFO, write "1" to the FIFO transmission data request clear bit (FCR1C:FDRQC) to clear the interrupt request.*

[bit9] FTIE: Transmission FIFO Interrupt Enable Bit

This bit enables a transmission FIFO interrupt. If this bit is set to "1", an interrupt occurs when the FDRQ bit is "1".

- This bit is reset when the FCR1C:FTIEC bit in the clear register is set to "1".
- This bit is set when the FCR1S:FTIES bit in the set register is set to "1".

Bit	Description
0	Disable transmission FIFO interrupts.
1	Enable transmission FIFO interrupts.

Note:

- *Set FTIE=0 if FIFO is not used when block transfer.*

[bit8] FSEL: FIFO Selection Bit

This bit selects the transmission and reception FIFO.

- This bit is reset when the FCR1C:FSELC bit in the clear register is set to "1".
- This bit is set when the FCR1S:FSELS bit in the set register is set to "1".

"0": Allocate as the transmission FIFO: FIFO1, and as the reception FIFO: FIFO2.
"1": Allocate as the transmission FIFO: FIFO2, and as the reception FIFO: FIFO1.

Bit	Description
0	Transmission FIFO: FIFO1, reception FIFO: FIFO2
1	Transmission FIFO: FIFO2, reception FIFO: FIFO1

Notes:

- *This bit is not cleared by a FIFO reset (FCL2, FCL1 = 1).*
- *To change this bit, first disable FIFO operation (FCR0:FE2, FE1 = 0).*
- *When FDRQ = 0, the changing of this bit is prohibited.*
- *Set this bit before setting the FIFO Byte Register(FBYTE) and the Transmission FIFO Interrupt Control Register(FTICR).*
- *This bit can't be accessed with the FIFO Byte Register(FBYTE) at the same time.*

9.15. FIFO Control Register 0 (FCR0)

FIFO control register 0 (FCR0) enables/disables FIFO operation, resets the FIFO, saves the read pointer, and makes the retransmission setting.

Bit Configuration of FIFO Control Register 0 (FCR0)

Figure 9-18 shows the bit configuration of FIFO control register 0 (FCR0).

Figure 9-18 Bit Configuration of FIFO Control Register 0 (FCR0)

Bit	15 ... 8	7	6	5	4	3	2	1	0
Field	(FCR1)	Reserved	FLST	FLD	FSET	FCL2	FCL1	FE2	FE1
R/W Attribute		R0,W0	R,WX	R,W	R0,W	R0,W	R0,W	R/W	R/W
Protection Attribute		-	-	-	-	-	-	-	-
Initial Value		0	0	0	0	0	0	0	0

[bit7] Reserved: Reserved Bit

[bit6] FLST: FIFO Retransmission Data Lost Flag Bit

This bit indicates that the retransmission data of the transmission FIFO has been lost.

FLST set condition

- Data is written in the FIFO when the FLSTE bit in FIFO control register 1 (FCR1) is "1" and the write pointer of the transmission FIFO and the read pointer saved by the FSET bit coincide.

FLST reset conditions

- The transmission FIFO reset(When FCR1:FSEL=0, set "1" to the FCR0:FCL1. When FCR1:FSEL=1, set "1" to the FCR0:FCL2.)
- Writing "1" to the FSET bit

If "1" is set in this bit, the data pointed to by the read pointer saved with the FSET bit is overwritten. For this reason, re-transmission with the FLD bit cannot be set even if an error occurs. When performing retransmission with this bit set to "1", reset the FIFO and then write the data to the FIFO again.

Bit	Description
0	Data has not been lost.
1	Data has been lost.

[bit5] FLD: FIFO Pointer Reload Bit

This bit reloads, to the read pointer, the data saved in the transmission FIFO by the FSET bit. This bit is used for retransmission in cases such as communication errors.
This bit is set to "0" when the retransmission setting has been completed.

- This bit is set when the FCR0S:FLDS bit in the set register is set to "1".

Bit	Description
0	Do not execute reload.
1	Execute reload.

Notes:

- *While this bit is set to "1", reloading to the read pointer is in progress, so writing of other than a FIFO reset is prohibited.*
- *Setting this bit to "1" is prohibited when a FIFO is enabled or transmission is in progress.*
- *First set the SCR:TIE bit and the SCR:TBIE bit to "0" and then write "1" to this bit, and after the transmission FIFO is enabled, set the SCR:TIE and SCR:TBIE bits to "1".*

[bit4] FSET: FIFO Pointer Saving Bit

This bit stores the read pointer of the transmission FIFO.
If the read pointer is saved before transmission starts and then a communication error occurs, retransmission is possible if the FLST bit is "0".

■ This bit is set when the FCR0S:FSETS bit in the set register is set to "1".

"1": Save the current read pointer value.
"0": No effect.

Bit	Description	
	Write	Read
0	Do not save.	Always read "0".
1	Save the read pointer value.	

Note:

- *Set this bit to "1" when the transmission byte count (FBYTE) is 0.*

[bit3] FCL2: FIFO2 Reset Bit

This bit resets FIFO2.
If this bit is set to "1", the internal status of FIFO2 is initialized.
Only the FCR0:FLST bit is initialized and the values of other bits in the FCR1/0 registers are retained.

■ This bit is set when the FCR0S:FCL2S bit in the set register is set to "1".

Bit	Description	
	Write	Read
0	No effect	"0" is always read.
1	Reset FIFO2.	

Notes:

- *Disable transmission/reception first and then execute a FIFO2 reset.*
- *Execute the reset after setting the transmission FIFO interrupt enable bit to "0".*
- *The valid data count of the FBYTE2 register is set to 0.*
- *Neither TDR register nor the RDR register is initialized.*

[bit2] FCL1: FIFO1 Reset Bit

This bit resets FIFO1.
If this bit is set to "1", the internal status of FIFO1 is initialized.
Only the FCR0:FLST bit is initialized and the values of other bits in the FCR1/0 registers are retained.

- This bit is set when the FCR0S:FCL1S bit in the set register is set to "1".

Bit	Description	
	Write	Read
0	No effect	"0" is always read.
1	Reset FIFO1.	

Notes:

- *Disable transmission/reception first and then execute a FIFO1 reset.*
- *Execute this reset after setting the transmission FIFO interrupt enable bit to "0".*
- *The valid data count of the FBYTE1 register is set to 0.*
- *Neither TDR register nor the RDR register is initialized.*

[bit1] FE2: FIFO2 Operation Enable Bit

This bit enables or disables the operation of FIFO2.

- Set this bit to "1" when using FIFO2.
- When FIFO2 is set as the transmission FIFO (FCR1:FSEL = 1), FIFO2 contains data when "1" is written to this bit, and CSIO is transmission-enabled (SCR:TXE = 1), transmission is started immediately. At this time, to set the SCR:TIE and SCR:TBIE bits to "1", first set them to "0" and write "1" to this bit, and then set the SCR:TIE and SCR:TBIE bits to "1".
- A reception error clears this bit to "0" if the FIFO is selected as the reception FIFO by the FSEL bit. After this, this bit cannot be set to "1" unless the reception error is cleared.
- To use FIFO1 as the transmission or reception FIFO, set this bit to "1" or "0" when the transmission or reception buffer is empty ((SSR:TDRE = 1) or (SSR:RDRF = 0)), respectively.
- To use FIFO1 as a reception FIFO, first disable reception (SCR:RXE = 0) and then change the setting of this bit when the reception buffer is empty (SSR:RDRF = 0).
- Disabling FIFO2 does not change the state of FIFO2.
- This bit is reset when the FCR0C:FE2C bit in the clear register is set to "1".
- This bit is set when the FCR0S:FE2S bit in the set register is set to "1".

Bits	Description
0	Disable FIFO2 operation.
1	Enable FIFO2 operation.

[bit0] FE1: FIFO1 Operation Enable Bit

This bit enables or disables the operation of FIFO1.

- Set this bit to "1" when using FIFO1.
- When FIFO1 is set as the transmission FIFO (FCR1:FSEL = 0), FIFO1 contains data when "1" is written to this bit, and CSIO is transmission-enabled (SCR:TXE = 1), transmission is started immediately. At this time, to set the SCR:TIE and SCR:TBIE bits to "1", first set them to "0" and write "1" to this bit, and then set the TIE and TBIE bits to "1".

- A reception error clears this bit to "0" if the FIFO is selected as the reception FIFO by the FSEL bit. Subsequently, this bit cannot be set to "1" unless the reception error is cleared.
- To use FIFO1 as the transmission or reception FIFO, set this bit to "1" or "0" when the transmission or reception buffer is empty ((SSR:TDRE = 1) or (SSR:RDRF = 0)), respectively.
- To use FIFO1 as a reception FIFO, first disable reception (SCR:RXE = 0) and then change the setting of this bit when the reception buffer is empty (SSR:RDRF = 0).
- Disabling FIFO1 does not change the state of FIFO1.
- This bit is reset when the FCR0C:FE1C bit in the clear register is set to "1".
- This bit is set when bit FCR0S:FE1S in the set register is set to "1".

Bit	Description
0	Disable FIFO1 operation.
1	Enable FIFO1 operation.

9.16. FIFO Byte Register (FBYTE)

The FIFO byte register (FBYTE) indicates the valid data count of the FIFO. This register also specifies whether a reception interrupt is generated when the predefined amount of data is received by the reception FIFO.

Bit Configuration of the FIFO Byte Register (FBYTE)

Figure 9-19 shows the bit configuration of the FIFO byte register (FBYTE).

Figure 9-19 Bit Configuration of FIFO Byte Register (FBYTE)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	FBYTE2								FBYTE1							
R/W Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W
Protection Attribute	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The FBYTE register indicates the valid data count for the FIFO. The settings made with the FCR1:FSEL bit are listed below.

Table 9-3 Data Count Indication

FSEL	FIFO Selection	Data Count Indication
0	FIFO2: Reception FIFO, FIFO1: Transmission FIFO	FIFO2:FBYTE2, FIFO1:FBYTE1
1	FIFO2: Transmission FIFO, FIFO1: Reception FIFO	FIFO2:FBYTE2, FIFO1:FBYTE1

- The initial transfer count value of the FBYTE register is 0x08.
- The FBYTE for the reception FIFO contains the data count for setting the reception interrupt flag. If the transfer count that is set matches the data display of the FBYTE register, the interrupt flag (SSR:RDRF) is set to "1".
- When both of the following conditions are satisfied, the continuation of reception idle status for 8 baud rate clocks or longer sets the interrupt flag (RDRF) to "1".
  - The reception FIFO idle detection enable bit (FRIIE) is "1".
  - The number of data items in the reception FIFO does not reach the transfer count.

  During an 8-clock count, the counter is reset to 0 when RDR is read, and the system starts counting the 8 clocks again. The counter is reset to 0 when the reception FIFO is disabled. If the reception FIFO is enabled while there is data remaining in the reception FIFO, counting restarts.
- To receive data with the master operation (master reception), set the SCR:TIE bit and the SCR:TBIE bit to "0", set the number of reception data items in the FBYTE register of the transmission FIFO, and write "1" to the FCR1C:FDRQC bit. Later, when the SCR:TXE bit is "1", the serial clock pulses needed for the data items that have been set are output, and as many data items as set can be received. To set the SCR:TIE bit and the SCR:TBIE bit to "1", set them to "1" after FCR1:FDRQ is set to "1".
- Whenever transmission data is written to TDR, the FBYTE of the transmission FIFO is incremented by 1.
- Whenever reception data is read from RDR, the FBYTE of the reception FIFO is decremented by 1.
- During block transfer, the value of this register is the block transfer count.

[bit15:8] FBYTE2: FIFO2 Data Count Indication Bit
[bit7:0] FBYTE1: FIFO1 Data Count Indication Bit

FBYTE2, FBYTE1	Description
Write	Set the transfer count.
Read	Read valid data count.

Read (valid data count)

Transmission: Data count written to the FIFO but not yet transmitted
Reception: Data count that has been received but not read by the reception FIFO

Write (transfer count)

Transmission: Set 0x00.
Reception: Set the data count that triggers reception interrupts.

Table 9-4 Data Count Stored in FIFO

FIFO Capacity	Data Length	SSR:AWC	Maximum FBYTE Value	Number of Data Items That Can Be Stored
16 bytes	5 to 16 bits	0	8	8
		1	4	4
	20, 24, and 32 bits	1	4	
32 bytes	5 to 16 bits	0	16	16
		1	8	8
	20, 24, and 32 bits	1	8	
64 bytes	5 to 16 bits	0	32	32
		1	16	16
	20, 24, and 32 bits	1	16	

Notes:

- *Set the FBYTE of the transmission FIFO to 0x00, When only transmitting data during master operation..*
- *When receiving data during master operation, set the number of transmission data items when the transmission FIFO is empty and the SCR:TIE and SSR:TBIE bits are both "0".*
- *To disable reception (SCR:RXE = 0) while receiving data during master operation, disable the transmission FIFO first and then disable transmission/reception.*
- *When the reception FIFO is not used, the reception block size to be assumed during block transfer is set to "1", regardless of the setting of this register.*
- *To change the FBYTE of the reception FIFO, first disable reception and then change the setting.*
- *Any setting that exceeds the FIFO capacity is prohibited.*
- *Set the FIFO byte register (FBYTE) after setting the FIFO selection bit (FCR1:FSEL).*
- *The FIFO selection bit (FCR1:FSEL) and the FIFO byte register (FBYTE) cannot be set at the same time.*
- *When transmitting, FBYTE shows the value of the number of writing the TDR register times minus 1. This is the reason that the data to transmit written to the TDR register is stored to the FIFO when a previous data to transmit is remained in the TDR register. When the data in the*

TDR register is transmitted, the data not transmitted in the FIFO is transferred to the TDR register.

- *When receiving, FBYTE shows the value of the number of the received data in the FIFO not read from the RDR register. The data that has not received yet to the RDR register is not included.*

9.17. Transmission FIFO Interrupt Control Register (FTICR)

The transmission FIFO interrupt control register (FTICR) sets the condition for interrupts triggered by the valid data count of FIFO transmission.

Transmission FIFO Interrupt Control Register (FTICR)

Figure 9-20 shows the bit configuration of the transmission FIFO interrupt control register (FTICR).

Figure 9-20 Bit Configuration of Transmission FIFO Interrupt Control Register (FTICR)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	FTICR2								FTICR1							
R/W Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W
Protection Attribute	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The function of this bit depends on the setting of the ECR:TXBLKEN bit. The details are as follows:

- ECR:TXBLKEN = 0
  The FTICR register sets the trigger level for interrupts according to the valid transmission data count (remaining amount) of the transmission FIFO.
- FCR:TXBLKEN = 1
  The FTICR register sets the trigger level for interrupts according to the vacant data count of the transmission FIFO.

The table below lists the settings made by the FCR1:FSEL bit.

Table 9-5 Transmission FIFO Setting by FCR1:FSEL

FSEL	Selection of Transmission FIFO	Transmission FIFO Interrupt Control Register
0	FIFO1	FTICR1
1	FIFO2	FTICR2

- The initial valid data count for triggering interrupts of the FTICR register is 0x00.
- Set the number of data items that generate a transmission interrupt in the FTICR of the transmission FIFO.
  - For ECR:TXBLKEN = 0, if the data count that was set becomes equal to or smaller than the valid data count of the transmission FIFO (FTICR or FBYTE), the interrupt flag (FDRQ) is set to "1".
  - For ECR:TXBLKEN = 1, if the data count that was set becomes equal to or smaller than the vacant data count of the transmission FIFO, the interrupt flag (FDRQ) is set to "1". If the data count that was set in this bit is greater than the vacant data count of the transmission FIFO, the interrupt flag (FDRQ) is set to "0".
- Set FTICR so that it satisfies: FTICR ≤ FIFO capacity - 2.
- The read value indicates the valid data count of the FIFO.
- Transmission FIFO: Data count written to the transmission FIFO that has not yet been transmitted.
- Reception FIFO: Data count received by in the reception FIFO that has not yet been read.

FTICR2: FIFO2 Data Count Indication Bit
FTICR1: FIFO1 Data Count Indication Bit

FTICR2, FTICR1	Description
Write	Set the number of valid data items that generate an interrupt.
Read	Read valid data count.

Notes:

- *Any setting that would exceed the FIFO capacity is prohibited.*
- *The setting value cannot be read.*
- *As the FIFO data count for transmission, the transmission data count that has been written, minus 1, is displayed as the valid data count. This is because any attempt to write data to the TDR register stores the data in the transmission FIFO if the TDR register contains data that has yet to be transmitted. When the data in the TDR register is transmitted, data in the transmission FIFO that has yet to be transmitted is transferred to the TDR register.*
- *The FIFO data count for reception represents the data count received by the reception FIFO and which has not yet been read. The data count does not include the data being received by the RDR register.*
- *For ECR:TXBLKEN = 0, if block transfer is performed during DMA transfer, only 1 can be set as the block size.*
- *To perform block transfer with ECR:TXBLKEN = 1, set the value of the block size to this register.*

9.18. Extended Control Register (ECR)

The extended control register (ECR) can be used to set block transfer.

Bit Configuration of the Extended Control Register (ECR)

Figure 9-21 shows the bit configuration of the extended control register (ECR).

Figure 9-21 Bit Configuration of Extended Control Register (ECR)

Bit	15 ... 8	7	6	5	4	3	2	1	0
Field	(ESR)	Reserved	Reserved	Reserved	EISEL	REIE	TEIE	RX BLKEN	TX BLKEN
R/W Attribute		R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W	R/W
Protection Attribute		-	-	-	-	-	-	-	-
Initial Value		0	0	0	0	0	0	0	0

[bit7:5] Reserved: Reserved Bits

[bit4] EISEL: Error Interrupt Request Output Selection Bit

This bit selects the output of the error interrupts from the interrupt request pin or outputs them through the error interrupt request pin. For details on interrupt output selection, see Table 2-2 and Table 2-3.

Bit	Description
0	Output a reception error interrupt request from the reception interrupt request pin. Output a transmission error interrupt request from the transmission interrupt request pin.
1	Output a reception error interrupt request from the reception error interrupt request pin. Output a transmission error interrupt request from the transmission error interrupt request pin.

[bit3] REIE: Reception Error Interrupt Enable Bit

This bit enables/disables reception error interrupt request output.
For details on the target interrupt factors, see Table 2-3.

Bit	Description
0	Disable reception error interrupts.
1	Enable reception error interrupts.

[bit2] TEIE: Transmission Error Interrupt Enable Bit

This bit enables/disables transmission error interrupt request output.
For details on the target interrupt factors, see Table 2-2.

Bit	Description
0	Disable transmission error interrupts.
1	Enable transmission error interrupts.

[bit1] RXBLKEN: Reception Block Transfer Set Bit

This bit sets reception-side DMA transfer mode.

- "0": Perform DMA transfer in demand transfer mode.
- "1": Perform DMA transfer in block transfer mode.

Bit	Description
0	Demand transfer mode
1	Block transfer mode

[bit0] TXBLKEN: Transmission Block Transfer Set Bit

This bit sets transmission-side DMA transfer mode.

- "0": Perform DMA transfer in demand transfer mode.
- "1": Perform DMA transfer in block transfer mode.

Bit	Description
0	Demand transfer mode
1	Block transfer mode

9.19. Extended Status Register (ESR)

The extended status register (ESR) checks the reception block transfer error flag and transmission block transfer error flag.

Extended Status Register (ESR)

Figure 9-22 shows the bit configuration of the extended status register (ESR).

Figure 9-22 Bit Configuration of Extended Status Register (ESR)

Bit	15	14	13	12	11	10	9	8	7 ... 0
Field	Reserved	Reserved	Reserved	Reserved	RXUDR	TXOVR	RBERR	TBERR	(ECR)
R/W Attribute	R0,W0	R0,W0	R0,W0	R0,W0	R,WX	R,WX	R,WX	R,WX	
Protection Attribute	-	-	-	-	-	-	-	-	
Initial Value	0	0	0	0	0	0	0	0	

[bit15:12] Reserved: Reserved Bits

[bit11] RXUDR: Reception FIFO under Run Flag Bit

If the reception FIFO becomes empty as a result of block transfer and reading is generated, this bit is set to "1".

- This bit is reset when the ESRC:RXUDRC bit in the clear register is set to "1".
- Please refer to Table 2-3 for output of interrupt request.

Bit	Description
0	Indicate that reception FIFO under run is not occurring.
1	Indicate that reception FIFO under run is occurring.

Notes:

- *Software reset (SCR:UPCL = 1) resets this bit to "0".*
- *If reception block transfer enable (ECR:RXBLKEN) is "0", this bit is set to "0".*
- *This bit is set only when the reception FIFO is used.*
- *This bit is set to "0" when Reception FIFO is disabled.*

[bit10] TXOVR: Transmission FIFO Overrun Flag Bit

If the transmission FIFO becomes full by block transfer and writing occurs, this bit is set to "1".

- This bit is reset when the ESRC:TXOVRC bit in the clear register is set to "1".
- Please refer to Table 2-2 for output of interrupt request.

Bit	Description
0	Indicate that transmission FIFO overrun is not occurring.
1	Indicate that transmission FIFO overrun is occurring.

Notes:

- *Software reset (SCR:UPCL = 1) resets this bit to "0".*
- *If transmission block transfer enable (ECR:TXBLKEN) is "0", this bit is set to "0".*
- *This bit is set only when the transmission FIFO is used.*
- *This bit is set to "0" when Transmission FIFO is disabled.*

[bit9] RBERR: Reception Block Transfer Error Bit

This bit indicates that a block transfer error has occurred on the reception side.
If block transfer is performed with a value greater than the threshold set in the FBYTE register, this bit is set to "1" to indicate a block transfer error. A reception interrupt occurs if reception interrupts are enabled (SCR:RIE = 1).

- This bit is reset when the ESRC:RBERRC bit in the clear register is set to "1".
- Please refer to Table 2-3 for output of interrupt request.

Bit	Description
0	Indicate that a reception block transfer error is not occurring.
1	Indicate that a reception block transfer error is occurring.

Notes:

- *Software reset (SCR:UPCL = 1) resets this bit to "0".*
- *If reception block transfer enable (ECR:RXBLKEN) is "0", or if reception interrupt enable (SCR:RIE) is "0", this bit is set to "0".*

[bit8] TBERR: Transmission Block Transfer Error Bit

This bit indicates that a block transfer error has occurred on the transmission side.
If block transfer is performed with a value greater than the threshold set in the FTICR register, this bit is set to "1" to indicate a block transfer error. At that time, a transmission interrupt occurs if transmission interrupt is enabled (SCR:TIE = 1).

- This bit is reset when the ESRC:TBERRC bit in the clear register is set to "1".
- Please refer to Table 2-2 for output of interrupt request.

Bit	Description
0	Indicate that a transmission block transfer error is not occurring.
1	Indicate that a transmission block transfer error is occurring.

Notes:

- *Software reset (SCR:UPCL = 1) resets this bit to "0".*
- *If transmission block transfer enable (ECR:TXBLKEN) is "0", this bit is set to "0".*
- *When the transmission FIFO is OFF and block transfer is performed under one of the following conditions, this bit is set to "1" to indicate a block transfer error.*
  - A transmission interrupt is enabled (SCR:TIE = 1) and the transmission data register contains data (SSR:TDRE = 0).
  - Transmission is in progress (SSR:TBI = 0) with transmission interrupt disabled (SCR:TIE = 0) and with transmission bus idle interrupt enabled (SCR:TBIE = 1).
  - If transmission interrupt is enabled (SCR:TIE = 1) and transmission bus idle interrupt is enabled (SCR:TBIE = 1), the condition for enabling transmission interrupt (SCR:TIE = 1) is given priority.

9.20. Transmission Block Size Register (TBSIZE)

This register sets the block size for transmission block transfer.

Bit Configuration of the Transmission Block Size Register (TBSIZE)

Figure 9-23 shows the bit configuration of the transmission block size register (TBSIZE).

Figure 9-23 Bit Configuration of Transmission Block Size Register (TBSIZE)

Bit	15 ... 8	7	6	5	4	3	2	1	0
Field	-	TBSIZE							
R/W Attribute		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute		-	-	-	-	-	-	-	-
Initial Value		0	0	0	0	0	0	0	0

The TBSIZE register sets the block size of the transmission block transfer.

:[bit7:0] TBSIZE: Transmission Block Size byte

TBSIZE	Description
Write	Sets the number of transmission blocks.
Read	Read the setting value.

Notes:

- *Any setting that would exceed the FIFO capacity is prohibited.*
- *Setting this bit to "0" is prohibited.*
- *Any setting that would exceed the value set in the transmission FIFO interrupt control register (FTICR) is prohibited.*
- *In this register, set the same value as the transmission block size set in the DMA controller.*
- *When the transmission FIFO is not used, the transmission block size to be assumed at block transfer is "1", regardless of the setting of this register.*

9.21. Serial Auxiliary Control Status Clear Register (SACSRC)

The serial auxiliary control status clear register (SACSRC) can clear the bits in the serial auxiliary control status register (SACSR).

Note:

- *For details on the operations that can be performed on this register, also see the description of the target register SACSR.*

Bit Configuration of the Serial Auxiliary Control Status Clear Register (SACSRC)

Figure 9-24 shows the bit configuration of the serial auxiliary control status clear register (SACSRC).

Figure 9-24 Bit Configuration of Serial Auxiliary Control Status Clear Register (SACSRC)

Bit	15	14	13	12	11	10	9	8
Field	STSTC	Reserved	TBEENC	CSEIEC	CSEC	Reserved		TINTC
R/W Attribute	R0,W	R0,W0	R0,W	R0,W	R0,W	R0,W0		R0,W
Protection Attribute	-	-	-	-	-	-		-
Initial Value	0	0	0	0	0	00		0

Bit	7	6	5	4	3	2	1	0
Field	TINTEC	TSYNEC	TRGEC	Reserved				TMREC
R/W Attribute	R0,W	R0,W	R0,W	R0,W0				R0,W
Protection Attribute	-	-	-	-				-
Initial Value	0	0	0	0000				0

[bit15] STSTC: Clearing the Serial Test Bit

Writing "1" to this bit resets the SACSR:STST to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit14] Reserved: Reserved Bit

[bit13] TBEENC: Clearing the Transfer Byte Error Enable Bit

If "1" is written to this bit, SACSR:TBEEN is reset to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit12] CSEIEC: Clearing the Chip Select Error Interrupt Enable Bit

If "1" is written to this bit, SACSR:CSEIE is reset to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit11] CSEC: Clearing the Chip Select Error Flag
If "1" is written to this bit, SACSR:CSE is reset to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit10:9] Reserved: Reserved Bits

[bit8] TINTC: Clearing the Timer Interrupt Flag
Writing "1" to this bit resets SACSR:TINT to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit7] TINTEC: Clearing the Timer Interrupt Enable Bit
Writing "1" to this bit resets SACSR:TINTE to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit6] TSYNEC: Clearing the Synchronous Transmission Enable Bit
Writing "1" to this bit resets SACSR:TSYNE to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit5] TRGEC: Clearing the External Trigger Enable Bit
Writing "1" in this bit resets the SACSR:TRGE to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit4:1] Reserved: Reserved Bits

[bit0] TMREC: Clearing the Serial Timer Enable Bit
Writing "1" to this bit resets SACSR:TMRE to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

9.22. FIFO Control Clear Register 1 (FCR1C)

FIFO control clear register 1 (FCR1C) can clear a bit in FIFO control register 1 (FCR1).

Note:

- *For operations on this register, also see the description of the target register FCR1.*

Bit Configuration of FIFO Control Clear Register 1 (FCR1C)

Figure 9-25 shows the bit configuration of FIFO control clear register 1 (FCR1C).

Figure 9-25 Bit Configuration of FIFO Control Clear Register 1 (FCR1C)

Bit	15	14	13	12	11	10	9	8	7 ... 0
Field	Reserved	Reserved	Reserved	FLSTEC	FRIIEC	FDRQC	FTIEC	FSELC	(FCR0C)
R/W Attribute	R0,W0	R0,W0	R0,W0	R0,W	R0,W	R0,W	R0,W	R0,W	
Protection Attribute	-	-	-	-	-	-	-	-	
Initial Value	0	0	0	0	0	0	0	0	

[bit15:13] Reserved: Reserved Bits

[bit12] FLSTEC: Clearing the Retransmission Data Lost Detection Enable Bit

Writing "1" to this bit resets FCR1:FLSTE to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit11] FRIIEC: Clearing the Reception FIFO Idle Detection Enable Bit

Writing "1" to this bit resets FCR1:FRIIE to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit10] FDRQC: Clearing the Transmission FIFO Data Request Bit

Writing "1" to this bit resets FCR1:FDRQ to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit9] FTIEC: Clearing the Transmission FIFO Interrupt Enable Bit

Writing "1" to this bit resets FCR1:FTIE to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit8] FSELC: Clearing the FIFO Selection Bit

Writing "1" to this bit resets FCR1:FSEL to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

9.23. FIFO Control Clear Register 0 (FCR0C)

FIFO control clear register 0 (FCR0C) can clear a bit in FIFO control register 0 (FCR0).

Note:

- *For details on the operations that can be performed on this register, also see the description of target register FCR0.*

Bit Configuration of FIFO Control Clear Register 0 (FCR0C)

Figure 9-26 shows the bit configuration of FIFO control clear register 0 (FCR0C).

Figure 9-26 Bit Configuration of FIFO Control Clear Register 0 (FCR0C)

Bit	15 ... 8	7	6	5	4	3	2	1	0
Field	(FCR1C)	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	FE2C	FE1C
R/W Attribute		R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W	R0,W
Protection Attribute		-	-	-	-	-	-	-	-
Initial Value		0	0	0	0	0	0	0	0

[bit7:2] Reserved: Reserved Bits

[bit1] FE2C: Clearing the FIFO2 Operation Enable Bit

Writing "1" to this bit resets FCR0:FE2 to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit0] FE1C: Clearing the FIFO1 Operation Enable Bit

Writing "1" to this bit resets FCR0:FE1 to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

9.24. Extended Status Clear Register (ESRC)

The extended status clear register (ESRC) can clear a bit in the extended status register (ESR).

Extended Status Clear Register (ESRC)

Figure 9-27 shows the bit configuration of the extended status clear register (ESRC).

Figure 9-27 Bit Configuration of Extended Status Clear Register (ESRC)

Bit	15	14	13	12	11	10	9	8	7 ... 0
Field	Reserved	Reserved	Reserved	Reserved	RXUDRC	TXOVRC	RBERRC	TBERRC	-
R/W Attribute	R0,W0	R0,W0	R0,W0	R0,W0	R0,W	R0,W	R0,W	R0,W	
Protection Attribute	-	-	-	-	-	-	-	-	
Initial Value	0	0	0	0	0	0	0	0	

[bit15:12] Reserved: Reserved Bits

[bit11] RXUDRC: Reception FIFO underrun flag clear bit

Writing "1" to this bit resets ESR:RXUDR to "0".
Writing "0" to this bit is invalid.
"0" is always read from this bit.

[bit10] TXOVRC: Transmission FIFO overrun flag clear bit

Writing "1" to this bit resets ESR:TXOVR to "0".
Writing "0" to this bit is invalid.
"0" is always read from this bit.

[bit9] RBERRC: Reception Block Transfer Error Clear Bit

Writing "1" to this bit resets ESR:RBERR to "0".
Writing "0" to this bit is invalid.
"0" is always read from this bit.

[bit8] TBERRC: Transmission Block Transfer Error Clear Bit

Writing "1" to this bit resets ESR:TBERR to "0".
Writing "0" to this bit is invalid.
"0" is always read from this bit.

9.25. Serial Auxiliary Control Status Set Register (SACSRS)

The serial auxiliary control status set register (SACSRS) can clear a bit in the serial auxiliary control status register (SACSR).

Note:

- *For operations on this register, also see the description of the target register SACSR.*

Bit Configuration of Serial Auxiliary Control Status Set Register (SACSRS)

Figure 9-28 shows the bit configuration of the serial auxiliary control status set register (SACSRS).

Figure 9-28 Bit Configuration of Serial Auxiliary Control Status Set Register (SACSRS)

Bit	15	14	13	12	11	10	9	8
Field	STSTS	Reserved	TBEENS	CSEIES	Reserved			
R/W Attribute	R0,W	R0,W0	R0,W	R0,W	R0,W0			
Protection Attribute	-	-	-	-	-			
Initial Value	0	0	0	0	0000			

Bit	7	6	5	4	3	2	1	0
Field	TINTES	TSYNES	TRGES	Reserved				TMRES
R/W Attribute	R0,W	R0,W	R0,W	R0,W0				R0,W
Protection Attribute	-	-	-	-				-
Initial Value	0	0	0	0000				0

[bit15] STSTS: Setting the Serial Test Bit

Writing "1" to this bit sets the SACSR:STST to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit14] Reserved: Reserved Bit

[bit13] TBEENS: Transfer Byte Error Enable Set Bit

Writing "1" to this bit sets SACSR:TBEEN to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit12] CSEIES: Chip Select Error Interrupt Enable Set Bit

Writing "1" to this bit sets SACSR:CSEIE to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit11:8] Reserved: Reserved Bits

[bit7] TINTES: Setting the Timer Interrupt Enable Bit

Writing "1" to this bit sets SACSR:TINTE to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit6] TSYNES: Setting the Synchronous Transmission Enable Bit

Writing "1" to this bit sets SACSR:TSYNE to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit5] TRGES: Setting the External Trigger Enable Bit

Writing "1" to this bit sets SACSR:TRGE to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit4:1] Reserved: Reserved Bits

[bit0] TMRES: Setting the Serial Timer Enable Bit

Writing "1" to this bit sets SACSR:TMRE to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

9.26. FIFO Control Set Register 1 (FCR1S)

FIFO control set register 1 (FCR1S) can set a bit in FIFO control register 1 (FCR1).

Note:

- *For details on the operations that can be performed on this register, also see the description of target register FCR1.*

Bit Configuration of FIFO Control Set Register 1 (FCR1S)

Figure 9-29 shows the bit configuration of FIFO control set register 1 (FCR1S).

Figure 9-29 Bit Configuration of FIFO Control Set Register 1 (FCR1S)

Bit	15	14	13	12	11	10	9	8	7 ... 0
Field	Reserved	Reserved	Reserved	FLSTES	FRIIES	Reserved	FTIES	FSELS	(FCR0S)
R/W Attribute	R0,W0	R0,W0	R0,W0	R0,W	R0,W	R0,W0	R0,W	R0,W	
Protection Attribute	-	-	-	-	-	-	-	-	
Initial Value	0	0	0	0	0	0	0	0	

[bit15:13] Reserved: Reserved Bits

[bit12] FLSTES: Setting the Retransmission Data Lost Detection Enable Bit

Writing "1" to this bit sets FCR1:FLSTE to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit11] FRIIES: Setting the Reception FIFO Idle Detection Enable Bit

Writing "1" to this bit sets FCR1:FRIIE to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit10] Reserved: Reserved Bit

[bit9] FTIES: Setting the Transmission FIFO Interrupt Enable Bit

Writing "1" to this bit sets FCR1:FTIE to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit8] FSELS: Setting the FIFO Selection Bit

Writing "1" to this bit sets FCR1:FSEL to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

9.27. FIFO Control Set Register 0 (FCR0S)

FIFO control set register 0 (FCR0S) can set a bit in FIFO control register 0 (FCR0).

Note:

- *For details on the operations that can be performed on this register, also see the description of target register FCR0.*

Bit Configuration of FIFO Control Set Register 0 (FCR0S)

Figure 9-30 shows the bit configuration of FIFO control set register 0 (FCR0S).

Figure 9-30 Bit Configuration of FIFO Control Set Register 0 (FCR0S)

Bit	15 ... 8	7	6	5	4	3	2	1	0
Field	(FCR1S)	Reserved	Reserved	FLDS	FSETS	FCL2S	FCL1S	FE2S	FE1S
R/W Attribute		R0,W0	R0,W0	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
Protection Attribute		-	-	-	-	-	-	-	-
Initial Value		0	0	0	0	0	0	0	0

[bit7:6] Reserved: Reserved Bits

[bit5] FLDS: Setting the FIFO Pointer Reload Bit

Writing "1" to this bit sets FCR0:FLD to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit4] FSETS: Setting the FIFO Pointer Saving Bit

Writing "1" to this bit sets FCR0:FSET to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit3] FCL2S: Setting the FIFO2 Reset Bit

Writing "1" to this bit sets FCR0:FCL2 to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit2] FCL1S: Setting the FIFO1 Reset Bit

Writing "1" to this bit sets FCR0:FCL1 to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit1] FE2S: Setting the FIFO2 Operation Enable Bit

Writing "1" to this bit sets FCR0:FE2 to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit0] FE1S: Setting the FIFO1 Operation Enable Bit

Writing "1" to this bit sets FCR0:FE1 to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

CHAPTER 38: I²C Interface (I²C Communication Control Interface)



This chapter explains the I²C function supported by operation mode 4 of the multi-function serial interface.

1. Overview of the I²C Interface (I²C Communication Control Interface)
2. I²C Interface Interrupts
3. I²C Interface Operation
4. Serial Timer Operation
5. Dedicated Baud Rate Generator
6. Block Transfer
7. I²C Communication Operation Flowchart Examples
8. I²C Interface Registers

1. Overview of the I²C Interface (I²C Communication Control Interface)

The I²C interface (I²C communication control interface) supports the I²C bus and operates as a master/slave device on the I²C bus. Also, this interface has transmission and reception FIFOs (64 bytes each).

I²C Interface (I²C Communication Control Interface) Functions

	Function	Description
1	Data buffer	– Full duplex, double buffering (when FIFOs are not used) – Transmission and reception FIFOs (64 bytes each) (when FIFOs are used)
2	Serial input	– For serial clock and data input, noise from 2 to 38 bus clocks is filtered out.
3	Transfer mode	– Synchronous
4	Baud rate	– Dedicated baud rate generator is provided (configured from a 15-bit reload counter).
5	Data length	– 8 bits
6	Signaling method	– NRZ (Non Return to Zero)
7	Interrupt request	– Reception interrupt – Transmission interrupt – Status interrupt (INT interrupt, stop condition interrupt, repeated start detection interrupt, and serial timer interrupt) – Transmission FIFO interrupt (when the transmission FIFO is not higher than the interrupt trigger level or when the transmission FIFO is empty, reception FIFO under run) – Reception FIFO interrupt (reception FIFO under run) – DMA transfer is supported for both transmission and reception.
8	I²C	– Master/slave transmission and reception function – Arbitration function – Clock synchronization function – Transfer direction detection function – Function to generate and detect the repeated start condition – Bus error detection function – General call addressing function – 7-bit addressing as master and slave – An interrupt can be generated upon transfer and bus error occurrence. – 10-bit addressing function can be supported by program. – Standard-mode and Fast-mode are supported. Fast-mode Plus, High-speed mode, and Ultra Fast-mode are not supported.
9	Timer function	– A 16-bit serial timer is provided. – A division value can be selected for the operation clock (division by 1 to 256). – Activation by external trigger is available.
10	FIFO	– Transmission and reception FIFOs are provided (64 bytes for the transmission FIFO and 64 bytes for the reception FIFO). – Transmission FIFO and reception FIFO can be selected. – Transmission data can be retransmitted. – The timing of the reception FIFO interrupt can be changed by software. – Independent FIFO reset is supported.

2. I²C Interface Interrupts

The following factors may generate an interrupt request for an I²C interface interrupt:

- After the 1st byte is transmitted and received / After data is transmitted and received
- Stop condition
- Repeated start condition
- FIFO transmission data request
- FIFO reception data completion
- When the serial timer comparison value (STMCR) matches the serial timer value (STMR)

I²C Interface Interrupts

Table 2-1 lists the interrupt control bits and interrupt factors of the I²C interface.

Table 2-1 Interrupt Control Bits and Interrupt Factors of the I²C Interface

Interrupt Type	Interrupt Request Flag Bit	Flag Register	Interrupt Factor	Interrupt Factor Enable Bit	Interrupt Request Flag Clearing
Status	INT	IBCR	After 1st byte is transmitted and received*1 (except when master is operating with SSR:DMA= 1)	IBCR:INTE	Writing "1" to interrupt flag clear bit (IBCRC.INTC)
			After data is transmitted and received*1 (where SSR:DMA= 0)		
			Bus error (EIBCR:BEC= 0)		
			Arbitration lost detected		
			Reserved address detected		
			NACK received		
			Reception FIFO full during slave reception operation (where SSR:DMA= 0)		Writing "1" to IBCRC:INTC, after reception data is read until reception FIFO is empty
	SPC	IBSR	Stop condition	IBCR:CNDE	Writing "1" to interrupt flag clear bit IBSRC:SPCC
	RSC		Repeated start detected		Writing "1" to interrupt flag clear bit IBSRC:RSCC
	TINT	SACSR	When serial timer register (STMR) matches serial timer comparison register (STMCR)	SACSR:TINTE	Writing "1" to timer interrupt flag clear bit (SACSRC:TINTC)

Interrupt Type	Interrupt Request Flag Bit	Flag Register	Interrupt Factor	Interrupt Factor Enable Bit	Interrupt Request Flag Clearing
Reception	RDRF	SSR	Reserved address received	SMR:RIE	Reading reception data (RDR)
			After receiving data		
			Receiving as much data as FBYTE setting value		Reading reception data until reception FIFO is empty
			Reception idle detected when FRIIE= 1		
	ORE	SSR	Overrun error	SMR:RIE ECR:REIE	Writing "1" to reception error flag bit (SSR:REC)
	RXUDR	ESR	Reception FIFO under run	ECR:REIE	- Writing "1" to reception FIFO under run flag clear bit (ESRC:RXUDRC) - Disable reception FIFO 1)When FCR1:FSEL=0, FCR0:FE2=0 2)When FCR1:FSEL=1, FCR0:FE1=0
	RBERR	ESR	Reception block transfer error	ECR:REIE	- Writing "1" to reception block transfer error clear bit (ESRC:RBERRC) - Disable Block transfer mode (ECR:RXBLKEN=0)
Transmission	TDRE	SSR	Transmission register is empty	SMR:TIE	Writing to transmission data register (TDR), or writing "1" to transmission FIFO operation enable bit (retransmission) when bit is "0" and transmission FIFO has valid data*2
			Writing "1" to transmission buffer empty flag set bit (SSR:TSET)		
	FDRQ	FCR1	Transmission FIFO stores data count equal to or less than FTICR setting value or is empty	FCR1:FTIE	Writing "1" to FIFO transmission data request clear bit (FCR1C:FDRQC), or transmission FIFO is full
	TBI (SSR: DMA=1)	SSR	No transmission operation	SSR:TBIE	Writing to transmission data register (TDR), or writing "1" to transmission FIFO operation enable bit (retransmission) when bit is "0" and transmission FIFO has valid data*3
			Writing "1" to transmission buffer empty flag set bit (SSR:TSET)		
	TXOVR	ESR	Transmission FIFO overrun	ECR:TEIE	- Writing "1" to transmission FIFO overrun flag clear bit (ESRC:TXOVRC) - Disable transmission FIFO 1)When FCR1:FSEL=0, FCR0:FE1=0 2)When FCR1:FSEL=1, FCR0:FE2=0
	TBERR	ESR	Transmission block transfer error	ECR:TEIE	- Writing "1" to transmission block transfer error clear bit (ESRC:TBERRC) - Disable Block transfer mode (ECR:TXBLKEN=0)

*1: If normal data can be transmitted and received and SSR:TDRE is "0", no interrupt is generated. The purpose is to support DMA transfer.
If you want to generate an IBCR:INT flag when transmitting and receiving data, the SSR:TDRE bit must be "1" before the time that the IBCR:INT flag is set.

*2 After the SSR:TDRE bit becomes "0", set the SMR:TIE bit to "1".

*3 After the SSR:TBI bit becomes "0", set the SSR:TBIE bit to "1".

Switching of Error Interrupt Request Output

The error interrupt request output method varies depending on the setting of the error interrupt request output switch bit (ECR:EISEL), the transmission/reception error interrupt enable bit (ECR:REIE, TEIE), or the transmission/reception interrupt enable bit (SMR:RIE, TIE, SSR:TBIE, FCR1:FTIE).

Table 2-2 Transmission Interrupt Request Output Methods

ECR:EISEL	Interrupt Output	Target Interrupt Enable Bit	Interrupt Factor
0	Transmission interrupt request output	SMR:TIE	Transmission data empty (SSR:TDRE)
		SSR:TBIE	No transmission operation (SSR:TBI)
		FCR1:FTIE	Transmission FIFO empty (FCR1:FDRQ)
		ECR:TEIE	Transmission FIFO overrun (ESR:TXOVR) Transmission block transfer error (ESR:TBERR)
	Transmission error interrupt request output	-	No interrupt factor
1	Transmission interrupt request output	SMR:TIE	Transmission data empty (SSR:TDRE)
		SSR:TBIE	No transmission operation (SSR:TBI)
		FCR1:FTIE	Transmission FIFO empty (FCR1:FDRQ)
	Transmission error interrupt request output	ECR:TEIE	Transmission FIFO overrun (ESR:TXOVR) Transmission block transfer error (ESR:TBERR)

Table 2-3 Reception Interrupt Request Output Methods

ECR:EISEL	Interrupt Output	Target Interrupt Enable Bit	Interrupt Factor
0	Reception interrupt request output	SMR:RIE	Reception FIFO full (SSR:RDRF) Overrun error (SSR:ORE)
		ECR:REIE	Reception FIFO under run (ESR:RXUDR) Reception block transfer error (ESR:RBERR)
	Reception error interrupt request output	-	No interrupt factor
1	Reception interrupt request output	SMR:RIE	Reception FIFO full (SSR:RDRF)
	Reception error interrupt request output	ECR:REIE	Overrun error (SSR:ORE) Reception FIFO under run (ESR:RXUDR) Reception block transfer error (ESR:RBERR)

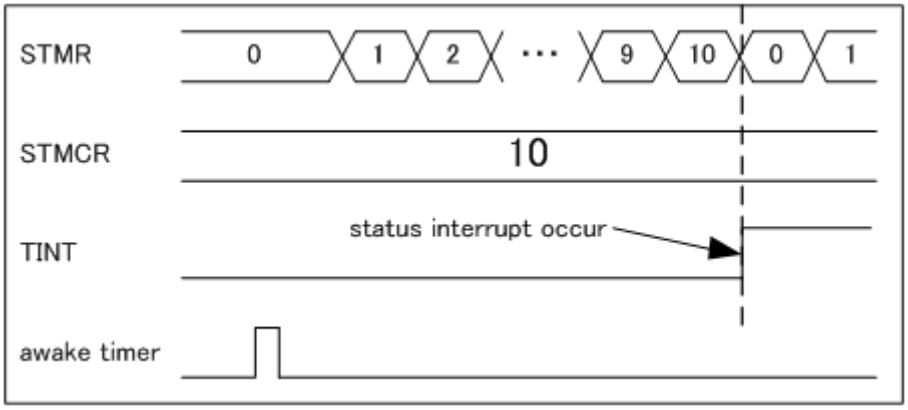
2.1. Timer Interrupt Generation and Flag Set Timing

A timer interrupt is generated when the serial timer register (STMR) and the serial timer comparison register (STMCR) match.

Timer Interrupt Generation and Flag Set Timing

- The timer interrupt flag (SACSR:TINT) is set to "1" when the serial timer register (STMR) matches the serial timer comparison register.
  If timer interrupts are enabled (SACSR:TINTE= 1) at this time, a status interrupt is generated.

Figure 2-1 Timer Interrupt Generation Timing



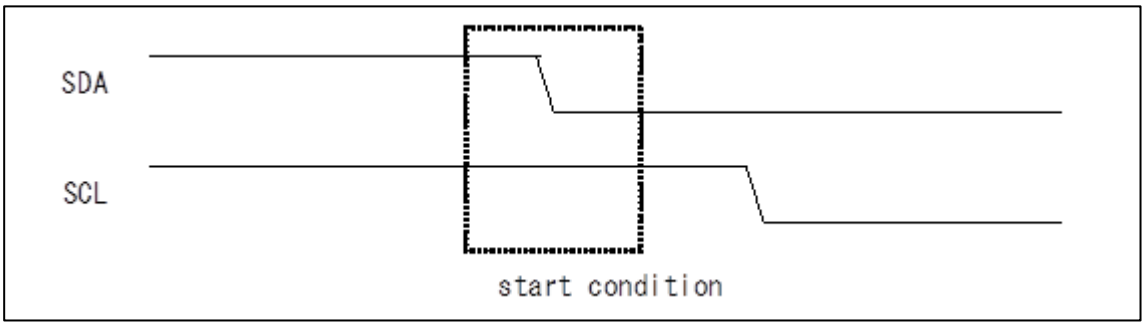
3. I²C Interface Operation

The I²C interface communicates using two bidirectional bus lines: a serial data line (SDA) and a serial clock line (SCL).

I²C Bus Start Condition

The following figure shows the start condition of the I²C bus.

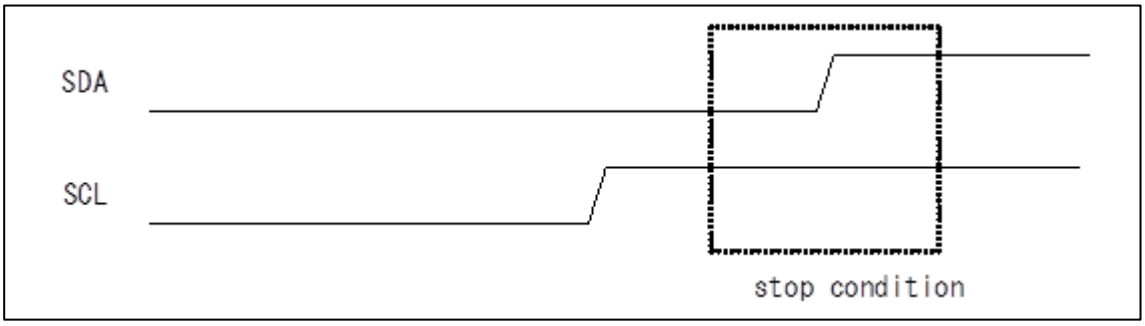
Figure 3-1 Start Condition



I²C Bus Stop Condition

The following figure shows the stop condition of the I²C bus.

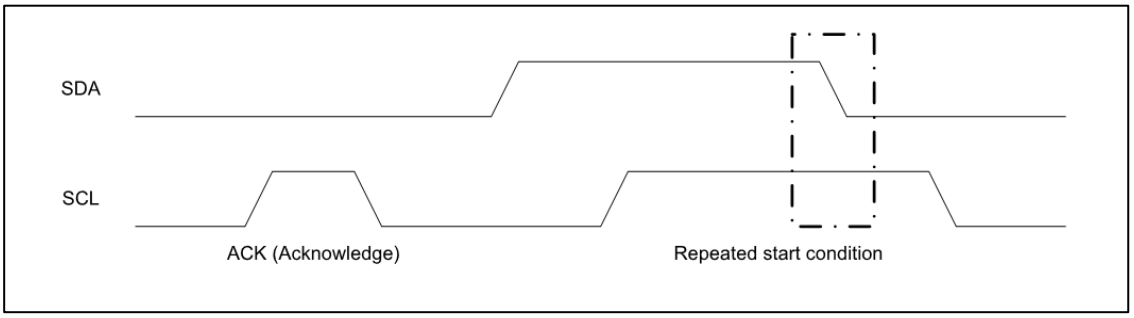
Figure 3-2 Stop Condition



I²C Bus Repeated Start Condition

The following figure shows the repeated start condition of the I²C bus.

Figure 3-3 Repeated Start Condition



3.1. Master Mode

In master mode, the clock is output to the I²C bus upon the occurrence of the start condition for the I²C bus. Master mode is selected and the ACT bit in the IBCR register is set to "1" when the I²C bus is in the idle state (SCL="H" and SDA="H") and the MSS bit in the IBCR register is set to "1".

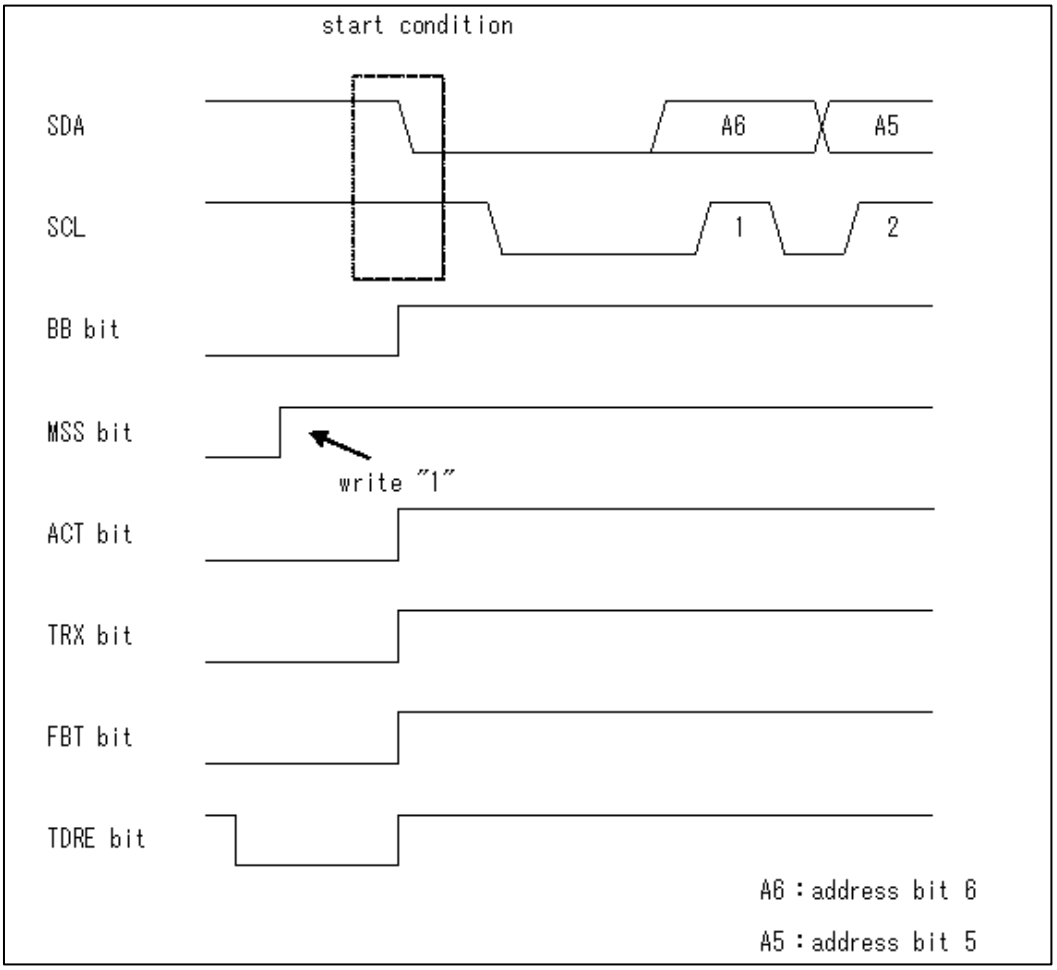
Start Condition Occurrence

The start condition is output under the following condition.

- "1" is written to the IBCR:MSS bit when SDA="H", SCL="H", ISMK:EN=1, and IBSR:BB=0.

The IBCR:ACT bit is set to "1" when the start condition is output to the I²C bus. Then, the IBSR:BB bit is set to "1" when the start condition is received, indicating that communication with the I²C bus is in progress (see Figure 3-4).

Figure 3-4 Relationship between Start Condition Output and Individual Bits



Note:

- *In operation mode 4 (I²C mode), use the bus clock at 8 MHz or more. A baud rate generator setting exceeding 400 kbps is prohibited in this mode.*

Slave Address Output

After the start condition is output, the set data in the TDR register is output as the address, beginning with bit7. If FIFO is enabled, the data written first in the TDR register is output. bit0 is used as a data direction bit (R/W). When the data direction bit (R/W) is "0", the data indicates the write direction (from master to slave). Set the address in the TDR register before writing IBCR:MSS= 1 or IBCR:SCC= 1.

Figure 3-5 and Figure 3-6 show the output timing of the address and data direction.

Figure 3-5 Address and Data Direction (When FIFO Is Disabled)

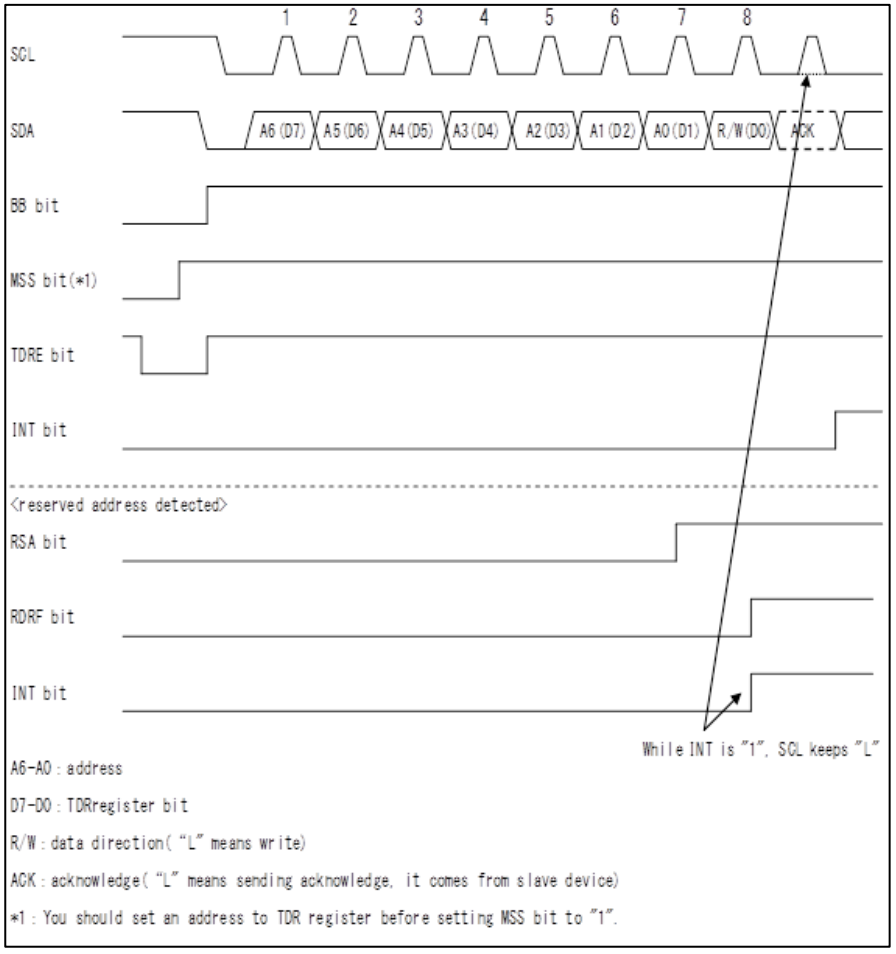
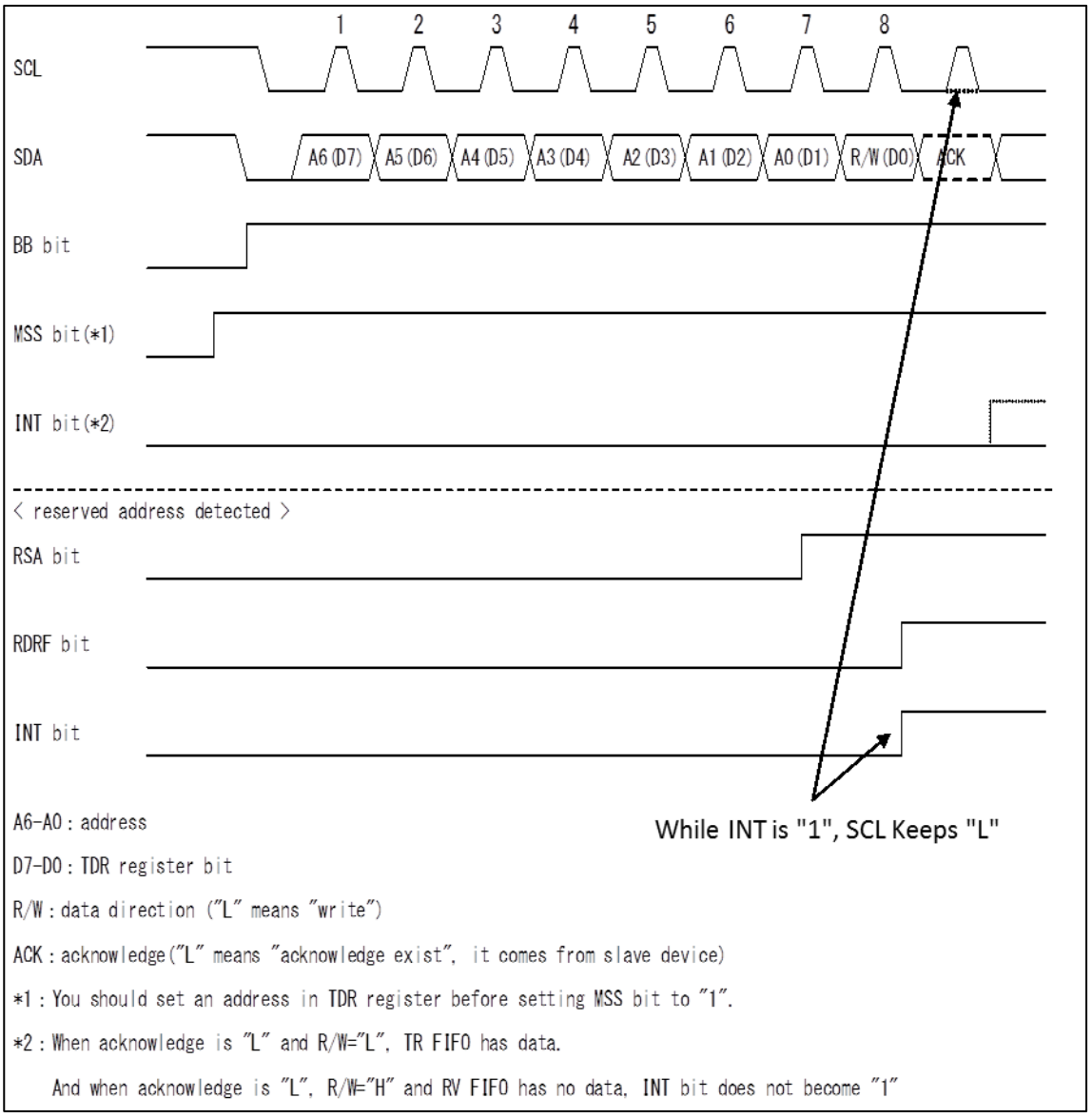


Figure 3-6 Address and Data Direction (When the Transmission and Reception FIFOs Are Enabled)



Acknowledgment Reception by Transmission of the 1st Byte

After the data direction bit (R/W) is output, the I²C interface receives an acknowledgment from the slave. The following tables describe the operations when FIFO is enabled or disabled.

Table 3-1 Operations after Acknowledgment Reception When DMA Mode Is Disabled (IBSR:RSA Bit = 0 and SSR:DMA Bit = 0)

Transmission FIFO	Reception FIFO	Transmission FIFO State	Reception FIFO State	Data Direction Bit (R/W)	Operation Immediately after Acknowledgment Reception	
					Acknowledgment Is ACK	Acknowledgment Is NACK
Disabled	Disabled	-	-	0	If the SSR:TDRE bit is "1", the IBCR:INT bit is set to "1", and a wait is performed. If the SSR:TDRE bit is "0", the IBCR:INT bit remains "0", and no wait is performed.	The IBCR:INT bit is set to "1", and a wait is performed.
				1		
Disabled	Enabled	-	Without data	0	If the SSR:TDRE bit is "1", the IBCR:INT bit is set to "1", and a wait is performed. If the SSR:TDRE bit is "0", the IBCR:INT bit remains "0", and no wait is performed.	The IBCR:INT bit is set to "1", and a wait is performed.
			With data		The IBCR:INT bit is set to "1", and a wait is performed.	
			-	1	If the SSR:TDRE bit is "1", the IBCR:INT bit is set to "1", and a wait is performed. If the SSR:TDRE bit is "0", the IBCR:INT bit remains "0", and no wait is performed.	
Enabled	Disabled	-	-	0	If the SSR:TDRE bit is "1", the IBCR:INT bit is set to "1", and a wait is performed. If the SSR:TDRE bit is "0", the IBCR:INT bit remains "0", and no wait is performed.	The IBCR:INT bit is set to "1", and a wait is performed.
				1		
Enabled	Enabled	-	Without data	0	If the SSR:TDRE bit is "1", the IBCR:INT bit is set to "1", and a wait is performed. If the SSR:TDRE bit is "0", the IBCR:INT bit remains "0", and no wait is performed.	The IBCR:INT bit is set to "1", and a wait is performed.
			With data		The IBCR:INT bit is set to "1", and a wait is performed.	
			-	1	If the SSR:TDRE bit is "1", the IBCR:INT bit is set to "1", and a wait is performed. If the SSR:TDRE bit is "0", the IBCR:INT bit remains "0", and no wait is performed.	

Table 3-2 Operations after Acknowledgment Reception When DMA Mode Is Enabled (IBSR:RSA Bit = 0 and SSR:DMA Bit = 1)

Transmission FIFO	Reception FIFO	Transmission FIFO State	Reception FIFO State	Data Direction Bit (R/W)	Operation Immediately after Acknowledgment Reception	
					Acknowledgment Is ACK	Acknowledgment Is NACK
Disabled	Disabled	-	-	0	If the SSR:TDRE bit is "1", the SSR:TBI bit is set to "1", and a wait is performed. If the SSR:TDRE bit is "0", the SSR:TBI bit remains "0", and no wait is performed.	The IBCR:INT bit is set to "1", and a wait is performed.
				1		
Disabled	Enabled	-	Without data	0	If the SSR:TDRE bit is "1", the SSR:TBI bit is set to "1", and a wait is performed. If the SSR:TDRE bit is "0", the SSR:TBI bit remains "0", and no wait is performed.	The IBCR:INT bit is set to "1", and a wait is performed.
			With data		The IBCR:INT bit is set to "1", and a wait is performed.	
			-	1	If the SSR:TDRE bit is "1", the SSR:TBI bit is set to "1", and a wait is performed. If the SSR:TDRE bit is "0", the SSR:TBI bit remains "0", and no wait is performed.	
Enabled	Disabled	-	-	0	If the SSR:TDRE bit is "1", the SSR:TBI bit is set to "1", and a wait is performed. If the SSR:TDRE bit is "0", the SSR:TBI bit remains "0", and no wait is performed.	The IBCR:INT bit is set to "1", and a wait is performed.
				1		

Transmission FIFO	Reception FIFO	Transmission FIFO State	Reception FIFO State	Data Direction Bit (R/W)	Operation Immediately after Acknowledgment Reception	
					Acknowledgment Is ACK	Acknowledgment Is NACK
Enabled	Enabled	-	Without data	0	If the SSR:TDRE bit is "1", the SSR:TBI bit is set to "1", and a wait is performed. If the SSR:TDRE bit is "0", the SSR:TBI bit remains "0", and no wait is performed.	The IBCR:INT bit is set to "1", and a wait is performed.
			With data		The IBCR:INT bit is set to "1", and a wait is performed.	
Enabled	Enabled	-		1	If the SSR:TDRE bit is "1", the SSR:TBI bit is set to "1", and a wait is performed. If the SSR:TDRE bit is "0", the SSR:TBI bit remains "0", and no wait is performed.	The IBCR:INT bit is set to "1", and a wait is performed.

When DMA Mode is Disabled (SSR:DMA = 0)

FIFO disabled (transmission and reception FIFOs both disabled)

- If the IBSR:RSA bit is "0" and if the SSR:TDRE bit is "1" after acknowledgment reception, the interrupt flag (IBCR:INT) is set to "1", the SCL remains "L", and a wait is performed. For the wait, the interrupt flag is set to "0" and the wait is released by writing "1" to the interrupt flag clear bit (IBCRC:INTC). If the SSR:TDRE bit is "0" and ACK is received, a clock is generated on the SCL without setting the interrupt flag to "1".
- If the IBSR:RSA bit is "1", the interrupt flag (IBCR:INT) is set to "1" after the reserved address is received (before acknowledgment), the SCL remains "L", and a wait is performed. After the RDR register is read, the IBCR:ACKE bit and transmission data are set and "1" is written to the interrupt flag clear bit (IBCRC:INTC). This sets the interrupt flag to "0" and releases the wait.
- The received acknowledgment is set in the IBSR:RACK bit. The IBSR:RACK bit is verified during the wait. If the acknowledgment is NACK, "0" is written to the IBCR:MSS bit to generate the stop condition, or "1" is written to the IBCR:SCC bit to generate the repeated start condition. At this time, the IBCR:INT bit is automatically cleared to "0".

FIFO enabled

- The following settings are required for the FIFOs before the IBCR:MSS bit is set to "1".
  - To transmit to a slave (data direction bit = 0), set the data including the slave address for the transmission FIFO.
  - To receive data from a slave (data direction bit = 1), set the number of bytes to be received in the FIFO byte count register. Also, write to the transmission data register by using the slave address, the data direction bit, and as many dummies as the desired quantity of data to be received.
- If the IBSR:RSA bit is "0" and the received acknowledgment is ACK, the interrupt flag (IBCR:INT) is not set to "1", and data is transmitted and received according to the data direction bit (without a wait). If the acknowledgment is NACK, the interrupt flag (IBCR:INT) is set to "1", the SCL remains "L", and a wait is performed.
- The IBSR:RACK bit stores the received acknowledgment. The IBSR:RACK bit is verified during the wait. If the acknowledgment is NACK, "0" is written to the IBCR:MSS bit to generate the stop condition, or "1" is written to the IBCR:SCC bit to generate the repeated start condition. At this time, the IBCR:INT bit is automatically cleared to "0".

When DMA Mode is Enabled (SSR:DMA = 1)

FIFO disabled (transmission and reception FIFOs both disabled)

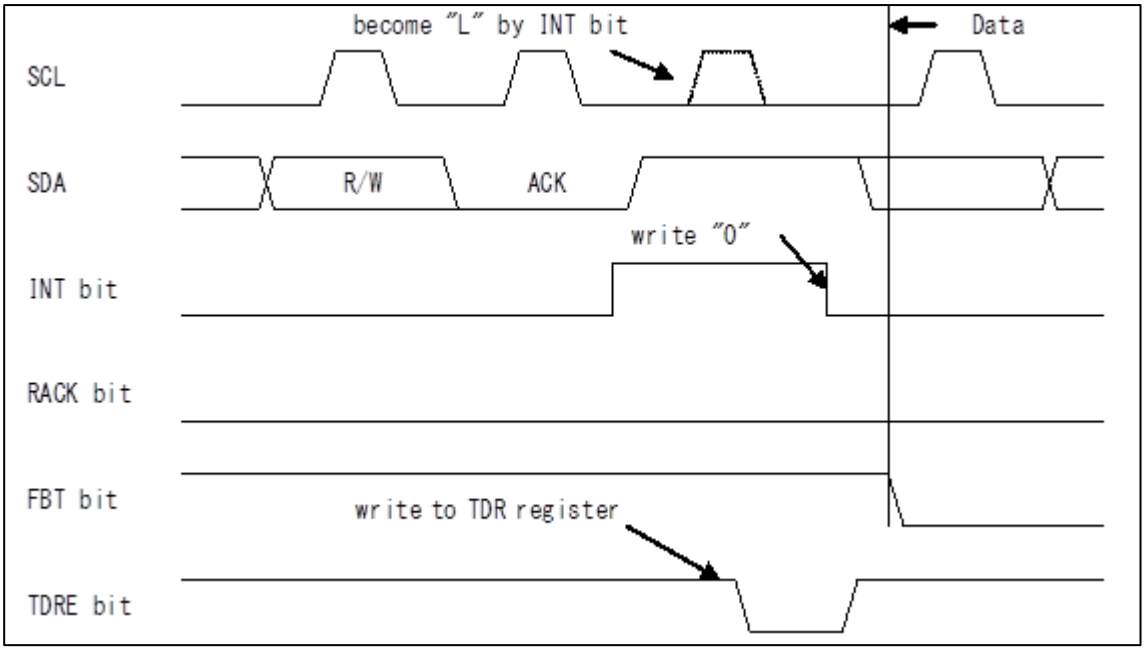
- If the IBSR:RSA bit is "0" and if the SSR:TDRE bit is "1" after an acknowledgment is received, the transmission bus idle flag (SSR:TBI) is set to "1", the SCL remains "L", and a wait is performed. The transmission bus idle flag is set to "0", which releases the wait, when the data to be transmitted is written to the TDR register. If the SSR:TDRE bit is "0" and ACK is received, a clock is generated on the SCL without setting the transmission bus idle flag (SSR:TBI) to "1".
- If the IBSR:RSA bit is "1", the interrupt flag (IBCR:INT) is set to "1" after the reserved address is received (before an acknowledgment), the SCL remains "L", and a wait is performed. After the RDR register is read, the IBCR:ACKE bit and transmission data are set, the writing "1" to the IBCRC:INTC sets the interrupt flag to "0", which releases the wait.
- The received acknowledgment is set in the IBSR:RACK bit. The IBSR:RACK bit is verified during the wait. If the acknowledgment is NACK, "0" is written to the IBCR:MSS bit to generate the stop condition, or "1" is written to the IBCR:SCC bit to generate the repeated start condition. At this time, the IBCR:INT bit is automatically cleared to "0".

FIFO enabled

- The following settings are required for the FIFOs before the IBCR:MSS bit is set to "1".
  - To transmit to a slave (data direction bit = 0), set the data including the slave address for the transmission FIFO.
  - To receive data from a slave (data direction bit = 1), set the number of bytes to be received in the FIFO byte count register. Also, write to the transmission data register by using the slave address, the data direction bit, and as many dummies as the desired quantity of data to be received.
- If the IBSR:RSA bit is "0" and the received acknowledgment is ACK, the interrupt flag (IBCR:INT) is not set to "1", and data is transmitted and received according to the data direction bit (without a wait). If the acknowledgment is NACK, the interrupt flag (IBCR:INT) is set to "1", the SCL remains "L", and a wait is performed.
- The IBSR:RACK bit stores the received acknowledgment. The IBSR:RACK bit is verified during the wait. If the acknowledgment is NACK, "0" is written to the IBCR:MSS bit to generate the stop condition,

or "1" is written to the IBCR:SCC bit to generate the repeated start condition. At this time, the IBCR:INT bit is automatically cleared to "0".

Figure 3-7 Acknowledgment (with FIFO Disabled, IBSR:RSA= 0, DMA= 0, and an ACK Response)



The wait for an address is as follows:

- Wait after acknowledgment reception if the IBSR:RSA bit is "0"
- Wait before acknowledgment reception if the IBSR:RSA bit is "1"

It does not depend on the IBCR:WSEL setting.

Figure 3-8 Acknowledgment (with FIFO Disabled, IBSR:RSA= 0, and a NACK Response)

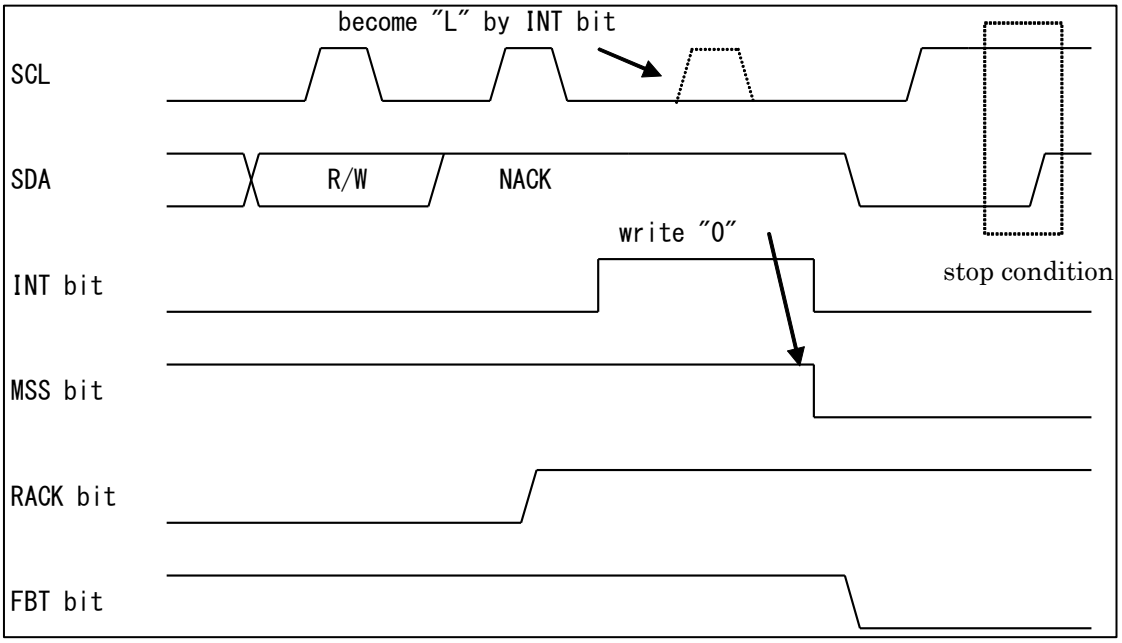


Figure 3-9 Acknowledgment (with FIFO Disabled, IBSR:RSA= 1, and an ACK Response)

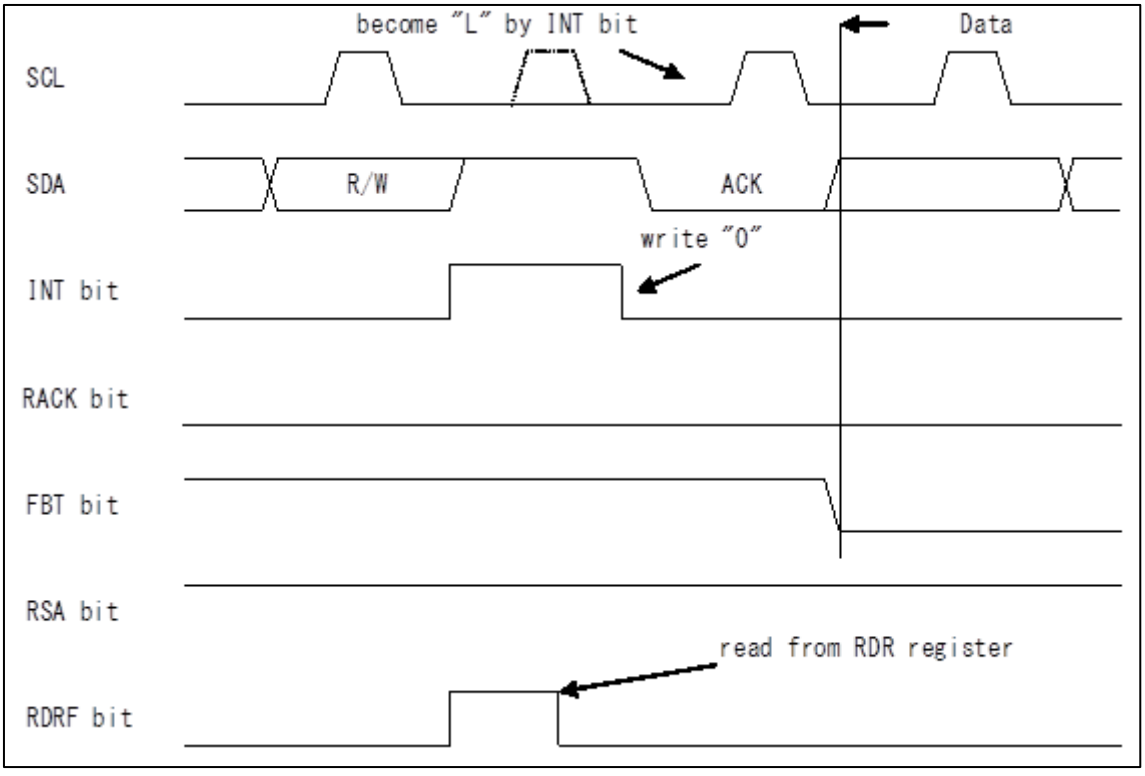


Figure 3-10 Acknowledgment (with FIFO Disabled, IBSR:RSA= 1, and a NACK Response)

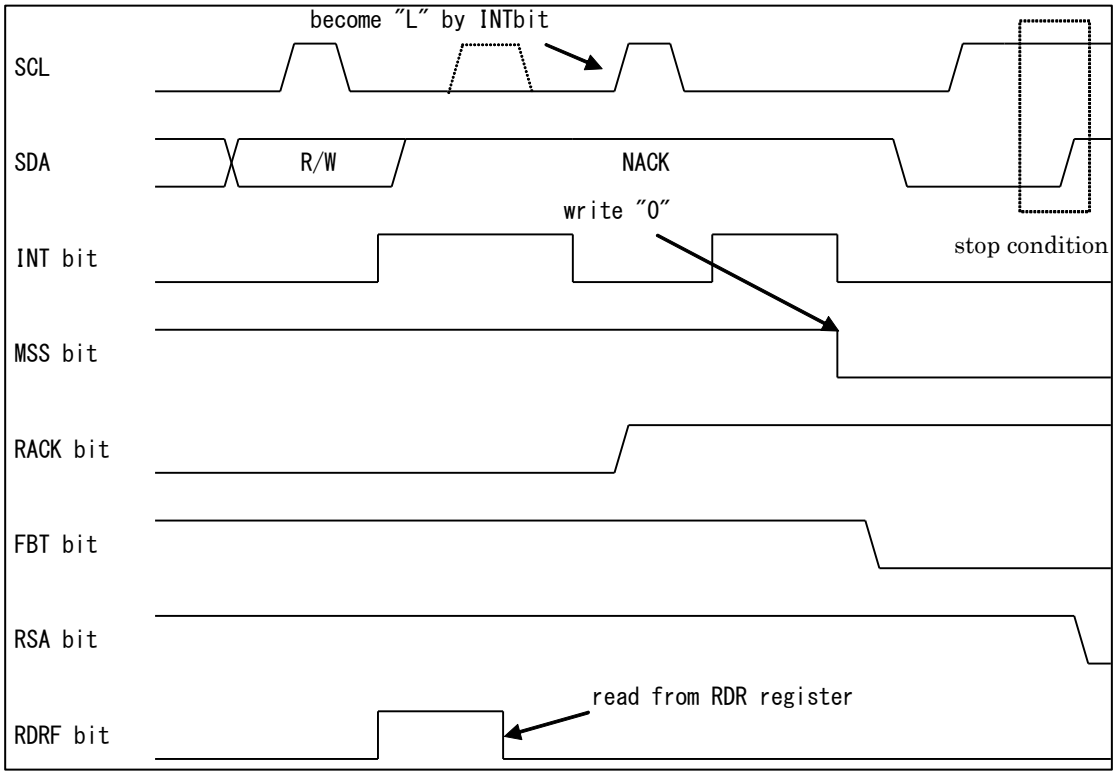
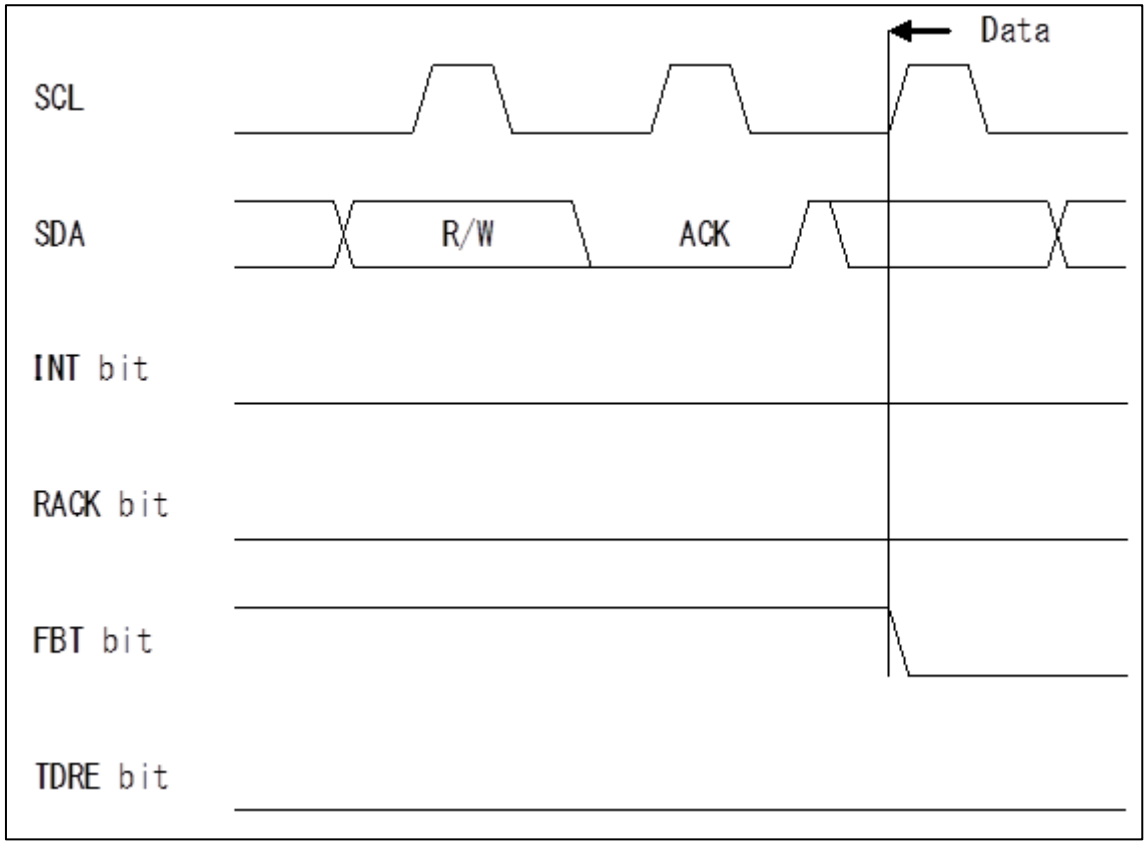


Figure 3-11 Acknowledgment (with FIFO Enabled, the Transmission FIFO with Data, the Reception FIFO without Data, IBSR:RSA= 0, and an ACK Response)



Data Transmission by the Master

If the data direction bit (R/W) is "0", data is transmitted from the master. The slave responds with ACK or NACK each time that 1 byte is transmitted.

The following tables describe the location where a wait is generated, which varies depending on the setting of the IBCR:WSEL bit.

Table 3-3 IBCR:WSEL Bit for Master Data Transmission When DMA Mode Is Disabled (SSR:DMA= 0)

WSEL Bit	Operation
0	<FIFO not used> At the 2nd or subsequent byte, the interrupt flag (IBCR:INT) is set to "1" and the SCL is set to "L" after an acknowledgment when the SSR:TDRE bit is "1" or arbitration lost is detected. Then, the wait state is entered. <FIFO used> After an acknowledgment when arbitration lost is detected or the transmission data register no longer has valid data (SSR:TDRE= 1), the interrupt flag (IBCR:INT) is set to "1". Then, the wait state is entered.
1	<FIFO not used> At the 2nd or subsequent byte, the interrupt flag (IBCR:INT) is set to "1" and the SCL is set to "L" after 1-byte data is transmitted from the master when the SSR:TDRE bit is "1" or arbitration lost is detected. Then, the wait state is entered. <FIFO used> After data transmission when arbitration lost is detected or the transmission data register no longer has valid data (SSR:TDRE= 1), the interrupt flag (IBCR:INT) is set to "1". Then, the wait state is entered.

Table 3-4 IBCR:WSEL Bit for Master Data Transmission When DMA Mode Is Enabled (SSR:DMA= 1)

WSEL Bit	Operation
0	<FIFO not used> At the 2nd or subsequent byte, the transmission bus idle flag (SSR:TBI) is set to "1" and the SCL is set to "L" after an acknowledgment when the SSR:TDRE bit is "1". Then, the wait state is entered. <FIFO used> After an acknowledgment when the transmission data register no longer has valid data (SSR:TDRE= 1), the transmission bus idle flag (SSR:TBI) is set to "1". Then, the wait state is entered.
1	<FIFO not used> At the 2nd or subsequent byte, the transmission bus idle flag (SSR:TBI) is set to "1" and the SCL is set to "L" after 1-byte data is transmitted from the master when the SSR:TDRE bit is "1". Then, the wait state is entered. <FIFO used> After 1-byte data is transmitted from the master when the transmission data register no longer has valid data (SSR:TDRE= 1), the transmission bus idle flag (SSR:TBI) is set to "1". Then, the wait state is entered.

However, in the following case, the interrupt flag (IBCR:INT) is set after an acknowledgment, regardless of the IBCR:WSEL setting.

- NACK is received at a time other than when the stop condition is set (IBCR:MSS= 0 and ACT= 1).

The following procedures are examples showing how to transmit data to a slave.

Data Transmission to a Slave When DMA Mode is Disabled (SSR:DMA= 0)

1. Transmission to a non-reserved address

- When the transmission FIFO is disabled
  1. Set the slave address (including the data direction bit) in the TDR register, and write "1" to the IBCR:MSS bit.
  2. Transmit the slave address, and then receive ACK. The interrupt flag (IBCR:INT) is set to "1".
  3. Write the data to be transmitted, to the TDR register.
  4. To update the IBCR:WSEL bit, write "1" to the interrupt flag clear bit (IBCRC:INTC) to release the wait for the I^2C bus.

5. After the transmission of 1 byte, set the interrupt flag to "1" as follows to wait for the I²C bus. If IBCR:WSEL= 0, set it after receiving an acknowledgment. If IBCR:WSEL= 1, set it immediately after the 1-byte transmission. Repeat steps 3. to 5. until the predefined quantity of data has been transmitted. However, suppose NACK is received after the release of the wait when IBCR:WSEL= 1. In that case, after an acknowledgment is received, an interrupt is generated once again to wait for the bus.
6. Set the IBCR:MSS bit to "0" to generate the stop condition, or set the IBCR:SCC bit to "1" to generate the repeated start condition.

■ When the transmission FIFO is enabled

1. Write the slave address (including the data direction bit) and transmission data to the TDR register.
2. Set the IBCR:WSEL bit, and write "1" to the IBCR:MSS bit.
3. If NACK is received during transmission, set the interrupt flag (IBCR:INT) to "1" immediately after the reception, and wait for the I²C bus. If all received responses are ACK, set the interrupt flag to "1" after the last byte is transmitted according to the IBCR:WSEL setting, and wait for the I²C bus.
4. Set the IBCR:MSS bit to "0" to generate the stop condition, or set the IBCR:SCC bit to "1" to generate the repeated start condition.

2. Transmission to the reserved address

■ When the transmission FIFO is disabled

1. Set the reserved address as the slave address in the TDR register, and write "1" to the IBCR:MSS bit.
2. Transmit the slave address. Then, the interrupt flag (IBCR:INT) is set to "1".
3. Read the RDR register, and verify the reserved address. (*1)
4. Write the data to be transmitted, to the TDR register.
5. To update the IBCR:WSEL bit, write "1" to the interrupt flag clear bit (IBCRC:INTC) to release the wait for the I²C bus.
6. After the transmission of 1 byte, set the interrupt flag to "1" as follows to wait for the I²C bus. If IBCR:WSEL= 0, set it after receiving an acknowledgment. If IBCR:WSEL= 1, set it immediately after the 1-byte transmission. Repeat steps 4. to 6. until the predefined quantity of data has been transmitted. However, suppose NACK is received after the release of the wait when IBCR:WSEL= 1. In that case, after an acknowledgment is received, an interrupt is generated once again to wait for the bus.
7. Set the IBCR:MSS bit to "0" to generate the stop condition, or set the IBCR:SCC bit to "1" to generate the repeated start condition.

■ When the transmission FIFO is enabled

1. Set the reserved address as the slave address in the TDR register, and write "1" to the IBCR:MSS bit.
2. Transmit the slave address. Then, the interrupt flag (IBCR:INT) is set to "1".
3. Read the RDR register, and verify the reserved address. (*1)
4. Write all transmission data to the TDR register (until the transmission FIFO is as full as possible).
5. If NACK is received during the transmission, set the interrupt flag (IBCR:INT) to "1" immediately after the reception, and wait for the I²C bus.
   If all received responses are ACK, set the interrupt flag to "1" after the last byte is transmitted according to the IBCR:WSEL setting, and wait for the I²C bus.
6. Set the IBCR:MSS bit to "0" to generate the stop condition, or set the IBCR:SCC bit to "1" to generate the repeated start condition.

*1 If either of the following conditions is satisfied, the IBCR:ACKE bit and the IBCR:WSEL bit must both be set to "1". Verification that the device operates as the master or a slave for the next data is also necessary.

- In a multi-master configuration, the reserved address is a general call address.
- The device is likely operating as a slave due to arbitration lost.

Data Transmission to a Slave When DMA Mode is Enabled (SSR:DMA=1)

1. Transmission to a non-reserved address

- When the transmission FIFO is disabled
  1. Set the slave address (including the data direction bit) in the TDR register, and write "1" to the IBCR:MSS bit.
  2. Transmit the slave address, and then receive ACK. The transmission bus idle flag (SSR:TBI) is set to "1".
  3. Write the data to be transmitted, to the TDR register to release the wait for the I^2C bus.
  4. After the transmission of 1 byte, set the transmission bus idle flag (SSR:TBI) to "1" as follows to wait for the I^2C bus. If IBCR:WSEL= 0, set it after receiving an acknowledgment. If IBCR:WSEL= 1, set it immediately after the 1-byte transmission.
  5. Write the data to be transmitted, to the TDR register to release the wait for the I^2C bus.
  6. After the transmission of 1 byte, set the transmission bus idle flag to "1" as follows to wait for the I^2C bus. If IBCR:WSEL= 0, set it after receiving an acknowledgment. If IBCR:WSEL= 1, set it immediately after the 1-byte transmission. Repeat steps 5. to 6. until the predefined quantity of data has been transmitted. However, suppose NACK is received after the release of the wait when IBCR:WSEL= 1. In that case, after an acknowledgment is received, the interrupt flag (IBCR:INT) is set to "1" to wait for the bus.
  7. Set the IBCR:MSS bit to "0" to generate the stop condition, or set the IBCR:SCC bit to "1"*2 to generate the repeated start condition.

- When the transmission FIFO is enabled
  1. Write the slave address (including the data direction bit) and transmission data to the TDR register.
  2. Set the IBCR:WSEL bit, and write "1" to the IBCR:MSS bit.
  3. If NACK is received during the transmission, set the interrupt flag (IBCR:INT) to "1" immediately after the reception, and wait for the I^2C bus. If all received responses are ACK, set the transmission bus idle flag (SSR:TBI) to "1" after the last byte is transmitted according to the IBCR:WSEL setting, and wait for the I^2C bus.
  4. Set the IBCR:MSS bit to "0" to generate the stop condition, or set the IBCR:SCC bit to "1"*2 to generate the repeated start condition.

2. Transmission to the reserved address

- When the transmission FIFO is disabled
  1. Set the reserved address as the slave address in the TDR register, and write "1" to the IBCR:MSS bit.
  2. Transmit the slave address. Then, the interrupt flag (IBCR:INT) is set to "1".
  3. Read the RDR register, and verify the reserved address. (*1)
  4. Write the data to be transmitted, to the TDR register.

5. To update the IBCR:WSEL bit, write "1" to the interrupt flag clear bit (IBCRC:INTC) to release the wait for the I²C bus.
6. After the transmission of 1 byte, set the transmission bus idle flag (SSR:TBI) to "1" as follows to wait for the I²C bus. If IBCR:WSEL= 0, set it after receiving an acknowledgment. If IBCR:WSEL= 1, set it immediately after the 1-byte transmission.
7. Write the data to be transmitted, to the TDR register to release the wait for the I²C bus.
8. After the transmission of 1 byte, set the transmission bus idle flag to "1" as follows to wait for the I²C bus. If IBCR:WSEL= 0, set it after receiving an acknowledgment. If IBCR:WSEL= 1, set it immediately after the 1-byte transmission. Repeat steps 7. to 8. until the predefined quantity of data has been transmitted. However, suppose NACK is received after the release of the wait when IBCR:WSEL= 1. In that case, after an acknowledgment is received, the interrupt flag (IBCR:INT) is set to "1" to wait for the bus.
9. Set the IBCR:MSS bit to "0" to generate the stop condition, or set the IBCR:SCC bit to "1"*2 to generate the repeated start condition.

■ When the transmission FIFO is enabled

1. Set the reserved address as the slave address in the TDR register, and write "1" to the IBCR:MSS bit.
2. Transmit the slave address. Then, the interrupt flag (IBCR:INT) is set to "1".
3. Read the RDR register, and verify the reserved address. (*1)
4. Write all transmission data to the TDR register (until the transmission FIFO is as full as possible).
5. If NACK is received during the transmission, set the interrupt flag (IBCR:INT) to "1" immediately after the reception, and wait for the I²C bus. If all received responses are ACK, set the transmission bus idle flag (SSR:TBI) to "1" after the last byte is transmitted according to the IBCR:WSEL setting, and wait for the I²C bus.
6. Set the IBCR:MSS bit to "0" to generate the stop condition, or set the IBCR:SCC bit to "1"*2 to generate the repeated start condition.

*1 If either of the following conditions is satisfied, the IBCR:ACKE bit and the IBCR:WSEL bit must both be set to "1". Verification that the device operates as the master or a slave for the next data is also necessary.

- In a multi-master configuration, the reserved address is a general call address.
- The device is likely operating as a slave due to arbitration lost.

*2 To issue the repeated start condition when DMA mode is enabled (SSR:DMA= 1), the SSR:TBI bit is "1", and the IBCR:INT bit is "0", perform the following procedure.

1. Write "1" to the IBCR:INT bit.
2. Verify that the IBCR:INT bit is set to "1".
3. Write the slave address to the TDR.
4. Set the IBCR:SCC bit to "1".

Notes:

- *Specifying a 7-bit slave address in master mode is prohibited when 7-bit slave address detection is enabled (ISBA:SAEN= 1).*
- *To change the IBCR register during transmission and reception, do so when the interrupt flag (IBCR:INT) is "1".*

- *If the IBCR:WSEL bit is changed, it is used as a generation condition for the interrupt flag (IBCR:INT) of the next data and the transmission bus idle flag (SSR:TBI) when DMA mode is enabled (SSR:DMA= 1).*
- *If transmission data is written to the TDR when an ACK response is detected and SSR:TDRE is "1" during data transmission, operation is as follows.*
  - When DMA mode is disabled (SSR:DMA= 0), the interrupt flag (IBCR:INT) is not set to "1", and the written data is transmitted.
  - When DMA mode is enabled (SSR:DMA= 1), the transmission bus idle flag (SSR:TBI) is not set to "1", and the written data is transmitted.
- *If transmission data is written to the TDR and an ACK response is returned when SSR:TDRE is "1" while data is being received, operation is as follows.*
  - When DMA mode is disabled (SSR:DMA= 0), the interrupt flag (IBCR:INT) is not set to "1" and only SSR:RDRF is set to "1". (If the reception FIFO is enabled, this is done when as much data as set by the FBYTE register is received.)
  - When DMA mode is enabled (SSR:DMA= 1), the transmission bus idle flag (SSR:TBI) is not set to "1" and only SSR:RDRF is set to "1". (If the reception FIFO is enabled, this is done when as much data as set by the FBYTE register is received.)

Figure 3-12 Master Transmission Interrupts 1 When FIFO Is Disabled (SSR:DMA= 0, IBCR:WSEL= 0, and IBSR:RSA= 0)



S : Start condition

W : Data direction bit (Write direction)

P : Stop condition

Sr : Repeated start condition

△ : Interrupt because of INTE = "1"

▲ : Interrupt because of CNDE = "1"

① An interrupt generated by slave address transmission + direction bit transmission + acknowledgment reception

— Write INT = "0" after the transmission data is written to the TDR register

② An interrupt generated by 1 byte transmission + acknowledgment reception

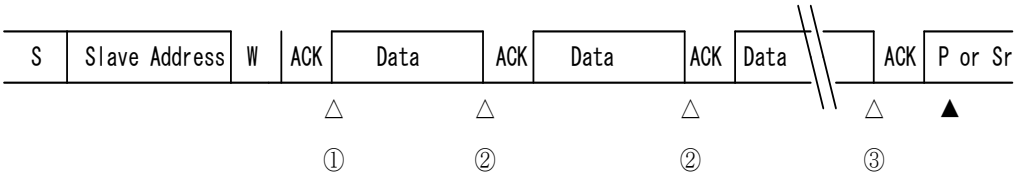
— Write INT = "0" after the transmission data is written to the TDR register

③ An interrupt generated by 1 byte transmission + acknowledgment reception

— Set MSS = "0" or MSS = "1" and SCC = "1"

*) The TDRE bit is "1" upon the generation of the interrupt flag (INT)

Figure 3-13 Master Transmission Interrupts 2 When FIFO Is Disabled (SSR:DMA= 0, IBCR:WSEL= 1, IBSR:RSA= 0, and ACK Response)



S : Start condition

W : Data direction bit (Write direction)

P : Stop condition

Sr : Repeated start condition

△ : Interrupt because of INTE = "1"

▲ : Interrupt because of CNDE = "1"

① An interrupt generated by slave address transmission + direction bit transmission + acknowledgment reception

— Write INT = "0" after the transmission data is written to the TDR register

② An interrupt generated by 1 byte transmission

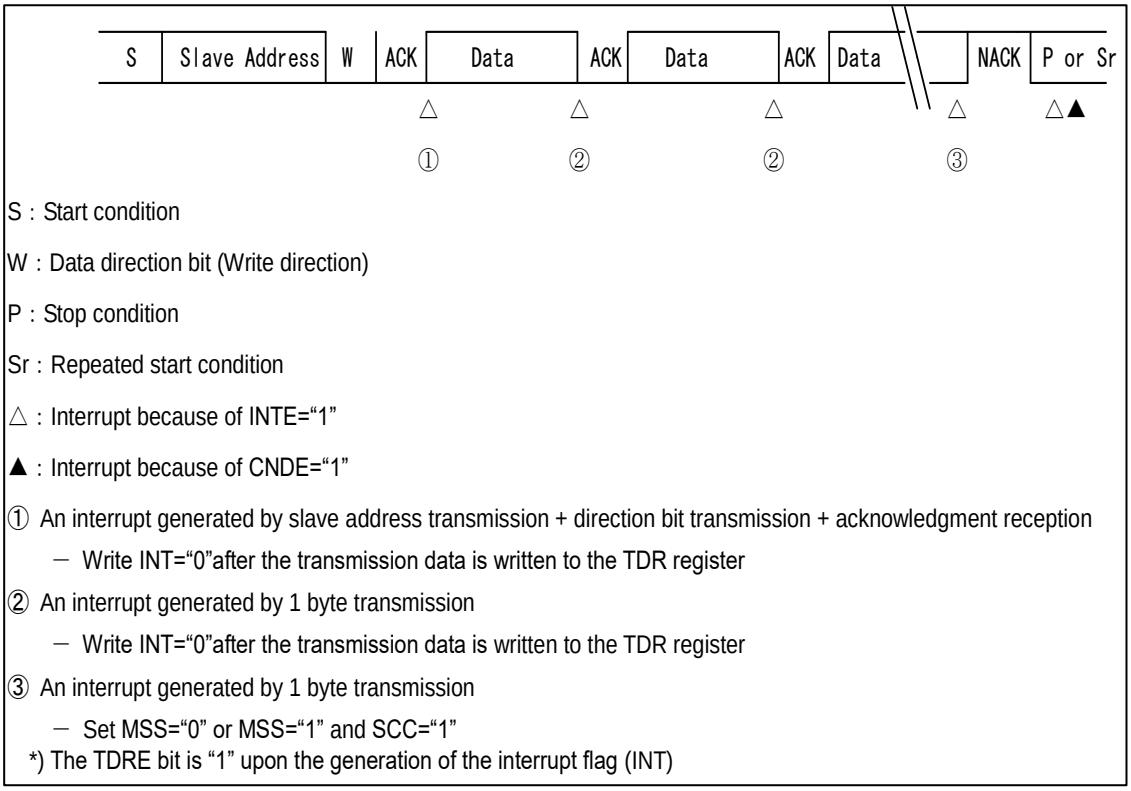
— Write INT = "0" after the transmission data is written to the TDR register

③ An interrupt generated by 1 byte transmission

— Set MSS = "0" or MSS = "1" and SCC = "1"

*) The TDRE bit is "1" upon the generation of the interrupt flag (INT)

Figure 3-14 Master Transmission Interrupts 3 When FIFO Is Disabled (SSR:DMA= 0, IBCR:WSEL= 1, IBSR:RSA= 0, and NACK Response)



S : Start condition

W : Data direction bit (Write direction)

P : Stop condition

Sr : Repeated start condition

△ : Interrupt because of INTE="1"

▲ : Interrupt because of CNDE="1"

① An interrupt generated by slave address transmission + direction bit transmission + acknowledgment reception

– Write INT="0"after the transmission data is written to the TDR register

② An interrupt generated by 1 byte transmission

– Write INT="0"after the transmission data is written to the TDR register

③ An interrupt generated by 1 byte transmission

– Set MSS="0" or MSS="1" and SCC="1"

*) The TDRE bit is "1" upon the generation of the interrupt flag (INT)

Figure 3-15 Master Transmission Interrupts 4 When FIFO Is Disabled (SSR:DMA= 0, IBCR:WSEL= 1, IBSR:RSA= 0, and Intermediate NACK Response)



S : Start condition

W : Data direction bit (Write direction)

P : Stop condition

Sr : Repeated start condition

△ : Interrupt because of INTE="1"

▲ : Interrupt because of CNDE="1"

① An interrupt generated by slave address transmission + direction bit transmission + acknowledgment reception

– Write INT="0"after the transmission data is written to the TDR register

② An interrupt generated by 1 byte transmission

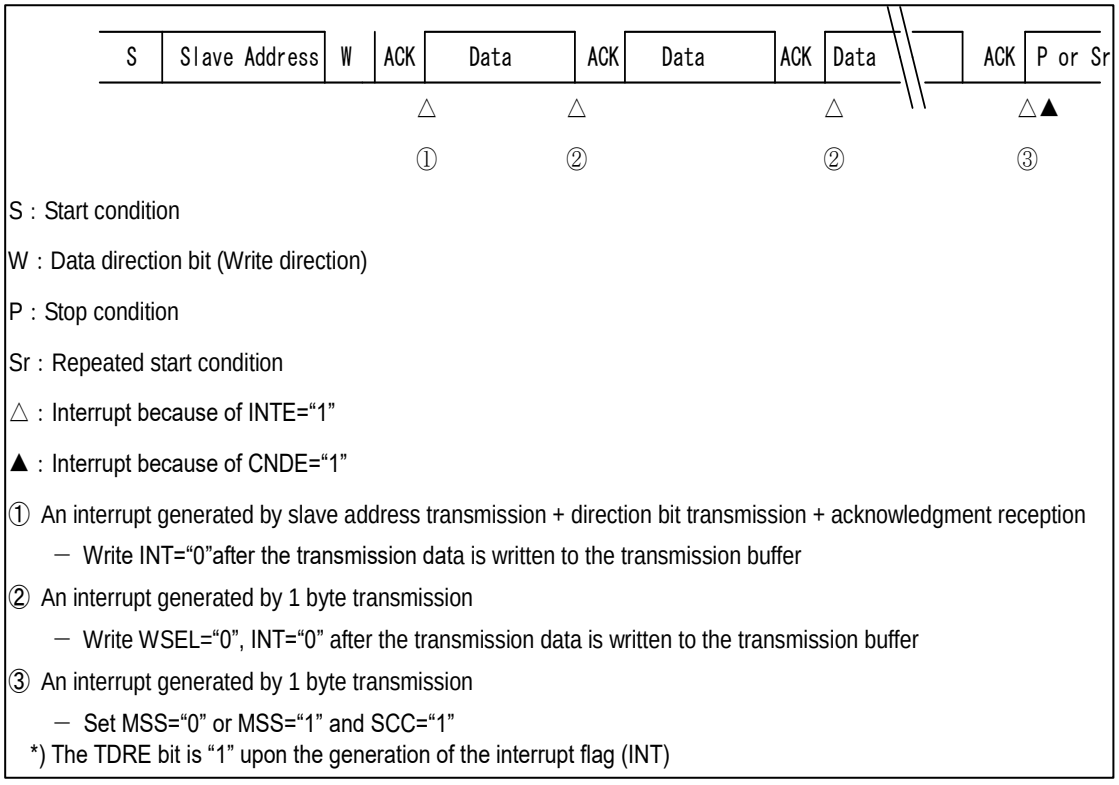
– Write INT="0"after the transmission data is written to the TDR register

③ An interrupt generated by NACK response

– Set MSS="0" or MSS="1" and SCC="1"

*) The TDRE bit is "1" upon the generation of the interrupt flag (INT)

Figure 3-16 Master Transmission Interrupts 5 When FIFO Is Disabled (SSR:DMA= 0, IBCR:WSEL= 1->"0", IBSR:RSA= 0, and ACK Response)



S : Start condition

W : Data direction bit (Write direction)

P : Stop condition

Sr : Repeated start condition

△ : Interrupt because of INTE="1"

▲ : Interrupt because of CNDE="1"

① An interrupt generated by slave address transmission + direction bit transmission + acknowledgment reception

- Write INT="0"after the transmission data is written to the transmission buffer

② An interrupt generated by 1 byte transmission

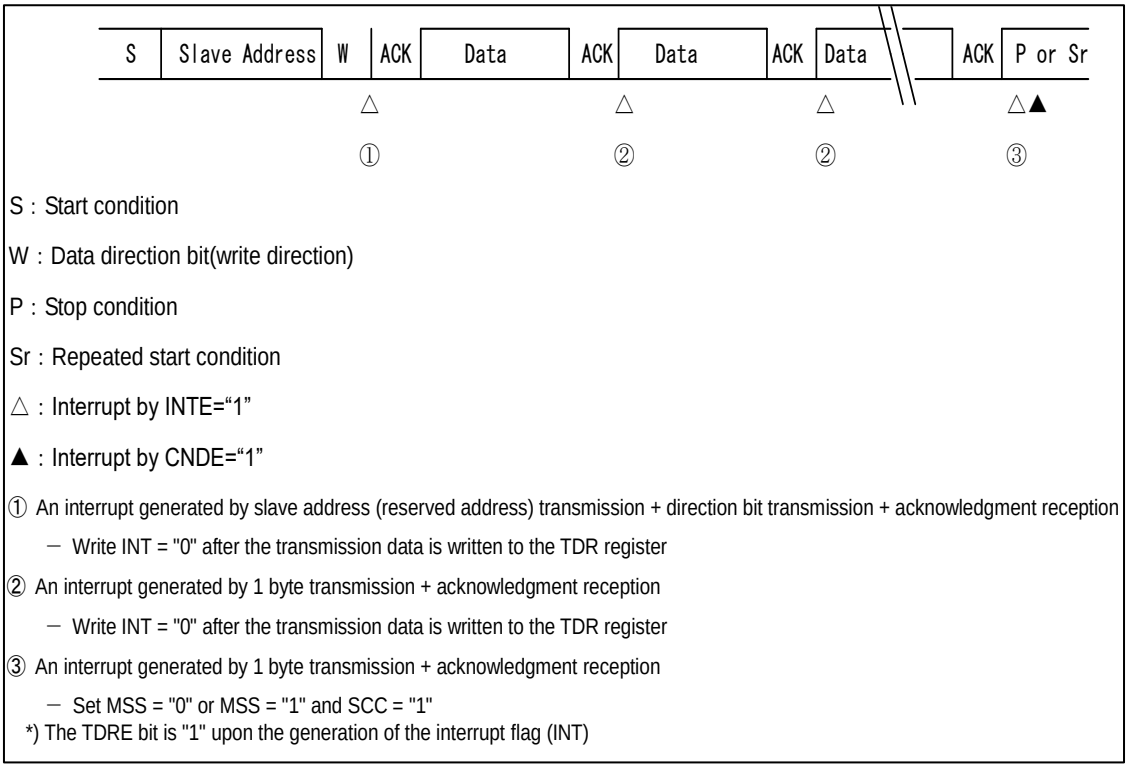
- Write WSEL="0", INT="0" after the transmission data is written to the transmission buffer

③ An interrupt generated by 1 byte transmission

- Set MSS="0" or MSS="1" and SCC="1"

*) The TDRE bit is "1" upon the generation of the interrupt flag (INT)

Figure 3-17 Master Transmission Interrupts 6 When FIFO Is Disabled (SSR:DMA= 0, IBCR:WSEL= 0, and IBSR:RSA= 1)



S : Start condition

W : Data direction bit(write direction)

P : Stop condition

Sr : Repeated start condition

△ : Interrupt by INTE="1"

▲ : Interrupt by CNDE="1"

① An interrupt generated by slave address (reserved address) transmission + direction bit transmission + acknowledgment reception

– Write INT = "0" after the transmission data is written to the TDR register

② An interrupt generated by 1 byte transmission + acknowledgment reception

– Write INT = "0" after the transmission data is written to the TDR register

③ An interrupt generated by 1 byte transmission + acknowledgment reception

– Set MSS = "0" or MSS = "1" and SCC = "1"

*) The TDRE bit is "1" upon the generation of the interrupt flag (INT)

Figure 3-18 Master Transmission Interrupts 7 When FIFO Is Enabled (SSR:DMA= 0, IBCR:WSEL= 0, IBSR:RSA= 0, and ACK Response)

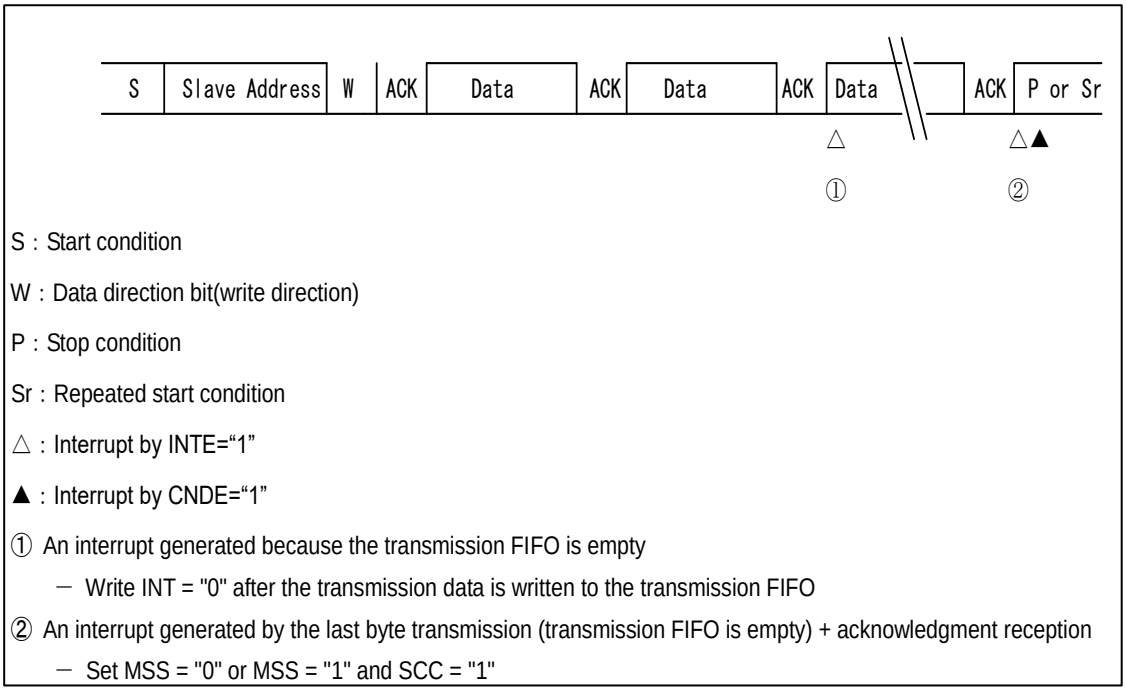


Figure 3-19 Master Transmission Interrupts 8 When FIFO Is Enabled (SSR:DMA= 0, IBCR:WSEL= 1, and IBSR:RSA= 0)

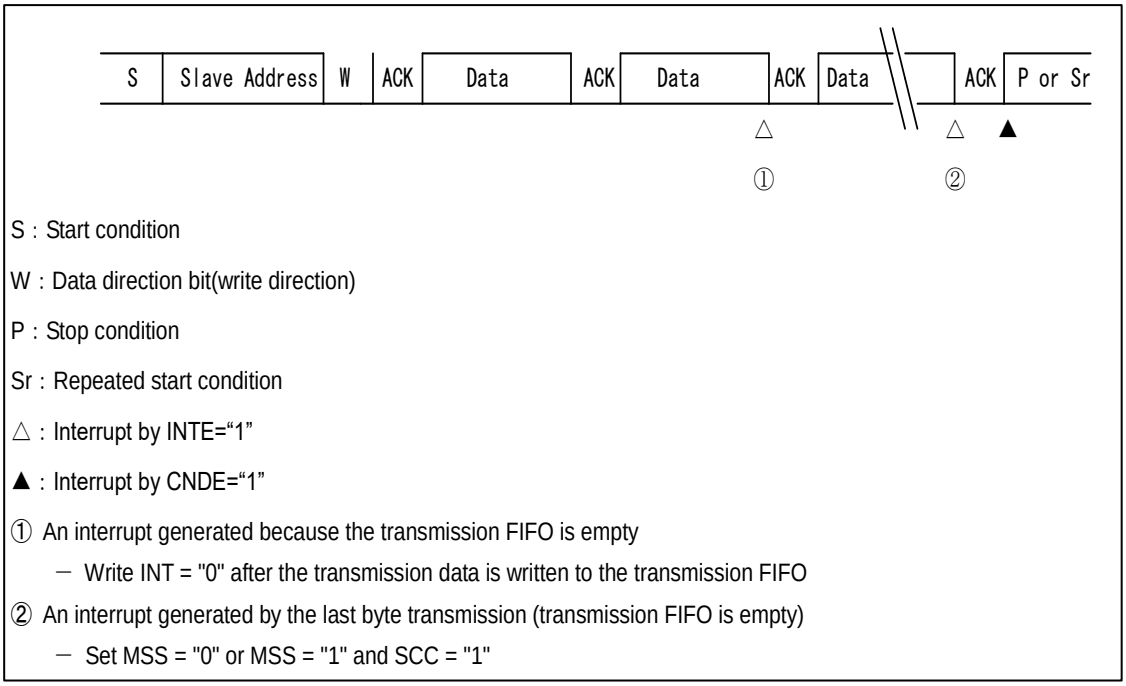
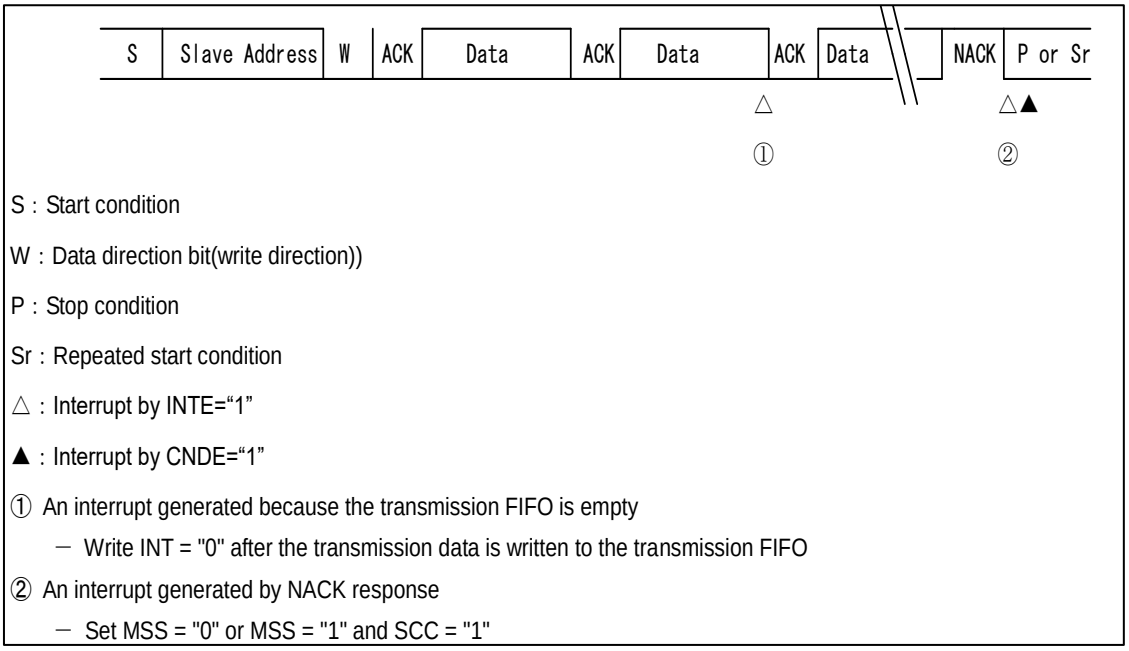


Figure 3-20 Master Transmission Interrupts 9 When FIFO Is Enabled (SSR:DMA= 0, IBCR:WSEL= 1, IBSR:RSA= 0, and NACK Response)



S : Start condition

W : Data direction bit(write direction))

P : Stop condition

Sr : Repeated start condition

△ : Interrupt by INTE="1"

▲ : Interrupt by CNDE="1"

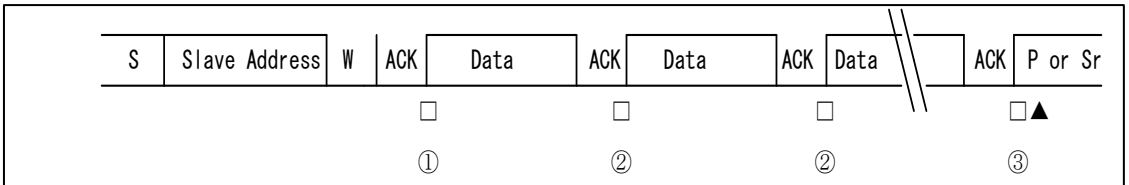
① An interrupt generated because the transmission FIFO is empty

– Write INT = "0" after the transmission data is written to the transmission FIFO

② An interrupt generated by NACK response

– Set MSS = "0" or MSS = "1" and SCC = "1"

Figure 3-21 Master Transmission Interrupts 10 When FIFO Is Disabled (SSR:DMA= 1, IBCR:WSEL= 0, and IBSR:RSA= 0)



S : Start condition

W : Data direction bit(write direction)

P : Stop condition

Sr : Repeated start condition

▲ : Interrupt by CNDE="1"

□ : Interrupt by TBIE="1"

① An interrupt generated by slave address transmission ＋ direction bit transmission＋acknowledgment reception

– the transmission data is written to the TDR register

② An interrupt generated by 1 byte transmission＋acknowledgment reception

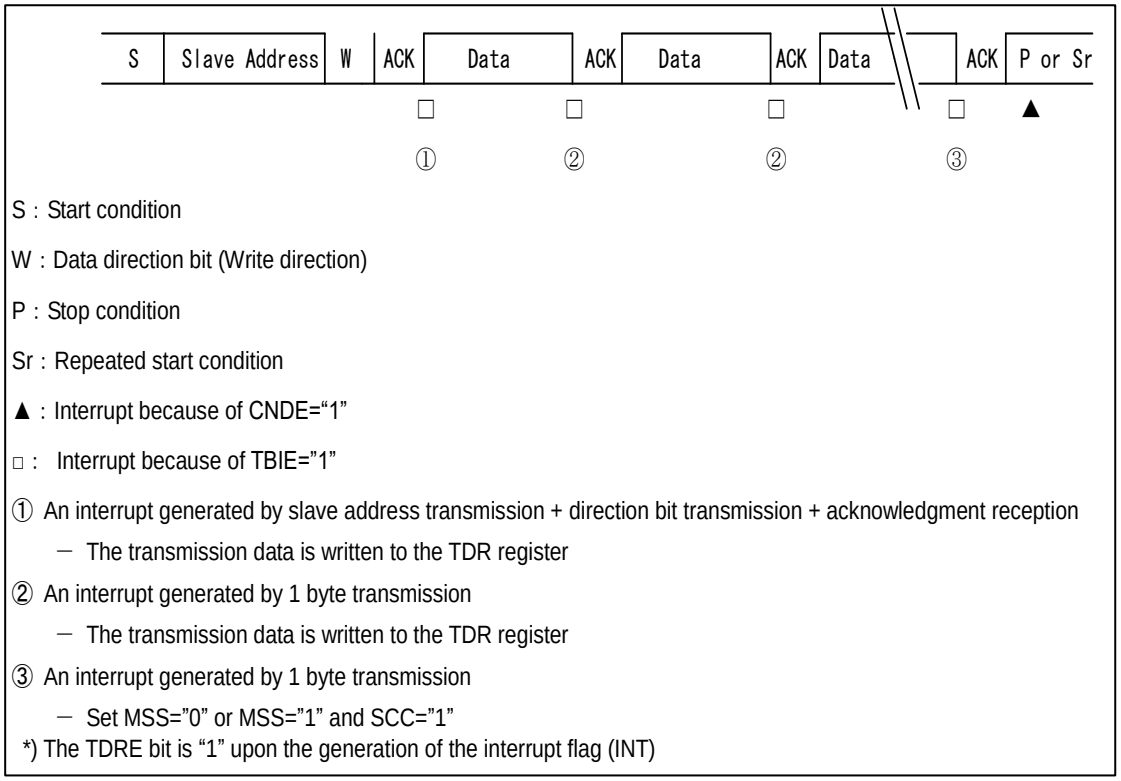
– the transmission data is written to the TDR register

③ An interrupt generated by 1 byte transmission＋acknowledgment reception

– Set MSS="0" or MSS="1" and SCC="1"

*) The TDRE bit is "1" upon the generation of the interrupt flag (TBI)

Figure 3-22 Master Transmission Interrupts 11 When FIFO Is Disabled (SSR:DMA= 1, IBCR:WSEL= 1, IBSR:RSA= 0, and ACK Response)



S : Start condition

W : Data direction bit (Write direction)

P : Stop condition

Sr : Repeated start condition

▲ : Interrupt because of CNDE="1"

□ : Interrupt because of TBIE="1"

① An interrupt generated by slave address transmission + direction bit transmission + acknowledgment reception

— The transmission data is written to the TDR register

② An interrupt generated by 1 byte transmission

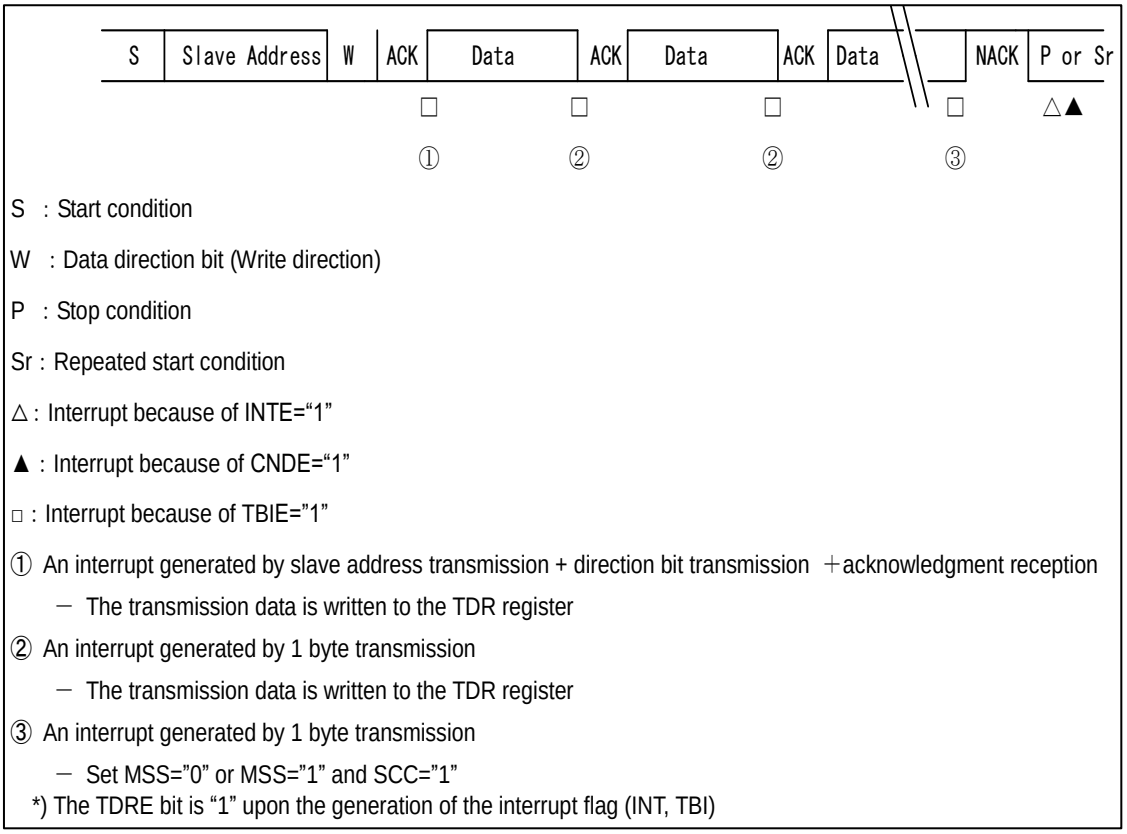
— The transmission data is written to the TDR register

③ An interrupt generated by 1 byte transmission

— Set MSS="0" or MSS="1" and SCC="1"

*) The TDRE bit is "1" upon the generation of the interrupt flag (INT)

Figure 3-23 Master Transmission Interrupts 12 When FIFO Is Disabled (SSR:DMA= 1, IBCR:WSEL= 1, IBSR:RSA= 0, and NACK Response)



S : Start condition

W : Data direction bit (Write direction)

P : Stop condition

Sr : Repeated start condition

△ : Interrupt because of INTE="1"

▲ : Interrupt because of CNDE="1"

□ : Interrupt because of TBIE="1"

① An interrupt generated by slave address transmission + direction bit transmission ＋acknowledgment reception

— The transmission data is written to the TDR register

② An interrupt generated by 1 byte transmission

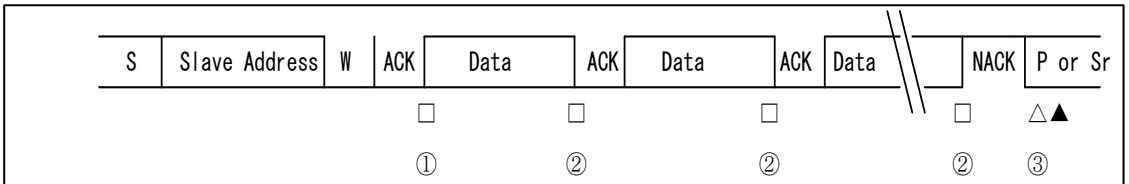
— The transmission data is written to the TDR register

③ An interrupt generated by 1 byte transmission

— Set MSS="0" or MSS="1" and SCC="1"

*) The TDRE bit is "1" upon the generation of the interrupt flag (INT, TBI)

Figure 3-24 Master Transmission Interrupts 13 When FIFO Is Disabled (SSR:DMA= 1, IBCR:WSEL= 1, IBSR:RSA= 0, and Intermediate NACK Response)



S : Start condition

W : Data direction bit (Write direction)

P : Stop condition

Sr : Repeated start condition

△ : Interrupt because of INTE="1"

▲ : Interrupt because of CNDE="1"

□ : Interrupt because of TBIE="1"

① An interrupt generated by slave address transmission + direction bit transmission + acknowledgment reception

– The transmission data is written to the TDR register

② An interrupt generated by 1 byte transmission

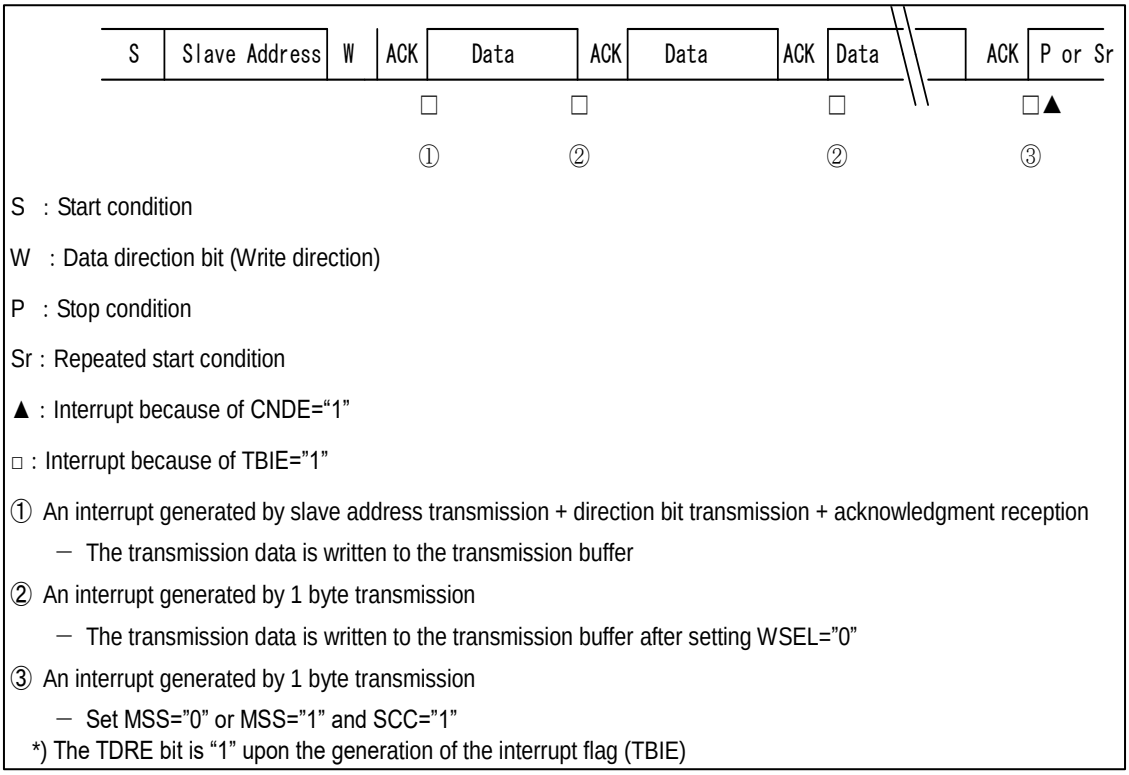
– The transmission data is written to the TDR register

③ An interrupt generated by NACK response

– Set MSS="0" or MSS="1" and SCC="1"

*) The TDRE bit is "1" upon the generation of the interrupt flag (INT, TBI)

Figure 3-25 Master Transmission Interrupts 14 When FIFO Is Disabled (SSR:DMA= 1, IBCR:WSEL= 1->"0", IBSR:RSA= 0, and ACK Response)



S : Start condition

W : Data direction bit (Write direction)

P : Stop condition

Sr : Repeated start condition

▲ : Interrupt because of CNDE="1"

□ : Interrupt because of TBIE="1"

① An interrupt generated by slave address transmission + direction bit transmission + acknowledgment reception

– The transmission data is written to the transmission buffer

② An interrupt generated by 1 byte transmission

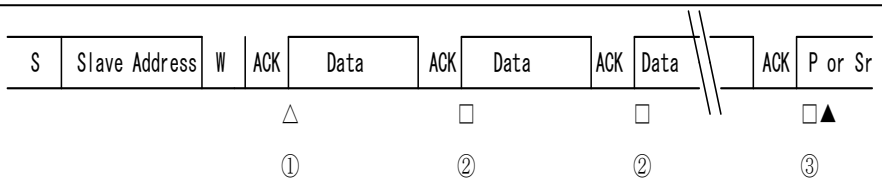
– The transmission data is written to the transmission buffer after setting WSEL="0"

③ An interrupt generated by 1 byte transmission

– Set MSS="0" or MSS="1" and SCC="1"

*) The TDRE bit is "1" upon the generation of the interrupt flag (TBIE)

Figure 3-26 Master Transmission Interrupts 15 When FIFO Is Disabled (SSR:DMA= 1, IBCR:WSEL= 0, and IBSR:RSA= 1)



S : Start condition

W : Data direction bit (Write direction)

P : Stop condition

Sr : Repeated start condition

△ : Interrupt because of INTE="1"

▲ : Interrupt because of CNDE="1"

□ : Interrupt because of TBIE="1"

① An interrupt generated by slave address (reserved address) transmission + direction bit transmission + acknowledgment reception

– Write INT = "0" after the transmission data is written to the TDR register

② An interrupt generated by 1 byte transmission + acknowledgment reception

– The transmission data is written to the TDR register

③ An interrupt generated by 1 byte transmission + acknowledgment reception

– Set MSS="0" or MSS="1" and SCC="1"

*) The TDRE bit is "1" upon the generation of the interrupt flag (INT, TBI)

Figure 3-27 Master Transmission Interrupts 16 When FIFO Is Enabled (SSR:DMA= 1, IBCR:WSEL= 0, IBSR:RSA= 0, and ACK Response)

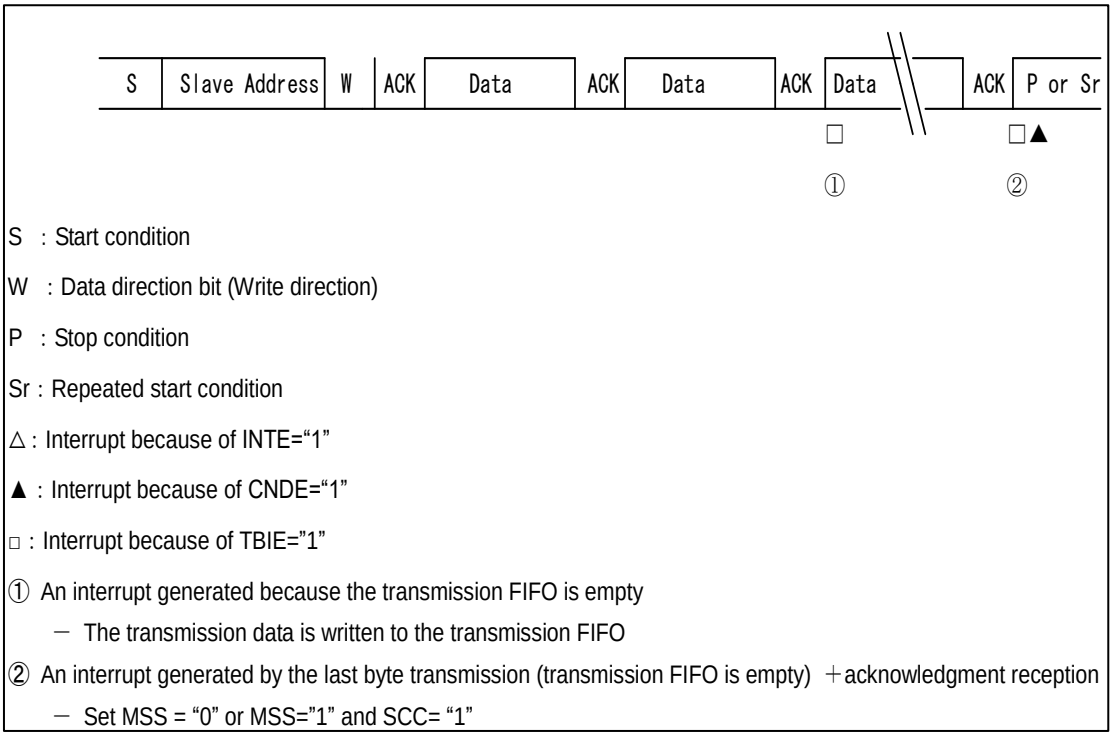


Figure 3-28 Master Transmission Interrupts 17 When FIFO Is Enabled (SSR:DMA= 1, IBCR:WSEL= 1, and IBSR:RSA= 0)

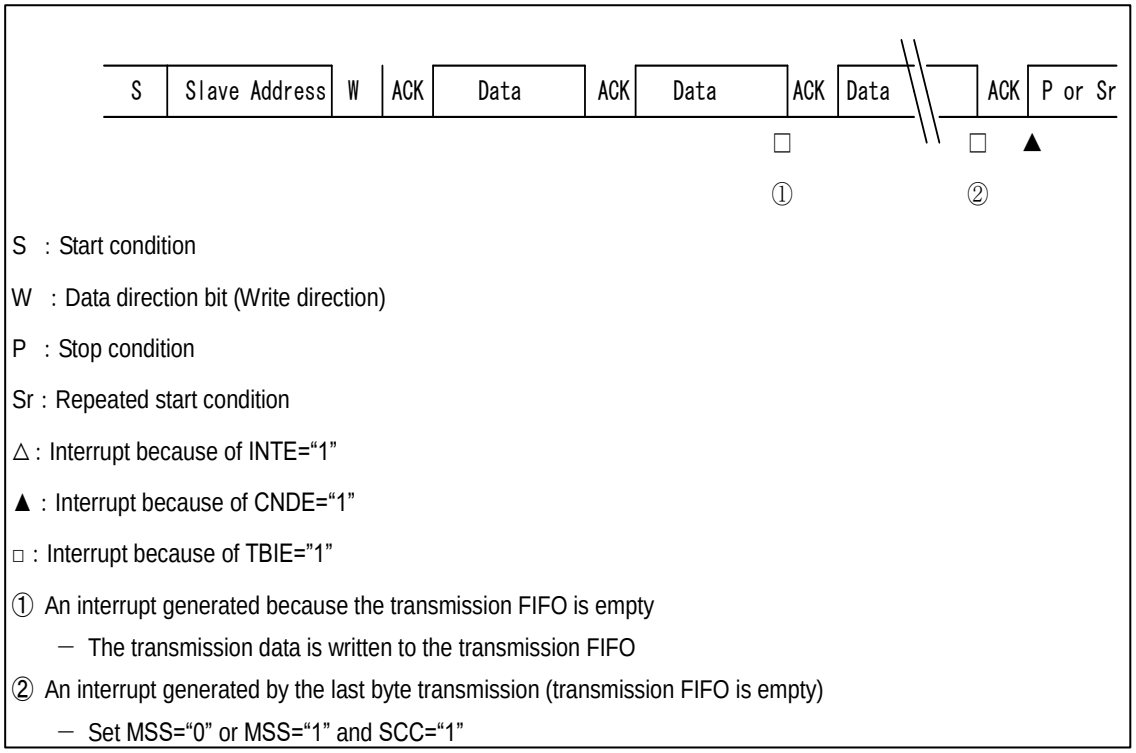
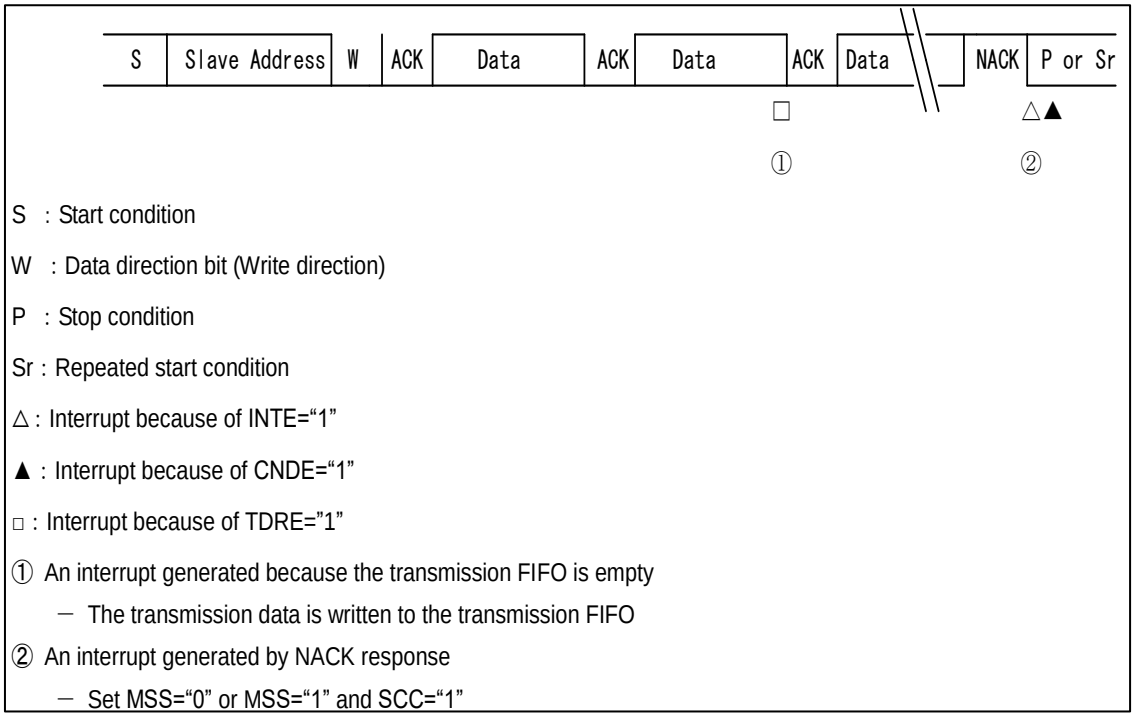


Figure 3-29 Master Transmission Interrupts 18 When FIFO Is Enabled (SSR:DMA= 1, IBCR:WSEL= 1, IBSR:RSA= 0, and NACK Response)



Data Reception by the Master

When DMA Mode is Disabled (SSR:DMA= 0)

If the data direction bit (R/W) is "1", the master receives data from a slave.

If FIFO is disabled, the master operates as follows.

- If the SSR:TDRE bit is "1", a wait is generated each time that 1 byte is received (IBCR:INT = 1 and SSR:RDRF = 1). At this time, the master returns an ACK or NACK response based on the setting of the ACKE bit in the IBCR register as follows. If the IBCR:WSEL bit is "0", it does so before the wait. If the IBCR:WSEL bit is "1", it does so after the wait.
- If the SSR:TDRE bit is "0", the master receives the next data with a response based on the setting of the ACKE bit in the IBCR register. If the response is ACK, no wait is generated (IBCR:INT = 0). If the response is NACK, a wait is generated (IBCR:INT = 1).

If FIFO is enabled, the SSR:RDRF bit is set to "1" when the master receives the same number of bytes as the reception byte count setting. The interrupt flag is set to "1" when the SSR:TDRE bit is "1", to wait for the I^2C bus. At this time, the acknowledgment operates as follows. Note that even if NACK is output, data is stored as reception data in the reception FIFO.

- If IBCR:WSEL = 0, the master returns a NACK response based on the setting of the ACKE bit, when the SSR:TDRE bit is set to "1".
- If IBCR:WSEL = 1, the interrupt flag is set to "1" after the last byte is received, generating a wait. The IBCR:ACKE bit is set and the interrupt flag is cleared to "0" during the wait. Then, the master returns an ACK or NACK response based on the IBCR:ACKE setting.

For details on the wait due to an interrupt, see the following table.

Table 3-5 IBCR:WSEL Bit for Master Data Reception When DMA Mode Is Disabled (SSR:DMA= 0)

WSEL Bit	Operation
0	At the 2nd or subsequent byte, the interrupt flag (IBCR:INT) is set to "1" and the SCL is set to "L" after an acknowledgment when the SSR:TDRE bit is "1". Then, the wait state is entered.
1	At the 2nd or subsequent byte, the interrupt flag (IBCR:INT) is set to "1" and the SCL is set to "L" after the master receives 1-byte data when the SSR:TDRE bit is "1". Then, the wait state is entered.

The following procedures are examples showing how to receive data from a slave.

- When the reception FIFO is disabled
  1. Set the slave address (including the data direction bit) in the TDR register, and write "1" to the IBCR:MSS bit.
  2. Transmit the slave address, and then receive ACK. The interrupt flag (IBCR:INT) is set to "1".
  3. To update the IBCR:WSEL bit, write "1" to the interrupt flag clear bit (IBCRC:INTC) to release the wait for the I^2C bus.
  4. After receiving 1 byte, set the interrupt flag to "1" as follows to wait for the I^2C bus. If IBCR:WSEL= 0, set it after transmitting an acknowledgment. If IBCR:WSEL = 1, set it immediately after the reception of 1 byte. Repeat steps 3. to 4. until the predefined quantity of data has been received.
  5. After receiving the last data, output NACK. Then, set the IBCR:MSS bit to "0" to generate the stop condition, or set the IBCR:SCC bit to "1" to generate the repeated start condition.

- When the transmission and reception FIFOs are enabled
  1. Set the number of bytes to be received in the FBYTE register.
  2. Write the following to the TDR register: the slave address (including the data direction bit) and as much dummy data as the quantity to be received.
  3. Write "1" to the IBCR:MSS bit.
  4. While the SSR:TDRE bit is "0", return an ACK response and continue to receive data. During reception, if the received data reaches the set count in FBYTE, set SSR:RDRF to "1". Read the RDR register when SSR:RDRF is set to "1".
  5. When the SSR:TDRE bit is set to "1", set the interrupt flag to "1" as follows to wait for the I^2C bus. If IBCR:WSEL = 0, set it after NACK output. If IBCR:WSEL = 1, set it immediately after the reception of 1 byte.
  6. If IBCR:WSEL= 1, set the IBCR:ACKE bit to "0". If IBCR:WSEL = 0, this IBCR:ACKE bit setting is not necessary, but either set the IBCR:MSS bit to "0" to generate the stop condition, or set the IBCR:SCC bit to "1" to generate the repeated start condition.

When DMA Mode is Enabled (SSR:DMA= 1)

If the data direction bit (R/W) is "1", the master receives data from a slave.

If FIFO is disabled, the master operates as follows.

- If the SSR:TDRE bit is "1", a wait is generated each time that 1 byte is received (SSR:TBI= 1 and SSR:RDRF= 1). At this time, the master returns an ACK or NACK response based on the setting of the ACKE bit in the IBCR register as follows. If the IBCR:WSEL bit is "0", it does so before the wait. If the IBCR:WSEL bit is "1", it does so after the wait.
- If the SSR:TDRE bit is "0", a wait is generated each time that 1 byte is received (SSR:RDRF = 1). At this time, the master returns an ACK or NACK response based on the setting of the ACKE bit in the IBCR register as follows. If the IBCR:WSEL bit is "1", it does so before the wait. If the IBCR:WSEL bit is "0", it does so after the wait.

If FIFO is enabled, the SSR:RDRF bit is set when the master receives the same number of bytes as the reception byte count setting. When the SSR:TDRE bit is "1", the transmission bus idle flag (SSR:TBI) is set to "1" to wait for the I^2C bus. At this time, the acknowledgment operates as follows. Note that even if NACK is output, data is stored as reception data in the reception FIFO.

- If IBCR:WSEL = 0, the master returns a NACK response based on the setting of the ACKE bit, when the SSR:TDRE bit is set to "1".
- If IBCR:WSEL = 1, a wait is generated (SSR:TBI = 1) after the last byte is received. During the wait, the IBCR:ACKE bit is set, and the transmission bus idle flag (SSR:TBI) is cleared. Then, the master returns an ACK or NACK response based on the IBCR:ACKE setting.

For details on the wait due to an interrupt, see the following table.

Table 3-6 IBCR:WSEL Bit for Master Data Reception When DMA Mode Is Enabled (SSR:DMA= 1)

WSEL Bit	Operation
0	At the 2nd or subsequent byte, the transmission bus idle flag (SSR:TBI) is set to "1" and the SCL is set to "L" after an acknowledgment when the SSR:TDRE bit is "1". Then, a wait state is entered. At the 2nd or subsequent byte, if "1" is set for the reception data full flag (SSR:RDRF) after an acknowledgment when the reception FIFO is not used, the SCL is set to "L". Then, the wait state is entered.
1	At the 2nd or subsequent byte, the interrupt flag (SSR:TBI) is set to "1" and the SCL is set to "L" after the master receives 1-byte data when the SSR:TDRE bit is "1". Then, the wait state is entered. At the 2nd or subsequent byte, after data is received when the reception data full flag (SSR:RDRF) is set to "1" and the reception FIFO is not used, the SCL is set to "L". Then, the wait state is entered.

The following procedures are examples showing how to receive data from a slave.

■ When the reception FIFO is disabled

1. Set the slave address (including the data direction bit) in the TDR register, and write "1" to the IBCR:MSS bit.
2. Transmit the slave address, and then receive ACK. The transmission bus idle flag (SSR:TBI) is set to "1".
3. Write the data to be transmitted, to the TDR register to release the wait for the I^2C bus.
4. After receiving 1 byte, set the transmission bus idle flag (SSR:TBI) and the reception data full flag (SSR:RDRF)*2 to "1" under the following conditions to wait for the I^2C bus:
   - After acknowledgment transmission if IBCR:WSEL = 0
   - Immediately after reception of 1 byte if IBCR:WSEL = 1
5. Update the IBCR:WSEL bit, and read the RDR register. Then, write dummy data to the TDR register, and release the wait.
6. After receiving 1 byte, set the transmission bus idle flag (SSR:TBI) and the reception data full flag (SSR:RDRF)*2 to "1" under the following conditions to wait for the I^2C bus:
   - After acknowledgment transmission if IBCR:WSEL = 0
   - Immediately after reception of 1 byte if IBCR:WSEL = 1

   Repeat steps 5. to 6. until the predefined quantity of data has been received.
7. After receiving the last data, output NACK. Then, set the IBCR:MSS bit to "0" generate the stop condition, or set the IBCR:SCC*1 bit to "1" to generate the repeated start condition.

■ When the transmission and reception FIFOs are enabled

1. Set the number of bytes to be received in the FBYTE register.
2. Write the following to the TDR register: the slave address (including the data direction bit) and as much dummy data as the quantity to be received.
3. If IBCR:WSEL= 0, set NACK with the ACKE bit setting, and write "1" to the IBCR:MSS bit.
4. While the SSR:TDRE bit is "0", return an ACK response and continue to receive data. During reception, if the received data reaches the set count in FBYTE, set SSR:RDRF to "1". Read the RDR register when SSR:RDRF is set to "1".
5. If IBCR:WSEL = 0 when the SSR:TDRE bit is set to "1", set the interrupt flag to "1" after NACK output, and wait for the I^2C bus. If IBCR:WSEL = 1, set the transmission bus idle flag (SSR:TBI) to "1" immediately after receiving 1 byte, and wait for the I^2C bus.

6. If IBCR:WSEL = 1, set the IBCR:ACKE bit to "0". If IBCR:WSEL = 0, this IBCR:ACKE bit setting is not necessary, but either set the IBCR:MSS bit to "0" to generate the stop condition, or set the IBCR:SCC*1 bit to "1" to generate the repeated start condition.

   *1 To issue the repeated start condition when DMA mode is enabled (SSR:DMA= 1), the SSR:TBI bit is "1", and the IBCR:INT bit is "0", perform the following procedure.

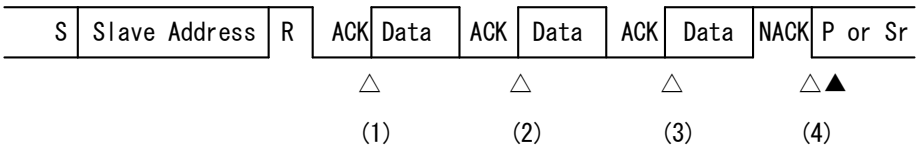
   1. Write "1" to the IBCR:INT bit.
   2. Verify that the IBCR:INT bit is set to "1".
   3. Write the slave address to the TDR.
   4. Set the IBCR:SCC bit to "1".

   *2 Regardless of the IBCR:WSEL setting, the reception data full flag (SSR:RDRF) is set to "1" immediately after the reception of 1 byte. For the 2nd or subsequent byte, there is the following wait for the I^2C bus when the reception data full flag (SSR:RDRF) is set to "1". If IBCR:WSEL = 0, the wait is performed after the transmission of an acknowledgment. If IBCR:WSEL = 1, the wait is performed immediately after the reception of 1 byte.

Notes:

- *Specifying a 7-bit slave address in master mode is prohibited when 7-bit slave address detection is enabled (ISBA:SAEN = 1).*
- *When SSR:TDRE is "0", an acknowledgment is output according to the setting of the IBCR:ACKE bit, and the next processing is performed even if an overrun error occurs.*
- *To change the IBCR register during transmission and reception, do so when the interrupt flag (IBCR:INT) is "1" or when the transmission bus idle flag is "1" (SSR:TBI = 1) and DMA mode is enabled (SSR:DMA = 1).*
- *During master reception with DMA mode disabled (SSR:DMA = 0), the SSR:TDRE bit may be "0" at the time that the interrupt flag (IBCR:INT) is set to "1" after dummy data is written to the TDR register. If so, the interrupt flag (IBCR:INT) remains "0", and the next data is received.*
- *During master reception with DMA mode enabled (SSR:DMA = 1), the SSR:TDRE bit may be "0" at the time that the transmission bus idle flag (SSR:TBI) is set to "1" after dummy data is written to the TDR register. If so, the transmission bus idle flag (SSR:TBI) remains "0", and the next data is received.*
- *If data is received while the reception FIFO is enabled and IBCR:WSEL is "0", the SSR:RDRF bit is set to "1" after the last bit is received. Then, after ACK transmission, the interrupt flag (IBCR:INT) is set to "1".*

Figure 3-30 Master Reception Interrupts 1 When FIFO Is Disabled (SSR:DMA= 0, IBCR:WSEL= 0, and IBSR:RSA= 0)



△ : Interrupt by INTE = "1"

▲ : Interrupt by CNDE = "1"

(1) An interrupt is generated by a slave address transmission, a direction bit transmission and an acknowledgment reception.

- The interrupt is cleared to "0" by writing INT = "0".

(2) An interrupt is generated by a byte reception and an acknowledgment transmission.

- Read the reception data. Then write INT = "0".

(3) An interrupt is generated by a byte reception and an acknowledgment transmission.

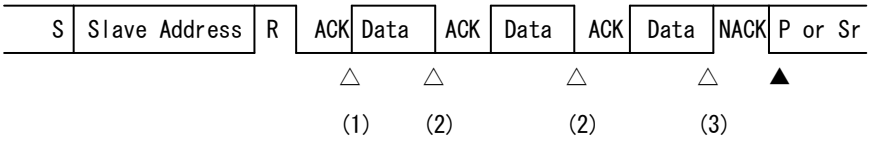
- Read the reception data. Then set ACKE = "0" and write INT = "0".

(4) An interrupt is generated by a byte reception and an acknowledgment transmission.

- Set MSS = "0". Or Set MSS = "1" and SCC = "1"

*: TDRE bit is "1" when the interrupt flag (INT) is set to “1”.

Figure 3-31 Master Reception Interrupts 2 When FIFO Is Disabled (SSR:DMA= 0, IBCR:WSEL= 1, and IBSR:RSA= 0)



△ : Interrupt by INTE="1"

▲ : Interrupt by CNDE=”1”

(1) An interrupt is generated by a slave address transmission, a direction bit transmission and an acknowledgement reception.

- The Interrupt is cleared to "0" by writing INT=”0”.

(2) An interrupt is generated by a byte reception.

- Read the reception data. Then write "0" to INT.

(3) An interrupt is generated by a byte reception.

- Read the reception data. Then set "0" to ACKE. Then set MSS="0". Or set MSS="1" and SCC="1".

*) TDRE bit is "1" when the interrupt flag (INT) is set to “1”.

Figure 3-32 Master Reception Interrupts 3 When FIFO Is Enabled (SSR:DMA= 0, IBCR:WSEL= 0, IBCR:ACKE= 0, and IBSR:RSA= 0)

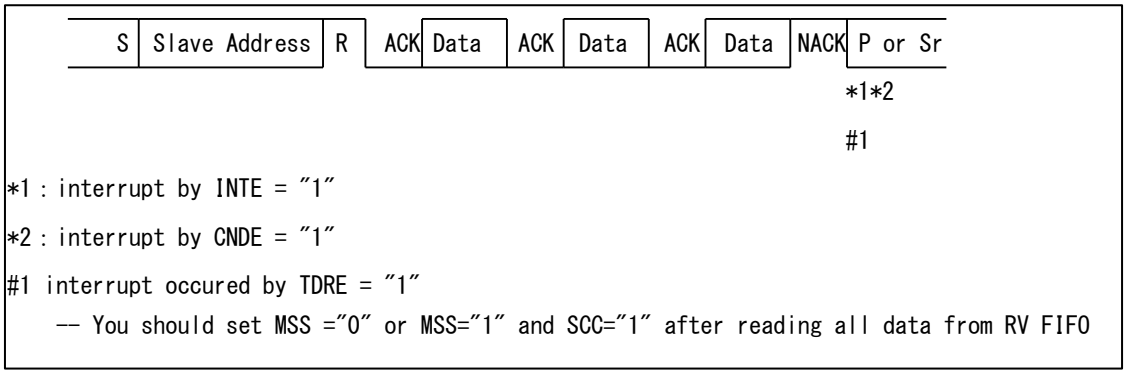


Figure 3-33 Master Reception Interrupts 4 When FIFO Is Enabled (SSR:DMA= 0, IBCR:WSEL= 1, and IBSR:RSA= 0)

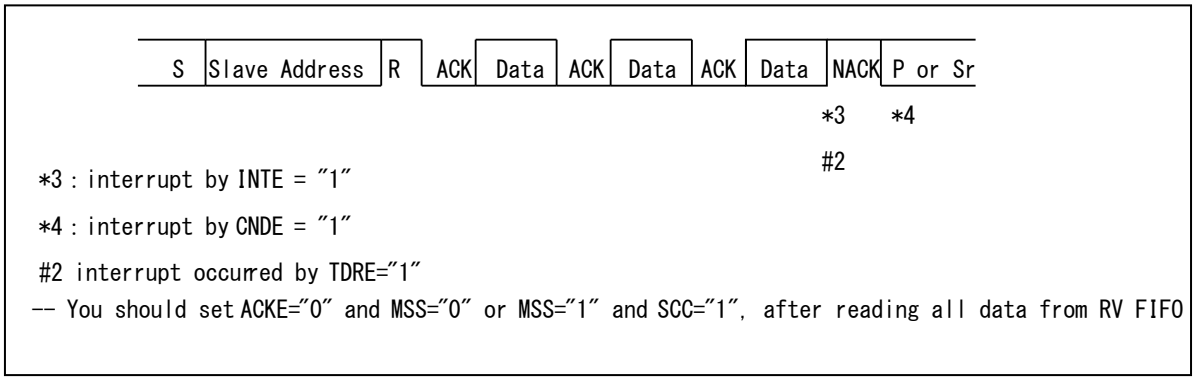
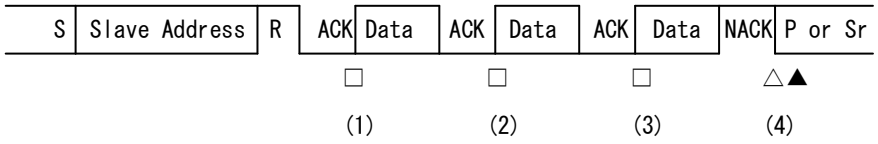


Figure 3-34 Master Reception Interrupts 5 When FIFO Is Disabled (SSR:DMA= 1, IBCR:WSEL= 0, and IBSR:RSA= 0)



△ : Interrupt by INTE="1"
▲ : Interrupt by CNDE="1"
□ : Interrupt by TBIE="1"

(1) An interrupt is generated by a slave address transmission, a direction bit transmission and an acknowledgment reception.
— Write a dummy data to TDR register.

(2) An interrupt is generated by a byte reception and an acknowledgment transmission.
— Read the reception data. Then write a dummy data to TDR register.

(3) An interrupt is generated by a byte reception and an acknowledgment transmission.
— Read the reception data. Then write a dummy data to TDR register and set ACKE="0".

(4) An interrupt is generated by a byte reception and an acknowledgment transmission.
— Set MSS="0". Or set MSS="1" and SCC="1".

*) TDRE bit is "1" when the Interrupt flag(INT or TBI) is set to "1".

Figure 3-35 Master Reception Interrupts 6 When FIFO Is Disabled (SSR:DMA= 1, IBCR:WSEL= 1, and IBSR:RSA= 0)

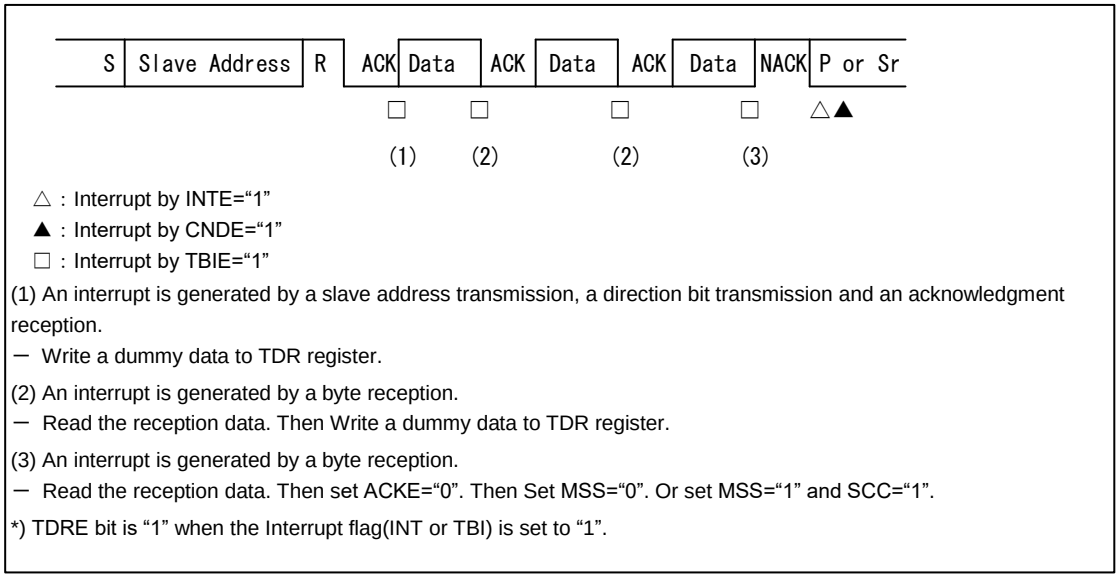


Figure 3-36 Master Reception Interrupts 7 When FIFO Is Enabled (SSR:DMA= 1, IBCR:WSEL= 0, IBCR:ACKE= 0, and IBSR:RSA= 0)

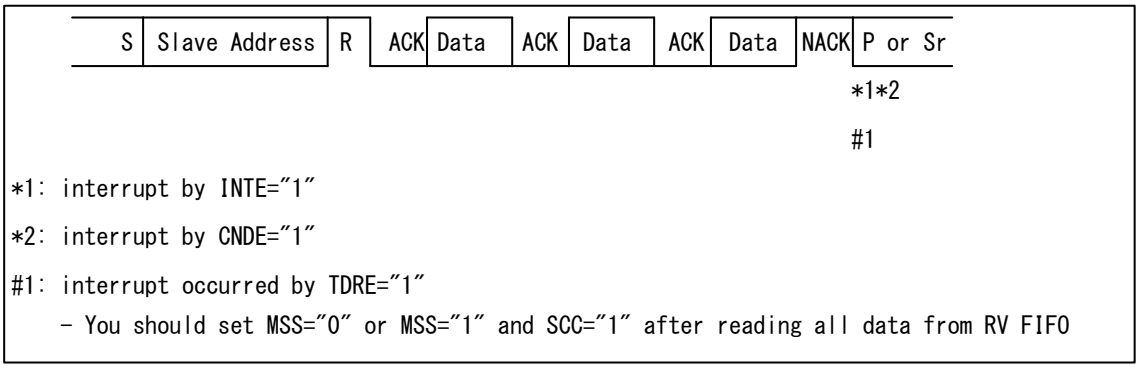
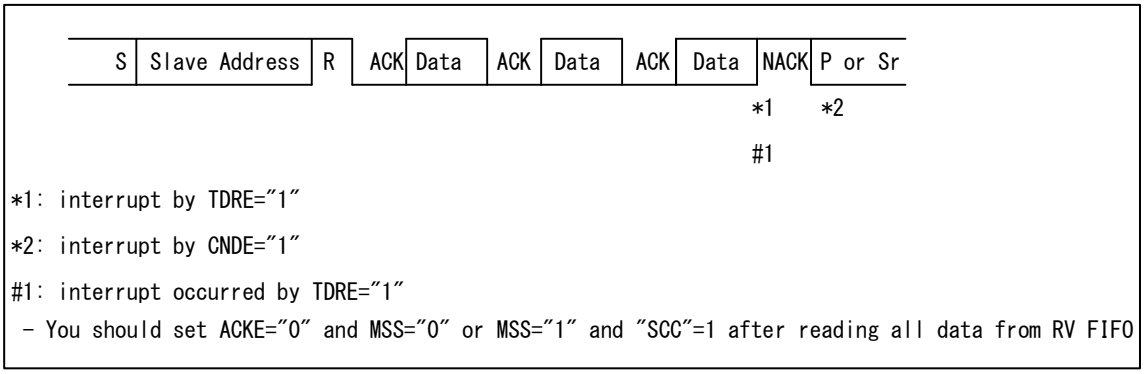


Figure 3-37 Master Reception Interrupts 8 When FIFO Is Enabled (SSR:DMA= 1, IBCR:WSEL= 1, and IBSR:RSA= 0)



Arbitration Lost

Arbitration lost is determined as a case where data from one master conflicts with data from another master and the received data differs from the transmitted data. At this time, the IBCR:MSS bit is set to "0" and the IBSR:AL bit to "1", enabling the master to operate in slave mode.

The IBSR:AL bit can be cleared to "0" under the following conditions.

- "1" is written to the IBCR:MSS bit.
- "0" is written to the IBCR:INT bit.
- "1" is written to the IBSRC:SPCC bit when the IBSR:AL bit is "1" and the IBSR:SPC bit is "1".
- I²C interface operation is disabled (ISMK:EN bit = 0).

If arbitration lost occurs, the interrupt flag (IBCR:INT) is set to "1" according to the IBCR:WSEL setting, and the SCL of the I²C bus is set to "L".

Wait for Master Mode

If both of the following conditions are satisfied, a wait for master mode is performed while the IBSR:BB bit is "1", and the start condition is transmitted when the IBSR:BB bit is set to "0".

- The IBCR:MSS bit is set to "1" when the IBSR:BB bit is "1".
- The device does not operate in slave mode.

To determine whether a wait for master mode is in progress, check the IBCR:MSS and IBCR:ACT bits (the wait state if IBCR:MSS = 1 and IBCR:ACT = 0). If the device operates in slave mode after the IBCR:MSS bit is set to "1", set the IBSR:AL bit to "1", the IBCR:MSS bit to "0", and the IBCR:ACT bit to "1".

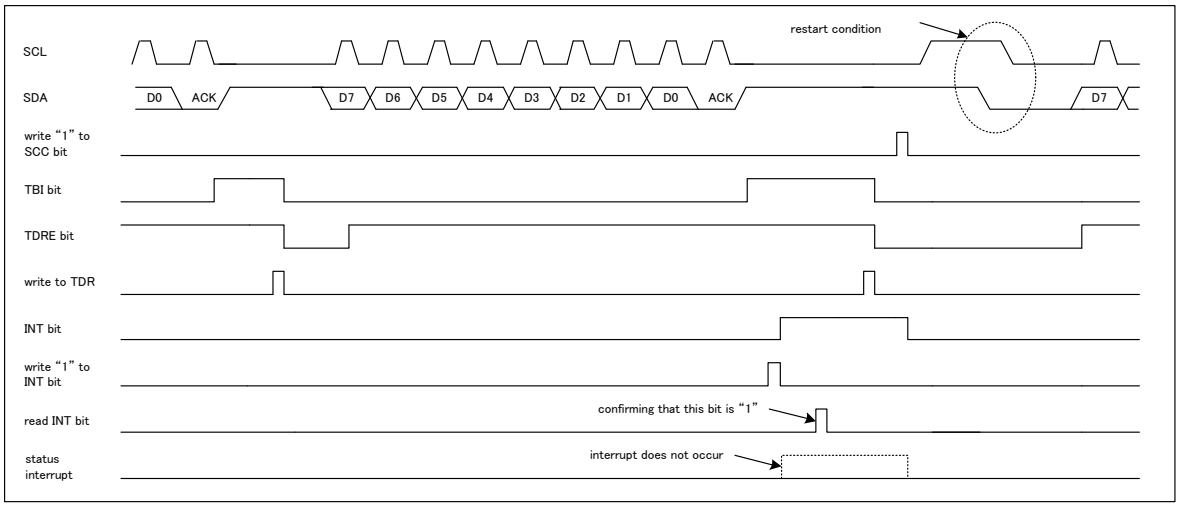
Issuing the Repeated Start Condition When DMA Mode is Enabled (SSM:DMA=1)

If a slave address is written to the TDR register when the transmission bus is idle (SSR:TBI = 1) and the interrupt flag (IBCR:INT) is "0", the transmission operation begins, and the repeated start condition cannot be issued.

Therefore, to issue the repeated start condition when the transmission bus is idle (SSR:TBI = 1) and the interrupt flag (IBCR:INT) is "0", perform the following procedure.

1. Write "1" to the IBCR:INT bit. No status interrupt is generated at this time.
2. Verify that the IBCR:INT bit is set to "1".
3. Write the slave address to the TDR.
4. Issue the repeated start (IBCR:SCC = 1).

Figure 3-38 Issuing the Repeated Start Condition When DMA Mode Is Enabled (SSR:DMA= 1, IBCR:WSEL= 0, IBSR:RSA= 0, and ACK Response)



3.2. Slave Mode

Operation is in slave mode after the start condition or the repeated start condition is detected and an ACK response is returned when the combination of the ISBA register and the ISMK register matches the received address.

Notes:

- *When EIBCR:BEC=0, if 2nd start condition is detected after 1st start condition is detected (during transfer of bit2 to bit9), a bus error is detected (IBCR:BER=1) and reception is stopped.*
- *In this case, after the interrupt flag (IBCR:INT) is cleared, retransmission of the start condition from the master is necessary.*

Slave Address Match Detection

The 1st receiving data after the start condition or the repeated start condition is detected contains 7-bit slave address and the bit that shows the direction of the data transmission. The ISMK register is set the value that masks the slave address of the ISBA register. The bit value "0" means "don't care", and "1" means that it needs that the bit value of address corresponds with the data of received slave address. That is, the bit whose the value is set to "0" in the ISMK register is not compared to the bit of the address.

When SAEN is set to "1", it enables the slave address detection.

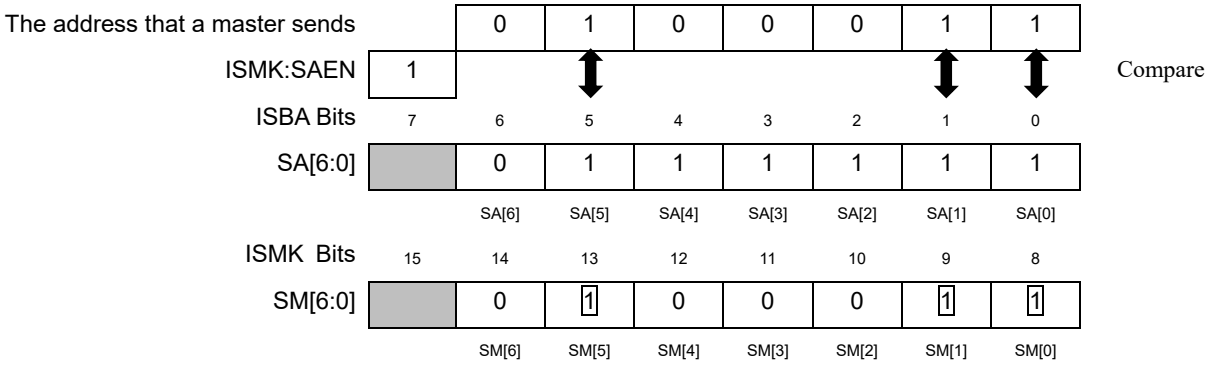
The data as the address sent by a master is compared with some slave address bits(SA[6:0]) that mask bits(SM[6:0]) corresponded to is set to "1".

If they match, ACK is output. If not, ACK is not output.

Example of the address detection

For example, a master send the slave address 0x23.

Figure 3-39 The example of the slave address detection



Just SA5, SA1, SA0 are compared the data as the address sent by the master. Because SM[6], SM[4:2] are set to "0", those bits are "don't care". As a result, the Multi-function serial interface outputs the ACK response.

Table 3-7 Operations Immediately after Acknowledgment Output for a Slave Address

Transmission FIFO	Reception FIFO	Transmission FIFO State	Reception FIFO State	Data Direction Bit (R/W)	Operation Immediately after Acknowledgment	
					Acknowledgment Is ACK	Acknowledgment Is NACK
Disabled	Disabled	-	-	0	If the SSR:TDRE bit is "1", the IBCR:INT bit is set to "1", and a wait is performed. If the SSR:TDRE bit is "0", the IBCR:INT bit remains "0", and no wait is performed.	The IBCR:INT bit remains "0", and no wait is performed.
				1		
Disabled	Enabled	-	Without data	0	The IBCR:INT bit remains "0", and no wait is performed.	The IBCR:INT bit remains "0", and no wait is performed.
			With data		The IBCR:INT bit is set to "1", and a wait is performed.	
			-	1	If the SSR:TDRE bit is "1", the IBCR:INT bit is set to "1", and a wait is performed. If the SSR:TDRE bit is "0", the IBCR:INT bit remains "0", and no wait is performed.	

Transmission FIFO	Reception FIFO	Transmission FIFO State	Reception FIFO State	Data Direction Bit (R/W)	Operation Immediately after Acknowledgment	
					Acknowledgment Is ACK	Acknowledgment Is NACK
Enabled	Disabled	-	-	0	If the SSR:TDRE bit is "1", the IBCR:INT bit is set to "1", and a wait is performed. If the SSR:TDRE bit is "0", the IBCR:INT bit remains "0", and no wait is performed.	The IBCR:INT bit remains "0", and no wait is performed.
				1		
Enabled	Enabled	-	Without data	0	The IBCR:INT bit remains "0", and no wait is performed.	The IBCR:INT bit remains "0", and no wait is performed.
			With data		The IBCR:INT bit is set to "1", and a wait is performed.	
Enabled	Enabled	-	-	1	If the SSR:TDRE bit is "1", the IBCR:INT bit is set to "1", and a wait is performed. If the SSR:TDRE bit is "0", the IBCR:INT bit remains "0", and no wait is performed.	The IBCR:INT bit remains "0", and no wait is performed.

■ Reserved address detection

If the 1st byte matches the reserved address (0b0000xxxx or 0b1111xxxx), the IBCR:INT bit is set to "1" after the 8th bit of data is received, regardless of whether the transmission and reception FIFOs are enabled, to wait for the I^2C bus. After reading the reception data at this time, set the following.

- If you want the device to operate as a slave, set IBCR:ACKE to "1", and check the data direction bit (IBSR:TRX). If the bit indicates the transmission direction, write transmission data to the TDR, and clear the IBCR:INT bit. Then, the device operates as a slave.
- If you do not want the device to operate as a slave, set IBCR:ACKE to "0", and clear the IBCR:INT bit. After acknowledgment output, the device does not operate as a slave.

Data Direction Bit

After receiving an address, the interface receives the data direction bit that determines whether data is transmitted or received. When this bit is "0", it indicates transmission from the master, and the slave receives the data.

Reception by a Slave

A matching slave address and "0" in the data direction bit together indicate reception in slave mode. The following procedures are examples for reception in slave mode.

When DMA Mode is Disabled (SSR:DMA = 0)

- When the reception FIFO is disabled
  1. After ACK transmission, set the interrupt flag (IBCR:INT) to "1" to wait for the I^{2}C bus. An interrupt due to a slave address match is determined from the IBCR:MSS, IBCR:ACT, and IBSR:FBT bits. Write "1" to the IBCR:ACKE bit and "1" to the interrupt flag clear bit (IBCRC:INTC) to release the wait for the I^{2}C bus. (See Table 3-7.)
  2. After transmitting 1-byte data, set the interrupt flag (IBCR:INT) to "1" according to the IBCR:WSEL setting to wait for the I^{2}C bus.
  3. Read the received data from the RDR register, and set the IBCR:ACKE bit. After that, write "1" to the interrupt flag clear bit (IBCRC:INTC) to release the wait for the I^{2}C bus.
  4. Repeat steps 2. to 3. until the stop condition or the repeated start condition is detected.

- When the reception FIFO is enabled
  1. When NACK is detected or the reception FIFO is full, the interrupt flag (IBCR:INT) is set to "1" to wait for the I^{2}C bus. If the stop condition or the repeated start condition is detected, the IBSR:SPC bit or the IBSR:RSC bit is set to "1", and the interrupt flag (IBCR:INT) is not set to "1" (no wait for the I^{2}C bus). The reception FIFO sets the SSR:RDRF bit to "1" when the received data count matches the FBYTE register setting value. If the SMR:RIE bit is "1" at this time, a reception interrupt is generated.
     1. If the interrupt flag (IBCR:INT) was set to "1", read the received data from the RDR register. After reading all the data, write "1" to the interrupt flag clear bit (IBCRC:INTC) to release the wait for the I2C bus. If the stop condition or the repeated start condition was detected, read all the received data from the RDR register, and clear the IBSR:SPC bit or the IBSR:RSC bit to "0".

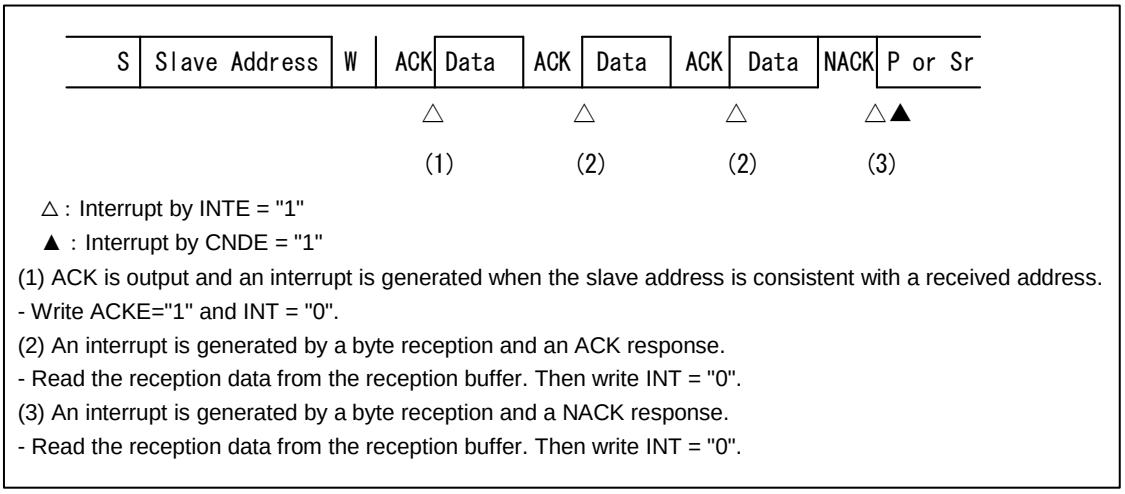
When DMA Mode is Enabled (SSR:DMA=1)

- When the reception FIFO is disabled
  1. After ACK transmission, set the interrupt flag (IBCR:INT) to "1" to wait for the I^{2}C bus. An interrupt due to a slave address match is determined from the IBCR:MSS, IBCR:ACT, and IBSR:FBT bits. Write "1" to the IBCR:ACKE bit and "1" to the interrupt flag clear bit (IBCRC:INTC) to release the wait for the I^{2}C bus. (See Table 3-7.)
  2. After receiving 1-byte data or immediately after receiving 1 byte, set the reception data full flag (SSR:RDRF) to "1". There is the following wait for the I^{2}C bus when the reception data full flag (SSR:RDRF) is set to "1". If IBCR:WSEL = 0, the wait is performed after the transmission of an acknowledgment. If IBCR:WSEL = 1, the wait is performed immediately after the reception of 1 byte.
  3. After setting the IBCR:ACKE bit, read the received data from the RDR register. Then, clear the reception data full flag (SSR:RDRF) to "0" to release the wait for the I^{2}C bus.
  4. Repeat steps 2. to 3. until the stop condition or the repeated start condition is detected.

■ When the reception FIFO is enabled

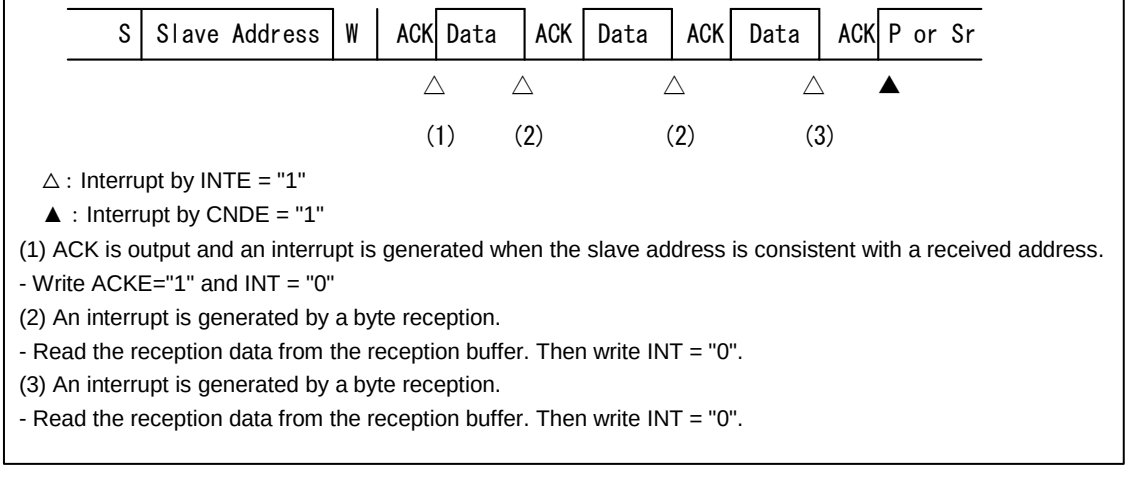
1. The detection of NACK sets the interrupt flag (IBCR:INT) to "1" to wait for the I²C bus. Also, when the reception FIFO is full, wait for the I²C bus. If the stop condition or the repeated start condition is detected, the IBSR:SPC bit or the IBSR:RSC bit is set to "1", and the interrupt flag (IBCR:INT) is not set to "1" (no wait for the I²C bus). The reception FIFO sets the SSR:RDRF bit to "1" when the received data count matches the FBYTE register setting value. If the SMR:RIE bit is "1" at this time, a reception interrupt is generated.
2. If the interrupt flag (IBCR:INT) was set to "1", read the received data from the RDR register. After reading all the data, write "1" to the interrupt flag clear bit (IBCRC:INTC) to release the wait for the I²C bus. If the reception FIFO is full, read the received data from the RDR register at least once to release the wait for the I²C bus. If the stop condition or the repeated start condition was detected, read all the received data from the RDR register, and clear the IBSR:SPC bit or the IBSR:RSC bit to "0".

Figure 3-40 Slave Reception Interrupts 1 When FIFO Is Disabled (SSR:DMA= 0, IBCR:WSEL= 0, and IBSR:RSA= 0)



△ : Interrupt by INTE = "1"
▲ : Interrupt by CNDE = "1"
(1) ACK is output and an interrupt is generated when the slave address is consistent with a received address.
- Write ACKE="1" and INT = "0".
(2) An interrupt is generated by a byte reception and an ACK response.
- Read the reception data from the reception buffer. Then write INT = "0".
(3) An interrupt is generated by a byte reception and a NACK response.
- Read the reception data from the reception buffer. Then write INT = "0".

Figure 3-41 Slave Reception Interrupts 2 When FIFO Is Disabled (SSR:DMA= 0, IBCR:WSEL= 1, and IBSR:RSA= 0)



△ : Interrupt by INTE = "1"
▲ : Interrupt by CNDE = "1"
(1) ACK is output and an interrupt is generated when the slave address is consistent with a received address.
- Write ACKE="1" and INT = "0"
(2) An interrupt is generated by a byte reception.
- Read the reception data from the reception buffer. Then write INT = "0".
(3) An interrupt is generated by a byte reception.
- Read the reception data from the reception buffer. Then write INT = "0".

Figure 3-42 Slave Reception Interrupts 3 When FIFO Is Disabled (SSR:DMA= 0, IBCR:WSEL= 1, and IBSR:RSA= 0)

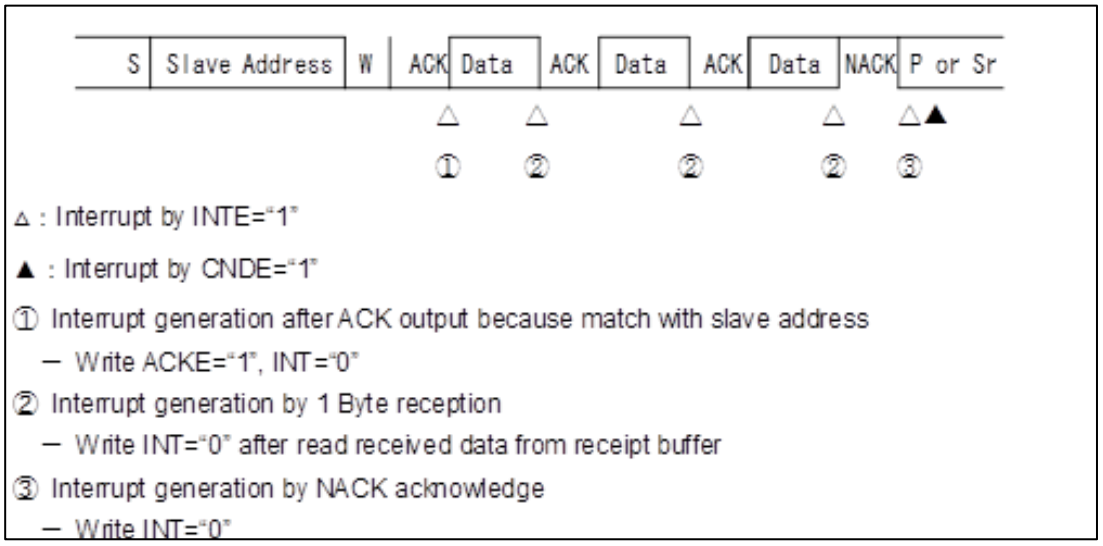


Figure 3-43 Slave Reception Interrupts 4 When the Reception FIFO Is Enabled (SSR:DMA= 0 and IBSR:RSA= 0)

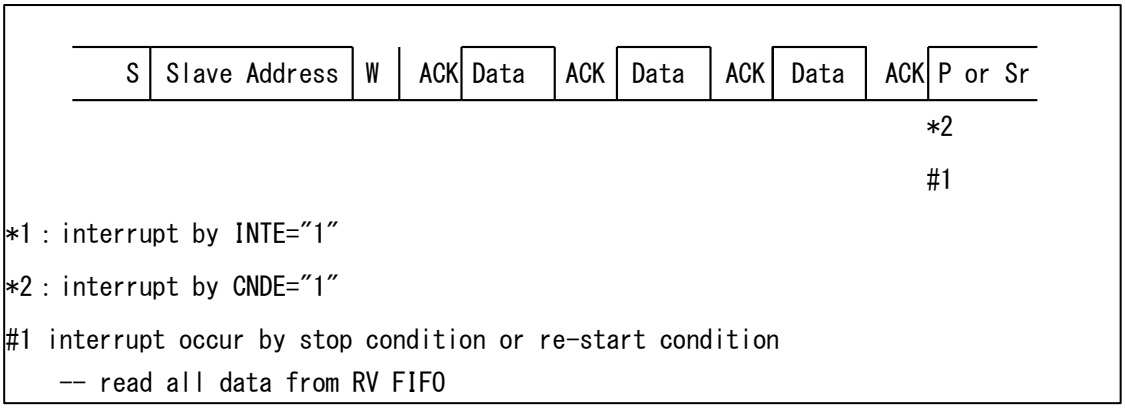


Figure 3-44 Slave Reception Interrupts 5 When the Reception FIFO Is Enabled (SSR:DMA= 0 and IBSR:RSA= 0)

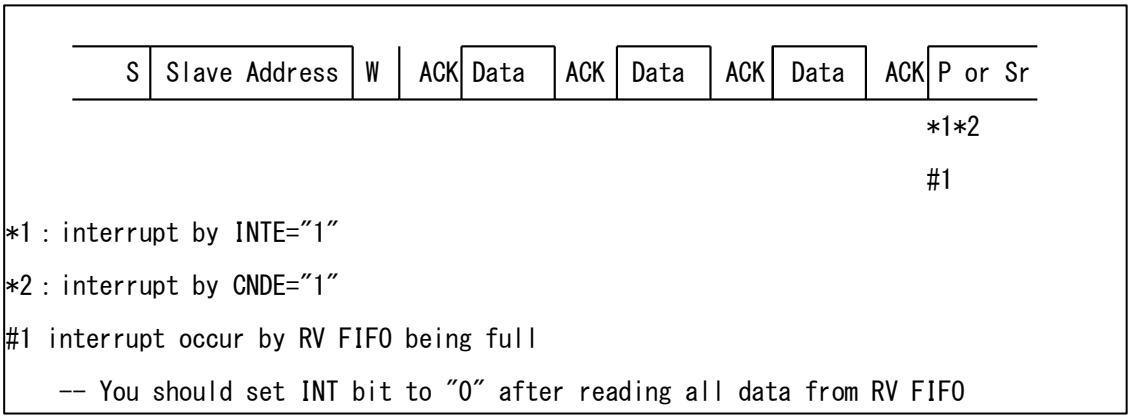
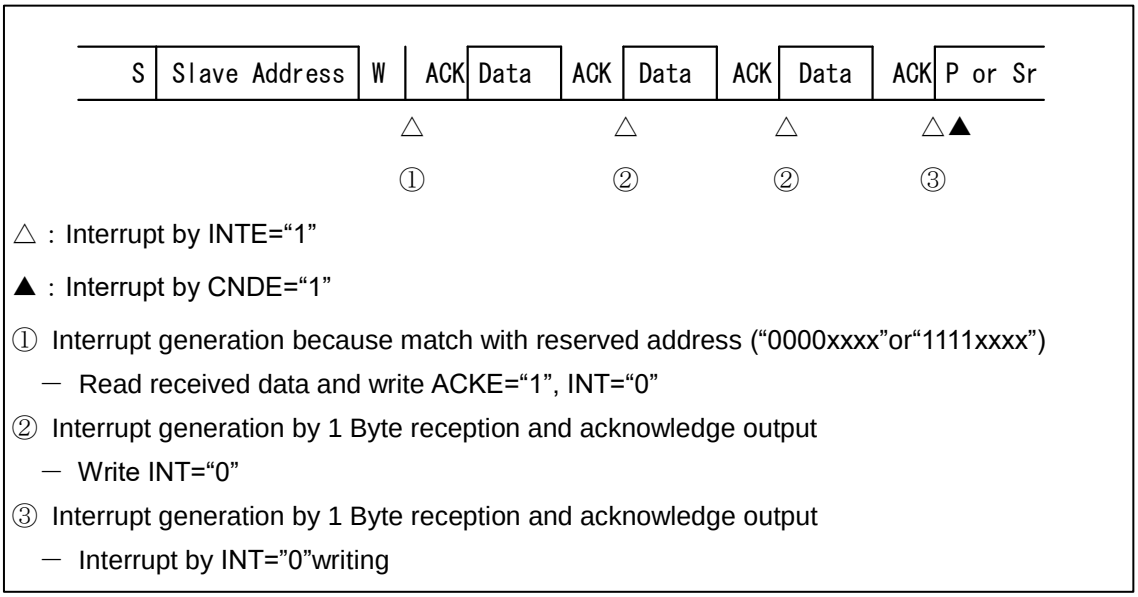


Figure 3-45 Slave Reception Interrupts 6 When FIFO Is Disabled (SSR:DMA= 0, IBCR:WSEL= 0, and IBSR:RSA= 1)



△ : Interrupt by INTE="1"

▲ : Interrupt by CNDE="1"

① Interrupt generation because match with reserved address ("0000xxxx"or"1111xxxx")

— Read received data and write ACKE="1", INT="0"

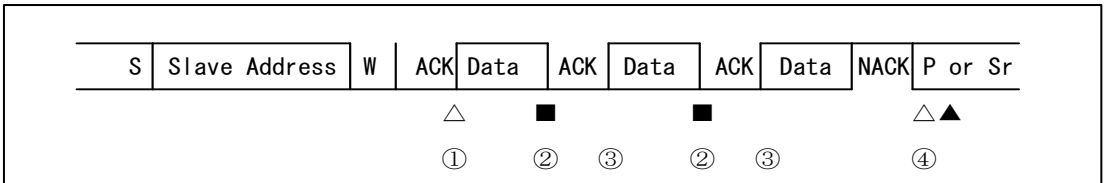
② Interrupt generation by 1 Byte reception and acknowledge output

— Write INT="0"

③ Interrupt generation by 1 Byte reception and acknowledge output

— Interrupt by INT="0"writing

Figure 3-46 Slave Reception Interrupts 7 When FIFO Is Disabled (SSR:DMA= 1, IBCR:WSEL= 0, and IBSR:RSA= 0)



△ : Interrupt by INTE="1"

▲ : Interrupt by CNDE="1"

■ : Interrupt by RIE="1"

① Interrupt generation after ACK output because match with slave address

— Write ACKE="1", INT="0"

② Interrupt generation by 1 Byte reception (No wait for I^2C bus)

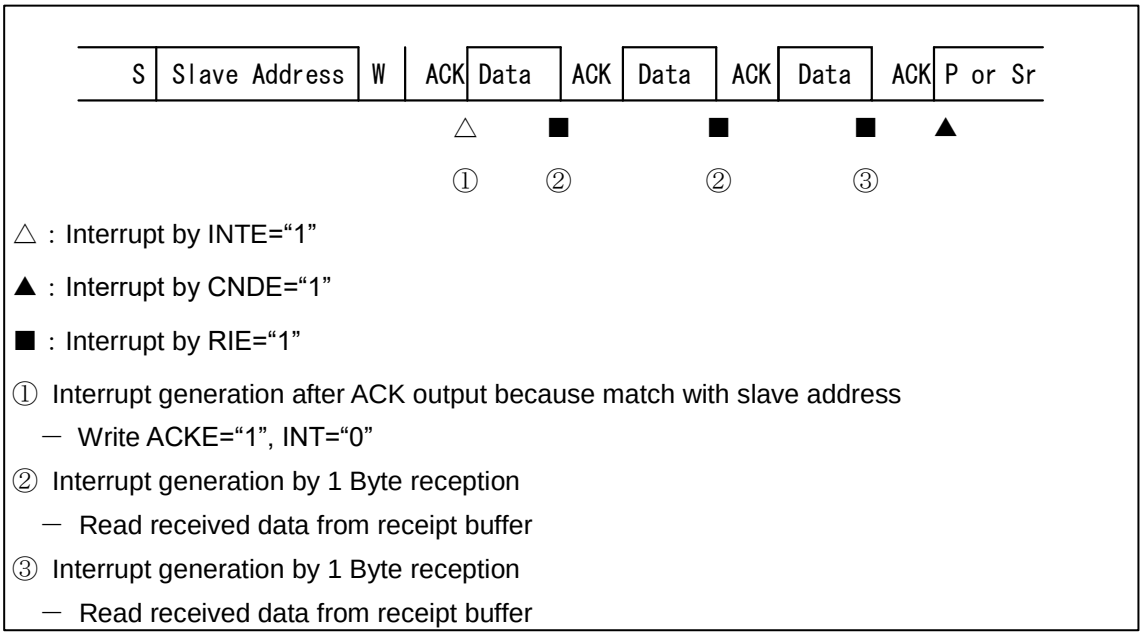
— Read received data from receipt buffer

③ I^2C bus wait by ACK acknowledge

④ Interrupt generation by 1 Byte receive and NACK acknowledge

—Read received data from receipt buffer and write INT="0"

Figure 3-47 Slave Reception Interrupts 8 When FIFO Is Disabled (SSR:DMA= 1, IBCR:WSEL= 1, and IBSR:RSA= 0)



△ : Interrupt by INTE="1"

▲ : Interrupt by CNDE="1"

■ : Interrupt by RIE="1"

① Interrupt generation after ACK output because match with slave address

— Write ACKE="1", INT="0"

② Interrupt generation by 1 Byte reception

— Read received data from receipt buffer

③ Interrupt generation by 1 Byte reception

— Read received data from receipt buffer

Figure 3-48 Slave Reception Interrupts 9 When FIFO Is Disabled (SSR:DMA= 1, IBCR:WSEL= 1, and IBSR:RSA= 0)



△ : Interrupt by INTE="1"

▲ : Interrupt by CNDE="1"

■ : Interrupt by RIE="1"

① Interrupt generation after ACK output because match with slave address

— Write ACKE="1", INT="0"

② Interrupt generation by 1 Byte reception

— Read received data from receipt buffer

③ Interrupt generation by NACK acknowledge

— Write INT="0"

Figure 3-49 Slave Reception Interrupts 10 When the Reception FIFO Is Enabled (SSR:DMA= 1 and IBSR:RSA= 0)

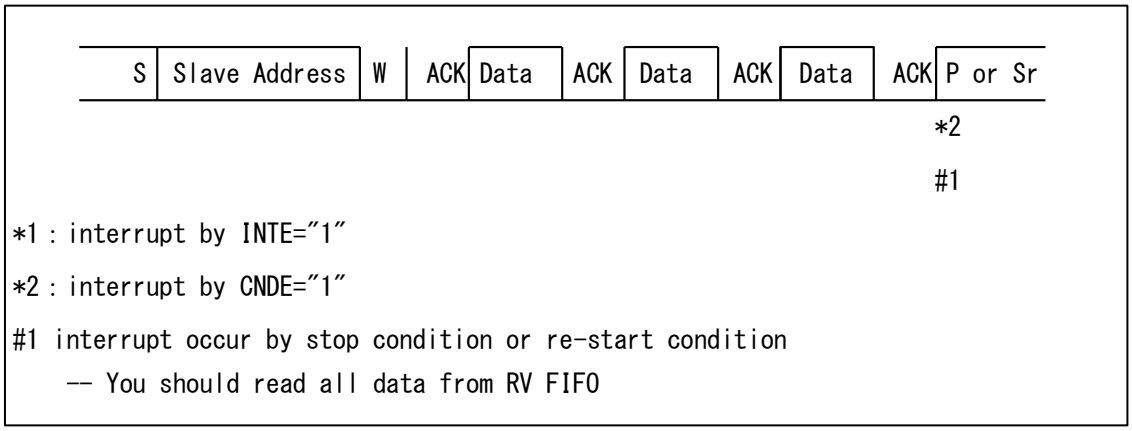


Figure 3-50 Slave Reception Interrupts 11 When the Reception FIFO Is Enabled (SSR:DMA= 1 and IBSR:RSA= 0)

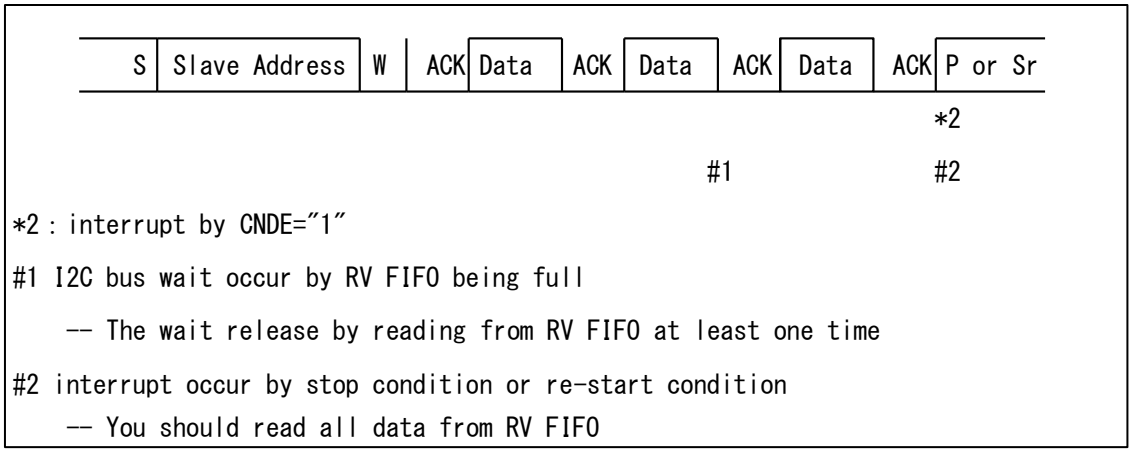
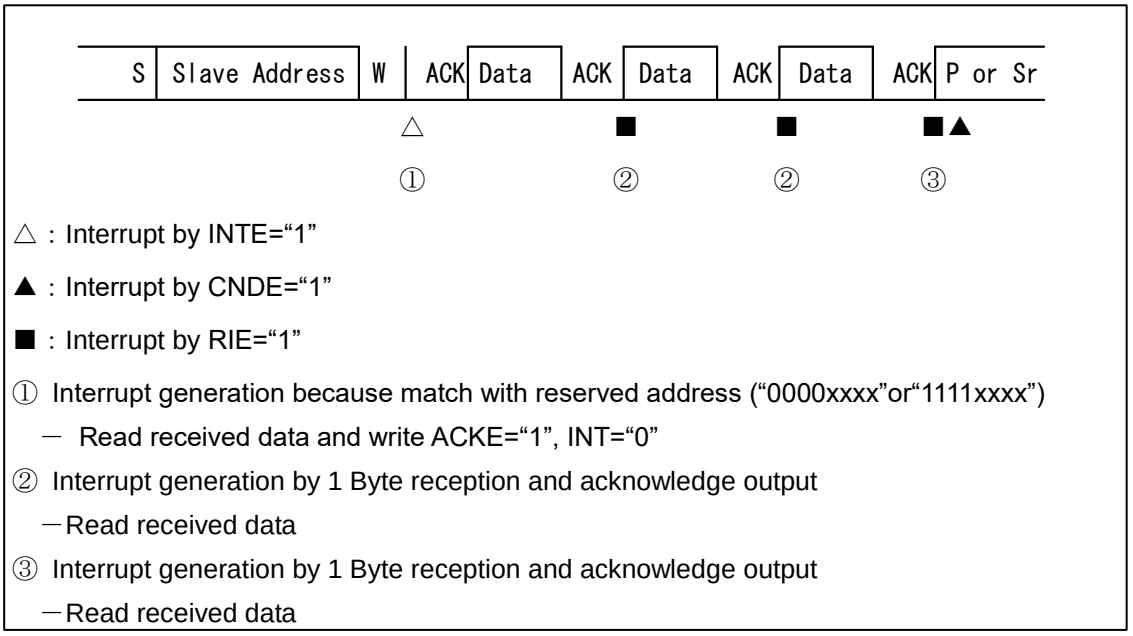


Figure 3-51 Slave Reception Interrupts 12 When FIFO Is Disabled (SSR:DMA= 1, IBCR:WSEL= 0, and IBSR:RSA= 1)



Transmission by a Slave

A matching slave address and "1" in the data direction bit together indicate transmission by a slave. If FIFO is disabled, the interrupt flag (IBCR:INT) is set to "1" according to the IBCR:WSEL setting, to generate a wait after the transmission of 1 byte or after an acknowledgment response. (See Table 3-7.)

You can verify acknowledgment output from the master by checking the IBSR:RACK bit. A NACK response from the master indicates that the master could not receive data correctly or that data reception ended. If NACK is detected when IBCR:WSEL = 1, an interrupt is generated and a wait is performed.

3.3. Bus Error

The detection of the stop or (repeated) start condition on the I^2C bus during data transmission or reception is handled as a bus error.

Bus Error Occurrence Conditions

For a bus error, the IBCR:BER bit is set to "1" under the following conditions.

- The (repeated) start or stop condition is detected during transfer of the 1st byte.
- The (repeated) start or stop condition is detected in the 2nd to 9th (acknowledgment) bits of data.

Bus Error Operation

When EIBCR:BEC=0

Check the IBCR:BER bit when the interrupt flag (IBCR:INT) is set to "1" according to the status. If the IBCR:BER bit is "1", take action against the error. The IBCR:BER bit is cleared by writing "1" to the interrupt flag clear bit (IBCRC:INTC).

If a bus error occurs, the IBCR:INT bit is set to "1", but the SCL of the I^2C bus is not set to "L". Then, the wait state is entered.

When EIBCR:BEC=1

Check the IBCR:BER bit when the interrupt flag (IBSR:SPC or IBSR:RSC) is set to "1" according to the status. If the IBCR:BER bit is "1", take action against the error. The IBCR:BER bit is cleared by the following operations:

- Writing "1" to the IBCRC:INTC bit when IBCR:INT = 1
- Writing "1" to the IBSRC:SPCC bit when IBSR:SPC = 1
- Writing "1" to the IBSRC:RSCC bit when IBSR:RSC = 1

4. Serial Timer Operation

The serial timer can be used for timer functions.

Serial Timer Operation

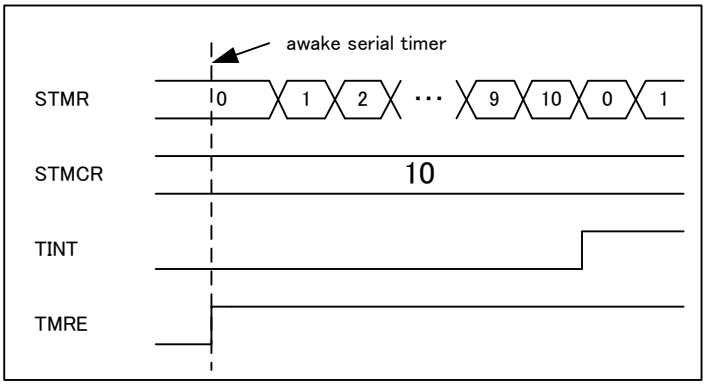
Activating the Serial Timer

There are two ways to activate the serial timer: setting the serial timer enable bit (SACSR:TMRE) to "1", and activation by external trigger.

- Activation by the serial timer enable bit (SACSR:TMRE)

  If the external trigger enable bit (SACSR:TRGE) is set to "0", the serial timer is activated after the serial timer enable bit (SACSR:TMRE) is set to "1". Then, the serial timer register (STMR) starts counting from 0.

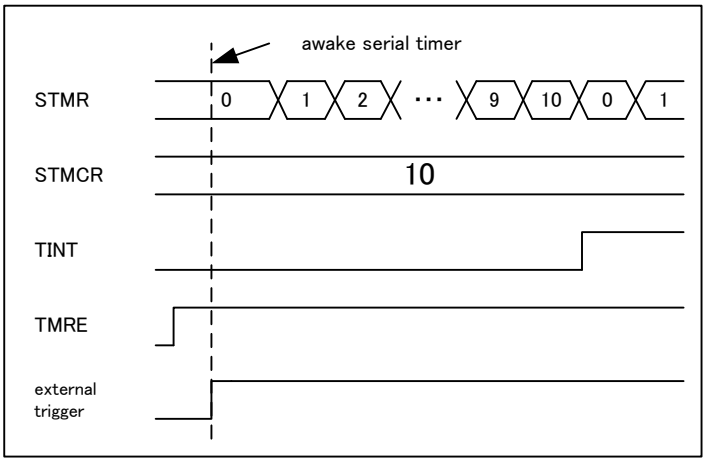
Figure 4-1 Activation by the Serial Timer Enable Bit (STMCR=10, SACSR:TRGE=0)



- Activation by external trigger

  If the external trigger enable bit (SACSR:TRGE) is set to "1", the serial timer is activated after the edge of the set external trigger in the trigger selection bits (TRG1,0) is detected when the serial timer enable bit (SACSR:TMRE) is "1". Then, the serial timer register (STMR) starts counting from 0.

Figure 4-2 Activation by External Trigger (STMCR=10, SACSR:TRGE=1, TRG1=0, TRG0=0)



Note:

- *If the external trigger enable bit (SACSR:TRGE) is "1" and the serial timer enable bit (SACSR:TMRE) is "0", even the detection of the edge of the external trigger that is set in the trigger selection bits (SACSR:TRG1,0) does not start the operation of the serial timer.*

Stopping the Serial Timer

The serial timer stops when the serial timer enable bit (SACSR:TMRE) is set to "0". The value of the serial timer register (STMR) at this time is retained.

Timer Operation

The serial timer operates as the timer.

If the serial timer register (STMR) matches the serial timer comparison register (STMCR), the timer interrupt flag (SACSR:TINT) is set to "1", and the serial timer register (STMR) is reset to 0.

Figure 4-3 Timer Operation (STMCR=10)

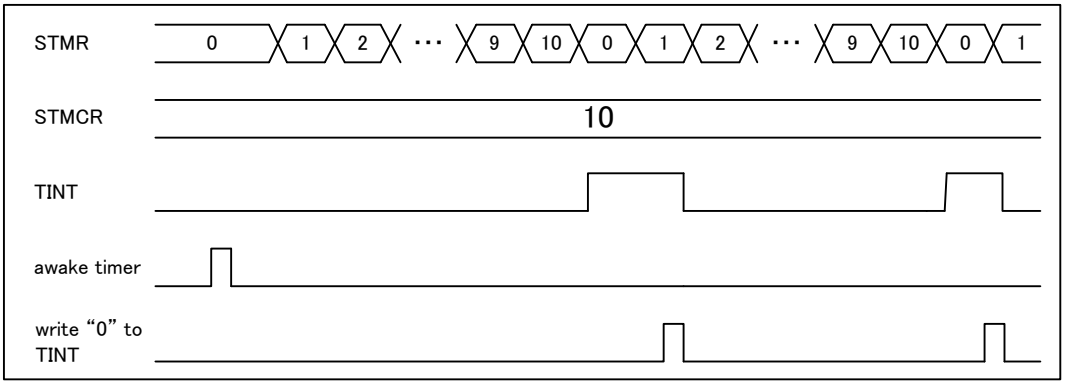


Figure 4-4 Serial Timer Initialization Flowchart

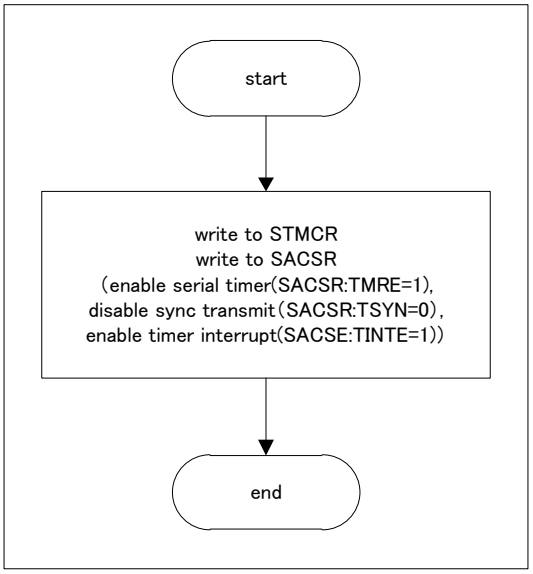
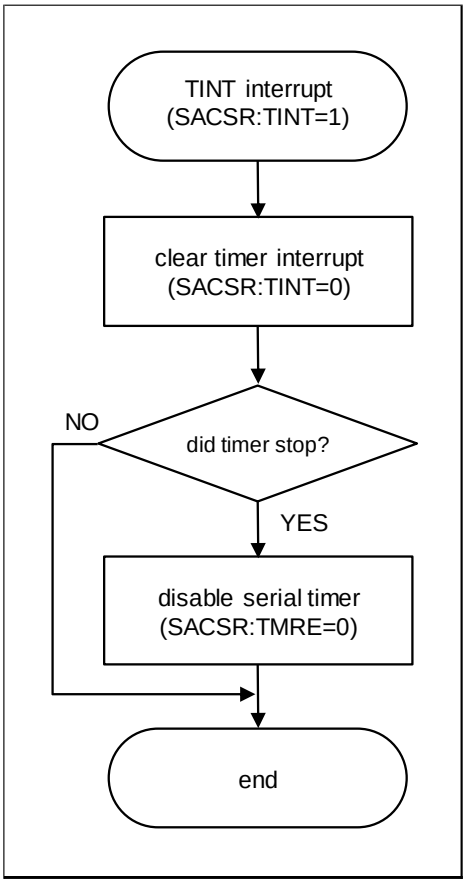


Figure 4-5 Serial Timer Interrupt Processing Flowchart



Note:

– *If the setting of the timer comparison register (STMCR) is 0x0000, the timer is operating, and the timer operation clock divider value (SACSR:TDIV) is set to 0b0000, the timer interrupt flag (SACSR:TINT) is fixed at "1".*

5. Dedicated Baud Rate Generator

A dedicated baud rate generator sets the frequency of the serial clock.

Baud Rate Selection

Baud Rate Obtained by Dividing the Internal Clock by a Dedicated Baud Rate Generator (Reload Counter)

There are two internal reload counters, with each corresponding to the transmission or reception serial clock. A baud rate can be selected with the setting of a 15-bit reload value in baud rate generator registers 1 and 0 (BGR1, BGR0).

The reload counters divide the internal clock according to the set value.

Baud Rate Calculation

The two 15-bit reload counters are set by baud rate generator registers 1 and 0 (BGR1, BGR0).
The following shows the baud rate calculation formulas.

(1) Reload value

```
V = Phy / b - 1

V : reload value  b : baud rate   Phy :  frequency of bus clocl or that of external clock

If baud rate is affected by the rising time of I2C bus, you should make a cut-and-try to
the reload value.
```

(2) Calculation example

```
When bus clock is 16MHz, baud rate is 400k bps, the reload value can be
calculated as follows.
                reload value :
                    V = (16 x 1000000)/400000- 1 = 39
                          then baud rate will be
                    b = (16 x 1000000)/(39  + 1) = 400k bps
```

Notes:

- *Use 16-bit access when writing values to baud rate generator registers 1 and 0 (BGR1, BGR0).*
- *Set the baud rate generator registers when the ISMK:EN bit in the ISMK register is "0".*
- *In operation mode 4 (I^2C mode), use the bus clock at 8 MHz or more. A baud rate generator setting exceeding 400 kbps is prohibited in this mode.*
- *Setting the reload value to "0" stops the reload counter.*
- *When considering the allowable baud rate range, also consider the impact of the jitter of the input clock to the macro.*

Examples of Reload Values and Baud Rate Settings for Each Bus Clock Frequency

The following table shows examples of reload values and baud rate settings.

Table 5-1 Examples of Reload Values and Baud Rate Settings

Baud Rate [bps]	8 MHz	10 MHz	16 MHz	20 MHz	24 MHz	32 MHz
	Value	Value	Value	Value	Value	Value
400000	19	24	39	49	59	79
200000	39	49	79	99	119	159
100000	79	99	159	199	239	319

Baud Rate [bps]	40 MHz	48 MHz	72 MHz	80 MHz
	Value	Value	Value	Value
400000	99	119	179	199
200000	199	239	359	399
100000	399	479	719	799

These numerical values assume that the SCL rising of the I^2C bus is 0 s. If the SCL rising of the I^2C bus is slower, the resulting baud rates are slower than the above numerical values.

Reload Counter Function

The reload counters consist of 15-bit registers for reload values, and they generate transmission/reception clocks from internal clocks. Also, the count value of the transmission reload counter can be read from the baud rate generator registers (BGR1, BGR0).

Start of Counting

A reload counter starts counting when a reload value is written to a baud rate generator register (BGR1, BGR0).

6. Block Transfer

This section explains the block transfer operation.

Block Transfer Setting

The setting of RXBLKEN= 1 in the extended control register (ECR) enables reception block transfer. The setting of TXBLKEN= 1 in the ECR enables transmission block transfer.

The block size is set in the transmission block size register or FBYTE register(reception side).

The threshold values used to output block transfer requests are set in the FBYTE register for reception and in the FTICR register for transmission. A reception block transfer request is output when the data count in the reception FIFO exceeds the FBYTE register setting value. A transmission block transfer request is output when the available capacity of the transmission FIFO exceeds the FTICR register setting value.

6.1. RW Access Mode

This section explains operation in RW access mode for block transfer.

Reception Block Transfer

Reception block transfer operates as follows.

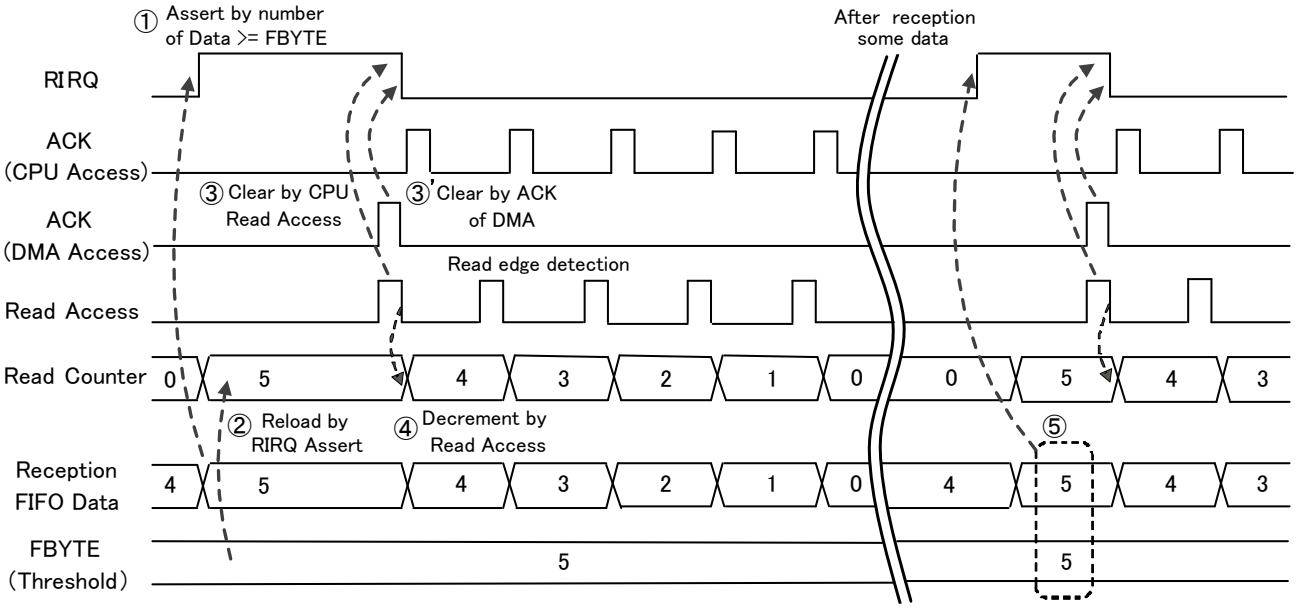
A reception block transfer request (RIRQ) is asserted when the data count in the reception FIFO exceeds the threshold value (FBYTE). At this time, the reception block size value is reloaded from the FBYTE register to the Read counter. The reception block size value is "1" when the reception FIFO is not used.

After the reception block transfer is requested, the reception block transfer request (RIRQ) is cleared when a read access to the reception data occurs. The read access to the reception data decrements the Read counter.

If the data count in the reception FIFO exceeds the threshold value (FBYTE) when the Read counter becomes "0", the reception block transfer request (RIRQ) is asserted again. Asserting the reception block transfer request (RIRQ) reloads the Read counter.

During application operation, if the next block transfer request occurs before the completion of the previous block transfer, this state might be reported to the system by a block transfer error. This state does not occur if you configure FBYTE=1. Completion of block transfer by DMA depends on occupation of bus transaction by application. You need to fully evaluate the application integrally, design the system, and choose the optimal number of block transfers.

Figure 6-1 Reception Block Transfer in RW Access Mode



Transmission Block Transfer

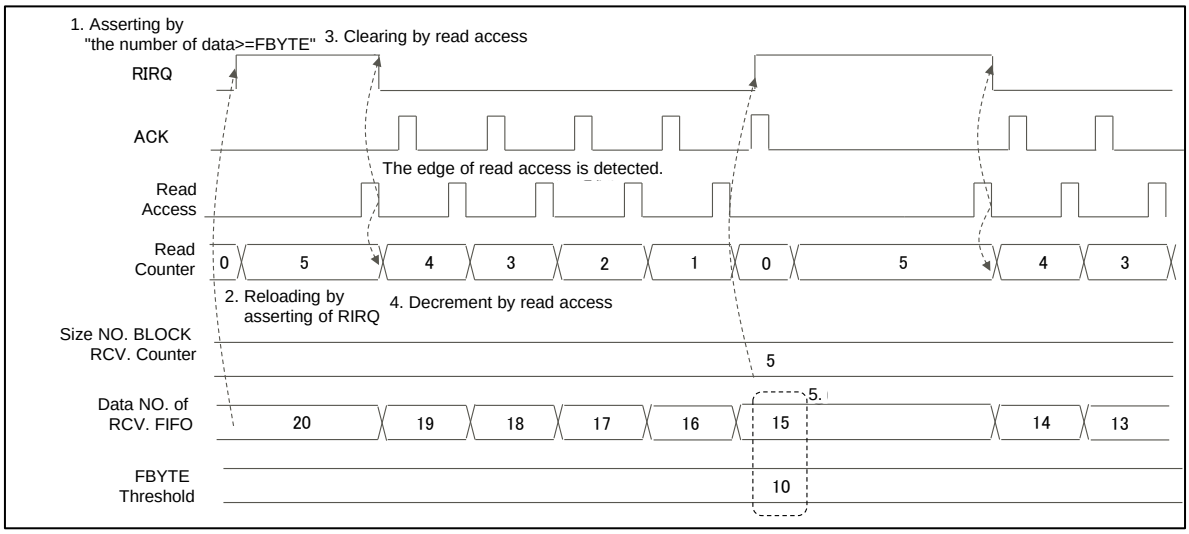
Transmission block transfer operates as follows.

A transmission block transfer request (TIRQ) is asserted when the available capacity in the transmission FIFO exceeds the threshold value (FTICR). At this time, the transmission block size value is reloaded from the TBSIZE register to the Write counter. The transmission block size value is "1" when the transmission FIFO is not used.

After the transmission block transfer is requested, the transmission block transfer request (TIRQ) is cleared when write access to the transmission FIFO occurs. The write access to the transmission FIFO decrements the Write counter.

If the available capacity in the transmission FIFO exceeds the threshold value (FTICR) when the Write counter becomes "0", the transmission block transfer request (TIRQ) is asserted again. Asserting the transmission block transfer request (TIRQ) reloads the Write counter.

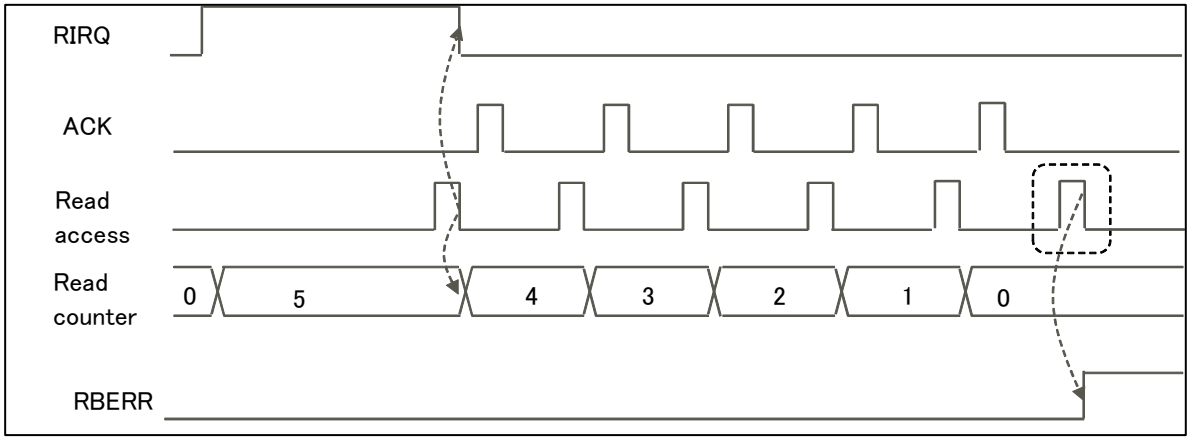
Figure 6-2 Transmission Block Transfer in RW Access Mode



Reception Block Transfer Error

The occurrence of read access before the assertion of the next reception block transfer request (RIRQ) while the Read counter is "0" causes a reception block transfer error. Then, the ESR:RBERR bit is asserted.

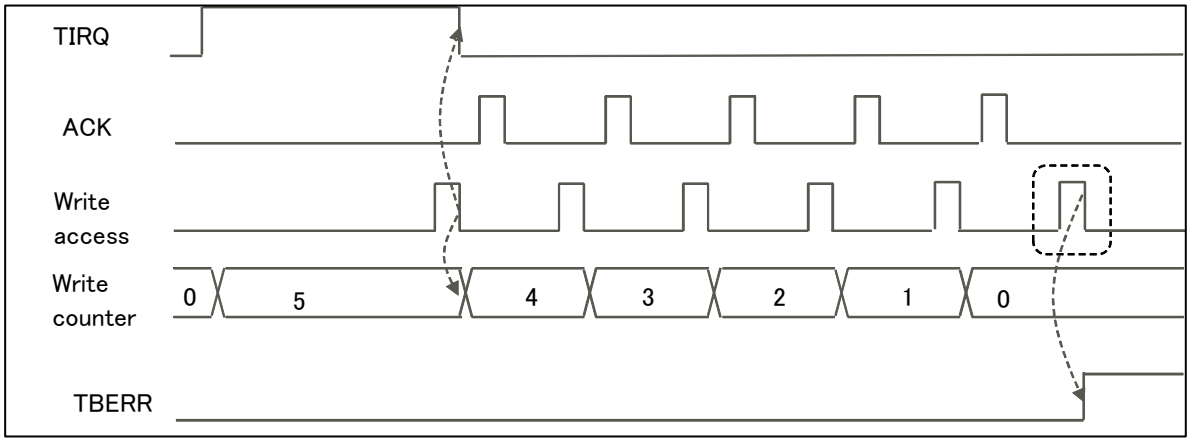
Figure 6-3 Reception Block Transfer Error in RW Access Mode



Transmission Block Transfer Error

The occurrence of write access before the assertion of the next transmission block transfer request (TIRQ) while the Write counter is "0" causes a transmission block transfer error. Then, the ESR:TBERR bit is asserted.

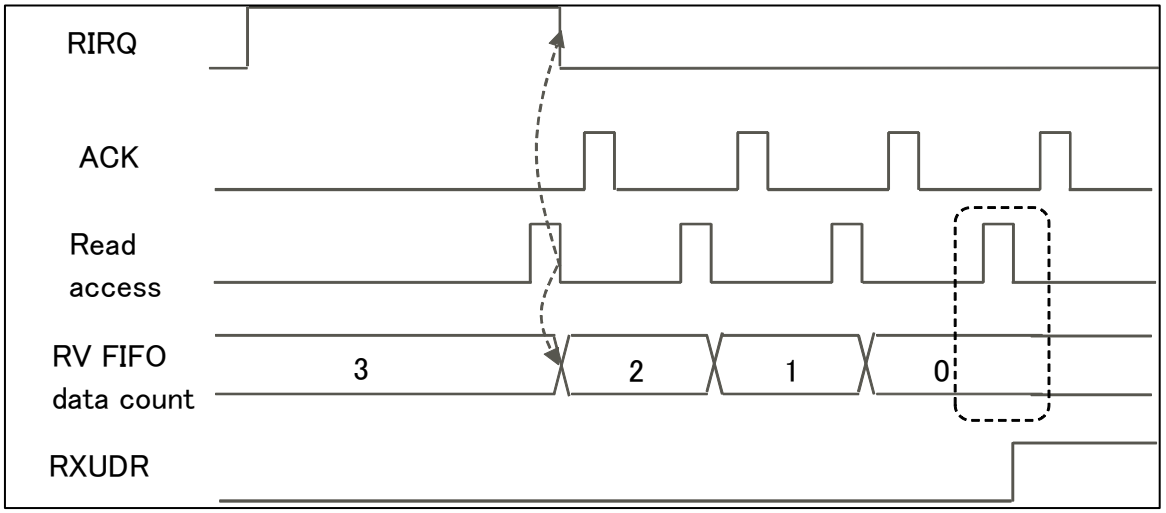
Figure 6-4 Transmission Block Transfer Error in RW Access Mode



Reception FIFO under run

Reading reception data from the reception FIFO while the data count in the reception FIFO is "0" causes a reception FIFO under run. Then, the ESR:RXUDR bit is asserted.

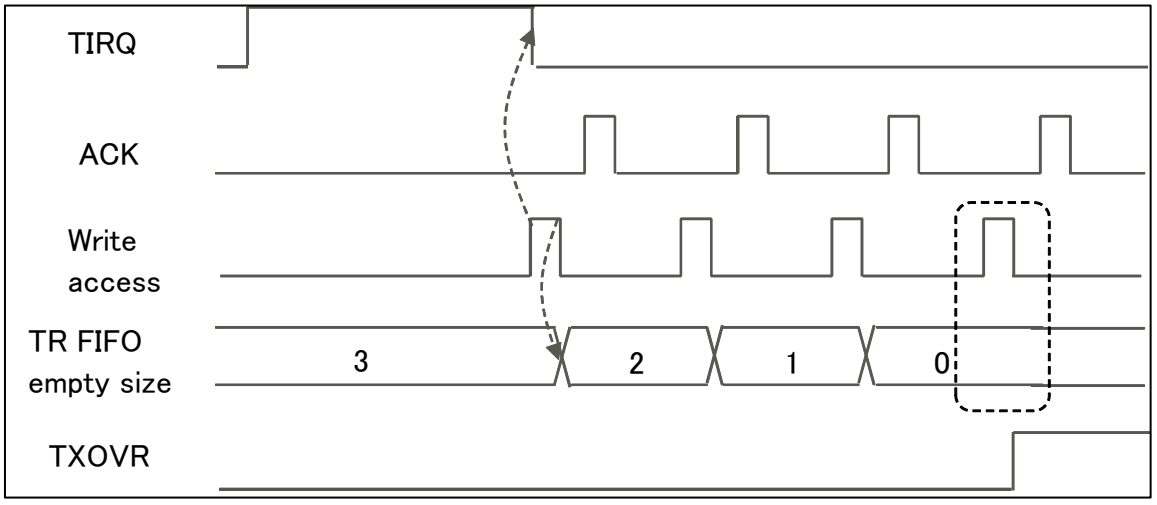
Figure 6-5 Reception FIFO Under run in RW Access Mode



Transmission FIFO overrun

Writing transmission data to the transmission FIFO while the available capacity of the transmission FIFO is "0" causes a transmission FIFO overrun. Then, the ESR:TXOVR bit is asserted.

Figure 6-6 Transmission FIFO Overrun in RW Access Mode



7. I²C Communication Operation Flowchart Examples

This section shows examples of flowcharts of I²C communication operations.

I²C Flowchart Examples When DMA Mode is Disabled (SSR:DMA=0) (FIFO is Not Used)

Figure 7-1 I²C Flowchart Example Where DMA Mode Is Disabled (SSR:DMA=0) (FIFO Not Used) 1/3

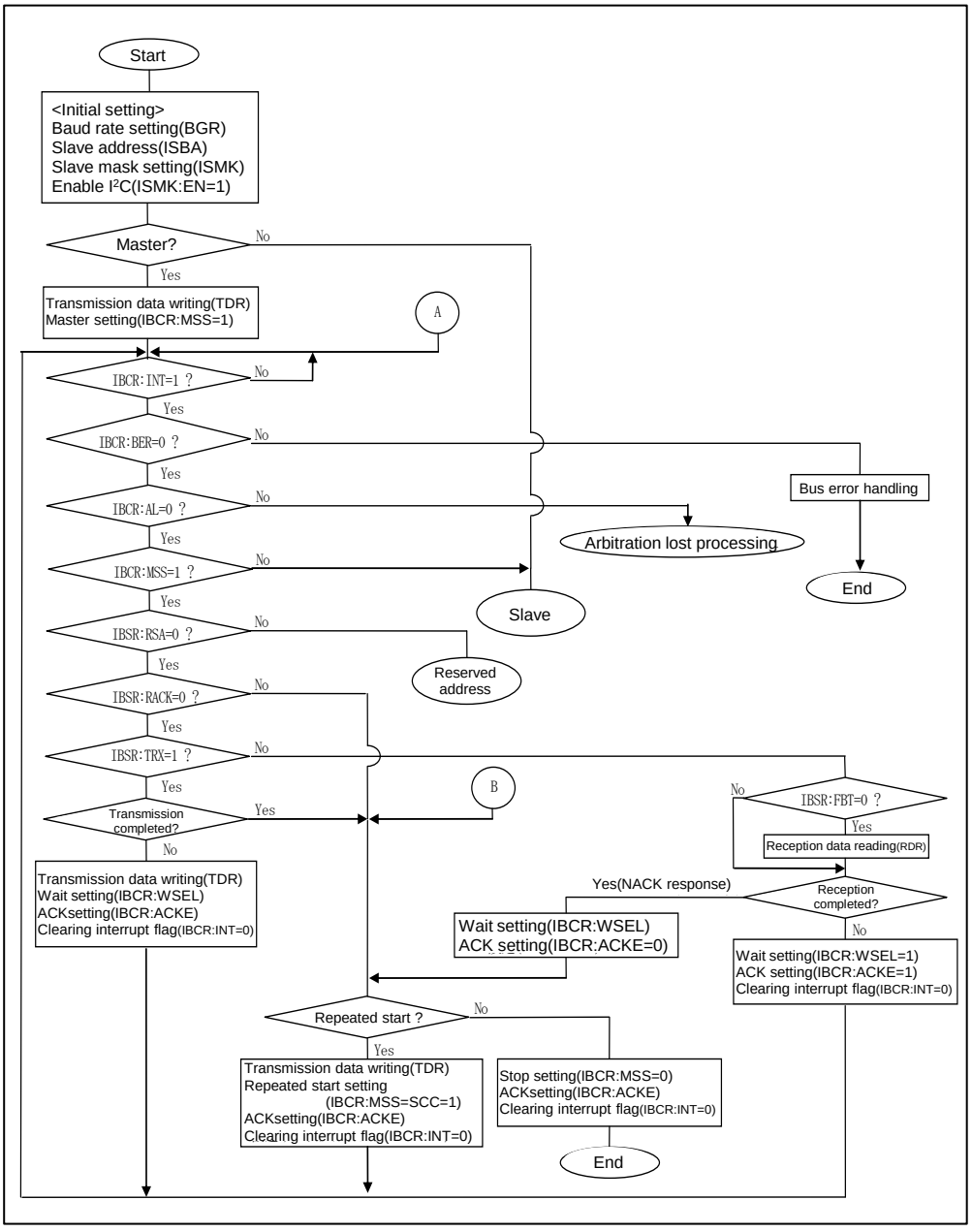


Figure 7-2 I²C Flowchart Example Where DMA Mode Is Disabled (SSR:DMA=0) (FIFO Not Used) 2/3

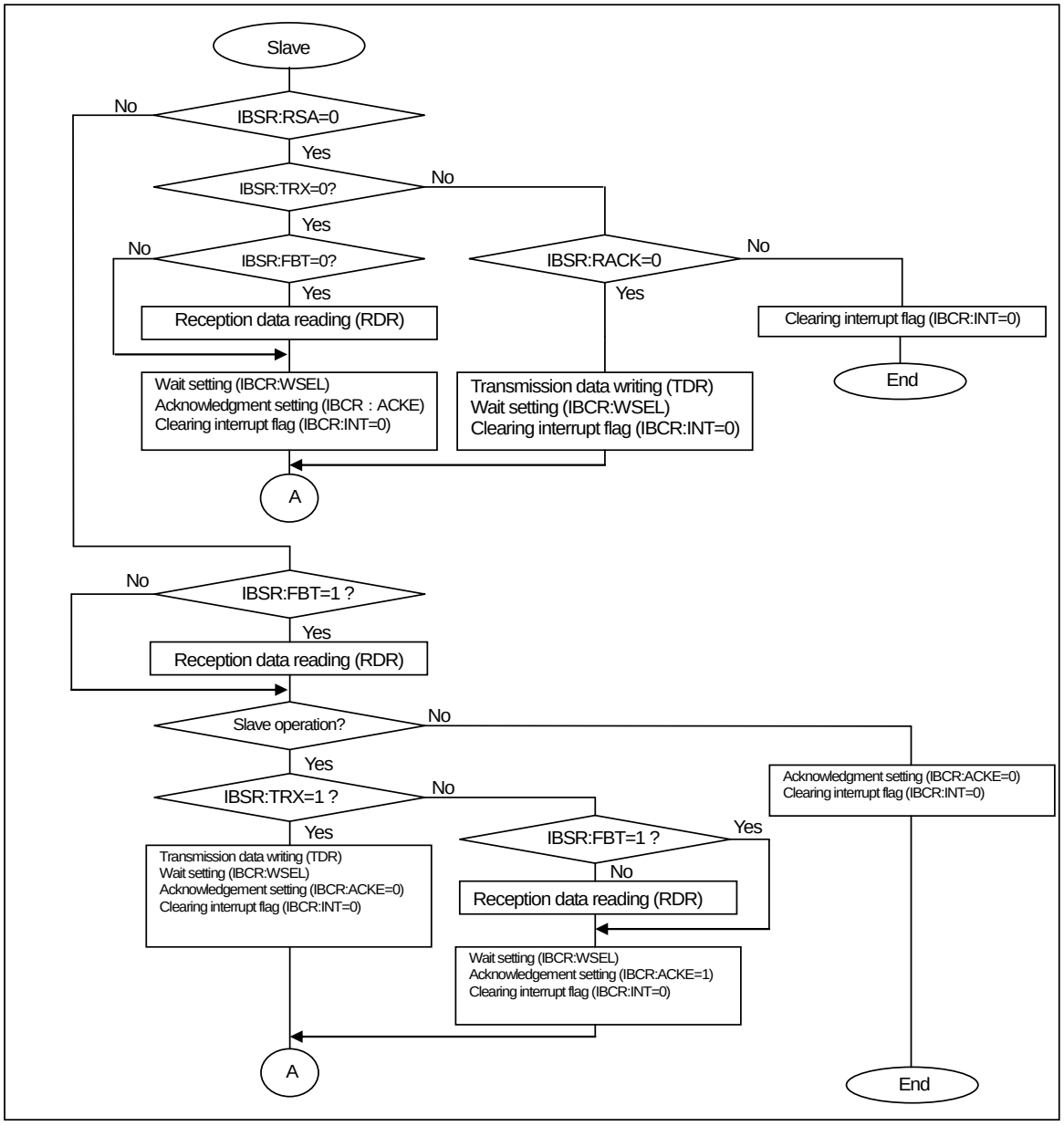
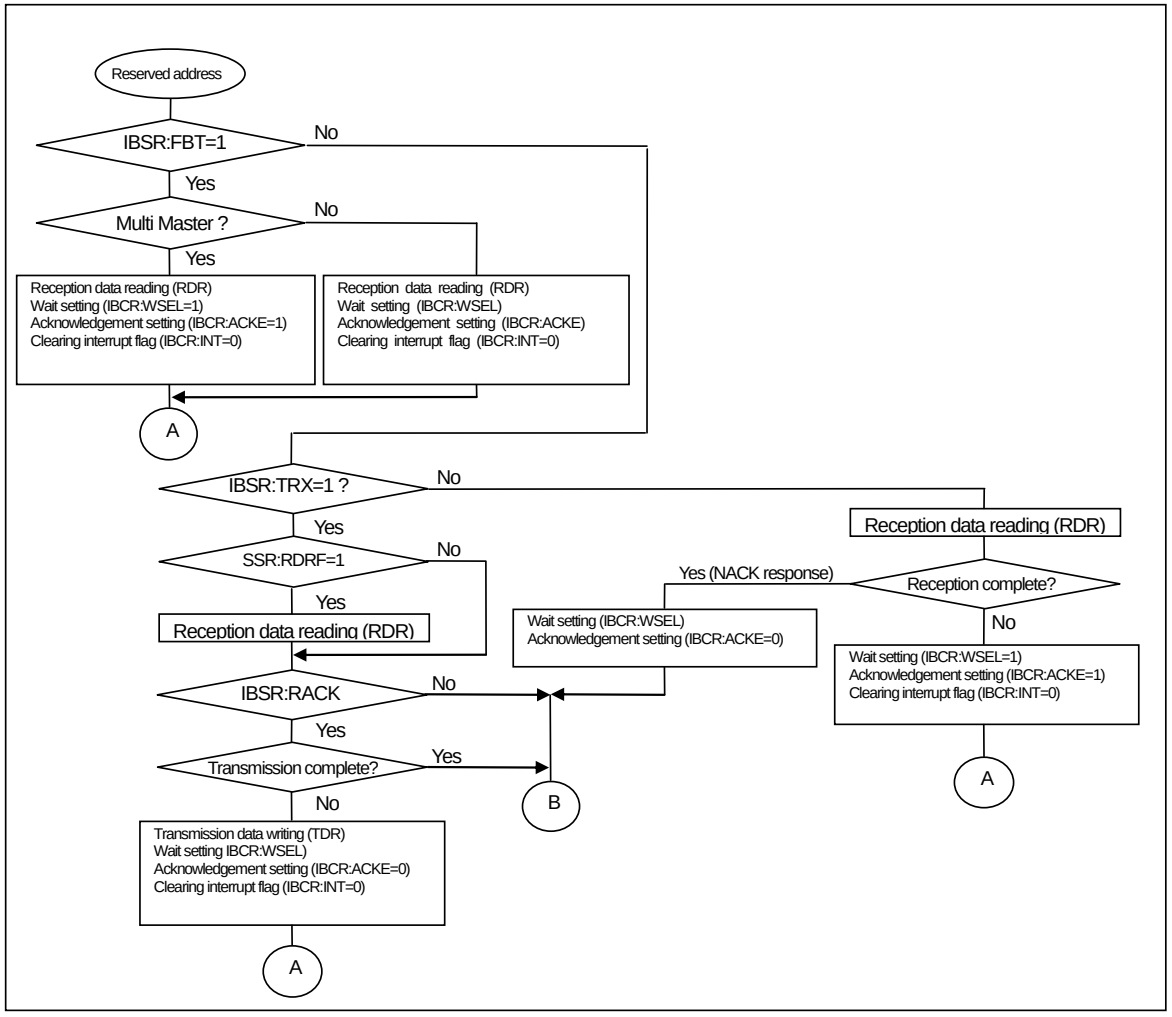


Figure 7-3 I²C Flowchart Example Where DMA Mode Is Disabled (SSR:DMA=0) (FIFO Not Used) 3/3



I²C Flowchart Example Where DMA Mode is Enabled (SSR:DMA=1) (FIFO Not Used)

Figure 7-4 I²C Flowchart Example Where DMA Mode Is Enabled (SSR:DMA=1) (FIFO Not Used) 1/5

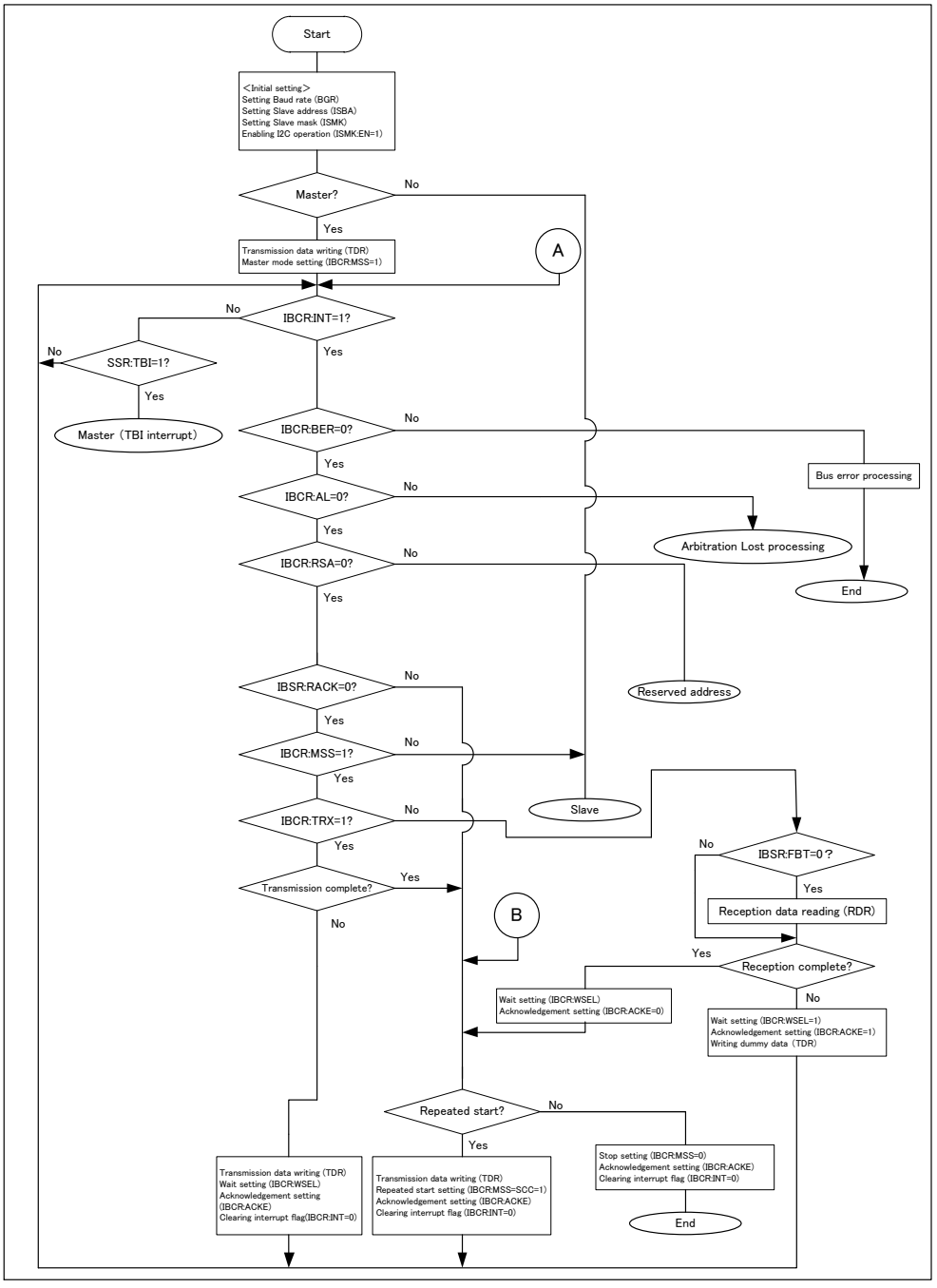


Figure 7-5 I²C Flowchart Example Where DMA Mode Is Enabled (SSR:DMA=1) (FIFO Not Used) 2/5

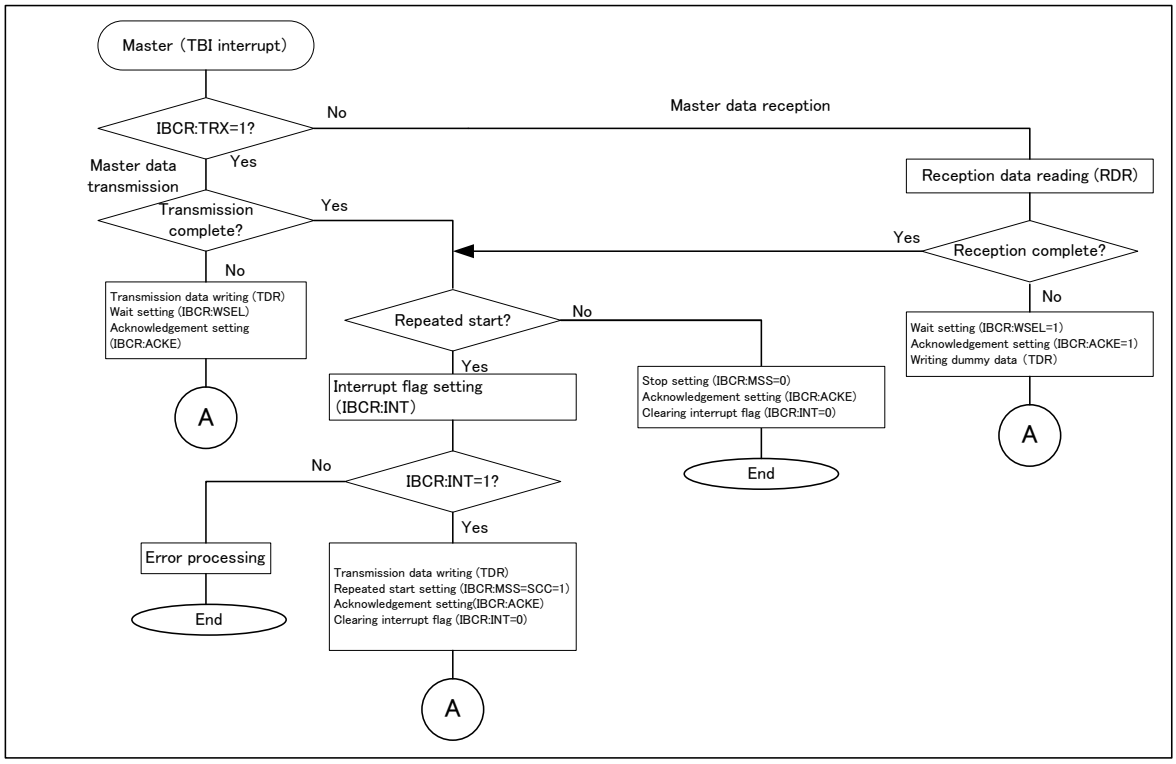


Figure 7-6 I²C Flowchart Example Where DMA Mode Is Enabled (SSR:DMA=1) (FIFO Not Used) 3/5

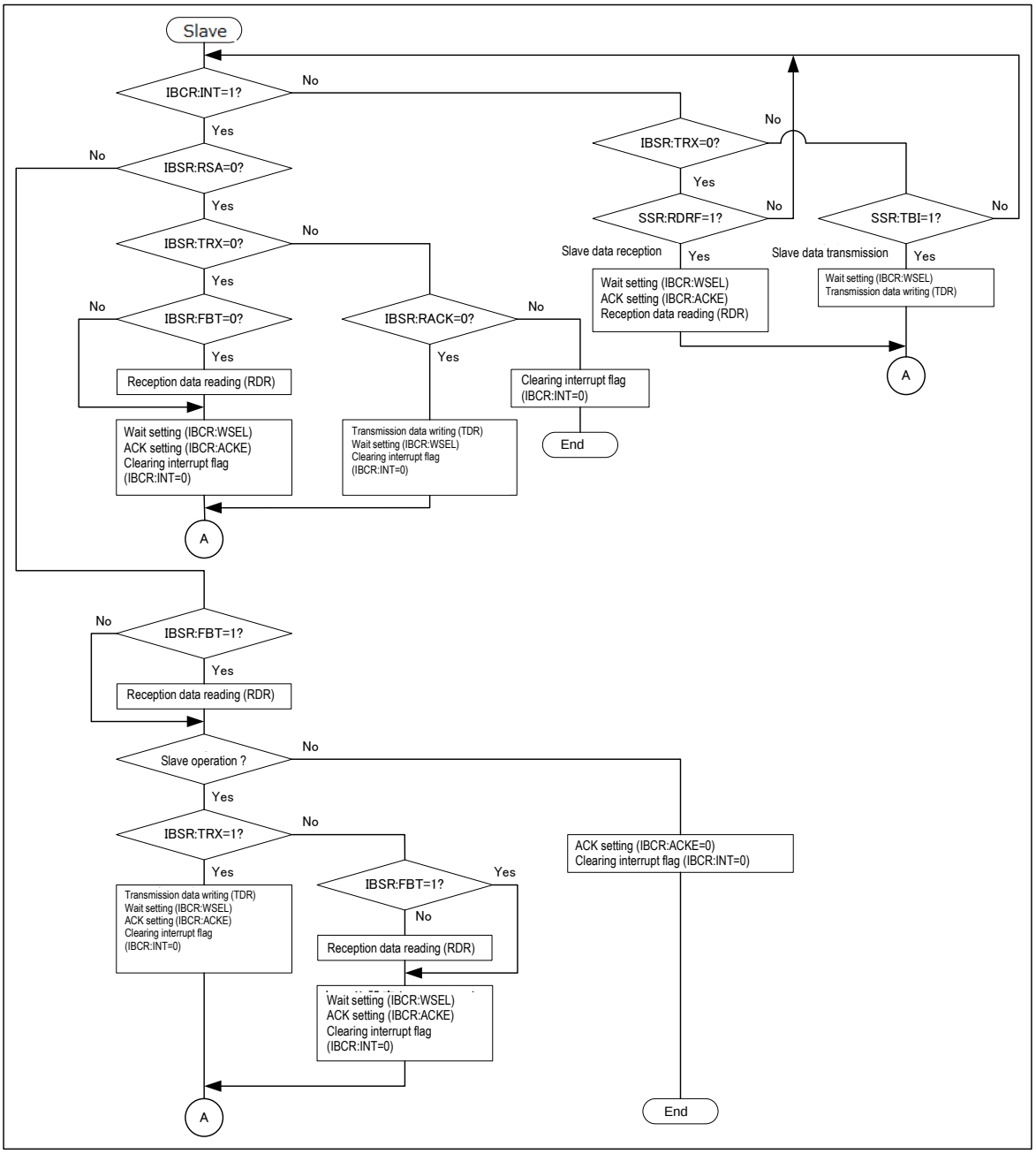


Figure 7-7 I²C Flowchart Example Where DMA Mode Is Enabled (SSR:DMA=1) (FIFO Not Used) 4/5

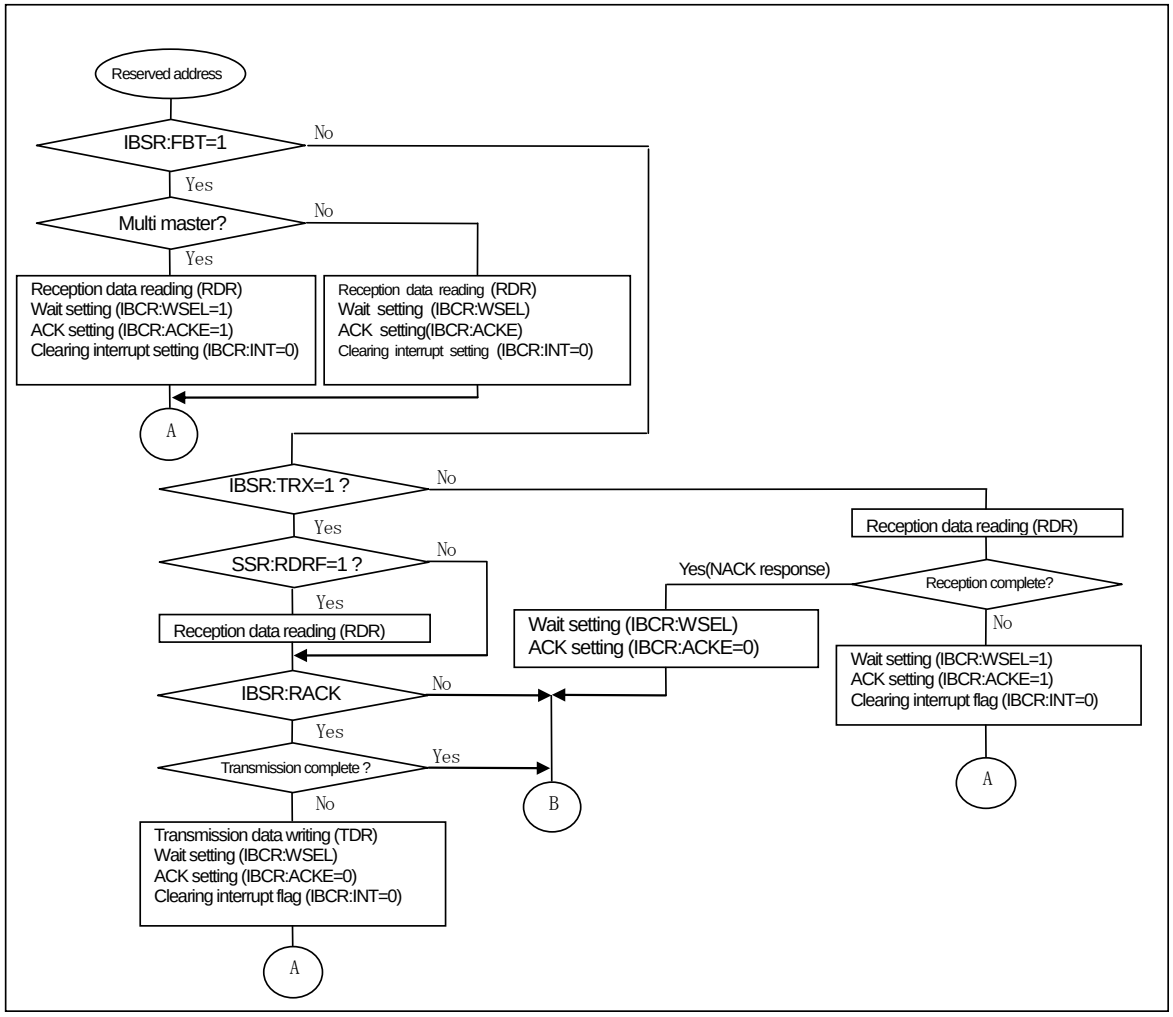
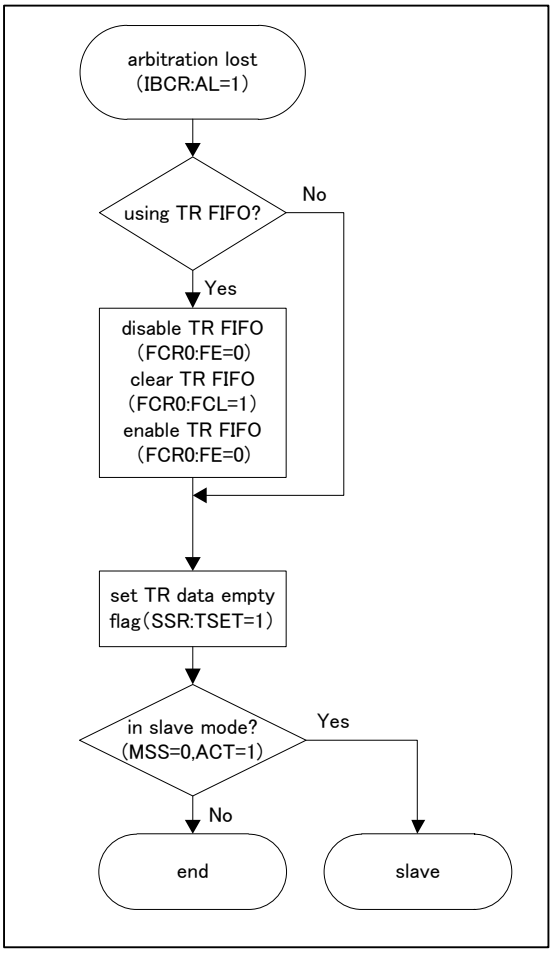


Figure 7-8 Arbitration Lost Processing Flowchart Example 5/5



Note:

- *The flows roughly show the settings for operation in I^2C mode. Processing must take into account error handling, etc. according to the application.*

8. I²C Interface Registers

This section lists the I²C interface registers.

I²C Interface Registers

Table 8-1 I²C Interface Registers

	bit15 bit8	bit7 bit0
I²C	IBCR (I²C bus control register)	SMR (serial mode register)
	SSR (serial status register)	IBSR (I²C bus status register)
	-	RDR/TDR (reception data register/transmission data register)
	-	-
	SACSR1 (serial auxiliary control status register 1)	SACSR0 (serial auxiliary control status register 0)
	STMR1 (serial timer register 1)	STMR0 (serial timer register 0)
	STMCR1 (serial timer comparison register 1)	STMCR0 (serial timer comparison register 0)
	-	-
	EIBCR (extended I²C bus control register)	NFCR (noise filter control register)
	-	-
	-	-
	-	-
	-	-
	-	-
	BGR1 (baud rate generator register 1)	BGR0 (baud rate generator register 0)
	ISMK (7-bit slave address mask register)	ISBA (7-bit slave address register)
FIFO	FCR1 (FIFO control register 1)	FCR0 (FIFO control register 0)
	FBYTE2 (FIFO2 byte register)	FBYTE1 (FIFO1 byte register)
	FTICR2 (transmission FIFO interrupt control register 2)	FTICR1 (transmission FIFO interrupt control register 1)
Auxiliary Register	ESR (extended status register)	ECR (extended control register)
	-	TBSIZE (transmission block size register)
	-	-
Clear Registers	IBCRC (I²C bus control clear register)	SMRC (serial mode clear register)
	SSRC (serial status clear register)	IBSRC (I²C bus status clear register)
	SACSR1C (serial auxiliary control status clear register 1)	SACSR0C (serial auxiliary control status clear register 0)
	-	-
	-	-
	-	-
	-	-
	-	-
	FCR1C (FIFO control clear register 1)	FCR0C (FIFO control clear register 0)
	-	-
	-	-
	ESRC (extended status clear register)	-
Set	IBCRS (I²C bus control set register)	SMRS (serial mode set register)

	bit15 bit8	bit7 bit0
Registers	SSRS (serial status set register)	-
	SACSR1S (serial auxiliary control status set register 1)	SACSR0S (serial auxiliary control status set register 0)
	-	-
	-	-
	-	-
	-	-
	-	-
	FCR1S (FIFO control set register 1)	FCR0S (FIFO control set register 0)
	-	-
Reception/Transmission Register	-	RDR/TDR (transmission/reception data register)*1
	-	-
	-	RDR/TDR (transmission/reception data register)*1
	-	-
	-	RDR/TDR (transmission/reception data register)*1
	-	-
	-	RDR/TDR (transmission/reception data register)*1
	-	-
	-	RDR/TDR (transmission/reception data register)*1
	-	-
	-	RDR/TDR (transmission/reception data register)*1
	-	-
	-	RDR/TDR (transmission/reception data register)*1
	-	-
	-	RDR/TDR (transmission/reception data register)*1
	-	-
	-	RDR/TDR (transmission/reception data register)*1
	-	-
	-	RDR/TDR (transmission/reception data register)*1
	-	-
	-	RDR/TDR (transmission/reception data register)*1
	-	-
	-	RDR/TDR (transmission/reception data register)*1
	-	-
	-	RDR/TDR (transmission/reception data register)*1
	-	-
	-	RDR/TDR (transmission/reception data register)*1
	-	-
	-	RDR/TDR (transmission/reception data register)*1
	-	-
	-	RDR/TDR (transmission/reception data register)*1
	-	-
	-	RDR/TDR (transmission/reception data register)*1
	-	-
	-	RDR/TDR (transmission/reception data register)*1
	-	-
	-	RDR/TDR (transmission/reception data register)*1
	-	-
	-	RDR/TDR (transmission/reception data register)*1
	-	-

*1 Access via a mirror address area

8.1. I²C Bus Control Register (IBCR)

The I²C bus control register (IBCR) sets the selection of master/slave mode, generation of the repeated start condition, enabling of acknowledgments, and enabling of interrupts. The IBCR also displays an interrupt flag.

I²C Bus Control Register (IBCR)

Figure 8-1 shows the bit configuration of the I²C bus control register (IBCR).

Figure 8-1 Bit Configuration of the I²C Bus Control Register (IBCR)

Bit	15	14	13	12	11	10	9	8	7 ... 0
Field	MSS	ACT/ SCC	ACKE	WSEL	CNDE	INTE	BER	INT	(SMR)
R/W Attribute	R/W	R,W	R/W	R/W	R/W	R/W	R,WX	R,WX	
Protection Attribute	-	-	-	-	-	-	-	-	
Initial Value	0	0	0	0	0	0	0	0	

[bit15] MSS : Master/Slave Selection Bit

- If this bit is set to "1" when the I²C bus is in the idle state (ISMK:EN= 1 and IBSR:BB= 0), the mode is set to master mode.
- After this bit is set to "1" when the BB bit in the IBSR register is "1", a wait for start condition generation is performed until the IBSR:BB bit is set to "0". If a slave address is found to match during this wait and the device operates as a slave, this bit is set to "0" and the AL bit in the IBSR register is set to "1".
- Writing "0" to this bit during master operation (MSS= 1 and ACT= 1) when the interrupt flag (INT) is "1" generates the stop condition.
- This bit is reset when the IBCRC:MSSC bit in the clear register is set to "1".
- This bit is set when the IBCRS:MSSS bit in the set register is set to "1".

MSS is cleared under the following conditions.

1. I²C interface operation is disabled (ISMK:EN bit = 0).
2. Arbitration lost occurs.
3. A bus error is detected (BER bit = 1) (except in cases where the stop condition is detected in the acknowledgment field during master operation).
4. "0" is written to the MSS bit when INT= 1.
5. "0" is written to the MSS bit when DMA mode is enabled (SSR:DMA= 1) and SSR:TBI= 1.

The following table shows the relationship between the MSS bit and the ACT bit.

MSS Bit	ACT Bit	State
0	0	Idle
0	1	Slave operation in progress (slave mode) after ACK response*1 is returned for slave address match or reserved address
1	0	Master operation waiting
1	1	Master operation in progress (master mode)

*1 ACK response: Indicates that the SDA of the I2C bus is "L" during the acknowledgment interval.

MSS Bit	Description
0	Select slave mode.
1	Select master mode.

Notes:

- *To change the MSS bit to "0" when DMA mode is disabled (SSR:DMA= 0) and the MSS bit is set to "1", do so when the MSS bit is "1" and the INT bit is "1". Writing "0" to the MSS bit when the ACT bit is "1" also clears the INT bit to "0".*
- *To change the MSS bit to "0" when DMA mode is enabled (SSR:DMA= 1) and the MSS bit is set to "1", do so when the MSS bit is "1" and the INT bit is "1" or the SSR:TBI bit is "1". Writing "0" to the MSS bit when the ACT bit is "1" also clears the INT bit to "0".*
- *During master operation, "1" is read while the ACT bit is "1", even if "0" is written to the MSS bit.*
- *If "1" is written to the MSS bit during slave operation, this bit is set to "0" and the AL bit in the IBSR register is set to "1". If you want the device to operate in master mode after slave operation ends, perform the following procedure.*

  1. Verify the end of slave operation through detection of the stop condition (IBCR:SPC= 1).
  2. Write the slave address and transmission data (only when the transmission FIFO is used).
  3. Write "1" to this bit.

[bit14] ACT/SCC: Operation Flag/Repeated Start Condition Generation Bit

The meaning of this bit varies depending on whether the operation is read or write.

ACT/SCC	Description
Write	SCC bit
Read	ACT bit

The ACT bit indicates whether the device is operating in master or slave mode.

■ This bit is set when the IBCRS:ACTS bit in the set register is set to "1".

ACT bit set conditions:

1. The start condition is output to the I2C bus (master mode).
2. The slave address matches the address transmitted by the master (slave mode).
3. The reserved address is detected, and an acknowledgment response to the detection is returned (slave mode is selected when MSS= 0).

ACT bit reset conditions:

<Master mode>

1. The stop condition is detected (except in cases where the stop condition is detected in the acknowledgment field).

2. Arbitration lost is detected.
3. A bus error is detected (BER bit = 1) (except in cases where the stop condition is detected in the acknowledgment field).
4. I^2C interface is operation disabled (ISMK:EN bit = 0).

<Slave mode>

1. The (repeated) start condition is detected.
2. The stop condition is detected.
3. No acknowledgment response is returned in the detection of the reserved address (IBSR:RSA= 1).
4. I^2C interface operation is disabled (ISMK:EN bit = 0).
5. A bus error is detected (BER bit = 1).

Writing "1" to this bit in master mode executes a repeated start. Writing "0" is invalid.

Bit	Description	
	Write	**Read**
0	No effect	No operation
1	Generate the repeated start condition.	The I^2C is operating.

Notes:

- *When writing "1" to the SCC bit, do so during an interrupt in master mode (MSS= 1, ACT= 1, and INT= 1). Writing "1" to the SCC bit when the ACT bit is "1" clears the INT bit to "0".*
- *Writing "1" to this bit in slave mode (MSS= 0 and ACT= 1) is prohibited.*
- *If "1" is written to the SCC bit and "0" to the MSS bit, the MSS bit takes priority.*
- *If both of the following conditions are satisfied, the INT bit is set to "1" to wait for the I^2C bus (SCL="L"). To generate the repeated start condition, it is necessary to write "1" to the SCC bit again and clear the INT bit.*
  - "1" is written to the SCC bit when a master mode interrupt (MSS= 1, ACT= 1, INT= 1, and WSEL= 1) is generated at the 8th bit.
  - NACK is received as the 9th bit.
- *To issue the repeated start condition when DMA mode is enabled (SSR:DMA= 1), the SSR:TBI bit is "1", and the IBCR:INT bit is "0", perform the following procedure.*
  1. Write "1" to the IBCR:INT bit.
  2. Verify that the IBCR:INT bit is set to "1".
  3. Write the slave address to the TDR.
  4. Set this bit to "1".

[bit13] ACKE : Data Byte acknowledgment Enable Bit

- If this bit is set to "1", "L" is output at the acknowledgment timing.
- Under any of the following conditions, change this bit.
  - The INT bit is "1" when ACT= 1.
  - DMA mode is enabled (SSM:DMA= 1) and the SSR:TBI bit is "1" when ACT= 1.

- DMA mode is enabled (SSM:DMA= 1) and SSR:RDRF is "1" during slave reception when ACT= 1.
- ACT is "0".
- This bit is reset when the IBCRC:ACKEC bit in the clear register is set to "1".
- This bit is set when the IBCRS:ACKES bit in the set register is set to "1".

This bit is invalid under the following conditions.

1. There is an acknowledgment for an address field for other than the reserved address (automatic generation).
2. Data is being transmitted (IBSR:RSA= 0, IBSR:TRX= 1, and IBSR:FBT= 0).
3. An ACK response is always returned during slave reception when the reception FIFO is enabled (FCR0:FE= 1, MSS= 0, and ACT= 1).
4. The response during master reception when the reception FIFO is enabled and WSEL is "0" (FCR0:FE= 1, MSS= 1, ACT= 1, and WSEL= 0) is as follows. If the SSR:TDRE bit is "0", an ACK response is returned. If the SSR:TDRE bit is "1", a NACK response is returned.
5. An ACK response is always returned during slave transmission when the reception FIFO is enabled, WSEL= 0, and the reserved address is detected (IBSR:RSA= 1, IBSR:TRX= 1, IBSR:FBT= 1). If a NACK response is returned, disable the reception FIFO and set ACKE= 0 at the interrupt time after the reserved address is detected.
6. The transmission data register has data during master reception when the reception FIFO is enabled and WSEL is "1" (FCR0:FE= 1, MSS= 1, ACT= 1, WSEL= 1, and SSR:TDRE= 0).

Bit	Description
0	Disable acknowledgment.
1	Enable acknowledgment.

[bit12] WSEL : Wait Selection Bit

- When DMA mode is disabled (SSR:DMA= 0), this bit generates an interrupt (INT= 1) before or after acknowledgment and selects whether to wait for the I^2C bus.
- When DMA mode is enabled (SSR:DMA= 1), this bit selects whether to wait for the I^2C bus before or after an acknowledgment.
- The WSEL bit is invalid under the following conditions.
  1. An interrupt for the 1st byte*1 is generated (INT= 1).
  2. The reserved address is detected (IBSR:FBT= 1 and IBSR:RSA= 1).
  3. A NACK response*2 is detected during data transmission with FIFO used (FCR0:FE= 1, IBSR:RACK= 1, and ACT= 1).
  4. The reception FIFO becomes full during use.

  *1 1st byte: Indicates the data after the (repeated) start condition.

  *2 NACK response: Indicates that the SDA of the I^2C bus is "H" during an acknowledgment period.
- This bit is reset when the IBCRC:WSELC bit in the clear register is set to "1".
- This bit is set when the IBCRS:WSELS bit in the set register is set to "1".

Bit	Description
0	Wait after acknowledgment (9th bit).
1	Wait after data transmission is completed (8th bit).

[bit11] CNDE : Condition Detection Interrupt Enable Bit

This bit enables interrupt generation when the stop or repeated start condition is detected in master or slave mode (ACT= 1). An interrupt is generated when the RSC or SPC bit in the IBSR register is "1" and this bit is "1".

- This bit is reset when the IBCRC:CNDEC bit in the clear register is set to "1".
- This bit is set when the IBCRS:CNDES bit in the set register is set to "1".

Bit	Description
0	Disable repeated start or stop condition interrupts.
1	Enable repeated start or stop condition interrupts.

[bit10] INTE : Interrupt Enable Bit

This bit enables interrupts (INT= 1) for data transmission and reception and for bus errors in master or slave mode.

- This bit is reset when the IBCRC:INTEC bit in the clear register is set to "1".
- This bit is set when the IBCRS:INTES bit in the set register is set to "1".

Bit	Description
0	Disable interrupts.
1	Enable interrupts.

[bit9] BER : Bus Error Flag Bit

This bit indicates that an error has been detected on the I^2C bus.

BER bit set conditions:

1. The start or stop condition is detected during transfer of the 1st byte*1.
2. At the 2nd or subsequent byte, the (repeated) start or stop condition is detected in the 2nd to 9th (acknowledgment) bits of data.

BER bit reset conditions:

1. "1" is written to the IBCRC:INTC bit when EIBCR:BEC= 0 and BER= 1.
2. I2C interface operation is disabled (ISMK:EN= 0).
3. "1" is written to the IBCRC:INTC bit when EIBCR:BEC= 1 and IBCR:INT= 1.
4. "1" is written to the IBSRC:SPCC bit when EIBCR:BEC= 1 and IBSR:SPC= 1.
5. "1" is written to the IBSRC:RSCC bit when EIBCR:BEC= 1 and IBSR:RSC= 1.

*1 1st byte: Indicates the data after the (repeated) start condition.

Bit	Description
0	There is no error.
1	An error is detected.

Notes:

- *Check this bit in the following cases. If it is "1", data was either not transmitted or received correctly. Perform retransmission or other processing accordingly.*
  - When EIBCR:BEC= 0, the interrupt flag (INT bit) was set to "1".
  - When EIBCR:BEC= 1, the repeated start condition check bit (IBSR:RSC bit) was set to "1".
  - When EIBCR:BEC= 1, the stop condition check bit (IBSR:SPC bit) was set to "1".
- *If all of the following conditions are true when EIBCR:BEC= 1, issue the stop condition:*

- IBCR:BER=1
- IBCR:INT=1
- IBCR:ACT=1
- IBCR:MSS=1

[bit8] INT : Interrupt Flag Bit

This bit sets the flag to "1" in master or slave mode when a bus error occurs or after 8 or 9 bits (ACK) of data are transmitted or received. The SCL is set to "L" when the INT bit is set to "1" and released from the "L" state when the INT bit is set to "0", although this does not apply when a bus error occurs.

INT bit set conditions:

<8th bit>

<When DMA mode is not involved>

1. Write "1" to IBCRC:INTS.
2. The reserved address is detected in the 1st byte.
3. Arbitration lost is detected in the 2nd or subsequent byte when WSEL is "1".

<When DMA mode is disabled (SSR:DMA=0)>

1. The SSR:TDRE bit is "1" at the 2nd or subsequent byte during master operation when WSEL is "1".
2. The SSR:TDRE bit is "1" at the 2nd or subsequent byte during slave operation when WSEL is "1" and the reception FIFO is disabled.
3. The SSR:TDRE bit is "1" at the 2nd or subsequent byte during slave transmission when WSEL is "1".
4. Slave reception occurs when the reception FIFO is disabled and WSEL is "1".

<When DMA mode is enabled (SSR:DMA=1)>

1. "1" is written to the INT bit at the 2nd or subsequent byte during master operation when WSEL is "1" and the SSR:TBI bit is "1".

<9th bit>

<When DMA mode is not involved>

1. Arbitration lost is detected in the 1st byte.
2. A NACK is received at a time other than when stop condition output is set ("0" is written to the MSS bit during master operation).
3. Arbitration lost is detected in the 2nd or subsequent byte when WSEL is set to "0".
4. The reserved address is not detected in the 1st byte, and the reception FIFO has data in the reception direction (IBSR:TRX= 0) in master or slave mode when the reception FIFO is enabled.
5. IBSR:BER is "1" during master operation (MSS= 1 and ACT= 1) when EIBCR:BEC= 1.

<When DMA mode is disabled (SSR:DMA=0)>

1. The reserved address is not detected in the 1st byte, and the SSR:TDRE bit is "1" in the transmission direction (IBSR:TRX= 1) in master or slave mode.
2. The reserved address is not detected in the 1st byte, and the SSR:TDRE bit is "1" in the reception direction (IBSR:TRX= 0) in master or slave mode when the reception FIFO is disabled.

3. The SSR:TDRE bit at the 2nd or subsequent byte is "1" during operation in master mode when WSEL is set to "0".
4. The SSR:TDRE bit at the 2nd or subsequent byte is "1" during slave transmission when WSEL is set to "0".
5. The reception FIFO is disabled during slave reception when WSEL is set to "0". However, an interrupt is not generated in the 9th bit during slave reception of the 1st byte where the reserved address has been detected.
6. The reception FIFO becomes full during slave reception with the reception FIFO enabled.

<When DMA mode is enabled (SSR:DMA=1)>

1. The reserved address is not detected in the 1st byte, and the SSR:TDRE bit is "1" in the transmission direction (IBSR:TRX= 1) in slave mode.
2. The reserved address is not detected in the 1st byte, and the SSR:TDRE bit is "1" in the reception direction (IBSR:TRX= 0) in slave mode with the reception FIFO disabled.
3. "1" is written to the INT bit at the 2nd or subsequent byte during operation in master mode, where the SSR:TBI bit is "1" when WSEL is set to "0".

<Other>

1. A bus error is detected when IBCR:BEC= 0.

INT bit reset conditions:

1. "1" is written to the IBCRC:INTC bit.
2. "0" is written to the MSS bit when the INT bit is "1" and the ACT bit is "1".
3. "1" is written to the SCC bit when the INT bit is "1" and the ACT bit is "1".

Writing "1" to the INT bit is invalid when DMA mode is disabled (SSR:DMA= 0).

Bit	Description	
	Write	Read
0	No effect	No interrupt request
1	No effect	Interrupt request

Notes:

- *If "1" is written to the INT bit at the 2nd or subsequent byte during operation in master mode when DMA mode is enabled (SSR:DMA= 1) and the SSR:TBI bit is "1", a status interrupt is not generated.*
- *To issue the repeated start condition when DMA mode is enabled (SSR:DMA= 1), the SSR:TBI bit is "1", and the IBCR:INT bit is "0", perform the following procedure.*
  1. Write "1" to the IBCR:INT bit.
  2. Verify that the IBCR:INT bit is set to "1".
  3. Write the slave address to the TDR.
  4. Set the IBCR:SCC bit to "1".
- *When "1" is set to INT flag, write "1" to IBCRC:INTC to release the wait for the I^2C bus.*
- *If the ISMK:EN bit is set to "0", the SSR:RDRF and INT bits may be set to "1" depending on the reception timing. In this case, read the reception data, and clear the INT bit.*

- *Even if the reception FIFO becomes full during master reception operation when the reception FIFO is enabled, the INT bit is not set to "1".*
- *Write "1" to this bit when the start condition is issued (IBCR:MSS= 1).*

8.2. Serial Mode Register (SMR)

The serial mode register (SMR) sets the operation mode and enables/disables transmission and reception interrupts.

Serial Mode Register (SMR)

Figure 8-2 shows the bit configuration of the serial mode register (SMR).

Figure 8-2 Bit Configuration of the Serial Mode Register (SMR)

Bit	15 ... 8	7	6	5	4	3	2	1	0
Field	(IBCR)	MD2	MD1	MD0	WUCR	RIE	TIE	Reserved	Reserved
R/W Attribute		R/W	R/W	R/W	R/W	R/W	R/W	R0,W0	R0,W0
Protection Attribute		-	-	-	-	-	-	-	-
Initial Value		0	0	0	0	0	0	0	0

[bit7:5] MD[2:0] : Operation Mode Setting Bits

These bits set the operation mode.

Bits			Description
0	0	0	Operation mode 0 (asynchronous normal mode)
0	0	1	Operation mode 1 (asynchronous multi-processor mode)
0	1	0	Operation mode 2 (clock synchronous mode)
0	1	1	Operation mode 3 (LIN communication mode)
1	0	0	Operation mode 4 (I^2C mode)

* This section explains the register and its operations in operation mode 4.

Notes:

- *Settings other than above are prohibited.*
- *To switch the operation mode, disable the I^2C (ISMK:EN= 0), and then switch the operation mode without a break.*
- *After setting the operation mode, set each register.*

[bit4] WUCR: WAKE UP Control Bit

WUCR is not supported

This bit selects the pin used for external interrupts.

- This bit is reset when the SMRC:WUCRC bit in the clear register is set to "1".
- This bit is set when the SMRS:WUCRS bit in the set register is set to "1".

"0": Set the INT pin for external interrupts.
"1": Set the SDA or SCL pin for external interrupts.

Bit	Description
0	Disable the WAKE UP function.
1	Enable the WAKE UP function.

Note:

− The WAKE UP function is not supported.

[bit3] RIE: Reception Interrupt Enable Bit

- This bit enables/disables reception interrupt request output to the CPU.
- A reception interrupt request is output when the RIE bit and the reception data flag bit (SSR:RDRF) are "1" or when any of the error flag bits (SSR:ORE) is "1".
- This bit is reset when the SMRC:RIEC bit in the clear register is set to "1".
- This bit is set when the SMRS:RIES bit in the set register is set to "1".

Bit	Description
0	Disable reception interrupts.
1	Enable reception interrupts.

Note:

− *To receive data using the INT bit in the I^2C bus control register (IBCR) when DMA mode is disabled (SSR:DMA= 0), set this bit to "0".*

[bit2] TIE: Transmission Interrupt Enable Bit

- This bit enables/disables transmission interrupt request output to the CPU.
- A transmission interrupt request is output when the TIE bit and the SSR:TDRE bit are "1".
- This bit is reset when the SMRC:TIEC bit in the clear register is set to "1".
- This bit is set when the SMRS:TIES bit in the set register is set to "1".

Bit	Description
0	Disable transmission interrupts.
1	Enable transmission interrupts.

Note:

− *To transmit data using the INT bit in the I^2C bus control register (IBCR) when DMA mode is disabled (SSR:DMA= 0), set this bit to "0".*

[bit1:0]Reserved : Reserved Bits

8.3. I²C Bus Status Register (IBSR)

The I²C bus status register (IBSR) indicates whether the repeated start, an acknowledgment, the data direction, arbitration lost, the stop condition, the I²C bus status, and a bus error have been detected.

I²C Bus Status Register (IBSR)

Figure 8-3 shows the bit configuration of the I²C bus status register (IBSR).

Figure 8-3 Bit Configuration of the I²C Bus Status Register (IBSR)

Bit	15 ... 8	7	6	5	4	3	2	1	0
Field	(SSR)	FBT	RACK	RSA	TRX	AL	RSC	SPC	BB
R/W Attribute		R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute		-	-	-	-	-	-	-	-
Initial Value		0	0	0	0	0	0	0	0

[bit7] FBT : First Byte Bit

This bit indicates the 1st byte.

FBT bit set condition:

1. The (repeated) start condition is detected.

FBT bit clear conditions:

1. The 2nd byte starts to be transmitted and received.
2. The stop condition is detected (except in cases where the stop condition is detected in the acknowledgment field during master operation).
3. I²C interface operation is disabled (ISMK:EN bit= 0).
4. A bus error is detected (IBCR:BER bit= 1) when EIBCR:BEC= 0.

Bit	Description
0	Not the 1st byte
1	The 1st byte is being transmitted and received.

[bit6] RACK : Acknowledgment Flag Bit

This bit indicates an acknowledgment received for the 1st byte, in master or slave mode.

RACK bit update conditions:

1. Acknowledgment for the 1st byte
2. Acknowledgment of data in master or slave mode

RACK bit clear conditions(RACK bit = 0):

1. The (repeated) start condition is detected.
2. I^2C interface operation is disabled (ISMK:EN bit = 0).
3. A bus error is detected (IBCR:BER bit = 1) when EIBCR:BEC= 0.

Bit	Description
0	"L" reception
1	"H" reception

[bit5] RSA : Reserved Address Detection Bit

This bit indicates that the reserved address has been detected.

RSA bit set condition (RSA= 1):

1. The 1st byte is (0000xxxx) or (1111xxxx). The letter "x" represents "0" or "1".

RSA bit reset conditions (RSA= 0)

1. The (repeated) start condition is detected.
2. The stop condition is detected (except in cases where the stop condition is detected in the acknowledgment field during master operation).
3. I^2C interface operation is disabled (ISMK:EN bit = 0).
4. A bus error is detected (IBCR:BER bit = 1) (except in cases where the stop condition is detected in the acknowledgment field during master operation).

If the RSA bit is set to "1" in the 1st byte, the interrupt flag (IBCR:INT) is set to "1" and the SCL to "L" at the SCL falling of the 8th bit in the 1st byte. This occurs regardless of whether FIFO is enabled or disabled. To read reception data and have the device operate as a slave at this time, set IBCR:ACKE to "1", and clear the interrupt flag (IBCR:INT) to "0". After that, if the TRX bit is "0", the device receives data as a slave. To prevent the device from receiving data in midstream, set the IBCR:ACKE bit to "0". Then, the device does not receive subsequent data.

Bit	Description
0	Reserved address not detected
1	Reserved address detected

Notes:

- *If IBCR:ACKE is set to "0" during data transmission, "1" is a prohibited setting for IBCR:ACKE until the stop or repeated start condition is detected.*
- *If slave transmission has been confirmed at the time of an interrupt due to the detection of the reserved address, an ACK response is returned when the reception FIFO is enabled. Therefore, disable the reception FIFO and set IBCR:ACKE= 0.*

[bit4] TRX : Data Direction Bit

This bit indicates the direction of data.

TRX bit set conditions:

1. The (repeated) start condition is transmitted in master mode.
2. The 8th bit in the 1st byte in slave mode is "1" (transmission direction as a slave).

TRX bit reset conditions:

1. Arbitration lost occurs (AL= 1).
2. The 8th bit in the 1st byte in slave mode is "0" (reception direction as a slave)
3. The 8th bit in the 1st byte in master mode is "1" (reception direction as a master).
4. The stop condition is detected (except in cases where the stop condition is detected in the acknowledgment field during master operation).
5. The (repeated) start condition is detected in a mode other than master mode.
6. I^2C interface operation is disabled (ISMK:EN bit = 0).
7. A bus error is detected (IBCR:BER bit = 1) (except in cases where the stop condition is detected in the acknowledgment field during master operation).

Bit	Description
0	Reception direction
1	Transmission direction

[bit3] AL: Arbitration Lost Bit

This bit indicates arbitration lost.

AL bit set conditions:

1. The data output in master mode differs from the received data.
2. The IBCR:MSS bit is set to "1" but the device operates as a slave.
3. The repeated start condition is detected in the 1st bit of the 2nd or subsequent byte of data in master mode when EIBCR:BEC= 0.
4. The repeated start condition is detected in master mode when EIBCR:BEC= 1.
5. The stop condition is detected in the 1st bit of the 2nd or subsequent byte of data in master mode when EIBCR:BEC= 0.
6. The stop condition is detected in master mode when EIBCR:BEC= 1 (except in cases where the stop condition is detected in the acknowledgment field).
7. Generation of the repeated start condition is attempted in master mode but fails.
8. Generation of the stop condition is attempted in master mode but fails.

AL bit reset conditions:

1. "1" is written to the IBCR:MSS bit.
2. "1" is written to the IBCRC:INTC bit.
3. "1" is written to the IBSRC:SPCC bit when the AL bit is "1" and the SPC bit is "1".
4. I^2C interface operation is disabled (ISMK:EN bit= 0).
5. A bus error is detected (IBCR:BER bit= 1) when EIBCR:BEC= 0.

Bit	Description
0	No arbitration lost occurred.
1	Arbitration lost occurred.

[bit2] RSC: Repeated Start Condition Check Bit

This bit indicates that the repeated start condition has been detected in master or slave mode.

RSC bit set conditions:

1. The repeated start condition is detected after an acknowledgment during operation in slave or master mode when EIBCR:BEC= 0.
2. The repeated start condition is detected in the 1st byte during operation in slave or master mode when EIBCR:BEC= 1.

RSC bit reset conditions:

1. "1" is written to the IBSRC:RSCC bit.
2. "1" is written to the IBCR:MSS bit.
3. I^2C interface operation is disabled (ISMK:EN bit = 0).

Writing to this bit is invalid.

Bit	Description
0	The repeated start condition was not detected.
1	The repeated start condition was detected.

Note:

- *If no acknowledgment response is returned during the reception operation in slave mode following the detection of the reserved address, slave mode ends. Consequently, even when the repeated start condition is next detected, this bit is not set to "1".*

[bit1] SPC: Stop Condition Check Bit

This bit indicates that the stop condition has been detected in master or slave mode.

SPC bit set conditions:

1. The stop condition is detected after an acknowledgment during operation in slave or master mode when EIBCR:BEC= 0.
2. The stop bit is detected in any of the following cases when EIBCR:BEC= 1:
   - During transfer of the 1st byte
   - Slave mode operation in progress
   - Master mode operation in progress (except in cases where the stop condition is detected in the acknowledgment field)
3. Arbitration lost occurs in the stop condition generation operation in master mode.

SPC bit reset conditions:

1. "1" is written to the IBSRC:SPCC bit.
2. "1" is written to the IBCR:MSS bit.
3. I^2C interface operation is disabled (ISMK:EN bit = 0).

Writing to this bit is invalid.

Bit	Description	
0	The stop condition was not detected.	
1	Master	The stop condition was detected, or arbitration lost occurred when the stop condition was output.
	Slave	The stop condition was detected.

Notes:

- *If no acknowledgment response is returned during the reception operation in slave mode following the detection of the reserved address, slave mode ends. Consequently, even when the stop condition is next detected, this bit is not set to "1".*
- *If all of the following conditions are true, this bit is not set to "1", and master operation continues even when the stop condition is detected:*
  - EIBCR:BEC=1
  - Master operation in progress
  - Acknowledgment field transfer in progress

[bit0] BB : Bus Status Bit

This bit indicates the bus status.

BB bit set condition:

1. "L" is detected in the SDA or SCL of the I^2C bus.

BB bit reset conditions:

1. The stop condition is detected (except in cases where the stop condition is detected in the acknowledgment field during master operation).
2. I^2C interface operation is disabled (ISMK:EN bit = 0).
3. A bus error is detected (IBCR:BER bit = 1) when EIBCR:BEC= 0.

Bit	Description
0	Bus idle state
1	Bus transmission and reception state

8.4. Serial Status Register (SSR)

The serial status register (SSR) is used to check the transmission and reception status.

Serial Status Register (SSR)

Figure 8-4 shows the bit configuration of the serial status register (SSR).

Figure 8-4 Bit Configuration of the Serial Status Register (SSR)

Bit	15	14	13	12	11	10	9	8	7 ... 0
Field	REC	TSET	DMA	TBIE	ORE	RDRF	TDRE	TBI	(IBSR)
R/W Attribute	R0,W	R0,W	R/W	R/W	R,WX	R,WX	R,WX	R,WX	
Protection Attribute	-	-	-	-	-	-	-	-	
Initial Value	0	0	0	0	0	0	1	1	

[bit15] REC : Reception Error Flag Clear Bit

This bit clears the ORE bit in the serial status register (SSR).

- This bit is set when the SSRS:RECS bit in the set register is set to "1".
- Writing "1" clears the ORE bit.
- Writing "0" does not have any effect.

If it is read, "0" is always read.

Bit	Description	
	Write	Read
0	No effect	"0" is always read.
1	Clear the reception error flag (ORE).	

[bit14] TSET : Transmission Buffer Empty Flag Set Bit

This bit sets the TDRE bit in the serial status register (SSR).

- Writing "1" sets the TDRE bit. This also sets the TBI bit when DMA mode is enabled (DMA= 1).
- Writing "0" does not have any effect.
- This bit is set when the SSRS:TSETS bit in the set register is set to "1".

If it is read, "0" is always read.

Bit	Description	
	Write	Read
0	No effect	"0" is always read.
1	Set the TDRE bit.	

Note:

- *Write "1" to this bit when the IBCR:INT bit is "1".*

[bit13] DMA: DMA Mode Enable Bit

This bit disables/enables DMA mode.

- This bit is reset when the SSRC:DMAC bit in the clear register is set to "1".
- This bit is set when the SSRS:DMAS bit in the set register is set to "1".

- If this bit is set to "1", it means interrupt conditions corresponding to DMA transfer.
- If this bit is set to "0", it means interrupt conditions for cases where DMA transfer is not performed.

For details, see Table 2-1.

Bit	Description
0	Disable DMA mode.
1	Enable DMA mode.

Notes:

- *This bit can be changed only when ISMK:EN= 0.*
- *If block transfer is performed during DMA transfer, the only block size that can be set is 1.*

[bit12] TBIE : Transmission Bus Idle Interrupt Enable Bit (Valid Only when DMA Mode is Enabled)

- This bit enables/disables transmission bus idle interrupt request output to the CPU.
- A transmission bus idle interrupt request is output when DMA mode is enabled (DMA= 1) and the TBIE and TBI bits are "1".
- This bit is set to "0" when DMA mode is disabled (DMA= 0), even during writing, in which case the writing is ignored and the "0" state is retained.
- This bit is reset when the SSRC:TBIEC bit in the clear register is set to "1".
- This bit is set when the SSRS:TBIES bit in the set register is set to "1".

Bit	Description
0	Disable transmission bus idle interrupts.
1	Enable transmission bus idle interrupts.

[bit11] ORE : Overrun Error Flag Bit

- This bit is set to "1" when an overrun occurs in reception, and it is cleared by the writing of "1" to the REC bit in the serial status register (SSR).
- A reception interrupt request is output when the ORE bit and the SMR:RIE bit are "1".
- For details on interrupt request output, see Table 2-3.
- If this flag is set, the reception data register (RDR) is invalid.
- If this flag is set, no reception data is stored in the reception FIFO when used.

Bit	Description
0	No overrun error
1	Overrun error found

[bit10] RDRF : Reception Data Full Flag Bit

- This flag indicates the status of the reception data register (RDR).
- A reception interrupt request is output when the SMR:RIE bit and the reception data flag bit (RDRF) are "1".
- This flag is set to "1" when reception data is loaded into the reception data register (RDR), and it is cleared to "0" when the data in the RDR is read.
- This bit is set at the SCL falling time of the 8th bit of data.
- This bit is also set when a NACK response*1 is returned.
- RDRF is set to "1" when the predefined quantity of data is received by the reception FIFO in use.
- When ECR:RXBLKEN=0 and reading of the reception FIFO empties it during use, this bit is cleared to "0".
- When ECR:RXBLKEN=1 and using reception FIFO, this bit is clear to "0" when the number of the data in the reception FIFO is less than or equal to the value of the FBYTE register.
- If all of the following conditions are satisfied and the reception idle state continues for 8 or more clocks of the baud rate clock, the interrupt flag (SSR:RDRF) is set to "1".
  - The reception FIFO idle detection enable bit (FCR:FRIIE) is "1".
  - The data count in the reception FIFO has not reached the transfer count.
  - The IBCR:BER bit is "0".

  During an 8-clock count, the counter is reset to 0 when the RDR is read, and the system starts counting the 8 clocks again.

  *1 NACK response: Indicates that the SDA of the I^2C bus is "H" during an acknowledgment period.

Bit	Description
0	The reception data register (RDR) is empty.
1	The reception data register RDR contains data.

Notes:

- *After RDRF becomes "1" when the reception FIFO is used, resetting the reception FIFO (FCR0:FCL2,FCL1= 1) does not cause RDRF to become "0". Therefore, to set RDRF to "0" after resetting the reception FIFO, perform a dummy read on the reception data register while I2C mode is disabled (ISMK.I2CEN=0).*

- *If all of the following conditions are satisfied, the SCL is set to "L" after ACK transmission. The SCL is released from the "L" state when the RDRF bit is set to "0".*
  - The reception FIFO is not used.
  - DMA mode is enabled (IBCR:DMA= 1).
  - The RDRF bit is "1" during the reception operation (IBSR:TRX= 0) for the 2nd or subsequent byte.
  - IBCR:WSEL=0
- *If all of the following conditions are satisfied, the SCL is set to "L" after the reception of 1-byte data. The SCL is released from the "L" state when the RDRF bit is set to "0".*
  - The reception FIFO is not used.
  - DMA mode is enabled (IBCR:DMA= 1).
  - The RDRF bit is "1" during the reception operation (IBSR:TRX= 0) for the 2nd or subsequent byte.

- IBCR:WSEL=1

- *If the reception FIFO is used to receive data with DMA mode enabled (DMA= 1), the SCL is set to "L" when the reception FIFO becomes full. The SCL is released from the "L" state when data is read from the RDR at least once.*

[bit9] TDRE : Transmission Data Empty Flag Bit

- This bit indicates the status of the transmission data register (TDR).
- A transmission interrupt request is output when the SMR:TIE bit and the TDRE bit are "1".
- Writing transmission data in the TDR sets this bit to "0", which indicates that the TDR contains valid data. Loading the data into the transmission shift register to start transmission sets this bit to "1", which indicates that TDR does not contain valid data.
- This bit is set when "1" is written to the TSET bit in the serial status register (SSR). Use it when the desired setting for the TDRE bit is "1" at the detection time of arbitration lost, a bus error, etc.

Bit	Description
0	The transmission data register (TDR) contains data.
1	The transmission data register is empty.

[bit8] TBI: Transmission Bus Idle Flag Bit (Valid Only when DMA Mode is Enabled)

This bit indicates that the I^2C does not perform the transmission operation when DMA mode is enabled (DMA= 1). The SCL is set to “L” when DMA mode is enabled (DMA= 1) and the TBI bit is set to "1". Writing transmission data to the TDR sets the TBI bit to "0" and releases the SCL from the “L” state.

TBI bit set conditions:

<8th bit>

1. The SSR:TDRE bit is "1" at the 2nd or subsequent byte during master operation when WSEL is "1".
2. The SSR:TDRE bit is "1" at the 2nd or subsequent byte during slave transmission when WSEL is "1".

<9th bit>

1. The reserved address is not detected in the 1st byte and the SSR:TDRE bit is "1" during master operation.
2. The SSR:TDRE bit is "0" at the 2nd or subsequent byte during master operation when WSEL is "1".
3. The SSR:TDRE bit is "0" at the 2nd or subsequent byte during slave transmission when WSEL is "1".

<Other>

"1" is written to the transmission buffer empty flag set bit (TSET).

TBI bit reset condition:

1. Transmission data is written to the transmission data register (TDR).

A transmission interrupt request is output when this bit is set to "1" and transmission bus idle interrupts are enabled (SSR:TBIE= 1).

■ This bit is "1" when DMA mode is disabled (DMA= 0).

Bit	Description
0	Transmission in progress
1	No transmission

8.5. Reception Data Register/Transmission Data Register (RDR/TDR)

The reception and transmission data registers are allocated to the same address. This register functions as the reception data register when data is read, and as the transmission data register when data is written. When the FIFO mode is enabled, RDR/TDR register address is the address of reading/writing of the FIFO.

Reception Data Register (RDR)

Figure 8-5 shows the bit configuration of the register for serial data reception (RDR).

Figure 8-5 Bit Configuration of the Reception Data Register (RDR)

Bit	15 … 8	7	6	5	4	3	2	1	0
Field	-	D7	D6	D5	D4	D3	D2	D1	D0
R/W Attribute		R	R	R	R	R	R	R	R
Protection Attribute		-	-	-	-	-	-	-	-
Initial Value		0	0	0	0	0	0	0	0

The reception data register (RDR) is the data buffer register used to receive serial data.

- The shift register converts serial data signals transmitted to the serial data line (SDA pin), and they are stored in the reception data register (RDR).
- When the 1st byte*1 is received, the lowest bit (RDR:D0) indicates the data direction.
- The reception data full flag bit (SSR:RDRF) is set to "1" when reception data is stored in the reception data register (RDR).
- The reception data full flag bit (SSR:RDRF) is automatically cleared to "0" when the data in the reception data register (RDR) is read.
- *1 1st byte: Indicates the data after the (repeated) start condition.

Notes:

- *SSR:RDRF is set to "1" when the predefined quantity of data is received by the reception FIFO in use.*
- *SSR:RDRF is cleared to "0" when the reception FIFO becomes empty during use.*

Transmission Data Register (TDR)

Figure 8-6 shows the bit configuration of the transmission data register.

Figure 8-6 Bit Configuration of the Transmission Data Register (TDR)

Bit	15 ... 8	7	6	5	4	3	2	1	0
Field	-	D7	D6	D5	D4	D3	D2	D1	D0
R/W Attribute		W	W	W	W	W	W	W	W
Protection Attribute		-	-	-	-	-	-	-	-
Initial Value		1	1	1	1	1	1	1	1

The transmission data register (TDR) is the data buffer register used to transmit serial data.

- The transmission data register (TDR) value is output MSB first to the serial data line (SDA pin).
- When the 1st byte*1 is transmitted, the lowest bit (TDR:D0) indicates the data direction.
- The transmission data empty flag (SSR:TDRE) is cleared to "0" when transmission data is written to the transmission data register (TDR).
- The transmission data empty flag (SSR:TDRE) is set to "1" when data is transferred to the transmission shift register.
- Transmission data cannot be written to the transmission data register (TDR) when the transmission FIFO is disabled and the data empty flag (SSR:TDRE) is "0".
- Transmission data can be written to the transmission FIFO in use until the transmission FIFO capacity is reached, even when the data empty flag (SSR:TDRE) is "0".

Note:

– *The transmission data register is a write-only register, and the reception data register is a read-only register. The two registers are located at the same address, so the written value differs from the read one.*

8.6. Serial Auxiliary Control Status Register (SACSR)

The serial auxiliary control status register (SACSR) can select the method of serial timer activation, enable/disable timer interrupts, set the division value of the serial timer operation clock, and enable/disable the serial timer.

Bit Configuration of the Serial Auxiliary Control Status Register (SACSR)

Figure 8-7 shows the bit configuration of the serial auxiliary control status register (SACSR).

Figure 8-7 Bit Configuration of the Serial Auxiliary Control Status Register (SACSR)

Bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved	Reserved	Reserved	Reserved	TRG1	TRG0	TINT
R/W Attribute	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R/W	R/W	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	TINTE	Reserved	TRGE	TDIV3	TDIV2	TDIV1	TDIV0	TMRE
R/W Attribute	R/W	R0,W0	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

[bit15:11] Reserved : Reserved Bits

[bit10:9] TRG[1:0] : Trigger Selection Bits

These bits select the method of detecting the edge for an external trigger of serial timer activation.

Bits		Description
0	0	Rising edge detection
0	1	Falling edge detection
1	0	Both edge detection
1	1	Setting prohibited

Notes:

- *These bits can be changed only when the serial timer enable bit (TMRE) is "0".*
- *These bits are invalid when the external trigger enable bit (TRGE) is "0".*

[bit8] TINT : Timer Interrupt Flag

If the serial timer register (STMR) and the serial timer comparison register (STMCR) match, the serial timer register (STMR) is set to "0", and this bit is set to "1".

A status interrupt request is output when this bit is "1" and the timer interrupt enable bit (TINTE) is "1".

Writing "0" to this bit resets it to "0".

Writing "1" to this bit is invalid.

- This bit is reset when the SACSRC:TINTC bit in the clear register is set to "1".

Bit	Description
0	No timer interrupt request
1	Timer interrupt request issued

[bit7] TINTE : Timer Interrupt Enable Bit

This bit enables/disables timer interrupts to the CPU.

A status interrupt request is output when this bit is "1" and the timer interrupt flag (TINT) is "1".

- This bit is reset when the SACSRC:TINTEC bit in the clear register is set to "1".
- This bit is set when the SACSRS:TINTES bit in the set register is set to "1".

Bit	Description
0	Disable interrupts triggered by the serial timer.
1	Enable interrupts triggered by the serial timer.

[bit6] Reserved : Reserved Bit

[bit5] TRGE : External Trigger Enable Bit

This bit selects the method of activating the serial timer.

- This bit is reset when the SACSRC:TRGEC bit in the clear register is set to "1".
- This bit is set when the SACSRS:TRGES bit in the set register is set to "1".

Bit	Description
0	The serial timer starts when the serial timer enable bit (TMRE) changes from "0" to "1".
1	If the serial timer enable bit (TMRE) is set to "1", the detection of the edge of the external trigger that is set in the trigger selection bits (TRG1, 0) starts the serial timer.

Notes:

- *This bit can be changed only when the serial timer enable bit (TMRE) is "0".*
- *If the serial timer enable bit (TMRE) is set to "0", even the detection of the edge of the external trigger that is set in the trigger selection bits (TRG1, 0) does not start the serial timer.*

[bit4-1] TDIV[3:0] : Timer Operation Clock Division Bits

These bits set the division ratio of the serial timer.

Bits				Timer Operation Clock						
				Division Ratio	Φ= 8 MHz	Φ= 10 MHz	Φ= 16 MHz	Φ= 20 MHz	Φ= 24 MHz	Φ= 32 MHz
0	0	0	0	Φ	125 ns	100 ns	62.5 ns	50 ns	41.67 ns	31.25 ns
0	0	0	1	Φ/2	250 ns	200 ns	125 ns	100 ns	83.33 ns	62.5 ns
0	0	1	0	Φ/4	500 ns	400 ns	250 ns	200 ns	166.67 ns	125 ns
0	0	1	1	Φ/8	1 us	800 ns	500 ns	400 ns	333.33 ns	250 ns
0	1	0	0	Φ/16	2 us	1.6 us	1 us	800 ns	666.67 ns	500 ns
0	1	0	1	Φ/32	4 us	3.2 us	2 us	1.6 us	1.33 us	1 us
0	1	1	0	Φ/64	8 us	6.4 us	4 us	3.2 us	2.67 us	2 us
0	1	1	1	Φ/128	16 us	12.8 us	8 us	6.4 us	5.33 us	4 us
1	0	0	0	Φ/256	32 us	25.6 us	16 us	12.8 us	10.67 us	8 us

Φ: Bus clock

Notes:

- *These bits can be changed only when the serial timer enable bit (TMRE) is "0".*
- *Settings other than the above are prohibited.*

[bit0] TMRE : Serial Timer Enable Bit

This bit enables or disables serial timer operation.

- This bit is reset when the SACSRC:TMREC bit in the clear register is set to "1".
- This bit is set when the SACSRS:TMRES bit in the set register is set to "1".

Bit	Description
0	Stop the operation of the serial timer. The value of the serial timer register (STMR), when stopped, is retained.
1	Initialize the serial timer register (STMR) to "0" and start the operation of the serial timer when this bit changes from "0" to "1".

Note:

- *When external triggers are enabled (TRGE= 1), setting this bit to "1" does not start the serial timer until the detection of the edge of the external trigger that is set in the trigger selection bits (SACSR:TRG1, 0).*

8.7. Serial Timer Register (STMR)

The serial timer register (STMR) represents the timer value of the serial timer.

Bit Configuration of the Serial Timer Register (STMR)

Figure 8-8 shows the bit configuration of the serial timer register (STMR).

Figure 8-8 Bit Configuration of the Serial Timer Register (STMR)

Bit	15	14	13	12	11	10	9	8
Field	TM15	TM14	TM13	TM12	TM11	TM10	TM9	TM8
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	TM7	TM6	TM5	TM4	TM3	TM2	TM1	TM0
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

[bit15:0] TM15-0 : Timer Data Bits

These bits indicate the timer value of the serial timer.

While the timer is operating, the timer value of the serial timer is incremented by 1 for every timer operation clock (as set in SACSR:TDIV3 to 0).

Note:

- *These bits are initialized to "0" when timer operation starts.*

8.8. Serial Timer Comparison Register (STMCR)

The serial timer comparison register (STMCR) sets the timer comparison value of the serial timer.

Bit Configuration of the Serial Timer Comparison Register (STMCR)

Figure 8-9 shows the bit configuration of the serial timer comparison register (STMCR).

Figure 8-9 Bit Configuration of the Serial Timer Comparison Register (STMCR)

Bit	15	14	13	12	11	10	9	8
Field	TC15	TC14	TC13	TC12	TC11	TC10	TC9	TC8
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	TC7	TC6	TC5	TC4	TC3	TC2	TC1	TC0
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

[bit15:0] TC15-0 : Compare Bits

These bits set the comparison value of the serial timer.

These bits are compared with the serial timer register (STMR). If the serial timer register (STMR) matches these bits when updated, the serial timer register is set to "0". At this time, the timer interrupt flag (SACSR:TINT) is set to "1".

The interval of the following operation is (STMCR:TC + 1) * timer operation clock (as set in SACSR:TDIV3 to 0):

- Setting SACSR:TINT to "1"

Notes:

- *If this register is set to 0x0000, the timer is operating, and the division value (SACSR:TDIV) of the timer operation clock is set to 0b0000, the timer interrupt flag (SACSR:TINT) is fixed at "1".*
- *This register can be changed only when the serial timer is disabled (SACSR:TMRE= 0).*

8.9. Noise Filter Control Register (NFCR)

The noise filter control register (NFCR) sets the noise filter time.

Noise Filter Control Register (NFCR)

Figure 8-10 shows the bit configuration of the noise filter control register (NFCR).

Figure 8-10 Bit Configuration of the Noise Filter Control Register (NFCR)

Bit	15 ... 8	7	6	5	4	3	2	1	0
Field	(EIBCR)	Reserved	Reserved	Reserved	NFT4	NFT3	NFT2	NFT1	NFT0
R/W Attribute		R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W	R/W
Protection Attribute		-	-	-	-	-	-	-	-
Initial Value		0	0	0	0	0	0	0	0

[bit7:5] Reserved : Reserved Bits

[bit4:0] NFT : Noise Filter Time Selection Bits

These bits select the noise filter time for serial clock input (SCL) and serial data input (SDA).

The calculation formula for the noise filter time is as follows:

Noise filter time = (NFT + 1) * 2 * bus clock cycle time

Table 8-2 shows the relationship between the noise filter time selection bits and bus clock frequency. Adjust the noise filter time selection bits according to the bus clock frequency.

Notes:

- *Set these bits when the EN bit in the ISMK register is "0".*
- *Combinations not listed in Table 8-2 are prohibited.*

Table 8-2 Relationship between the Noise Filter Time Selection Bits and Bus Clock Frequency

Bits					Bus Clock Frequency [MHz]
0	0	0	0	0	8 MHz to less than 40 MHz*1
0	0	0	0	1	40 MHz to less than 60 MHz
0	0	0	1	0	60 MHz to less than 80 MHz
0	0	0	1	1	80 MHz to less than 100 MHz
0	0	1	0	0	100 MHz to less than 120 MHz
0	0	1	0	1	120 MHz to less than 140 MHz
0	0	1	1	0	140 MHz to less than 160 MHz
0	0	1	1	1	160 MHz to less than 180 MHz
0	1	0	0	0	180 MHz to less than 200 MHz
0	1	0	0	1	200 MHz to less than 220 MHz
0	1	0	1	0	220 MHz to less than 240 MHz
0	1	0	1	1	240 MHz to less than 260 MHz
0	1	1	0	0	260 MHz to less than 280 MHz
0	1	1	0	1	280 MHz to less than 300 MHz
0	1	1	1	0	300 MHz to less than 320 MHz
0	1	1	1	1	320 MHz to less than 340 MHz
1	0	0	0	0	340 MHz to less than 360 MHz
1	0	0	0	1	360 MHz to less than 380 MHz
1	0	0	1	0	380 MHz to less than 400 MHz

*1 In standard mode, this is 2 MHz to less than 40 MHz.

8.10. Extended I²C Bus Control Register (EIBCR)

The extended I²C bus control register (EIBCR) controls SDA/SCL output and sets whether to continue operation after a bus error occurs.

Extended I²C Bus Control Register (EIBCR)

Figure 8-11 shows the bit configuration of the extended I²C bus control register (EIBCR).

Figure 8-11 Bit Configuration of the Extended I²C Bus Control Register (EIBCR)

Bit	15	14	13	12	11	10	9	8	7 ... 0
Field	Reserved	Reserved	SDAS	SCLS	SDAC	SCLC	SOCE	BEC	(NFCR)
R/W Attribute	R0,W0	R0,W0	R,WX	R,WX	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	1	1	0	0	

[bit15-14] Reserved : Reserved Bits

[bit13] SDAS : SDA Status Bit

This bit indicates the signal level of the SDA line after a signal passes through the noise filter.

Bit	Description
0	The SDA line is "L".
1	The SDA line is "H".

Note:

- *This bit is valid only when the I²C is enabled (ISMK:EN= 1). When the I²C is disabled (ISMK:EN= 0), this bit always reads "0".*

[bit12] SCLS : SCL Status Bit

This bit indicates the signal level of the SCL line after a signal passes through the noise filter.

Bit	Description
0	The SCL line is "L".
1	The SCL line is "H".

Note:

- *This bit is valid only when the I²C is enabled (ISMK:EN= 1). When the I²C is disabled (ISMK:EN= 0), this bit always reads "0".*

[bit11] SDAC : SDA Output Control Bit

This bit controls SDA output when serial output control is enabled (SOCE= 1).

Bit	Description
0	SDA output is "L".
1	SDA output is "H".

[bit10] SCLC : SCL Output Control Bit

This bit controls SCL output when serial output control is enabled (SOCE= 1).

Bit	Description
0	SCL output is "L".
1	SCL output is "H".

[bit9] SOCE : Serial Output Enable Bit

This bit enables serial output control.

If this bit is set to "1", operation is as follows.

- The SDA output control bit (SDAC) controls SDA output.
- The SCL output control bit (SCLC) controls SCL output.

Bit	Description
0	Disable serial output control.
1	Enable serial output control.

Note:

- *Set this bit to "1" only when IBCR:MSS= 0 and IBCR:ACT= 0.*

[bit8] BEC : Bus Error Control Bit

This bit selects whether to continue or stop I^2C operation after a bus error occurs (IBSR:BER= 1).

Bit	Description
0	Stop I^2C operation.
1	Continue I^2C operation.

Note:

- *If the start condition is detected again during address data transfer after the start condition was detected or during transfer of bit2 to bit9 (acknowledgment bits) when EIBCR:BEC= 0, a bus error is detected (IBCR:BER= 1) and reception is stopped. Consequently, the next data cannot be received.*
  In this case, after the interrupt flag (IBCR:INT) is cleared, retransmission of the start condition from the master is necessary.

8.11. 7-Bit Slave Address Mask Register (ISMK)

The 7-bit slave address mask register (ISMK) specifies whether to compare slave address bits.

7-Bit Slave Address Mask Register (ISMK)

Figure 8-12 shows the bit configuration of the 7-bit slave address mask register (ISMK).

Figure 8-12 Bit Configuration of the 7-Bit Slave Address Mask Register (ISMK)

Bit	15	14	13	12	11	10	9	8	7 ... 0
Field	EN	SM6	SM5	SM4	SM3	SM2	SM1	SM0	(ISBA)
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Protection Attribute	-	-	-	-	-	-	-	-	
Initial Value	0	1	1	1	1	1	1	1	

[bit15] EN : I²C Interface Operation Enable Bit

This bit enables/disables I²C interface operation.

"0": Disable I²C interface operation.
"1": Enable I²C operation.

Bit	Description
0	Disable
1	Enable

Notes:

- *This bit is not cleared to "0" even when the BER bit in the IBCR register is set to "1".*
- *Set the baud rate generator when this bit is "0".*
- *Set the 7-bit slave address and 7-bit slave address mask registers when this bit is "0".*
- *Transmission and reception are disabled immediately after the I²C interface is disabled (EN= 0).*
- *If you want to disable I²C interface operation after generating the stop condition by writing "0" to the IBCR:MSS bit, verify that the stop condition is generated, and then disable operation (EN= 0).*
- *If the EN bit is set to "0" during transmission, a pulse may be generated on the SDA/SCL of the I²C bus.*

[bit14:8] SM[n] n=6 to 0 : Slave Address Mask Bits

These bits specify whether a 7-bit slave address and the received address are excluded from comparison.

Bits set to "1": Compare them.
Bits set to "0": Handle them as a match.

SM[n] n=6 to 0	Description
0	Do not compare bits.
1	Compare bits.

Note:

- *Set this register when the EN bit is "0".*

8.12. 7-Bit Slave Address Register (ISBA)

The 7-bit slave address register (ISBA) sets a slave address.

7-Bit Slave Address Register (ISBA)

Figure 8-13 shows the bit configuration of the 7-bit slave address register (ISBA).

Figure 8-13 Bit Configuration of the 7-Bit Slave Address Register (ISBA)

Bit	15 ... 8	7	6	5	4	3	2	1	0
Field	(ISMK)	SAEN	SA6	SA5	SA4	SA3	SA2	SA1	SA0
R/W Attribute		R/W	R,W	R,W	R,W	R,W	R,W	R,W	R,W
Protection Attribute		-	-	-	-	-	-	-	-
Initial Value		0	0	0	0	0	0	0	0

[bit7] SAEN : Slave Address Enable Bit

This bit enables slave address detection.

"0": Do not detect the slave address.

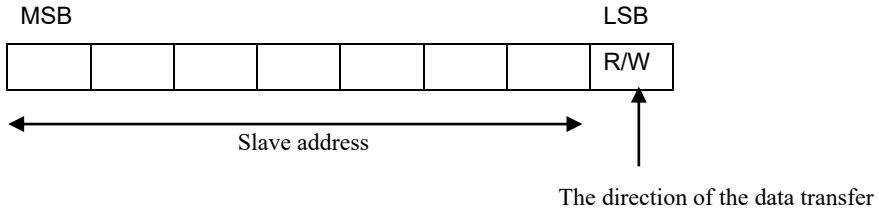
"1": Compare the ISBA and ISMK settings with the 1st byte received.

Bit	Description
0	Disable
1	Enable

[bit6:0] SA[6:0] : 7-Bit Slave Address

- If slave address detection is enabled (SAEN= 1), the 7-bit slave address register (ISBA) compares the 7-bit data received after the detection of the (repeated) start condition with this register. If all bits match, the device operates in slave mode, and ACK is output. Then, the received slave address is set in this register. (If SAEN= 0, ACK is not output.)
- The 1st received data after detecting the (repeated) start condition consists of 7-bit slave address and the bit that specifies the direction of data transfer. It compares the slave address included in the received data to these bits.
- The address bit for which "0" is set in ISMK is excluded from the comparison.

Figure 8-14 The 1st byte format after detecting the (repeated) start condition



Compare the slave address with the 7-bit slave address register (ISBA)

SA[6:0]	Description
Write / Read	7-bit slave address

Notes:

- *Setting the reserved address is prohibited.*
- *Set this register when the EN bit in the ISMK register is "0".*

8.13. Baud Rate Generator Registers 1, 0 (BGR1, BGR0)

Baud rate generator registers 1, 0 (BGR1, BGR0) set the division ratio of the serial clock.

Bit Configuration of Baud Rate Generator Registers 1 and 0 (BGR1, BGR0)

Figure 8-15 shows the bit configuration of baud rate generator registers 1 and 0 (BGR1, BGR0).

Figure 8-15 Bit Configuration of Baud Rate Generator Registers 1 and 0 (BGR1, BGR0)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved	BGR1							BGR0							
R/W Attribute	R0, W0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The baud rate generator registers set the division ratio of the serial clock.
BGR1 represents the upper bits, and BGR0 represents the lower bits. The reload value for counting can be written to these registers, and the setting value can be read.
The reload counter starts counting when a reload value is written to baud rate generator registers 0 and 1 (BGR1, BGR0).

[bit15] Reserved : Reserved Bit

[bit14:8] BGR1 : Baud Rate Generator Register 1

BGR1	Description
Write	Write a value to reload counter bit8 to 14.
Read	Read the BGR1 setting value.

[bit7:0] BGR0 : Baud Rate Generator Register 0

BGR0	Description
Write	Write a value to reload counter bit0 to 7.
Read	Read the BGR0 setting value.

Notes:

- *Use 16-bit access when writing a value to the baud rate generator registers (BGR1, BGR0).*
- *Set the baud rate generator registers when the EN bit in the ISMK register is "0".*
- *Set a baud rate regardless of whether the mode is master or slave.*
- *In operation mode 4 (I^2C mode), use the bus clock at 8 MHz or more. A baud rate generator setting exceeding 400 kbps is prohibited in this mode.*

8.14. FIFO Control Register 1 (FCR1)

FIFO control register 1 (FCR1) sets the FIFO test, selects the transmission and reception FIFOs, enables transmission FIFO interrupts, and controls the interrupt flag.

Bit Configuration of FIFO Control Register 1 (FCR1)

Figure 8-16 shows the bit configuration of FIFO control register 1 (FCR1).

Figure 8-16 Bit Configuration of FIFO Control Register 1 (FCR1)

Bit	15	14	13	12	11	10	9	8	7 ... 0
Field	Reserved	Reserved	Reserved	FLSTE	FRIIE	FDRQ	FTIE	FSEL	(FCR0)
R/W Attribute	R0,W0	R0,W0	R0,W0	R/W	R/W	R,WX	R/W	R/W	
Protection Attribute	-	-	-	-	-	-	-	-	
Initial Value	0	0	0	0	0	1	0	0	

[bit15:13] Reserved : Reserved Bits

[bit12] FLSTE : Retransmission Data Lost Detection Enable Bit

This bit enables the detection of the FIFO Retransmission Data Lost Flag Bit (FCR0:FLST).

- This bit is reset when the FCR1C:FLSTEC bit in the clear register is set to "1".
- This bit is set when the FCR1S:FLSTES bit in the set register is set to "1".

"0": Disable FCR0:FLST bit detection.
"1": Enable FCR0:FLST bit detection.

Bit	Description
0	Disable data lost detection.
1	Enable data lost detection.

Note:

- *To set this bit to "1", set the FSET bit to "1" first, and then set this bit to "1".*

[bit11] FRIIE : Reception FIFO Idle Detection Enable Bit

When the reception FIFO has valid data, this bit enables or disables detection of the reception idle state that continues for an 8-bit time or longer. If reception interrupts are enabled (SMR:RIE= 1), the detection of the reception idle state generates a reception interrupt.

- This bit is reset when the FCR1C:FRIIEC bit in the clear register is set to "1".
- This bit is set when the FCR1S:FRIIES bit in the set register is set to "1".

"0": Disable detection of the reception idle state.
"1": Enable detection of the reception idle state.

Bit	Description
0	Disable detection of reception FIFO idle.
1	Enable detection of reception FIFO idle.

Note:

- *To use the reception FIFO, set this bit to "1".*

[bit10] FDRQ : Transmission FIFO Data Request Bit

This bit requests transmission FIFO data.
"1" in this bit indicates that transmission data is requested. At this time, if transmission interrupts are enabled (FTIE= 1), a transmission FIFO interrupt request is output .

FDRQ set conditions

- When transmission FIFO interrupt control is not used
  - FBYTE (for transmission) = 0 (The transmission FIFO is empty.)
  - The transmission FIFO is reset.
- When transmission FIFO interrupt control is used
  - When ECR:TXBLKEN= 0
    - FTICR setting value ≥ FTICR read value (The data count stored in the transmission FIFO is at the interrupt trigger level or lower.)
  - When ECR:TXBLKEN= 1
    - FTICR setting value ≤ Free data count in transmission FIFO
  - The transmission FIFO is reset.

FDRQ reset conditions

- "1" is written to the FCR1C:FDRQC bit.
- When ECR:TXBLKEN= 0
  - The transmission FIFO becomes full.
- When ECR:TXBLKEN= 1
  - FTICR setting value ≥ Free data count in transmission FIFO

Bit	Description
0	No transmission FIFO data request
1	Transmission FIFO data request issued

Notes:

- *Writing "0" to this bit is inhibited when FBYTE(for transmission) = "0".*
- *The FSEL bit cannot be changed when this bit is "0".*
- *Writing to this bit is invalid.*
- *If a transmission interrupt is generated, and the necessary data is written to the transmission FIFO, write "1" to the FIFO transmission data request clear bit (FCR1C:FDRQC) to clear the interrupt request.*

[bit9] FTIE : Transmission FIFO Interrupt Enable Bit

This bit enables a transmission FIFO interrupt. If this bit is set to "1" when the FDRQ bit is "1", an interrupt is generated.

- This bit is reset when the FCR1C:FTIEC bit in the clear register is set to "1".
- This bit is set when the FCR1S:FTIES bit in the set register is set to "1".

Bit	Description
0	Disable transmission FIFO interrupts.
1	Enable transmission FIFO interrupts.

Note:

- *Set FTIE=0 if FIFO is not used when block transfer.*

[bit8] FSEL : FIFO Selection Bit

This bit selects the transmission and reception FIFOs.

- This bit is reset when the FCR1C:FSELC bit in the clear register is set to "1".
- This bit is set when the FCR1S:FSELS bit in the set register is set to "1".

"0": Allocate FIFO1 as the transmission FIFO and FIFO2 as the reception FIFO.
"1": Allocate FIFO2 as the transmission FIFO and FIFO1 as the reception FIFO.

Bit	Description
0	Transmission FIFO: FIFO1; reception FIFO: FIFO2
1	Transmission FIFO: FIFO2; reception FIFO: FIFO1

Notes:

- *This bit is not cleared by a FIFO reset (FCL2, FCL1= 1).*
- *To change the value of this bit, disable FIFO operation (FCR0:FE2, FE1= 0) first.*
- *This bit cannot be changed when FDRQ= 0.*
- *Set the FIFO Selection Bit (FCR1:FSEL) before setting the FIFO byte register (FBYTE) and teh Transmission FIFO Interrupt Control Register (FTICR).*
- *It can not access the FIFO byte register (FBYTE) at the same access.*

8.15. FIFO Control Register 0 (FCR0)

FIFO control register 0 (FCR0) enables/disables FIFO operation, performs a FIFO reset, saves the read pointer, and sets retransmission.

Bit Configuration of FIFO Control Register 0 (FCR0)

Figure 8-17 shows the bit configuration of FIFO control register 0 (FCR0).

Figure 8-17 Bit Configuration of FIFO Control Register 0 (FCR0)

Bit	15 ... 8	7	6	5	4	3	2	1	0
Field	(FCR1)	Reserved	FLST	FLD	FSET	FCL2	FCL1	FE2	FE1
R/W Attribute		R0,W0	R,WX	R,W	R0,W	R0,W	R0,W	R/W	R/W
Protection Attribute		-	-	-	-	-	-	-	-
Initial Value		0	0	0	0	0	0	0	0

[bit7] Reserved : Reserved Bit

[bit6] FLST : FIFO Retransmission Data Lost Flag Bit

This bit indicates that the retransmission data of the transmission FIFO has been lost.

FLST set condition

- Data is written to the FIFO when the FLSTE bit in FIFO control register 1 (FCR1) is "1" and the write pointer of the transmission FIFO coincides with the read pointer saved by the FSET bit.

FLST reset conditions

- Transmission FIFO reset.
  - When FCR1:FSEL="0", set "1" to the FCR0:FCL1.
  - When FCR1:FSEL="1", set "1" to the FCR0:FCL2.
- "1" is written to the FSET bit.

When this bit is set to "1", the data pointed to by the read pointer stored by the FSET bit is overwritten. Also, the FLD bit cannot set retransmission even if an error occurs. To perform retransmission with this bit set to "1", perform a FIFO reset, and write data to the FIFO again.

Bit	Description
0	Data has not been lost.
1	Data has been lost.

[bit5] FLD : FIFO Pointer Reload Bit

This bit reloads the data saved in the transmission FIFO by the FSET bit, into the read pointer. This bit is used for retransmission in such cases as communication errors.
This bit is set to "0" when retransmission setting is completed.

- This bit is set when the FCR0S:FLDS bit in the set register is set to "1".

Bit	Description
0	Do not reload.
1	Reload.

Notes:

- *Since a reload to the read pointer is in progress while this bit is "1", operations other than FIFO reset are prohibited.*
- *Setting this bit to "1" is prohibited while FIFO is enabled or transmission is in progress.*
- *To set the SMR:TIE bit to "1", first set it to "0", write "1" to this bit, and enable the transmission FIFO. Then, set the SMR:TIE bit to "1".*

[bit4] FSET : FIFO Pointer Saving Bit

This bit stores the read pointer of the transmission FIFO.
With the read pointer saved before transmission, retransmission is possible in such cases as communication errors, provided that the FLST bit is "0".

- This bit is set when the FCR0S:FSETS bit in the set register is set to "1".

"1": Save the current read pointer value.
"0": No effect

Bit	Description
0	Do not save.
1	Save.

Note:

- *Set this bit to "1" when the transmission byte count (FBYTE) is 0.*

[bit3] FCL2 : FIFO2 Reset Bit

This bit resets FIFO2.
If this bit is set to "1", the internal status of FIFO2 is initialized.
Only the FCR0:FLST bit is initialized, and the values of other bits in the FCR1/0 register are retained.

- This bit is set when the FCR0S:FCL2S bit in the set register is set to "1".

Bit	Description	
	Write	Read
0	No effect	"0" is always read.
1	Reset FIFO2.	

Notes:

- *After disabling FIFO2, reset FIFO2.*
- *Before performing the reset, set the transmission FIFO interrupt enable bit to "0".*

- *The valid data count of the FBYTE2 register is set to 0.*
- *TDR register and RDR register are not initialized.*

[bit2] FCL1 : FIFO1 Reset Bit

This bit resets FIFO1.

If this bit is set to "1", the internal status of FIFO1 is initialized.

Only the FCR0:FLST bit is initialized, and the values of other bits in the FCR1/0 register are retained.

- This bit is set when the FCR0S:FCL1S bit in the set register is set to "1".

Bit	Description	
	Write	Read
0	No effect	"0" is always read.
1	Reset FIFO1.	

Notes:

- *After disabling FIFO1, reset FIFO1.*
- *Before performing the reset, set the transmission FIFO interrupt enable bit to "0".*
- *The valid data count of the FBYTE1 register is set to 0.*
- *TDR register and RDR register are not initialized.*

[bit1] FE2 : FIFO2 Operation Enable Bit

This bit enables or disables FIFO2 operation.

- Set this bit to "1" when using FIFO2.
- If a reception error occurs when this FIFO is selected as the reception FIFO by the FCR1:FSEL bit, this bit is cleared to "0". After that, this bit cannot be set to "1" until the reception error is cleared.
- To use FIFO2 as the transmission FIFO, set this bit to "1" when the transmission buffer is empty (SSR:TDRE= 1). To use it as the reception FIFO, set this bit to "0" when the reception buffer is empty (SSR:RDRF= 0).
- Disabling FIFO2 does not change the state of FIFO2.
- This bit is reset when the FCR0C:FE2C bit in the clear register is set to "1".
- This bit is set when the FCR0S:FE2S bit in the set register is set to "1".

Bit	Description
0	Disable FIFO2 operation.
1	Enable FIFO2 operation.

Notes:

- *When the IBSR:BB bit is "0" or the IBCR:INT bit is "1", change the enable/disable setting accordingly.*
- *If this FIFO is selected as the reception FIFO and operates for slave transmission after the reserved address is detected, set this bit to "0" upon an interrupt triggered by the detection of the reserved address. Also set IBCR:ACKE= 0.*

- *If the SSR:RDRF bit is "1" when this FIFO is used as the reception FIFO and this bit changes from "1" to "0", the reception FIFO is not disabled until the SSR:RDRF bit is set to "0".*
- *To change this bit from "0" to "1" when FIFO2 is used as the transmission FIFO and contains data, set the SMR:TIE bit to "0" before writing "1" to this bit. Then, set the SMR:TIE bit to "1".*

[bit0] FE1 : FIFO1 Operation Enable Bit

This bit enables or disables FIFO1 operation.

- Set this bit to "1" when using FIFO1.
- If a reception error occurs when this FIFO is selected as the reception FIFO by the FCR1:FSEL bit, this bit is cleared to "0". After that, this bit cannot be set to "1" until the reception error is cleared.
- To use FIFO1 as the transmission FIFO, set this bit to "1" when the transmission buffer is empty (SSR:TDRE= 1). To use it as the reception FIFO, set this bit to "0" when the reception buffer is empty (SSR:RDRF= 0).
- Disabling FIFO1 does not change the state of FIFO1.
- This bit is reset when the FCR0C:FE1C bit in the clear register is set to "1".
- This bit is set when the FCR0S:FE1S bit in the set register is set to "1".

Bit	Description
0	Disable FIFO1 operation.
1	Enable FIFO1 operation.

Notes:

- *When the IBSR:BB bit is "0" or the IBCR:INT bit is "1", change the enable/disable setting accordingly.*
- *If this FIFO is selected as the reception FIFO and operates for slave transmission after the reserved address is detected, set this bit to "0" upon an interrupt triggered by the detection of the reserved address. Also set IBCR:ACKE= 0.*
- *If the SSR:RDRF bit is "1" when this FIFO is used as the reception FIFO and this bit changes from "1" to "0", the reception FIFO is not disabled until the SSR:RDRF bit is set to "0".*
- *To change this bit from "0" to "1" when FIFO1 is used as the transmission FIFO and contains data, set the SMR:TIE bit to "0" before writing "1" to this bit. Then, set the SMR:TIE bit to "1".*

8.16. FIFO Byte Register (FBYTE)

The FIFO byte register (FBYTE) indicates the valid data count of the FIFO. This register can also specify whether to generate a reception interrupt when the reception FIFO receives the predefined quantity of data.

Bit Configuration of the FIFO Byte Register (FBYTE)

Figure 8-18 shows the bit configuration of the FIFO byte register (FBYTE).

Figure 8-18 Bit Configuration of the FIFO Byte Register (FBYTE)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	FBYTE2								FBYTE1							
R/W Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W
Protection Attribute	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The FBYTE register indicates the valid data count of the FIFO. Table 8-3 shows the relationship between the FCR1:FSEL bit settings and FBYTE.

Table 8-3 Data Count Indication

FSEL	FIFO Selection	Data Count Indication
0	FIFO2: Reception FIFO; FIFO1: Transmission FIFO	FIFO2:FBYTE2, FIFO1:FBYTE1
1	FIFO2: Transmission FIFO; FIFO1: Reception FIFO	FIFO2:FBYTE2, FIFO1:FBYTE1

- The initial transfer count value of the FBYTE register is 0x08.
- In FBYTE of the reception FIFO, set the data count that generates a reception interrupt flag. The interrupt flag (SSR:RDRF) is set to "1" when this set transfer count matches the data count indication in the FBYTE register.
- If both of the following conditions are satisfied and the reception idle state continues for 8 or more clocks of the baud rate clock, the interrupt flag (SSR:RDRF) is set to "1".
  - The reception FIFO idle detection enable bit (FCR:FRIIE) is "1".
  - The data count in the reception FIFO has not reached the transfer count.

  During an 8-clock count, the counter is reset to 0 when the RDR is read, and the system starts counting the 8 clocks again. The counter is reset to 0 when the reception FIFO is disabled. If the reception FIFO is enabled when data remains in the reception FIFO, the counting resumes.
- To receive data during master operation (master reception), set the SMR:TIE bit to "0", set the reception data count in the FBYTE register of the transmission FIFO, and write "0" to the FCR1:FDRQ bit. SCL clocks equivalent to the set data are output. Then, the IBCR:INT bit is set to "1". If you want to set the SMR:TIE bit to "1", do so after the FCR1:FDRQ is set to "1".
- When writing a transmission data to the TDR register one time, the FBYTE of the transmission FIFO is incremented by one ("+1").
- When reading a reception data from the RDR register one time, the FBYTE of the reception FIFO is decremeted by one ("-1").
- During block transfer, the value of this register is the block transfer count.

[bit15:8] FBYTE2: FIFO2 Data Count Indication byte
[bit7:0] FBYTE1: FIFO1 Data Count Indication Byte

FBYTE2, FBYTE1	Description
Write	Set the transfer count.
Read	Read the valid data count.

Read (valid data count)

Transmission: Data count that was written but not sent to the transmission FIFO
Reception: Data count that has been received but not read by the reception FIFO

Write (transfer count)

Transmission: Set 0x00.
Reception: Set the data count that generates a reception interrupt.

Table 8-4 FIFO-Stored Data Count

FIFO Capacity	Maximum FBYTE Count	Data Count That Can Be Stored
16 bytes	16	16
32 bytes	32	32
64 bytes	64	64

Notes:

- *Set FBYTE of the transmission FIFO to 0x00, except when receiving data during master operation.*
- *For the transmission data count for data reception during master operation, set the count when the transmission FIFO is empty and the SMR:TIE bit is "0".*
- *Before disabling the I^2C interface (ISMK:EN= 0) while data is being received in master operation, first disable the transmission and reception FIFOs.*
- *When the reception FIFO is not used, the reception block size is "1" for block transfer regardless of the setting of this register.*
- *Under any of the following conditions, change the setting.*
  - The I2C interface is disabled (ISMK:EN= 0).
  - IBCR:INT is "1" during master reception when SSR:DMA= 0.
  - SSR:TBI is "1" during master reception when SSR:DMA= 1.
- *Settings that exceed the FIFO capacity are prohibited.*
- *To receive data during master operation (master reception), set the SMR:TIE bit to "0", and do not write dummy data to the transmission data register (TDR) when setting the reception data count in the FBYTE register of the transmission FIFO.*
- *After setting the FIFO Selection Bit (FCR1:FSEL), set the FIFO byte register (FBYTE).*
- *The FIFO Selection Bit and the FIFO byte register can not be set at the same time.*
- *The indication of transmission FIFO data number is indicated as the valid value that is the number of writing times of the transmission data decrimented by one. This is the reason that a transmission data written is transferred to transmission FIFO when TDR register has a data not transmitted. When the data in TDR register is transmitted, the data not transmitted in transmission FIFO is transferred to TDR register.*

- *The indication of reception FIFO data number is indicated the number of the data received to reception FIFO and not read from reception FIFO. The data receiving in RDR register is not included.*

8.17. Transmission FIFO Interrupt Control Register (FTICR)

The transmission FIFO interrupt control register (FTICR) is used to set the interrupts triggered by the valid data count for FIFO transmission.

Transmission FIFO Interrupt Control Register (FTICR)

Figure 8-19 shows the bit configuration of the transmission FIFO interrupt control register (FTICR).

Figure 8-19 Bit Configuration of the Transmission FIFO Interrupt Control Register (FTICR)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	FTICR2								FTICR1							
R/W Attribute	R, W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W
Protection Attribute	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The function of this bit is changed by the value of ECR:TXBLKEN.

- ECR:TXBLKEN="0"

The FTICR register sets the interrupt trigger level by the valid transmission data count (remaining amount) of the transmission FIFO.

- ECR:TXBLKEN="1"

The FTICR register sets the interrupt trigger level by the number of the empty data in transmission FIFO.

Table 8-5 Transmission FIFO Setting by FCR1:FSEL

FSEL	Selection of Transmission FIFO	Transmission FIFO Interrupt Control Register
0	FIFO1	FTICR1
1	FIFO2	FTICR2

- The initial value of the valid data count for generating interrupts is 0x00 in the FTICR register.
- Set the data count that generates a transmission interrupt in FTICR of the transmission FIFO.
- When ECR:TXBLKEN="0", the interrupt flag (FDRQ) is set to "1" when the set data count matches or is smaller than the valid data count of the transmission FIFO (FTICR or FBYTE).
- When ECR:TXBLKEN="1", the interrupt flag (FDRQ) is set to "1" when the set data count matches or is smaller than the number of the empty data in transmission FIFO. When the number of data set to this register is bigger than the number of the empty data in transmission FIFO, the interrupt flag (FCR1:FDRQ) is set to "0".
- Set FTICR such that the setting satisfies FTICR ≤ FIFO capacity - 2.
- The read value shows the valid data count of the FIFO.
- Transmission FIFO: Quantity of written but not-yet-transmitted data in the transmission FIFO
- Reception FIFO: Quantity of received but unread data in the reception FIFO

[bit15:8] FTICR2: FIFO2 Data Count Indication byte
[bit7:0] FTICR1: FIFO1 Data Count Indication Byte

FTICR2, FTICR1	Description
Write	Set the valid data count that generates an interrupt.
Read	Read the valid data count.

Notes:

- *Settings that exceed the FIFO capacity are prohibited.*
- *The setting value cannot be read.*
- *For the display of the FIFO data count for transmission, the written count for transmission data minus 1 is displayed as the valid data count. This is because the transmission FIFO stores any transmission data written to the TDR register that still contains data not yet transmitted. When the data in the TDR register is transmitted, the not-yet-transmitted data in the transmission FIFO is transferred to the TDR register.*
- *For the display of the FIFO data count for reception, the quantity of received but unread data in the reception FIFO is displayed. The data count does not include the data being received in the RDR register.*
- *For ECR:TXBLKEN = 0, if block transfer is performed during DMA transfer, only 1 can be set as the block size.*
- *To perform block transfer with ECR:TXBLKEN = 1, set the value of the block size to this register.*

8.18. Extended Control Register (ECR)

The extended control register (ECR) can be used to set block transfer.

Bit Configuration of the Extended Control Register (ECR)

Figure 8-20 shows the bit configuration of the extended control register (ECR).

Figure 8-20 Bit Configuration of the Extended Control Register (ECR)

Bit	15 ... 8	7	6	5	4	3	2	1	0
Field	(ESR)	Reserved	Reserved	Reserved	EISEL	REIE	TEIE	RX BLKEN	TX BLKEN
R/W Attribute		R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W	R/W
Protection Attribute		-	-	-	-	-	-	-	-
Initial Value		0	0	0	0	0	0	0	0

[bit7:5] Reserved : Reserved Bits

[bit4] EISEL : Error Interrupt Request Output Selection Bit

This error interrupt output bit selects output from an interrupt request pin or output from an error interrupt request pin. For details on the interrupt output selection, see Table 2-2 and Table 2-3.

Bit	Description
0	Output a reception error interrupt request from the reception interrupt request pin. Output a transmission error interrupt request from the transmission interrupt request pin.
1	Output a reception error interrupt request from the reception error interrupt request pin. Output a transmission error interrupt request from the transmission error interrupt request pin.

[bit3] REIE : Reception Error Interrupt Enable Bit

This bit enables/disables reception error interrupt request output. For details on the target interrupt factors, see Table 2-3.

Bit	Description
0	Disable reception error interrupts.
1	Enable reception error interrupts.

[bit2] TEIE : Transmission Error Interrupt Enable Bit

This bit enables/disables transmission error interrupt request output. For details on the target interrupt factors, see Table 2-2.

Bit	Description
0	Disable transmission error interrupts.
1	Enable transmission error interrupts.

[bit1] RXBLKEN : Reception Block Transfer Setting Bit

This bit sets the DMA transfer mode for reception.

- "0": Perform DMA transfer in demand transfer mode.
- "1": Perform DMA transfer in block transfer mode.

Bit	Description
0	Demand transfer mode
1	Block transfer mode

[bit0] TXBLKEN : Transmission Block Transfer Setting Bit

This bit sets the DMA transfer mode for transmission.

- "0": Perform DMA transfer in demand transfer mode.
- "1": Perform DMA transfer in block transfer mode.

Bit	Description
0	Demand transfer mode
1	Block transfer mode

8.19. Extended Status Register (ESR)

The extended status register (ESR) is used to check the reception block transfer error flag and transmission block transfer error flag.

Extended Status Register (ESR)

Figure 8-21 shows the bit configuration of the extended status register (ESR).

Figure 8-21 Bit Configuration of the Extended Status Register (ESR)

Bit	15	14	13	12	11	10	9	8	7 ... 0
Field	Reserved	Reserved	Reserved	Reserved	RXUDR	TXOVR	RBERR	TBERR	(ECR)
R/W Attribute	R0,W0	R0,W0	R0,W0	R0,W0	R,WX	R,WX	R,WX	R,WX	
Protection Attribute	-	-	-	-	-	-	-	-	
Initial Value	0	0	0	0	0	0	0	0	

[bit15:12] Reserved : Reserved Bits

[bit11] RXUDR : Reception FIFO under Run Flag Bit

If the reception FIFO is read when empty due to a block transfer, this bit is set to "1".

- This bit is reset when the ESRC:RXUDRC bit in the clear register is set to "1".
- For details on interrupt request output, see Table 2-3.

Bit	Description
0	No reception FIFO under run has occurred.
1	A reception FIFO under run has occurred.

Notes:

- *This bit is "0" when the reception block transfer enable (ECR:RXBLKEN) is "0".*
- *This bit is set only when the reception FIFO is used.*
- *This bit is "0" when the reception FIFO is disabled.*

[bit10] TXOVR : Transmission FIFO Overrun Flag Bit

If the transmission FIFO is written when full due to a block transfer, this bit is set to "1".

- This bit is reset when the ESRC:TXOVRC bit in the clear register is set to "1".
- For details on interrupt request output, see Table 2-2.

Bit	Description
0	No transmission FIFO overrun has occurred.
1	A transmission FIFO overrun has occurred.

Notes:

- *This bit is "0" when the transmission block transfer enable (ECR:TXBLKEN) is "0".*
- *This bit is set only when the transmission FIFO is used.*
- *This bit is "0" when the transmission FIFO is disabled.*

[bit9] RBERR : Reception Block Transfer Error Bit

This bit indicates that a block transfer error occurred during reception.
If a block transfer is performed using a value larger than the set threshold value in the FBYTE register, it is handled as a block transfer error. If reception interrupts are enabled (SMR:RIE= 1), a reception interrupt is generated.

- This bit is reset when the ESRC:RBERRC bit in the clear register is set to "1".
- For details on interrupt request output, see Table 2-3.

Bit	Description
0	No reception block transfer error has occurred.
1	Reception block transfer error has occurred.

Notes:

- *This bit is "0" when the reception block transfer enable (ECR:RXBLKEN) is "0" or when the reception interrupt enable bit is "0".*

[bit8] TBERR : Transmission Block Transfer Error Bit

This bit indicates that a block transfer error occurred during transmission.
If a block transfer is performed using a value larger than the set threshold value in the FTICR register, it is handled as a block transfer error. Then, this bit is set to "1". If transmission interrupts are enabled (SMR:TIE= 1), a transmission interrupt is generated.

- This bit is reset when the ESRC:TBERRC bit in the clear register is set to "1".
- For details on interrupt request output, see Table 2-2.

Bit	Description
0	No transmission block transfer error has occurred.
1	A transmission block transfer error has occurred.

Notes:

- *If the transmission block transfer enable (ECR:TXBLKEN) is "0", this bit is "0".*
- *This bit is set "1" as the block transfer error when the transmission FIFO is OFF and the block transfer is executed under either the condition:*
  - The transmission interrupt is enabled (SMR:TIE="1") and the TDR register has a data (SSR:TDRE="0")
  - The transmission interrupt is disabled(SMR:TIE="0") and the transmission bus idle interrupt is enabled (SSR:TBIE="1") and in transmitting (SSR:TBI="0")
- *When the transmission interrupt is enabled (SMR:TIE="1") and the transmission bus idle interrupt is enabled (SSR:TBIE="1"), the condition of the transmission interrupt enabled (SMR:TIE="1") is prior.*

8.20. Transmission Block Size Register (TBSIZE)

This register is used to set the block size for transmission block transfer.

Bit Configuration of the Transmission Block Size Register (TBSIZE)

Figure 8-22 shows the bit configuration of the transmission block size register (TBSIZE).

Figure 8-22 Bit Configuration of the Transmission Block Size Register (TBSIZE)

Bit	15 ... 8	7	6	5	4	3	2	1	0
Field	-	TBSIZE							
R/W Attribute		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute		-	-	-	-	-	-	-	-
Initial Value		0	0	0	0	0	0	0	0

The TBSIZE register sets the block size for transmission block transfer.

[bit7:0] TBSIZE : Transmission Block size byte

TBSIZE	Description
Write	Set the number of transmission blocks.
Read	Read the setting value.

Notes:

- *Settings that exceed the FIFO capacity are prohibited.*
- *Setting "0" to this bit is prohibited.*
- *Settings that exceed the setting value in the transmission FIFO interrupt control register (FTICR) are prohibited.*
- *Set the same value in this register as the transmission block size that is set for the DMA controller.*
- *When the transmission FIFO is not used, the transmission block size is "1" for block transfer regardless of the register setting.*

8.21. I²C Bus Control Clear Register (IBCRC)

The I²C bus control clear register (IBCRC) can clear a bit in the I²C bus control register (IBCR).

Note:

- *For details on operations of this register, see also the descriptions of IBCR, which is the targeted register.*

I²C Bus Control Clear Register (IBCRC)

Figure 8-23 shows the bit configuration of the I²C bus control clear register (IBCRC).

Figure 8-23 Bit Configuration of the I²C Bus Control Clear Register (IBCRC)

Bit	15	14	13	12	11	10	9	8	7 ... 0
Field	MSSC	Reserved	ACKEC	WSELC	CNDEC	INTEC	Reserved	INTC	SMRC
R/W Attribute	R0,W	R0,W0	R0,W	R0,W	R0,W	R0,W	R0,W0	R0,W	
Protection Attribute	-	-	-	-	-	-	-	-	
Initial Value	0	0	0	0	0	0	0	0	

[bit15] MSSC : Clearing the Master/Slave Selection Bit

Writing "1" to this bit resets IBCR:MSS to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit14] Reserved : Reserved Bit

[bit13] ACKEC : Clearing the Data Byte acknowledgment Enable Bit

Writing "1" to this bit resets IBCR:ACKE to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit12] WSELC : Clearing the Wait Selection Bit

Writing "1" to this bit resets IBCR:WSEL to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit11] CNDEC : Clearing the Condition Detection Interrupt Enable Bit

Writing "1" to this bit resets IBCR:CNDE to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit10] INTEC : Clearing the Interrupt Enable Bit

Writing "1" to this bit resets IBCR:INTE to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit9] Reserved : Reserved Bit

[bit8] INTC : Clearing the Interrupt Flag

Writing "1" to this bit resets IBCR:INT to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

8.22. Serial Mode Clear Register (SMRC)

The serial mode clear register (SMRC) can clear a bit in the serial mode register (SMR).

Note:

- *For details on operations of this register, see also the descriptions of SMR, which is the targeted register.*

Bit Configuration of the Serial Mode Clear Register (SMRC)

Figure 8-24 shows the bit configuration of the serial mode clear register (SMRC).

Figure 8-24 Bit Configuration of the Serial Mode Clear Register (SMRC)

Bit	15 ... 8	7	6	5	4	3	2	1	0
Field	(IBCRC)	Reserved	Reserved	Reserved	WUCRC	RIEC	TIEC	Reserved	Reserved
R/W Attribute		R0,W0	R0,W0	R0,W0	R0,W	R0,W	R0,W	R0,W0	R0,W0
Protection Attribute		-	-	-	-	-	-	-	-
Initial Value		0	0	0	0	0	0	0	0

[bit7:5] Reserved : Reserved Bits

[bit4] WUCRC : Clearing the WAKE UP Control Bit

Writing "1" to this bit resets SMR:WUCR to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit3] RIEC : Clearing the Reception Interrupt Enable Bit

Writing "1" to this bit resets SMR:RIE to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit2] TIEC : Clearing the Transmission Interrupt Enable Bit

Writing "1" to this bit resets SMR:TIE to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit1:0] Reserved ; Reserved Bits

8.23. Serial Status Clear Register (SSRC)

The serial status clear register (SSRC) can clear a bit in the serial status register (SSR).

Note:

- *For details on operations of this register, see also the descriptions of SSR, which is the targeted register.*

Bit Configuration of the Serial Status Clear Register (SSRC)

Figure 8-25 shows the bit configuration of the serial status clear register (SSRC).

Figure 8-25 Bit Configuration of the Serial Status Clear Register (SSRC)

Bit	15	14	13	12	11	10	9	8	7 ... 0
Field	Reserved	Reserved	DMAC	TBIEC	Reserved	Reserved	Reserved	Reserved	(IBSRC)
R/W Attribute	R0,W0	R0,W0	R0,W	R0,W	R0,W0	R0,W0	R0,W0	R0,W0	
Protection Attribute	-	-	-	-	-	-	-	-	
Initial Value	0	0	0	0	0	0	0	0	

[bit15:14] Reserved : Reserved Bits

[bit13] DMAC : Clearing the DMA Mode Enable Bit

Writing "1" to this bit resets SSR:DMA to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit12] TBIEC : Clearing the Transmission Bus Idle Interrupt Enable Bit (Valid Only when DMA Mode is Enabled)

Writing "1" to this bit resets SSR:TBIE to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit11:8] Reserved : Reserved Bits

8.24. I²C Bus Status Clear Register (IBSRC)

The I²C bus status clear register (IBSRC) can clear a corresponding bit in the I²C bus status register (IBSR).

Note:

- *For details on operations of this register, see also the descriptions of IBSR, which is the targeted register.*

I²C Bus Status Clear Register (IBSRC)

Figure 8-26 shows the bit configuration of the I²C bus status clear register (IBSRC).

Figure 8-26 Bit Configuration of the I²C Bus Status Clear Register (IBSRC)

Bit	15 ... 8	7	6	5	4	3	2	1	0
Field	SSRC	Reserved	Reserved	Reserved	Reserved	Reserved	RSCC	SPCC	Reserved
R/W Attribute		R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W	R0,W	R0,W0
Protection Attribute		-	-	-	-	-	-	-	-
Initial Value		0	0	0	0	0	0	0	0

[bit7:3] Reserved : Reserved Bits

[bit2] RSCC : Clearing the repeated Start Condition Check Bit

Writing "1" to this bit resets IBSR:RSC to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit1] SPCC : Clearing the Stop Condition Check Bit

Writing "1" to this bit resets IBSR:SPC to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit0] Reserved : Reserved Bit

8.25. Serial Auxiliary Control Status Clear Register(SACSRC)

The serial auxiliary control status clear register (SACSRC) can clear a corresponding bit in the serial auxiliary control status register (SACSR).

Note:

- *For details on operations of this register, see also the descriptions of SACSR, which is the targeted register.*

Bit Configuration of the Serial Auxiliary Control Status Clear Register (SACSRC)

Figure 8-27 Bit Configuration of the Serial Auxiliary Control Status Register (SACSRC)

Bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	TINTC
R/W Attribute	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	TINTEC	Reserved	TRGEC	Reserved	Reserved	Reserved	Reserved	TMREC
R/W Attribute	R0,W	R0,W0	R0,W	R0,W0	R0,W0	R0,W0	R0,W0	R0,W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

[bit15:9] Reserved : Reserved Bits

[bit8] TINTC : Clearing the Timer Interrupt Flag

Writing "1" to this bit resets SACSR:TINT to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit7] TINTEC : Clearing the Timer Interrupt Enable Bit

Writing "1" to this bit resets SACSR:TINTE to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit6] Reserved : Reserved Bit

[bit5] TRGEC : Clearing the External Trigger Enable Bit

Writing "1" to this bit resets SACSR:TINTE to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit4:1] Reserved : Reserved Bits

[bit0] TMREC : Clearing the Serial Timer Enable Bit

Writing "1" to this bit resets SACSR:TMRE to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

8.26. FIFO Control Clear Register 1 (FCR1C)

FIFO control clear register 1 (FCR1C) can clear a bit in FIFO control register 1 (FCR1).

Note:

- *For details on operations of this register, see also the descriptions of FCR1, which is the targeted register.*

8.26.1.1 Bit Configuration of FIFO Control Clear Register 1 (FCR1C)

Figure 8-28 Bit Configuration of FIFO Control Clear Register 1 (FCR1C)

Bit	15	14	13	12	11	10	9	8	7 ... 0
Field	Reserved	Reserved	Reserved	FLSTEC	FRIIEC	FDRQC	FTIEC	FSELC	(FCR0C)
R/W Attribute	R0,W0	R0,W0	R0,W0	R0,W	R0,W	R0,W	R0,W	R0,W	
Protection Attribute	-	-	-	-	-	-	-	-	
Initial Value	0	0	0	0	0	0	0	0	

[bit15:13] Reserved : Reserved Bits

[bit12] FLSTEC : Clearing the Retransmission Data Lost Detection Enable Bit

Writing "1" to this bit resets FCR1:FLSTE to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit11] FRIIEC : Clearing the Reception FIFO Idle Detection Enable Bit

Writing "1" to this bit resets FCR1:FRIIE to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit10] FDRQC : Clearing the Transmission FIFO Data Request Bit

Writing "1" to this bit resets FCR1:FDRQ to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit9] FTIEC : Clearing the Transmission FIFO Interrupt Enable Bit

Writing "1" to this bit resets FCR1:FTIE to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit8] FSELC : Clearing the FIFO Selection Bit

Writing "1" to this bit resets FCR1:FSEL to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

8.27. FIFO Control Clear Register 0 (FCR0C)

FIFO control clear register 0 (FCR0C) can clear a bit in FIFO control register 0 (FCR0).

Note:

- *For details on operations of this register, see also the descriptions of FCR0, which is the targeted register.*

Bit Configuration of FIFO Control Clear Register 0 (FCR0C)

Figure 8-29 Bit Configuration of FIFO Control Clear Register 0 (FCR0C)

Bit	15 .. 8	7	6	5	4	3	2	1	0
Field	(FCR1C)	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	FE2C	FE1C
R/W Attribute		R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W	R0,W
Protection Attribute		-	-	-	-	-	-	-	-
Initial Value		0	0	0	0	0	0	0	0

[bit7:2] Reserved : Reserved Bits

[bit1] FE2C : Clearing the FIFO2 Operation Enable Bit

Writing "1" to this bit resets FCR0:FE2 to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit0] FE1C : Clearing the FIFO1 Operation Enable Bit

Writing "1" to this bit resets FCR0:FE1 to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

8.28. Extended Status Clear Register (ESRC)

The extended status clear register (ESRC) can be used to clear a bit in the extended status register (ESR).

Extended Status Clear Register (ESRC)

Figure 8-30 shows the bit configuration of the extended status clear register (ESRC).

Figure 8-30 Bit Configuration of the Extended Status Clear Register (ESRC)

Bit	15	14	13	12	11	10	9	8	7 ... 0
Field	Reserved	Reserved	Reserved	Reserved	RXUDRC	TXOVRC	RBERRC	TBERRC	-
R/W Attribute	R0,W0	R0,W0	R0,W0	R0,W0	R0,W	R0,W	R0,W	R0,W	
Protection Attribute	-	-	-	-	-	-	-	-	
Initial Value	0	0	0	0	0	0	0	0	

[bit15:12] Reserved : Reserved Bits

[bit11] RXUDRC : Reception FIFO under Run Flag Clear Bit

Writing "1" to this bit resets ESR:RXUDR to "0".
Writing "0" to this bit is invalid.
"0" is always read from this bit.

[bit10] TXOVRC : Transmission FIFO Overrun Flag Clear Bit

Writing "1" to this bit resets ESR:TXOVR to "0".
Writing "0" to this bit is invalid.
"0" is always read from this bit.

[bit9] RBERRC : Reception Block Transfer Error Clear Bit

Writing "1" to this bit resets ESR:RBERR to "0".
Writing "0" to this bit is invalid.
"0" is always read from this bit.

[bit8] TBERRC : Transmission Block Transfer Error Clear Bit

Writing "1" to this bit resets ESR:TBERR to "0".
Writing "0" to this bit is invalid.
"0" is always read from this bit.

8.29. I²C Bus Control Set Register (IBCRS)

The I²C bus control set register (IBCRS) can set a bit in the I²C bus control register (IBCR).

Note:

- *For details on operations of this register, see also the descriptions of IBCR, which is the targeted register.*

I²C Bus Control Set Register (IBCRS)

Figure 8-31 shows the bit configuration of the I²C bus control set register (IBCRS).

Figure 8-31 Bit Configuration of the I²C Bus Control Set Register (IBCRS)

Bit	15	14	13	12	11	10	9	8	7 ... 0
Field	MSSS	ACTS	ACKES	WSELS	CNDES	INTES	Reserved	INTS	(SMRS)
R/W Attribute	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W0	R0,W	
Protection Attribute	-	-	-	-	-	-	-	-	
Initial Value	0	0	0	0	0	0	0	0	

[bit15] MSSS : Setting the Master/Slave Selection Bit

Writing "1" to this bit sets IBCR:MSS to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit14] ACTS : Setting the repeated Start Condition Generation Bit

Writing "1" to this bit sets IBCR:ACT to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit13] ACKES : Setting the Data Byte acknowledgment Enable Bit

Writing "1" to this bit sets IBCR:ACKE to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit12] WSELS : Setting the Wait Selection Bit

Writing "1" to this bit sets IBCR:WSEL to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit11] CNDES : Setting the Condition Detection Interrupt Enable Bit

Writing "1" to this bit sets IBCR:CNDE to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit10] INTES : Setting the Interrupt Enable Bit

Writing "1" to this bit sets IBCR:INTE to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit9] Reserved : Reserved Bit

[bit8] INTS : Setting the Interrupt Flag

Writing "1" to this bit sets IBCR:INT to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

8.30. Serial Mode Set Register (SMRS)

The serial mode set register (SMRS) can set a bit in the serial mode register (SMR).

Note:

- *For details on operations of this register, see also the descriptions of SMR, which is the targeted register.*

Bit Configuration of the Serial Mode Set Register (SMRS)

Figure 8-32 Bit Configuration of the Serial Mode Set Register (SMRS)

Bit	15 ... 8	7	6	5	4	3	2	1	0
Field	(IBCRS)	Reserved	Reserved	Reserved	WUCRS	RIES	TIES	Reserved	Reserved
R/W Attribute		R0,W0	R0,W0	R0,W0	R0,W	R0,W	R0,W	R0,W0	R0,W0
Protection Attribute		-	-	-	-	-	-	-	-
Initial Value		0	0	0	0	0	0	0	0

[bit7:5] Reserved : Reserved Bits

[bit4] WUCRS : Setting the WAKE UP Control Bit

Writing "1" to this bit sets SMR:WUCR to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit3] RIES : Setting the Reception Interrupt Enable Bit

Writing "1" to this bit sets SMR:RIE to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit2] TIES : Setting the Transmission Interrupt Enable Bit

Writing "1" to this bit sets SMR:TIE to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit1:0] Reserved : Reserved Bits

8.31. Serial Status Set Register (SSRS)

The serial status set register (SSRS) can set a bit in the serial status register (SSR).

Note:

- *For details on operations of this register, see also the descriptions of SSR, which is the targeted register.*

Bit Configuration of the Serial Status Set Register (SSRS)

Figure 8-33 Bit Configuration of the Serial Status Set Register (SSRS)

Bit	15	14	13	12	11	10	9	8	7 ... 0
Field	RECS	TSETS	DMAS	TBIES	Reserved	Reserved	Reserved	Reserved	-
R/W Attribute	R0,W	R0,W	R0,W	R0,W	R0,W0	R0,W0	R0,W0	R0,W0	
Protection Attribute	-	-	-	-	-	-	-	-	
Initial Value	0	0	0	0	0	0	0	0	

[bit15] RECS : Setting the Reception Error Flag Clear Bit

Writing "1" to this bit sets SSR:REC to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit14] TSETS : Setting the Transmission Buffer Empty Flag Set Bit

Writing "1" to this bit sets SSR:TSET to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit13] DMAS : Setting the DMA Mode Enable Bit

Writing "1" to this bit sets SSR:DMA to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit12] TBIES : Setting the Transmission Bus Idle Interrupt Enable Bit (Valid Only when DMA Mode is Enabled)

Writing "1" to this bit sets SSR:TBIE to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit11:8] Reserved : Reserved Bits

8.32. Serial Auxiliary Control Status Set Register (SACSRS)

The serial auxiliary control status set register (SACSRS) can set a bit in the serial auxiliary control status register (SACSR).

Note:

- *For details on operations of this register, see also the descriptions of SACSR, which is the targeted register.*

Bit Configuration of the Serial Auxiliary Control Status Set Register (SACSRS)

Figure 8-34 Bit Configuration of the Serial Auxiliary Control Status Set Register (SACSRS)

Bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W Attribute	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	TINTES	Reserved	TRGES	Reserved	Reserved	Reserved	Reserved	TMRES
R/W Attribute	R0,W	R0,W0	R0,W	R0,W0	R0,W0	R0,W0	R0,W0	R0,W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

[bit15:8] Reserved : Reserved Bits

[bit7] TINTES : Setting the Timer Interrupt Enable Bit

Writing "1" to this bit sets SACSR:TINTE to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit6] Reserved : Reserved Bit

[bit5] TRGES : Setting the External Trigger Enable Bit

Writing "1" to this bit sets SACSR:TRGE to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit4:1] Reserved : Reserved Bits

[bit0] TMRES : Setting the Serial Timer Enable Bit

Writing "1" to this bit sets SACSR:TMRE to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

8.33. FIFO Control Set Register 1 (FCR1S)

FIFO control set register 1 (FCR1S) can set a bit in FIFO control register 1 (FCR1).

Note:

- *For details on operations of this register, see also the descriptions of FCR1, which is the targeted register.*

Bit Configuration of FIFO Control Set Register 1 (FCR1S)

Figure 8-35 Bit Configuration of FIFO Control Set Register 1 (FCR1S)

Bit	15	14	13	12	11	10	9	8	7 ... 0
Field	Reserved	Reserved	Reserved	FLSTES	FRIIES	Reserved	FTIES	FSELS	(FCR0S)
R/W Attribute	R0,W0	R0,W0	R0,W0	R0,W	R0,W	R0,W0	R0,W	R0,W	
Protection Attribute	-	-	-	-	-	-	-	-	
Initial Value	0	0	0	0	0	0	0	0	

[bit15:13] Reserved : Reserved Bits

[bit12] FLSTES : Setting the Retransmission Data Lost Detection Enable Bit

Writing "1" to this bit sets FCR1:FLSTE to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit11] FRIIES : Setting the Reception FIFO Idle Detection Enable Bit

Writing "1" to this bit sets FCR1:FRIIE to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit10] Reserved : Reserved Bit

[bit9] FTIES : Setting the Transmission FIFO Interrupt Enable Bit

Writing "1" to this bit sets FCR1:FTIE to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit8] FSELS : Setting the FIFO Selection Bit

Writing "1" to this bit sets FCR1:FSEL to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

8.34. FIFO Control Set Register 0 (FCR0S)

FIFO control set register 0 (FCR0S) can set a bit in FIFO control register 0 (FCR0).

Note:

- *For details on operations of this register, see also the descriptions of FCR0, which is the targeted register.*

Bit Configuration of FIFO Control Set Register 0 (FCR0S)

Figure 8-36 Bit Configuration of FIFO Control Set Register 0 (FCR0S)

Bit	15 ... 8	7	6	5	4	3	2	1	0
Field	(FCR1S)	Reserved	Reserved	FLDS	FSETS	FCL2S	FCL1S	FE2S	FE1S
R/W Attribute		R0,W0	R0,W0	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
Protection Attribute		-	-	-	-	-	-	-	-
Initial Value		0	0	0	0	0	0	0	0

[bit7:6] Reserved : Reserved Bits

[bit5] FLDS : Setting the FIFO Pointer Reload Bit

Writing "1" to this bit sets FCR0:FLD to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit4] FSETS : Setting the FIFO Pointer Saving Bit

Writing "1" to this bit sets FCR0:FSET to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit3] FCL2S : Setting the FIFO2 Reset Bit

Writing "1" to this bit sets FCR0:FCL2 to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit2] FCL1S : Setting the FIFO1 Reset Bit

Writing "1" to this bit sets FCR0:FCL1 to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit1] FE2S : Setting the FIFO2 Operation Enable Bit

Writing "1" to this bit sets FCR0:FE2 to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit0] FE1S : Setting the FIFO1 Operation Enable Bit

Writing "1" to this bit sets FCR0:FE1 to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

CHAPTER 39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1))

This chapter describes the LIN communication function supported by operation mode 3 of the multi-function serial interface functions.

1. LIN Interface (V2.1) (LIN Communication Control Interface (V2.1))
2. LIN Interface (v2.1) Interrupts
3. Serial Timer Operation
4. Test Mode
5. Dedicated Baud Rate Generator
6. Block Transfer
7. LIN Interface (V2.1) Operation
8. Setup Procedure and Program Flow for Operation Mode 3 (LIN Communication Mode)
9. Interface (V2.1) Registers

LIN-TXXPT03P01R01L09-E1-XX

1. LIN Interface (V2.1) (LIN Communication Control Interface (V2.1))

1.1. Manual Mode

The LIN interface (v2.1) (LIN communication control interface (v2.1)) provides the functions to support the LIN bus.
In addition, this interface incorporates transmission/reception FIFOs (64 bytes each).

LIN interface (v2.1) (LIN communication control interface (v2.1)) functions (manual mode)

	Function	Description
1	Data buffer	– Full duplex, double buffering (when FIFOs are not used) – Transmission and reception FIFOs (64 bytes each) (when FIFOs are used)
2	Serial input	– Oversampling is performed 3 times by the bus clock. The reception value is determined by majority decision.
3	Transfer mode	– Asynchronous
4	Baud rate	– Dedicated baud rate generator is provided (configured from a 15-bit reload counter). – The external clock can be adjusted by the reload counter. – Auto baud rate adjustment with Sync Field reception.
5	Data length	– 8 bits
6	Signaling method	– NRZ (Non Return to Zero)
7	Start bit detection	– Synchronized to falling edges of the start bit
8	Reception error detection	– Framing error – Overrun error
9	Interrupt request	– Reception interrupt (reception completion, framing error, overrun error, and reception block transfer error) – Reception FIFO interrupt (reception FIFO under run) – Transmission interrupt (transmission data empty, transmission bus idle, and transmission block transfer error) – Status interrupt (LIN Break field detection and serial timer interrupt) – Interrupt request to ICU (LIN Sync field detection: LSYN) – Transmission FIFO interrupt (when the transmission FIFO is not higher than the interrupt trigger level or when the transmission FIFO is empty, transmission FIFO overrun) – DMA transfer is supported for both transmission and reception.
10	Timer function	– A 16-bit serial timer is provided. – A division value can be selected for the operation clock (division by 1 to 256). – Activation by external trigger is available.
11	LIN bus option	– Support for LIN Protocol Revision 2.1 – Master device operation – Slave device operation – LIN Break field generation (changeable from 13 to 16 bits) – LIN Break delimiter generation (changeable from 1 to 4 bits) – LIN Break field detection – Detection of the start/stop edge of the LIN Sync field connected to an input capture
12	FIFO option	– Transmission and reception FIFOs are provided (64 bytes for the transmission FIFO and 64 bytes for the reception FIFO). – Transmission FIFO and reception FIFO can be selected. – Transmission data can be retransmitted. – The timing of the reception FIFO interrupt can be changed by software. – Independent FIFO reset is supported.

Note:

– *The LIN Wake Up function is not supported.*

1.2. Assist Mode

The LIN interface (v2.1) (LIN communication control interface (v2.1)) provides functions for supporting the LIN bus. The header section can be automatically transmitted/detected in LIN communication. In addition, this interface incorporates transmission/reception FIFOs (64 bytes each).

LIN Interface (v2.1) (LIN Communication Control Interface (v2.1)) Functions (Assist Mode)

	Function	Description
1	Data buffer	− Full duplex, double buffering (when FIFOs are not used) − Transmission and reception FIFOs (64 bytes each) (when FIFOs are used)
2	Serial input	− Oversampling is performed 3 times by the bus clock. The reception value is determined by majority decision.
3	Transfer mode	− Asynchronous
4	Baud rate	− Dedicated baud rate generator is provided (configured from a 15-bit reload counter). − The external clock can be adjusted by the reload counter. − Auto baud rate adjustment with Sync Field reception.
5	Data length	− 8 bits
6	Signaling method	− NRZ (Non Return to Zero)
7	Start bit detection	− Synchronized to falling edges of the start bit
8	Reception error detection	− Framing error − Overrun error − LIN bus error − LIN ID parity error − LIN checksum error
9	Interrupt request	− Transmission interrupt ➢ Data transmission interrupt (transmission data empty, transmission bus idle, and transmission block transfer error) ➢ Transmission FIFO interrupt (when the transmission FIFO is not higher than the interrupt trigger level or when the transmission FIFO is empty, transmission FIFO overrun) − Reception interrupt ➢ Data reception interrupt (reception completion and reception block transfer error) ➢ Reception FIFO interrupt (when the reception FIFO is higher than the interrupt threshold, reception FIFO under run) ➢ Various types of error interrupts (LIN bus error, LIN Sync Data error, LIN ID parity error, framing error, overrun error, and LIN checksum error) − Status interrupt ➢ Auto header completion interrupt ➢ Sync Field detection interrupt ➢ Checksum calculation completion interrupt − DMA transfer is supported for both transmission and reception.
10	Timer function	− A 16-bit serial timer is provided. − A division value can be selected for the operation clock (division by 1 to 256). − Activation by external trigger is available.

11	LIN bus option	– Support for LIN Protocol Revision 2.1 – Automatic transmission/reception of the header of the master/slave device ➢ LIN Break field generation (changeable from 13 to 20 bits) ➢ LIN Break delimiter generation (changeable from 1 to 4 bits) ➢ Automatic generation of the Sync Field and automatic check of data value (0x55) ➢ Automatic generation/check of the parity value of the ID Field ➢ Automatic generation/check of a checksum (support for standard/extended checksum)
12	FIFO option	– Transmission and reception FIFOs are provided (64 bytes for the transmission FIFO and 64 bytes for the reception FIFO). – Transmission FIFO and reception FIFO can be selected. – Transmission data can be retransmitted. – The timing of the transmission/reception FIFO interrupt can be changed by software. – Independent FIFO reset is supported.
13	LIN communication test function	– Serial communication test function – Pseudo error generation functions (LIN bus error, LIN ID parity error, LIN checksum error, LIN Sync Field error, and framing error)

Note:

– *The LIN Wake Up function is not supported.*

2. LIN Interface (v2.1) Interrupts

2.1. Manual Mode

The LIN interface (v2.1) can generate reception interrupts, transmission interrupts, and status interrupts. In manual mode, an interrupt request can be generated by any of the following factors.

- When reception data is set in the reception data register (RDR) or when a reception error occurs.
- When transmission is started after transmission data is transferred from the transmission data register (TDR) to the transmission shift register.
- Transmission bus idle (no transmission operation)
- Transmission FIFO data request
- LIN Break Field detection
- LIN Sync Field detection
- The serial timer comparison value (STMCR) matches the serial timer value (STMR).
  LIN interface (v2.1) interrupts (manual mode)

Table 2-1 shows the interrupt control bits of the LIN interface (v2.1), together with the interrupt factors in manual mode.

Table 2-1 Interrupt Control Bits and Interrupt Factors of LIN Interface (v2.1) (Manual Mode)

Interrupt Type	Interrupt Request Flag Bit	Flag Register	Interrupt Factor	Interrupt Factor Enable Bit	Clearing of Interrupt Request Flag
Reception	RDRF	SSR	1-byte reception	SCR:RIE	- Reading of reception data (RDR) - Software reset (SCR:UPCL = 1)
			Reception of as much data as the amount set in FBYTE		- Reading of reception data until the reception FIFO becomes empty - Software reset (SCR:UPCL = 1)
			Detection of reception idle for 8-bit time or longer while the FRIIE bit is "1" and the reception FIFO contains valid data		
	ORE	SSR	Overrun error	SCR:RIE ECR:REIE	- Writing "1" to the reception error flag clear bit (SSR:REC) - Software reset (SCR:UPCL = 1)
	FRE	SSR	Framing error		
	RXUDR	ESR	Reception FIFO underrun	ECR:REIE	- Writing "1" to the reception FIFO underrun clear flag bit (ESRC:RXUDRC) - Software reset (SCR:UPCL = 1) - Disable reception FIFO - When FCR1:FSEL=0, set FCR0:FE2=0 - When FCR1:FSEL=1, set FCR0:FE1=0
	RBERR	ESR	Reception block transfer error	ECR:REIE	- Writing "1" to the reception block transfer error clear bit (ESRC:RBERRC) - Software reset(SCR:UPCL=1) - Disable Block transfer mode (ECR:RXBLKEN=0)
Transmission	TDRE	SSR	The transmission register is empty.	SCR:TIE	Writing to the transmission data (TDR), or writing "1" to the transmission FIFO operation enable bit (FCR0:FE1 or FCR0:FE2) (retransmission) while the bit value is "0" and the transmission FIFO contains valid data *1

Interrupt Type	Interrupt Request Flag Bit	Flag Register	Interrupt Factor	Interrupt Factor Enable Bit	Clearing of Interrupt Request Flag
	TBI	SSR	No transmission operation	SCR:TBIE	Writing to the transmission data (TDR), writing "1" to the LIN Break Field setting bit (LBR), or writing "1" to the transmission FIFO operation enable bit (FCR0:FE1 or FCR0:FE2) (retransmission) while the bit value is "0" and the transmission FIFO contains valid data*1
	FDRQ	FCR1	The amount of data in the transmission FIFO is equal to or smaller than the FTICR setting value or it is empty.	FCR1:FTIE	Writing "1" to the FIFO transmission data request clear bit (FCR1C:FDRQC) , or transmission FIFO full
	TXOVR	ESR	Transmission FIFO overrun	ECR:TEIE	- Writing "1" to the transmission FIFO overrun flag clear bit (ESRC:TXOVRC) - Software reset (SCR:UPCL = 1) - Disable transmission FIFO - When FCR1:FSEL=0, set FCR0:FE1=0 - When FCR1:FSEL=1, set FCR0:FE2=0
	TBERR	ESR	Transmission block transfer error	ECR:TEIE	- Writing "1" to the transmission block transfer error clear bit (FSRC:TBERRC) - Software reset (SCR:UPCL=1) - Disable Block transfer mode (ECR:TXBLKEN=0)
Status (Manual Mode)	LBD	SSR	LIN Break Field detection	ESCR:LBIE	- Writing "1" to the SSRC:LBDC bit - Software reset (SCR:UPCL = 1)
	SFD	SACSR	Sync Field detection	SACSR:SFDE	- Writing "1" to the Sync Field detection flag clear bit (SACSRC:SFDC) - Software reset (SCR:UPCL=1)
	TINT	SACSR	Coincidence of the serial timer register (STMR) and the serial timer comparison register (STMCR) values	SACSR:TINTE	- Writing "1" to the timer interrupt flag clear bit (SACSRC:TINTC) - Software reset (SCR:UPCL=1)
Input Capture	ICP0	ICS0	1st falling edge of the LIN Sync Field	ICS0:ICE0	Disabling the ICP0
	ICP0	ICS0	5th falling edge of the LIN Sync Field		

*1: Set the TIE bit to "1" after the TDRE bit becomes "0".

*2: When auto baud rate adjustment is enabled (SACSR:AUTE = 1),this interrupt by input capture is not generated.

Switching of Error Interrupt Request Output

The method of outputting an error interrupt request varies depending on the settings of the error interrupt request output switching bit (ECR:EISEL), transmission/reception error interrupt enable bits (ECR:REIE, TEIE), and transmission/reception interrupt enable bits (SCR:RIE, TIE, TBIE, FCR1:FTIE).

Table 2-2 Method of Outputting Transmission Interrupt Requests

ECR:EISEL	Interrupt Output	Target Interrupt Enable Bit	Interrupt Factor
0	Transmission interrupt request output	SCR:TIE	Transmission data empty (SSR:TDRE)
		SCR:TBIE	No transmission operation (SSR:TBI)
		FCR1:FTIE	Transmission FIFO empty (FCR1:FDRQ)
		ECR:TEIE	Transmission FIFO overrun (ESR:TXOVR) Transmission block transfer error (ESR:TBERR)
	Transmission error interrupt request output	-	No interrupt factor
1	Transmission interrupt request output	SCR:TIE	Transmission data empty (SSR:TDRE)
		SCR:TBIE	No transmission operation (SSR:TBI)
		FCR1:FTIE	Transmission FIFO empty (FCR1:FDRQ)
	Transmission error interrupt request output	ECR:TEIE	Transmission FIFO overrun (ESR:TXOVR) Transmission block transfer error (ESR:TBERR)

Table 2-3 Method of Outputting Reception Interrupt Requests

ECR:EISEL	Interrupt Output	Target Interrupt Enable Bit	Interrupt Factor
0	Reception interrupt request output	SCR:RIE	Reception full (SSR:RDRF) Framing error (SSR:FRE) Overrun error (SSR:ORE)
		ECR:REIE	Reception FIFO underrun (ESR:RXUDR) Reception block transfer error (ESR:RBERR)
	Reception error interrupt request output	-	No interrupt factor
1	Reception interrupt request output	SCR:RIE	Reception full (SSR:RDRF)
	Reception error interrupt request output	ECR:REIE	Framing error (SSR:FRE) Overrun error (SSR:ORE) Reception FIFO underrun (ESR:RXUDR) Reception block transfer error (ESR:RBERR)

2.1.1. Occurrence of Reception Interrupts and Flag Set Timing

There are 2 reception interrupt types: reception completion (SSR:RDRF) and reception error (SSR:ORE, FRE).

Occurrence of Reception Interrupts and Flag Set Timing

Detection of the first stop bit stores the reception data to the reception data register (RDR). Related flags are set upon the completion of reception (SSR:RDRF = 1) or upon a reception error (SSR:ORE, FRE = 1). At this time, a reception interrupt occurs if reception interrupt is enabled (SCR:RIE = 1).

Note:

- *If a reception error occurs, the data of the reception data register (RDR) is invalid.*

Figure 2-1 Set Timing of RDRF (Reception Data Full) Flag Bit

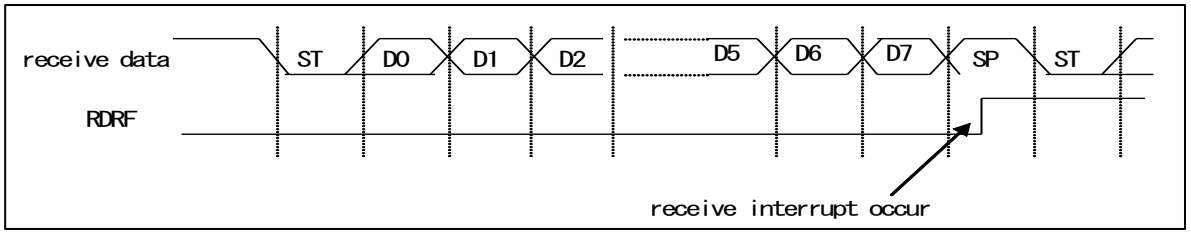
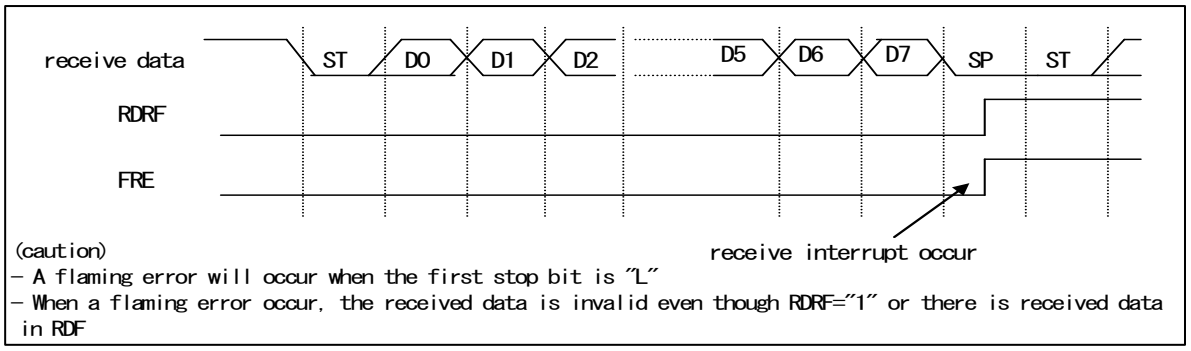


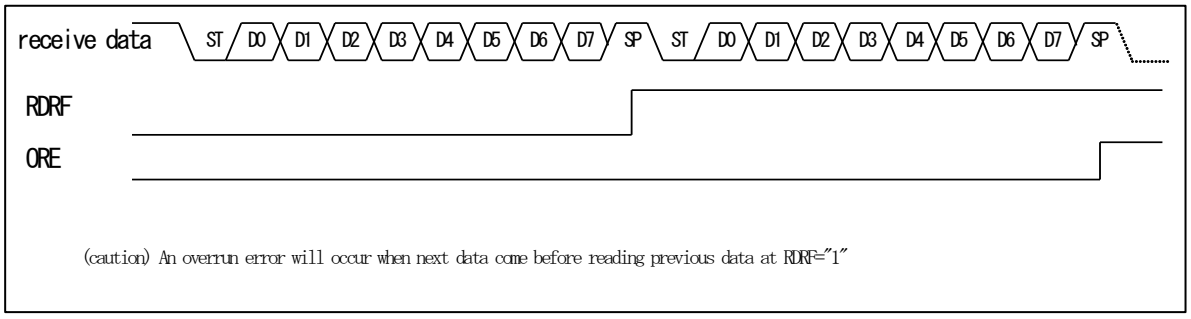
Figure 2-2 Set Timing of FRE (Framing Error) Flag Bit



Note:

- *During data reception, the following occurs if a falling edge of serial data is detected at the same time as, or 1 or 2 bus clocks earlier than, the sampling point: The edge becomes invalid and the data that follows may not be received normally. Consecutive frames must have intervals between them.*

Figure 2-3 Set Timing of ORE (Overrun Error) Flag Bit



2.1.2. Occurrence of Interrupts and Flag Set Timing When Using Reception FIFO

An interrupt occurs during the use of the reception FIFO when as much data as the amount set in the FIFO byte register (FBYTE) is received.

Occurrence of Reception Interrupts and Flag Set Timing When Using Reception FIFO

The occurrence of interrupts during the use of the reception FIFO is determined by the value set in the FBYTE register.

- If as much transfer data as the amount set in the FBYTE register is received, the reception data full flag (SSR:RDRF) in the serial status register is set to "1". At this time, if reception interrupts are enabled (SCR:RIE), a reception interrupt is generated.
- When the amount of data in the reception FIFO is "1" or greater and both of the following conditions are satisfied, the continuation of the reception idle status for 8 baud rate clocks or longer sets the interrupt flag (SSR:RDRF) to "1".
  - The reception FIFO idle detection enable bit (FCR1:FRIIE) is "1".
  - The number of data items in the reception FIFO does not reach the transfer count.

  During an 8-clock count, the counter is reset to 0 when RDR is read, and the system starts counting the 8 clocks again. The counter is reset to 0 when the reception FIFO is disabled. After the reception FIFO is enabled while there is still data in the reception FIFO, counting restarts.
- When the reception FIFO becomes empty as a result of reading the reception data (RDR), the reception data full flag (SSR:RDRF) is cleared.
- If the reception valid data count becomes equal to the value of the FIFO capacity, the reception of the next data triggers an overrun error (SSR:ORE = 1).

Figure 2-4 Occurrence Timing of Reception Interrupt When Reception FIFO Is Used

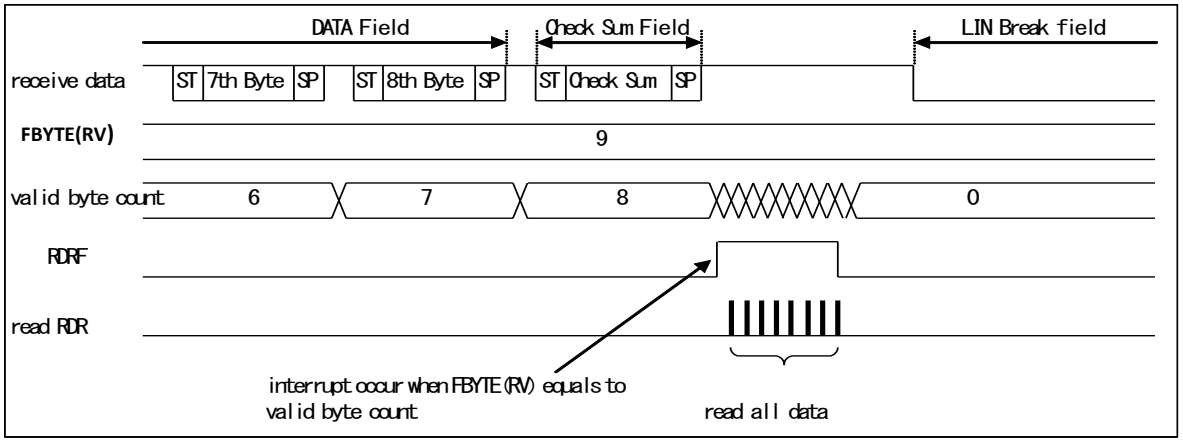
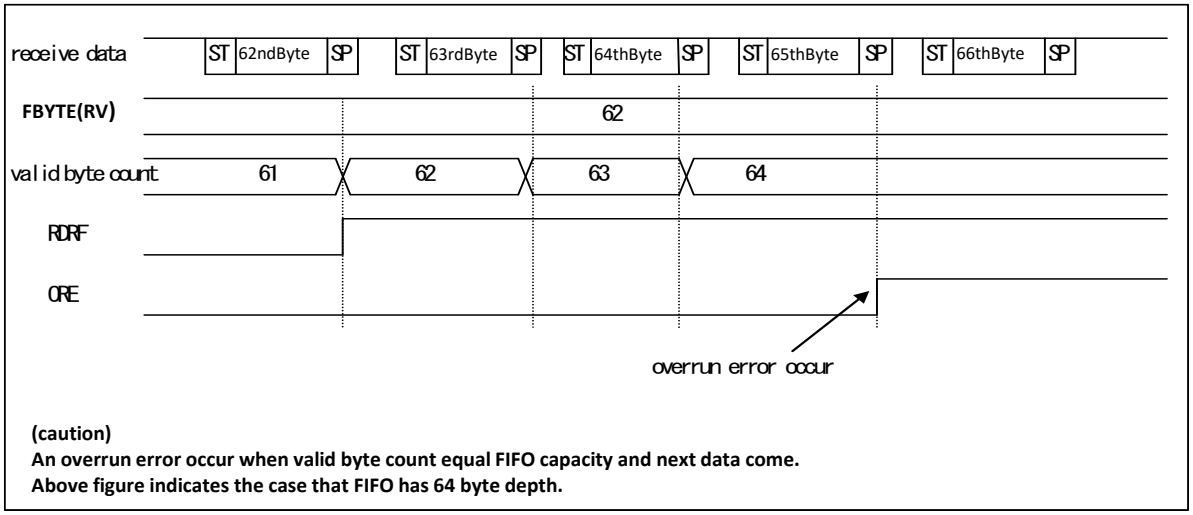


Figure 2-5 Set Timing of ORE (Overrun Error) Flag Bit



(caution)
An overrun error occur when valid byte count equal FIFO capacity and next data come.
Above figure indicates the case that FIFO has 64 byte depth.

2.1.3. Occurrence of Transmission Interrupts and Flag Set Timing

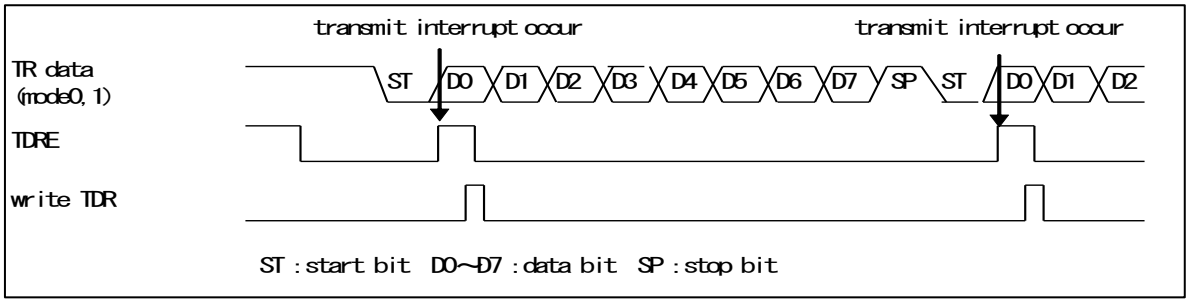
A data transmission interrupt occurs in the following cases: 1. The transmission data is transferred from the transmission data register (TDR) to the transmission shift register (SSR:TDRE = 1) to start data transmission. 2. Transmission is not performed (SSR:TBI = 1).

Occurrence of Transmission Interrupts and Flag Set Timing

Set Timing of Transmission Data Empty Flag (TDRE)

The transfer of data from the transmission data register (TDR) to the transmission shift register allows the next data to be written (SSR:TDRE = 1). At this time, a transmission interrupt occurs if transmission interrupt is enabled (SCR:TIE = 1). The TDRE bit is a read-only bit, and the SSR:TDRE bit is cleared to "0" when data is written to the transmission data register (TDR).

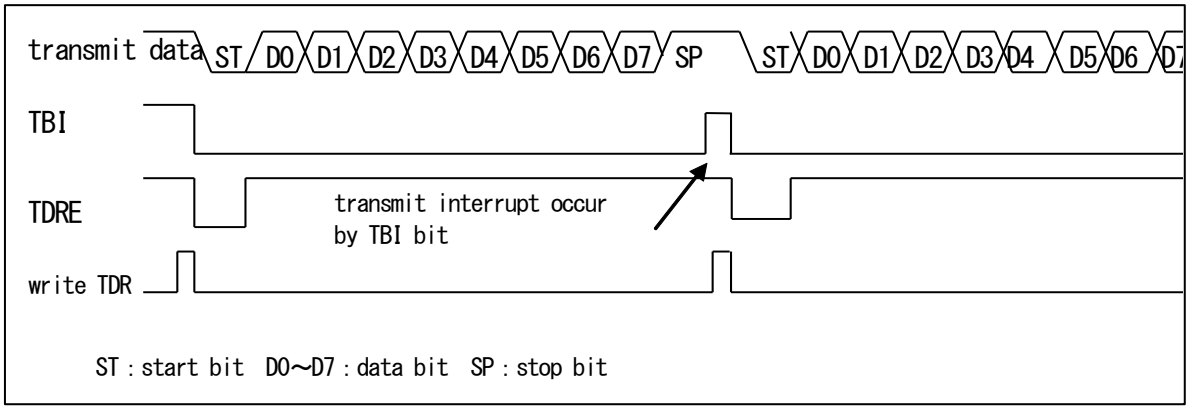
Figure 2-6 Set Timing of Transmission Data Empty Flag (SSR:TDRE)



Set Timing of Transmission Bus Idle Flag (TBI)

The SSR:TBI bit is set to "1" during those periods in which the transmission data register is empty (SSR:TDRE = 1) and transmission is not performed. At this time, a transmission interrupt occurs if transmission bus idle interrupt is enabled (SCR:TBIE = 1). Setting transmission data to the transmission data register (TDR) clears the TBI bit and the transmission interrupt request.

Figure 2-7 Set Timing of Transmission Bus Idle Flag (TBI)



2.1.4. Occurrence of Interrupts and Flag Set Timing When Using Transmission FIFO

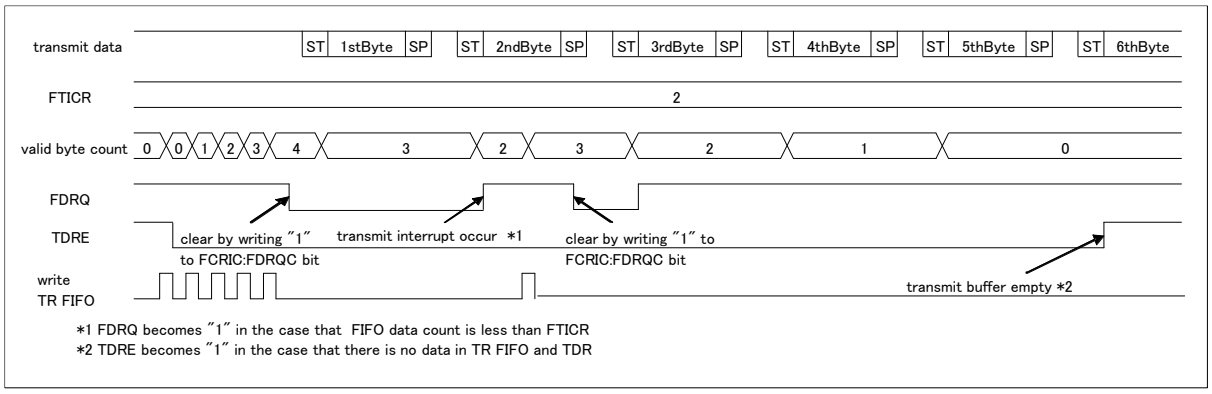
An interrupt occurs during the use of the transmission FIFO when the data count in the transmission FIFO is equal to or less than the value set in the transmission FIFO interrupt control register (FTICR).

Occurrence of Transmission Interrupts and Flag Set Timing When Using Transmission FIFO

The occurrence of interrupts during the use of the transmission FIFO is determined by the value set in the FTICR register.

- When the data count in the transmission FIFO is equal to or less than the value set in the FTICR register, the FIFO transmission data request bit (FCR1:FDRQ) is set to "1".
  At this time, a transmission interrupt occurs if the FIFO transmission interrupt is enabled (FCR1:FTIE = 1).
- When you write the necessary data to the transmission FIFO after a transmission interrupt occurs, write "1" to the FIFO transmission data request clear bit (FCR1C:FDRQC) to clear the interrupt request.
- The FIFO transmission data request bit (FCR1:FDRQ) is set to "0" when the transmission FIFO becomes full.
- Whether data is present in the transmission FIFO can be checked by reading the FIFO byte register (FBYTE). This can also be checked by reading the transmission FIFO interrupt control register (FTICR).
  FBYTE = 0x00 indicates that the transmission FIFO does not contain data.

Figure 2-8 Timing of Transmission Interrupt Occurrence When Transmission FIFO Is Used



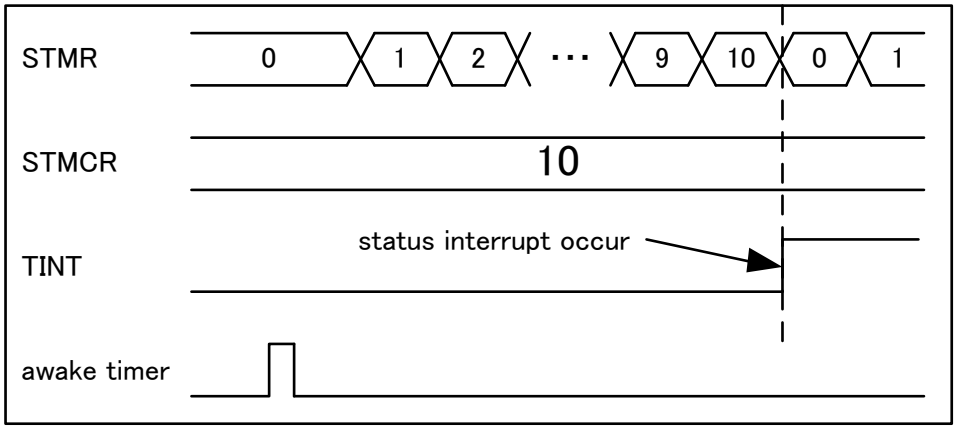
2.1.5. Occurrence of Timer Interrupts and Flag Set Timing

A timer interrupt occurs when the value of the serial timer register (STMR) coincides with that of the serial timer comparison register (STMCR).

Timer Interrupt Generation and Flag Set Timing

The timer interrupt flag (SACSR:TINT) is set to "1" when the value of the serial timer register (STMR) coincides with that of the serial timer comparison register.
At this time, a status interrupt occurs if the timer interrupt is enabled (SACSR:TINTE = 1).

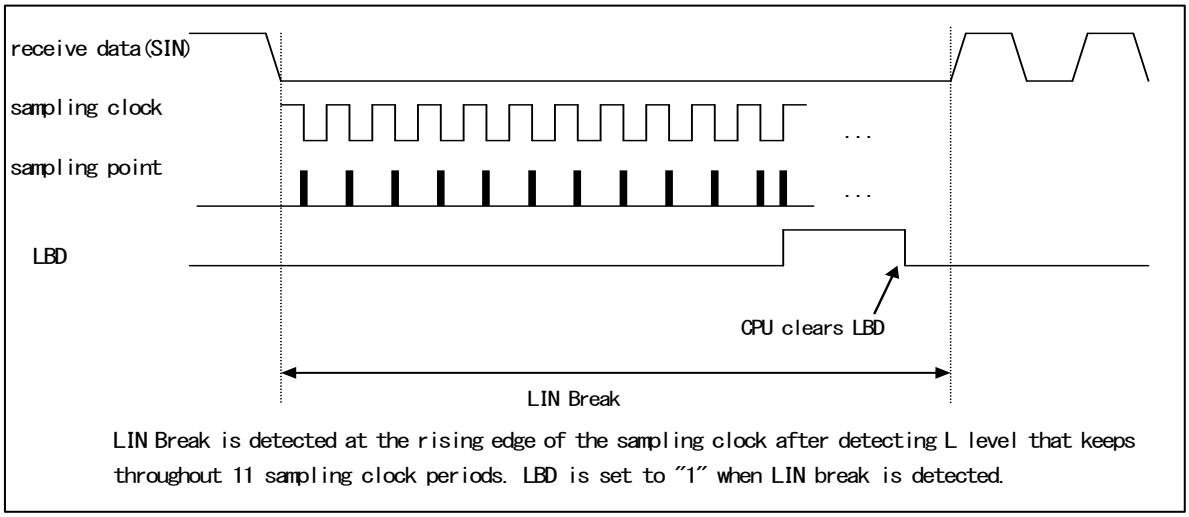
Figure 2-9 Occurrence Timing of Timer Interrupt



2.1.6. Set Timing of LIN Break Field Detection Flag (LBD)

The LBD bit is set to "1" when the serial input (SIN) has "0" input for an interval of 11 bits or more. At this time, a status interrupt occurs if the LIN Break Field interrupt is enabled (ESCR:LBIE = 1).

Figure 2-10 Set Timing of LBD (LIN Break Field Detection) Flag



Note:

- *During the reception of a LIN Break Field, a framing error (SSR:FRE=1) is detected before the detection of a LIN Break Field if reception is enabled (SCR:RXE=1).*

2.1.7. Occurrence of Sync Field Detection Interrupt and Flag Set Timing

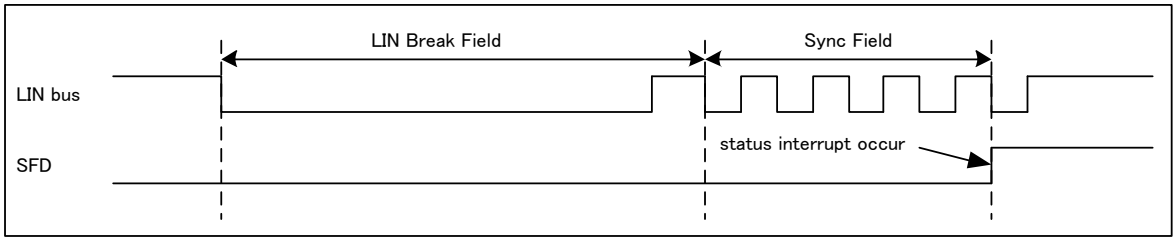
A Sync Field detection interrupt occurs if the detection of the Sync Field is complete.

Occurrence of Sync Field Detection Interrupt and Flag Set Timing

The Sync Field detection flag (SACSR:SFD) is set to "1" once the 5th falling edge of the LIN bus is detected in the SyncField when auto baud rate adjustment is enabled (SACSR:AUTE = 1).

At this time, a status interrupt occurs if the Sync Field interrupt is enabled (SACSR:SFDE = 1).

Figure 2-11 Timing of Sync Field Detection Interrupt



2.2. Assist Mode

The LIN interface (v2.1) can generate reception interrupts, transmission interrupts, and status interrupts. In assist mode, an interrupt request can be generated due to the following factors:

- When reception data is set in the reception data register (RDR) or when a reception error occurs.
- When transmission is started after transmission data is transferred from the transmission data register (TDR) to the transmission shift register.
- Transmission bus idle (no transmission operation)
- Transmission FIFO data request
- LIN Break Field detection
- LIN Sync Field detection
- The serial timer comparison value (STMCR) matches the serial timer value (STMR)
- Detection of the completion of the LIN auto header or the checksum calculation

LIN Interface (v2.1) Interrupts (Assist Mode)

Table 2-4 shows the interrupt control bits of the LIN interface (v2.1) and the interrupt factors in assist mode.

Table 2-4 LIN Interface (v2.1) Interrupt Control Bits and Interrupt Factors (Assist Mode)

Interrupt Type	Interrupt Request Flag Bit	Flag Register	Interrupt Factor	Interrupt Factor Enable Bit	Clearing of Interrupt Request Flag
Reception	RDRF	SSR	1-byte reception	SCR:RIE	- Reading of reception data (RDR) - Software reset (SCR:UPCL = 1)
			Reception of as much data as the amount set in FBYTE		- Reading of reception data (RDR) until the reception FIFO becomes empty - Software reset (SCR:UPCL = 1)
			Detection of reception idle for the 8-bit time or longer while the FRIIE bit is "1" and the reception FIFO contains valid data		
	ORE	SSR	Overrun error	SCR:RIE ECR:REIE	- Writing "1" to the reception error flag clear bit (SSR:REC) - Software reset (SCr:UPCL = 1)
	FRE	SSR	Framing error		
	LBSER	LAMESR	LIN bus error detection	LAMIER: LBSERIE	- Writing "1" to LAMESRC:LBSERC - Software reset (SCR:UPCL = 1)
	LSFER	LAMESR	LIN Sync Data error detection	LAMIER: LSFERIE	- Writing "1" to LAMESRC:LSFERC - Software reset (SCR:UPCL = 1)
	LPTER	LAMESR	LIN ID parity error detection	LAMIER: LPTERIE	- Writing "1" to LAMESRC:LPTERC - Software reset (SCR:UPCL = 1)
	LCSER	LAMESR	LIN checksum error detection	LAMIER: LCSERIE	- Writing "1" to LAMESRC:LCSERC - Software reset (SCR:UPCL = 1)
	RXUDR	ESR	Reception FIFO underrun	ECR:REIE	- Writing "1" to the reception FIFO underrun flag clear bit (ESRC:RXUDRC) - Software reset (SCR:UPCL = 1) - Disable reception FIFO - When FCR1:FSEL=0, set FCR0:FE2=0 - When FCR1:FSEL=1, set FCR0:FE1=0

Interrupt Type	Interrupt Request Flag Bit	Flag Register	Interrupt Factor	Interrupt Factor Enable Bit	Clearing of Interrupt Request Flag
	RBERR	ESR	Reception block transfer error	ECR:REIE	- Writing "1" to the reception block transfer error clear bit - Software reset (SCR:UPCL=1) - Disable Block transfer mode (ECR:RXBLKEN=0)
Transmission	TDRE	SSR	The transmission register is empty.	SCR:TIE	Writing to the transmission data (TDR) , or writing "1" to the transmission FIFO operation enable bit (FCR0:FE1 or FCR0:FE2) (retransmission) when the bit value is "0" and the transmission FIFO contains valid data*1
	TBI	SSR	No transmission operation	SCR:TBIE	Writing to the transmission data (TDR), writing "1" to the LIN Break Field setting bit (LBR), or writing "1" to the transmission FIFO operation enable bit (retransmission) when the bit value is "0" and the transmission FIFO contains valid data *1
	FDRQ	FCR1	The amount of data in the transmission FIFO is equal to or less than the FTICR setting value or it is empty.	FCR1:FTIE	Writing "1" to the FIFO transmission data request clear bit (FCR1C:FDRQC) , or the transmission FIFO is full
	TXOVR	ESR	Transmission FIFO overrun	ECR:TEIE	- Writing "1" to the transmission FIFO overrun flag clear bit (ESRC:TXOVRC) - Software reset (SCR:UPCL=1) - Disable transmission FIFO - When FCR1:FSEL=0, set FCR0:FE1=0 - When FCR1:FSEL=1, set FCR0:FE2=0
	TBERR	ESR	Transmission block transfer error	ECR:TEIE	- Writing "1" to the transmission block transfer error clear bit (ESRC:TBERRC) - Software reset (SCR:UPCL=1) - Disable Block transfer (ECR:TXBLKEN=0)
Status (Assist Mode)	LBD	SSR	LIN Break Field detection	ESCR:LBIE	- Writing "1" to the SSRC:LBDC bit - Software reset (SCR:UPCL=1)
	SFD	SACSR	Sync Field detection	SACSR: SFDE	- Writing "1" to the Sync Field detection flag clear bit (SACSRC:SFDC) - Software reset (SCR:UPCL=1)
	TINT	SACSR	Coincidence of the value in the serial timer register (STMR) and that in the serial timer comparison register (STMCR)	SACSR: TINTE	- Writing "1" to the timer interrupt flag clear bit (SACSRC:TINTC) - Software reset (SCR:UPCL=1)
	LAHC	LAMSR	Completion of auto header	LAMIER: LAHCIE	- Writing "1" to LAMSRC:LAHCC - Reading from the ID register (LAMRID) - Software reset (SCR:UPCL = 1)
	LCSC	LAMSR	Completion of checksum calculation	LAMIER: LCSCIE	- Writing "1" to LAMSRC:LCSCC - Software reset (SCR:UPCL = 1)

*1: Set the TIE bit to "1" after the TDRE bit becomes "0".

Switching of Error Interrupt Request Output

The method of outputting an error interrupt request varies depending on the setting of the error interrupt request output switching bit (ECR:EISEL), transmission/reception error interrupt enable bits (ECR:REIE, TEIE), and transmission/reception interrupt enable bits (SCR:RIE, TIE).

Table 2-5 Method of Outputting Transmission Interrupt Requests

ECR:EISEL	Interrupt Output	Target Interrupt Enable Bit	Interrupt Factor
0	Transmission interrupt request output	SCR:TIE	Transmission data empty (SSR:TDRE)
		SCR:TBIE	No transmission operation (SSR:TBI)
		FCR1:FTIE	Transmission FIFO empty (FCR1:FDRQ)
		ECR:TEIE	Transmission FIFO overrun (ESR:TXOVR) Transmission block transfer error (ESR:TBERR)
	Transmission error interrupt request output	-	No interrupt factor
1	Transmission interrupt request output	SCR:TIE	Transmission data empty (SSR:TDRE)
		SCR:TBIE	No transmission operation (SSR:TBI)
		FCR1:FTIE	Transmission FIFO empty (FCR1:FDRQ)
	Transmission error interrupt request output	ECR:TEIE	Transmission FIFO overrun (ESR:TXOVR) Transmission block transfer error (ESR:TBERR)

Table 2-6 Method of Outputting Reception Interrupt Requests

ECR:EISEL	Interrupt Output	Target Interrupt Enable Bit	Interrupt Factor
0	Reception interrupt request output	SCR:RIE	Reception full (SSR:RDRF) Framing error (SSR:FRE) Overrun error (SSR:ORE)
		LAMIER:LBSERIE	LIN bus error (LAMESR:LBSER)
		LAMIER:LSFERIE	LIN Sync Data error (LAMESR:LSFER)
		LAMIER:LPTERIE	LIN ID parity error (LAMESR:LPTER)
		LAMIER:LCSERIE	LIN Check Sum error (LAMESR:LCSER)
		ECR:REIE	Reception FIFO underrun (ESR:RXUDR) Reception block transfer error (ESR:RBERR)
	Reception error interrupt request output	-	No interrupt factor
1	Reception interrupt request output	SCR:RIE	Reception full (SSR:RDRF)
	Reception error interrupt request output	LAMIER:LBSERIE	LIN bus error (LAMESR:LBSER)
		LAMIER:LSFERIE	LIN Sync Data error (LAMESR:LSFER)
		LAMIER:LPTERIE	LIN ID parity error (LAMESR:LPTER)
		LAMIER:LCSERIE	LIN Check Sum error (LAMESR:LCSER)
		ECR:REIE	Framing error (SSR:FRE) Overrun error (SSR:ORE) Reception FIFO underrun (ESR:RXUDR) Reception block transfer error (ESR:RBERR)

2.2.1. Occurrence of Reception Interrupts and Flag Set Timing in Assist Mode

There are 2 reception interrupt types: Reception completion (SSR:RDRF) and reception error (SSR:ORE, FRE, LAMESR:LBSER, LSFER, LPTER, LCSER).

Occurrence of Reception Completion Interrupts and Flag Set Timing

In assist mode (LAMCR:LAMEN = 1), reception data is stored in the reception data register (RDR) every time the stop bit of each of the following fields is detected. A flag is set (SSR:RDRF = 1) once reception is complete. At this time, a reception interrupt occurs if reception interrupt is enabled (SCR:RIE = 1).

- Data Field for Response
- ID Field when the ID register is not used (LAMCR:LIDEN = 0)

The timing at which the reception data full flag bit (SSR:RDRF) is set is the same as that in manual mode, as described in "2.1.1 Occurrence of Reception Interrupts and Flag Set Timing." See "Figure 2-1 Set Timing of RDRF (Reception Data Full) Flag Bit."

Notes:

- *If a reception error occurs, the data in the reception data register (RDR) will be invalid.*
- *Suppose that the setting is such that the LIN assist mode reception ID register is used for the reception of ID Fields (LAMCR:LIDEN = 1). Then, when an ID Field is received, the received ID value is not stored into the reception data register (RDR) and the reception data full flag bit (SSR:RDRF) is not set.*
- *In slave operation, suppose that the setting is such that the reception data register (RDR) is used for the reception of ID Fields (LAMCR:LIDEN = 0). Then, when an ID Field is received, the received ID value is stored to the reception data register (RDR) and the reception data full flag bit is set (SSR:RDRF = 1). Check the ID value after the LIN auto header completion flag is set (LAMSR:LAHC = 1).*
- *Neither the Sync Field nor the checksum are stored to the reception data register (RDR) and the reception data full flag bit (SSR:RDRF) is not set.*

Occurrence of Framing Error Interrupt and Flag Set Timing

In assist mode (LAMCR:LAMEN = 1), a framing error is detected and the framing error flag is set (SSR:FRE = 1) if the "L" level is detected on the stop bit when the Sync Field, ID Field, data, and checksum are received. A reception interrupt occurs if reception interrupt is enabled (SCR:RIE = 1). The master node that transmits the Sync Field and ID Field or the master/slave node that transmits the checksum executes self-check on the transmitted data. If the "L" level is detected on the stop node, then a framing error is detected and the framing error flag is set (SSR:FRE = 1). A reception interrupt occurs if reception interrupt is enabled (SCR:RIE = 1).

Transmission/reception processing of a header and response in assist mode stops once a framing error is detected.

The operation enable bit for the reception FIFO is cleared (FCR0:FE1 = 0 or FCR0:FE2 = 0) while the framing error flag is set (SSR:FRE = 1).

The timing at which the framing error flag bit (SSR:FRE) is set is the same as that in manual mode, as described in "2.1.1 Occurrence of Reception Interrupts and Flag Set Timing." See "Figure 2-2 Set Timing of FRE (Framing Error) Flag Bit."

Occurrence of Overrun Error Interrupt and Flag Set Timing

An overrun error is detected if the following data reception has been detected before reading of the reception data (SSR:RDRF = 1). The overrun error flag is set (SSR:ORE = 1) when reception is complete (SSR:RDRF = 1). A reception interrupt occurs if reception interrupt is enabled (SCR:RIE = 1).

The reception processing of a header section and response section in assist mode stops once an overrun error is detected.

The operation enable bit for the reception FIFO is cleared (FCR0:FE1 = 0 or FCR0:FE2 = 0) while the overrun error flag is set (SSR:ORE = 1).

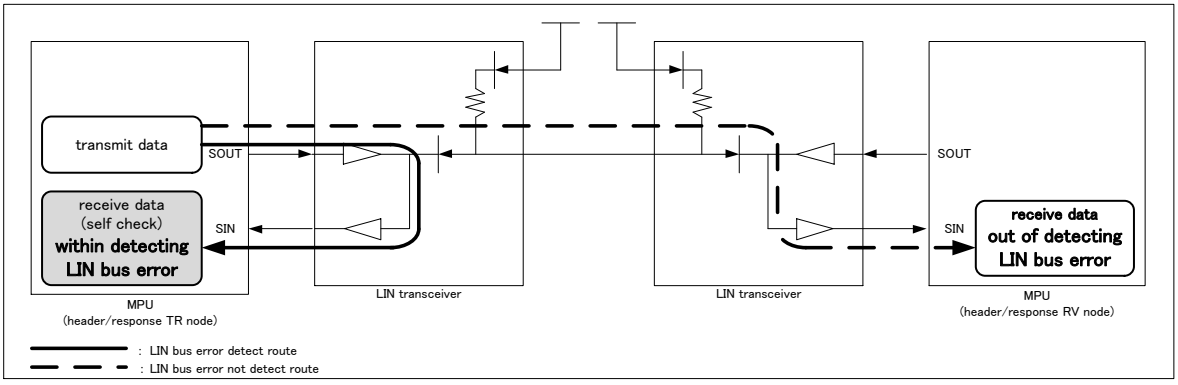
The timing at which the overrun error flag bit (SSR:ORE) is set is the same as that in manual mode, as described in "2.1.1 Occurrence of Reception Interrupts and Flag Set Timing." See "Figure 2-3 Set Timing of ORE (Overrun Error) Flag Bit."

Occurrence of LIN Bus Error Detection Interrupt and Flag Set Timing

In assist mode (LAMCR:LAMEN = 1), LIN bus error detection is executed by the self-check on the side that transmits the header/response. The side that receives the header/response cannot execute LIN bus error detection.

Figure 2-12 shows the LIN bus error detection target.

Figure 2-12 LIN BUS Error Detection Target



The target range of the LIN bus error detection is the start bit and byte data of the LIN Break and Sync Field/ID Field/Data Field/Check Sum Field. The stop bit is not included in the LIN bus error detection target. A framing error (SSR:FRE = 1) is detected when the "L" level is detected on the stop bit.

The transmission processing of a header section and response section in assist mode stops once a LIN bus error is detected.

The LIN auto header completion flag (LAMSR:LAHC = 1) is set even if a LIN bus error occurs when the ID Field transmission is complete.

Occurrence of LIN Bus Error Detection Interrupt and Flag Set Timing on the Master Side

The master side (SCR:MS = 0) performs LIN bus error detection when transmitting a header/response.

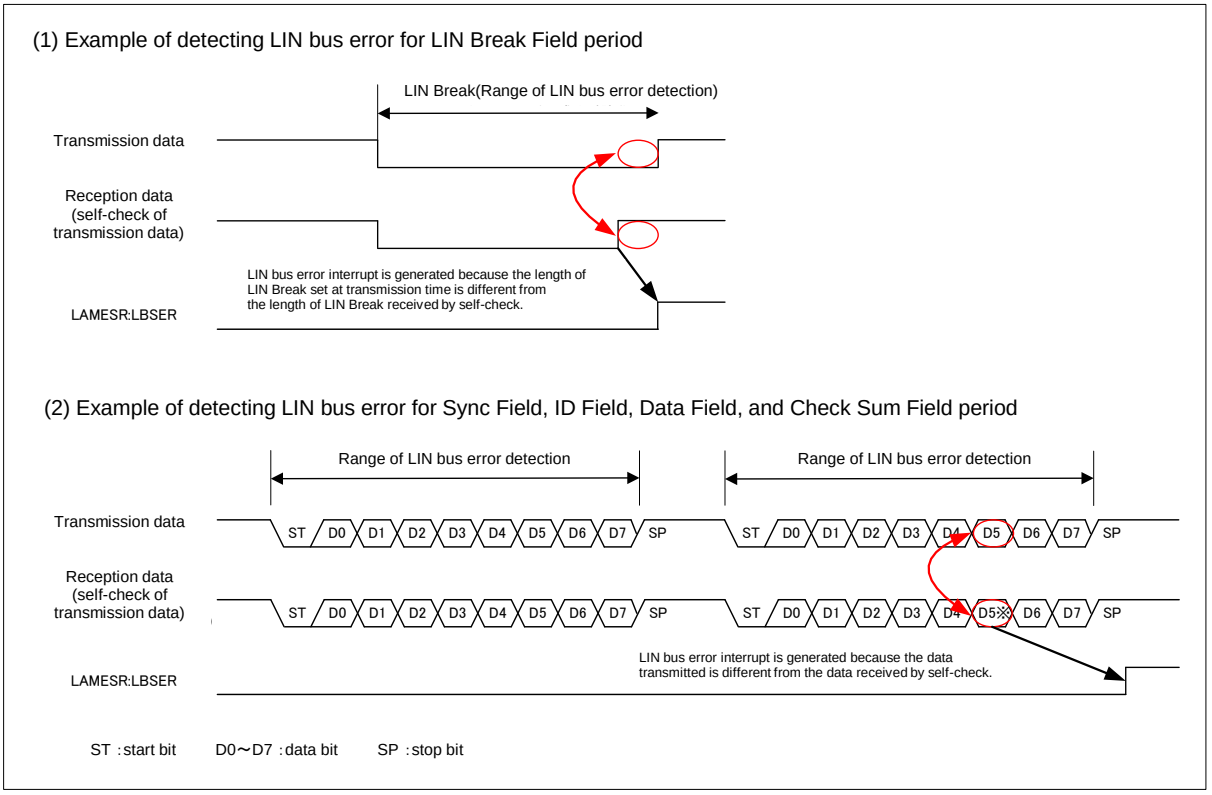
If an abnormality is detected as a result of comparing the transmission LIN Break length and the reception LIN Break length or the transmission data and reception data, then a LIN bus error is detected and a flag is set (LAMESR:LBSER = 1). A reception interrupt occurs if the interrupt is enabled (LAMIER:LBSERIE = 1).

Occurrence of LIN Bus Error Detection Interrupt and Flag Set Timing on the Slave Side

The slave side (SCR:MS = 1) performs LIN bus error detection when transmitting a response.

If an abnormality is detected as a result of comparing transmission data and reception data, then a LIN bus error is detected and a flag is set (LAMESR:LBSER = 1). A reception interrupt occurs if the interrupt is enabled (LAMIER:LBSERIE = 1).

Figure 2-13 Set Timing of LIN Bus Error Detection Flag (LAMESR:LBSER)

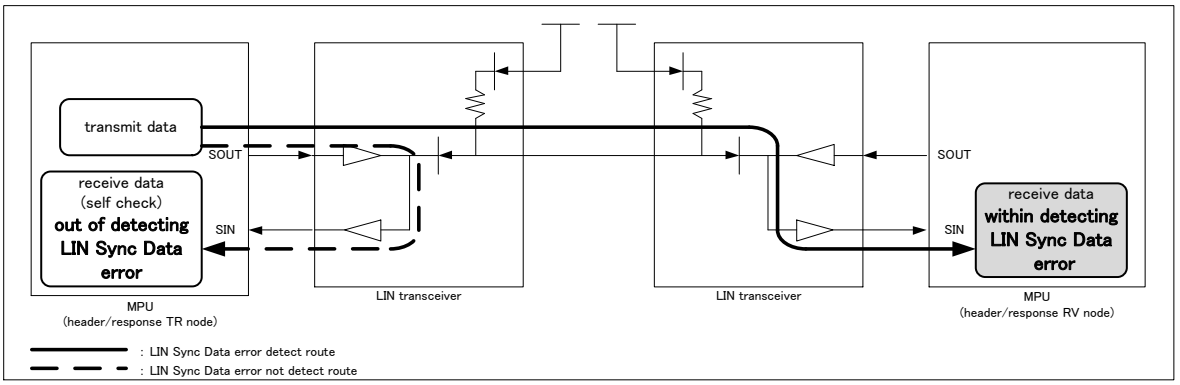


Occurrence of LIN Sync Data Error Detection Interrupt and Flag Set Timing

The LIN Sync Data error detection is executed if the auto baud rate adjustment (SACSR:AUTE = 0) is disabled when slave mode (SCR:MS = 1) is set in assist mode (LAMCR:LAMEN = 1).

Figure 2-14 shows the LIN Sync Data error detection target.

Figure 2-14 LIN Sync Data Error Detection Target

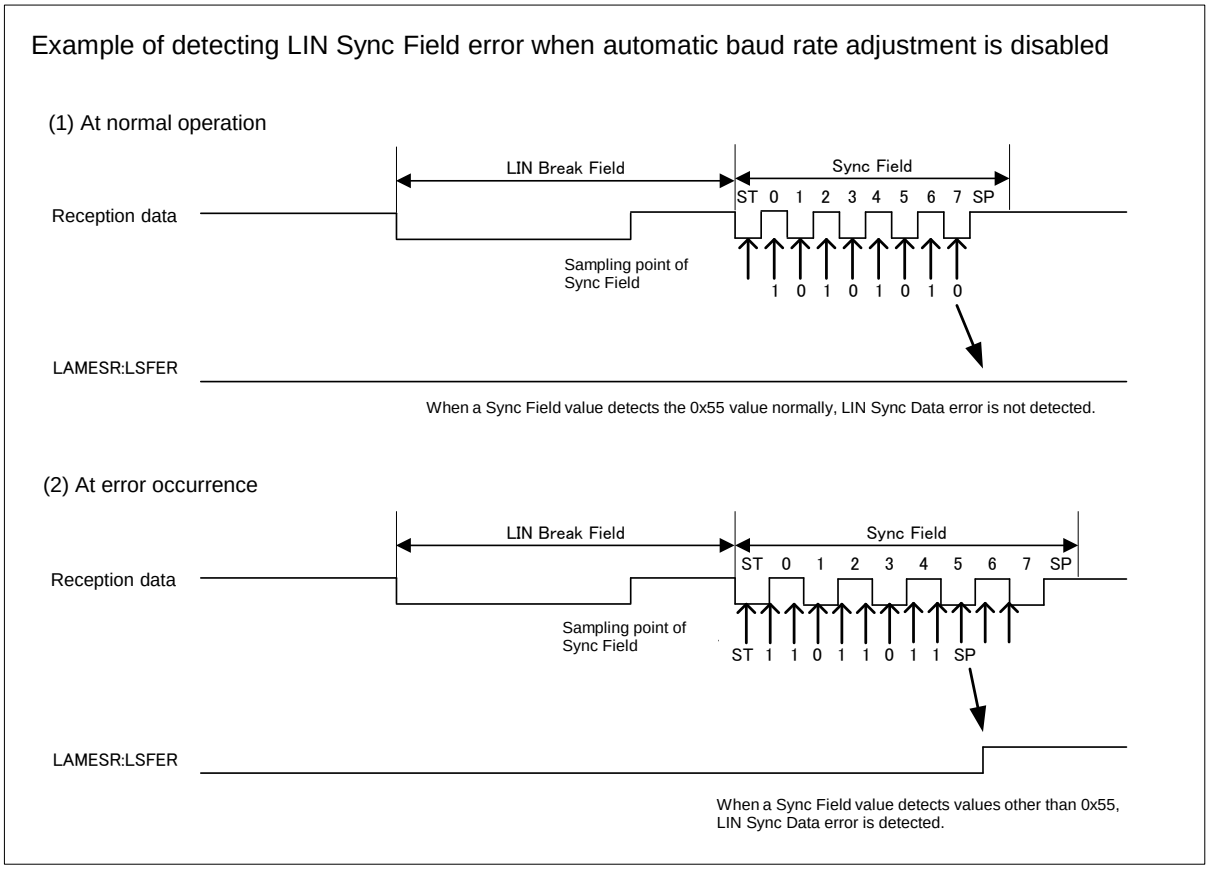


The target range of LIN Sync Data error detection is the start bit and byte data of the Sync Field. The stop bit is not included in the LIN Sync Data error detection target. A framing error (SSR:FRE = 1) is detected when the "L" level is detected on the stop bit.

Occurrence of LIN Sync Data Error Eetection Interrupt and Flag Set Timing When the Auto Baud Rate Adjustment is Disabled

In slave mode (SCR:MS = 1) when auto baud rate adjustment is disabled (SACSR:AUTE = 0), the data value of the Sync Field is checked. If any value other than 0x55 is detected, then a LIN Sync Data error is detected and a flag is set (LAMESR:LSFER = 1). At this time, a reception interrupt occurs if the interrupt is enabled (LAMIER:LSFERIE = 1).

Figure 2-15 Set Timing of LIN Sync Data Error Detection Flag (LAMESR:LSFER) (When Auto Baud Rate Adjustment is Disabled)

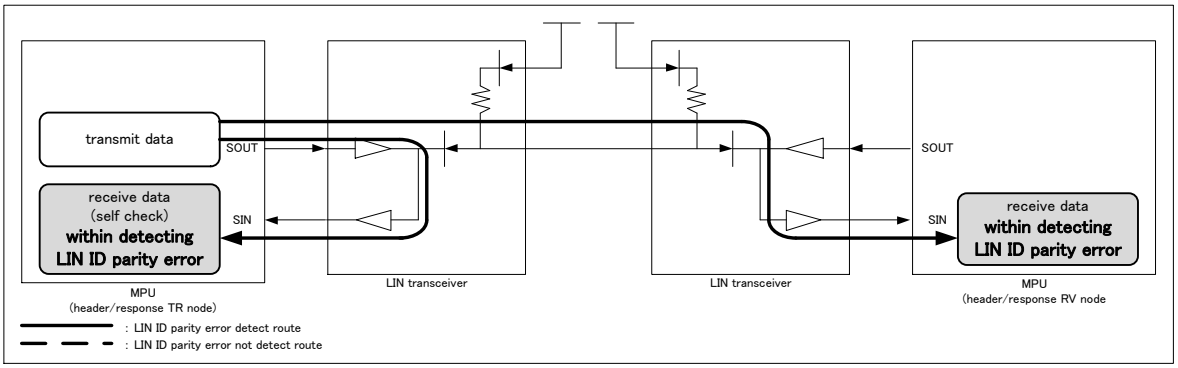


Occurrence of LIN ID Parity Error Detection Flag Interrupt and Flag Set Timing

In assist mode (LAMCR:LAMEN = 1), the LIN ID parity error detection is executed on the slave that receives the ID Field and the self-check is performed on the master that transmits the ID Field.

Figure 2-16 shows the LIN ID parity error detection target.

Figure 2-16 LIN ID Parity Error Detection Target



The target range of LIN ID parity error detection is the ID data and the byte data of the parity. The start and stop bits are not included in the LIN ID parity error detection target. A framing error occurs (SSR:FRE = 1) if the "L" level is detected on the stop bit.

In LIN assist mode (LAMCR:LAMEN = 1), the auto header completion flag is set (LAMSR:LAHC = 1) if a LIN ID parity error occurs during auto header transmission/reception.

Transmission/reception processing for a response in assist mode stops once a LIN ID parity error is detected.

Occurrence of LIN ID Parity Error Detection Interrupt and Flag Set Timing on the Master Side

The master (SCR:MS = 0) set in assist mode (LAMCR:LAMEN = 1) performs LIN ID parity error detection when transmitting the ID Field. The master executes a parity operation for the 6-bit Frame ID set in the transmission data register (TDR) or the LIN assist mode transmission ID register (LAMTID). Then, the master automatically creates and transmits an ID Field.

If an ID Field is received for self-check and if the parity operation result of the Frame ID value differs from the received parity value, then a LIN ID parity error is detected and a flag is set (LAMESR:LPTER = 1).

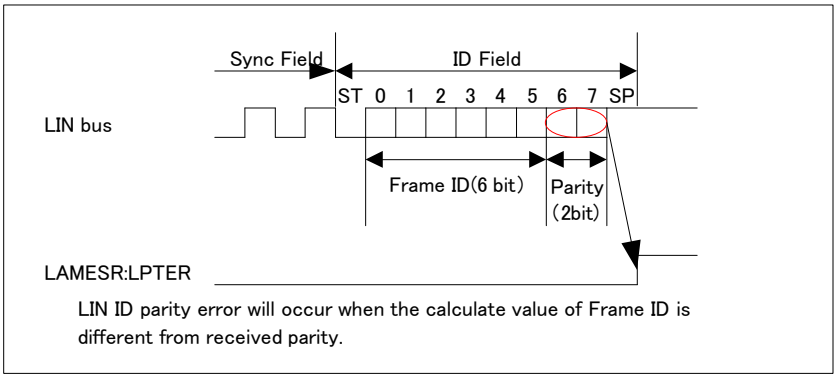
At this time, a reception interrupt occurs if the interrupt is enabled (LAMIER:LPTERIE = 1).

Occurrence of LIN ID Parity Error Detection Interrupt and Flag Set Timing on the Slave Side

The slave (SCR:MS = 1) set in assist mode (LAMCR:LAMEN = 1) performs LIN ID parity error detection when it receives the ID Field. If the ID Field is received and if the parity operation result of the Frame ID value differs from the received parity value, then a LIN ID parity error is detected and a flag is set (LAMESR:LPTER = 1).

At this time, a reception interrupt occurs if the interrupt is enabled (LAMIER:LPTERIE = 1).

Figure 2-17 Set Timing of LIN ID Parity Error Detection Flag (LAMESR:LPTER)

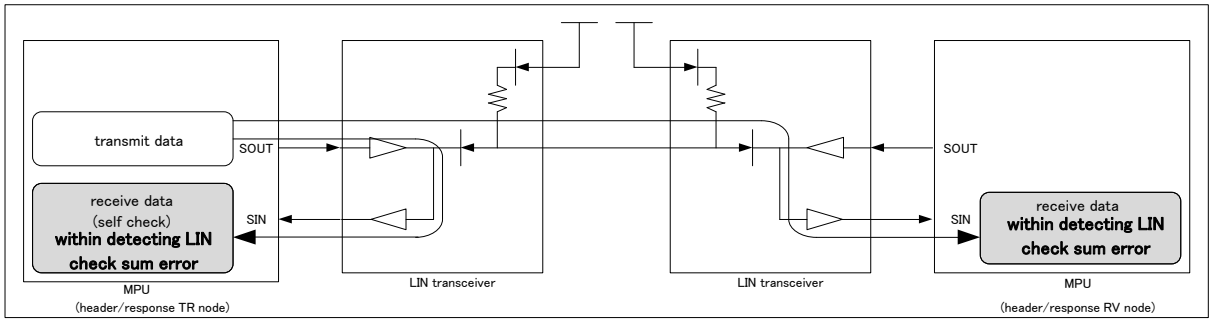


Occurrence of LIN Checksum Error Detection Flag Interrupt and Flag Set Timing

In assist mode (LAMCR:LAMEN = 1), LIN checksum error detection is executed by the self-check on the side that transmits the checksum and is also executed by the side that receives the checksum.

Figure 2-18 shows the LIN checksum error detection target.

Figure 2-18 LIN Checksum Error Detection Target



As the calculation method for the checksum that is automatically transmitted, select standard (target: data) or extended (target: ID Field + data) checksum with the LAMCR:LCSTYP bit.

Occurrence of LIN Checksum Error Detection Interrupt and Flag Set Timing When Setting Standard Checksum Calculation

When the standard checksum calculation is set (LAMCR:LCSTYP = 0), the side that transmits a response (data and checksum) performs the checksum calculation based on the transmission data for the specified LIN data length (LAMCR:LDL3-0). Then, after transmitting the last data, the transmission side automatically transmits the checksum.

The side that receives a response (data and checksum) executes the checksum calculation based on the reception data for the specified LIN data length (LAMCR:LDL3-0). If the received checksum differs from the calculation result value, then a LIN checksum error is detected and a flag is set (LAMESR:LCSER = 1).

At this time, a reception interrupt occurs if the interrupt is enabled (LAMIER:LCSERIE = 1).

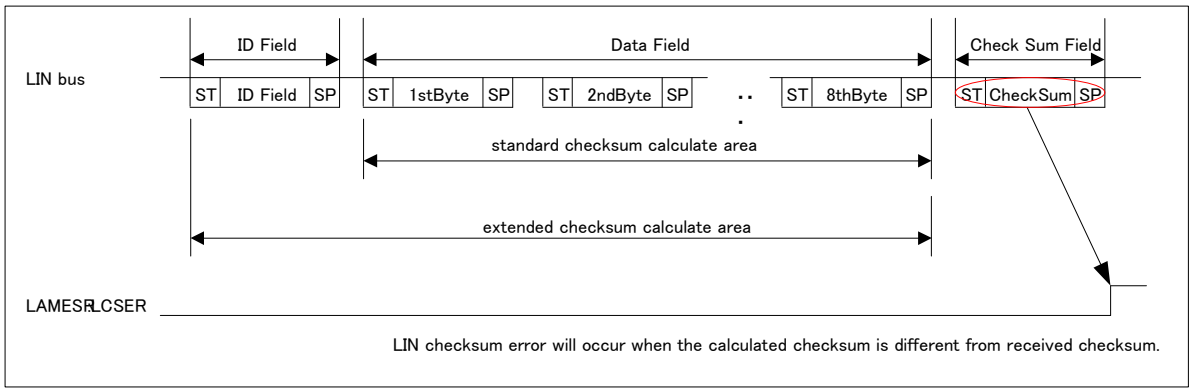
Occurrence of LIN Checksum Error Detection Interrupt and Flag Set Timing When Setting Extended Checksum Calculation

When extended checksum calculation is set (LAMCR:LCSTYP = 1), the side that transmits a response (data and checksum) performs the checksum calculation based on the ID Field value and the transmission data for the specified LIN data length (LAMCR:LDL3-0). Then, after transmitting the last data, the transmission side automatically transmits the checksum.

The side that receives a response (data and checksum) executes the checksum calculation based on the ID Field value and the reception data for the specified LIN data length (LAMCR:LDL3-0). If the received checksum differs from the calculation result value, then a LIN checksum error is detected and a flag is set (LAMESR:LCSER = 1).

At this time, a reception interrupt occurs if the interrupt is enabled (LAMIER:LCSERIE = 1).

Figure 2-19 Set Timing of LIN Checksum Error Detection Flag (LAMESR:LCSER)



Note:

- *Regardless of the standard/extended checksum calculation setting, the processing performed in assist mode stops and checksum calculation is not executed if an error (LIN bus error, LIN ID parity error, LIN Sync Data error, or framing error) is detected in the data of the header section and the response section.*

2.2.2. Occurrence of Interrupts and Flag Set Timing When Using Reception FIFO

The timing is the same as that in manual mode, as described in "2.1.2 Occurrence of Interrupts and Flag Set Timing When Using Reception FIFO"

2.2.3. Occurrence of Transmission Interrupts and Flag Set Timing

An interrupt occurs by the SSR:TDRE flag and the SSR:TBI flag at transmission.

Occurrence of Transmission Interrupts and Flag Set Timing

Set Timing of Transmission Data Empty Flag (TDRE)

The timing is the same as that in manual mode, as described in "2.1.3 Occurrence of Transmission Interrupts and Flag Set Timing."

Set Timing of Transmission Bus Idle Flag (TBI)

The transmission bus idle flag bit (SSR:TBI) is set to "1" when all the following conditions are satisfied. At this time, a transmission interrupt occurs if the transmission bus idle interrupt is enabled (SCR:TBIE = 1).

- The transmission data register is empty (SSR:TDRE = 1) and transmission is not performed.
- A header (LIN Break Field, Sync Field, or ID Field) has not been transmitted when the master operation (SCR:MS = 0) is used in assist mode (LAMCR:LAMEN = 1).
- A response (data and checksum) has not been transmitted in assist mode (LAMCR:LAMEN = 1).

The transmission bus idle flag bit (SSR:TBI) and a transmission interrupt request are cleared in any of the following cases.

- When writing transmission data to the transmission data register (TDR)
  (The transmission data register is not empty (SSR:TDRE = 0))
- When a header (LIN Break Field, Sync Field, or ID Field) is being transmitted while the master is being used (SCR:MS = 0) in assist mode (LAMCR:LAMEN = 1).
- A response (data and checksum) is being transmitted in assist mode (LAMCR:LAMEN = 1)

Figure 2-20 Set Timing of Transmission Bus Idle Flag (TBI) (When Master Response Is Transmitted and ID Register Is Used)

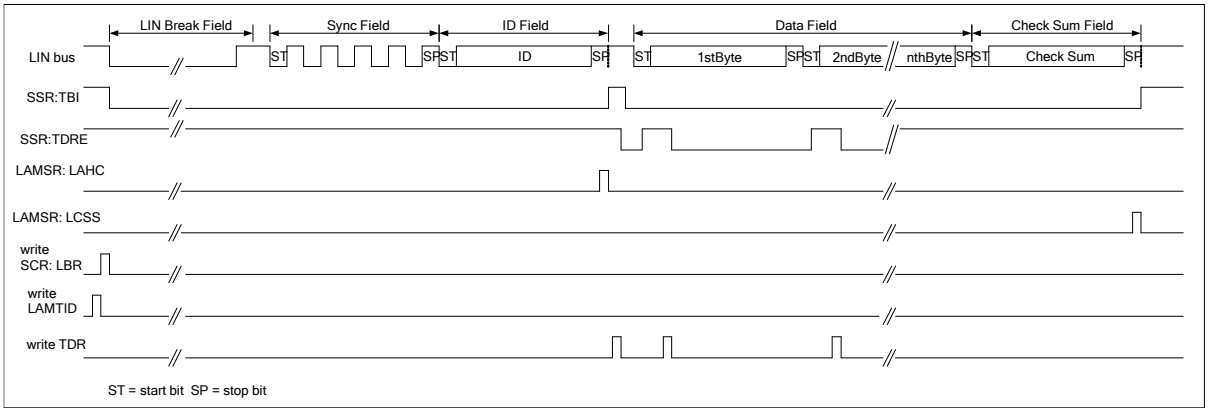


Figure 2-21 Set Timing of Transmission Bus Idle Flag (TBI) (When Master Response Is Transmitted and ID Register Is Not Used)

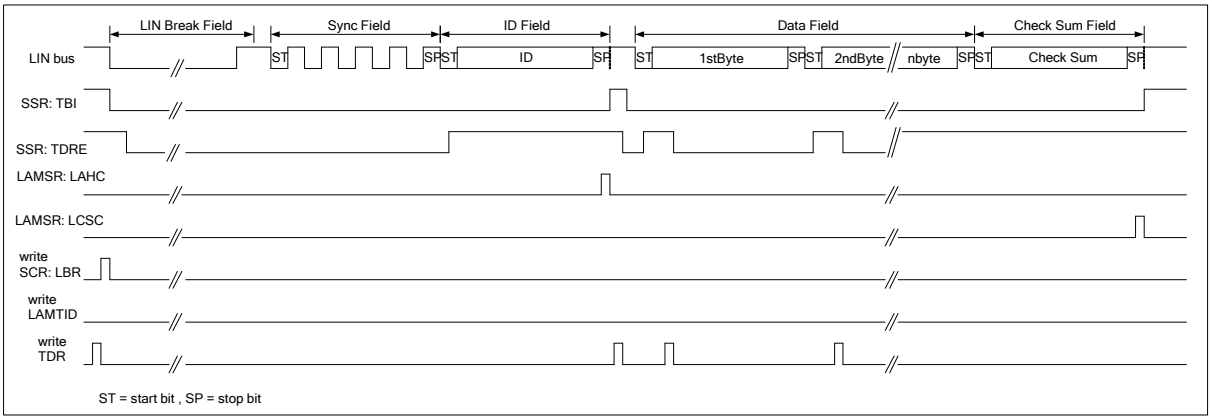
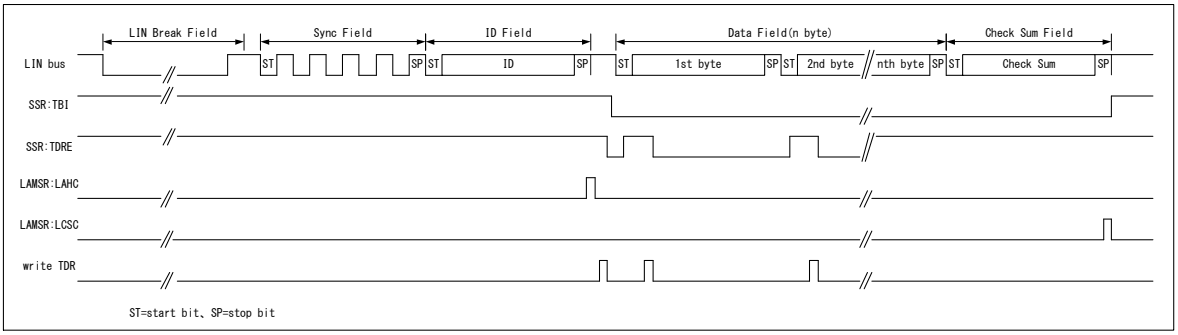


Figure 2-22 Set Timing of Transmission Bus Idle Flag (TBI) (When Slave Response Is Transmitted)



2.2.4. Occurrence of Interrupts and Flag Set Timing When Using Transmission FIFO

The timing is the same as that in manual mode, as described in "2.1.4 Occurrence of Interrupts and Flag Set Timing When Using Transmission FIFO."

2.2.5. Occurrence of Timer Interrupts and Flag Set Timing

The timing is the same as that in manual mode, as described in "2.1.5 Occurrence of Timer Interrupts and Flag Set Timing"

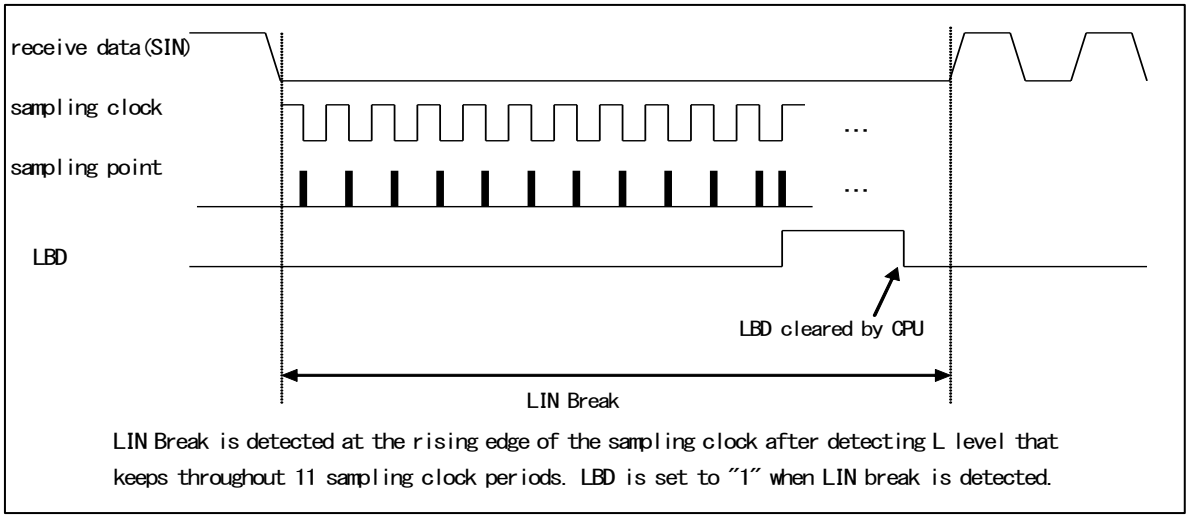
2.2.6. Occurrence of Status Interrupt and Flag Set Timing in Assist Mode

In assist mode, a status interrupt occurs when the LIN Break Field (SSR:LBD) is detected, the Sync Field is detected (SACSR:SFD), the auto header is complete (LAMSR:LAHC), or checksum calculation is complete (LAMSR:LCSC).

Set Timing of LIN Break Field Detection Flag (LBD)

The LBD bit is set to "1" when the serial input (SIN) has "0" input for an interval of 11 bits or more. At this time, a status interrupt occurs if the LIN Break Field interrupt is enabled (ESCR:LBIE = 1).

Figure 2-23 Set Timing of LBD (LIN Break Field Detection) Flag



Note:

- *During the reception of a LIN Break Field, a framing error is detected before the detection of a LIN Break Field if reception is enabled (SCR:RXE = 1).*

Sync Field Detection Interrupt and Flag Set Timing

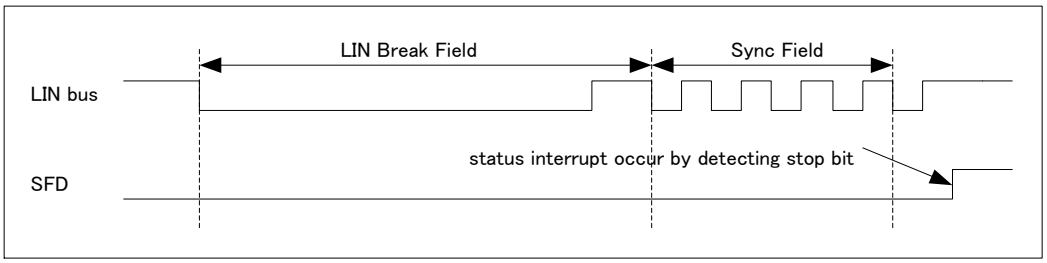
The Sync Field detection interrupt and flag set timing varies as follows based on the setting value of the auto baud rate adjustment bit (SACSR:AUTE).

■ When auto baud rate adjustment is enabled (SACSR:AUTE = 1)

The timing is the same as that in manual mode, as described in "2.1.7Occurrence of Sync Field Detection Interrupt and Flag Set Timing."

- When auto baud rate adjustment is disabled (SACSR:AUTE = 0)
  The Sync Field detection flag is set (SACSR:SFD = 1) when the stop bit of the Sync Field is detected. A status interrupt occurs if the interrupt is enabled (SACSR:SFDE = 1).

Figure 2-24 Set Timing of Sync Field Detection Flag (SACSR:SFD) When Auto Baud Rate Adjustment is Disabled (SACSR:AUTE = 0)

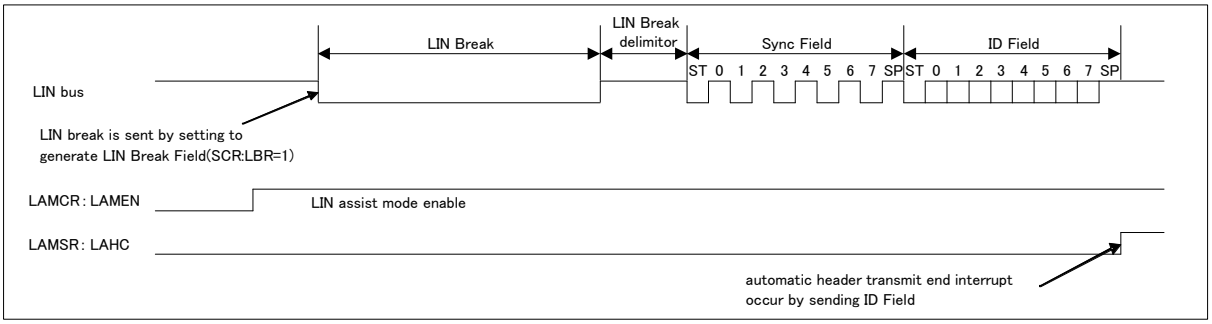


Auto Header Completion Interrupt and Flag Set Timing at Transmission

When the master is set in LIN assist mode (LAMCR:LAMEN = 1), a flag is set (LAMSR.LAHC = 1) once transmission of the headers from the LIN Break to the ID Field is complete. A status interrupt occurs if the interrupt is enabled (LAMIER:LAHCIE = 1).

In LIN assist mode, the auto header completion flag is set (LAMSR:LAHC = 1) even when a LIN bus error/LIN ID parity error/framing error occurs in the ID Field. However, the transmission/reception processing of the response section stops in LIN assist mode.

Figure 2-25 Set Timing of Auto Header Completion Flag (LAMSR:LAHC)

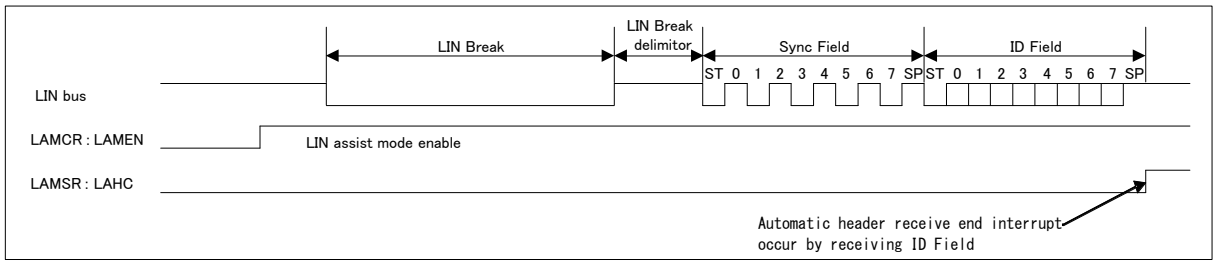


Auto Header Reception Completion Interrupt and Flag Set Timing at Reception

When the slave is set in LIN assist mode (LAMCR:LAMEN = 1), a flag is set (LAMSR:LAHC = 1) once reception of the headers from the LIN Break to the ID Field is complete. A status interrupt occurs if the interrupt is enabled (LAMIER:LAHCIE = 1).

In LIN assist mode, the auto header completion flag is set (LAMSR:LAHC = 1) even if a LIN bus error/LIN ID parity error/framing error occurs in the ID Field. However, transmission/reception processing for the response section stops in LIN assist mode.

Figure 2-26 Set Timing of Auto Header Completion Flag (LAMSR:LAHC)

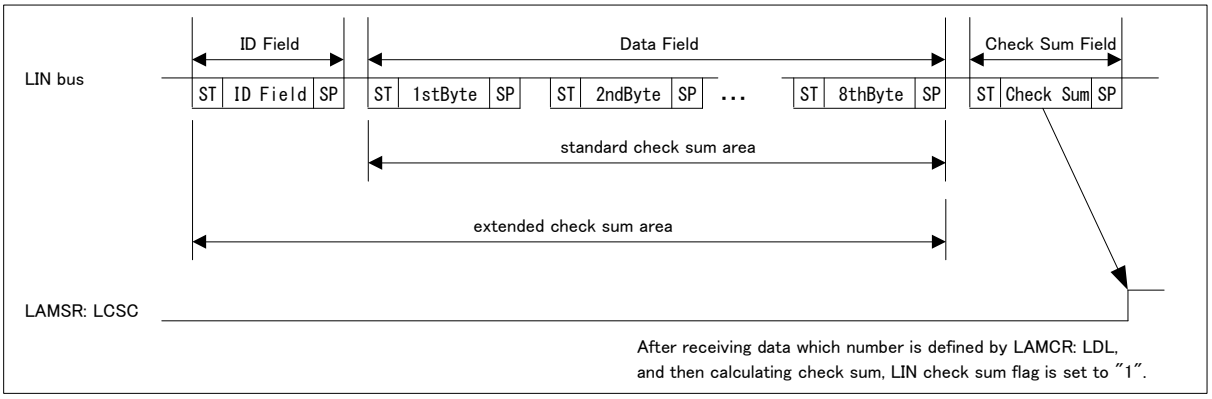


Occurrence of LIN Checksum Detection Completion Flag Interrupt and Flag Set Timing

When assist mode is set (LAMCR:LAMEN = 1), checksum detection is executed by the self-check performed on the side that transmits the checksum, and is also executed by the side that receives the checksum. When the data for the specified data length (LAMCR:LDL3 to LDL0) and checksum are received, the checksum calculation is complete so a flag is set (LAMSR:LCSC = 1). A status interrupt occurs if the interrupt is enabled (LAMIER:LCSCIE = 1).

When checksum reception is complete, the received checksum value is not stored in the RDR register and the SSR:RDRF is not set to "1". When a FIFO is used, the value is not stored in the reception FIFO.

Figure 2-27 Set Timing of LIN Checksum Detection Completion Flag (LAMSR:LCSC)



Notes:

- *The checksum calculation stops as soon as a framing error is detected in the last data of the specified data length (LAMCR:LDL3 to LDL0).*
- *If a framing error is detected in the checksum, the calculation result is not guaranteed.*

3. Serial Timer Operation

The serial timer provides timer functions.

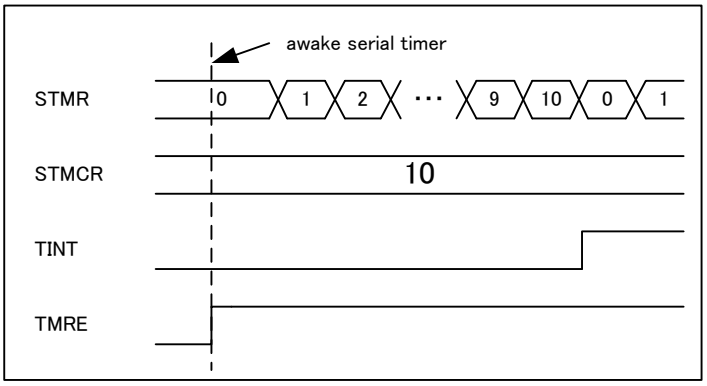
Serial Timer Operation

Activating Serial Timer

The serial timer can be activated by setting the serial timer enable bit (SACSR:TMRE) to "1", either by an external trigger or by the Sync Field.

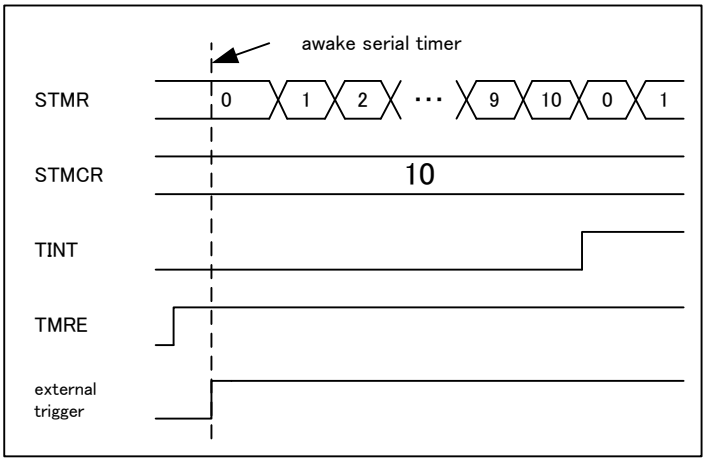
- Activation by the serial timer enable bit (SACSR:TMRE)
  When the external trigger enable bit (SACSR:TRGE) is set to "0", setting the serial timer enable bit (SACSR:TMRE) to "1" activates the serial timer and starts counting of the serial timer register (STMR) from 0.

Figure 3-1 Activation by Serial Timer Enable Bit (STMCR = 10, SACSR:TRGE = 0)



- Activation by an external trigger
  The serial timer is activated and counting of the serial timer register (STMR) starts from 0 when all of the following are satisfied: 1. The external trigger enable bit (SACSR:TRGE) is set to "1". 2. An external trigger edge specified by the trigger selection bit (TRG1, 0) is detected. 3. The serial timer enable bit (SACSR:TMRE) is set to "1".

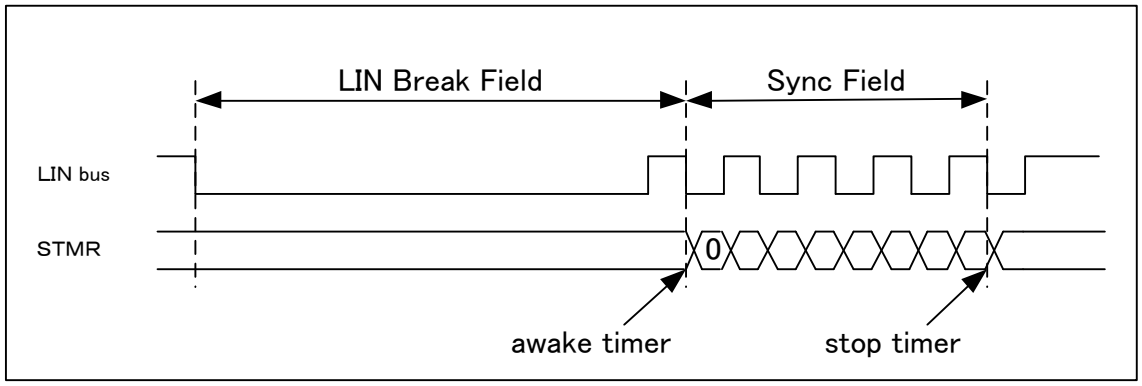
Figure 3-2 Activation by an External Trigger (STMCR = 10, SACSR:TRGE = 1, TRG1 = 0, TRG0 = 0)



■ Activation by Sync Field reception

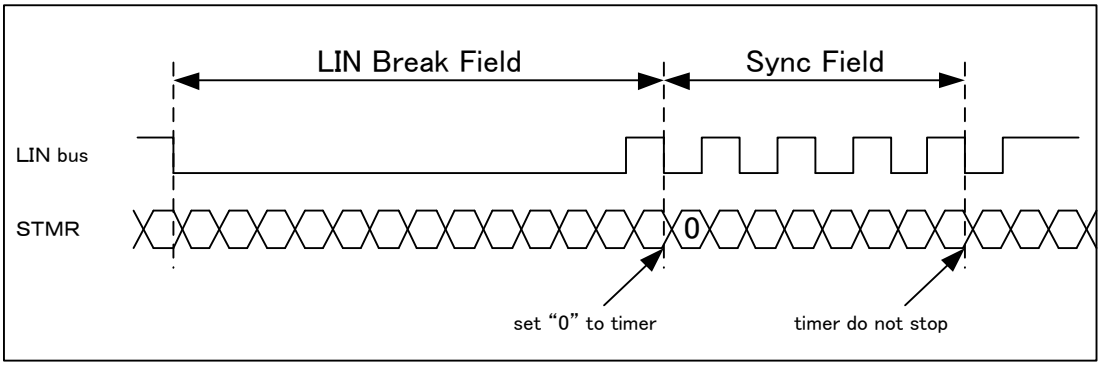
The serial timer is activated and counting of the serial timer register (STMR) starts from 0 when all of the following are satisfied: 1. The serial timer is stopped. 2. The auto baud rate adjustment bit (SACSR:AUTE) is set to "1". 3. The LIN interface (v2.1) detects the first falling edge of the Sync Field.

Figure 3-3 Reception of Sync Field While Serial Timer Is Stopped (SACSR:AUTE = 1, TMRE = 0)



Counting of the serial timer register (STMR) starts from 0 when all of the following are satisfied: 1. The serial timer is operating. 2. The auto baud rate adjustment bit (SACSR:AUTE) is set to "1". 3. The LIN interface (v2.1) detects the first falling edge of the Sync Field.

Figure 3-4 Reception of Sync Field While Serial Timer Is Operating (SACSR:AUTE = 1, TMRE = 1)



Note:

- *Detection of an external trigger edge specified by the trigger selection bit (SACSR:TRG1, 0) does not start the serial timer when both of the following are satisfied: 1. The external trigger enable bit (SACSR:TRGE) is set to "1". 2. The serial timer enable bit (SACSR:TMRE) is set to "0".*

Stopping the Serial Timer

The serial timer stops under the following conditions.

- The serial timer stops when both of the following are satisfied: 1. The auto baud rate adjustment bit (SACSR:AUTE) is set to "0". 2. The serial timer enable bit (SACSR:TMRE) is reset to "0". At this time, the value of the serial timer register (STMR) is retained.
- The serial timer stops when all of the following are satisfied: 1. The auto baud rate adjustment bit (SACSR:AUTE) is set to "1". 2. The serial timer enable bit (SACSR:TMRE) is set to "1". 3. The serial timer enable bit (SACSR:TMRE) is reset to "0" in any status other than that indicating that the Sync Field is being received. At this time, the value of the serial timer register (STMR) is retained.
- The serial timer stops and the value of the serial timer register (STMR) is retained if the LIN interface (v2.1) detects the 5th falling edge of the Sync Field, when both of the following are satisfied: 1. The auto baud rate adjustment bit (SACSR:AUTE) is set to "1". 2. The serial timer enable bit (SACSR:TMRE) is set to "0".

Note:

- *The serial timer does not stop and continues operating even if the LIN interface (v2.1) detects the 5th falling edge of the Sync Field, when both of the following are satisfied: 1. The auto baud rate adjustment bit (SACSR:AUTE) is set to "1". 2. The serial timer enable bit (SACSR:TMRE) is set to "1".*

Timer Operation

If the serial timer register (STMR) and the serial timer comparison register (STMCR) match, the timer interrupt flag (SACSR:TINT) is set to "1", and the serial timer register (STMR) is reset to "0".

Figure 3-5 Timer Operation (STMCR = 10)

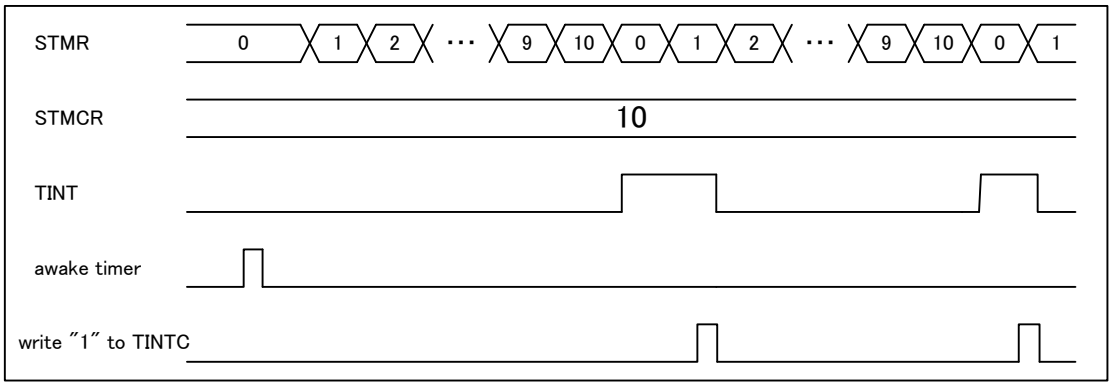


Figure 3-6 Serial Timer Initialization

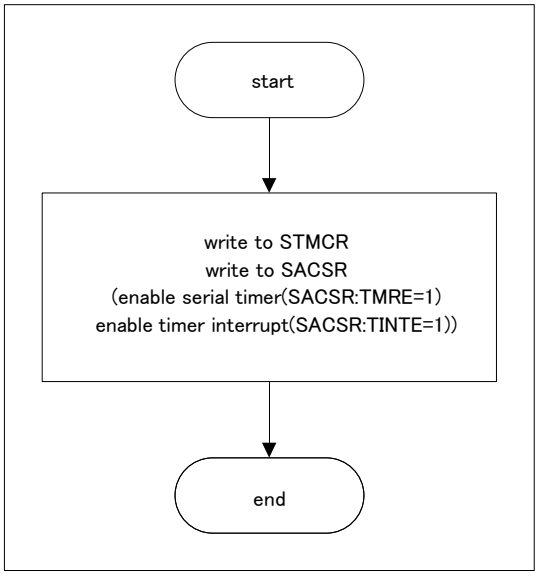
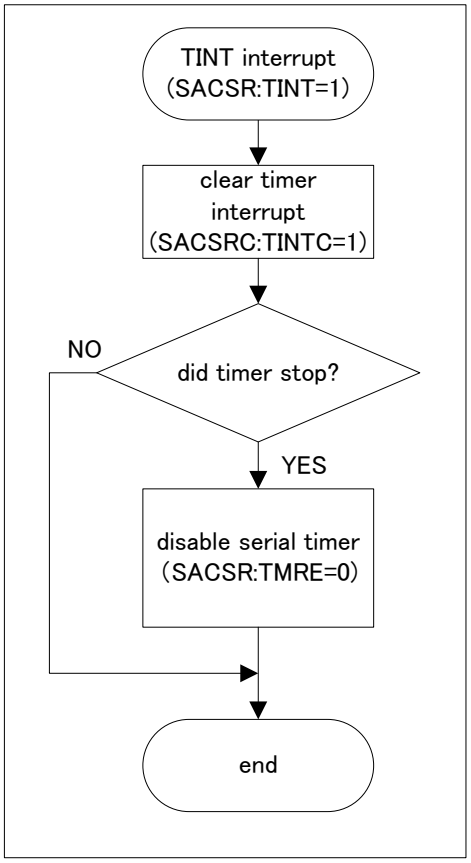


Figure 3-7 Serial Timer Interrupt



Notes:

- *The timer interrupt flag (SACSR:TINT) is fixed to "1" when the timer comparison register (STMCR) is set to 0x0000, the timer is operating, and the division value (SACSR:TDIV) of the timer operation clock is set to 0b0000.*
- *The serial timer register (STMR) is reset to "0" after receiving the Sync Field when the auto baud rate adjustment bit (SACSR:AUTE) is set to "1".*

4. Test Mode

This chapter describes the operation in test mode.

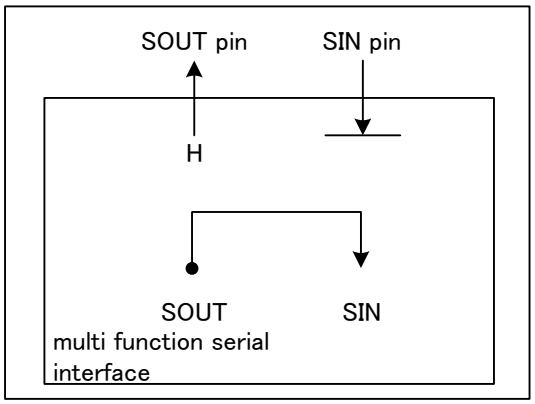
4.1. Manual Mode

Serial Test Mode

When serial test mode is enabled (SACSR:STST = 1), SOUT and SIN are connected together within the multi-function serial interface, so that the data transmitted from SOUT can be received through SIN directly.

When serial test mode is enabled (SACSR:STST = 1), the SOUT pin is fixed to "H", and the data input to the SIN pin is ignored.

Figure 4-1 Serial Test Mode



Notes:

- *The value of the serial test mode enable bit (SACSR:STST) can be changed only when transmission and reception are prohibited (SCR:TXE = RXE = 0).*
- *In manual mode (LAMCR:LAMEN = 0), do not set the pseudo error test mode.*

4.2. Assist Mode

Serial test mode

This mode is the same as Serial Test Mode in manual mode.

However, in LIN assist mode (LAMCR:LAMEN = 1), the serial test can be executed only on the master node (SCR:MS = 0). The serial test result is checked by the transmission/reception flag and the status flag. For the transmission/reception flag and status flag, see "Table 2-4 LIN Interface (v2.1) Interrupt Control Bits and Interrupt Factors (Assist Mode)."

Notes:

- *In LIN assist mode (LAMCR:LAMEN = 1), the serial test cannot be executed on the slave node (SCR:MS = 1).*
- *When the serial test is executed on the master node (SCR:MS = 0) in LIN assist mode (LAMCR:LAMEN = 1), transmission data (the Sync Field, ID Field, and response data, as well as the checksum) cannot be read by the reception data register (RDR).*

Pseudo Error Test Mode

In assist mode (LAMCR:LAMEN = 1), a LIN bus error, LIN Sync Data error, LIN ID parity error, LIN checksum error, and framing error can be generated as a pseudo error. More than one of these errors can be generated at the same time.

Any of the following self-diagnostics can be executed by implementing serial test mode together.

- Pseudo LIN bus error test mode
- Pseudo LIN ID parity error test mode
- Pseudo LIN checksum error test mode
- Pseudo framing error test mode

Activating Pseudo Error Test Mode

To activate pseudo error test mode, it is necessary to enable the pseudo error setting by writing values to the key code control bit (LAMERT:KEY1, KEY0) according to the following procedure.

- KEY1, KEY0 = 0b00 + write a pseudo error setting value
- KEY1, KEY0 = 0b01 + write the pseudo error setting value (same as the previous value)
- KEY1, KEY0 = 0b10 + write the pseudo error setting value (same as the previous value)
- KEY1, KEY0 = 0b11 + write the pseudo error setting value (same as the previous value)
- The pseudo error setting value is enabled when the value is written for the 4th time.

The value written to this register is invalid if you do not follow this setting procedure (if a value is written or read to/from another register during the writing process, if the value written is incorrect, or if a value is read from this register during the writing process).

To clear a pseudo error setting, follow the same procedure for making a pseudo error setting.

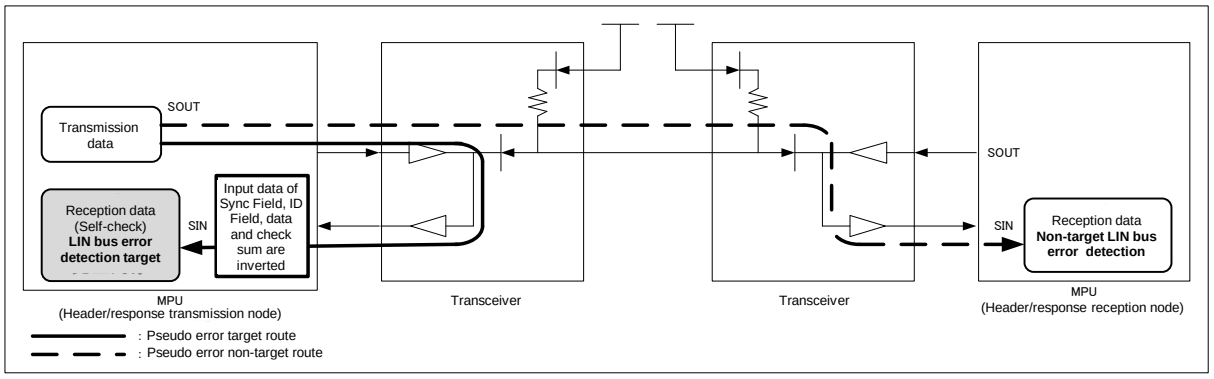
Notes:

- *LIN communication stops when any of the following pseudo errors occurs in assist mode.*
  - LIN bus error
  - LIN ID parity error
  - Framing error
- *In manual mode (LAMCR:LAMEN = 0), do not set pseudo error test mode.*

Overview of pseudo LIN bus error test mode

During a pseudo LIN bus error test, a node that transmits data receives the transmitted data by inverting it.

Figure 4-2 Overview of Pseudo LIN Bus Error Test Mode



To activate pseudo LIN bus error test mode, set the LIN bus error pseudo error setting bit (LAMERT:LBSERT = 1) according to the method used to activate pseudo error test mode.

Activating pseudo LIN bus error test mode causes the following operation to be performed.

- Master

  The Sync Field, ID Field, data, and checksum are transmitted.
  Once the LIN bus error pseudo error setting is set (LAMERT:LBSERT = 1), the reception data is inverted at the stop bit timing. Then, a LIN bus error occurs during the self-check and the flag bit (LAMESR:LBSER) is set to "1".

- Slave

  Data and the checksum are transmitted.
  Once the LIN bus error pseudo error setting is set (LAMERT:LBSERT = 1), the reception data is inverted at the stop bit timing. Then, a LIN bus error occurs during the self-check and the flag bit (LAMESR:LBSER) is set to "1".

Notes:

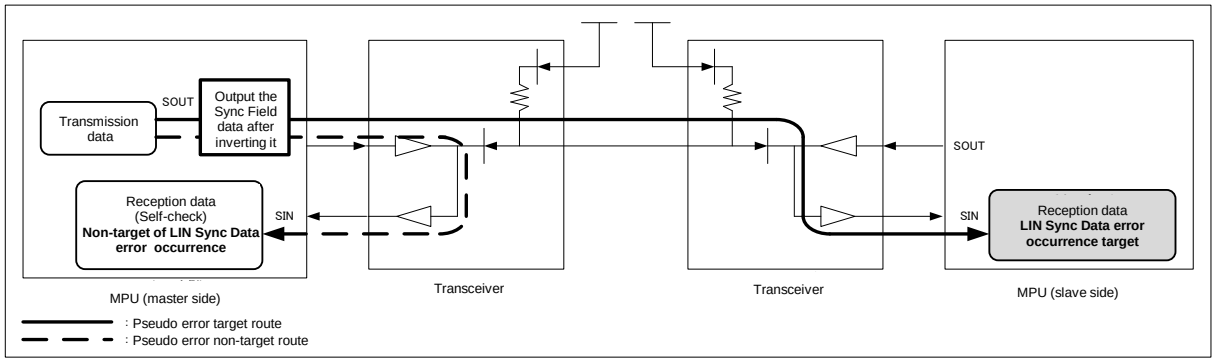
- *When a LIN bus error is detected (LAMESR:LBSER = 1), the transmission/reception processing of the header and response sections in assist mode stops.*
- *Enable this test mode only when transmitting a header or response.*

Overview of Pseudo LIN Sync Data Error Test Mode

In a pseudo LIN Sync Data error test, the Sync Field value (0x55) is transmitted after being inverted.

The master that transmits the Sync Field cannot detect a pseudo LIN Sync Data error.

Figure 4-3 Overview of Pseudo LIN Sync Data Error Test Mode



To activate pseudo LIN Sync Data error test mode, it is necessary to enable the LIN Sync Data error pseudo error setting bit (LAMERT:LSFERT = 1) according to the method used to activate pseudo error test mode.

When the pseudo LIN Sync Data error pseudo error setting (LAMERT:LSFERT = 1) is set in the master before the start bit of the Sync Field, the master outputs the value (0x55) after inverting it when transmitting the Sync Field.

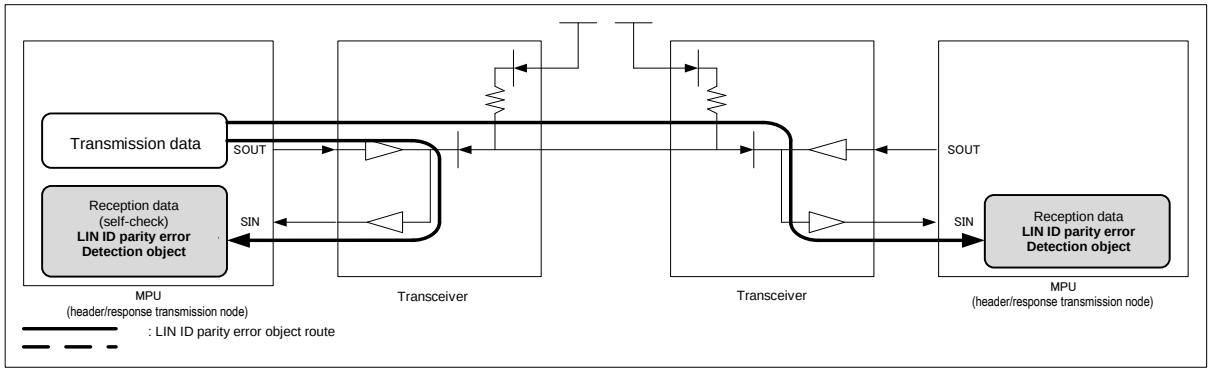
Notes:

- *A LIN Sync Data error is detected in the slave (SCR:MS = 1) in assist mode (LAMCR:LAMEN = 1).*
- *When a LIN Sync Data error is detected (LAMESR:LSFER = 1), the reception processing of the header and transmission/reception processing for the response in assist mode stops.*
- *Both master and slave modes can be set; however, a pseudo LIN Sync Data error can be generated only in master mode.*
- *Set (SCR:LBR = 1) before LIN communication starts.*
- *Set the LIN bus error pseudo error setting bit (LAMERT:LBSERT = 1) at the same time as this bit (LAMERT:LSFERT = 1) is set.*

Overview of Pseudo LIN ID Parity Error Test Mode

During the pseudo LIN ID parity error test, a parity bit is transmitted after being inverted.

Figure 4-4 Overview of Pseudo LIN ID Parity Error Test Mode



To activate pseudo LIN ID parity error test mode, it is necessary to enable the LIN ID parity error pseudo error setting bit (LAMERT:LPTERT = 1) according to the method used to activate the pseudo error test mode.

When the pseudo LIN ID parity error pseudo error setting (LAMERT:LPTERT = 1) is made on the master before the start bit of the ID Field, the master outputs the parity values (2 bits) to the ID Field after inverting them all when transmitting the ID Field.

When receiving the ID Field, a LIN ID parity error occurs and the flag bit (LAMESR:LPTER) is set to "1".

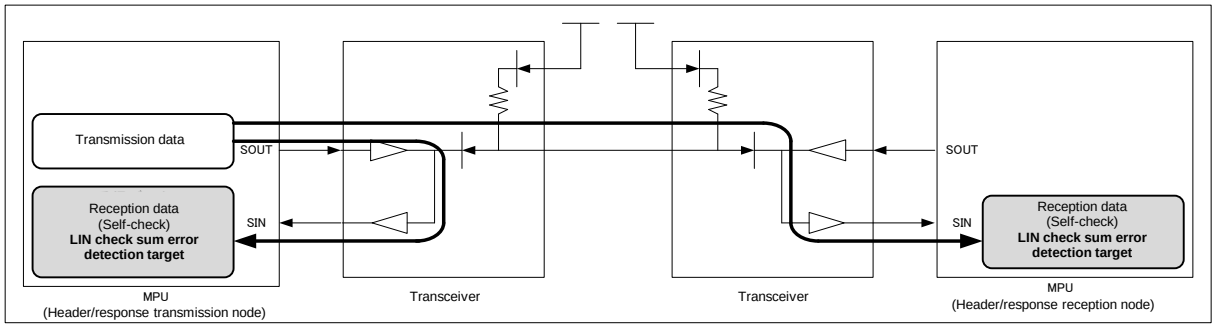
Notes:

- *The transmission/reception processing for a response in assist mode stops once a LIN ID parity error is detected (LAMESR:LPTER = 1).*
- *Both master and slave mode can be set; however, a pseudo LIN parity error can be generated only in master mode.*
- *Set (SCR:LBR = 1) before the LIN communication starts.*

Overview of Pseudo LIN Checksum Error Test Mode

During the pseudo LIN checksum error test, the checksum data is transmitted after being inverted.

Figure 4-5 Overview of Pseudo LIN Checksum Error Test Mode



To activate pseudo LIN checksum error test mode, it is necessary to enable the LIN checksum error pseudo error setting bit (LAMERT:LCSERT = 1) according to the method used to activate pseudo error test mode.

When transmitting the checksum, the checksum data is transmitted after being inverted. At this time, the flag bit is set to "1" (LAMESR:LCSER) because the LIN bus is monitored.

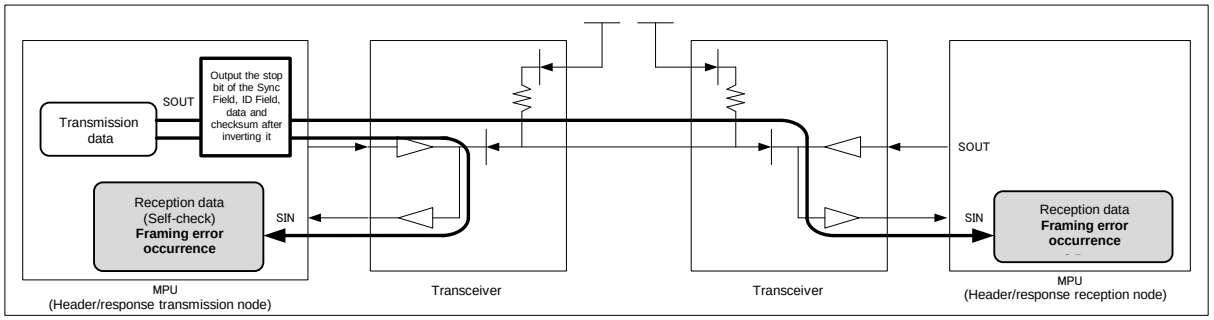
Notes:

- *In master mode, set (SCR:LBR = 1) before LIN communication starts.*
- *In slave mode, make this setting before transmitting a response.*

Overview of Pseudo Framing Error Test Mode

During the pseudo framing error test, the stop bit is transmitted after being inverted.

Figure 4-6 Overview of Pseudo Framing Error Test Mode



To activate pseudo framing error test mode, it is necessary to enable the framing error pseudo error setting bit (LAMERT:FRET = 1) according to the method used to activate pseudo error test mode.

Activating pseudo framing error test mode causes the following operation to be performed.

- Master

  When the framing error pseudo error setting is made (LAMERT:FRET = 1) before the stop bit of each Field, the stop bit value ("H" level) is output after being inverted when transmitting the Sync Field, ID Field, data, and checksum.

  During reception, a framing error occurs and the flag bit (SSR:FRE) is set to "1".

- Slave

  When the framing error pseudo error setting is made (LAMERT:FRET = 1) before the stop bit of each Field, the stop bit value ("H" level) is output after being inverted when transmitting the data and checksum.

  During reception, a framing error occurs and the flag bit (SSR:FRE) is set to "1".

Note:

- *When a framing error is detected (SSR:FRE = 1), transmission/reception processing of the header and response sections in assist mode stops.*

5. Dedicated Baud Rate Generator

The transmission and reception clock source for the LIN interface (v2.1) can be selected from the following.

- Dedicated baud rate generator (reload counter)
- Input of an external clock to the baud rate generator (reload counter)

LIN Interface (v2.1) Baud Rate

A baud rate can be selected from the following 2 types:

Baud Rate Obtained by Dividing the Internal Clock by the Dedicated Baud Rate Generator (Reload Counter)

There are 2 internal reload counters, each of which corresponds to the transmission or reception serial clock. A baud rate can be selected by setting a 15-bit reload value in the baud rate generator register 1 or 0 (BGR1, BGR0).

The reload counters divide the internal clock according to the set value.

Set the clock source by selecting an internal clock (BGR1:EXT = 0).

Note:

- *When auto baud rate adjustment is enabled (SACSR:AUTE = 1), set in the internal clock(BGR1:EXT=0),*

Baud Rate Obtained by Dividing an External Clock by the Dedicated Baud Rate Generator (Reload Counter)

An external clock is used as the clock source for the reload counter.

A baud rate can be selected by setting a 15-bit reload value in the baud rate generator register 1 or 0 (BGR1, BGR0).

The reload counters divide the external clock according to the set value.

Set a clock source by selecting the use of an external clock and the baud rate generator clock (BGR1:EXT = 1).

This mode is provided for dividing and using an oscillator with a special frequency.

Notes:

- *Make the setting of the external clock (BGR1:EXT = 1) under the condition whereby the reload counter is stopped (BGR1/0 = 0x0000).*
- *When the external clock setting is specified (BGR1:EXT = 1), the "H" width and "L" width of the external clock must be 2 bus clocks or more.*
- *When the external clock setting is specified (BGR1:EXT = 1), auto baud rate adjustment is disabled (SACSR:AUTE = 0).*

5.1. Setting Baud Rate

This section describes baud rate setting. This section also describes the result of calculating the serial clock frequency.

Calculation of Baud Rate

The 2 15-bit reload counters are set by baud rate generator registers 1, 0 (BGR1, BGR0).
The baud rate calculation formulas are shown below.

(1) Reload value:

```
V = Φ / b - 1

V : reload value  b : baud rate   Φ :  frequency of bus clocl or that of external clock
```

(2) Calculation example

```
When bus clock is 16MHz, using internal clock, baud rate is 19200 bps, the reload value can be
calculated as follows.
                reload value :
                    V = (16 x 1000000)/19200 - 1 = 832
                          then baud rate will be
                    b = (16 x 1000000)/(832 + 1) = 19208bps
```

(3) Baud rate error

The baud rate error is obtained by the following formula.

```
error rate(%) = (calculated value - target value)/target value * 100
  (ex. When bus clock is 20MHz, target baud rate is 153600 bps,
       the result will be as follows.
               reload value    = (20 x 1000000)/153600 - 1 = 129
          calculated baud rate =  20 x 1000000)/(129 + 1) = 153846 (bps)
               error rate (%) = (153846 - 153600)/153600 x 100 = 0.16 (%)
```

Notes:

- *Setting the reload value to "0" stops the reload counter.*
- *When the reload value is even, the "L" width of the serial clock is 1-bus-clock-cycle longer than the "H" width. When the reload value is odd, the "L" width and the "H" width of the serial clock are equal.*
- *Set the reload value to 3 or larger. However, data may not be received normally depending on the baud rate errors and the reload value setting.*
- *When considering the tolerable baud rate range, also consider the impact of any jitter in the clock input on the macro.*

Example of Reload Values and Baud Rate Settings for Individual Bus Clock Frequencies

The following table shows example reload values and baud rate settings.

Table 5-1 Example Reload Values and Baud Rate Settings

Baud Rate	8 MHz		10 MHz		16 MHz		20 MHz		24 MHz		32 MHz	
(bps)	Value	ERR	Value	ERR	Value	ERR	Value	ERR	Value	ERR	Value	ERR
8 M	-	-	-	-	-	-	-	-	-	-	3	0
6 M	-	-	-	-	-	-	-	-	3	0	-	-
5 M	-	-	-	-	-	-	3	0	-	-	-	-
4 M	-	-	-	-	3	0	4	0	5	0	7	0
2.5 M	-	-	3	0	-	-	7	0	-	-	-	-
2 M	3	0	4	0	7	0	9	0	11	0	15	0
1 M	7	0	9	0	15	0	19	0	23	0	31	0
500000	15	0	19	0	31	0	39	0	47	0	63	0
460800	-	-	-	-	-	-	-	-	51	0.16	-	-
250000	31	0	39	0	63	0	79	0	95	0	127	0
230400	-	-	-	-	-	-	86	-0.22	103	0.16	138	-0.08
153600	51	0.16	64	0.16	103	0.16	129	0.16	155	0.16	207	0.16
125000	63	0	79	0	127	0	159	0	191	0	255	0
115200	-	-	86	-0.22	138	-0.08	173	-0.22	207	0.16	277	-0.08
76800	103	0.16	129	0.16	207	0.16	259	0.16	312	-0.16	416	-0.08
57600	138	-0.08	173	-0.22	277	-0.08	346	0.06	416	-0.08	555	-0.08
38400	207	0.16	259	0.16	416	-0.08	520	-0.03	624	0	832	0.04
28800	277	-0.08	346	0.06	554	-0.01	693	0.06	832	0.04	1110	0.01
19200	416	-0.08	520	-0.03	832	0.04	1041	-0.03	1249	0	1666	-0.02
10417	767	<0.01	959	<0.01	1535	<0.01	1919	<0.01	2303	<0.01	3071	<0.01
9600	832	0.04	1041	<0.01	1666	-0.02	2082	0.01	2499	0	3332	0.01
7200	1110	<0.01	1388	<0.01	2221	<0.01	2777	<0.01	3332	<0.01	4443	0.01
4800	1666	-0.02	2082	0.01	3332	<0.01	4166	<0.01	4999	0	6666	<0.01
2400	3332	<0.01	4166	<0.01	6666	<0.01	8332	<0.01	9999	0	13332	<-0.01
1200	6666	<0.01	8332	<0.01	13332	<0.01	16666	<0.01	19999	0	26666	<0.01
600	13332	<0.01	16666	<0.01	26666	<0.01	-	-	-	-	-	-
300	26666	<0.01	-	-	-	-	-	-	-	-	-	-

- Value: Setting of the BGR1/0 register
- ERR: Baud rate error (%)

Table 5-2 Example Reload Values and Baud Rate Settings (Continued from Previous Page)

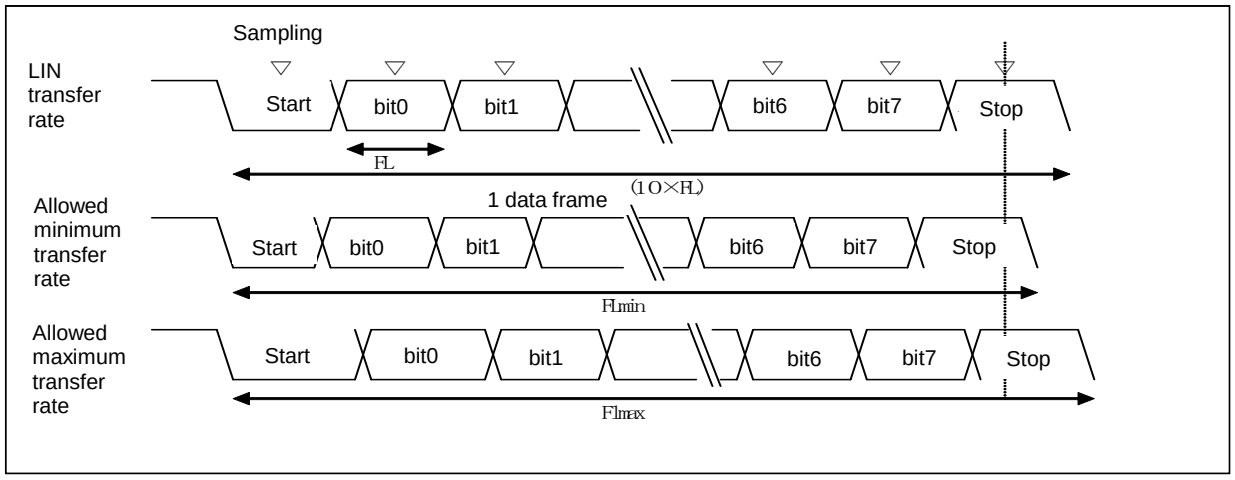
Baud Rate (bps)	40 MHz		48 MHz		72 MHz		80 MHz	
	Value	ERR	Value	ERR	Value	ERR	Value	ERR
8 M	4	0	5	0	8	0	9	0
6 M	-	-	7	0	11	0	-	-
5 M	7	0	-	-	-	-	15	0
4 M	9	0	11	0	17	0	19	0
2.5 M	15	0	-	-	-	-	31	0
2 M	19	0	23	0	35	0	39	0
1 M	39	0	47	0	71	0	79	0
500000	79	0	95	0	143	0	159	0
460800	86	-0.22	103	0.16	155	0.16	173	-0.22
250000	159	0	191	0	287	0	319	0
230400	173	-0.22	207	0.16	312	-0.16	346	0.06
153600	259	0.16	312	-0.16	468	-0.05	520	-0.03
125000	319	0	383	0	575	0	639	0
115200	346	0.06	416	-0.08	624	0	693	0.06
76800	520	-0.03	624	0	937	-0.05	1041	-0.03
57600	693	0.06	832	0.04	1249	0	1388	<0.01
38400	1041	-0.03	1249	0	1874	0	2082	0.01
28800	1388	<0.01	1666	-0.02	2499	0	2777	<0.01
19200	2082	0.01	2499	0	3749	0	4166	-0.01
10417	3839	<0.01	4607	<0.01	6911	<0.01	7679	0
9600	4166	<0.01	4999	0	7499	0	8332	0
7200	5555	<0.01	6666	<0.01	9999	0	11110	0
4800	8332	<0.01	9999	0	14999	0	16666	0
2400	16666	<0.01	19999	0	29999	0	-	-
1200	-	-	-	-	-	-	-	-
600	-	-	-	-	-	-	-	-
300	-	-	-	-	-	-	-	-

Reception Baud Rate Tolerance

Described here is the number of errors in the receiver baud rate that can be tolerated during reception.

Use the following formula to ensure that the baud rate errors during data reception fall within the tolerable range.

Figure 5-1 Reception Baud Rate Tolerance



After the detection of a start bit, the sampling timing for the reception data is determined by the counters set in registers BGR1/0 as shown in the figure. The data can be received normally if all the data up to the last data (stop bit) is included in this sampling timing.

Theoretically, this derives the following for 10-bit data reception.

When the sampling timing margin is 1 bus clock cycle (ϕ), the tolerable minimum transfer rate (FLmin) will be as follows:

FLmin = (10 bits * (V + 1) - (V + 1) / 2 + 2) / ϕ = (19 V + 23) / 2 ϕ (s) V: Reload value, ϕ: Bus clock

Therefore, the maximum transmitted baud rate (BGmax) that can be received will be as follows:

BGmax = 10 / FLmin = 20 ϕ / (19 V + 23) (bps) V: Reload value, ϕ: Bus clock

When receiving data at the tolerable maximum transfer rate (FLmax), sampling is done at the first point of the received 10th-bit.

Therefore, the tolerable maximum transfer rate (FLmax) will be as follows:

9 / 10 * FImax = (10 bits * (V + 1) - (V + 1) / 2) / ϕ V: Reload value, ϕ: Bus clock

FImax = (19 / 18 * 10 * (V + 1)) / ϕ

When the sampling timing margin (ϕ) is 2 clock cycles, the tolerable maximum transfer rate (FLmax) will be as follows:

9 / 10 * FImax = (10 bits * (V + 1) - (V + 1) / 2 - 2) / ϕ V: Reload value, ϕ: Bus clock

FImax = (19 / 18 * 10 * (V + 1) - 40 / 18) / ϕ = (190 V + 150) / 20 ϕ (s) V: Reload value, ϕ: Bus clock

Therefore, the minimum transmitted baud rate (BGmin) that can be received will be as follows:

BGmin = 10 / FLmax = 18 ϕ / (19 V + 15) (bps) V: Reload value, ϕ: Bus clock

The baud rate tolerable errors for the LIN interface (v2.1) and the transmission destination will be as follows, when calculated using the above formulas for the minimum and maximum baud rate values.

Reload Value (V)	Maximum Baud Rate Error Tolerable	Minimum Baud Rate Error Tolerable
3	0%	0
10	+3.28%	-3.41%
50	+4.83%	-4.87%
100	+5.04%	-5.07%
200	+5.15%	-5.16%
32767	+5.26%	-5.26%

Note:

- *The receiving accuracy depends on the number of bits in a frame, the bus clock, and the reload value. A higher accuracy is achieved through the use of a higher bus clock and a higher division ratio.*

External Clock

Writing "1" to the EXT bit in the baud rate generator register (BGR1) causes the baud rate generator to divide the external clock.

Note:

- *The external clock synchronizes with the internal clock of the LIN interface (v2.1). Therefore, an external clock that cannot be synchronized causes the operation to become unstable.*

Reload Counter Function

The transmission reload counter and reception reload counter both function as dedicated baud rate generators. They each consist of 15-bit registers for the reload values. They generate a transmission clock and a reception clock from either an external clock or the internal clock.

Start of Counting

When a reload value is written to the baud rate generator registers (BGR1 and BGR0), the reload counters start counting.

Restarting

A reload counter restarts under the following conditions:

For both the Transmission and Reception Reload Counters

- Programmable reset (SCR:UPCL bit)

Reception Reload Counter

- Detection of a falling edge of the start bit in asynchronous mode

6. Block Transfer

This section describes the operation of block transfer.

Setting Block Transfer

Block transfer can be performed for reception by setting RXBLKEN in the extended control register (ECR) to 1 and for transmission by setting TXBLKEN to 1.

Block transfer can be performed in that mode in which it is operated with the READ/WRITE access standard (RW access mode).

The block size can be set with the transmission block size register and FBYTE register (for the reception side).

The threshold at which to output a block transfer request can be set with the FBYTE register for reception and with the FTICR register for transmission. When the amount of data in the reception FIFO exceeds the setting made for the FBYTE register, a reception block transfer request is output. When the amount of free space in the transmission FIFO exceeds the setting made for the FTICR register, a transmission block transfer request is output.

6.1. RW Access Mode

This section describes block transfer in RW access mode.

Reception Block Transfer

Reception block transfer is performed as described below.

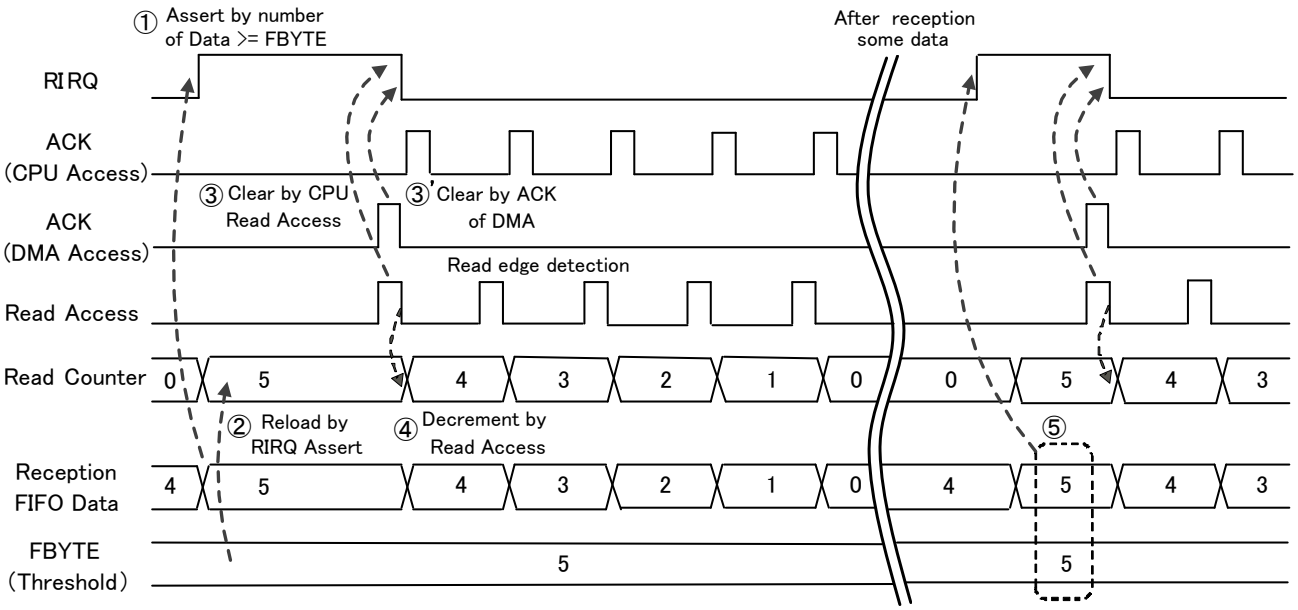
If the amount of data in the reception FIFO is over the threshold (FBYTE), a reception block transfer request (RIRQ) is asserted. At this time, the value of the reception block size is reloaded from the FBYTE register to the Read counter. When the reception FIFO is not used, the value of the reception block size is "1".

After the reception block transfer request, if reading a reception data occurs, the reception block transfer request (RIRQ) is cleared. With reading a reception data, the Read counter is decremented.

When the Read counter becomes "0", if the amount of data in the reception FIFO is over the threshold (FBYTE), the reception block transfer request (RIRQ) is asserted again. With the assertion of the reception block transfer request (RIRQ), the Read counter is reloaded.

During application operation, if the next block transfer request occurs before the completion of the previous block transfer, this state might be reported to the system by a block transfer error. This state does not occur if you configure FBYTE=1. Completion of block transfer by DMA depends on occupation of bus transaction by application. You need to fully evaluate the application integrally, design the system, and choose the optimal number of block transfers.

Figure 6-1 Reception Block Transfer in RW Access Mode



Transmission Block Transfer

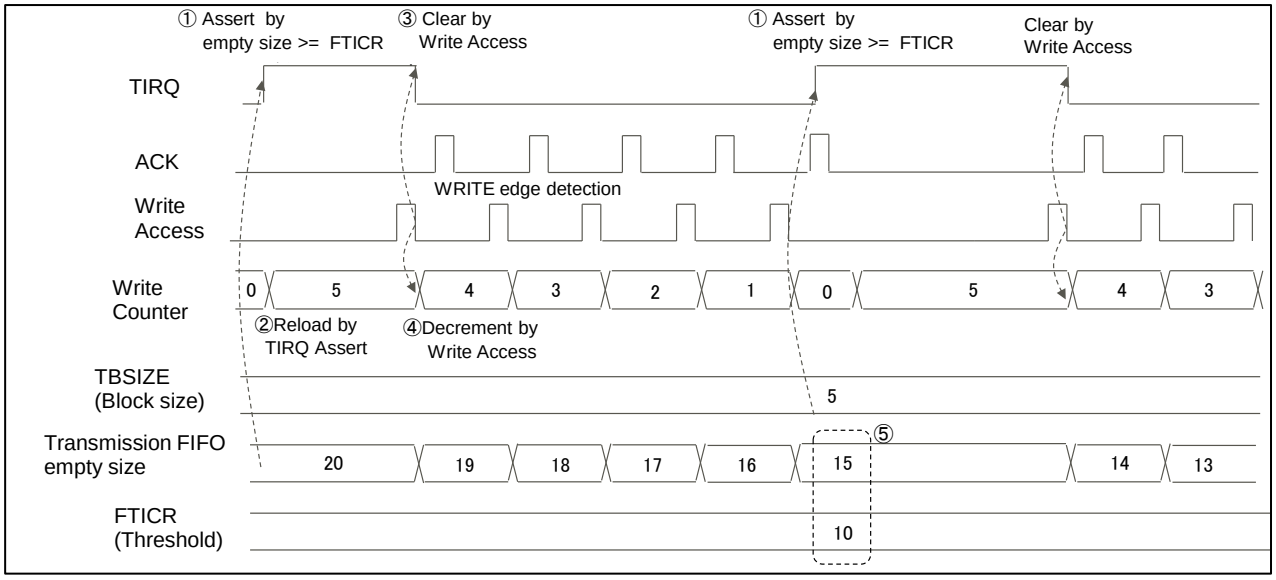
Transmission block transfer is performed as described below.

If the amount of free space in the transmission FIFO exceeds the threshold (FTICR), a transmission block transfer request (TIRQ) is asserted. At this time, the value of the transmission block size is reloaded from the TBSIZE register to the Write counter. When the transmission FIFO is not used, the value of the transmission block size is "1".

After the transmission block transfer request, if writing a transmission data occurs, the transmission block transfer request (TIRQ) is cleared. By writing a transmission data, the Write counter is decremented.

When the Write counter becomes "0", if the free space in the transmission FIFO exceeds the threshold (FTICR), the transmission block transfer request (TIRQ) is asserted again. With the assertion of the transmission block transfer request (TIRQ), the Write counter is reloaded.

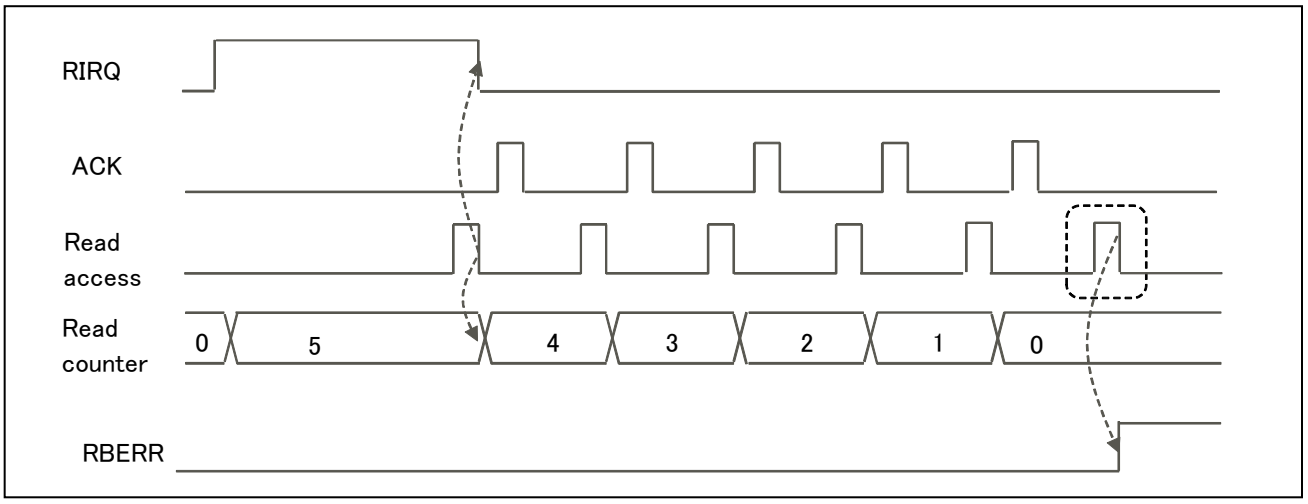
Figure 6-2 Transmission Block Transfer in RW Access Mode



Reception Block Transfer Error

If, while the Read counter is "0", a read access occurs before the next reception block transfer request (RIRQ) is asserted, a reception block transfer error occurs, and the ESR:RBERR bit is asserted.

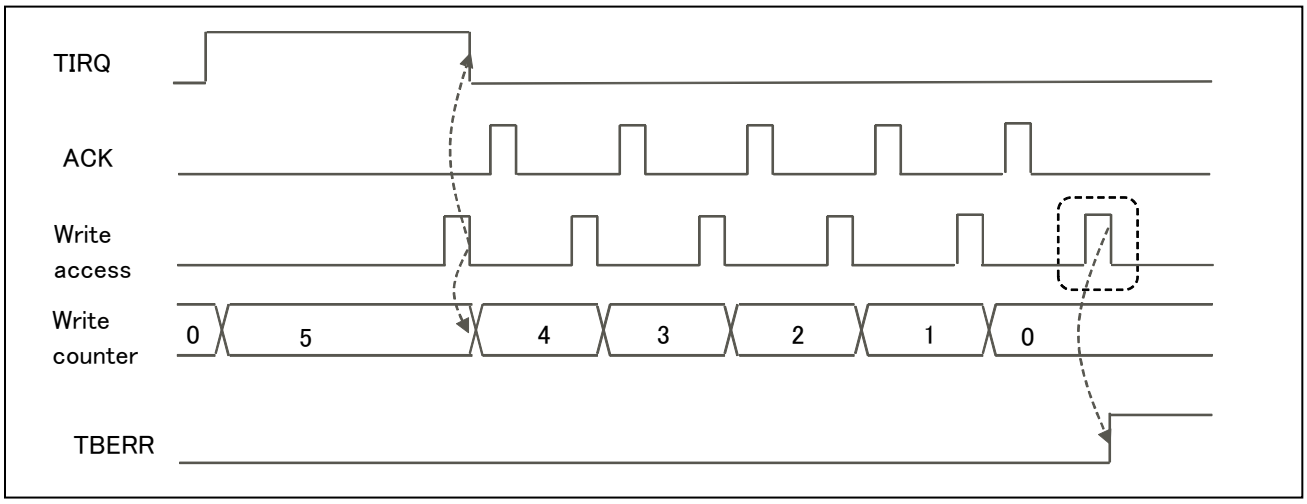
Figure 6-3 Reception Block Transfer Error in RW Access Mode



Transmission Block Transfer Error

If, while the Write counter is "0", a write access occurs before the next transmission block transfer request (TIRQ) is asserted, a transmission block transfer error occurs, and the ESR:TBERR bit is asserted.

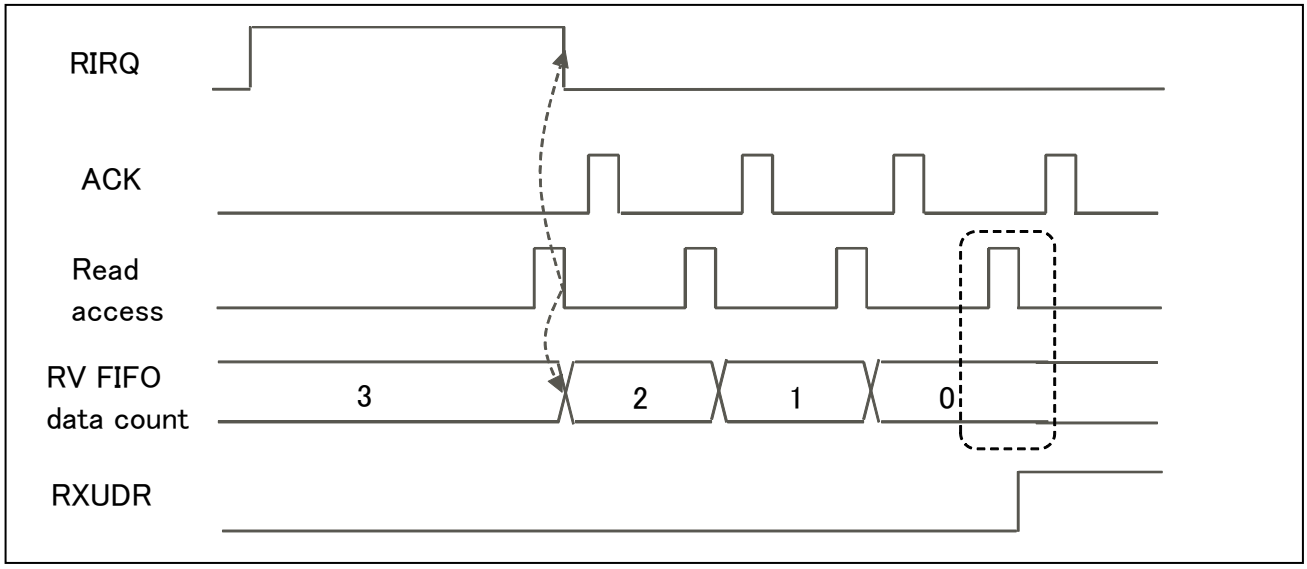
Figure 6-4 Transmission Block Transfer Error in RW Access Mode



Reception FIFO Underrun

If, while the amount of data in the reception FIFO is "0", reception data is read from the reception FIFO, a reception FIFO underrun occurs, and the ESR:RXUDR bit is asserted.

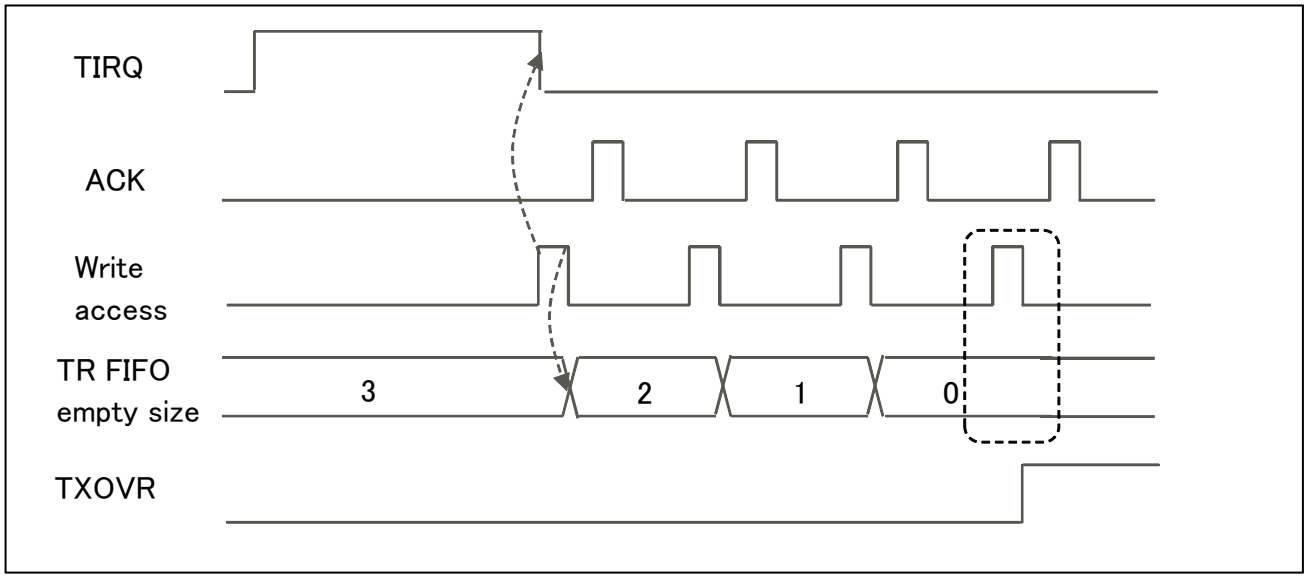
Figure 6-5 Reception FIFO Underrun in RW Access Mode



Transmission FIFO Overrun

If, while the amount of free space in the transmission FIFO is "0", transmission data is written to the transmission FIFO, a transmission FIFO overrun occurs, and the ESR:TXOVR bit is asserted.

Figure 6-6 Transmission FIFO Overrun in RW Access Mode



7. LIN Interface (V2.1) Operation

The LIN interface (v2.1) operates using master/slave bidirectional LIN communication.

7.1. Manual Mode

This section describes the operations in manual mode.

Master Operation

Selecting Master Operation

To enable operation as the master, set the SCR:MS bit to "0".

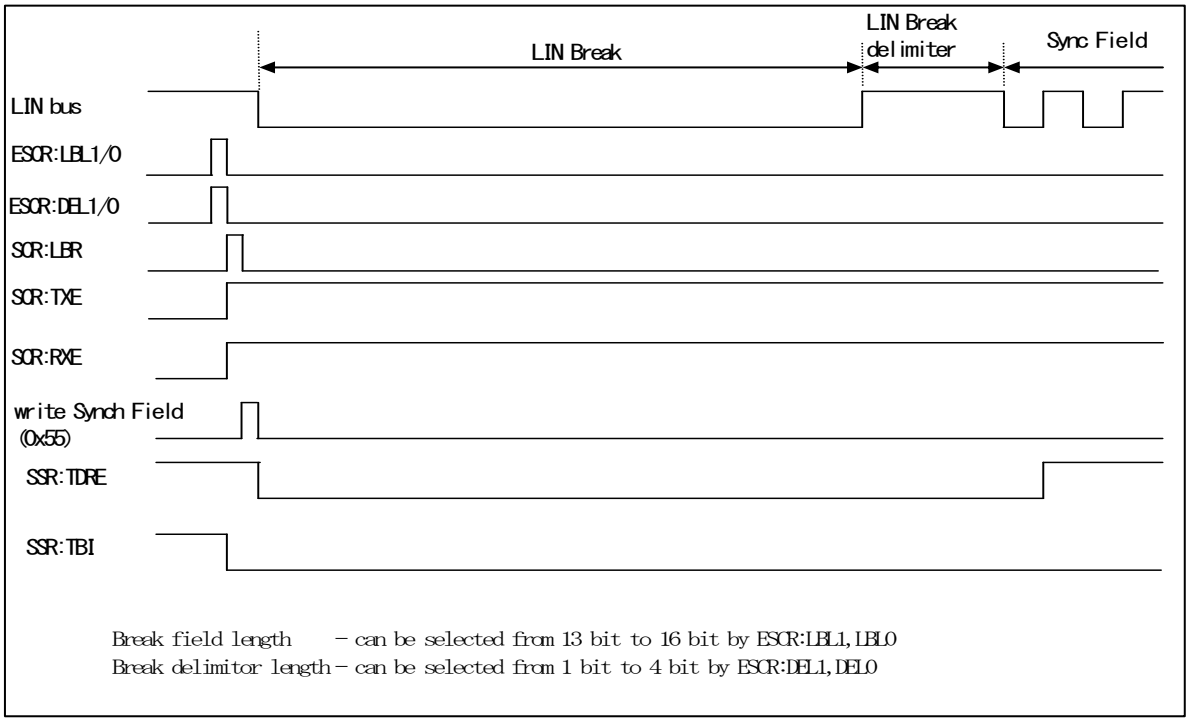
LIN Break Field Transmission to Sync Field Transmission

- The LIN Break Field length (ESCR:LBL1, LBL0) and LIN Break Field delimiter length (ESCR:DEL1, DEL0) can be selected.
- When transmission is enabled (SCR:TXE = 1) and the SCR:LBR bit (LIN Break Field setting bit) is set to "1", LIN Break Field is transmitted.
- Sync Field will be transmitted by writing 0x55 to the transmission data register (TDR).

Notes:

- *After setting the SCR:LBR bit (LIN Break Field setting bit) to "1", set the transmission data register (TDR) to 0x55.*
- *Even when the SCR:RXE bit (reception operation enable bit) is set to "1", the LIN Break Field part does not perform reception.*

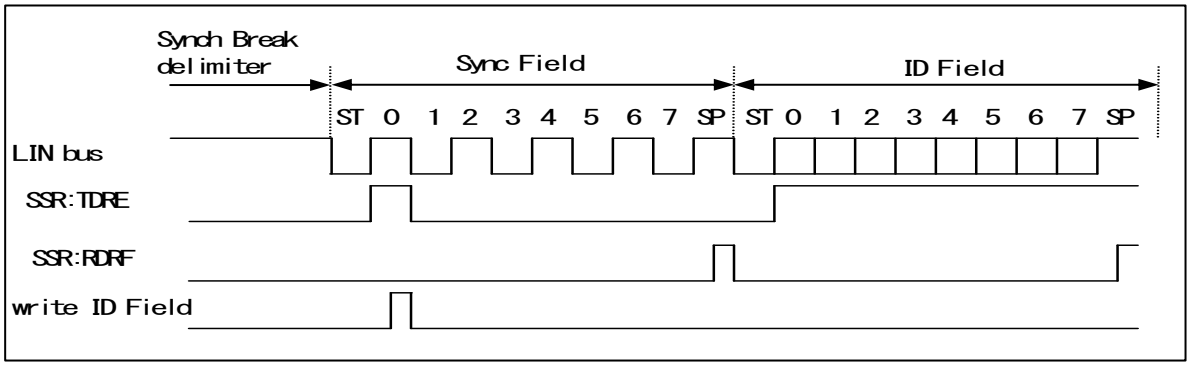
Figure 7-1 LIN Break Field to Sync Field Transmission



Sync Field Transmission to ID Field Transmission

- When the 1st bit of Sync Field (0x55) is transmitted, the SSR:TDRE (transmission data empty) bit is set to "1".
  At this time, a transmission interrupt occurs if transmission interrupt is enabled (SCR:TIE = 1).
- When a transmission interrupt occurs, the ID Field can be written in the transmission data register (TDR).
- When a reception interrupt occurs, the transmission data and reception data are compared to verify that there is no error.
- The data length of the ID Field is 8 bits, and the output is LSB first.

Figure 7-2 Sync Field to ID Field Transmission



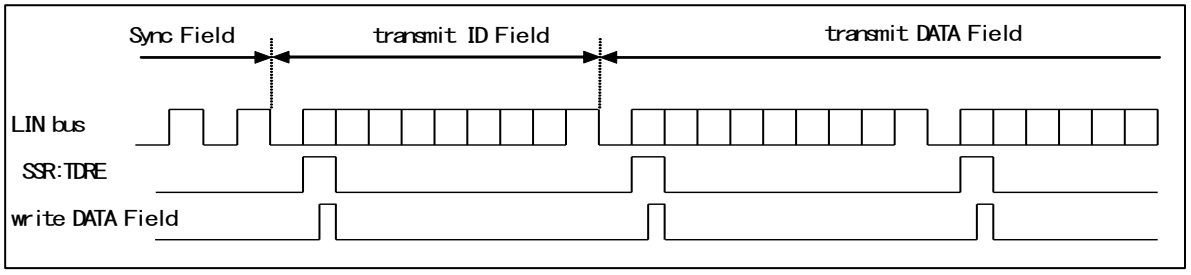
ID Field Transmission to DATA Field Transmission and Reception

This step selects whether the DATA Field is to be transmitted to or received by a slave device.

(When the DATA Field is transmitted)

When the 1st bit of the ID Field is transmitted, it is set as SSR:TDRE = 1. At this time, the DATA Field can be written.

Figure 7-3 ID Field Transmission to DATA Field Transmission

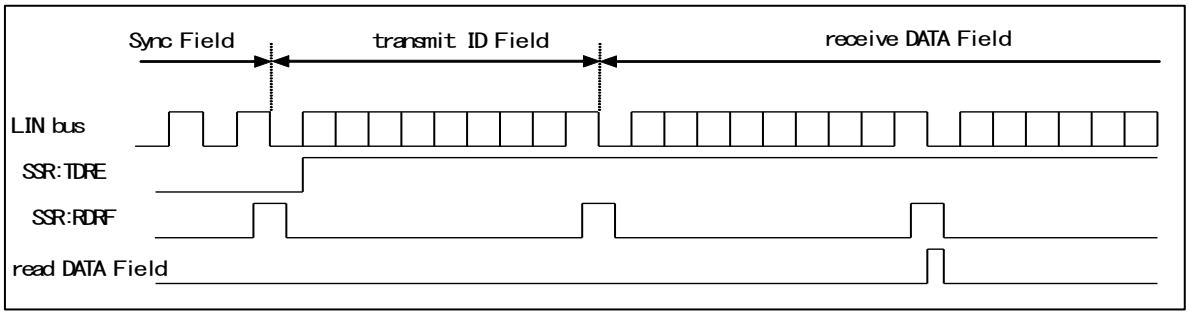


(When the DATA Field is received)

- When the 1st bit of the ID Field is transmitted, it is set as SSR:TDRE = 1. However, do not write any transmission data.
  In addition, disable transmission interrupt (SCR:TIE = 0).
- When the DATA Field is received, SSR:RDRF is set to 1. At this time, a reception interrupt occurs if the reception interrupt is enabled (SCR:RIE = 1).

- A start bit is detected when both of the following are satisfied: 1. The noise-filtered result (the result of the rule of majority on the 3-time bus clock sampling of the serial data input) shows falling. 2. The filtered data shows "L" at the sampling point.

Figure 7-4 ID Field Transmission to DATA Field Reception



Notes:

- *There is a built-in noise filter (the rule of majority being applied to 3-time bus clock sampling of the serial data input). However, we suggest that you design the board in such a way that noise does not pass through the filter. Alternatively, we suggest that you implement a communication method whereby noise passing through the filter does not cause a problem (by, for example, appending a checksum to the data at the end of transmission for retransmission in case of an error).*
- *During data reception, the following occurs if a falling edge of serial data is detected at the same time as, or 1 or 2 bus clocks earlier than, the sampling point: The edge becomes invalid and the frame that follows cannot be received normally. When frames are successively output, it is recommended that intervals be provided between the frames.*

Master Operation Timing Chart (When a FIFO is Not Used)

Figure 7-5 LIN Bus Timing (When a DATA Field Is Transmitted: When a FIFO Is Not Used)

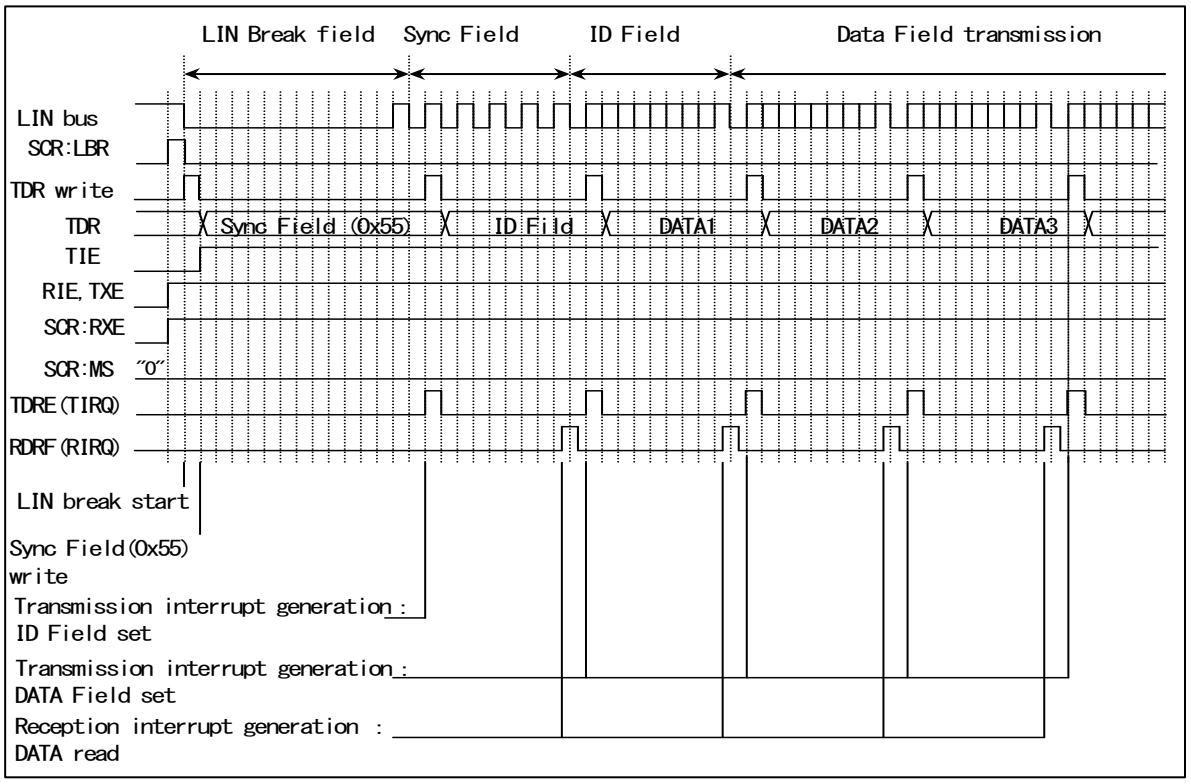
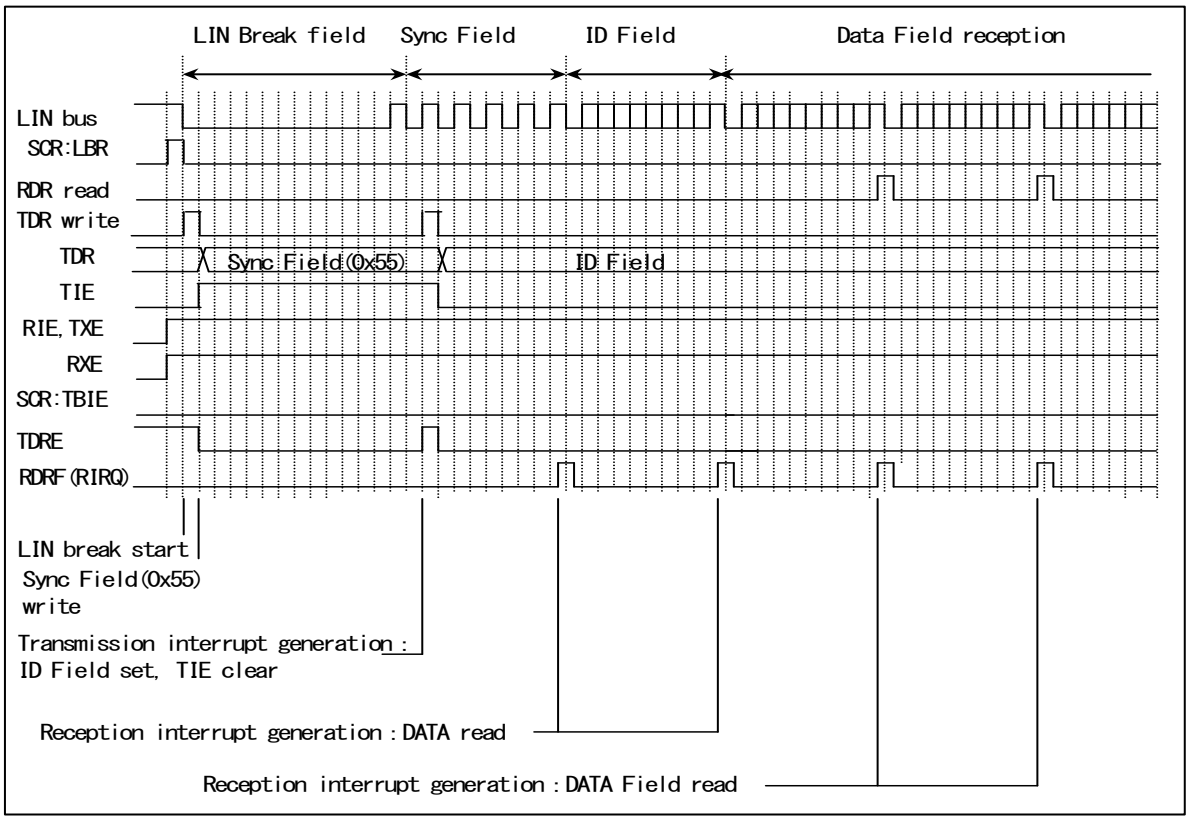


Figure 7-6 LIN Bus Timing (When a DATA Field Is Received: When a FIFO Is Not Used)



Master Operation Timing Chart (When a FIFO is Used)

Figure 7-7 LIN Bus Timing (When a DATA Field Is Transmitted: When a FIFO Is Used)

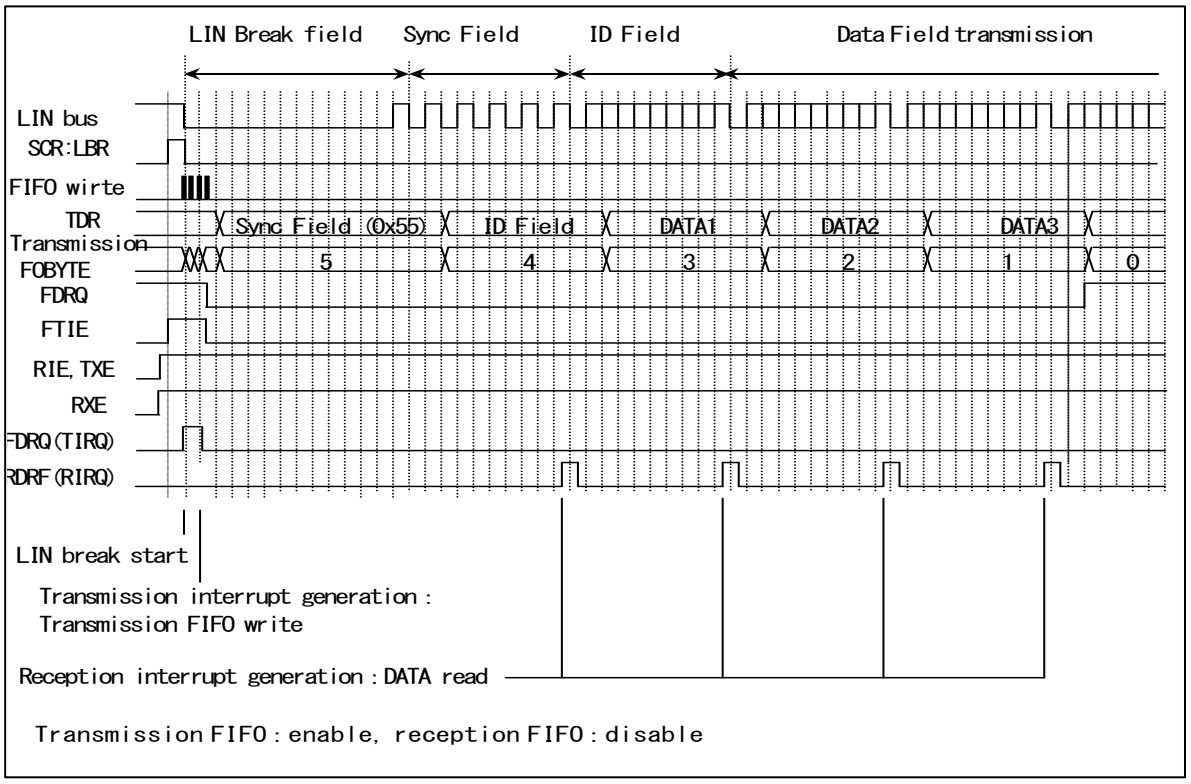
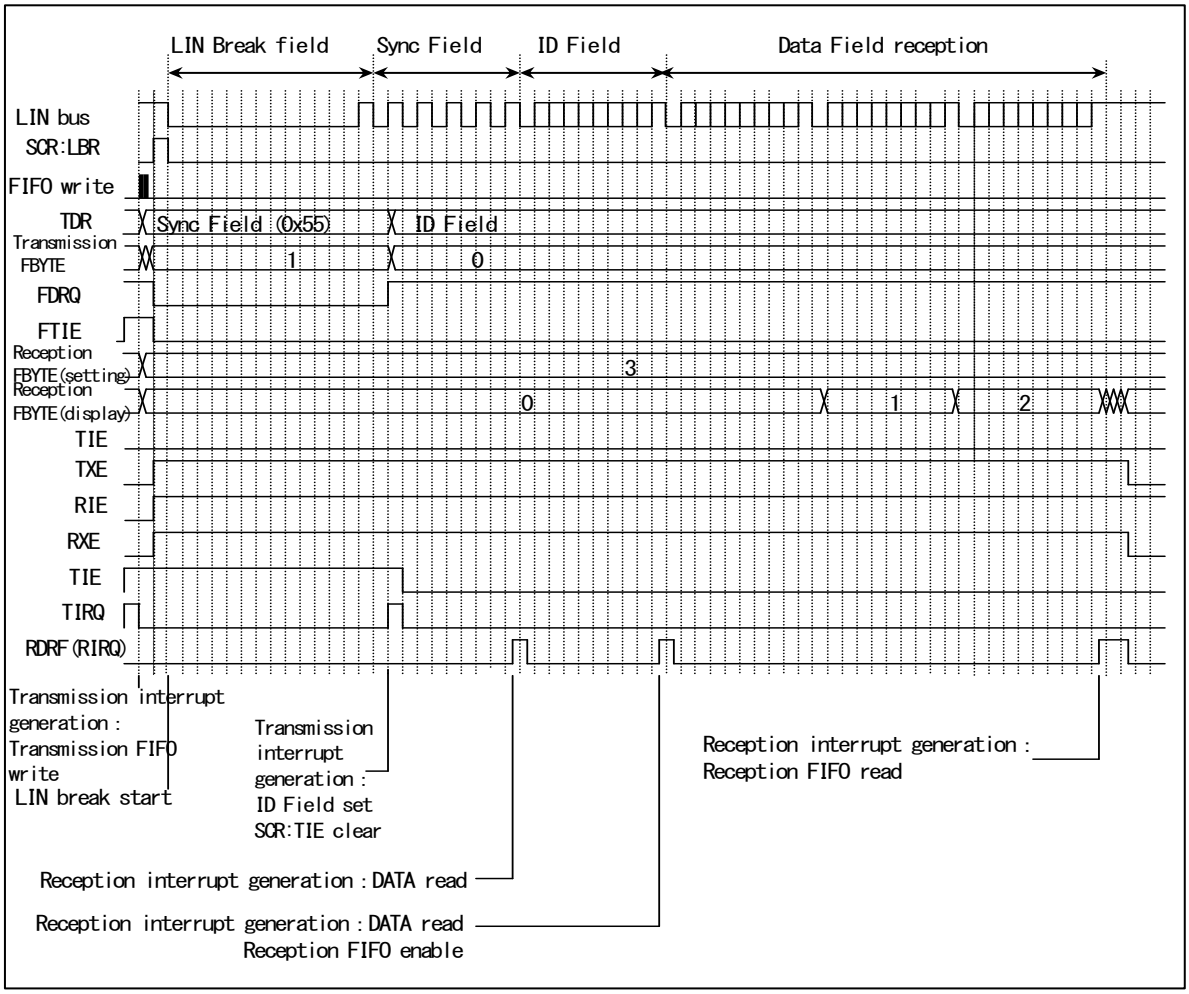


Figure 7-8 LIN Bus Timing (When a DATA Field Is Received: When a FIFO Is Used)



Slave Operation

Selecting Slave Operation

To enable operation as a slave, set SCR:MS bit to "1".

LIN Break Field Reception to Sync Field Reception

There are 2 methods to verify whether auto baud rate adjustment was applied between LIN Break Field reception and Sync Field reception, as follows:

- Method for comparing BGR and STMR
- Method for confirming the SACSR:BST bit

The process using each method is as follows:

1. Method for comparing BGR and STMR

   (1) Enable auto baud rate adjustment (SACSR:AUTE = 1)

   (2) When the LIN Break Field is entered, the LIN Break Field is detected (SSR:LBD = 1) at the 11th bit.
   At this time, a status interrupt occurs if the ESCR:LBIE bit is set to "1". After the LIN Break Field is detected (SSR:LBD = 1), the serial timer is disabled (SACSR:TMRE = 0).

   (3) When the LIN Interface (v2.1) detects the 1st falling edge of the Sync Field, it initializes the serial timer register (STMR) to 0.

   (4) When the 5th falling edge of the Sync Field is detected, the Sync Field detection flag (SACSR:SFD) is set to "1".

   (5) When the 5th falling edge of the Sync Field is detected, the Sync Field detection flag (SACSR:SFD) is set to "1". Then, the following is verified to check whether auto baud rate adjustment was performed.

   - When auto baud rate adjustment was performed, the data read from the serial timer register (STMR) and baud rate generator register (BGR) are the same upon the detection of the Sync Field (SACSR:SFD = 1).
   - When auto baud rate adjustment was not performed, the data read from the serial timer register (STMR) and baud rate generator register (BGR) differs upon the detection of the Sync Field (SACSR:SFD = 1).

   Note:
   - *When auto baud rate adjustment is enabled (SACSR:AUTE = 1), set in the internal clock(BGR1:EXT=0),*

2. Method of using the SACSR:BST bit

   (1) Enable auto baud rate adjustment (SACSR:AUTE = 1)

   (2) When the LIN Break Field is entered, the LIN Break Field will be detected (SSR:LBD = 1) at the 11th bit.
   At this time, a status interrupt occurs if ESCR:LBIE bit is set to "1".

   (3) When LIN Interface (v2.1) detects the 1st falling edge of the Sync Field, it initializes serial timer register (STMR) to 0.

   (4) When the 5th falling edge of the Sync Field is detected, the Sync Field detection flag (SACSR:SFD) is set to "1".

(5) When the falling edge of the 5th Sync Field is detected, the following operation is performed depending on the value of the serial timer register (STMR).

- When the value of serial timer register (STMR) is between that of the Sync Field lower limit register (SFLR) and Sync Field upper limit register (SFUR), the value of the serial timer register (STMR) is set in the baud rate generator register (BGR1, BGR0), and the baud rate setting flag (SACSR:BST) is set to "1".
- When the value of serial timer register (STMR) is below that of the Sync Field lower limit register (SFLR) or exceeds that of the Sync Field upper limit register (SFUR), the baud rate generator register (BGR1, BGR0) will not be changed and the baud rate setting flag (SACSR:BST) is reset to "0".

Figure 7-9 LIN Break Field Reception to Sync Field Reception (When STMR Is between SFUR and SFLR)

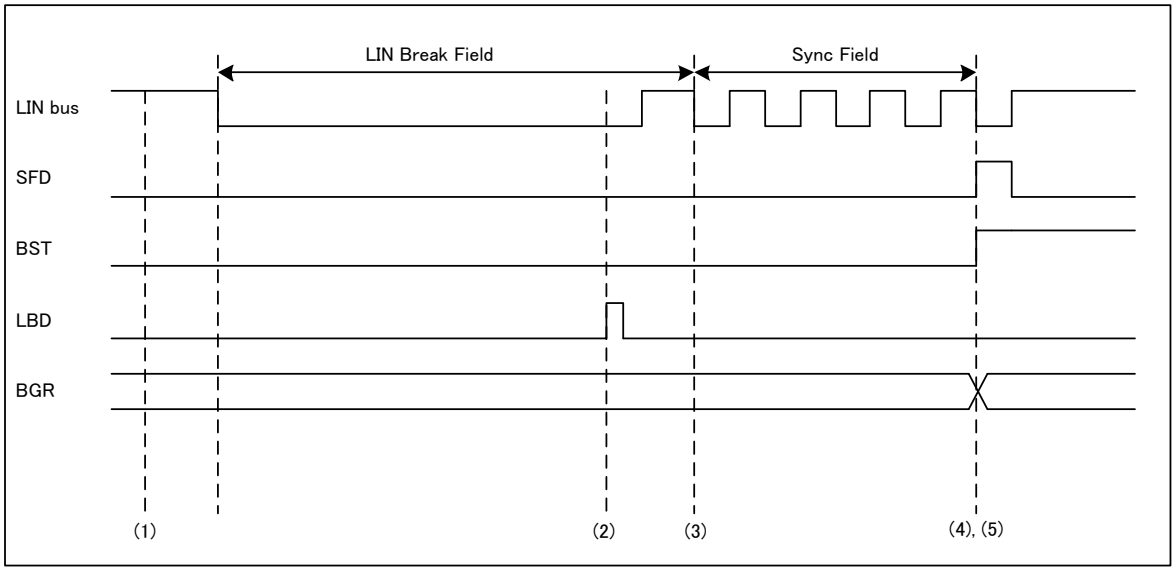
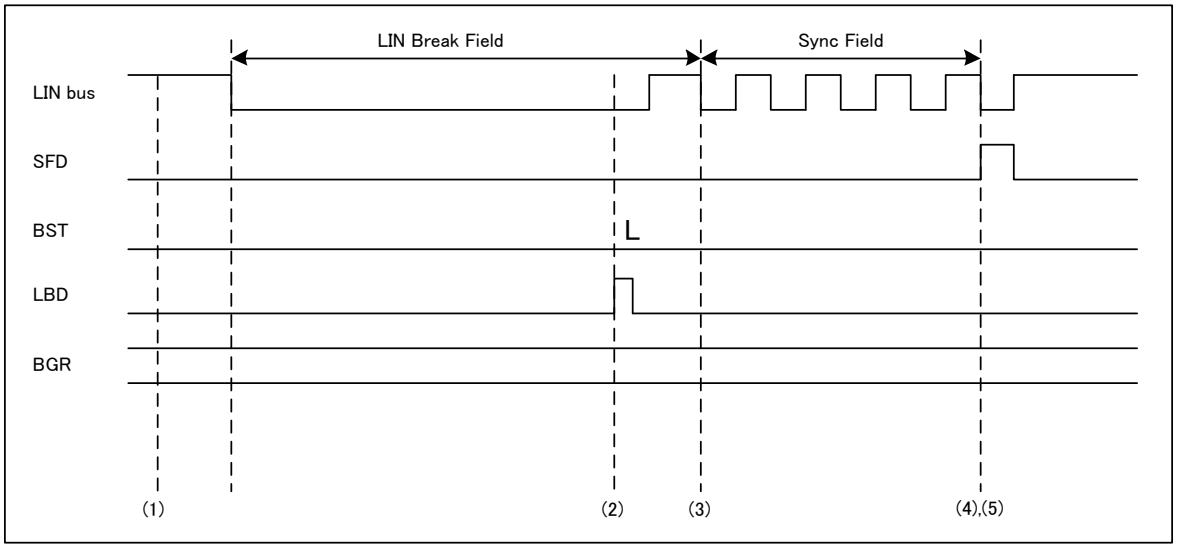


Figure 7-10 LIN Break Field Reception to Sync Field Reception (When STMR Is Not between SFUR and SFLR)



Notes:

- *Disable reception (SCR:RXE = 0) for the LIN Break Field and Sync Field.*
- *When auto baud rate adjustment is enabled (SACSR:AUTE = 1), set in the internal clock(BGR1:EXT=0),*

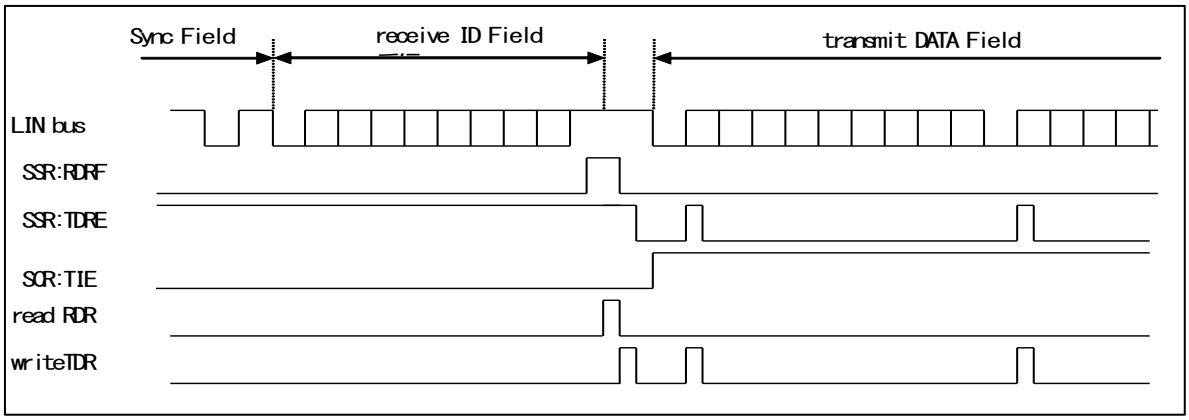
ID Field Reception to DATA Field Transmission and Reception

After the reception of the ID Field, it is possible to select whether the DATA Field is to be transmitted to or received by the master device.

(When the DATA Field is transmitted)

After receiving the ID Field, write the data into the transmission data register (TDR). In this case, enable the transmission interrupt (SCR:TIE = 1).

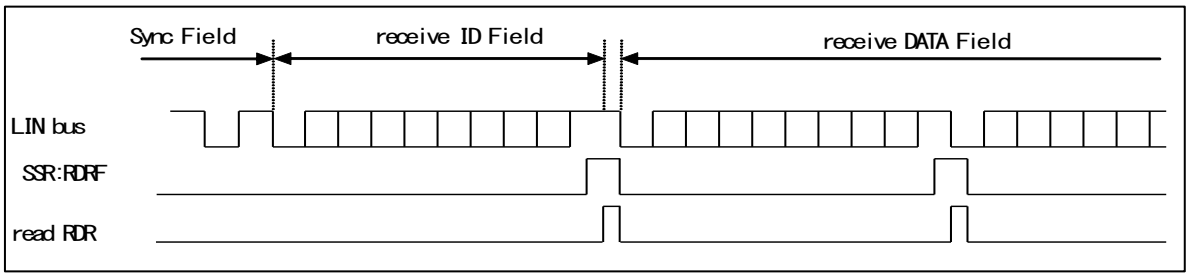
Figure 7-11 ID Field Reception to DATA Field Transmission



(When the DATA Field is received)

- Every time the DATA Field is received, SSR:RDRF is set to "1". At this time, a reception interrupt occurs if reception interrupt has been enabled (SCR:RIE = 1).
- A start bit is detected when both of the following are satisfied: 1. The noise-filtered result (the result of the rule of majority on the 3-time bus clock sampling of the serial data input) shows falling. 2. The filtered data shows "L" at the sampling point.

Figure 7-12 ID Field Reception to DATA Field Reception



Notes:

- *There is a built-in noise filter (the rule of majority being applied to 3-time bus clock sampling of the serial data input). However, we suggest that you design the board in such a way that noise does not pass through the filter. Alternatively, we suggest that you implement a communication method whereby noise passing through the filter does not cause a problem (by, for example, appending a checksum to the data at the end of transmission for retransmission in case of an error).*
- *During data reception, the following occurs if a falling edge of serial data is detected at the same time as, or 1 or 2 bus clocks earlier than, the sampling point: The edge becomes invalid*

and the frame that follows cannot be received normally. When frames are successively output, it is recommended that intervals be provided between the frames.

Slave Operation Timing Chart

Figure 7-13 LIN Bus Timing (When the DATA Field Is Transmitted: When a FIFO Is Not Used and SACSR:AUTE = 1)

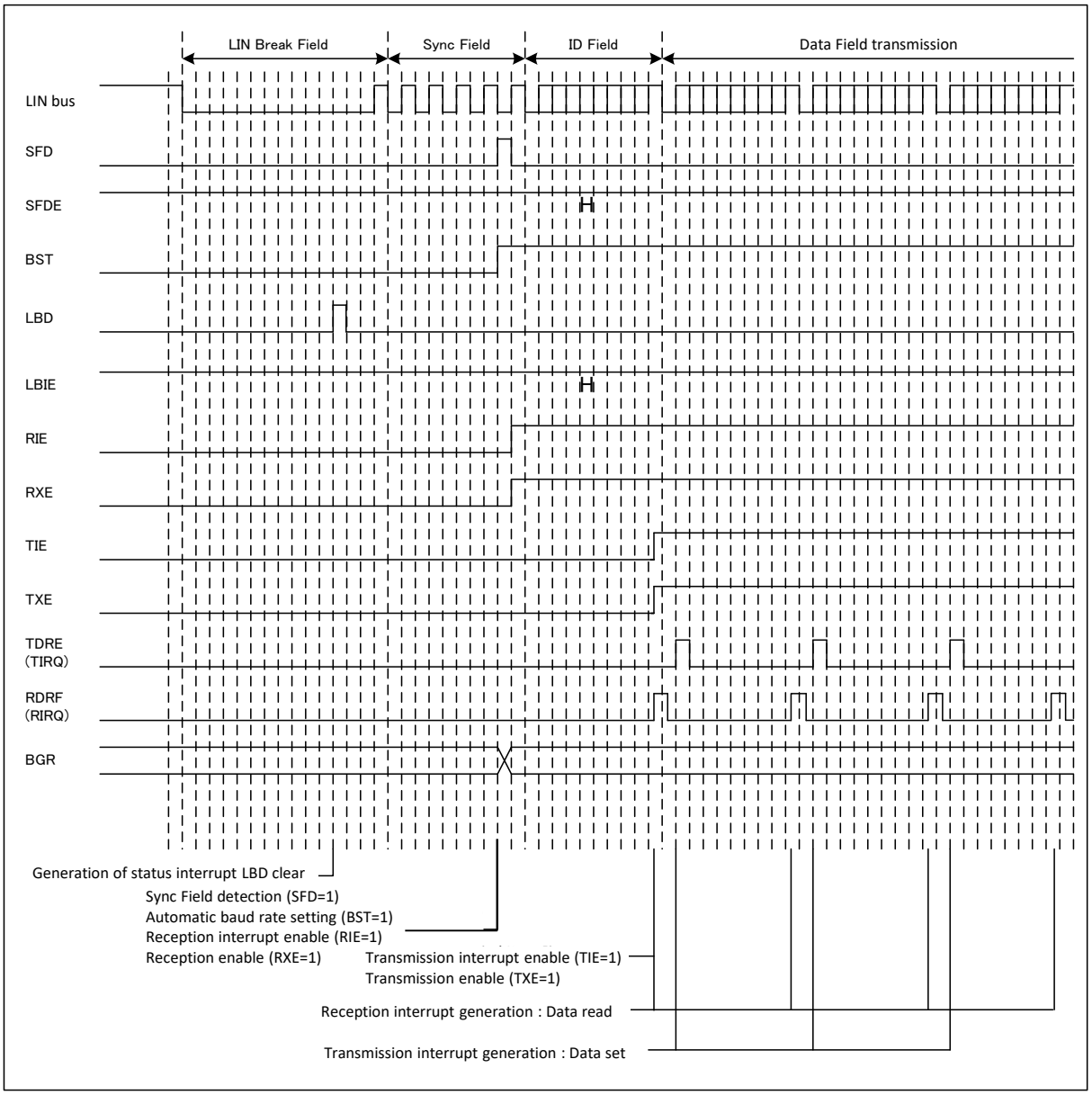
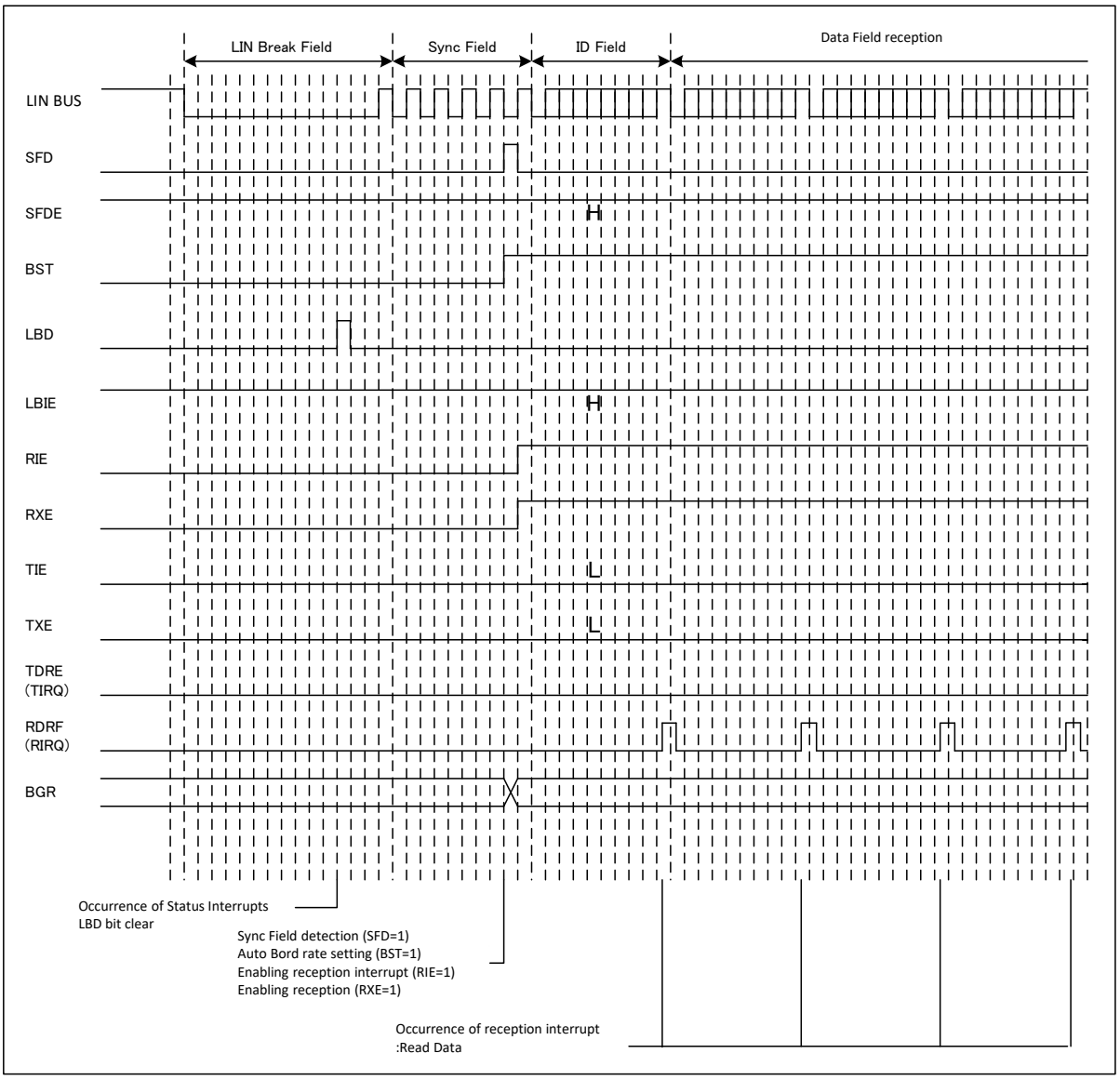


Figure 7-14 LIN Bus Timing (When the DATA Field Is Received: When a FIFO Is Not Used and SACSR:AUTE = 1)



When a FIFO is Used

Figure 7-15 LIN Bus Timing (When DATA Field Is Transmitted: When a FIFO Is Used and SACSR:AUTE = 1)

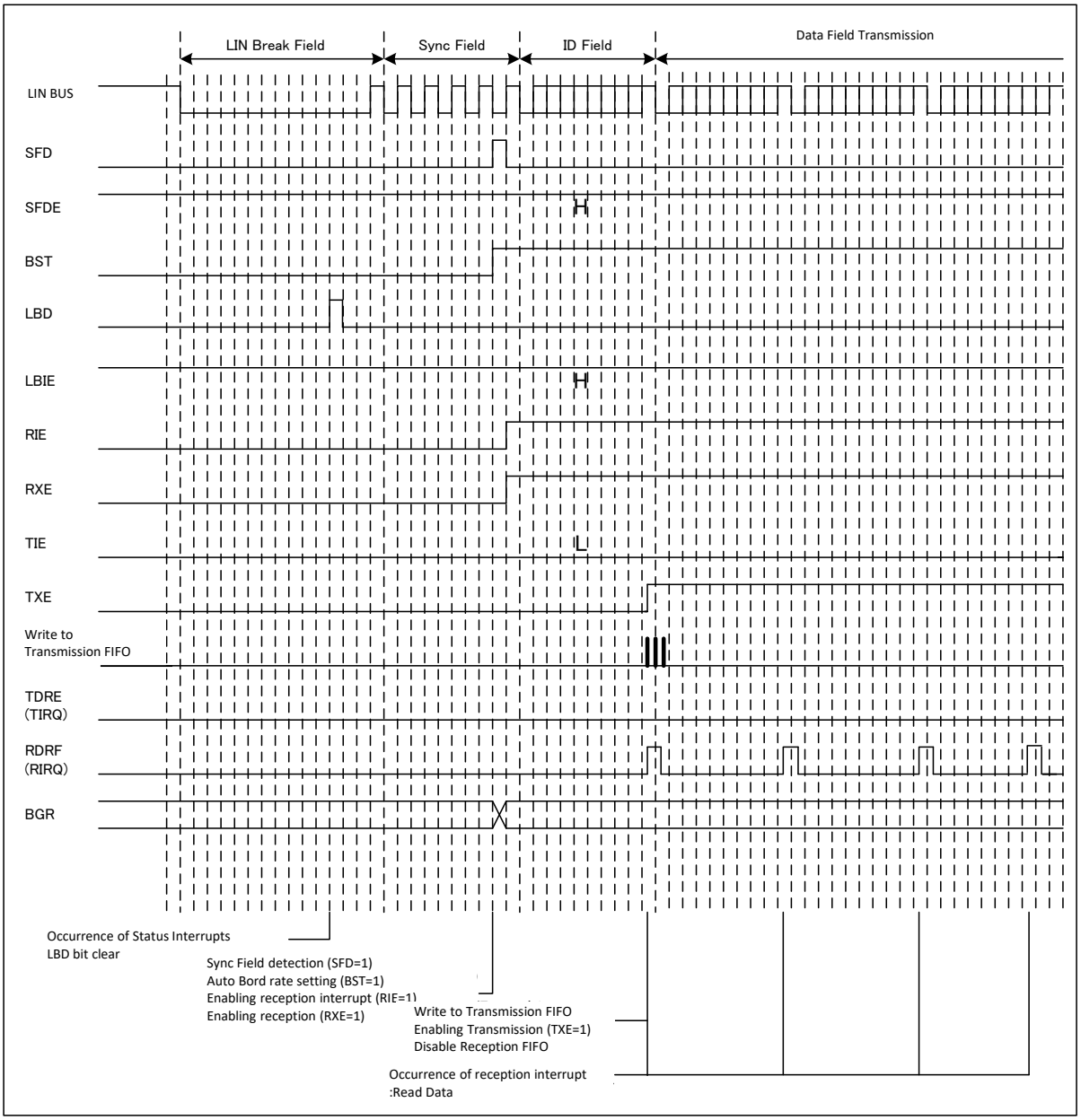
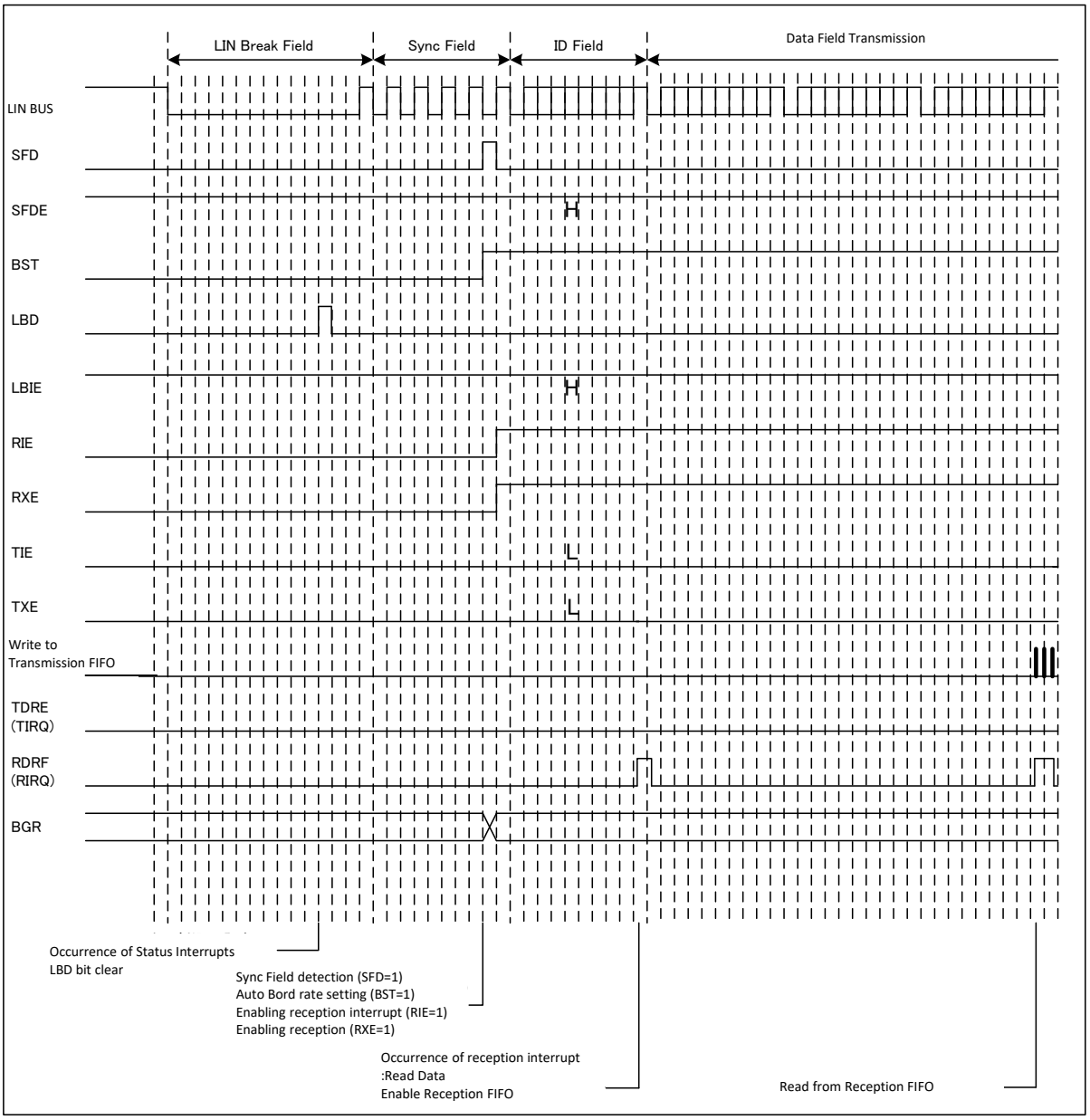


Figure 7-16 LIN Bus Timing (When DATA Field Is Received: When a FIFO Is Used and SACSR:AUTE = 1)



7.2. Assist Mode

Assist mode processes the auto transmission/reception of the LIN header and the following generation and detection.

- Generation and detection of parity for ID Field
- Generation and detection of checksum
- Detection of LIN bus error

Master Operation

Auto Header Transmission Setting

To process auto header transmission using assist mode, after initial setting, set the SCR:LBR bit (LIN Break Field setting bit) to "1". Setting to "1" initiates the auto transmission of the LIN Break Field, Sync Field and ID Field. How to set transmission is described below:

- To enable operation as the master, set the SCR:MS bit (master/slave function selection bit) to "0".
- Set the LAMCR:LAMEN bit (LIN assist mode processing enable bit) to "1".
- Select and set the LIN Break Field length (ESCR:LBL2, LBL1, LBL0) and LIN Break Field delimiter length (ESCR:DEL1, DEL0).
- Select and set the stop bit length (SMR:SBL and ESCR:ESBL).
- When the assist mode transmission ID register (LAMTID) is used for ID transmission, set the LAMCR:LIDEN bit (LIN ID register enable bit) to "1".
  When the data transmission register (TDR) is used for ID transmission, set the LAMCR:LIDEN bit (LIN ID register use enable bit) to "0".
- Set the ID value in the register selected for ID transmission.
- Set the LIN data length (LAMCR:LDL3 to LDL0) which corresponds to the ID.
- Select and set the checksum to either standard or extended. (LAMCR:LCSTYP).
- Disable the LIN Break Field interrupt enable bit (ESCR:LBIE = 0). When LIN Break Field interrupt enable bit is enabled (ESCR:LBIE = 1), since LIN Break Field is also detected at the master, a status interrupt (SSR:LBD bit (LIN Break Field detection flag)) occurs.
- Disable Sync Field detection interrupt enable bit (SACSR:SFDE = 0). When the Sync Field detection interrupt enable bit is enabled (SACSR:SFDE = 1), since the Sync Field is also detected at the master, a status interrupt (SACSR:SFD bit (Sync Field detection flag)) occurs.
- Set the reception operation enable bit (SCR:RXE) to "0" (reception disabled).

LIN Break Field to ID Field Transmission

- Set the transmission operation enable bit (SCR:TXE) to "1" (transmission enabled).
- Set the LIN Break Field setting bit (SCR:LBR) to "1" (LIN Break Field generation). The LIN Break Field set at ESCR:LBL2 to LBL0 is transmitted.
- The master receives the LIN Break Field transmitted by the master, and checks for bus errors.
- After the transmission of the LIN Break Field, the LIN Break Field delimiter set at ESCR:DEL1, DEL0 will be transmitted.
- After the transmission of the LIN Break Field delimiter, the Sync Field (0x55 fixed value) is transmitted.
- The master receives the Sync Field transmitted by the master, and checks for bus error/framing errors.
- After the transmission of the Sync Field, the set ID Field value is transmitted. When the LAMCR:LIDEN bit (LIN ID register use enable bit) is set to "0", the ID Field value is transmitted as set in the transmission data register (TDR). When the LAMCR:LIDEN bit (LIN ID register use enable bit)

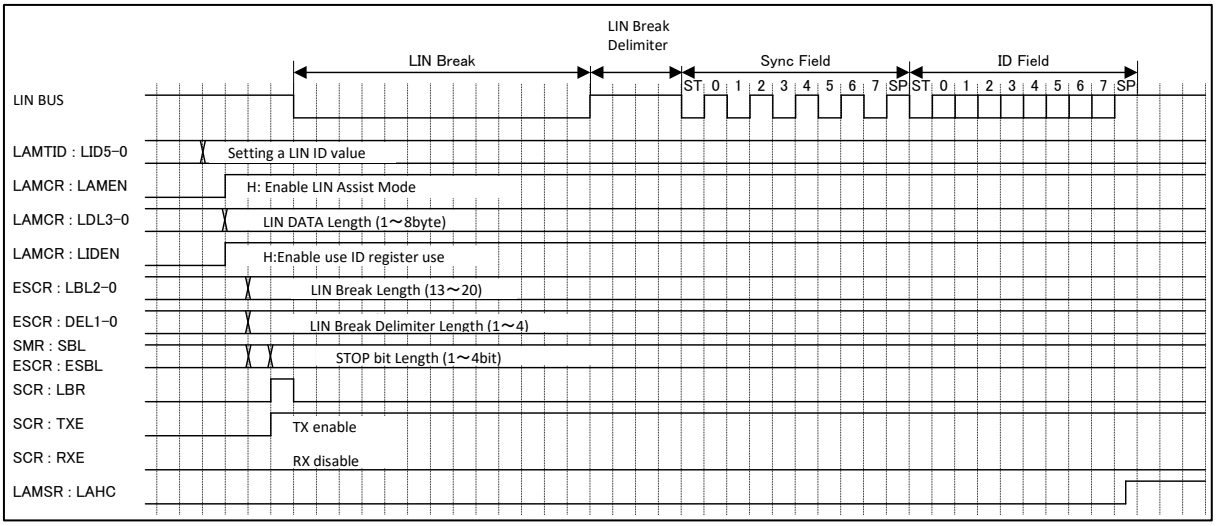
is set to "1", the ID field value is transmitted as set in the LIN assist mode transmission ID register (LAMTID).

- When the transmission data register (TDR) is used for ID Field transmission (LAMCR:LIDEN = 0), and the 1st bit of the ID Field is transmitted, the SSR:TDRE (transmission data empty) bit is set to "1". At this time, a transmission interrupt occurs if the transmission interrupt is enabled (SCR:TIE = 1). When a transmission interrupt (TDRE) occurs, the transmission data can be written in the transmission data register (TDR).
- The data length of the ID Field is 8 bits, and the output is LSB first. The LIN parity in the ID Field is calculated automatically.
- The master receives the ID Field transmitted by the master, and checks for a bus error/framing error.
- When the ID Field transmission is completed, the LIN auto header completion flag is set (LAMSR:LAHC = 1). In this case, when the LIN auto header transmission completion interrupt enable bit is enabled (LAMIER:LAHCIE=1), a status interrupt occurs.
- When the following errors occur, transmission is halted.
  - LIN bus error
  - LIN ID parity error
  - Framing error

Notes:

- *In the header transmission setting (from LBR activation (SCR:LBR = 1) to LIN auto header completion (LAMSR:LAHC = 1)), the auto header transmission setting shall not be changed.*
- *Disable reception (SCR:RXE = 0) while the master header is being transmitted in assist mode.*
- *While the master is operating in assist mode, the transmission data values of the Sync Field and ID Field are not stored to the RDR register.*
- *When response data is received without using the LIN assist mode transmission ID register (LAMTID), and the 1st bit of the ID Field is transmitted, it is set as SSR:TDRE = 1. However, do not write data. In addition, disable the transmission interrupt (SCR:TIE = 0).*
- *When the response data is received using the LIN assist mode transmission ID register (LAMTID), the transmission data register (TDR) can be written after the LIN Break Field setting bit (SCR:LBR) is set to "1". However, do not write data.*

Figure 7-17 LIN Break Field to ID Field Transmission



LIN Break Field Retransmission during Assist Mode Processing

The LIN Break Field (SCR:LBR = 1) shall be set at the following timing.

- After the completion of header transmission (LAMSR:LAHC = 1)
- After the completion of response transmission and reception (LAMSR:LCSC = 1)

Data Field Transmission and Reception

This step selects whether the DATA Field shall be transmitted to or received by a slave device.

(When the DATA Field is transmitted)

- When the LIN assist mode transmission ID register (LAMTID) is not used, after the 1st bit of the ID Field is transmitted, it is set as SSR:TDRE = 1. At this time, the DATA Field can be written.
- When the LIN assist mode transmission ID register (LAMTID) is used, and the LIN Break Field setting bit (SCR:LBR) is set to "1", the DATA Field becomes writable.
- After the LIN Break Field setting bit (SCR:LBR) is set to "1" but before the start of response transmission, enable transmission (SCR:TXE = 1).
- In LIN assist mode, the checksum calculation is performed automatically. For the checksum calculation, the calculation method can be selected with the LIN checksum type selection bit (LAMCR:LCSTYP).
- After the completion of the checksum calculation, the checksum calculation completion flag (LAMSR:LCSC) is set. In this case, when the checksum calculation completion interrupt enable bit is set (LAMIER:LCSCIE = 1), a status interrupt occurs.
- After the completion of response transmission (LAMSR:LCSC = 1), disable transmission (SCR:TXE = 0).

Notes:

- *While a response is being transmitted during an assist mode operation, do not enable reception (SCR:RXE = 1).*
- *During assist mode operation, response transmission data (Data Field, checksum) cannot be stored into the RDR register.*
- *When a FIFO is being used, after setting the LIN Break Field setting bit (SCR:LBR) to "1" (LIN Break Field generation bit), write data to the FIFO.*
- *When the LIN data length is set to 0 bytes (LAMCR:LDL3 to LDL0 = 0000) for response transmission, write dummy data into the TDR register to automatically perform the checksum calculation and then transmit the data. (Any value can be written.) The TDR setting value does not affect the checksum calculation.*
- *When the LIN data length is set to "0" (LAMCR:LDL3 to LDL0 = 0000), the checksum values will be as follows:*
  - With the standard checksum setting (LAMCR:LCSTYP = 0), the checksum value is 0xFF.
  - With the extended checksum setting (LAMCR:LCSTYP = 1), the checksum value is the inverse value of the ID Field.
- *During the transmission of response data, do not change the transmission data length (LAMCR:LDL3 to LDL0).*

Figure 7-18 ID Field Transmission to DATA Field Transmission (When ID Register Is Used and a FIFO Is Not Used)

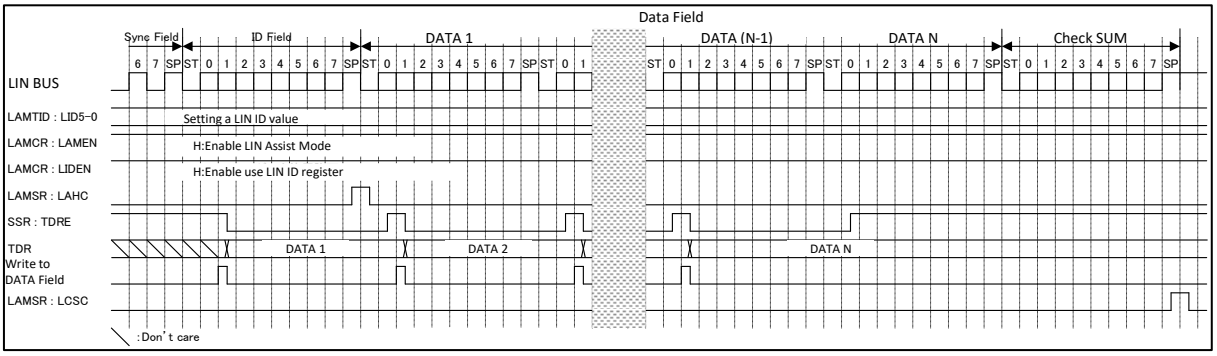
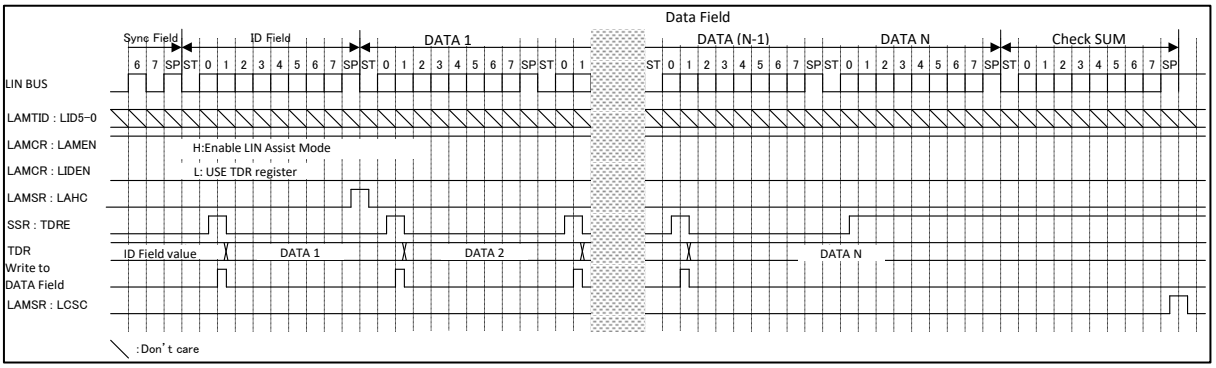


Figure 7-19 ID Field Transmission to DATA Field Transmission (When ID Register Is Not Used and a FIFO Is Not Used)



(When the DATA Field is received)

- Between LIN auto header completion (LAMSR:LAHC = 1) and the start of response reception, enable reception (SCR:RXE = 1) and disable transmission (SCR:TXE = 0).
- When the DATA Field is received, SSR:RDRF is set to 1. At this time, a reception interrupt occurs if a reception interrupt is enabled (SCR:RIE = 1).
- After the completion of checksum reception, the LIN checksum calculation completion flag is set (LAMSR:LCSC = 1). In this case, when the checksum calculation completion interrupt enable bit is set (LAMIER:LCSCIE = 1), a status interrupt occurs.
- After the completion of checksum reception (LAMSR:LCSC = 1), disable reception (SCR:RXE = 0).

Notes:

- *During data reception, the following occurs if a falling edge of serial data is detected at the same time as, or 1 or 2 bus clocks earlier than, the sampling point: The edge becomes invalid and the frame that follows cannot be received normally. When frames are successively output, it is recommended that intervals be provided between the frames.*
- *The checksum value for response reception during assist mode operation is not stored into the RDR register.*
- *When the LIN data length is set to "0" (LAMCR:LDL3 to LDL0 = 0000), the checksum values will be as follows:*

- With the standard checksum setting (LAMCR:LCSTYP = 0), the checksum value is 0xFF.
- With the extended checksum setting (LAMCR:LCSTYP = 1), the checksum value is the inverse value of the ID Field.

– *During the reception of response data, do not change the reception data length (LAMCR:LDL3 to LDL0).*

Figure 7-20 ID Field Transmission to DATA Field Reception (When the ID Register Is Used and a FIFO Is Not Used)

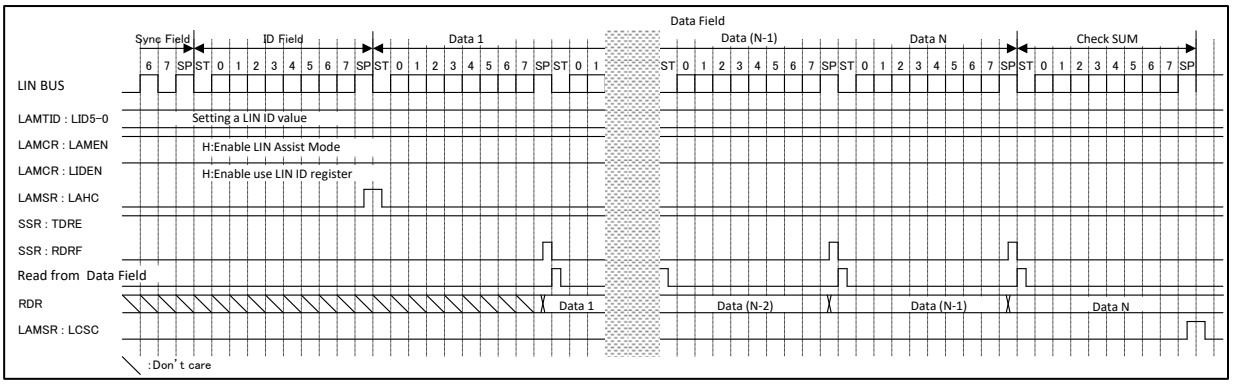
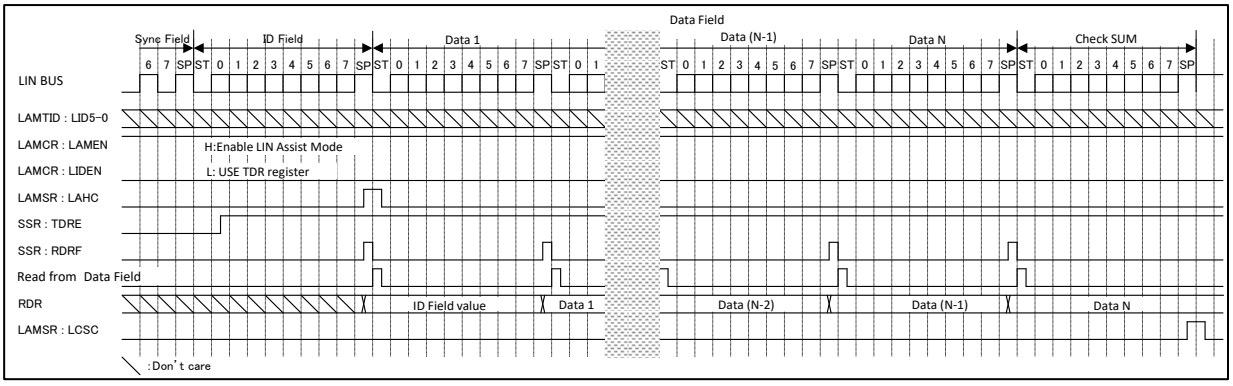


Figure 7-21 ID Field Transmission to DATA Field Reception (When the ID Register Is Not Used and a FIFO Is Not Used)



Master Operation Time Chart (when a FIFO is Not Used)

Figure 7-22 LIN Bus Timing (the ID Register Is Used, the DATA Field Is Transmitted, and a FIFO Is Not Used)

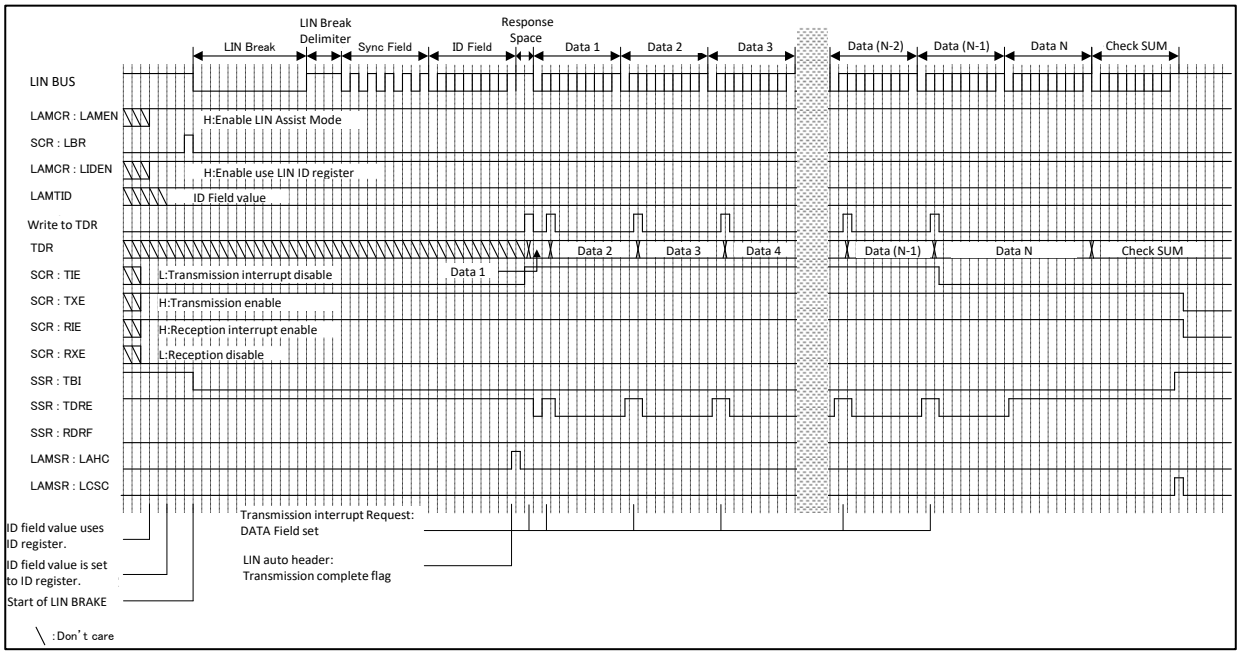


Figure 7-23 LIN Bus Timing (the ID Register Is Not Used, the DATA Field Is Transmitted, and a FIFO Is Not Used)

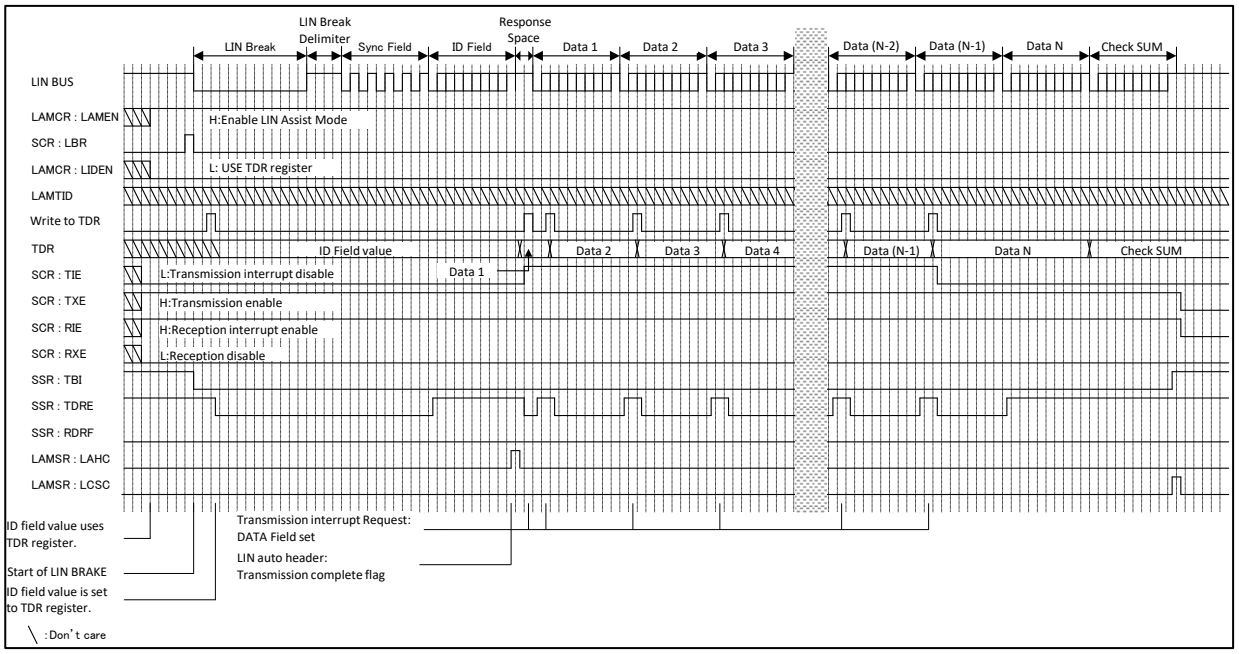


Figure 7-24 LIN Bus Timing (the ID Register Is Used, the DATA Field Is Received, and a FIFO Is Not Used)

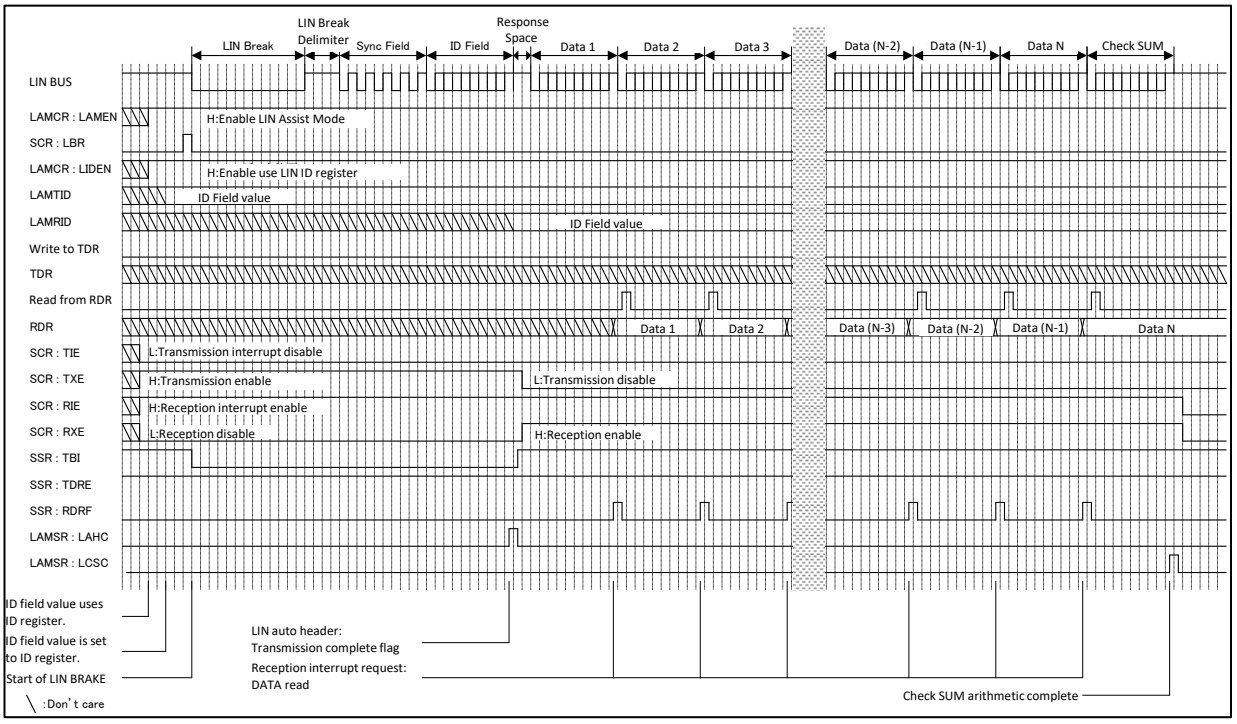


Figure 7-25 LIN Bus Timing (the ID Register Is Not Used, the DATA Field Is Received, and a FIFO Is Not Used)

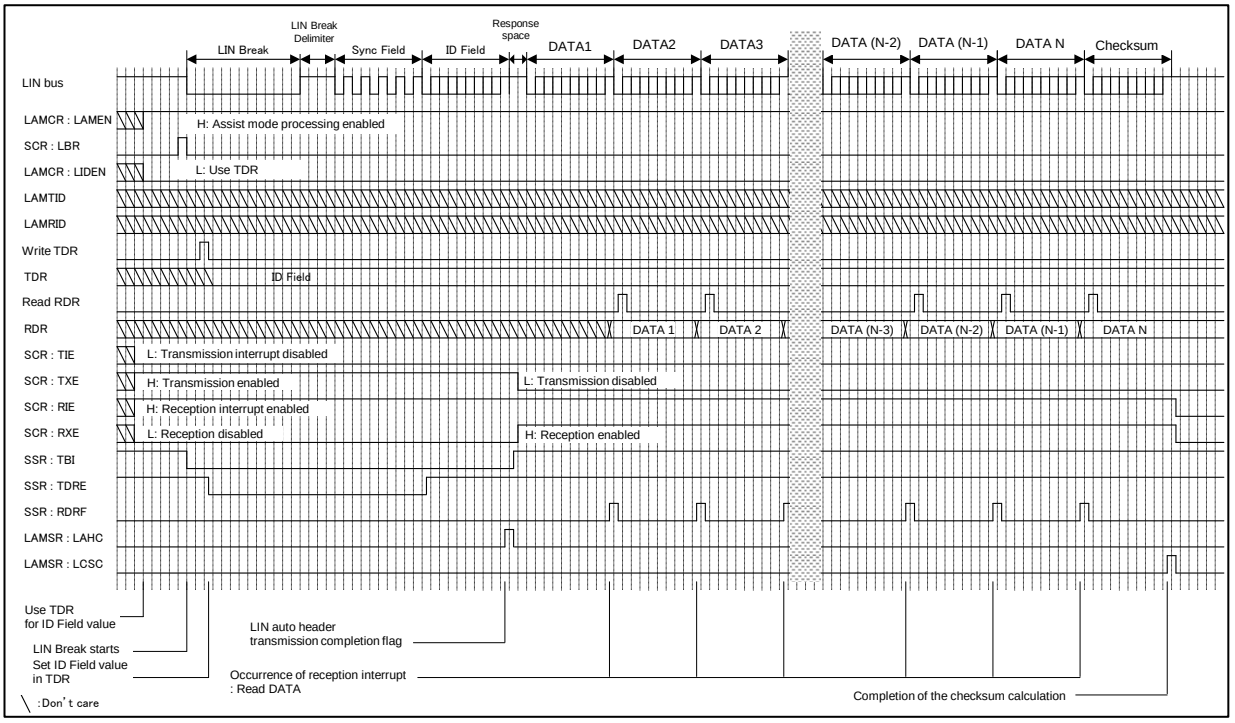


Figure 7-26 LIN Bus Timing (the ID Register Is Used, the DATA Field Is Transmitted, and a FIFO Is Used)

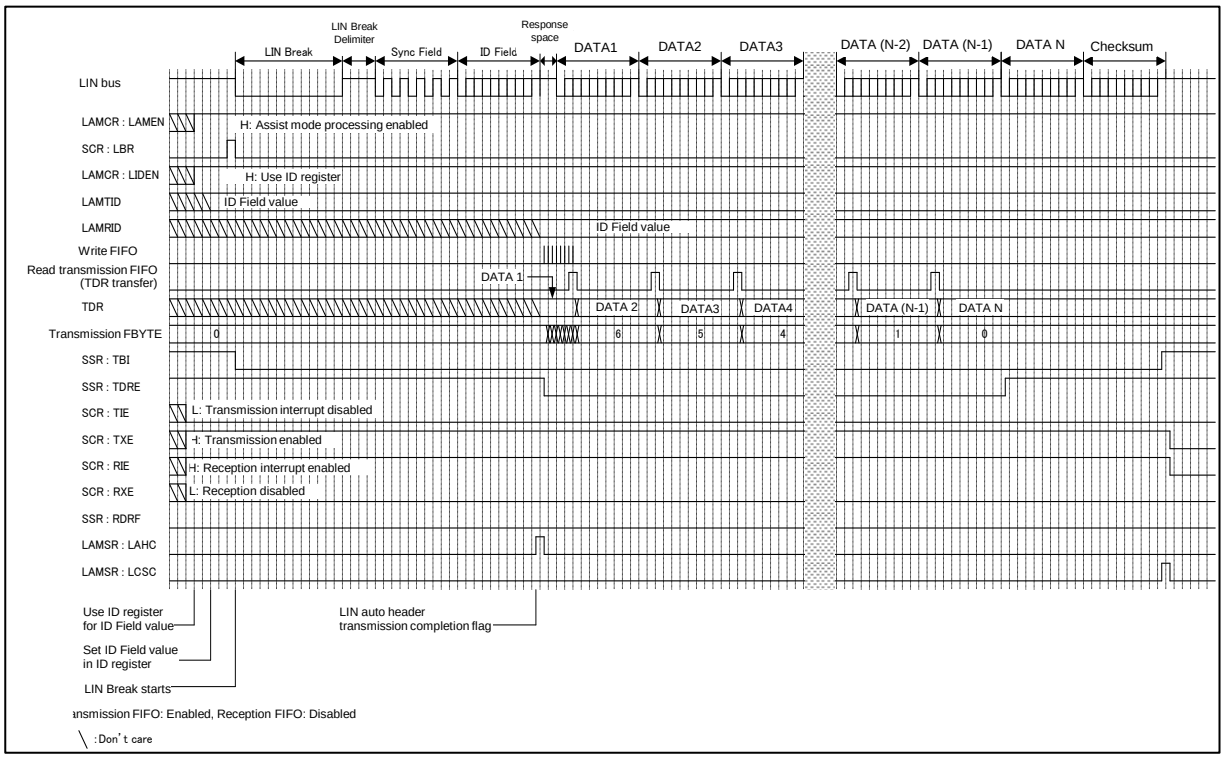


Figure 7-27 LIN Bus Timing (the ID Register Is Not Used, the DATA Field Is Transmitted, and a FIFO Is Used)

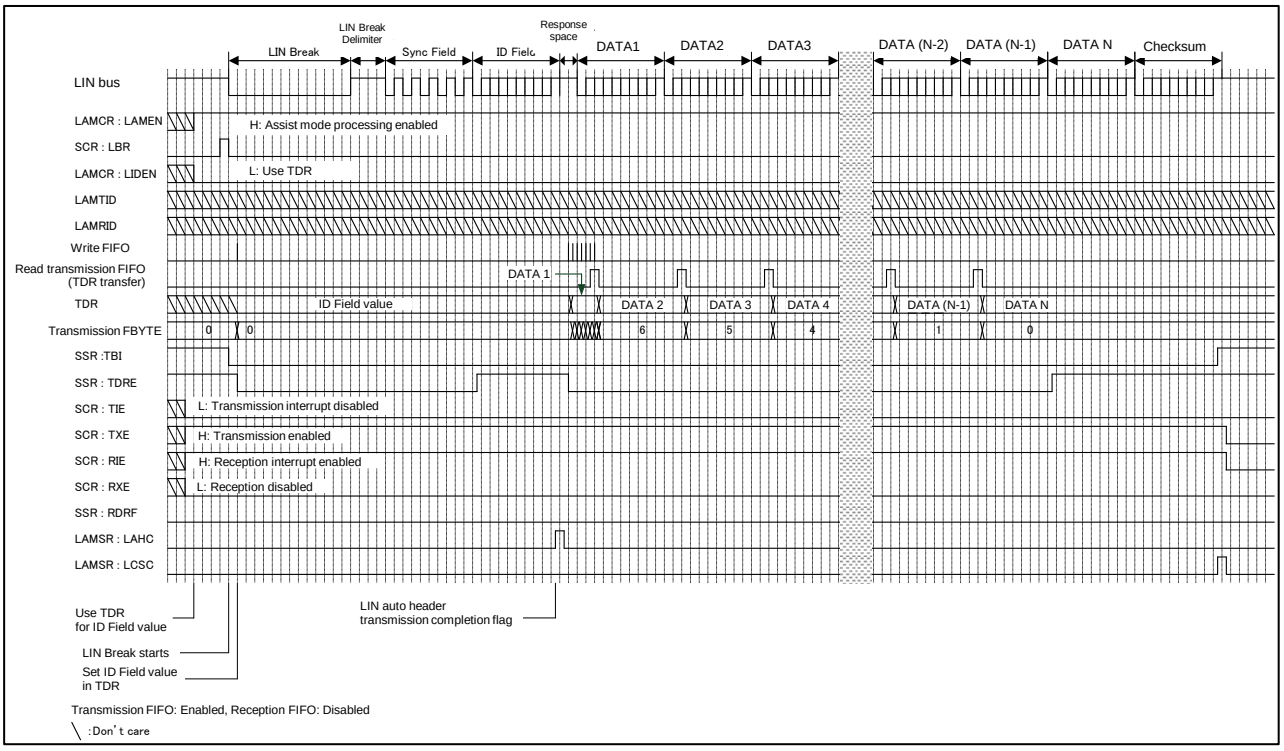


Figure 7-28 LIN Bus Timing (the ID Register Is Used, the DATA Field Is Received, and a FIFO Is Used)

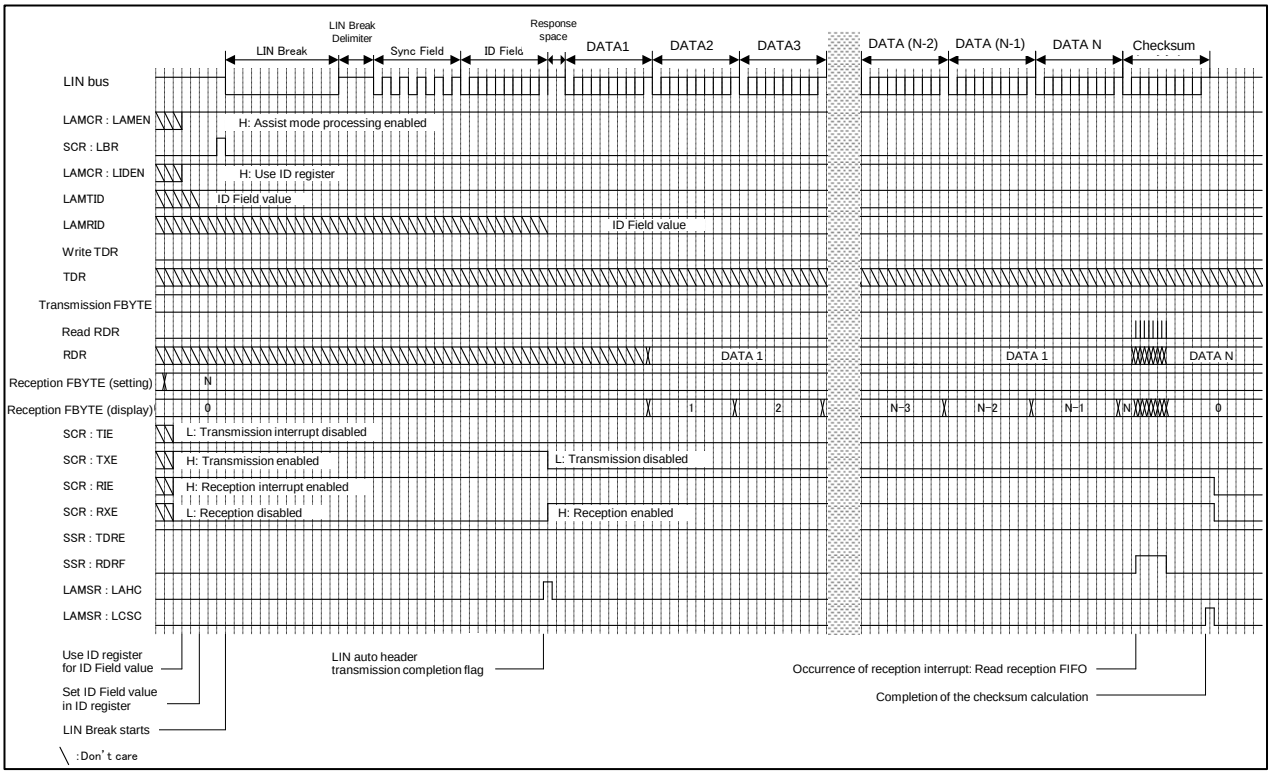
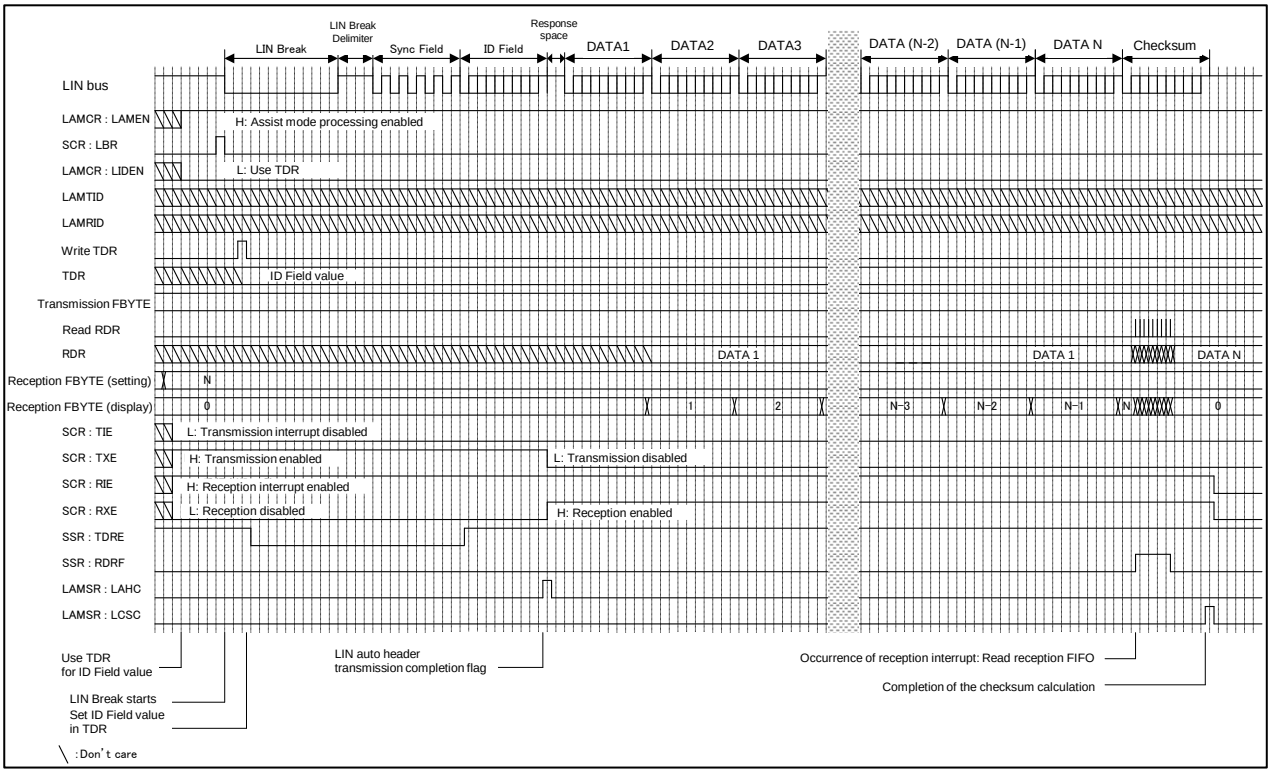


Figure 7-29 LIN Bus Timing (the ID Register Is Not Used, the DATA Field Is Received, and a FIFO Is Used)



Slave Operation

Auto Header Reception Setting

To process auto header reception using assist mode, set the following:

- To enable operation as a slave, set the SCR:MS bit (master/slave function selection bit) to "1".
- To enable operation using LIN assist mode, set the LAMCR:LAMEN bit (LIN assist mode processing enable bit) to "1".
- Select and set the stop bit length (SMR:SBL and ESCR:ESBL).
- When the assist mode reception ID register (LAMRID) is used for ID reception, set the LAMCR: LIDEN bit (LIN ID register enable bit) to "1".
  When the data reception register (RDR) is used for ID reception, set the LAMCR:LIDEN bit (LIN ID register use enable bit) to "0".
- To enable auto baud rate adjustment, set the SACSR:AUTE bit (auto baud rate adjustment bit) to "1" and set in the internal clock(BGR1:EXT = 0).
- Set the reception enable bit (SCR:RXE) to "0" (reception disabled).

LIN Break Field Reception to ID Field Reception

(1) When the LIN Break Field is entered, the LIN Break Field is detected (SSR:LBD = 1) at the 11th bit. At this time, a status interrupt occurs if the ESCR:LBIE bit is set to "1".

The following operations are performed as part of auto baud rate adjustment.

(2) When the LIN Interface (v2.1) detects the 1st falling edge of the Sync Field, it initializes the serial timer register (STMR) to 0.

(3) When the 5th falling edge of the Sync Field is detected, the Sync Field detection flag (SACSR:SFD) is set to "1". At this time, a status interrupt occurs if the SACSR:SFDE bit is set to "1".

(4) When the falling edge of the 5th Sync Field is detected, the following operation is performed depending on the value of the serial timer register (STMR).

- When the value of the serial timer register (STMR) is between that of the Sync Field lower limit register (SFLR) and Sync Field upper limit register (SFUR), the value of the serial timer register (STMR) is set in the baud rate generator register (BGR1, BGR0), and the baud rate setting flag (SACSR:BST) is set to "1".
- When the value of the serial timer register (STMR) is below that of the Sync Field lower limit register (SFLR) or exceeds that of the Sync Field upper limit register (SFUR), the baud rate generator register (BGR1, BGR0) will not be changed and the baud rate setting flag (SACSR:BST) is reset to "0".

Figure 7-30 LIN Break Field to ID Field Reception (When STMR Is Between SFUR and SFLR)

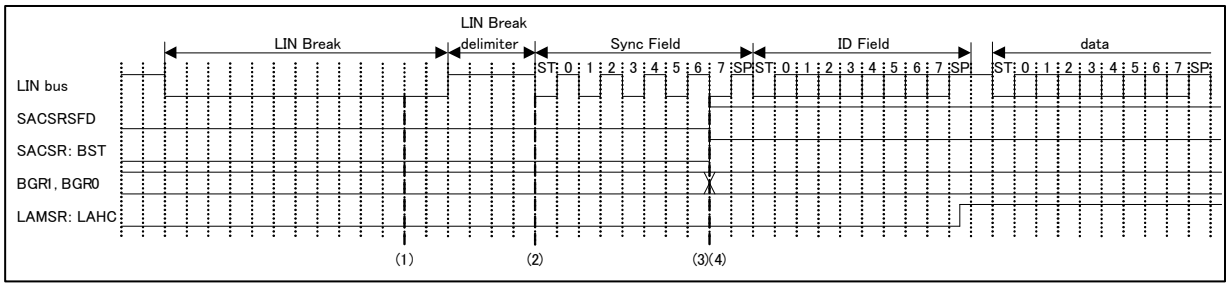
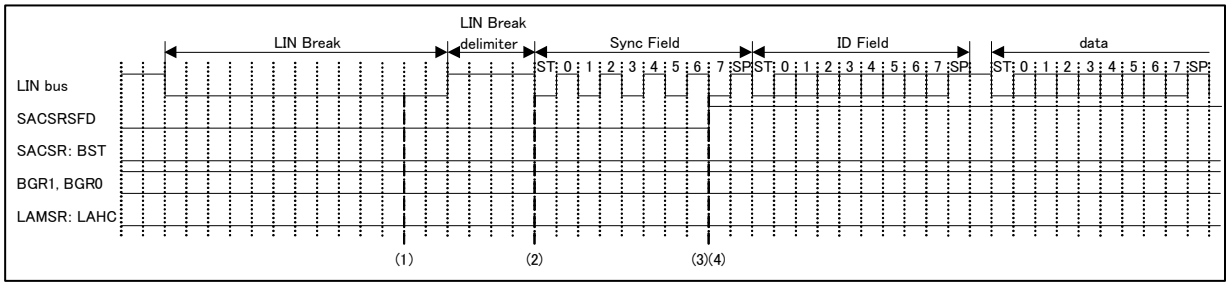


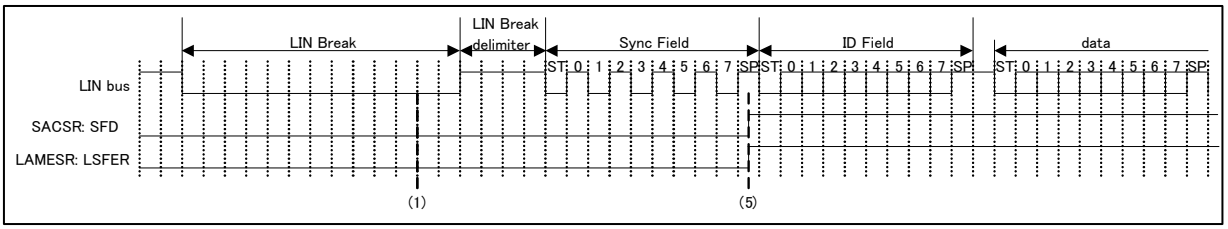
Figure 7-31 LIN Break Field to ID Field Reception (When STMR Is Not between SFUR and SFLR)



The following operations are performed when auto baud rate adjustment is not conducted.

(5) When auto baud rate adjustment is not conducted, set the SACSR:AUTE bit (auto baud rate adjustment bit) to "0". Treat the Sync Field value as data, and verify that the Sync Field value is 0x55. When it is 0x55, the Sync Field detection flag (SACSR:SFD) is set to "1". When it is other than 0x55, the Sync Data error flag bit (LAMESR:LSFER) is set to "1".

Figure 7-32 LIN Break Field to ID Field Reception (When Auto Baud Rate Adjustment Is Not Performed)



The following operations are common both to when auto baud rate adjustment is performed and when it is not performed.

(6) When auto header reception is completed in LIN assist mode, the LAMSR:LAHC bit (LIN auto header completion flag) is set to "1". If a LIN parity error (LAMESR:LPTER = 1) occurs in the ID Field, the LAMSR:LAHC bit (LIN auto header completion flag) is set to "1". So, if the LAMSR:LAHC bit is set to "1", verify that the error is not detected.

(7) When the ID Field is correctly received, set the checksum type (LAMCR:LCSTYP) and the LIN data length setting bit (LAMCR:LDL3 to LDL0).

Figure 7-33 LIN Break Field to ID Field Reception (When a Parity Error Occurs)

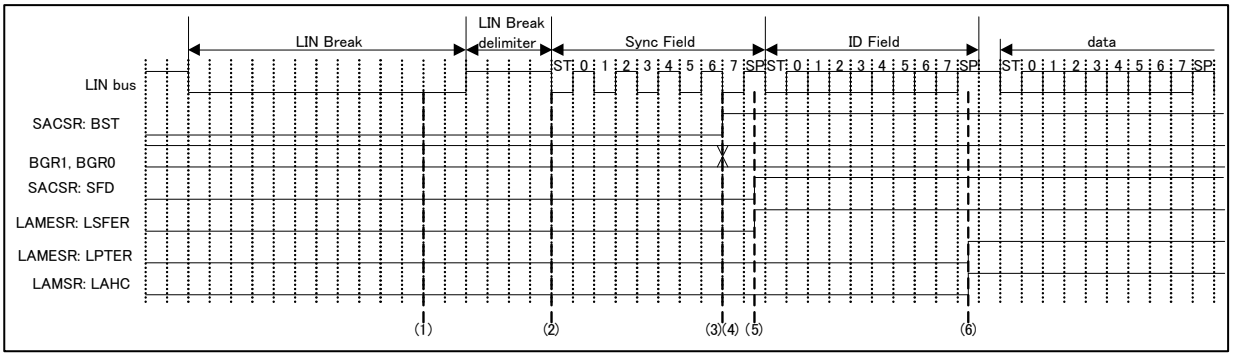
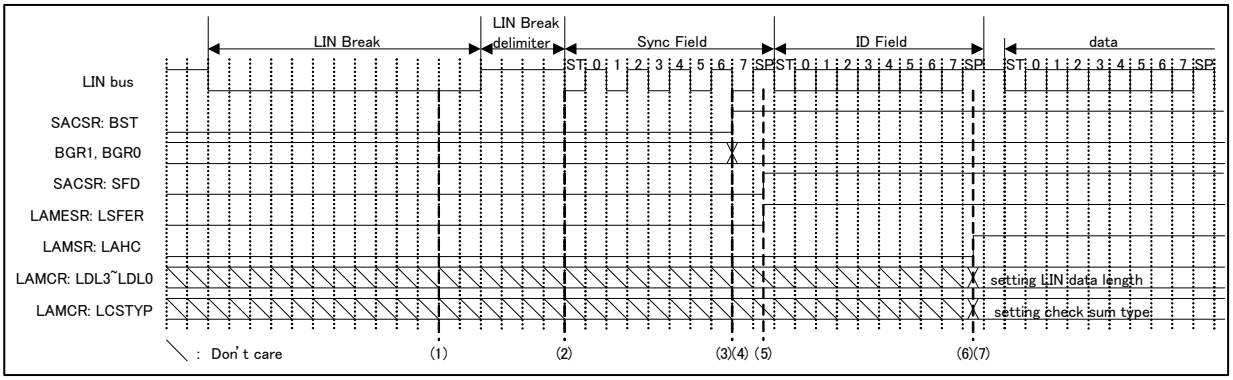


Figure 7-34 LIN Break Field to ID Field Reception (When the LIN Data Length Is Set)



Notes:

- *Disable reception (SCR:RXE = 0) while the slave header is being received in assist mode.*

- *The setting of the transmission enable bit (SCR:TXE) is ignored while the slave header is received in assist mode.*
- *While the slave is operating in assist mode, the Sync Field value cannot be stored in the reception data register (RDR).*
- *During slave operation, suppose that the setting is such that the reception data register (RDR) is used for the reception of the ID Fields (LAMCR:LIDEN = 0). Then, when an ID Field is received, the received ID value is stored into the reception data register (RDR) and the reception data full flag bit is set (SSR:RDRF = 1). Check the ID value after the LIN auto header completion flag is set (LAMSR:LAHC = 1).*
- *Even when it is determined that there is no need for auto baud rate adjustment (SACSR:BST = 0), there are cases that the reception of the ID Field continues, and LIN auto header completion flag (LAMSR:LAHC = 1) and error flag may be set. Therefore, when the LIN auto header completion flag (LAMSR:LAHC) is set to "1", verify the baud rate setting flag.*

ID Field Reception to DATA Field Transmission and Reception

After the reception of the ID Field, it is possible to select whether the DATA Field is to be transmitted to or received by the master device.

(When the DATA Field is transmitted)

- Using the received ID Field value, set the checksum type (LAMCR:LCSTYP) and LIN data length setting bit (LAMCR:LDL3 to LDL0).
- After receiving the ID Field, write the data into the transmission data register (TDR). In this case, enable transmission (SCR:TXE = 1) and the transmission interrupt (SCR:TIE = 1).
- Perform the checksum calculation based on the LIN data length setting bit (LAMCR:LDL3 to LDL0), and automatically transmit the checksum after transmitting the final data.
- For the checksum calculation, the calculation method can be selected with the LIN checksum type selection bit (LAMCR:LCSTYP).
- After the reception of checksum data, set the checksum calculation completion flag (LAMSR:LCSC). In this case, when the checksum calculation completion interrupt enable bit is set (LAMIER:LCSCIE = 1), a status interrupt occurs.
- After the completion of the checksum calculation (LAMSR:LCSC = 1), disable transmission (SCR:TXE = 0).

Notes:

- *While a response is being transmitted during an assist mode operation, disable reception (SCR:RXE = 0).*
- *During assist mode operations, response transmission data (data, checksum) cannot be stored in the reception data register (RDR).*
- *When the LIN data length is set to 0 bytes (LAMCR:LDL3 to LDL0 = 0000) for response transmission, write dummy data to the TDR register to automatically start the checksum calculation and transmit the data. (Any value can be written.) The TDR setting value does not affect the checksum calculation.*
- *When the LIN data length is set to "0" (LAMCR:LDL3 to LDL0 = 0000), the checksum values will be as follows:*
  - With the standard checksum setting (LAMCR:LCSTYP = 0), the checksum value is 0xFF.
  - With the extended checksum setting (LAMCR:LCSTYP = 1), the checksum value is the inverse value of the ID Field.

Figure 7-35 ID Field Reception to DATA Field Transmission (When the ID Register Is Used)

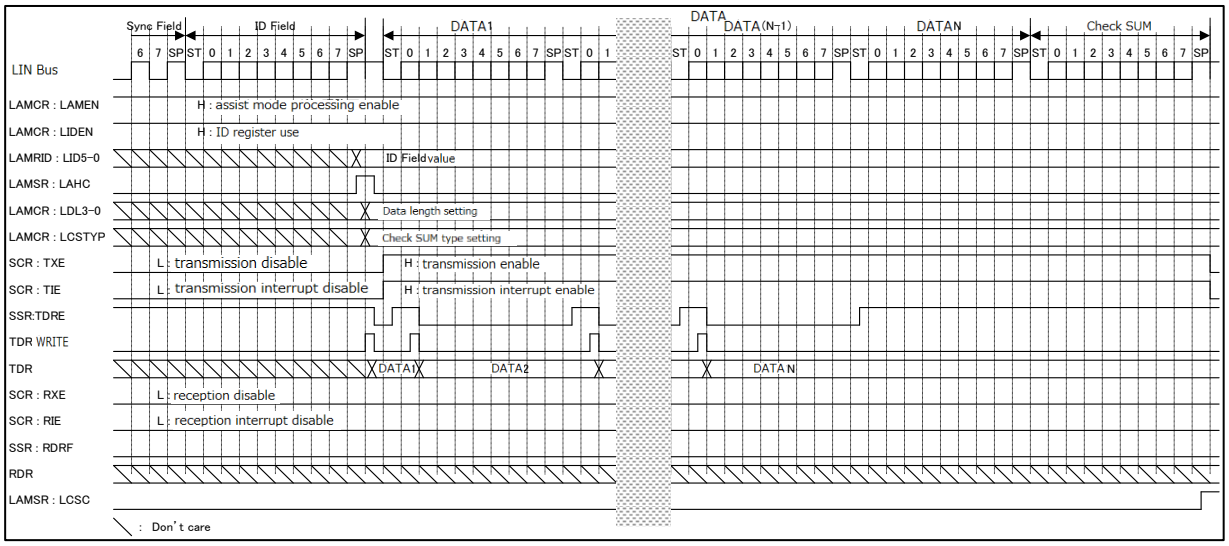
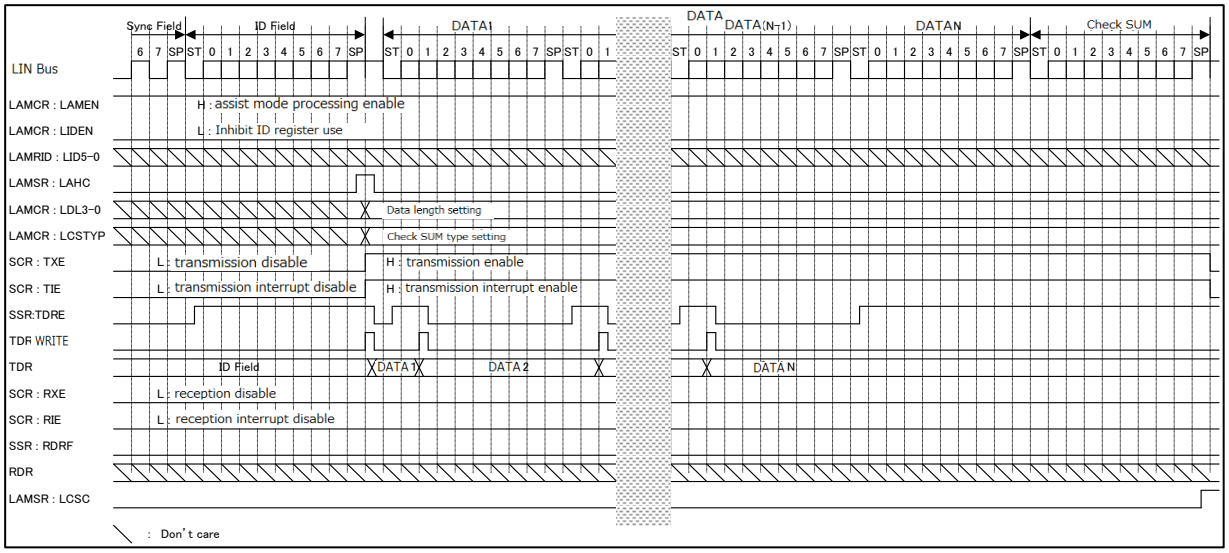


Figure 7-36 ID Field Reception to DATA Field Transmission (When the ID Register Is Not Used)



(When the DATA Field is received)

- Using the received ID Field value, set the checksum type (LAMCR:LCSTYP) and LIN data length setting bit (LAMCR:LDL3 to LDL0).
- Enable reception (SCR:RXE = 1).
- The checksum calculation is performed automatically. For the checksum calculation, the calculation method can be selected with the LIN checksum type selection bit (LAMCR:LCSTYP).
- For each reception of DATA Field, SSR:RDRF is set to "1". A reception interrupt occurs if the reception interrupt is enabled (SCR:RIE = 1).
- When the checksum calculation completion flag is set (LCSC = 1), check for a checksum error. In this case, when the checksum calculation completion interrupt is enabled, a status interrupt occurs. When the checksum error interrupt is enabled, a reception interrupt occurs.
- After the completion of checksum reception (LAMSR:LCSC = 1), disable reception (SCR:RXE = 0).

Notes:

- *During data reception, the following occurs if a falling edge of serial data is detected at the same time as, or 1 or 2 bus clocks earlier than, the sampling point: The edge becomes invalid and the frame that follows cannot be received normally. When frames are successively output, it is recommended that intervals be provided between the frames.*
- *The checksum value of response reception during assist mode operations is not stored in the reception data register (RDR).*
- *When the LIN data length is set to "0" (LAMCR:LDL3 to LDL0 = 0000), the checksum values will be as follows:*
  - With the standard checksum setting (LAMCR:LCSTYP = 0), the checksum value is 0xFF.
  - With the extended checksum setting (LAMCR:LCSTYP = 1), the checksum value is the inverse value of the ID Field.
- *For slave operation, when the LIN communication speed is 19.2 kbps, within about 25 µs after the auto header completion flag is set (LAMSR:LAHC = 1), set the checksum type (LAMCR:LCSTYP) and the LIN data length (LAMCR:LDL3 to LDL0), and disable the transmission (SCR:TXE), the transmission interrupt (SCR:TIE), the reception (SCR:RXE) and the reception interrupt (SCR:RIE) . (Set to a value less than half of one cycle time for LIN communication.)*

Figure 7-37 ID Field Reception to DATA Field Reception (When the ID Register Is Used)

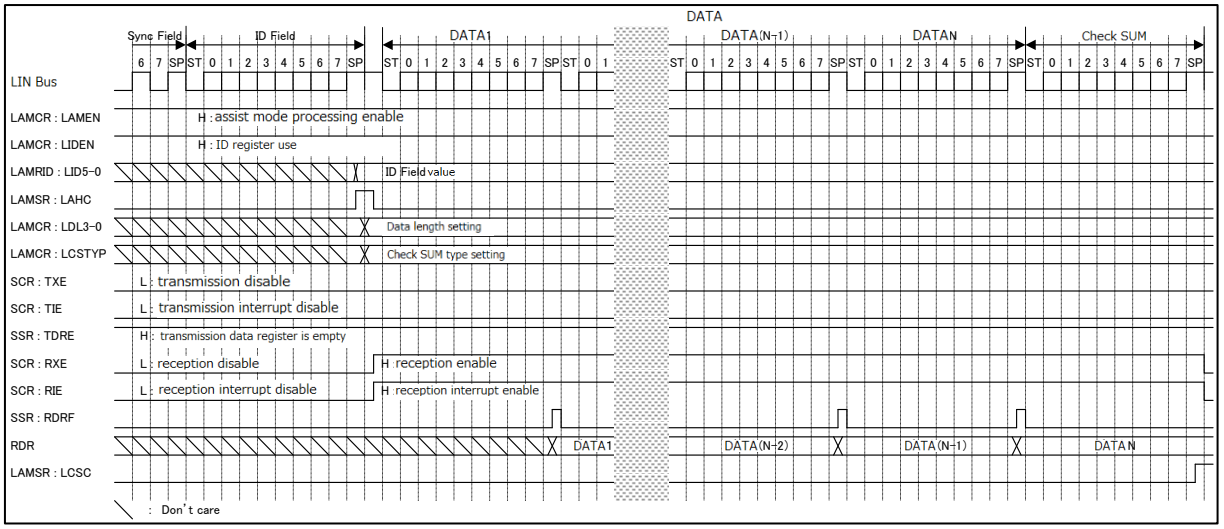
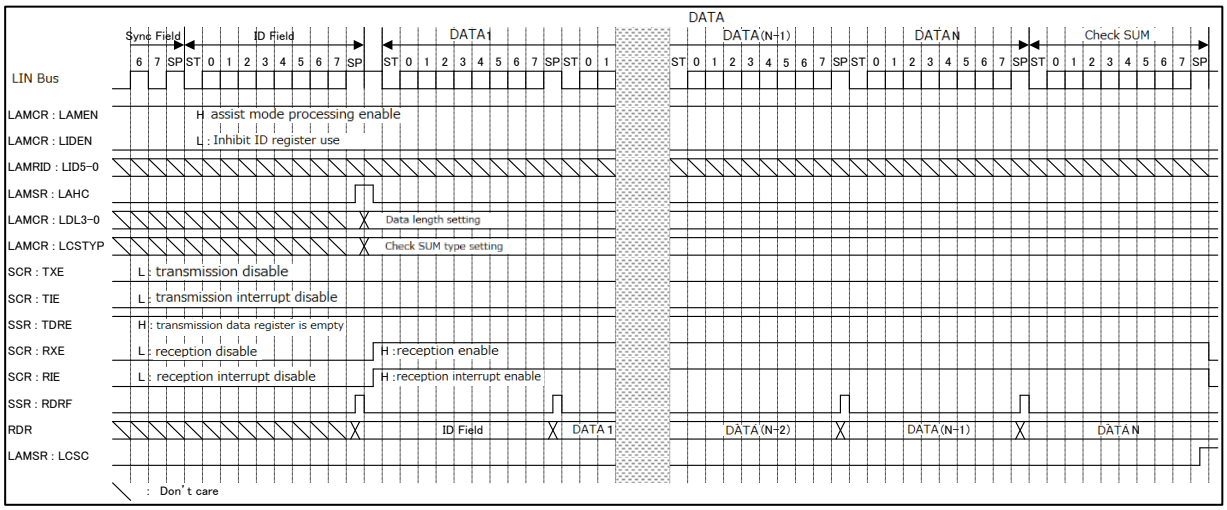


Figure 7-38 ID Field Reception to DATA Field Reception (When the ID Register Is Not Used)



LIN Break Field Reception during Assist Mode Processing

When the LIN Break Field (SSR:LBD = 1) was detected from Sync Field to Checksum, the following procedure is required:

- Disable reception (SCR:RXE = 0) and transmission (SCR:TXE = 0).
- Clear the error flag.
- Abandon any data received before LIN Break Field was detected.
  - When a reception FIFO is used, after disabling the reception FIFO operation (FCR0:FE1 = 0 or FCR0:FE2 = 0), reset the reception FIFO (FCR0:FCL1 = 1 or FCR0:FCL2 = 1).
  - Then, to clear the reception data register, read the RDR register.
- Abandon any data transmitted before the LIN Break Field is detected.
  - When a transmission FIFO is used, after disabling the transmission FIFO operation (FCR0:FE1 = 0 or FCR0:FE2 = 0), reset the transmission FIFO (FCR0:FCL1 = 1 or FCR0:FCL2 = 1).
  - Then, execute transmission data register clear (LAMCR:LTDRCL = 1) to attain the transmission bus idle status.

Slave Operation Timing Chart

Figure 7-39 LIN Bus Timing (When the DATA Field Is Transmitted: a FIFO Is Not Used, SACSR:AUTE = 1, and an ID Register Is Used)

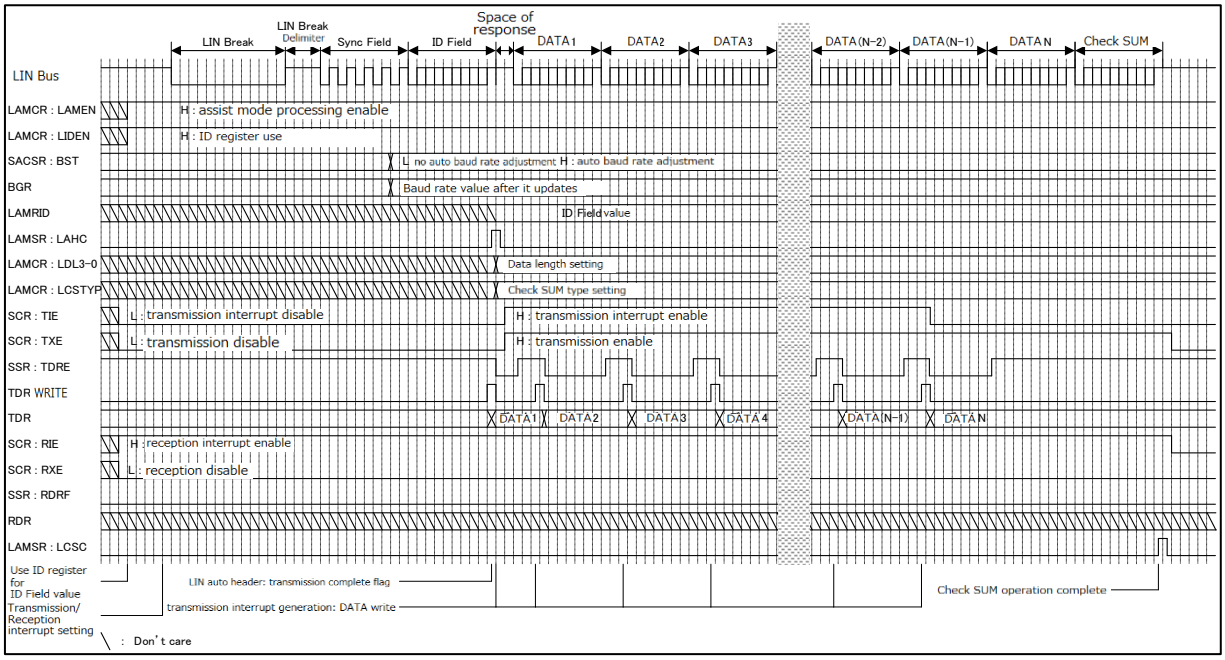


Figure 7-40 LIN Bus Timing (When the DATA Field Is Transmitted: a FIFO Is Not Used, SACSR:AUTE = 1, and the ID Register Is Not Used)

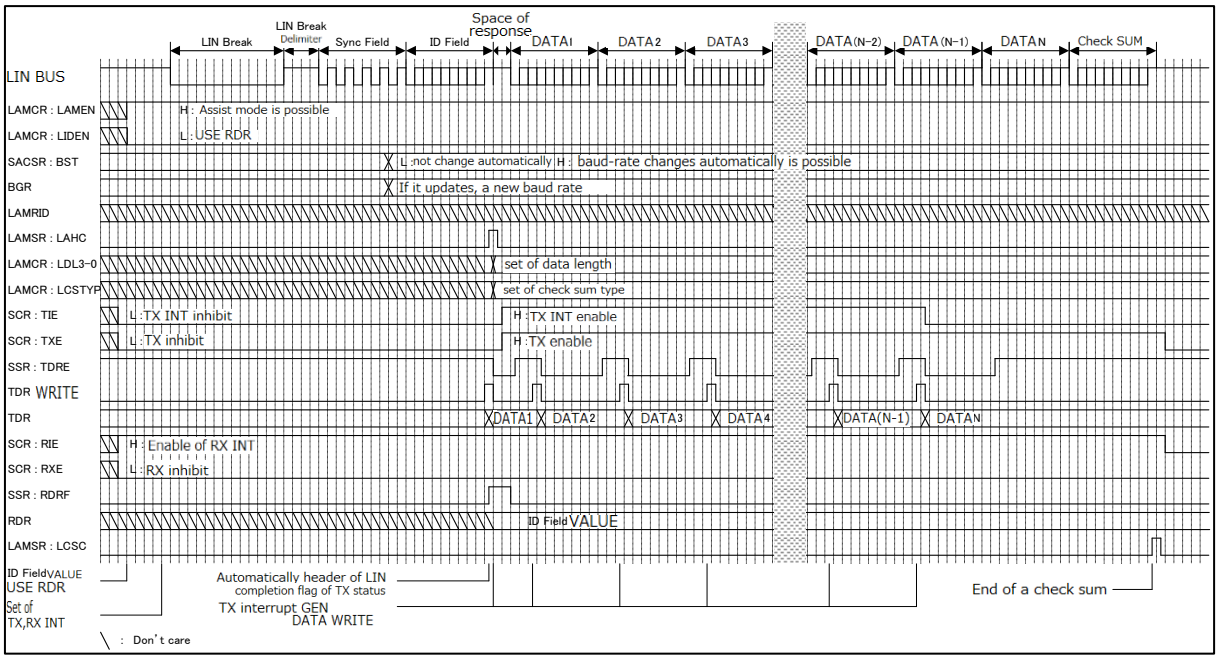


Figure 7-41 LIN Bus Timing (When the DATA Field Is Received: a FIFO Is Not Used, SACSR:AUTE = 1, and an ID Register Is Used)

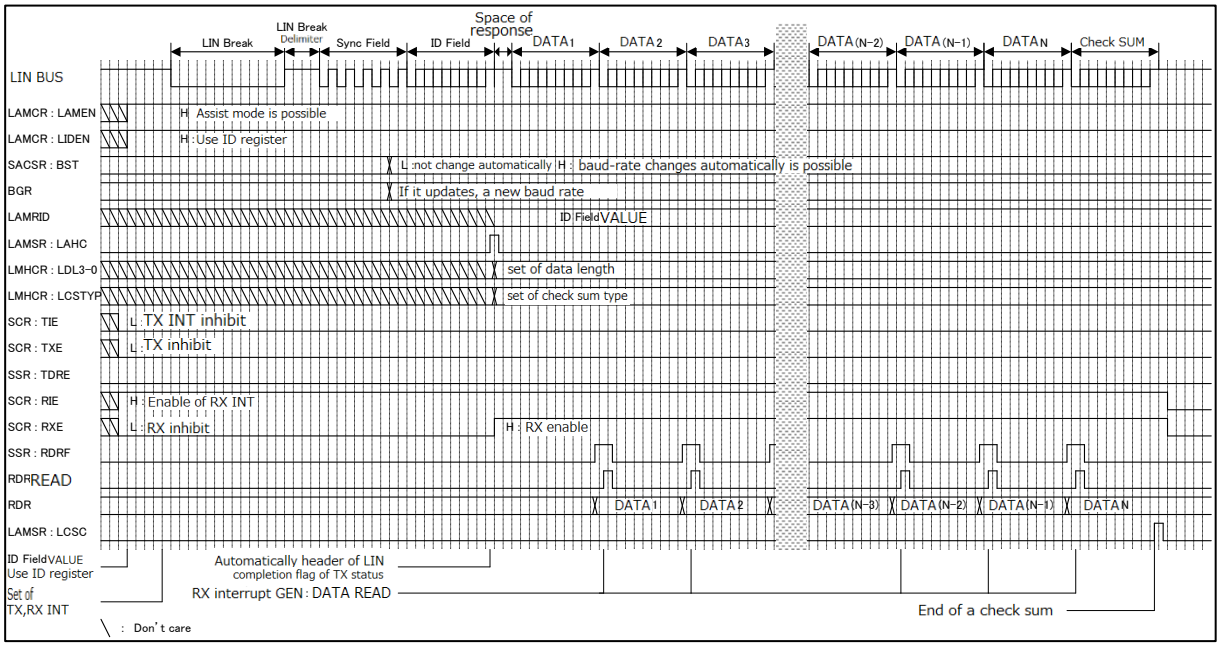


Figure 7-42 LIN Bus Timing (When the DATA Field Is Received: a FIFO Is Not Used, SACSR:AUTE = 1, and an ID Register Is Not Used)

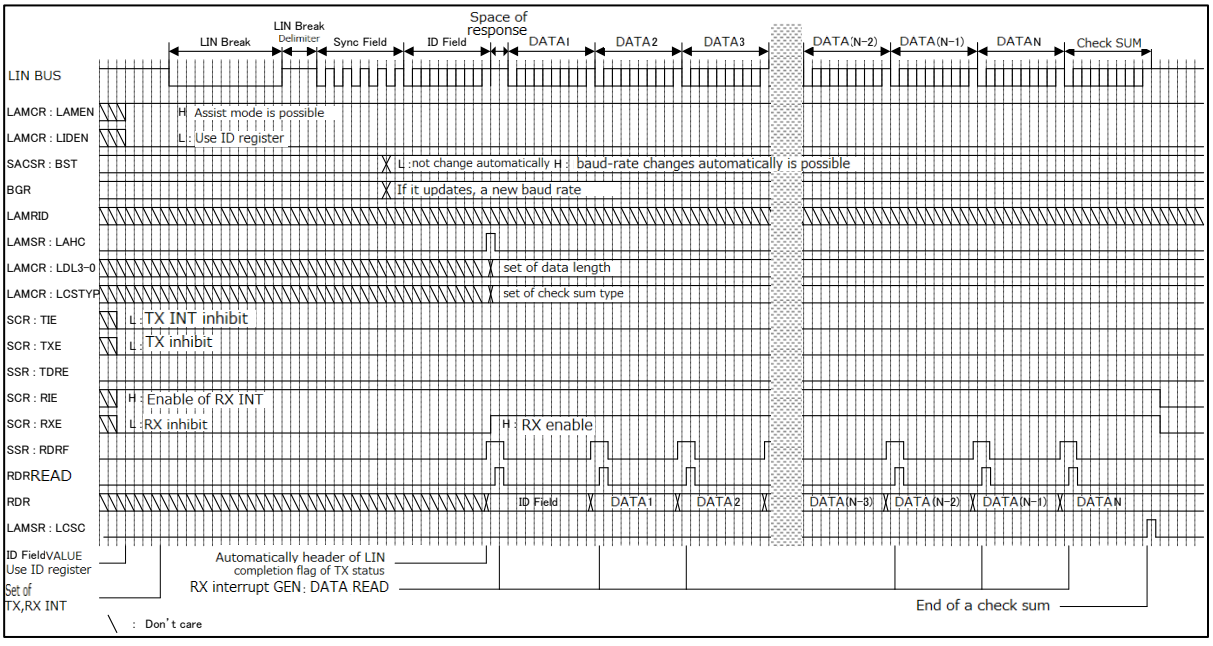


Figure 7-43 LIN Bus Timing (When the DATA Field Is Transmitted: a FIFO Is Used, SACSR:AUTE = 1, and the ID Register Is Used)

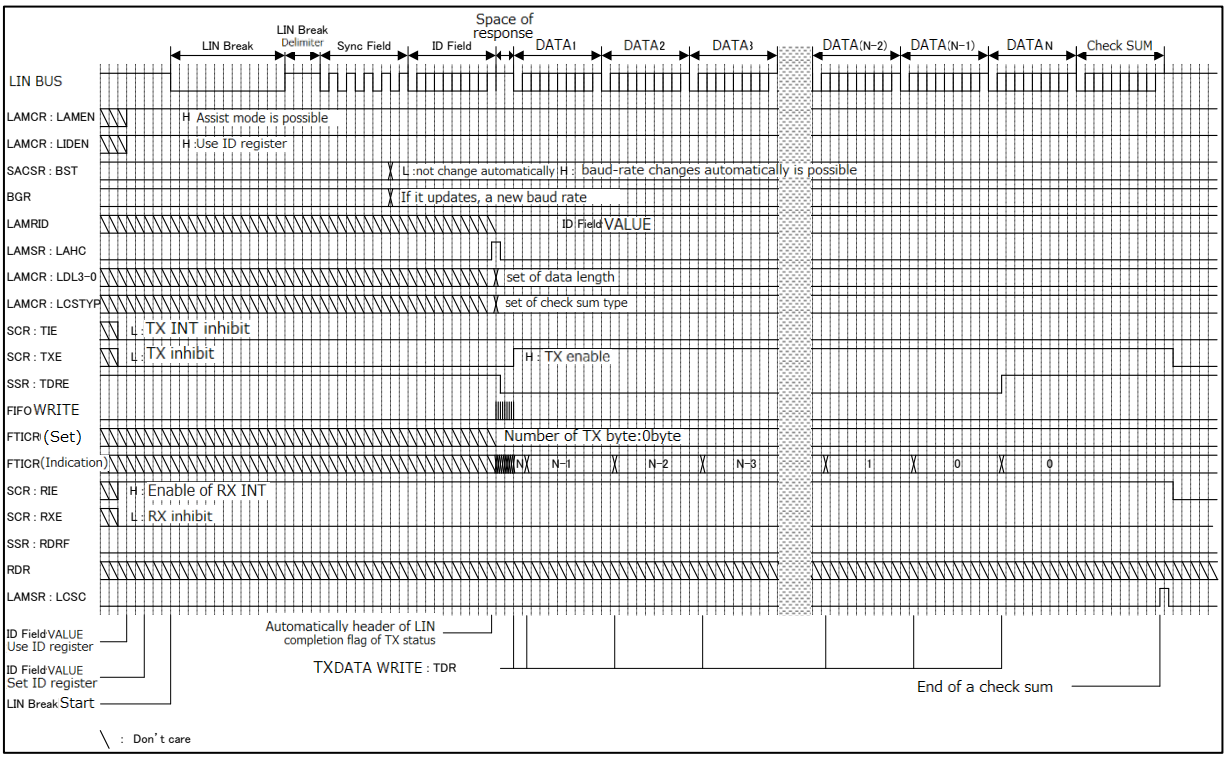


Figure 7-44 LIN Bus Timing (When the DATA Field Is Transmitted: a FIFO Is Used, SACSR:AUTE = 1, and the ID Register Is Not Used)

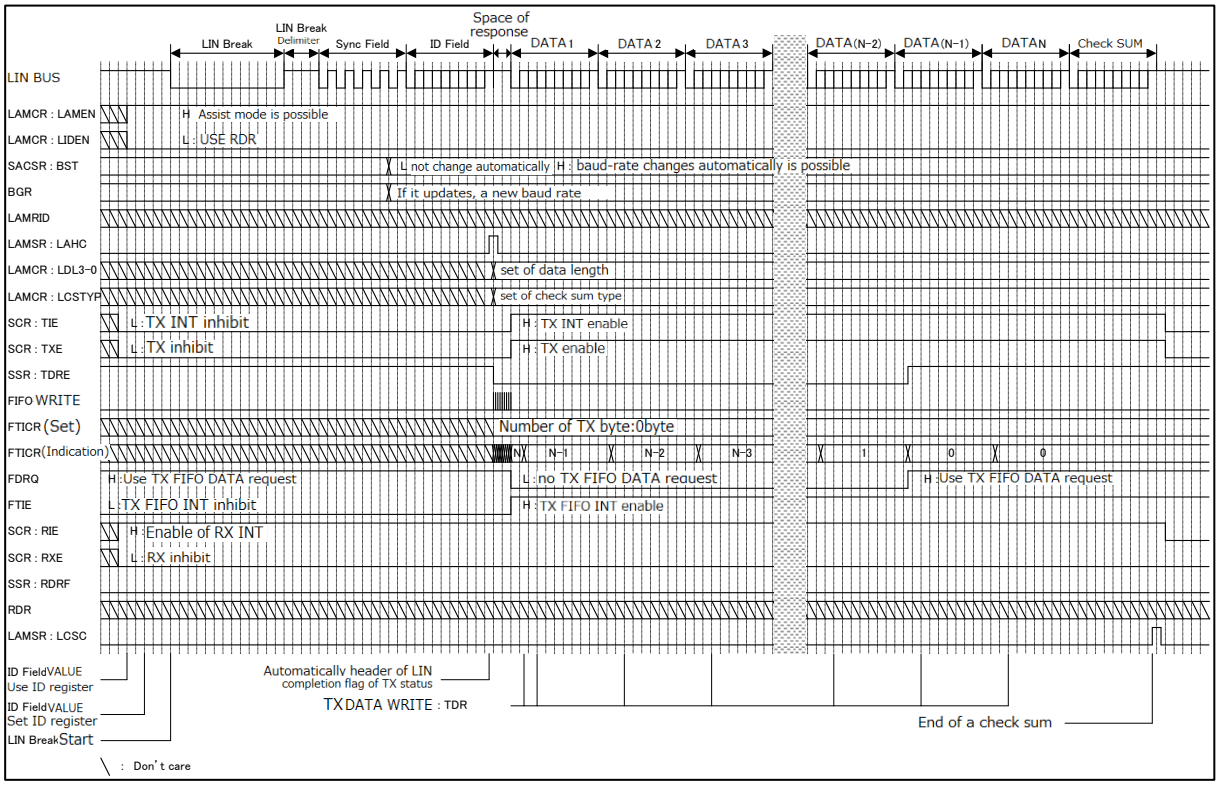


Figure 7-45 LIN Bus Timing (When the DATA Field Is Received: a FIFO Is Used, SACSR:AUTE = 1, and the ID Register Is Used)

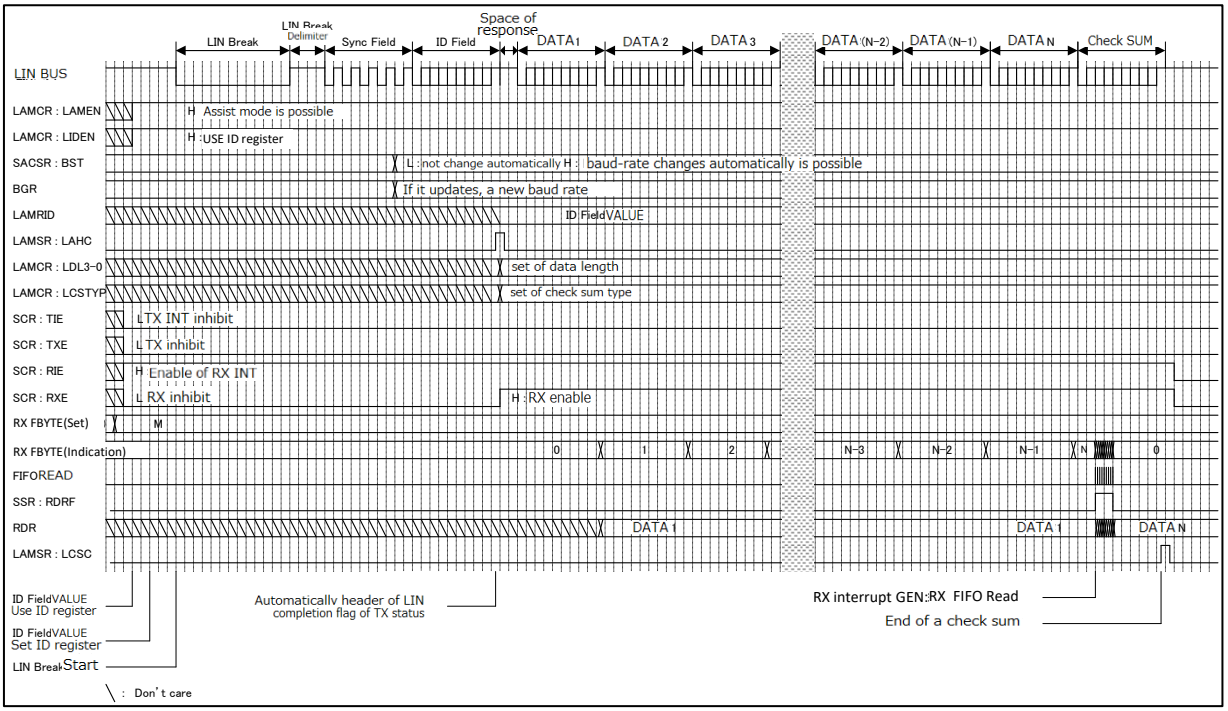
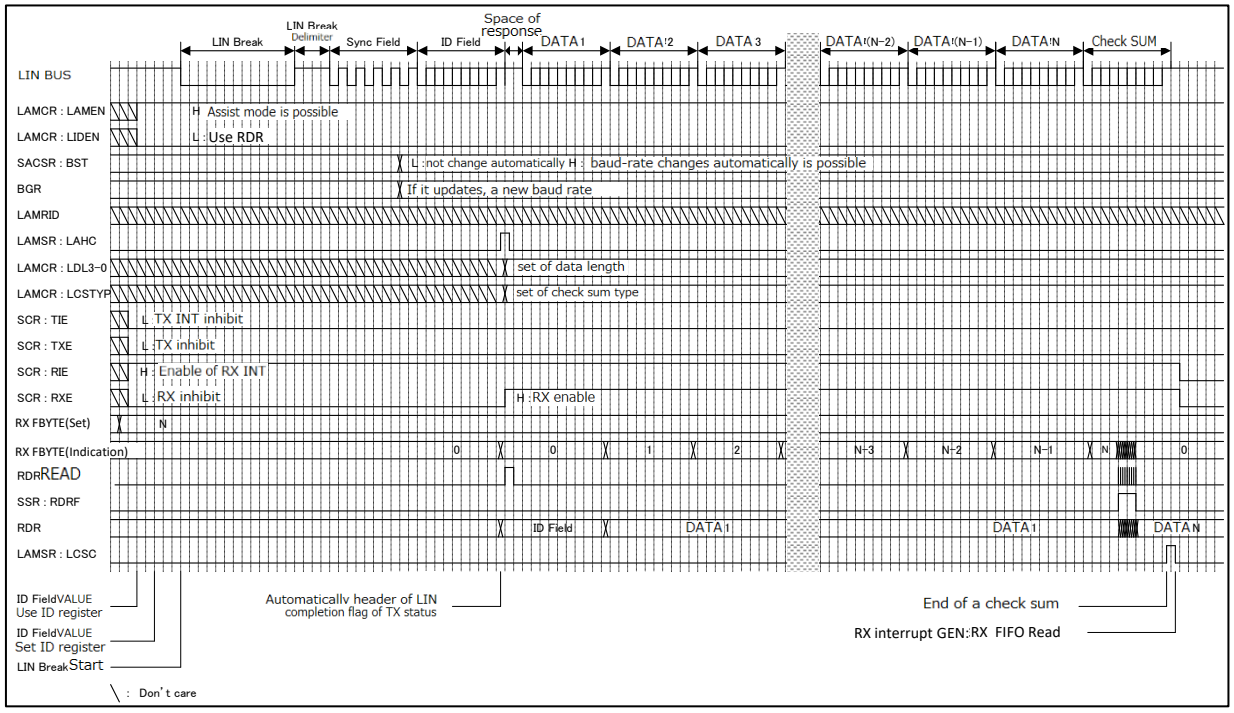


Figure 7-46 LIN Bus Timing (When the DATA Field Is Received: a FIFO Is Used, SACSR:AUTE = 1, and the ID Register Is Not Used)



8. Setup Procedure and Program Flow for Operation Mode 3 (LIN Communication Mode)

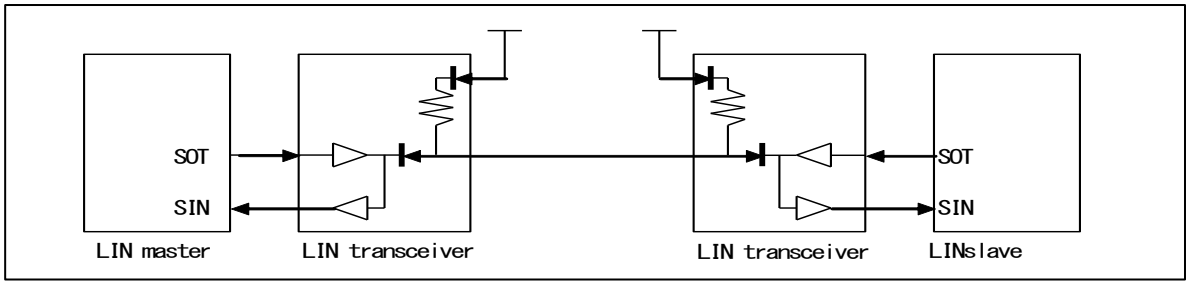
Operation mode 3 (LIN communication mode) can be used for either the LIN master system or the LIN slave system.

Register Setting

Connection between MCUs

Figure 8-1 illustrates the communication system between a LIN master and a LIN slave. The LIN interface (v2.1) can operate as either the LIN master or the LIN slave.

Figure 8-1 Example of LIN Bus System Communication



8.1. Manual Mode

This section provides an example of a flow chart for the master or slave side in manual mode.

Flow Chart Example

Master Operation

Figure 8-2 Example of LIN Communication Master Mode (a FIFO Is Not Used)

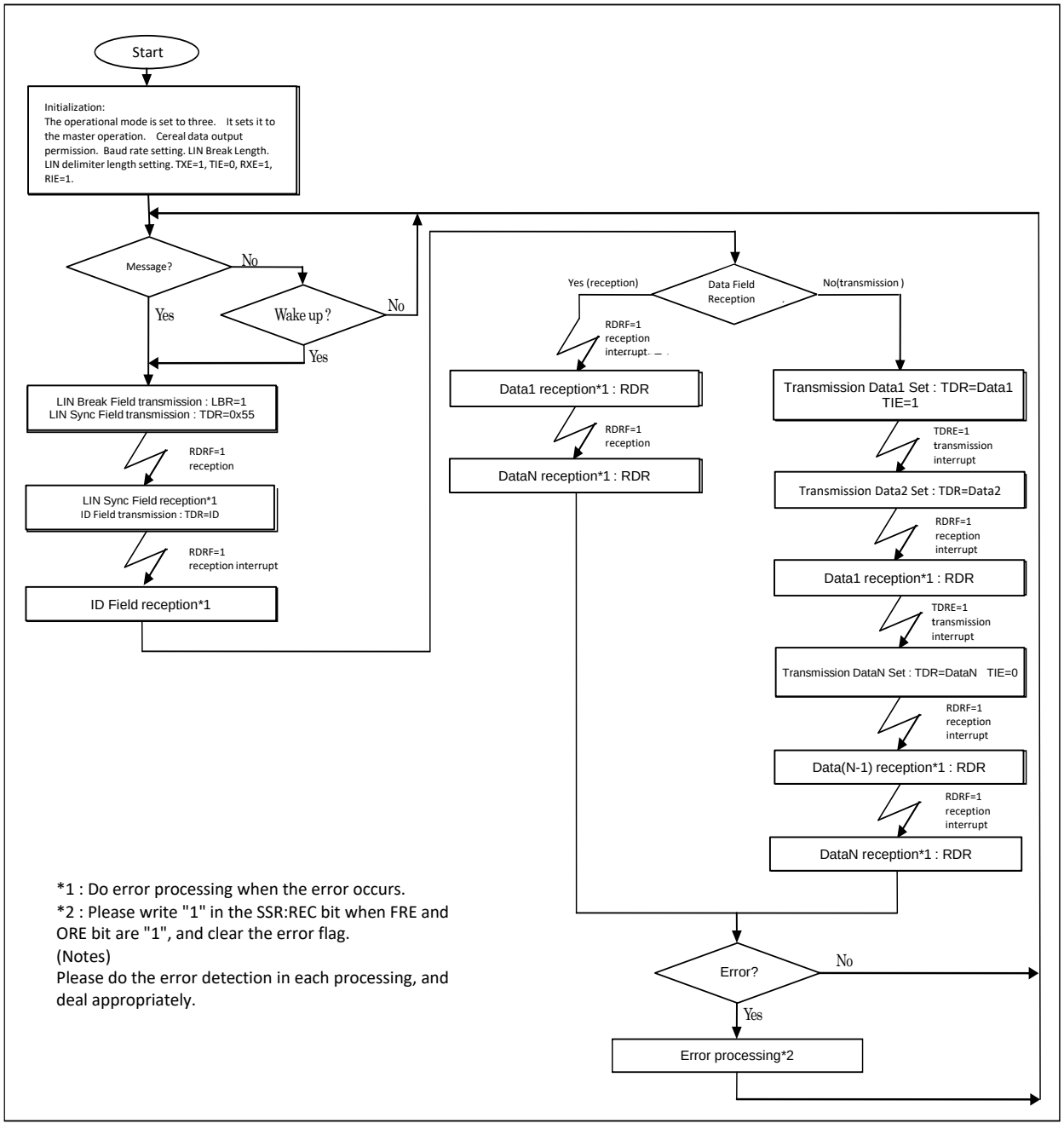
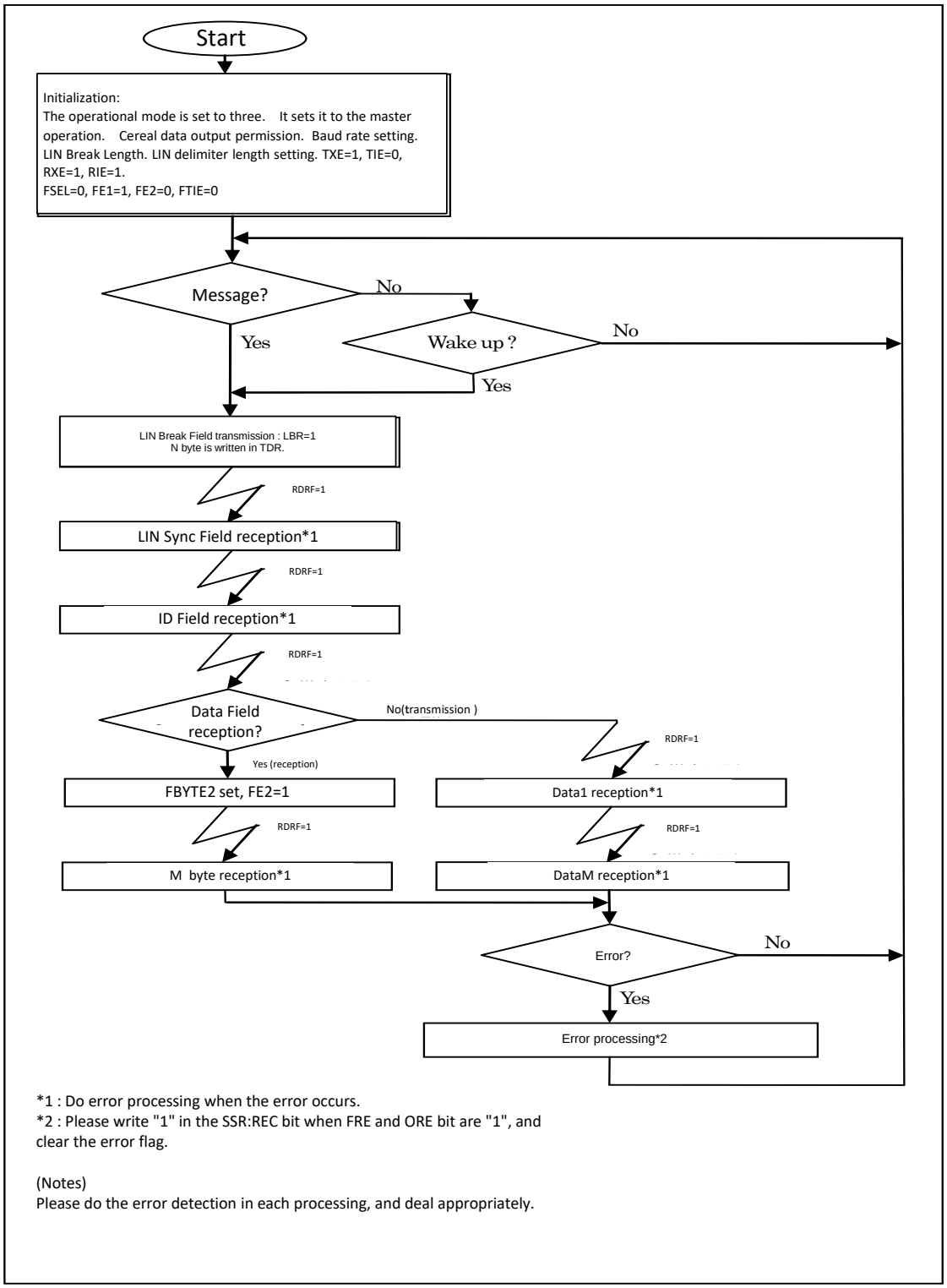


Figure 8-3 Example of LIN Communication Master Mode (a FIFO Is Used)

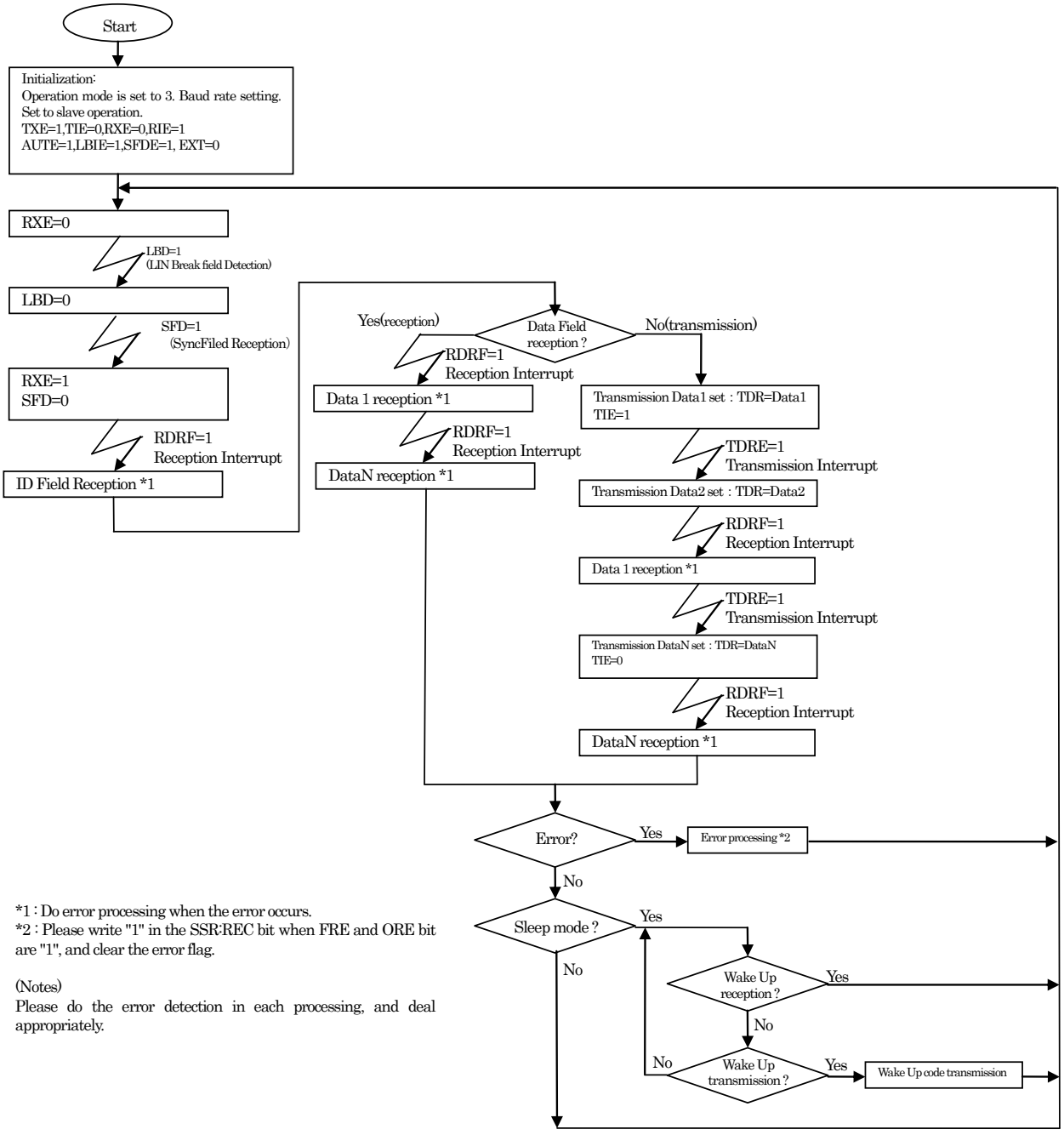


*1 : Do error processing when the error occurs.
*2 : Please write "1" in the SSR:REC bit when FRE and ORE bit are "1", and clear the error flag.

(Notes)
Please do the error detection in each processing, and deal appropriately.

Slave Operation

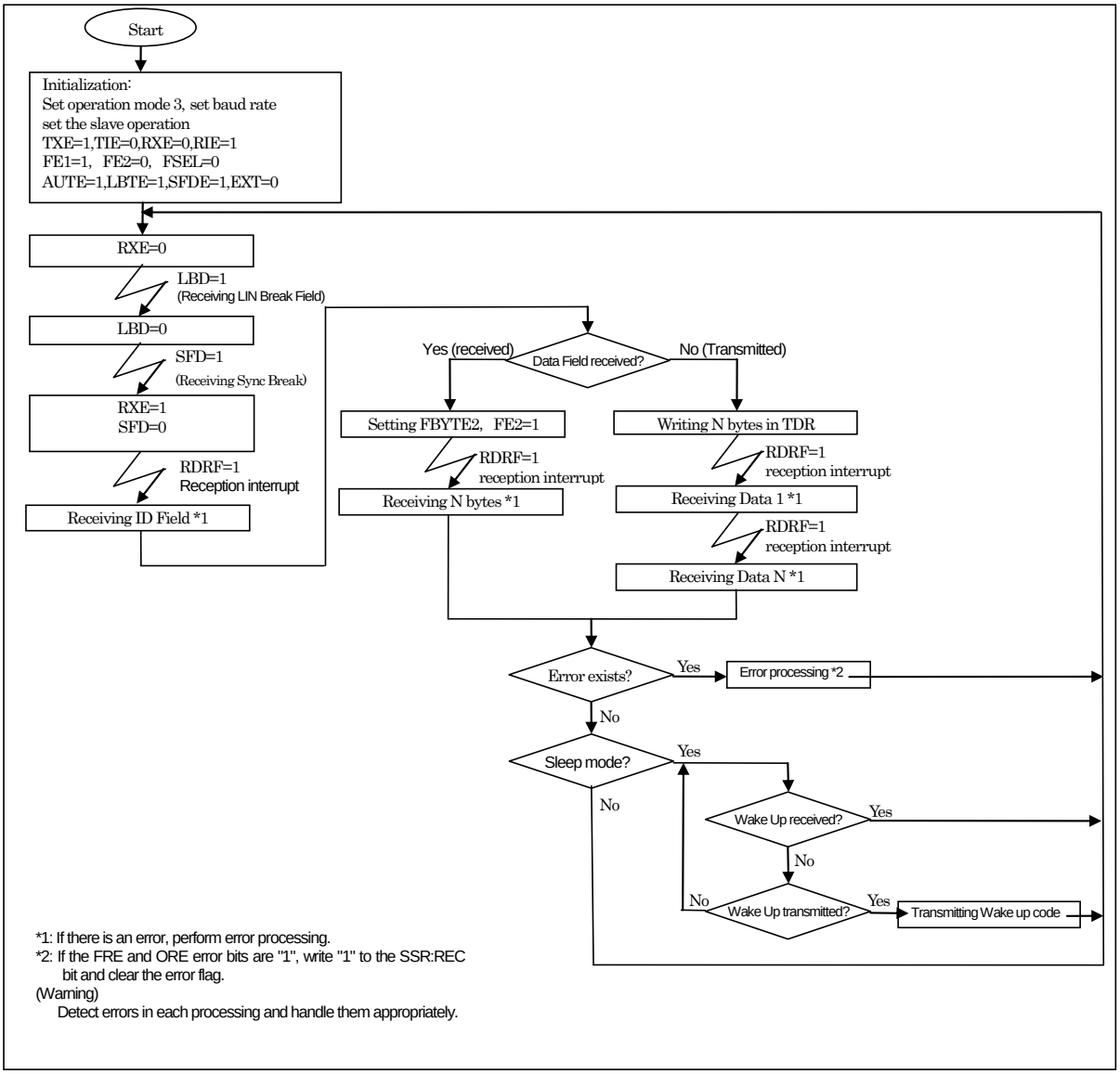
Figure 8-4 Example of LIN Communication Slave Mode (FIFO Is Not Used, Auto Baud Rate Adjustment is Allowed (SACSR:AUTE = 1))



*1 : Do error processing when the error occurs.
*2 : Please write "1" in the SSR:REC bit when FRE and ORE bit are "1", and clear the error flag.

(Notes)
Please do the error detection in each processing, and deal appropriately.

Figure 8-5 Example of LIN Communication Slave Mode (a FIFO Is Used, Auto Baud Rate Adjustment Allowed (SACSR:AUTE = 1))



8.2. Assist Mode

This section provides an example of a flow chart of the master or slave side in assist mode.

Flow Chart Example

Master Operation

Figure 8-6 Example of LIN Communication Master Mode (Assist Mode/a FIFO Is Not Used)

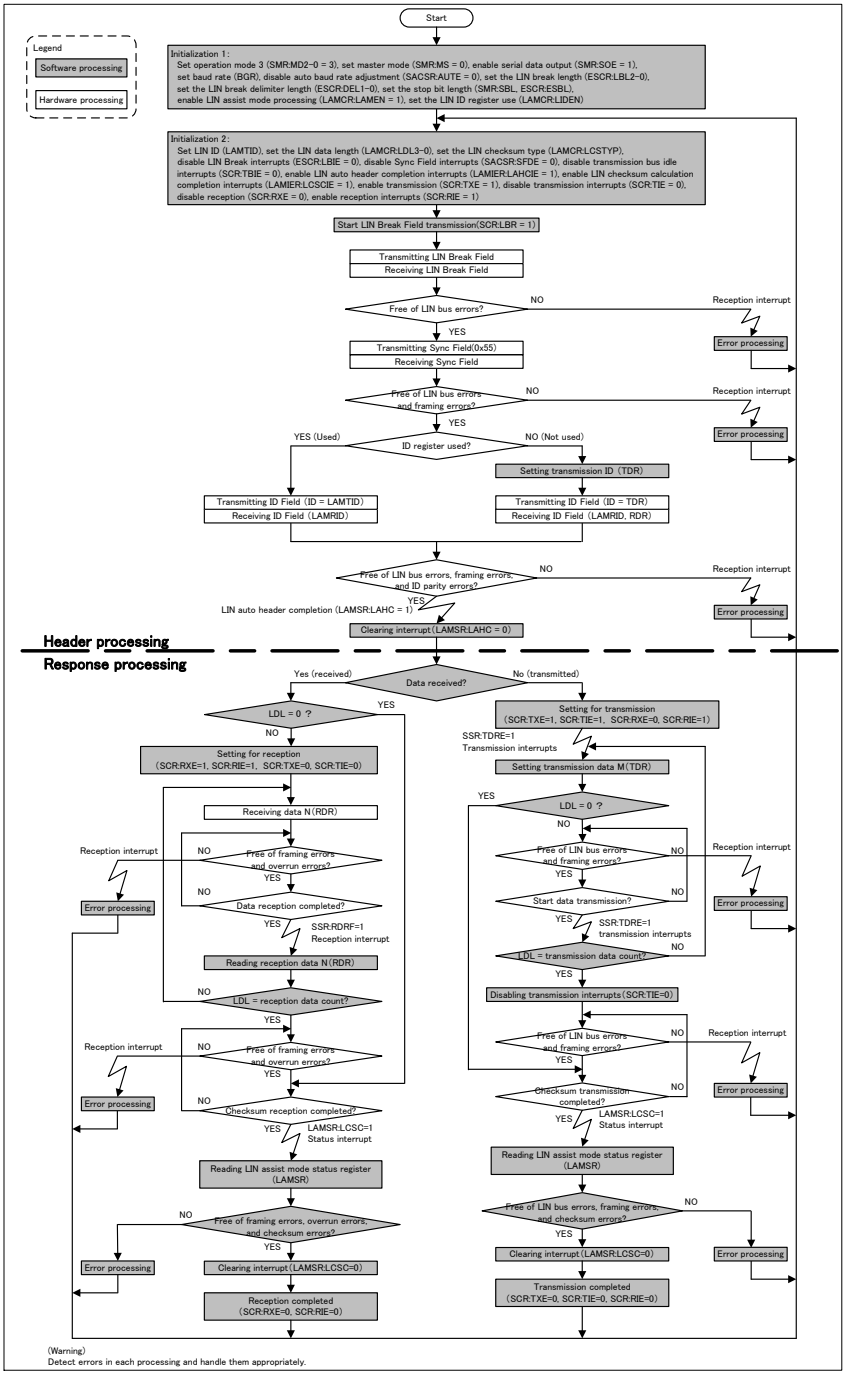
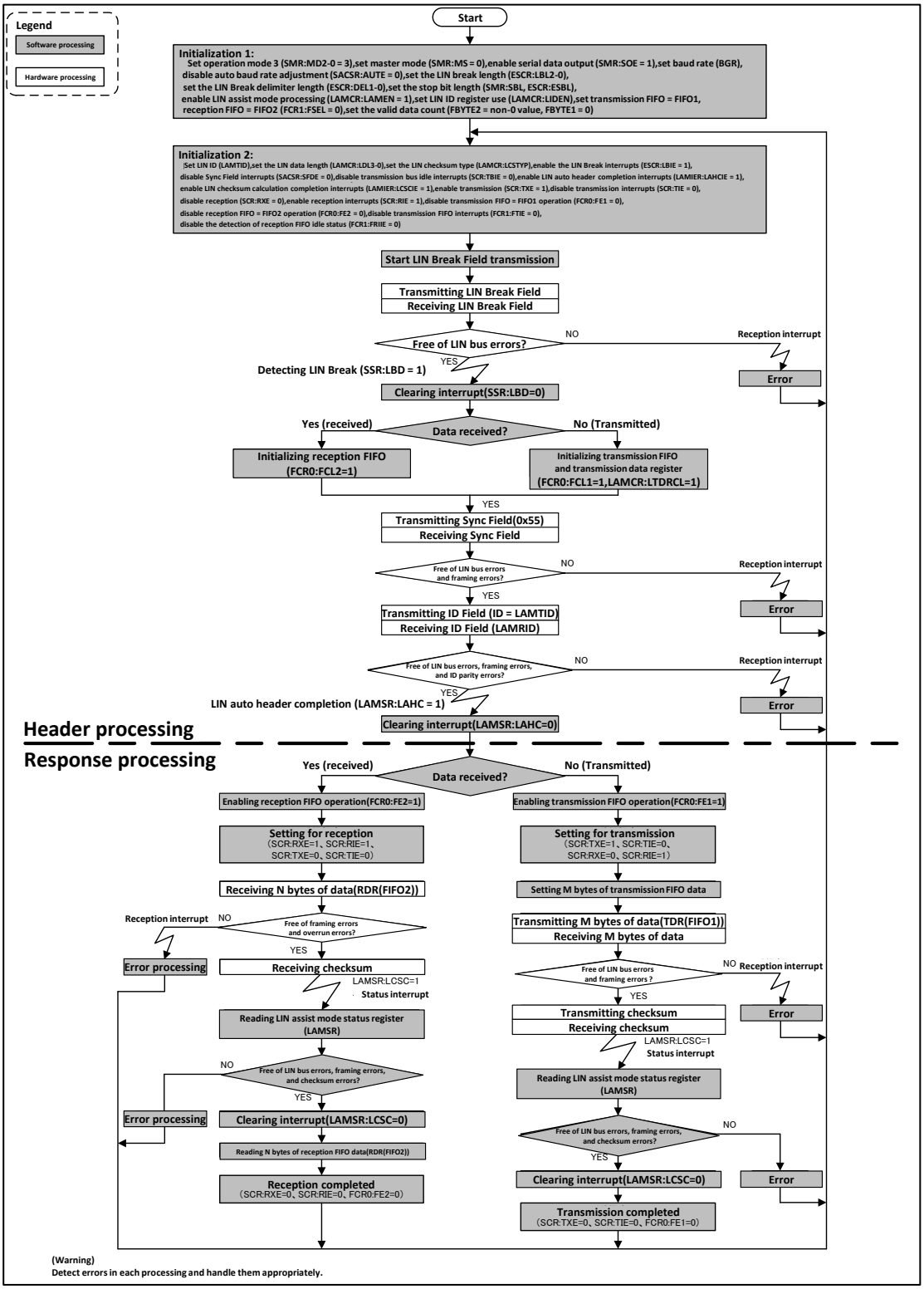


Figure 8-7 Example of LIN Communication Master Mode (Assist Mode/a FIFO Is Used)[*]



Slave Operation

Figure 8-8 Example of LIN Communication Slave Mode (Assist Mode/a FIFO Is Not Used)

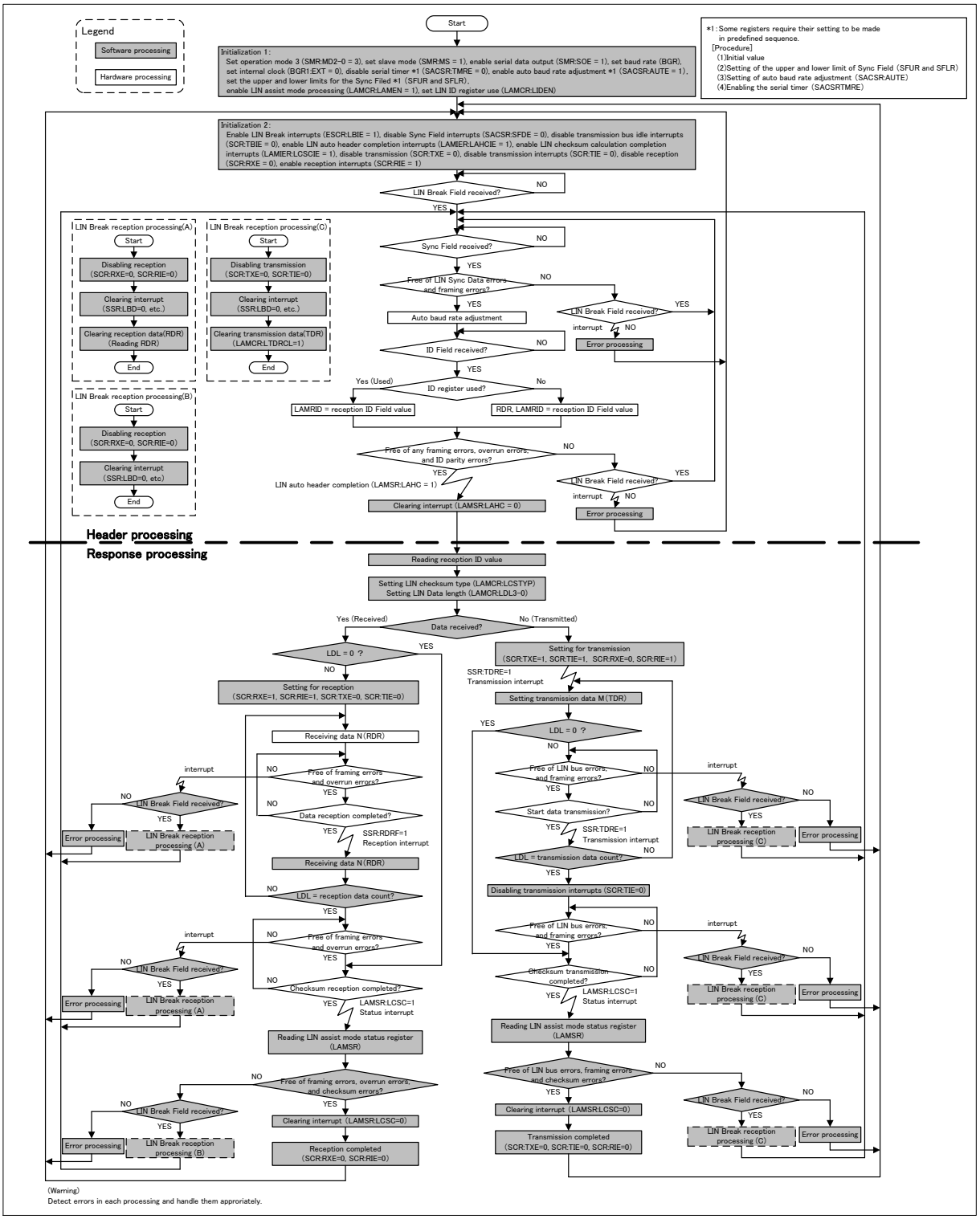
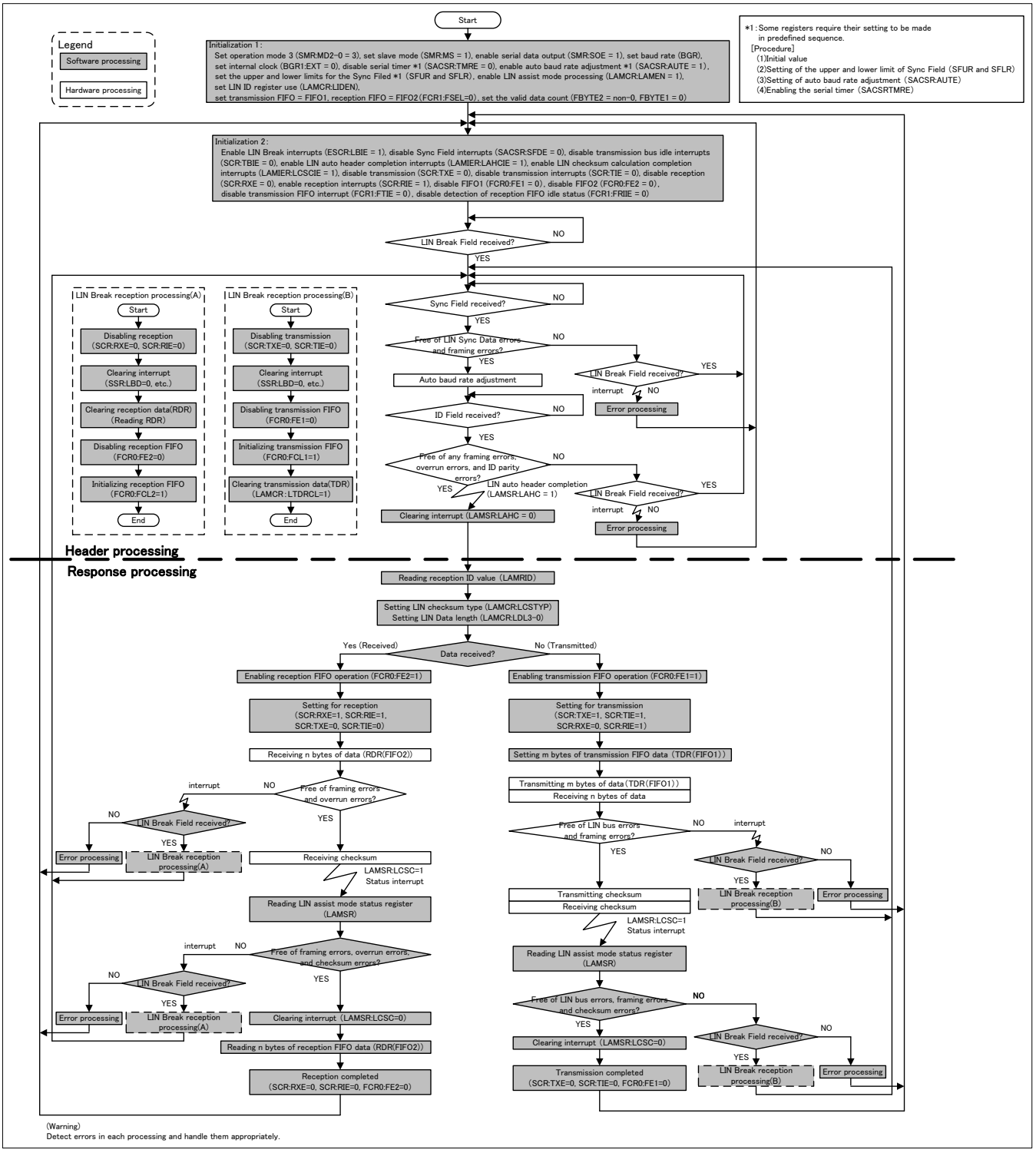


Figure 8-9 Example of LIN Communication Slave Mode (Assist Mode/a FIFO Is Used)



9. Interface (V2.1) Registers

This section provides a list of the registers of the LIN interface (v2.1).

List of Registers of the LIN Interface (v2.1)

Table 9-1 LIN Interface (V2.1) Registers

	bit15 bit8	bit7 bit0
LIN Interface (v2.1)	SCR (serial control register)	SMR (serial mode register)
	SSR (serial status register)	ESCR (extended communication control register)
	-	RDR0/TDR0 (reception data register/transfer data register)
	-	-
	SACSR1 (Serial auxiliary control status register 1)	SACSR0 (Serial auxiliary control status register 0)
	STMR1 (Serial timer register 1)	STMR0 (serial timer register 0)
	STMCR1 (serial timer comparison register 1)	STMCR0 (serial timer comparison register 0)
	SFUR1 (Sync Field upper limit register 1)	SFUR0 (Sync Field upper limit register 0)
	SFLR1 (Sync Field lower limit register 1)	SFLR0 (Sync Field lower limit register 0)
	LAMSR (LIN assist mode status register)	LAMCR (LIN assist mode control register)
	-	-
	-	-
	LAMIER (LIN assist mode interrupt enable register)	LAMTID/LAMRID (LIN assist mode transmission/reception ID register)
	LAMESR (LIN assist mode error status register)	LAMERT (LIN assist mode error test register)
	BGR1 (baud rate generator register 1)	BGR0 (baud rate generator register 0)
	-	-
FIFO	FCR1 (timer control register 1)	FCR0 (timer control register 0)
	FBYTE2 (FIFO2 byte register)	FBYTE1 (FIFO1 byte register)
	FTICR2 (transmission FIFO interrupt control register 2)	FTICR1 (transmission FIFO interrupt control register 1)
Auxiliary Register	ESR (extended status register)	ECR (extended control register)
	-	TBSIZE (transmission block size register)
	-	-
Clear Registers	SCRC (serial control clear register)	SMRC (serial mode clear register)
	SSRC (serial status clear register)	ESCRC (extended communication control clear register)
	SACSR1C (serial auxiliary control status clear register 1)	SACSR0C (serial auxiliary control status clear register 0)
	-	-
	-	-
	LAMSRC (LIN assist mode status clear register)	LAMCRC (LIN assist mode control clear register)
	LAMIERC (LIN assist mode interrupt enable clear register)	-
	LAMESRC (LIN assist mode error status clear register)	-
	FCR1C (FIFO control clear register 1)	FCR0C (FIFO control clear register 0)
	-	-
	-	-
	ESRC (extended status clear register)	-
Set	SCRS (serial control set register)	SMRS (serial mode set register)

	bit15 bit8	bit7 bit0
Registers	SSRS (serial status set register)	ESCRS (extended communication control set register)
	SACSR1S (serial auxiliary control status set register 1)	SACSR0S (serial auxiliary control status set register 0)
	-	-
	-	-
	-	LAMCRS (LIN assist mode control set register)
	LAMIERS (LIN assist mode interrupt enable set register)	-
	-	-
	FCR1S (FIFO control set register 1)	FCR0S (FIFO control set register 0)
	-	-
Transmission and reception data register	-	RDR0/TDR0 (reception data register/ transmission data register)*1
	-	-
	-	RDR0/TDR0 (reception data register/ transmission data register)*1
	-	-
	-	RDR0/TDR0 (reception data register/ transmission data register)*1
	-	-
	-	RDR0/TDR0 (reception data register/ transmission data register)*1
	-	-
	-	RDR0/TDR0 (reception data register/ transmission data register)*1
	-	-
	-	RDR0/TDR0 (reception data register/ transmission data register)*1
	-	-
	-	RDR0/TDR0 (reception data register/ transmission data register)*1
	-	-
	-	RDR0/TDR0 (reception data register/ transmission data register)*1
	-	-
	-	RDR0/TDR0 (reception data register/ transmission data register)*1
	-	-
	-	RDR0/TDR0 (reception data register/ transmission data register)*1
	-	-
	-	RDR0/TDR0 (reception data register/ transmission data register)*1
	-	-
	-	RDR0/TDR0 (reception data register/ transmission data register)*1
	-	-
	-	RDR0/TDR0 (reception data register/ transmission data register)*1
Transmission and reception data	-	RDR0/TDR0 (reception data register/ transmission data register)*1

	bit15	bit8	bit7	bit0
register	-		-	
	-		RDR0/TDR0 (reception data register/ transmission data register)*1	
	-		-	
	-		RDR0/TDR0 (reception data register/ transmission data register)*1	
	-		-	
	-		RDR0/TDR0 (reception data register/ transmission data register)*1	
	-		-	
	-		RDR0/TDR0 (reception data register/ transmission data register)*1	
	-		-	

*1: Access from the mirror address area

9.1. Serial Control Register (SCR)

The serial control register (SCR) is used to enable/disable transmission/reception interrupts, enable/disable transmission idle interrupts, and enable/disable transmission/reception operations. This register also includes the settings for LIN Break Field generation and LIN interface (v2.1) reset.

Serial Control Register (SCR)

Figure 9-1 shows the bit configuration of the serial control register (SCR).

Figure 9-1 Bit Configuration of Serial Control Register (SCR)

Bit	15	14	13	12	11	10	9	8	7 ... 0
Field	UPCL	MS	LBR	RIE	TIE	TBIE	RXE	TXE	(SMR)
R/W Attribute	R0,W	R/W	R0,W	R/W	R/W	R/W	R/W	R/W	
Protection Attribute	-	-	-	-	-	-	-	-	
Initial Value	0	0	0	0	0	0	0	0	

[bit15] UPCL: Programmable Clear Bit

This bit initializes the internal status of the LIN interface (v2.1).

- This bit is set when the SCRS:UPCLS bit in the set register is set to "1". [*]

When "1" is set:

- The LIN interface (v2.1) is directly reset (software reset). However, the register settings are retained. At this time, transmission and reception communications are immediately disconnected.
- The baud rate generator reloads the setting values of the BGR1/0 registers and then restarts.
- All transmission/reception and status interrupt factors (SSR:TDRE, TBI, RDRF, FRE, ORE, LBD, SACSR: TINT, SFD, LAMSR:LER, SER, RDRF, TDRE, TBI, LCSC, LAHC, LAMESR:LCSER, LPTER, LSFER, and LBSER) are initialized.
- The baud rate setting flag (SACSR:BST) is initialized.

When "0" is set:

There is no effect.

When it is read, "0" is always read.

Bit	Programmable Clear Bit	
	Write	Read
0	No effect	"0" is always read.
1	Programmable clear	

Notes:

- *Execute programmable clear after setting interrupt prohibition.*
- *When a FIFO is used, prohibit the FIFO (FCR0:FE2, FE1 = 0) before executing programmable clear.*
- *Programmable clear does not clear the transmission/reception FIFOs.*
- *Executing programmable clear (SCR:UPCL = 1) does not clear the value of the serial timer register (STMR).*

[bit14] MS: Master/Slave Function Select Bit

This bit selects either master or slave mode.

- This bit is reset when the SCRC:MSC bit in the clear register is set to "1". [*]

■ This bit is set when the SCRS:MSS bit in the set register is set to "1". [*]

When set to "0": The system is placed in master mode.
When set to "1": The system is placed in slave mode.

Bit	Master/Slave Function Select Bit
0	Master mode
1	Slave mode

[bit13] LBR: LIN Break Field Setting Bit (Effective Only for Master Operation)

■ In LIN manual mode operation (LAMCR:LAMEN = 0):
- Setting this bit to "1" generates the LIN Break Field and LIN Break delimiter in the lengths specified by the ESCR:LBL1/0 and ESCR:DEL1/0 bits.

■ In LIN assist mode operation (LAMCR:LAMEN = 1):
- Setting this bit to "1" generates the LIN Break Field and LIN Break delimiter in the lengths specified by the ESCR:LBL2/1/0 and ESCR:DEL1/0 bits, and then transmits the Sync Field and the ID Field.

■ This bit is set when the SCRS:LBRS bit in the set register is set to "1".

When writing:

- Writing "0": No effect
  Writing "1":
- In LIN manual mode operation (LAMCR: LAMEN = 0),
- a LIN Break Field is generated.
  In LIN assist mode operation (LAMCR: LAMEN = 1),
  a LIN Break Field is generated, and a Sync Field and an ID Field are transmitted.

When reading:

"0" is always read.

Bit	LIN Break Field Setting Bit	
	Write	Read
0	No effect	"0" always read
1	In LIN manual mode, a LIN Break Field is generated. In LIN assist mode, the LIN Break Field to ID Field are transmitted.	

Notes:
- *This bit is effective only for master operation (MS = 0).*
- *This bit must not be set to "1" when a header is being transmitted or when a response is being transmitted or received.*
- *If an attempt is made to perform programmable clear (writing UPCL = 1) and LBR setting (writing LBR= 1) simultaneously, the programmable clear takes precedence.*

[bit12] RIE: Reception Interrupt Enable Bit

■ This bit enables/disables reception interrupt request output to the CPU.

- For ECR:EISEL = 0, a reception interrupt request is issued when the RIE bit and the reception data flag bit (SSR:RDRF) are "1" or when one of the error flag bits (SSR:FRE, ORE, ESR:RXUDR, RBERR) is "1".
- For ECR:EISEL = 1, a reception interrupt request is issued when the RIE bit and the reception data flag bit (SSR:RDRF) are "1".
- This bit is reset when the SCRC:RIEC bit in the clear register is set to "1".
- This bit is set when the SCRS:RIES bit in the set register is set to "1".

Bit	Reception Interrupt Enable Bit
0	Disable reception interrupts.
1	Enable reception interrupts.

[bit11] TIE: Transmission Interrupt Enable Bit

- This bit enables/disables transmission interrupt request output to the CPU.
- For ECR:EISEL = 0, a transmission interrupt request is issued when the TIE bit and the SSR:TDRE bit are set to "1" or when one of the error flag bits (ESR:TXOVR, TBERR) is set to "1".
- For ECR:EISEL = 1, a transmission interrupt request is issued when the TIE bit and the SSR:TDRE bit are "1".
- This bit is reset when the SCRC:TIEC bit in the clear register is set to "1".
- This bit is set when the SCRS:TIES bit in the set register is set to "1".

Bit	Transmission Interrupt Enable Bit
0	Disable transmission interrupts.
1	Enable transmission interrupts.

[bit10] TBIE: Transmission Bus Idle Interrupt Enable Bit

- This bit enables/disables transmission bus idle interrupt request output to the CPU.
- A transmission bus idle interrupt request is output when the TBIE bit and the SSR:TBI bit are "1".
- This bit is reset when the SCRC:TBIEC bit in the clear register is set to "1"
- This bit is set when the SCRS:TBIES bit in the set register is set to "1".

Bit	Transmission Bus Idle Interrupt Enable Bit
0	Disable transmission bus idle interrupts.
1	Enable transmission bus idle interrupts.

[bit9] RXE: Reception Operation Enable Bit

This bit controls enable/disable of the receiving operation of the LIN interface (v2.1).

- This bit is reset when the SCRC:RXEC bit in the clear register is set to "1".
- This bit is set when the SCRS:RXES bit in the set register is set to "1".

- When set to "0": Data frame reception operation is disabled.
- When set to "1": Data frame reception operation is enabled.

Bit	Reception Operation Enable Bit
0	Disable reception.
1	Enable reception.

Notes:

- *Even if reception operation is enabled (RXE = 1), reception operation does not start unless a falling edge of a start bit is input.*
- *In master operation, even if reception operation is enabled (RXE = 1), data is not received while LIN Break Field is being transmitted.*
- *In manual mode (LAMCR:LAMEN = 0), disabling reception operation (RXE = 0) during reception immediately stops reception operation.*
- *In assist mode (LAMCR:LAMEN = 1), disable reception (RXE = 0) during transmission or reception of a header and during response transmission.*
- *In assist mode (LAMCR:LAMEN = 1), even if reception operation is disabled (RXE = 0) during header reception, header reception operation does not stop. To stop the reception, disable the reception (RXE = 0) and set manual mode (LAMCR:LAMEN = 0).*
- *In assist mode (LAMCR:LAMEN = 1), disabling reception operation (RXE = 0) during reception of a response immediately stops reception operation.*
- *Framing error detection (SSR:FRE = 1) is enabled when a LIN Break Field is received and reception is enabled (RXE = 1).*

[bit8] TXE: Transmission Operation Enable Bit

This bit enables/disables transmission by the LIN interface (v2.1).

- This bit is reset when the SCRC:TXEC bit in the clear register is set to "1".
- This bit is set when the SCRS:TXES bit in the set register is set to "1".

- When set to "0": Data frame transmission operation is disabled.
- When set to "1": Data frame transmission operation is enabled.

Bit	Transmission Operation Enable Bit
0	Disable transmission.
1	Enable transmission.

Notes:

- *In manual mode (LAMCR:LAMEN = 0), disabling transmission operation (TXE = 0) during transmission immediately stops transmission operation.*
- *In assist mode (LAMCR:LAMEN = 1), even if transmission operation is disabled (TXE = 0) during header transmission, header transmission operation does not stop. To stop the transmission, disable the transmission (TXE = 0) and set manual mode (LAMCR:LAMEN = 0).*
- *In assist mode (LAMCR:LAMEN = 1), disabling transmission operation (TXE = 0) during transmission of a response immediately stops transmission operation.*

9.2. Serial Mode Register (SMR)

The serial mode register (SMR) is responsible for the following: 1. Operation mode setting, 2. Selection of the transfer direction, the data length, and the stop bit length, and 3. Setting of enabling/disabling of output on the serial data and clock pins.

Serial Mode Register (SMR)

Figure 9-2 shows the bit configuration of the serial mode register (SMR).

Figure 9-2 Bit Configuration of Serial Mode Register (SMR)

Bit	15 ... 8	7	6	5	4	3	2	1	0
Field	(SCR)	MD2	MD1	MD0	WUCR	SBL	Reserved	Reserved	SOE
R/W Attribute		R/W	R/W	R/W	R/W	R/W	R0,W0	R0,W0	R/W
Protection Attribute		-	-	-	-	-	-	-	-
Initial Value		0	0	0	0	0	0	0	0

[bit7:5] MD[2:0]: Operation Mode Setting Bits

These bits set the operation mode.

Bits			Operation Mode Setting Bit
0	0	0	Operation mode 0 (asynchronous normal mode)
0	0	1	Operation mode 1 (asynchronous multi-processor mode)
0	1	0	Operation mode 2 (clock synchronous mode)
0	1	1	Operation mode 3 (LIN communication mode)
1	0	0	Operation mode 4 (I2C mode)

* This section describes the register and its operations in operation mode 3.

Notes:

- *Settings other than those described above are prohibited.*
- *To switch the operation mode, execute programmable clear (SCR:UPCL = 1) and then switch the operation mode.*
- *Set the registers only after setting an operation mode.*

[bit4] WUCR: WAKE UP Control Bit

WUCR is not supported

This bit selects a pin used for external interrupts.

- This bit is reset when the SMRC:WUCRC bit in the clear register is set to "1".
- This bit is set when the SMRS:WUCRS bit in the set register is set to "1".

"0": Set the INT pin for external interrupts.

"1": Set the SIN pin for external interrupts.

Bit	WAKE UP Control Bit
0	Disable the WAKE UP function.
1	Enable the WAKE UP function.

Note:

- *The WAKE UP function is not supported.*

[bit3] SBL: Stop Bit Length Selection Bit

This bit sets the length of the stop bits (frame end mark of the transmission data).

- This bit is reset when the SMRC:SBLC bit in the clear register is set to "1".
- This bit is set when the SMRS:SBLS bit in the set register is set to "1".

Settings where SBL is "0" and ESCR:ESBL is "0": The stop bit is set to 1 bit.
Settings where SBL is "1" and ESCR:ESBL is "0": The stop bit is set to 2 bits.
Settings where SBL is "0" and ESCR:ESBL is "1": The stop bit is set to 3 bits.
Settings where SBL is "1" and ESCR:ESBL is "1": The stop bit is set to 4 bits.

Bit	Stop Bit Length Selection Bit	
0	ESCR.ESBL = 0	1 bit
	ESCR.ESBL = 1	3 bits
1	ESCR.ESBL = 0	2 bits
	ESCR.ESBL = 1	4 bits

Notes:

- *Reception always detects only the first stop bit.*
- *Set this bit when transmission is disabled (SCR:TXE = 0).*

[bit2:1] Reserved: Reserved Bits

[bit0] SOE: Serial Data Output Enable Bit

This bit enables/disables the output of serial data.

- This bit is reset when the SMRC:SOEC bit in the clear register is set to "1".
- This bit is set when the SMRS:SOES bit in the set register is set to "1".

Bit	Serial Data Output Enable Bit
0	Serial data output disable
1	Serial data output enable

9.3. Serial Status Register (SSR)

The serial status register (SSR) checks the transmission and reception status, checks the reception error flag, detects the LIN Break Field, or clears the reception error flag.

Serial Status Register (SSR)

Figure 9-3 shows the bit configuration of the serial status register (SSR).

Figure 9-3 Bit Configuration of Serial Status Register (SSR)

Bit	15	14	13	12	11	10	9	8	7 ... 0
Field	REC	Reserved	LBD	FRE	ORE	RDRF	TDRE	TBI	(ESCR)
R/W Attribute	R0,W	R0,W0	R, WX	R, WX	R, WX	R, WX	R, WX	R, WX	
Protection Attribute	-	-	-	-	-	-	-	-	
Initial Value	0	0	0	0	0	0	1	1	

[bit15] REC: Reception Error Flag Clear Bit

This bit clears the FRE and ORE bits in the serial status register (SSR) and the SER flag in the LIN assist mode status register (LAMSR).

- This bit is set when the SSRS:RECS bit in the set register is set to "1".

- Writing "1" clears the error flag.
- Writing "0" does not have any effect.

If it is read, "0" is always read.

Bit	Reception Error Flag Clear Bit	
	Write	Read
0	No effect	"0" is always read.
1	Clear the reception error flags (FRE, ORE, and LAMSR:SER).	

[bit14] Reserved: Reserved Bit

[bit13] LBD: LIN Break Field Detection Flag Bit

This bit indicates LIN Break Field detection.

The LBD bit is set to "1" when the serial input (SIN) has "L" input for an interval of 11 bits or more. At this time, a status interrupt occurs if the LIN Break Field interrupt enable bit (LBIE) is set to "1".

- This bit is reset when the SSRC:LBDC bit in the clear register is set to "1".

(When reading)

"1": Detect a LIN Break Field.
"0": Do not detect a LIN Break Field.

(When writing)

Writing to this bit has no effect.

Bit	LIN Break Field Flag Bit	
	Write	Read
0	No effect	No LIN Break Field detected
1	No effect	LIN Break Field detected

Note:

- *This bit is set "0" when Software reset (SCR:UPCL=1) is done.*

[bit12] FRE: Framing Error Flag Bit

- This bit is set to "1" when a framing error occurs during reception. This bit is cleared by writing "1" to the REC bit in the serial status register (SSR).
- A reception interrupt request is issued when the FRE bit and the RIE bit are "1".
- For details on interrupt request output, see Table 2-3.
- If this flag is set, the reception data register (RDR) data will be invalid.
- If this flag is set when a reception FIFO is used, the reception FIFO enable bits are cleared, and no reception data is stored to the reception FIFO.

Bit	Framing Error Flag Bit
0	No framing error
1	Framing error found

Notes:

- *During reception of a LIN Break Field, a framing error is detected before detection of a LIN Break Field if the reception enable setting (SCR:RXE = 1) is made. However, header reception is performed normally without being stopped.*
- *In assist mode (LAMCR:LAMEN), suppose that a LIN Break Field is detected and a new LIN Break is then transmitted from the master before the completion of ID Field reception. Then, a framing error is detected by the "L" level at the 10th bit in the new LIN Break Field regardless of the reception disable setting (SCR:RXE = 0). However, header reception is performed normally without being stopped.*
- *This bit is set "0" when Software reset (SCR:UPCL=1) is done.*

[bit11] ORE: Overrun Error Flag Bit

- This bit is set to "1" if an overrun occurs during reception. This bit is cleared by writing "1" to the REC bit in the serial status register (SSR).
- A reception interrupt request is issued when the ORE bit and the RIE bit are "1".
- For details on the interrupt request output, see Table 2-3.
- If this flag is set, the reception data register (RDR) data will be invalid.
- If this flag is set when a reception FIFO is used, the reception FIFO enable bits are cleared, and no reception data is stored to the reception FIFO.

Bit	Overrun Error Flag Bit
0	No overrun error
1	Overrun error found

Note:

- *This bit is set "0" when Software reset (SCR:UPCL=1) is done.*

[bit10] RDRF: Reception Data Full Flag Bit

- This flag indicates the status of the reception data register (RDR).
- This bit is set to "1" when reception data is loaded into RDR. Reading the reception data register (RDR) causes this bit to be cleared to "0".
- A reception interrupt request is issued when the RDRF bit and the RIE bit are "1".
- When the reception FIFO is used, if the reception FIFO receives a prescribed number of data items, RDRF is set to "1".
- When reception FIFO available, SSR:RDRF is set "1" when conditions under below and the condition that the reception idle state continues for at least 8 clock pulses of the baud rate clock are satisfied.
  - The reception FIFO idle detection enable bit (FCR1:FRIIE) is set to "1".
  - The reception FIFO has not received the prescribed number of data items, and data remains in the reception FIFO.

  During an 8-clock count, the counter is reset to 0 when RDR is read, and starts to count the 8clocks again.
- For ECR:RXBLKEN = 0, when the reception FIFO is used, this bit is cleared to "0" when the reception FIFO becomes empty as a result of data being read from it.
- For ECR:RXBLKEN = 1, when the reception FIFO is used, this bit is cleared to "0" when the number of data items in the reception FIFO is equal to or less than the value set in the FBYTE register.

Bit	Reception Data Full Flag Bit
0	The reception data register (RDR) is empty.
1	The reception data register (RDR) contains data.

Notes:

- *When the reception FIFO is used and RDRF has become "1", resetting the reception FIFO (FCR0:FCL2, FCL1 = 1) does not cause RDRF to be set to "0". Therefore, in order to set RDRF to "0" after resetting the reception FIFO, perform a dummy reading of the reception data register in reception disable status (SCR:RXE = 0).*
- *During slave operation (SCR:MS = 1) in assist mode (LAMCR:LAMEN = 1), suppose that the setting is such that the reception data register (RDR) is used for the reception of ID Fields (LAMCR:LIDEN = 0). Then, the loading of an ID Field into the reception data register (RDR) sets the reception data full flag bit (RDRF = 1). The LIN auto header completion flag is also set (LAMSR:LAHC = 1) at this time.*
- *During slave operation (SCR:MS = 1) in assist mode (LAMCR:LAMEN = 1), suppose that the setting is such that the LIN assist mode reception ID register is used for the reception of ID Fields (LAMCR:LIDEN = 1). Then, when an ID Field is received, the received ID value is not stored into the reception data register (RDR) and the reception data full flag bit (SSR:RDRF) is not set.*
- *The Sync Field and the checksum are not stored into the reception data register (RDR) and the reception data full flag bit (SSR:RDRF) is not set.*
- *The transmitted fields are not stored in the reception data register (RDR) and the reception data full flag bit (SSR:RDRF) is not set.*
- *This bit is set "0" when Software reset (SCR:UPCL = 1) is done.*

[bit9] TDRE: Transmission Data Empty Flag Bit

- This bit indicates the status of the transmission data register (TDR).
- Writing transmission data to TDR sets this bit to "0", indicating that TDR contains valid data. Loading data into the transmission shift register to start transmission sets this bit to "1", indicating that TDR does not contain valid data.
- A transmission interrupt request is issued when the TDRE bit and the TIE bit are "1".
- The TDRE bit is set to "1" when programmable clear (SCR:UPCL=1) is done.
- During master operation (SCR:MS = 0) in assist mode (LAMCR:LAMEN = 1), suppose that the setting is such that the transmission data register (TDR) is used for the transmission of ID Fields (LAMCR:LIDEN = 0). Then, transmission of the first bit in the ID Field sets the transmission data empty flag (TDRE = 1).
- For details on the setting and clearing timing of the TDRE bit when the transmission FIFO is used, see "Occurrence of Interrupts and Flag Set Timing When Using Transmission FIFO."

Bit	Transmission Data Empty Flag Bit
0	The transmission data register (TDR) holds data.
1	The transmission data register is empty.

[bit8] TBI: Transmission Bus Idle Flag Bit

- This bit indicates that the LIN interface (v2.1) is not performing transmission.
- This bit is set to "0" when transmission data is written to the transmission data register (TDR).
- This bit is set to "1" during those periods in which the transmission data register (TDR) is empty (TDRE = 1) and transmission is not performed.
- In manual mode (LAMCR:LAMEN = 0):
  - Setting the LIN Break Field (SMR:LBR = 1) sets this bit to "0".
  - Once transmission of the LIN Break Field is complete and the transmission data register is empty, this bit is set to "1".
- In assist mode (LAMCR:LAMEN = 1):
  - This bit is set to "0" during transmission of a header in the master operation (SCR:MS = 0).
  - Once the header transmission (ID Field transmission) is complete and the transmission data register is empty, this bit is set to "1".
  - This bit is set to "0" during the transmission of a response.
  - Once response transmission (checksum transmission) is complete and the transmission data register is empty, this bit is set to "1".
- A transmission interrupt request is issued if this bit is set to "1" and transmission bus idle interrupt is enabled (SCR:TBIE = 1).

Bit	Transmission Bus Idle Flag Bit
0	Transmission in progress
1	No transmission

9.4. Extended Communication Control Register (ESCR)

The extended communication control register (ESCR) enables/disables LIN Break Field interrupts, detects LIN Break Fields, and selects lengths for the LIN Break Field, the LIN Break delimiter, and the stop bit.

Extended Communication Control Register (ESCR)

Figure 9-4 shows the bit configuration of the extended communication control register (ESCR).

Figure 9-4 Bit Configuration of Extended Communication Control Register (ESCR)

Bit	15 ... 8	7	6	5	4	3	2	1	0
Field	(SSR)	Reserved	ESBL	LBL2	LBIE	LBL1	LBL0	DEL1	DEL0
R/W Attribute		R0,W0	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute		-	-	-	-	-	-	-	-
Initial Value		0	0	0	0	0	0	0	0

[bit7] Reserved: Reserved Bit

[bit6] ESBL: Extended Stop Bit Length Selection Bit

This bit selects the length of the stop bits (frame end mark of the transmission data).

- This bit is reset when the ESCRC:ESBLC bit in the clear register is set to "1".
- This bit is set when the ESCRS:ESBLS bit in the set register is set to "1".

Settings where SBL is "0" and ESCR:ESBL is "0": The stop bit is set to 1 bit.
Settings where SBL is "1" and ESCR:ESBL is "0": The stop bit is set to 2 bits.
Settings where SBL is "0" and ESCR:ESBL is "1": The stop bit is set to 3 bits.
Settings where SBL is "1" and ESCR:ESBL is "1": The stop bit is set to 4 bits.

Bit	Extended Stop Bit Length Selection Bit	
0	SMR.SBL = 0	1 bit
	SMR.SBL = 1	2 bits
1	SMR.SBL = 0	3 bits
	SMR.SBL = 1	4 bits

Notes:

- *Reception always detects only the first stop bit.*
- *Set this bit when transmission is disabled (SCR:TXE = 0).*
- *In assist mode (LAMCR:LAMEN = 1), set this bit before setting the LIN Break Field (SCR:LBR = 1).*

[bit4] LBIE: LIN Break Field Detection Interrupt Enable Bit

This bit enables/disables LIN Break Field detection interrupts.
When the LIN Break Field detection flag (LBD) is "1", enabling interrupts (LBIE = 1) generates a status interrupt.

- This bit is reset when the ESCRC:LBIEC bit in the clear register is set to "1".

This bit is set when the ESCRS:LBIES bit in the set register is set to "1".

Bit	LIN Break Field Detection Interrupt Enable Bit
0	Disable LIN Break Field detection interrupts.
1	Enable LIN Break Field detection interrupts.

[bit5, bit3:2] LBL[2:0]: LIN Break Field Length Selection Bits (Effective Only for Master Operation)

- These bits set the LIN Break Field duration as a number of bits.
- Set these bits before setting the LIN Break Field setting bit (SCR:LBR) to "1".
- During slave operation, a LIN Break Field is detected at the 11th bit regardless of the setting of these bits.

Bits			LIN Break Field Length Selection Bit
0	0	0	13-bit length
0	0	1	14-bit length
0	1	0	15-bit length
0	1	1	16-bit length
1	0	0	17-bit length
1	0	1	18-bit length
1	1	0	19-bit length
1	1	1	20-bit length

Notes:

- *This function is effective only during master operation (SCR:MS = 0).*
- *Set these bits before setting the LIN Break Field (SCR:LBR = 1).*
- *In manual mode (LAMCR:LAMEN = 0), only 13-bit to 16-bit lengths are effective. Therefore, always set LBL2 to "0".*

[bit1:0] DEL[1:0]: LIN Break Delimiter Length Selection Bits (Effective Only for Master Operation)

- These bits set the length of the LIN Break delimiter as a number of bits.
- Set these bits before setting the LIN Break Field setting bit(SCR:LBR) to "1".

Bits		LIN Break Delimiter Length Selection Bit
0	0	1-bit length
0	1	2-bit length
1	0	3-bit length
1	1	4-bit length

Notes:

- *This function is effective only during master operation (SCR:MS = 0).*
- *In assist mode (LAMCR:LAMEN = 1), set this bit before setting the LIN Break Field (SCR:LBR = 1).*

9.5. Reception Data Register/Transmission Data Register (RDR/TDR)

The reception data register and the transmission data register are placed at the same address. If read, the register functions as a reception data register, and if written to, it functions as a transmission register.

Reception Data Register (RDR)

Figure 9-5 shows the bit configuration of the serial reception register (RDR).

Figure 9-5 Bit Configuration of Reception Data Register (RDR)

Bit	15 ... 8	7	6	5	4	3	2	1	0
Field		D7	D6	D5	D4	D3	D2	D1	D0
R/W Attribute		R	R	R	R	R	R	R	R
ProtectionAttribute		-	-	-	-	-	-	-	-
Initial Value		0	0	0	0	0	0	0	0

The reception data register (RDR) is the data buffer register used for the reception of serial data.

- Serial data signals transmitted to the serial input pin (SIN pin) are converted by the shift register, and are then stored into the reception data register (RDR).
- The reception data full flag bit (SSR:RDRF) is set to "1" when reception data is stored into the reception data register (RDR). If reception interrupts are enabled (SCR:RIE = 1), a reception interrupt request is generated.
- Read the reception data register (RDR) when the reception data full flag bit (SSR:RDRF) is "1". The reception data full flag bit (SSR:RDRF) is automatically cleared to "0" when reading data from the serial reception data register (RDR).
- If a reception error occurs (when SSR:ORE or FRE becomes "1"), the data in the reception data register (RDR) will be invalid.

Notes:

- *If a reception error occurs, the data in the reception data register (RDR) will be invalid.*
- *The operation in assist mode (LAMCR:LAMEN = 1) is as follows:*
  - During slave operation, suppose that the setting is such that the LIN assist mode reception ID register is used for the reception of ID Fields (LAMCR:LIDEN = 1). Then, when an ID Field is received, the received ID value is not stored into the reception data register (RDR) and the reception data full flag bit (SSR:RDRF) is not set.
  - During slave operation, suppose that the setting is such that the reception data register (RDR) is used for the reception of ID Fields (LAMCR:LIDEN = 0). Then, when an ID Field is received, the received ID value is stored into the reception data register (RDR) and the reception data full flag bit is set (SSR:RDRF = 1). Check the ID value after the LIN auto header completion flag is set (LAMSR:LAHC = 1).
  - The Sync Field and the checksum are not stored into the reception data register (RDR) and the reception data full flag bit (SSR:RDRF) is not set.
  - The transmitted fields are not stored into the reception data register (RDR) and the reception data full flag bit (SSR:RDRF) is not set.
  - Transmission and reception processing for assist mode stops when a reception error occurs (SSR:FRE, ORE, LAMESR:LCSER, LSFER, LBSER, or LPTER). At this time, the response reception processing stops the storing of the reception data in the reception data register, regardless of the reception enable setting (SCR:RXE = 1).

- *The operation when the reception FIFO is used is as follows:*
  - When the reception FIFO receives a predefined data count, the RDRF to "1".
  - When the reception FIFO becomes empty, RDRF is cleared to "0".
  - If a reception error occurs (when SSR:ORE, FRE becomes "1"), the enable bit in the reception FIFO is cleared and the reception data is not stored into the reception FIFO.

Transmission Data Register (TDR)

Figure 9-6 shows the bit configuration of the transmission data register.

Figure 9-6 Bit Configuration of Transmission Data Register (TDR)

Bit	15 ... 8	7	6	5	4	3	2	1	0
Field		D7	D6	D5	D4	D3	D2	D1	D0
R/W Attribute		W	W	W	W	W	W	W	W
Protection Attribute		-	-	-	-	-	-	-	-
Initial Value		1	1	1	1	1	1	1	1

The transmission data register (TDR) is the data buffer register used for the transmission of serial data.

- If transmission operation is enabled (SCR:TXE = 1), and the data to transmit is written to the transmission data register (TDR), the transmission data is transferred to the transmission shift register, is converted into serial data, and is then sent out from the serial data output pin (SOUT pin).
- The transmission data empty flag (SSR:TDRE) is cleared to "0" when the transmission data is written to the serial transmission data register (TDR).
- If the transmission FIFO is disabled or empty, the transmission data empty flag (SSR:TDRE) is set to "1" when transmission data is transferred to the transmission shift register to start transmission.
- If the transmission data empty flag (SSR:TDRE) is "1", the next data for transmission can be written. If transmission interrupts are enabled, a transmission interrupt is generated. Write the next transmission data after a transmission interrupt is generated or when the transmission data empty flag (SSR:TDRE) is "1".
- When the transmission data empty flag (SSR:TDRE) is "0" and the transmission FIFO is disabled or full, transmission data cannot be written to the transmission data register (TDR).

Notes:

- *The transmission data register is a write-only register, and the reception data register is a read-only register. The two registers are placed at the same address, so the written value differs from the read value.*
- *For details on the set timing of the transmission data empty flag (SSR:TDRE) when the transmission FIFO is used, see "Occurrence of Interrupts and Flag Set Timing When Using Transmission FIFO."*
- *The upper 2 bits of data are ignored when all of the following conditions are satisfied.*
  - LIN hardware assist mode (LAMCR:LAMEN = 1)
  - ID DATA register not used (LAMCR:LIDEN = 0)
  - ID DATA being transmitted

9.6. Serial Auxiliary Control Status Register (SACSR)

The serial auxiliary control status register (SACSR) controls the serial test operation, enables/disables auto baud rate adjustment, enables/disables Sync Field interrupts, selects a serial timer activation method, enables/disables timer interrupts, sets the division value of the serial timer operation clock, and enables/disables the serial timer.

Bit Configuration of Serial Auxiliary Control Status Register (SACSR)

Figure 9-7 shows the bit configuration of the serial auxiliary control status register (SACSR).

Figure 9-7 Bit Configuration of Serial Auxiliary Control Status Register (SACSR)

Bit	15	14	13	12	11	10	9	8
Field	STST	BST	SFD	SFDE	AUTE	TRG1	TRG0	TINT
R/W Attribute	R/W	R, WX	R, WX	R/W	R/W	R/W	R/W	R, WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	TINTE	Reserved	TRGE	TDIV3	TDIV2	TDIV1	TDIV0	TMRE
R/W Attribute	R/W	R0, W0	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

[bit15] STST: Serial Test Bit

This bit enables or disables serial test mode.

When serial test mode is enabled, SOUT and SIN are connected together inside the multi-function serial interface, so that the data transmitted from SOUT can be directly received from SIN.

When serial test mode is enabled, the SOUT pin is fixed to "H", and any data input to the SIN pin is ignored.

■ This bit is reset when the SACSRC:STSTC bit in the clear register is set to "1".
This bit is set when the SACSRS:STSTS bit in the set register is set to "1".

Bit	Serial Test Bit
0	Disable serial test mode.
1	Enable serial test mode.

Note:

- *This bit can be changed only when transmission/reception are disabled (SCR:TXE = 0, SCR:RXE = 0).*

[bit14] BST: Baud Rate Setting Flag

This bit indicates that auto baud rate adjustment has been performed upon the reception of a Sync Field.

This bit is updated upon the detection of a 5th falling edge of the LIN bus in the Sync Field.

Bit	Baud Rate Setting Flag	
	Write	Read
0	No effect	Without auto baud rate adjustment
1		With auto baud rate adjustment

Notes:

- *This bit is fixed to "0" when auto baud rate adjustment is disabled (AUTE = 0).*
- *A software reset (SCR:UPCL = 1) resets this bit to "0".*
- *This bit is effective only when the Sync Field detection flag (SACSR:SFD) is "1".*
- *Writing to this bit is invalid.*

[bit13] SFD: Sync Field Detection Flag

This bit indicates that a Sync Field has been detected.

This bit is set to "1" upon the detection of a 5th falling edge of the LIN bus in the Sync Field.

When this bit is "1" and the Sync Field detection interrupt enable bit (SFDE) is "1", a status interrupt request is issued.

This bit is reset when the SACSRC:SFDC bit in the clear register is set to "1".

Bit	Sync Field Detection Flag	
	Write	Read
0	No effect	Sync Field not detected
1	No effect	Sync Field detected

Notes:

- *A software reset (SCR:UPCL = 1) resets this bit to "0".*
- *Writing to this bit is invalid.*
- *This bit is effective both in master mode (SCR:MS = 0) and in slave mode (SCR:MS = 1).*

[bit12] SFDE: Sync Field Detection Interrupt Enable Bit

This bit enables/disables Sync Field interrupts to the CPU.

When this bit is "1" and the Sync Field detection flag (SFD) is "1", a status interrupt request is issued.

■ This bit is reset when the SACSRC:SFDEC bit in the clear register is set to "1".
This bit is set when the SACSRS:SFDES bit in the set register is set to "1".

Bit	Sync Field Detection Interrupt Enable Bit
0	Disable interrupts of Sync Field detection.
1	Enable interrupts of Sync Field detection.

[bit11] AUTE: Auto Baud Rate Adjustment Bit

This bit enables/disables auto baud rate adjustment.

- This bit is reset when the SACSRC:AUTEC bit in the clear register is set to "1".

This bit is set when the SACSRS:AUTES bit in the set register is set to "1".

Bit	Auto Baud Rate Adjustment Bit
0	Disable auto baud rate adjustment.
1	Enable auto baud rate adjustment.

Notes:

- *This bit is internally fixed to "0" in master mode (SCR:MS = 0).*
- *Setting this bit to "1" sets the timer operation clock division bits (TDIV[3:0]) to 0b0011 (division by 8).*
- *These bits can be changed from "0" to "1" only when the serial timer enable bit (TMRE) is "0".*
- *When auto baud rate adjustment is enabled (AUTE = 1), set in the internal clock(BGR1:EXT=0),*

[bit10:9] TRG[1:0]: Trigger Selection Bits

These bits select the detection method for the external trigger edges that are used to activate the serial timer.

Bits		Edge Detection Method for External Trigger
0	0	Rising edge detection
0	1	Falling edge detection
1	0	Both edge detection
1	1	Setting prohibited

Notes:

- *These bits are invalid if the external trigger enable bit (TRGE) is "0".*
- *These bits can be changed only when the serial timer enable bit (TMRE) is "0".*

[bit8] TINT: Timer Interrupt Flag

If the serial timer register (STMR) and the serial timer comparison register (STMCR) match, the serial timer register (STMR) is set to "0", and this bit is set to "1".

When this bit is "1" and the timer interrupt enable bit (TINTE) is "1", a status interrupt request is issued.

Writing to this bit is invalid.

- This bit is reset when the SACSRC:TINTC bit in the clear register is set to "1.

Bit	Description	
	Write	Read
0	No effect	No timer interrupt request
1	No effect	Timer interrupt request issued

Note:

- *A software reset (SCR:UPCL = 1) resets this bit to "0".*

[bit7] TINTE: Timer Interrupt Enable Bit

This bit enables/disables timer interrupts to the CPU.

When this bit is "1" and the timer interrupt flag (TINT) is "1", a status interrupt request is issued.

- This bit is reset when the SACSRC:TINTEC bit in the clear register is set to "1".
- This bit is set when the bit SACSRS:TINTES in the set register is set to "1".

Bit	Description
0	Disable interrupts triggered by the serial timer.
1	Enable interrupts triggered by the serial timer.

[bit6] Reserved: Reserved Bit

[bit5] TRGE: External Trigger Enable Bit

This bit selects the method used to activate the serial timer.

- This bit is reset when the SACSRC:TRGEC bit in the clear register is set to "1".

This bit is set when the SACSRS:TRGES bit in the set register is set to "1".

Bit	Description
0	The serial timer starts when value of the serial timer enable bit (TMRE) is changed from "0" to "1".
1	If the serial timer enable bit (TMRE) is set to "1", the detection of an external trigger edge specified by the trigger selection bit (TRG1, TRG0) starts the serial timer.

Notes:

- *These bits can be changed only when the serial timer enable bit (TMRE) is "0".*
- *If the serial timer enable bit (TMRE) is set to "0", the detection of an external trigger edge specified by the trigger selection bit (TRG1, TRG0) does not start the serial timer.*

[bit4:1] TDIV[3:0]: Timer Operation Clock Division Bits

These bits set the division ratio of the serial timer.

Bits				Timer Operation Clock						
				Division Ratio	Φ = 8 MHz	Φ = 10 MHz	Φ = 16 MHz	Φ = 20 MHz	Φ = 24 MHz	Φ = 32 MHz
0	0	0	0	Φ	125 ns	100 ns	62.5 ns	50 ns	41.67 ns	31.25 ns
0	0	0	1	Φ/2	250 ns	200 ns	125 ns	100 ns	83.33 ns	62.5 ns
0	0	1	0	Φ/4	500 ns	400 ns	250 ns	200 ns	166.67 ns	125 ns
0	0	1	1	Φ/8	1 us	800 ns	500 ns	400 ns	333.33 ns	250 ns
0	1	0	0	Φ/16	2 us	1.6 us	1 us	800 ns	666.67 ns	500 ns
0	1	0	1	Φ/32	4 us	3.2 us	2 us	1.6 us	1.33 us	1 us
0	1	1	0	Φ/64	8 us	6.4 us	4 us	3.2 us	2.67 us	2 us
0	1	1	1	Φ/128	16 us	12.8 us	8 us	6.4 us	5.33 us	4 us
1	0	0	0	Φ/256	32 us	25.6 us	16 us	12.8 us	10.67 us	8 us

Φ: Bus clock

Notes:

- *These bits can be changed only when the serial timer enable bit (TMRE) is "0".*
- *Settings other than the above are prohibited.*

[bit0] TMRE: Serial Timer Enable Bit

This bit enables or disables the serial timer operation.

■ This bit is reset when the SACSRC:TMREC bit in the clear register is set to "1".
This bit is set when the SACSRS:TMRES bit in the set register is set to "1".

Bit	Serial Timer Enable Bit
0	Stop serial timer operation. When stopped, the value of the serial timer register (STMR) is retained.
1	Changing this bit from "0" to "1" initializes the serial timer register (STMR) to "0" and starts serial timer operation.

Note:

- *If the external trigger is enabled (TRGE = 1), setting this bit to "1" does not start the serial timer until detection of an external trigger edge specified by the trigger selection bits (SACSR:TRG1, TRG0).*

9.7. Serial Timer Register (STMR)

The serial timer register (STMR) represents the timer value of the serial timer.

Bit Configuration of Serial Timer Register (STMR)

Figure 9-8 shows the bit configuration of the serial timer register (STMR).

Figure 9-8 Bit Configuration of Serial Timer Register (STMR)

Bit	15	14	13	12	11	10	9	8
Field	TM15	TM14	TM13	TM12	TM11	TM10	TM9	TM8
R/W Attribute	R, WX	R, WX	R, WX	R, WX	R, WX	R, WX	R, WX	R, WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	TM7	TM6	TM5	TM4	TM3	TM2	TM1	TM0
R/W Attribute	R, WX	R, WX	R, WX	R, WX	R, WX	R, WX	R, WX	R, WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

[bit15:0] TM15-0: Timer Data Bits

These bits indicate the timer value of the serial timer.

During timer operation, the timer value of the serial timer is incremented by 1 for each timer operation clock (specified by SACSR:TDIV3 to TDIV0).

Note:

- *These bits are initialized to "0" when the timer operation starts.*

9.8. Serial Timer Comparison Register (STMCR)

The serial timer comparison register (STMCR) sets the timer comparison value of the serial timer.

Bit Configuration of Serial Timer Comparison Register (STMCR)

Figure 9-9 shows the bit configuration of the serial timer comparison register (STMCR).

Figure 9-9 Bit Configuration of Serial Timer Comparison Register (STMCR)

Bit	15	14	13	12	11	10	9	8
Field	TC15	TC14	TC13	TC12	TC11	TC10	TC9	TC8
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	TC7	TC6	TC5	TC4	TC3	TC2	TC1	TC0
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

[bit15:0] TC15-0: Compare Bits

These bits set a comparison value for the serial timer.

These bits are compared with the serial timer register (STMR). If the serial timer register (STMR) coincides with these bits when updated, the serial timer register is set to "0". At this time, the timer interrupt flag (SACSR:TINT) is set to "1".

The period below is (STMCR:TC+1) X The timer clock (set by SACSR:TDIV3-0)

- SACSR:TINT is set "1"

Notes:

- *The timer interrupt flag (SACSR:TINT) is fixed to "1" when all of the following conditions are satisfied.*
  - Synchronous transmission is disabled (SACSR:TSYNE = 0).
  - 0x0000 is set in this register.
  - The timer is operating.
  - 0b0000 is set as the timer operating clock division value (SACSR:TDIV3-0).
- *This register can be changed only when the serial timer is disabled (SACSR:TMRE = 0).*
- *The serial timer register (STMR) may be reset to 0x0000 before baud rate adjustment when all of the following conditions are satisfied. For this reason, set these bits to a value larger than that set for the Sync Field upper limit bits (SFUR) when the auto baud rate adjustment bit (SACSR:AUTE) is "1".*
  - The auto baud rate adjustment bit (SACSR:AUTE) is "1".
  - The value of these bits is equal to or less than the value set for the Sync Field upper limit bits (SFUR).

9.9. Sync Field Upper Limit Register (SFUR)

The Sync Field upper limit register (SFUR) sets the upper limit on the values that can be set in the baud rate generator register for auto baud rate adjustment.

Bit Configuration of Sync Field Upper Limit Register (SFUR)

Figure 9-10 shows the bit configuration of the Sync Field upper limit register (SFUR).

Figure 9-10 Bit Configuration of Sync Field Upper Limit Register (SFUR)

Bit	15	14	13	12	11	10	9	8
Field	Reserved	TU14	TU13	TU12	TU11	TU10	TU9	TU8
R/W Attribute	R0, W0	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	TU7	TU6	TU5	TU4	TU3	TU2	TU1	TU0
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

[bit15] Reserved: Reserved Bit

[bit14:0] TU14-0: Upper Limit Bits

These bits set the upper limit on the values that can be set in the baud rate generator register (BGR1, BGR0) for auto baud rate adjustment.

Suppose that the auto baud rate adjustment bit (SACSR:AUTE) is "1" and that operation is being performed in slave mode (SCR:MS = 1). Also suppose that the value of the serial timer register (STMR) after receiving Sync Field does not exceed the value of these bits and is not less than the value of the Sync Field lower limit register (SFLR). Then, the baud rate generator register (BGR1, BGR0) is set to the value of the serial timer register (STMR).

Note:

- *These bits can be changed when the auto baud rate adjustment bit (SACSR:AUTE) is "0".*

9.10. Sync Field Lower Limit Register (SFLR)

The Sync Field lower limit register (SFLR) sets the lower limit for the values that can be set in the baud rate generator register for auto baud rate adjustment.

Bit Configuration of Sync Field Lower Limit Register (SFLR)

Figure 9-11 shows the bit configuration of the Sync Field lower limit register (SFLR).

Figure 9-11 Bit Configuration of Sync Field Lower Limit Register (SFLR)

Bit	15	14	13	12	11	10	9	8
Field	Reserved	TL14	TL13	TL12	TL11	TL10	TL9	TL8
R/W Attribute	R0, W0	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	TL7	TL6	TL5	TL4	TL3	TL2	TL1	TL0
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

[bit15] Reserved: Reserved Bit

[bit14:0] TL14-0: Lower Limit Bits

These bits set the lower limit on the values that can be set in the baud rate generator register (BGR1, BGR0) for auto baud rate adjustment.

Suppose that the auto baud rate adjustment bit (SACSR:AUTE) is "1" and that operation is being performed in slave mode (SCR:MS = 1). Also suppose that the value of the serial timer register (STMR) after receiving Sync Field does not exceed the value of the Sync Field upper limit register (SFUR) and is not less than the value of these bits. Then, the baud rate generator register (BGR1, BGR0) is set to the value of the serial timer register (STMR).

Note:

- *These bits can be changed when the auto baud rate adjustment bit (SACSR:AUTE) is "0".*

9.11. Baud Rate Generator Register 1, 0 (BGR1, BGR0)

Baud rate generator register 1, 0 (BGR1, BGR0) sets the division ratio of the serial clock. An external clock can also be selected as the clock source of the reload counter.

Bit Configuration of Baud Rate Generator Register 1, 0 (BGR0/1)

Figure 9-12 shows the bit configuration of baud rate generator register 1, 0 (BGR1, BGR0).

Figure 9-12 Bit Configuration of Baud Rate Generator Register 1, 0 (BGR1, BGR0)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	EXT	BGR1							BGR0							
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

- The baud rate generator registers set the division ratio of the serial clock.
- BGR1 represents the higher bits while BGR0 represents the lower bits. The reload value for counting can be written to these registers, and the set reload values can be read from these registers.
- The reload counter starts counting when a reload value is written in baud rate generator register 1, 0 (BGR1, BGR0).
- The EXT bit, bit15, selects the internal clock or an external clock as the clock source of the reload counter. EXT = 0 selects the internal clock. EXT = 1 selects an external clock.

[bit15] EXT: External Clock Selection Bit

Bit	External Clock Selection Bit
0	Use the internal clock.
1	Use an external clock.

[bit14:8] BGR1: Baud Rate Generator Register 1

BGR1	Baud Rate Generator Register 1
Write	Write to reload counter bits 8 to 14.
Read	Read the setting value of BGR1.

[bit7:0] BGR0: Baud Rate Generator Register 0

BGR0	Baud Rate Generator Register 0
Write	Write to reload counter bits 0 to 7.
Read	Read the setting value of BGR0.

Notes:

- *Use 16-bit access for writing a value to the baud rate generator register (BGR1, BGR0).*
- *If the values set in the baud rate generator registers (BGR1, BGR0) are changed, the new values are reloaded when the counter value becomes "00000"h. Therefore, to enable the new value immediately, execute programmable clear (UPCL) after changing the values of BGR1/0.*

- *When the reload value is even, the "L" width of the serial clock is 1-bus-clock-cycle longer than the "H" width. When the reload value is odd, the "L" width and the "H" width of the serial clock are equal.*
- *Set the reload value to 3 or larger. However, data may not be received normally depending on baud rate errors and the reload value setting.*
- *To change the clock setting to that of the external clock (EXT = 1) when the baud rate generator is operating, perform the following: 1. Write "0" to baud rate generators 1, 0 (BGR1, BGR0). 2. Execute programmable clear (UPCL). 3. Set external clock mode (EXT = 1).*
- *When auto baud rate adjustment is enabled (SACSR:AUTE = 1), set in the internal clock(BGR1:EXT=0),*

9.12. LIN Assist Mode Status Register (LAMSR)

The LIN assist mode status register (LAMSR) checks the auto header transmission/reception status and the reception error flag.

LIN Assist Mode Status Register (LAMSR)

Figure 9-13 shows the bit configuration of the LIN assist mode status register (LAMSR).

Figure 9-13 LIN Assist Mode Status Register (LAMSR)

Bit	15	14	13	12	11	10	9	8	7 ... 0
Field	LER	SER	RDRF	TDRE	TBI	LCSC	Reserved	LAHC	(LAMCR)
R/W Attribute	R, WX	R, WX	R, WX	R, WX	R, WX	R, WX	R0, W0	R, WX	
Protection Attribute	-	-	-	-	-	-	-	-	
Initial Value	0	0	0	1	1	0	0	0	

[bit15] LER: LIN Representative Error Flag Bit

- This bit is set to "1" when any of the following errors occurs. For details on the setting and clearing conditions for the error flag bit, see the explanations of the individual bits of the LIN assist mode error status register (LAMESR).
  - LIN bus error flag bit (LBSER)
  - LIN Sync Data error flag bit (LSFER)
  - LIN ID parity error flag bit (LPTER)
  - LIN checksum error flag bit (LCSER)

Bit	LIN Representative Error Flag Bit
0	There is no error.
1	An error has occurred.

Notes:

- *This bit is effective in assist mode (LAMCR:LAMEN = 1).*
- *In manual mode (LAMCR:LAMEN = 0), this bit is always read as "0".*

[bit14] SER: Serial Interface Representative Error Flag Bit

- This bit is set to "1" when any of the following errors occurs. For details on the setting and clearing conditions for the error flag bit, see the explanations of the individual bits of the serial status register (SSR).
  - Framing error flag bit (FRE)
  - Overrun error flag bit (ORE)

Bit	Serial Interface Representative Error Flag Bit
0	There is no error.
1	An error has occurred.

[bit13] RDRF: Reception Data Full Flag Bit

- This bit is the same as the reception data full flag bit (RDRF) in the serial status register (SSR). For an explanation of this bit, see the description of the serial status register (SSR).

[bit12] TDRE: Transmission Data Empty Flag Bit

- This bit is the same as the transmission data empty flag bit (TDRE) in the serial status register (SSR). For an explanation of this bit, see the description of the serial status register (SSR).

[bit11] TBI: Transmission Bus Idle Flag Bit

- This bit is the same as the transmission bus idle flag bit (TBI) in the serial status register (SSR). For an explanation of this bit, see the description of the serial status register (SSR).

[bit10] LCSC: LIN Checksum Calculation Completion Flag Bit

- This bit indicates the completion of checksum calculation.
- This bit is set to "1" upon the completion of checksum calculation, which happens when data of the configured length (LAMCR:LDL3 to LDL0) and a checksum are received during assist mode (LAMCR:LAMEN = 1) reception operation.
- A status interrupt request is issued when the LIN checksum calculation completion flag bit (LCSC) and the checksum calculation completion interrupt enable bit (LCSCIE) are "1".

(When reading)

"1": Checksum calculation completion has been detected.
"0": Checksum calculation completion has not been detected.

(When writing)

No effect.

Bit	Checksum Calculation Completion Flag Bit	
	Write	Read
0	No effect	Checksum calculation is in progress. Or Checksum calculation start is being awaited.
1	No effect	Checksum calculation has been completed.

Notes:

This bit is reset when the LAMSRC:LCSCC bit in the clear register is set to "1".

- *This bit is effective in assist mode (LAMCR:LAMEN = 1).*
- *This bit is set "0" when Software reset is done (SCR:UPCL=1).*

[bit9] Reserved: Reserved Bit

[bit8] LAHC: LIN Auto Header Completion Flag Bit

- This bit indicates the LIN auto header status.
- This bit is set to "1" when a LIN header is received in assist mode (LAMCR:LAMEN = 1).

- A status interrupt request is issued when the LIN auto header completion flag bit (LAHC) and the LIN auto header completion interrupt enable bit (LAHCIE) are "1".
- After this bit is set to "1", reading the LIN auto header reception ID register (LAMRID) sets this bit to "0".

(When reading)

"1": LIN auto header completion has been detected.
"0": LIN auto header completion has not been detected.

(When writing)

"No effect.

Bit	LIN Auto Header Completion Flag Bit	
	Write	Read
0	No effect	LIN auto header is being received. Or, its reception is being awaited.
1	No effect	LIN auto header has been received.

Notes:
This bit is reset when the LAMSRC:LAHCC bit in the clear register is set to "1".

- *This bit is effective in assist mode (LAMCR:LAMEN = 1).*
- *This bit is set "0" when Software reset is done (SCR:UPCL=1).*

9.13. LIN Assist Mode Control Register (LAMCR)

The LIN assist mode control register (LAMCR) enables LIN auto header processing, enables the use of the LIN ID register, selects a LIN checksum type, clears TDR, and sets the LIN data length in LIN assist mode.

LIN Assist Mode Control Register (LAMCR)

Figure 9-14 shows the bit configuration of the LIN assist mode control register (LAMCR).

Figure 9-14 Bit Configuration of LIN Assist Mode Control Register (LAMCR)

Bit	15 ... 8	7	6	5	4	3	2	1	0
Field	(LAMSR)	LDL3	LDL2	LDL1	LDL0	LTDRCL	LCSTYP	LIDEN	LAMEN
R/W Attribute		R/W	R/W	R/W	R/W	R0, W	R/W	R/W	R/W
Protection Attribute		-	-	-	-	-	-	-	-
Initial Value		0	0	0	0	0	0	0	0

[bit7:4] LDL[3:0]: LIN Data Length Setting Bit

- These bits set the length of the LIN response data within a range of 0 to 8 bytes.
- The setting value should represent the data length.
- Data of the length specified by these bits is sent in the transmission operation.
- Data of the length specified by these bits is received in the reception operation.

Bits				LIN Data Length Setting Bits
0	0	0	0	0 byte
0	0	0	1	1 byte
0	0	1	0	2 bytes
0	0	1	1	3 bytes
0	1	0	0	4 bytes
0	1	0	1	5 bytes
0	1	1	0	6 bytes
0	1	1	1	7 bytes
1	0	0	0	8 bytes
Other than the above				Setting prohibited

Notes:

- *This function is effective only in LIN assist mode (LAMEN = 1).*
- *During master mode operation, set these bits before the generation of LIN Break Fields (SCR:LBR = 1).*
- *During slave mode operation, set these bits before the start of response transmission/reception.*

[bit3] LTDRCL: LIN Transmission Data Register Clear Bit

- This bit clears the transmission data register (TDR).
- When set to "1": The transmission data empty flag bit (SSR:TDRE) and the transmission bus idle flag bit (SSR:TBI) are set to "1".
- When set to "0": No effect on the operation.

- When it is read, "0" is always read.
- This bit is set when the LAMCRS:LTDRCLS bit in the set register is set to "1".

Bit	LIN Transmission Data Register Clear Bit	
	Write	Read
0	No effect	"0" is always read.
1	Set the transmission data empty flag bit (SSR:TDRE) and the transmission bus idle flag bit (SSR:TBI) to "1".	

Notes:

- *Clearing the transmission data register does not reset the transmission FIFO.*
- *When the transmission FIFO is used, clear the transmission FIFO (FCR0:FCL1 or FCR0:FCL2) and then clear the transmission data register.*
- *Do not set this bit to "1" during transmission.*

[bit2] LCSTYP: LIN Checksum Type Selection Bit

This bit selects the LIN checksum type.

- This bit is reset when the LAMCRC:LCSTYPC bit in the clear register is set to "1".
- This bit is set when the LAMCRS:LCSTYPS bit in the set register is set to "1".

"0": The standard checksum is selected.
"1": The extended checksum is selected.

Bit	LIN Checksum Type Selection Bit
0	Standard checksum
1	Extended checksum

Notes:

- *This function is effective only in LIN assist mode (LAMEN = 1).*
- *During master mode operation, set these bits before the generation of LIN Break Fields (SCR:LBR = 1).*
- *During slave mode operation, set these bits before the start of response transmission/reception.*

[bit1] LIDEN: LIN ID Register Use Enable Bit

This bit sets whether to use the LIN assist mode transmission/reception ID register (LAMTID/LAMRID).

- This bit is reset when the LAMCRC:LIDENC bit in the clear register is set to "1".
- This bit is set when the LAMCRS:LIDENS bit in the set register is set to "1".

(In master mode (SCR:MS = 0))

When set to "0": Data written to the transmission data register (TDR) as the transmission data of LIN ID Field is used.

When set to "1": Data written to the LIN assist mode transmission ID register (LAMTID) as the transmission data of LIN ID Field is used.

(In slave mode (SCR:MS = 1))

When set to "0": The received data of LIN ID Field is stored into the reception data register (RDR).
When set to "1": The received data of LIN ID Field is stored into the LIN assist mode reception ID register (LAMRID).

Bit	LIN ID Register Use Enable Bit	
	Master	Slave
0	Use the transmission data register (TDR).	Use the reception data register (RDR).
1	Use the LIN assist mode transmission ID register (LAMTID).	Use the LIN assist mode reception ID register (LAMRID).

Note:

- *This function is effective only in LIN assist mode (LAMEN = 1).*

[bit0] LAMEN: LIN Assist Mode Processing Enable Bit

This bit sets whether LIN assist mode is used.

- This bit is reset when the LAMCRC:LAMENC bit in the clear register is set to "1".
- This bit is set when the LAMCRS:LAMENS bit in the set register is set to "1".

When set to "0": LIN manual mode is selected.
When set to "1": LIN assist mode is selected.

Bit	LIN Assist Mode Enable Bit
0	Manual mode
1	Assist mode

Notes:

- *In manual mode, change this bit when LIN transmission/reception is disabled (SCR:RXE = 0, SCR:TXE = 0).*
- *In assist mode, do not change this bit when LIN is operating.*
- *When changing this bit, execute programmable clear (SCR:UPCL = 1) immediately after changing the bit.*

9.14. LIN Assist Mode Interrupt Enable Register (LAMIER)

The LIN assist mode interrupt enable register (LAMIER) enables/disables LIN auto header completion interrupts, LIN checksum calculation completion interrupts, LIN bus error interrupts, LIN ID parity error interrupts, LIN Sync Data error interrupts, and LIN checksum error interrupts.

LIN Assist Mode Interrupt Enable Register (LAMIER)

Figure 9-15 shows the bit configuration of the LIN assist mode interrupt enable register (LAMIER).

Figure 9-15 Bit Configuration of LIN Assist Mode Interrupt Enable Register (LAMIER)

Bit	15	14	13	12	11	10	9	8	7 ... 0
Field	Reserved	LCSERIE	LPTERIE	LSFERIE	LBSERIE	LCSCIE	Reserved	LAHCIE	(LAMTID/LAMRID)
R/W Attribute	R0, W0	R/W	R/W	R/W	R/W	R/W	R0, W0	R/W	
Protection Attribute	-	-	-	-	-	-	-	-	
Initial Value	0	0	0	0	0	0	0	0	

Note:

- *Please use 8bit access to access this register.*

[bit15] Reserved: Reserved Bit

[bit14] LCSERIE: LIN Checksum Error Interrupt Enable Bit

- This bit enables/disables LIN checksum error interrupt request output to the CPU.
- A reception interrupt request is issued when the LCSERIE bit and the LAMESR:LCSER bit are "1".
- This bit is reset when the LAMIERC:LCSERIEC bit in the clear register is set to "1".
- This bit is set when the LAMIERS:LCSERIES bit in the set register is set to "1".

Bit	LIN Checksum Error Interrupt Enable Bit
0	Disable LIN checksum error interrupts.
1	Enable LIN checksum error interrupts.

[bit13] LPTERIE: LIN ID Parity Error Interrupt Enable Bit

- This bit enables/disables LIN ID parity error interrupt request output to the CPU.
- A reception interrupt request is issued when the LPTERIE bit and the LAMESR:LPTER bit are "1".
- This bit is reset when the LAMIERC:LPTERIEC bit in the clear register is set to "1".
- This bit is set when the LAMIERS:LPTERIES bit in the set register is set to "1".

Bit	LIN ID Parity Error Interrupt Enable Bit
0	Disable LIN ID parity error interrupts.
1	Enable LIN ID parity error interrupts.

[bit12] LSFERIE: LIN Sync Data error interrupt enable bit

- This bit enables/disables LIN Sync Data error interrupt request output to the CPU.

- A reception interrupt request is issued when the LSFERIE bit and the LAMESR:LSFER bit are "1".
- This bit is reset when the LAMIERC:LSFERIEC bit in the clear register is set to "1".
- This bit is set when the LAMIERS:LSFERIES bit in the set register is set to "1".

Bit	LIN Sync Data Error Interrupt Enable Bit
0	Disable LIN Sync Data error interrupts.
1	Enable LIN Sync Data error interrupts.

[bit11] LBSERIE: LIN Bus Error Interrupt Enable Bit

- This bit enables/disables LIN bus error interrupt request output to the CPU.
- A reception interrupt request is issued when the LBSERIE bit and the LAMESR:LBSER bit are "1".
- This bit is reset when the LAMIERC:LBSERIEC bit in the clear register is set to "1".
- This bit is set when the LAMIERS:LBSERIES bit in the set register is set to "1".

Bit	LIN Bus Error Interrupt Enable Bit
0	Disable LIN bus error interrupts.
1	Enable LIN bus error interrupts.

[bit10] LCSCIE: LIN Checksum Calculation Completion Interrupt Enable Bit

- This bit enables/disables LIN checksum calculation completion interrupt request output to the CPU.
- A status interrupt request is issued when the LCSCIE bit and LAMSR:LCSC bit are "1".
- This bit is reset when the LAMIERC:LCSCIEC bit in the clear register is set to "1".
- This bit is set when the LAMIERS:LCSCIES bit in the set register is set to "1".

Bit	LIN Checksum Calculation Completion Interrupt Enable Bit
0	Disable LIN checksum calculation completion interrupts.
1	Enable LIN checksum calculation completion interrupts.

[bit9] Reserved: Reserved Bit

[bit8] LAHCIE: LIN Auto Header Completion Interrupt Enable Bit

- This bit enables/disables LIN auto header completion interrupt request output to the CPU.
- A status interrupt request is issued when the LAHCIE bit and the LAMSR:LAHC bit are "1".
- This bit is reset when the LAMIERC:LAHCIEC bit in the clear register is set to "1".
- This bit is set when the LAMIERS:LAHCIES bit in the set register is set to "1".

Bit	LIN Auto Header Completion Interrupt Enable Bit
0	Disable LIN auto header completion interrupts.
1	Enable LIN auto header completion interrupts.

9.15. LIN Assist Mode Transmission/Reception ID Register (LAMTID/LAMRID)

The LIN assist mode transmission/reception ID register (LAMTID/LAMRID) displays the reception LIN ID parity, sets the transmission LIN ID, and displays the reception ID.

LIN Assist Mode Transmission ID Register (LAMTID)

Figure 9-16 shows the bit configuration of the LIN assist mode transmission ID register (LAMTID).

Figure 9-16 Bit Configuration of LIN Assist Mode Transmission ID Register (LAMTID)

Bit	15 ... 8	7	6	5	4	3	2	1	0
Field	(LAMIER)	Reserved	Reserved	LID5	LID4	LID3	LID2	LID1	LID0
R/W Attribute		W0	W0	W	W	W	W	W	W
Protection Attribute		-	-	-	-	-	-	-	-
Initial Value		0	0	0	0	0	0	0	0

[bit7:6] Reserved: Reserved Bits

[bit5:0] LID5-0: LIN ID Setting Bits

(When writing)

When assist mode is set to master and the LIN ID register use enable bit (LIDEN) is set to enabled, these bits set LIN ID Field data.

Note:

- *This function is effective only in LIN assist mode (LAMCR:LAMEN = 1).*

LIN Assist Mode Reception ID Register (LAMRID)

Figure 9-17 shows the bit configuration of the LIN assist mode reception ID register (LAMRID).

Figure 9-17 Bit Configuration of LIN Assist Mode Reception ID Register (LAMRID)

Bit	15 ... 8	7	6	5	4	3	2	1	0
Field	(LAMIER)	P1	P0	LID5	LID4	LID3	LID2	LID1	LID0
R/W Attribute	-	R	R	R	R	R	R	R	R
Protection Attribute		-	-	-	-	-	-	-	-
Initial Value	-	0	0	0	0	0	0	0	0

[bit7:6] P1, P0: LIN ID Parity Display Bits

(When reading)

When assist mode is set to slave and the LIN ID register use enable bit (LIDEN) is set to enabled, these bits indicate the parity value of the received LIN ID Field.

[bit5:0] LID5-0: LIN ID Display Bits

(When reading)

When assist mode is set to slave and the LIN ID register use enable bit (LIDEN) is set to enabled, these bits indicate the data of the received LIN ID Field.

Notes:

- *This function is effective only in LIN assist mode (LAMCR:LAMEN = 1).*
- *These bits indicate the data of the ID Field received in this register even if a LIN ID parity error occurs.*

9.16. LIN Assist Mode Error Status Register (LAMESR)

The LIN assist mode error status register (LAMESR) checks the flag of a LIN checksum error, LIN Sync Data error, LIN ID parity error, and LIN bus error.

LIN Assist Mode Error Status Register (LAMESR)

Figure 9-18 shows the bit configuration of the LIN assist mode error status register (LAMESR).

Figure 9-18 Bit Configuration of LIN Assist Mode Error Status Register (LAMESR)

Bit	15	14	13	12	11	10	9	8	7 ... 0
Field	Reserved	LCSER	LPTER	LSFER	LBSER	Reserved	Reserved	Reserved	(LAMERT)
R/W Attribute	R0,W0	R,WX	R,WX	R,WX	R,WX	R0,W0	R0,W0	R0,W0	
Protection Attribute	-	-	-	-	-	-	-	-	
Initial Value	0	0	0	0	0	0	0	0	

Note:

- *This register just can be accessed with 8bit access or 16bit access.*

[bit15] Reserved: Reserved Bit

[bit14] LCSER: LIN Checksum Error Flag Bit

- This bit is set to "1" when a LIN checksum error occurs.
- A reception interrupt request is issued when the LCSER bit and the LCSERIE bit are "1".
- For details on interrupt request output, see Table 2-6
- This bit is reset when the LAMESRC:LCSERC bit in the clear register is set to "1".

Bit	LIN Checksum Error Flag Bit	
	Write	Read
0	No effect	There is no error.
1	No effect	An error has occurred.

Notes:

- *This function is effective only in LIN assist mode (LAMCR:LAMEN = 1).*
- *This bit is set "0" when Software reset is done (SCR:UPCL=1).*

[bit13] LPTER: LIN ID Parity Error Flag Bit

- This bit is set to "1" if a LIN ID parity error occurs.
- A reception interrupt request is issued when the LPTER bit and the LPTER IE bit are "1".
- For details on interrupt request output, see Table 2-6
- When this flag is set, the received ID Field data is displayed in the LIN reception ID register (LAMRID) or the reception data register (RDR).
- This bit is reset when the LAMESRC:LPTERC bit in the clear register is set to "1".

Bit	LIN ID Parity Error Flag Bit	
	Write	Read
0	No effect	There is no error.
1	No effect	An error has occurred.

Notes:

- *This function is effective only in LIN assist mode (LAMCR:LAMEN = 1).*
- *This bit is set "0" when Software reset is done (SCR:UPCL=1).*

[bit12] LSFER: LIN Sync Data Error Flag Bit

- This bit detects whether the Sync Field value is 0x55 when auto baud rate adjustment is disabled (SACSR:AUTE = 0) in slave mode (SCR:MS = 1).
- A LIN Sync Data error is set to "1" when auto baud rate adjustment is disabled (SACSR:AUTE = 0) in slave mode (SCR:MS = 1).
- A reception interrupt request is issued when the LSFER bit and the LSFERIE bit are "1".
- For details on interrupt request output, see Table 2-6
- This bit is reset when the LAMESRC:LSFERC bit in the clear register is set to "1".

Bit	LIN Sync Data Error Flag Bit	
	Write	Read
0	No effect	There is no error.
1	No effect	An error has occurred.

Notes:

- *This function is effective only in LIN assist mode (LAMCR:LAMEN = 1).*
- *This bit is set "0" when Software reset is done (SCR:UPCL=1).*

[bit11] LBSER: LIN Bus Error Flag Bit

- This bit is set to "1" when a LIN bus error occurs.
- A reception interrupt request is issued when the LBSER bit and the LBSER IE bit are "1".
- For details on interrupt request output, see Table 2-6
- Reception data with an error is stored in the reception data register (RDR), when this flag is set in the ID Field and data field.
- This bit is reset when the LAMESRC:LBSERC bit in the clear register is set to "1".

Bit	LIN Bus Error Flag Bit	
	Write	Read
0	No effect	There is no error.
1	No effect	An error has occurred.

Notes:

- *This function is effective only in LIN assist mode (LAMCR:LAMEN = 1).*
- *This bit is set "0" when Software reset is done (SCR:UPCL=1).*

[bit10:8] Reserved: Reserved Bits

9.17. LIN Assist Mode Error Test Register (LAMERT)

The LIN assist mode error test register (LAMERT) sets the pseudo error setting for a framing error, LIN bus error, LIN Sync Data error, LIN ID parity error, and LIN checksum error based on the settings of the key code control bit and pseudo error setting bit.

LIN Assist Mode Error Test Register (LAMERT)

Figure 9-19 shows the bit configuration of the LIN assist mode error test register (LAMERT).

Figure 9-19 Bit Configuration of LIN Assist Mode Error Test Register (LAMERT)

Bit	15 ... 8	7	6	5	4	3	2	1	0
Field	(LAMESR)	KEY1	KEY0	Reserved	LCSERT	LPTERT	LSFERT	LBSERT	FRET
R/W Attribute		R0,W	R0,W	R0,W0	R/W	R/W	R/W	R/W	R/W
Protection Attribute		-	-	-	-	-	-	-	-
Initial Value		0	0	0	0	0	0	0	0

Notes:

- *In manual mode (LAMCR:LAMEN = 0), the setting of pseudo error test mode is prohibited.*
- *This register just can be accessed with 8bit access or 16bit access.*

[bit7:6] KEY1, KEY0: Key Code Control Bits

- This bit is a key code register that enables the following pseudo error settings.
  - Framing error pseudo error setting bit (FRET)
  - LIN bus error pseudo error setting bit (LBSERT)
  - LIN Sync Data error pseudo error setting bit (LSFERT)
  - LIN ID parity error pseudo error setting bit (LPTERT)
  - LIN checksum error pseudo error setting bit (LCSERT)
- To set the pseudo error setting, write a value according to the following procedure.
  - KEY1, KEY0 = 0b00 + write a pseudo error setting value
  - KEY1, KEY0 = 0b01 + write the pseudo error setting value (same as the previous value)
  - KEY1, KEY0 = 0b10 + write the pseudo error setting value (same as the previous value)
  - KEY1, KEY0 = 0b11 + write the pseudo error setting value (same as the previous value)
  - The pseudo error setting value is enabled when the value is written for the 4th time.
- The value written to this register is invalid if you do not follow this setting procedure (if a value is written or read to/from another register during the writing process, if the value written is incorrect, or if a value is read from this register during the writing process).
- To clear a pseudo error setting, follow the same procedure as that for making a pseudo error setting.
- The read value is "0".

Notes:

- *Assist mode is stopped when any of the following errors occurs.*
  - LIN bus error
  - LIN framing error
  - LIN Sync Data error
  - LIN ID parity error
  - LIN checksum error

[bit5] Reserved: Reserved Bit

[bit4] LCSERT: LIN Checksum Error Pseudo Error Setting Bit

- This bit controls the occurrence of a LIN checksum error.
- In assist mode, set this bit to "1" (an error has occurred) before sending response data. When sending a checksum, it is output after being inverted. When an inverted checksum is received, a LIN checksum error occurs and the flag bit (LAMESR:LCSER) is set to "1".
- The pseudo error function is enabled and the error continues to occur until the setting of this bit is disabled (= 0).

Bit	LIN Checksum Error Pseudo Error Setting Bit
0	No error
1	An error has occurred.

[bit3] LPTERT: LIN ID Parity Error Pseudo Error Setting Bit

- This bit controls the occurrence of a LIN ID parity error.
- Set this bit to "1" (an error has occurred) before setting the Lin Break Field (SCR:LBR = 0) when assist mode is set to master (SCR:MS = 0). When sending the ID Field, the ID parity bit (2-bit) is output after being inverted. When the ID Field of an inverted ID parity is received, a LIN ID parity error occurs and the flag bit (LAMESR:LPTER) is set to "1".
- The pseudo error function is enabled and the error continues to occur until the setting of this bit is disabled (= 0).

Bit	LIN ID Parity Error Pseudo Error Setting Bit
0	No error
1	An error has occurred.

[bit2] LSFERT: LIN Sync Data Error Pseudo Error Setting Bit

- This bit controls the occurrence of a LIN Sync Data error.
- Set this bit to "1" (an error has occurred) before setting the Lin Break Field (SCR:LBR = 0) when assist mode is set to master (SCR:MS = 0). All bits in the LIN Sync Field are output after being inverted.
- The pseudo error function is enabled and this bit continues to output the inverted Sync Field data until this bit set is disabled (= 0).

Bit	LIN Sync Data Error Pseudo Error Setting Bit
0	No error
1	An error has occurred.

Note:

- *Set the LIN bus error pseudo error setting bit (LBSERT = 1) at the same time as this bit (LSFERT = 1) is set.*

[bit1] LBSERT: LIN Bus Error Pseudo Error Setting Bit

- This bit controls the occurrence of a LIN bus error.
- When assist mode is set to master and when this bit is set to "1" (an error has occurred) in each field (Sync Field, ID Field, data, and checksum) to which data has been sent, a LIN bus error occurs and the flag bit (LAMESR:LBSER) is set to "1".
- When assist mode is set to slave and when this bit is set to "1" (an error has occurred) in each field (data and checksum) to which a response has been sent, a LIN bus error occurs and the flag bit (LAMESR:LBSER) is set to "1".
- The pseudo error function is enabled and the error continues to occur until the setting of this bit is disabled (= 0).

Bit	LIN Bus Error Pseudo Error Setting Bit
0	No error
1	An error has occurred.

Note:

– *The LIN bus error pseudo error setting cannot be set for the LIN Break Field.*

[bit0] FRET: Framing Error Pseudo Error Setting Bit

- This bit controls the occurrence of a LIN framing error.
- In assist mode, when this bit is set to "1" (an error has occurred) in each field (Sync Field, ID Field, data, and checksum), the stop bit is output after being inverted. If an inverted stop bit is received, a framing error occurs and the flag bit (SSR:FRE) is set to "1".
- The pseudo error function is enabled and the error continues to occur until the setting of this bit is disabled (= 0).

Bit	Framing Error Pseudo Error Setting Bit
0	No error
1	An error has occurred.

9.18. FIFO Control Register 1 (FCR1)

The FIFO control register 1 (FCR1) sets the FIFO test, selects transmission and reception FIFOs, enables transmission FIFO interrupts, and controls the interrupt flag.

Bit Configuration of FIFO Control Register 1 (FCR1)

Figure 9-20 shows the bit configuration of FIFO control register 1 (FCR1).

Figure 9-20 Bit Configuration of FIFO Control Register 1 (FCR1)

Bit	15	14	13	12	11	10	9	8	7 ... 0
Field	Reserved	Reserved	Reserved	FLSTE	FRIIE	FDRQ	FTIE	FSEL	(FCR0)
R/W Attribute	R0,W0	R0,W0	R0,W0	R/W	R/W	R,WX	R/W	R/W	
Protection Attribute	-	-	-	-	-	-	-	-	
Initial Value	0	0	0	0	0	1	0	0	

[bit15:13] Reserved: Reserved Bits

[bit12] FLSTE: Retransmission Data Lost Detection Enable Bit

This bit enables the detection of FIFO retransmission Data Lost Flag Bit (FCR0:FLST).

- This bit is reset when the FCR1C:FLSTEC bit in the clear register is set to "1".
- This bit is set when the FCR1S:FLSTES bit in the set register is set to "1".

"0": Disable FLST bit detection.
"1": Enable FLST bit detection.

Bit	Retransmission Data Lost Detection Enable Bit
0	Disable data lost detection.
1	Enable data lost detection.

Note:

- *To set this bit to "1", first set the FSET bit to "1" and then set this bit to "1".*

[bit11] FRIIE: Reception FIFO Idle Detection Enable Bit

When the reception FIFO contains valid data, this bit enables or disables the detection of continuation of reception idle status for an 8-bit time or longer. If the reception interrupt is enabled (SCR:RIE = 1), the detection of reception idle status triggers a reception interrupt.

- This bit is reset when the FCR1C:FRIIEC bit in the clear register is set to "1".
- This bit is set when the bit FCR1S:FRIIES in the set register is set to "1".

"0": Disable detection of reception idle status.
"1": Enable detection of reception idle status.

Bit	Reception FIFO Idle Detection Enable Bit
0	Disable detection of reception FIFO idle.
1	Enable detection of reception FIFO idle.

Note:

- *To use the reception FIFO, set this bit to "1".*

[bit10] FDRQ: Transmission FIFO Data Request Bit

This bit requests transmission FIFO data.
Value "1" of this bit indicates that transmission data is requested. At this time, a transmission interrupt request is issued if transmission interrupt is enabled (FTIE = 1).

FDRQ set conditions

- When transmission FIFO interrupt control is not used
  - FBYTE (for transmission) = 0 (Transmission FIFO is empty.)
  - Reset of the transmission FIFO
- When transmission FIFO interrupt control is used
  - When ECR:TXBLKEN=0
    - FTICR setting value ≥ FTICR read value (The amount of data in the transmission FIFO is equal to the interrupt trigger level or lower.)
  - When ECR:TXBLKEN=1
    - FTICR setting value ≦ the number of transmission FIFO empty data
  - Reset of the transmission FIFO

FDRQ clearing conditions

- "1" is written to FCR1C:FDRQC.
- For ECR:TXBLKEN = 0:
  - The transmission FIFO is full.
- For ECR.TXBLKEN = 1:
- FTICR setting value ≥ number of vacant data items in the transmission FIFO

Bit	Transmission FIFO Data Request Bit
0	No transmission FIFO data request
1	Transmission FIFO data request issued

Notes:

- *The FSEL bit cannot be changed when this bit is "0".*
- *Writing to this bit is invalid.*
- *Writing "0" to this bit is prohibited when the set value is equal to or less than a setting value. When you write required data to the transmission FIFO after a transmission interrupt occurs, write "1" to the FIFO transmission data request clear bit (FCR1C:FDRQC) to clear the interrupt request.*

[bit9] FTIE: Transmission FIFO Interrupt Enable Bit

This bit enables a transmission FIFO interrupt. If this bit is set to "1", a transmission interrupt occurs when the FDRQ bit is "1".

- This bit is reset when the FCR1C:FTIEC bit in the clear register is set to "1".
- This bit is set when the FCR1S:FTIES bit in the set register is set to "1".

Bit	Transmission FIFO Interrupt Enable Bit
0	Disable transmission FIFO interrupts.
1	Enable transmission FIFO interrupts.

Note:

- *During block transfer , set FTIE to "1" when transmission FIFO is not being used.*

[bit8] FSEL: FIFO Selection Bit

This bit selects the transmission and reception FIFO.

- This bit is reset when the FCR1C:FSELC bit in the clear register is set to "1".
- This bit is set when the FCR1S:FSELS bit in the set register is set to "1".

"0": Allocate as transmission FIFO: FIFO1 and as reception FIFO: FIFO2.
"1": Allocate as transmission FIFO: FIFO2 and as reception FIFO: FIFO1.

Bit	FIFO Selection Bit
0	Transmission FIFO: FIFO1, reception FIFO: FIFO2
1	Transmission FIFO: FIFO2, reception FIFO: FIFO1

Notes:

- *This bit is not cleared by a FIFO reset (FCL2, FCL1 = 1).*
- *To change the value of this bit, disable the FIFO operation (FE2, FE1 = 0) first.*
- *This bit cannot be changed when FDRQ = 0.*
- *Set the FIFO selection bit (FSEL) before setting the FIFO byte register (FBYTE) and the transmission FIFO interrupt control register (FTICR).*
- *Access cannot be performed at the same time as the FIFO byte register (FBYTE).*

9.19. FIFO Control Register 0 (FCR0)

FIFO control register 0 (FCR0) enables/disables FIFO operation, resets a FIFO, saves the read pointer, and makes the retransmission setting.

Bit Configuration of FIFO Control Register 0 (FCR0)

Figure 9-21 shows the bit configuration of FIFO control register 0 (FCR0).

Figure 9-21 Bit Configuration of FIFO Control Register 0 (FCR0)

Bit	15 ... 8	7	6	5	4	3	2	1	0
Field	(FCR1)	Reserved	FLST	FLD	FSET	FCL2	FCL1	FE2	FE1
R/W Attribute		R0,W0	R,WX	R,W	R0,W	R0,W	R0,W	R/W	R/W
Protection Attribute		-	-	-	-	-	-	-	-
Initial Value		0	0	0	0	0	0	0	0

[bit7] Reserved: Reserved Bit

When read, always it can be read "0".

When written, always it is set "0".

[bit6] FLST: FIFO Retransmission Data Lost Flag Bit

This bit indicates that the retransmission data for the transmission FIFO has been lost.

FLST set condition

- Data is written to the FIFO when the FLSTE bit in the FIFO control register 1 (FCR1) is "1" and the write pointer of the transmission FIFO and the read pointer saved by the FSET bit coincide.

FLST clearing conditions

- transmission FIFO reset
  - When FCR1:FSEL=0, set "1" to the FCR0:FCL1
  - When FCR1:FSEL=1, set "1" to the FCR0:FCL2
- Writing "1" to the FSET bit

If "1" is set in this bit, the data pointed to by the read pointer saved with the FSET bit is overwritten. For this reason, re-transmission with the FLD bit cannot be set even if an error occurs. When performing retransmission with this bit set to "1", reset the FIFO and then write the data to the FIFO again.

Bit	FIFO Retransmission Data Lost Flag Bit
0	Data has not been lost.
1	Data has been lost.

[bit5] FLD: FIFO Pointer Reload Bit

This bit reloads, to the read pointer, the data saved to the transmission FIFO by the FSET bit. This bit is used for retransmission in cases such as communication errors.
This bit is set to "0" when retransmission setting has been completed.

- This bit is set when the FCR0S:FLDS bit in the set register is set to "1".

Bit	FIFO Pointer Reload Bit
0	Do not execute reload.
1	Execute reload.

Notes:

- *While this bit is set to "1", reloading to the read pointer is in progress, so writing is prohibited except in the case of a FIFO reset.*
- *Setting this bit to "1" is prohibited when FIFO is enabled or transmission is in progress.*
- *To set the TIE and TBIE bits to "1", first set them to "0" and write "1" to this bit, and then enable the transmission FIFO and set the TIE and TBIE bits to "1".*

[bit4] FSET: FIFO Pointer Saving Bit

This bit stores the read pointer of the transmission FIFO.
If the read pointer is saved before transmission starts and then a communication error occurs, retransmission is possible if the FLST bit is "0".

■ This bit is set when the FCR0S:FSETS bit in the set register is set to "1".

"1": Save the current read pointer value.
"0": No effect.

Bit	FIFO Pointer Saving Bit	
	Write	Read
0	Do not save.	Always read "0"
1	Save the read pointer value.	

Note:

- *Set this bit to "1" when the transmission byte count (FBYTE) is 0.*

[bit3] FCL2: FIFO2 Reset Bit

This bit resets FIFO2.
If this bit is set to "1", the internal status of FIFO2 is initialized.
Only the FCR0:FLST bit is initialized. The values of the other bits in the FCR1/0 registers are retained.

■ This bit is set when the FCR0S:FCL2S bit in the set register is set to "1".

Bit	FIFO2 Reset Bit	
	Write	Read
0	No effect	"0" is always read.
1	Reset FIFO2.	

Notes:

- *Disable transmission/reception first and then execute a FIFO2 reset.*
- *Execute the reset after setting the transmission FIFO interrupt enable bit to "0".*
- *The valid data count for the FBYTE2 register is set to 0.*
- *The TDR register and RDR register are not initialized.*

[bit2] FCL1: FIFO1 Reset Bit

This bit resets FIFO1.

If this bit is set to "1", the internal status of FIFO1 is initialized.

Only the FCR0:FLST bit is initialized. The values of the other bits in the FCR1/0 registers are retained.

- This bit is set when the FCR0S:FCL1S bit in the set register is set to "1".

Bit	FIFO1 Reset Bit	
	Write	Read
0	No effect	"0" is always read.
1	Reset FIFO1.	

Notes:

- *Disable transmission/reception first and then execute a FIFO1 reset.*
- *Execute the reset after setting the transmission FIFO interrupt enable bit to "0".*
- *The valid data count of the FBYTE1 register is set to 0.*
- *The TDR register and RDR register are not initialized.*

[bit1] FE2: FIFO2 Operation Enable Bit

This bit enables or disables the operation of FIFO2.

- Set this bit to "1" when using FIFO2.
- When FIFO2 is specified as the transmission FIFO (FCR1:FSEL=1) and "1" is written to this bit, transmission starts immediately if FIFO2 contains data and LIN interface (v2.1) transmission is enabled (SCR:TXE = 1). At this time, to set the TIE and TBIE bits to "1", first set them to "0" and write "1" to this bit, and then set the TIE and TBIE bits to "1".
- A reception error caused this bit to be cleared to "0" if the FIFO is selected as the reception FIFO by the FSEL bit. After that, this bit cannot be set to "1" unless the reception error is cleared.
- To use FIFO2 as the transmission or reception FIFO, set this bit to "1" or "0" when the transmission or reception buffer is empty ((SSR:TDRE = 1) or (SSR:RDRF = 0)), respectively.
- To use FIFO2 as the reception FIFO, first disable reception (SCR:RXE = 0) and then set this bit to "0" when the reception buffer is empty (SSR:RDRF = 0) and the reception FIFO does not contain valid data (FBYTE2 = 0).

- To use FIFO2 as the reception FIFO, first disable reception (SCR:RXE = 0) and then set this bit to "1" when the reception buffer is empty (SSR:RDRF = 0).
- Disabling FIFO2 does not change the state of FIFO2.
- This bit is reset when the FCR0C:FE2C bit in the clear register is set to "1".
- This bit is set when the FCR0S:FE2S bit in the set register is set to "1".

Bit	FIFO2 Operation Enable Bit
0	Disable FIFO2 operation.
1	Enable FIFO2 operation.

[bit0] FE1: FIFO1 Operation Enable Bit

This bit enables or disables the operation of FIFO1.

- Set this bit to "1" when using FIFO1.
- When FIFO1 is specified as the transmission FIFO and "1" is written to this bit, transmission starts immediately if FIFO1 contains data and LIN interface (v2.1) transmission is enabled (SCR:TXE = 1). At this time, to set the TIE and TBIE bits to "1", first set them to "0" and write "1" to this bit, and then set the TIE and TBIE bits to "1".
- A reception error causes this bit to be cleared to "0" if the FIFO is selected as the reception FIFO by the FSEL bit. After that, this bit cannot be set to "1" unless the reception error is cleared.
- To use FIFO1 as the transmission or reception FIFO, set this bit to "1" or "0" when the transmission or reception buffer is empty ((SSR:TDRE = 1) or (SSR:RDRF = 0)), respectively.
- To use FIFO1 as the reception FIFO, first disable reception (SCR:RXE = 0) and then set this bit to "0" when the reception buffer is empty (SSR:RDRF = 0) and the reception FIFO does not contain valid data (FBYTE1 = 0).
- To use FIFO1 as the reception FIFO, first disable reception (SCR:RXE = 0) and then set this bit to "1" when the reception buffer is empty (SSR:RDRF = 0).
- Disabling FIFO1 does not change the state of FIFO1.
- This bit is reset when the FCR0C:FE1C bit in the clear register is set to "1".
- This bit is set when the FCR0S:FE1S bit in the set register is set to "1".

Bit	FIFO1 Operation Enable Bit
0	Disable FIFO1 operation.
1	Enable FIFO1 operation.

9.20. FIFO Byte Register (FBYTE)

The FIFO byte register (FBYTE) indicates the valid data count for the FIFO. This register also specifies whether a reception interrupt is generated when the predefined amount of data is received by the reception FIFO.

Bit Configuration of FIFO Byte Register (FBYTE)

Figure 9-22 shows the bit configuration of the FIFO byte register (FBYTE).

Figure 9-22 Bit Configuration of FIFO Byte Register (FBYTE)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	FBYTE2								FBYTE1							
R/W Attribute	R, W	R, W	R, W	R, W	R, W	R, W	R, W	R, W	R, W	R, W	R, W	R, W	R, W	R, W	R, W	R, W
Protection Attribute	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The FBYTE register indicates the valid data count for the FIFO. The table below shows the setting conditions for different FCR1:FSEL bit values.

Table 9-2 Data Count Indication

FSEL	FIFO Selection	Data Count Indication
0	FIFO2: Reception FIFO, FIFO1: Transmission FIFO	FIFO2:FBYTE2, FIFO1:FBYTE1
1	FIFO2: Transmission FIFO, FIFO1: Reception FIFO	FIFO2:FBYTE2, FIFO1:FBYTE1

- The initial transfer count value for the FBYTE register is 0x08.
- Set the data count, on which the generation of the reception interrupt flag is based, as the transfer count of the FBYTE for the reception FIFO. When the transferred data count and the data count indication in the FBYTE register coincide, the interrupt flag (RDRF) is set to "1".
- When both of the following conditions are satisfied, the continuation of the reception idle status for 8 baud rate clocks or longer sets the interrupt flag (RDRF) to "1".
  - - Reception FIFO idle detection enable bit (FRIIE) is "1".
  - - The number of data items in the reception FIFO does not reach the transfer count.

  During an 8-clock count, the counter is reset to 0 when RDR is read, and the system starts counting the 8 clocks again. The counter is reset to 0 when the reception FIFO is disabled. If the reception FIFO is enabled when data remains in the reception FIFO, counting restarts.
  - The value of FBYTE of transmission FIFO is incremented by 1 when a transmission data is written to TDR 1 time.
  - The value of FBYTE of reception FIFO is decremented by 1 when a reception data is read from RDR 1 time.
- During block transfer, the value of this register indicates the number of transfer blocks.

[bit15:8] FBYTE2: FIFO2 Data Count Indication Bit

[bit7:0] FBYTE1: FIFO1 Data Count Indication Bit

FBYTE2, FBYTE1	Description
Write	Set the transfer count.
Read	Read valid data count.

Read (valid data count)

Transmission: Data count that was written but not sent to the transmission FIFO
Reception: Data count that has been received but not read by the reception FIFO

Write (transfer count)

Transmission: Set 0x00.
Reception: Set the data count that triggers reception interrupts.

Table 9-3 Data Count Stored in FIFO

FIFO Capacity	Max FBYTE Count	Data Count That Can Be Stored
16 bytes	16	16
32 bytes	32	32
64 bytes	64	64

Notes:

- *Set 0x00 to FBYTE for the transmission FIFO.*
- *When the reception FIFO is not used, the reception block size during block transfer will be "1".*
- *Change the value after disabling the reception operation.*
- *Any setting that would exceed the FIFO capacity is prohibited.*
- *Set the FIFO byte register (FBYTE) after setting the FIFO selection bit (FCR1:FSEL).*
- *The FIFO selection bit (FCR1:FSEL) and the FIFO byte register (FBYTE) cannot be set at the same time.*
- *As the FIFO data count for transmission, the transmission data count that has been written, minus 1, is displayed as the valid data count. This is because an attempt to write transmission data to the TDR register stores the data in the transmission FIFO if the TDR register contains data that has yet to be transmitted. When the data in the TDR register is transmitted, the data in the transmission FIFO that is yet to be transmitted is transferred to the TDR register.*
- *The FIFO data count for reception represents the data count received by the reception FIFO but which has not yet been read. The data count does not include the data being received by the RDR register.*

9.21. Transmission FIFO Interrupt Control Register (FTICR)

The transmission FIFO interrupt control register (FTICR) sets the condition for interrupts triggered by the valid data count of the FIFO transmission.

Transmission FIFO Interrupt Control Register (FTICR)

Figure 9-23 shows the bit configuration of the transmission FIFO interrupt control register (FTICR).

Figure 9-23 Bit Configuration of Transmission FIFO Interrupt Control Register (FTICR)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	FTICR2								FTICR1							
R/W Attribute	R, W	R, W	R, W	R, W	R, W	R, W	R, W	R, W	R, W	R, W	R, W	R, W	R, W	R, W	R, W	R, W
Protection Attribute	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The function of these bits varies depending on the setting of the ECR:TXBLKEN bit. The details are as follows:

- ECR:TXBLKEN = 0
- The FTICR is set the trigger level to interrupt for transmission FIFO's transmission valid data number (remained data number).
- ECR:TXBLKEN = 1

  The FTICR register sets the trigger level for interrupts according to the vacant data count of the transmission FIFO.

  The table below shows setting values of the FCR1:FSEL bit.

Table 9-4 Transmission FIFO Setting by FCR1:FSEL

FSEL	Selection of Transmission FIFO	Transmission FIFO Interrupt Control Register
0	FIFO1	FTICR1
1	FIFO2	FTICR2

- The initial value of the valid data count for triggering interrupts of the FTICR register is 0x00.
- Set the number of data items that generate a transmission interrupt to the FTICR of the transmission FIFO.
  - For ECR:TXBLKEN = 0, if the data count that was set becomes equal to or smaller than, the indication of the valid data count in the transmission FIFO (FTICR or FBYTE), the interrupt flag (FDRQ) is set to "1".
  - For ECR:TXBLKEN = 1, the interrupt flag (FDRQ) is set to "1" when the data count that was set becomes equal to or smaller than the vacant data count of the transmission FIFO. If the data count set to these bits becomes greater than the vacant data count of the transmission FIFO, the interrupt flag (FDRQ) is set to "0".
- Set FTICR so that it satisfies "FTICR ≤ FIFO capacity - 2".
- The read value indicates the valid data count of the FIFO.
- Transmission FIFO: The data count written in the transmission FIFO that has not yet been transmitted.
- Reception FIFO: The data count received by the reception FIFO that has not yet been read.

[bit15:8]FTICR2: FIFO2 Data Count Indication Bit

[bit7:0]FTICR1: FIFO1 Data Count Indication Bit

FTICR2, FTICR1	Description
Write	Set the number of valid data items that generates an interrupt.
Read	Read valid data count.

Notes:

- *Any setting that would exceed the FIFO capacity is prohibited.*
- *The setting value cannot be read.*
- *As the FIFO data count for transmission, the transmission data count that has been written, minus 1, is displayed as the valid data count. This is because an attempt to write transmission data to the TDR register stores the data in the transmission FIFO if the TDR register contains data that has yet to be transmitted. When the data in the TDR register is transmitted, the data in the transmission FIFO that is yet to be transmitted is transferred to the TDR register.*
- *The FIFO data count for reception represents the data count received by the reception FIFO but which has not yet been read. The data count does not include the data being received by the RDR register.*
- *For ECR:TXBLKEN = 0, if block transfer is performed during DMA transfer, only 1 can be set as the block size.*
- *To perform block transfer with ECR:TXBLKEN = 1, set the value of the block size to this register.*

9.22. Extended Control Register (ECR)

The extended control register (ECR) can set block transfer.

Bit Configuration of Extended Control Register (ECR)

Figure 9-24 shows the bit configuration of the extended control register (ECR).

Figure 9-24 Bit Configuration of Extended Control Register (ECR)

Bit	15 ... 8	7	6	5	4	3	2	1	0
Field	(ESR)	Reserved	Reserved	Reserved	EISEL	REIE	TEIE	RXBLKEN	TXBLKEN
R/W Attribute		R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W	R/W
Protection Attribute		-	-	-	-	-			
Initial Value		0	0	0	0	0	0	0	0

[bit7:5] Reserved: Reserved Bits

[bit4] EISEL: Error Interrupt Request Output Selection Bit

This bit selects whether an error interrupt is output from the interrupt request pin or from the error interrupt request pin. For details on interrupt output selection, see Table 2-2, Table 2-3, Table 2-5, and Table 2-6.

Bit	Error Interrupt Request Output Selection Bit
0	Output the reception error interrupt request from the reception interrupt request pin. Output the transmission error interrupt request from the transmission interrupt request pin.
1	Output the reception error interrupt request from the reception error interrupt request pin. Output the transmission error interrupt request from the transmission error interrupt request pin.

[bit3] REIE: Reception Error Interrupt Enable Bit

This bit enables/disables reception error interrupt request output. For details on the target interrupt factors, see Table 2-3 and Table 2-6.

Bit	Reception Error Interrupt Enable Bit
0	Disable reception error interrupts.
1	Enable reception error interrupts.

[bit2] TEIE: Transmission Error Interrupt Enable Bit

This bit enables/disables transmission error interrupt request output. For details on the target interrupt factors, see Table 2-2 and Table 2-5.

Bit	Transmission Error Interrupt Enable Bit
0	Disable transmission error interrupts.
1	Enable transmission error interrupts.

[bit1] RXBLKEN: Reception Block Transfer Setting Bit

This bit sets the DMA transfer mode for reception.

- "0": Perform DMA transfer in demand transfer mode.
- "1": Perform DMA transfer in block transfer mode.

Bit	Reception Block Transfer Setting Bit
0	Demand transfer mode
1	Block transfer mode

[bit0] TXBLKEN: Transmission Block Transfer Setting Bit

This bit sets the DMA transfer mode for transmission.

- "0": Perform DMA transfer in demand transfer mode.
- "1": Perform DMA transfer in block transfer mode.

Bit	Transmission Block Transfer Setting Bit
0	Demand transfer mode
1	Block transfer mode

9.23. Extended Status Register (ESR)

The extended status register (ESR) checks the reception block transfer error flag and the transmission block transfer error flag.

Extended Status Register (ESR)

Figure 9-25 shows the bit configuration of the extended status register (ESR).

Figure 9-25 Bit Configuration of Extended Status Register (ESR)

Bit	15	14	13	12	11	10	9	8	7 ... 0
Field	Reserved	Reserved	Reserved	Reserved	RXUDR	TXOVR	RBERR	TBERR	(ECR)
R/W Attribute	R0,W0	R0,W0	R0,W0	R0,W0	R,WX	R,WX	R,WX	R,WX	
Protection Attribute	-	-	-	-	-	-	-	-	
Initial Value	0	0	0	0	0	0	0	0	

[bit15:12] Reserved: Reserved Bits

[bit11] RXUDR: Reception FIFO Underrun Flag Bit

This bit is set to "1" if the reception FIFO becomes empty due to block transfer, and then a read occurs.

- This bit is reset when the ESRC:RXUDRC bit in the clear register is set to "1".
- For details on interrupt request output, see Table 2-3 for manual mode, Table 2-6 for assist mode.

Bit	Reception FIFO Underrun Flag Bit
0	Indicate that no reception FIFO underrun has occurred.
1	Indicate that reception FIFO underrun has occurred.

Notes:

- *A software reset (SCR:UPCL = 1) resets this bit to "0".*
- *This bit is "0" when the reception block transfer enable bit (ECR:RXBLKEN) is "0".*
- *This bit is set only when the reception FIFO is used.*
- *This bit is "0" when reception FIFO is not available.*

[bit10] TXOVR: Transmission FIFO Overrun Flag Bit

This bit is set to "1" if the transmission FIFO becomes full due to block transfer, and then a write occurs.

- This bit is reset when the ESRC:TXOVRC bit in the clear register is set to "1".
- For details on interrupt request output, see Table 2-2 for manual mode, Table 2-5 for assist mode.

Bit	Transmission FIFO Overrun Flag Bit
0	Indicate that no transmission FIFO overrun has occurred.
1	Indicate that transmission FIFO overrun has occurred.

Notes:

- *A software reset (SCR:UPCL = 1) resets this bit to "0".*
- *This bit is "0" when the transmission block transfer enable bit (ECR:TXBLKEN) is "0".*
- *This bit is set only when the transmission FIFO is used.*

- *This bit is "0" when transmission FIFO is not available.*

[bit9] RBERR: Reception Block Transfer Error Bit

This bit indicates that a block transfer error has occurred during reception.
Suppose that block transfer is performed with a value larger than the threshold specified in the FBYTE register, this bit is set to "1" to indicate a block transfer error. A reception interrupt occurs if reception interrupt is enabled (SCR:RIE = 1).

- This bit is reset when the ESRC:RBERRC bit in the clear register is set to "1".
- For details on interrupt request output, see Table 2-3 for manual mode, Table 2-6 for assist mode.

Bit	Reception Block Transfer Error Bit
0	Indicate that no reception block transfer error has occurred.
1	Indicate that a reception block transfer error has occurred.

Notes:

- *A software reset (SCR:UPCL = 1) resets this bit to "0".*
- *This bit is "0" when the reception block transfer enable bit (ECR:RXBLKEN) is "0" or when the reception interrupt enable bit (SCR:RIE) is "0".*

[bit8] TBERR: Transmission Block Transfer Error Bit

This bit indicates that a block transfer error has occurred during transmission.
Suppose that block transfer is performed with a value larger than the threshold specified in the FTICR register, this bit is set to "1" to indicate a block transfer error to stop DMA transfer for transmission. At this time, a transmission interrupt occurs if transmission interrupts are enabled (SSR:TIE = 1).

- This bit is reset when the ESRC:TBERRC bit in the clear register is set to "1".
- For details on interrupt request output, see Table 2-2 for manual mode, Table 2-5 for assist mode.

Bit	Transmission Block Transfer Error Bit
0	Indicate that no transmission block transfer error has occurred.
1	Indicate that a transmission block transfer error has occurred.

Notes:

- *A software reset (SCR:UPCL = 1) resets this bit to "0".*
- *This bit is "0" while the transmission block transfer enable bit (ECR:TXBLKEN) is "0".*
- *When the transmission FIFO is OFF, and block transfer is performed under any one of the following conditions, this bit is set to "1" to indicate a block transfer error.*
  - A transmission interrupt is enabled (SCR:TIE = 1) and the transmission data register contains data (SSR:TDRE = 0).
  - A transmission interrupt is disabled (SCR:TIE = 0), a transmission bus idle interrupt is enabled (SCR:TBIE = 1), and transmission is in progress (SSR:TBI = 0).
    If a transmission interrupt is enabled (SCR:TIE = 1) and a transmission bus idle interrupt is enabled (SCR:TBIE = 1), the condition for enabling transmission interrupt (SCR:TIE = 1) is given priority.

9.24. Transmission Block Size Register (TBSIZE)

The transmission block size register sets the block size for transmission block transfer.

Bit Configuration of Transmission Block Size Register (TBSIZE)

Figure 9-26 shows the bit configuration of the transmission block size register (TBSIZE).

Figure 9-26 Bit Configuration of Transmission Block Size Register (TBSIZE)

Bit	15 ... 8	7	6	5	4	3	2	1	0
Field	-	TBSIZE							
R/W Attribute		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute		-	-	-	-	-	-	-	-
Initial Value		0	0	0	0	0	0	0	0

The TBSIZE register sets the block size for transmission block transfer.

[bit7:0]TBSIZE

TBSIZE	Description
Write	Set the transmission block count.
Read	Read a setting value.

Notes:

- *Any setting that would exceed the FIFO capacity is prohibited.*
- *Setting this bit to "0" is prohibited.*
- *Any setting that would exceed the setting value of the transmission FIFO interrupt control register (FTICR) is prohibited.*
- *In this register, set the same value as that set for the transmission block size in the DMA controller.*
- *When the transmission FIFO is not used, the transmission block size during block transfer will be "1", regardless of the setting of this register.*

9.25. Serial Control Clear Register (SCRC)

The serial control clear register (SCRC) can clear a bit in the serial control register (SCR).

Note:

- *For details on the operations performed on this register, also see the description of the target register SCR.*

Bit Configuration of Serial Control Clear Register (SCRC)

Figure 9-27 shows the bit configuration of the serial control clear register (SCRC).

Figure 9-27 Bit Configuration of Serial Control Clear Register (SCRC)

Bit	15	14	13	12	11	10	9	8	7 ... 0
Field	Reserved	MSC	Reserved	RIEC	TIEC	TBIEC	RXEC	TXEC	(SMRC)
R/W Attribute	R0, W0	R0,W	R0, W0	R0,W	R0,W	R0,W	R0,W	R0,W	
Protection Attribute	-	-	-	-	-	-	-	-	
Initial Value	0	0	0	0	0	0	0	0	

[bit15] Reserved: Reserved Bit

[bit14] MSC: Clearing the Master/Slave Function Selection Bit.

Writing "1" to this bit resets the SCR:MS to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit13] Reserved: Reserved Bit

[bit12] RIEC: Clearing the Reception Interrupt Enable Bit.

Writing "1" to this bit resets the SCR:RIE to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit11] TIEC: Clearing the Transmission Interrupt Enable Bit

Writing "1" to this bit resets the SCR:TIE to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit10] TBIEC: Clearing the Transmission Bus Idle Interrupt Enable Bit

Writing "1" to this bit resets the SCR:TBIE to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit9] RXEC: Clearing the Reception Operation Enable Bit

Writing "1" to this bit resets the SCR:RXE to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit8] TXEC: Clearing the Transmission Operation Enable Bit

Writing "1" to this bit resets the SCR:TXE to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

9.26. Serial Mode Clear Register (SMRC)

The serial mode clear register (SMRC) can clear a bit in the serial mode register (SMR).

Note:

- *For details on the operations performed on this register, also see the description of the target register SMR.*

Bit Configuration of Serial Mode Clear Register (SMRC)

Figure 9-28 shows the bit configuration of the serial mode clear register (SMRC).

Figure 9-28 Bit Configuration of Serial Mode Clear Register (SMRC)

Bit	15 ... 8	7	6	5	4	3	2	1	0
Field	(SCRC)	Reserved	Reserved	Reserved	WUCRC	SBLC	Reserved	Reserved	SOEC
R/W Attribute		R0, W0	R0, W0	R0, W0	R0,W	R0,W	R0, W0	R0, W0	R0,W
Protection Attribute		-	-	-	-	-	-	-	-
Initial Value		0	0	0	0	0	0	0	0

[bit7:5] Reserved: Reserved Bits

[bit4] WUCRC: Clearing the WAKE UP Control Bit

Writing "1" to this bit resets SMR:WUCR to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit3] SBLC: Clearing the Stop Bit Length Selection Bit

Writing "1" to this bit resets the SMR:SBL to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit2:1] Reserved: Reserved Bits

[bit0] SOEC: Clearing the Serial Data Output Enable Bit

Writing "1" to this bit resets the SMR:SOE to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

9.27. Serial Status Clear Register (SSRC)

The serial status clear register (SSRC) can clear a bit in the serial status register (SSR).

Note:

- *For details on the operations performed on this register, also see the description of the target register SSR.*

Bit Configuration of Serial Status Clear Register (SSRC)

Figure 9-29 shows the bit configuration of the serial status clear register (SSRC).

Figure 9-29 Bit Configuration of Serial Status Clear Register (SSRC)

Bit	15	14	13	12	11	10	9	8	7 ... 0
Field	Reserved	Reserved	LBDC	Reserved	Reserved	Reserved	Reserved	Reserved	(ESCRC)
R/W Attribute	R0, W0	R0, W0	R0,W	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	
Protection Attribute	-	-	-	-	-	-	-	-	
Initial Value	0	0	0	0	0	0	0	0	

[bit15:14] Reserved: Reserved Bits

[bit13] LBDC: Clearing the LIN Break Field Detection Flag Bit

Writing "1" to this bit resets the SSR:LBD to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit12:8] Reserved: Reserved Bits

9.28. Extended Communication Control Clear Register (ESCRC)

The extended communication control clear register (ESCRC) can be used to clear a bit in the extended communication control register (ESCR).

Note:

- *For details on the operations performed on this register, also see the description of the target register ESCR.*

Bit Configuration of Extended Communication Control Clear Register (ESCRC)

Figure 9-30 shows the bit configuration of the extended communication control clear register (ESCRC).

Figure 9-30 Bit Configuration of Extended Communication Control Clear Register (ESCRC)

Bit	15 ... 8	7	6	5	4	3	2	1	0
Field	(SSRC)	Reserved	ESBLC	Reserved	LBIEC	Reserved	Reserved	Reserved	Reserved
R/W Attribute		R0, W0	R0,W	R0, W0	R0,W	R0, W0	R0, W0	R0, W0	R0, W0
ProtectionAttribute		-	-	-	-	-	-	-	-
Initial Value		0	0	0	0	0	0	0	0

[bit7] Reserved: Reserved Bit

[bit6] ESBLC: Clearing the Extended Stop Bit Length Selection Bit

Writing "1" to this bit resets the ESCR:ESBL to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit5] Reserved: Reserved Bit

[bit4] LBIEC: Clearing the LIN Break Field Detection Interrupt Enable Bit

Writing "1" to this bit resets the ESCR:LBIE to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit3:0] Reserved: Reserved Bits

9.29. Serial Auxiliary Control Status Clear Register (SACSRC)

The serial auxiliary control status clear register (SACSRC) can clear the bits in the serial auxiliary control status register (SACSR).

Note:

- *For details on the operations performed on this register, also see the description of the target register SACSR.*

Bit Configuration of Serial Auxiliary Control Status Clear Register (SACSRC)

Figure 9-31 shows the bit configuration of the serial auxiliary control status clear register (SACSRC).

Figure 9-31 Bit Configuration of Serial Auxiliary Control Status Clear Register (SACSRC)

Bit	15	14	13	12	11	10	9	8
Field	STSTC	Reserved	SFDC	SFDEC	AUTEC	Reserved	Reserved	TINTC
R/W Attribute	R0,W	R0, W0	R0,W	R0,W	R0,W	R0, W0	R0, W0	R0,W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	TINTEC	Reserved	TRGEC	Reserved	Reserved	Reserved	Reserved	TMREC
R/W Attribute	R0,W	R0, W0	R0, W	R0, W0	R0, W0	R0, W0	R0, W0	R0,W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

[bit15] STSTC: Clearing the Serial Test Bit

Writing "1" to this bit resets SACSR:STST to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit14] Reserved: Reserved Bit

[bit13] SFDC: Clearing the Sync Field Detection Flag

Writing "1" to this bit resets SACSR:SFD to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit12] SFDEC: Clearing the Sync Field Detection Interrupt Enable Bit

Writing "1" to this bit resets SACSR:SFDE to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit11] AUTEC: Clearing the Auto Baud Rate Adjustment Bit

Writing "1" to this bit resets SACSR:AUTE to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit10:9] Reserved: Reserved Bits

[bit8] TINTC: Clearing the Timer Interrupt Flag

Writing "1" to this bit resets SACSR:TINT to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit7] TINTEC: Clearing the Timer Interrupt Enable Bit

Writing "1" to this bit resets SACSR:TINTE to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit6] Reserved: Reserved Bit

[bit5] TRGEC: Clearing the External Trigger Enable Bit

Writing "1" to this bit resets the SACSR:TRGE to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit4:1] Reserved: Reserved Bits

[bit0] TMREC: Clearing the Serial Timer Enable Bit

Writing "1" to this bit resets SACSR:TMRE to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

9.30. LIN Assist Mode Status Clear Register (LAMSRC)

The LIN assist mode status clear register (LAMSRC) can be used to clear a bit in the LIN assist mode status register (LAMSR).

Note:

- *For details on the operations performed on this register, also see the description of target register LAMSR.*

Bit Configuration of LIN Assist Mode Status Clear Register (LAMSRC)

Figure 9-32 shows the bit configuration of the LIN assist mode status clear register (LAMSRC).

Figure 9-32 Bit Configuration of LIN Assist Mode Status Clear Register (LAMSRC)

Bit	15	14	13	12	11	10	9	8	7 ... 0
Field	Reserved	Reserved	Reserved	Reserved	Reserved	LCSCC	Reserved	LAHCC	(LAMCRC)
R/W Attribute	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	R0,W	R0, W0	R0,W	
Protection Attribute	-	-	-	-	-	-	-	-	
Initial Value	0	0	0	0	0	0	0	0	

[bit15:11] Reserved: Reserved Bits

[bit10] LCSCC: Clearing the LIN Checksum Calculation Completion Flag

Writing "1" to this bit resets LAMSR:LCSC to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit9] Reserved: Reserved Bit

[bit8] LAHCC: Clearing the LIN Auto Header Completion Flag

Writing "1" to this bit resets the LAMSR:LAHC to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

9.31. LIN Assist Mode Control Clear Register (LAMCRC)

The LIN assist mode control clear register (LAMCRC) can be used to clear a bit in the LIN assist mode control register (LAMCR).

Note:

- *For details on the operations performed on this register, also see the description of target register LAMCR.*

Bit Configuration of LIN Assist Mode Control Clear Register (LAMCRC)

Figure 9-33 shows the bit configuration of the LIN assist mode control clear register (LAMCRC).

Figure 9-33 Bit Configuration of LIN Assist Mode Control Clear Register (LAMCRC)

Bit	15 ... 8	7	6	5	4	3	2	1	0
Field	(LAMSRC)	Reserved	Reserved	Reserved	Reserved	Reserved	LCSTYPC	LIDENC	LAMENC
R/W Attribute		R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	R0,W	R0,W	R0,W
Protection Attribute		-	-	-	-	-	-	-	-
Initial Value		0	0	0	0	0	0	0	0

[bit7:3] Reserved: Reserved Bits

[bit2] LCSTYPC: Clearing the LIN Checksum Type Selection Bit

Writing "1" to this bit resets the LAMCR:LCSTYP to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit1] LIDENC: Clearing the LIN ID Register Use Enable Bit

Writing "1" to this bit resets LAMCR:LIDEN to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit0] LAMENC: Clearing the LIN Assist Mode Processing Enable Bit

Writing "1" to this bit resets the LAMCR:LAMEN to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

9.32. LIN Assist Mode Interrupt Enable Clear Register (LAMIERC)

The LIN assist mode interrupt enable clear register (LAMIERC) can be used to clear a bit in the LIN assist mode interrupt enable register (LAMIER).

Note:

- *For details on the operations performed on this register, also see the description of target register LAMIER.*

Bit Configuration of LIN Assist Mode Interrupt Enable Clear Register (LAMIERC)

Figure 9-34 shows the bit configuration of the LIN assist mode interrupt enable clear register (LAMIERC).

Figure 9-34 Bit Configuration of LIN Assist Mode Interrupt Enable Clear Register (LAMIERC)

Bit	15	14	13	12	11	10	9	8	7 ... 0
Field	Reserved	LCSERIEC	LPTERIEC	LSFERIEC	LBSERIEC	LCSCIEC	Reserved	LAHCIEC	Reserved
R/W Attribute	R0, W0	R0,W	R0,W	R0,W	R0,W	R0,W	R0, W0	R0,W	R0,W0
Protection Attribute	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	00000000

[bit15] Reserved: Reserved Bit

[bit14] LCSERIEC: Clearing the LIN Checksum Error Interrupt Enable Bit

Writing "1" to this bit resets LAMIER:LCSERIE to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit13] LPTERIEC: Clearing the LIN ID Parity Error Interrupt Enable Bit

Writing "1" to this bit resets the LAMIER:LPTERIE to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit12] LSFERIEC: Clearing the LIN Sync Data Error Interrupt Enable Bit

Writing "1" to this bit resets the LAMIER:LSFERIE to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit11] LBSERIEC: Clearing the LIN Bus Error Interrupt Enable Bit

Writing "1" to this bit resets LAMIER:LBSERIE to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit10] LCSCIEC: Clearing the LIN Checksum Calculation Completion Interrupt Enable Bit

Writing "1" to this bit resets LAMIER:LCSCIE to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit9] Reserved: Reserved Bit

[bit8] LAHCIEC: Clearing the LIN Auto Header Completion Interrupt Enable Bit

Writing "1" to this bit resets LAMIER:LAHCIE to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit7:0] Reserved: Reserved Bits

All "0" are always read from these bits.

9.33. LIN Assist Mode Error Status Clear Register (LAMESRC)

The LIN assist mode error status clear register (LAMESRC) can be used to clear a bit in the LIN assist mode error status register (LAMESR).

Note:

- *For details on the operations performed on this register, also see the description of target register LAMESR.*

Bit Configuration of LIN Assist Mode Error Status Clear Register (LAMESRC)

Figure 9-35 shows the bit configuration of the LIN assist mode error status clear register (LAMESRC).

Figure 9-35 Bit Configuration of LIN Assist Mode Error Status Clear Register (LAMESRC)

Bit	15	14	13	12	11	10	9	8	7 ... 0
Field	Reserved	LCSERC	LPTERC	LSFERC	LBSERC	Reserved	Reserved	Reserved	Reserved
R/W Attribute	R0, W0	R0,W	R0,W	R0,W	R0,W	R0, W0	R0, W0	R0, W0	R0,W0
Protection Attribute	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	00000000

[bit15] Reserved: Reserved Bit

[bit14] LCSERC: Clearing the LIN Checksum Error Flag

Writing "1" to this bit resets LAMESR:LCSER to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit13] LPTERC: Clearing the LIN ID Parity Error Flag

Writing "1" to this bit resets LAMESR:LPTER to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit12] LSFERC: Clearing the LIN Sync Data Error Flag

Writing "1" to this bit resets LAMESR:LSFER to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit11] LBSERC: Clearing the LIN Bus Error Flag

Writing "1" to this bit resets LAMESR:LBSER to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit10:0] Reserved Bits

All "0" are always read from these bits.

All "0" are always written to these bits.

9.34. FIFO Control Clear Register 1 (FCR1C)

FIFO control clear register 1 (FCR1C) can clear a bit in FIFO control register 1 (FCR1).

Note:

- *For details on the operations performed on this register, also see the description of target register FCR1.*

Bit Configuration of FIFO Control Clear Register 1 (FCR1C)

Figure 9-36 shows the bit configuration of FIFO control clear register 1 (FCR1C).

Figure 9-36 Bit Configuration of FIFO Control Clear Register 1 (FCR1C)

Bit	15	14	13	12	11	10	9	8	7 ... 0
Field	Reserved	Reserved	Reserved	FLSTEC	FRIIEC	FDRQC	FTIEC	FSELC	(FCR0C)
R/W Attribute	R0, W0	R0, W0	R0, W0	R0,W	R0,W	R0,W	R0,W	R0,W	
Protection Attribute	-	-	-	-	-	-	-	-	
Initial Value	0	0	0	0	0	0	0	0	

[bit15:13] Reserved: Reserved Bits

[bit12] FLSTEC: Clearing the Retransmission Data Lost Detection Enable Bit

Writing "1" to this bit resets FCR1:FLSTE to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit11] FRIIEC: Clearing the Reception FIFO Idle Detection Enable Bit

Writing "1" to this bit resets FCR1:FRIIE to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit10] FDRQC: Clearing the Transmission FIFO Data Request Bit

Writing "1" to this bit resets FCR1:FDRQ to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit9] FTIEC: Clearing the Transmission FIFO Interrupt Enable Bit

Writing "1" to this bit resets FCR1:FTIE to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit8] FSELC: Clearing the FIFO Selection Bit

Writing "1" to this bit resets FCR1:FSEL to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

9.35. FIFO Control Clear Register 0 (FCR0C)

FIFO control clear register 0 (FCR0C) can clear a bit in FIFO control register 0 (FCR0).

Note:

- *For details on the operations performed on this register, also see the description of the target register FCR0.*

Bit Configuration of FIFO Control Clear Register 0 (FCR0C)

Figure 9-37 shows the bit configuration of FIFO control clear register 0 (FCR0C).

Figure 9-37 Bit Configuration of FIFO Control Clear Register 0 (FCR0C)

Bit	15 ... 8	7	6	5	4	3	2	1	0
Field	(FCR1C)	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	FE2C	FE1C
R/W Attribute		R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	R0,W	R0,W
Protection Attribute		-	-	-	-	-	-	-	-
Initial Value		0	0	0	0	0	0	0	0

[bit7:2] Reserved: Reserved Bits

[bit1] FE2C: Clearing the FIFO2 Operation Enable Bit

Writing "1" to this bit resets FCR0:FE2 to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit0] FE1C: Clearing the FIFO1 Operation Enable Bit

Writing "1" to this bit resets FCR0:FE1 to "0".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

9.36. Extended Status Clear Register (ESRC)

The extended status clear register (ESRC) can clear a bit in the extended status register (ESR).

Extended Status Clear Register (ESRC)

Figure 9-38 shows the bit configuration of the extended status clear register (ESRC).

Figure 9-38 Bit Configuration of Extended Status Clear Register (ESRC)

Bit	15	14	13	12	11	10	9	8	7 ... 0
Field	Reserved	Reserved	Reserved	Reserved	RXUDRC	TXOVRC	RBERRC	TBERRC	-
R/W Attribute	R0, W0	R0, W0	R0, W0	R0, W0	R0,W	R0,W	R0,W	R0,W	
Protection Attribute	-	-	-	-	-	-	-	-	
Initial Value	0	0	0	0	0	0	0	0	

[bit15:12] Reserved: Reserved Bits

[bit11] RXUDRC: Reception FIFO Underrun Flag Clear Bit

Writing "1" to this bit resets ESR:RXUDR to "0".
Writing "0" to this bit is invalid.
"0" is always read from this bit.

[bit10] TXOVRC: Transmission FIFO Overrun Flag Clear Bit

Writing "1" to this bit resets ESR:TXOVR to "0".
Writing "0" to this bit is invalid.
"0" is always read from this bit.

[bit9] RBERRC: Reception Block Transfer Error Clear Bit

Writing "1" to this bit resets ESR:RBERR to "0".
Writing "0" to this bit is invalid.
"0" is always read from this bit.

[bit8] TBERRC: Transmission Block Transfer Error Clear Bit

Writing "1" to this bit resets ESR:TBERR to "0".
Writing "0" to this bit is invalid.
"0" is always read from this bit.

9.37. Serial Control Set Register (SCRS)

The serial control set register (SCRS) can set a bit in the serial control register (SCR).

Note:

- *For details on the operations performed on this register, also see the description of target register SCR.*

Bit Configuration of Serial Control Set Register (SCRS)

Figure 9-39 shows the bit configuration of the serial control set register (SCRS).

Figure 9-39 Bit Configuration of Serial Control Set Register (SCRS)

Bit	15	14	13	12	11	10	9	8	7 ... 0
Field	UPCLS	MSS	LBRS	RIES	TIES	TBIES	RXES	TXES	(SMRS)
R/W Attribute	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	
Protection Attribute	-	-	-	-	-	-	-	-	
Initial Value	0	0	0	0	0	0	0	0	

[bit15] UPCLS: Setting the Programmable Clear Bit

Writing "1" to this bit sets SCR:UPCL to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit14] MSS: Setting the Master/Slave Function Selection Bit

Writing "1" to this bit sets SCR:MS to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit13] LBRS: Setting the LIN Break Field Setting Bit (Effective Only for the Master Operation)

Writing "1" to this bit sets SCR:LBR to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit12] RIES: Setting the Reception Interrupt Enable Bit

Writing "1" to this bit sets SCR:RIE to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit11] TIES: Setting the Transmission Interrupt Enable Bit

Writing "1" to this bit sets SCR:TIE to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit10] TBIES: Setting the Transmission Bus Idle Interrupt Enable Bit

Writing "1" to this bit sets SCR:TBIE to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit9] RXES: Setting the Reception Operation Enable Bit

Writing "1" to this bit sets the SCR:RXE to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit8] TXES: Setting the Transmission Operation Enable Bit

Writing "1" to this bit sets SCR:TXE to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

9.38. Serial Mode Set Register (SMRS)

The serial mode set register (SMRS) can set a bit in the serial mode register (SMR).

Note:

- *For details on the operations performed on this register, also see the description of target register SMR.*

Bit Configuration of Serial Mode Set Register (SMRS)

Figure 9-40 shows the bit configuration of the serial mode set register (SMRS).

Figure 9-40 Bit Configuration of Serial Mode Set Register (SMRS)

Bit	15 ... 8	7	6	5	4	3	2	1	0
Field	(SCRS)	Reserved	Reserved	Reserved	WUCRS	SBLS	Reserved	Reserved	SOES
R/W Attribute		R0, W0	R0, W0	R0, W0	R0,W	R0,W	R0, W0	R0, W0	R0,W
Protection Attribute		-	-	-	-	-	-	-	-
Initial Value		0	0	0	0	0	0	0	0

[bit7:5] Reserved: Reserved Bits

[bit4] WUCRS: Setting the WAKE UP Control Bit

Writing "1" to this bit sets SMR:WUCR to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit3] SBLS: Setting the Stop Bit Length Selection Bit

Writing "1" to this bit sets the SMR:SBL to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit2:1] Reserved: Reserved Bits

[bit0] SOES: Setting the Serial Data Output Enable Bit

Writing "1" to this bit sets SMR:SOE to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

9.39. Serial Status Set Register (SSRS)

The serial status set register (SSRS) can set a bit in the serial status register (SSR).

Note:

- *For details on the operations performed on this register, also see the description of target register SSR.*

Bit Configuration of Serial Status Set Register (SSRS)

Figure 9-41 shows the bit configuration of the serial status set register (SSRS).

Figure 9-41 Bit Configuration of Serial Status Set Register (SSRS)

Bit	15	14	13	12	11	10	9	8	7 ... 0
Field	RECS	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	(ESCRS)
R/W Attribute	R0,W	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	
Protection Attribute	-	-	-	-	-	-	-	-	
Initial Value	0	0	0	0	0	0	0	0	

[bit15] RECS: Setting the Reception Error Flag Clear Bit

Writing "1" to this bit sets SSR:REC to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit14:8] Reserved: Reserved Bits

9.40. Extended Communication Control Set Register (ESCRS)

The extended communication control set register (ESCRS) can be used to set a bit in the extended communication control register (ESCR).

Note:

- *For details on the operations performed on this register, also see the description of the target register ESCR.*

Bit Configuration of Extended Communication Control Set Register (ESCRS)

Figure 9-42 shows the bit configuration of the extended communication control set register (ESCRS).

Figure 9-42 Bit Configuration of Extended Communication Control Set Register (ESCRS)

Bit	15 ... 8	7	6	5	4	3	2	1	0
Field	(SSRS)	Reserved	ESBLS	Reserved	LBIES	Reserved	Reserved	Reserved	Reserved
R/W Attribute		R0, W0	R0,W	R0, W0	R0,W	R0, W0	R0, W0	R0, W0	R0, W0
Protection Attribute		-	-	-	-	-	-	-	-
Initial Value		0	0	0	0	0	0	0	0

[bit7] Reserved: Reserved Bit

[bit6] ESBLS: Setting the Extended Stop Bit Length Selection Bit

Writing "1" to this bit sets the ESCR:ESBL to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit5] Reserved: Reserved Bit

[bit4] LBIES: Setting the LIN Break Field Detection Interrupt Enable Bit

Writing "1" to this bit sets the ESCR:LBIE to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit3:0] Reserved: Reserved Bits

9.41. Serial Auxiliary Control Status Set Register (SACSRS)

The serial auxiliary control status set register (SACSRS) can set the bits in the serial auxiliary control status register (SACSR).

Note:

- *For details on the operations performed on this register, also see the description of target register SACSR.*

Bit Configuration of Serial Auxiliary Control Status Set Register (SACSRS)

Figure 9-43 shows the bit configuration of the serial auxiliary control status set register (SACSRS).

Figure 9-43 Bit Configuration of Serial Auxiliary Control Status Set Register (SACSRS)

Bit	15	14	13	12	11	10	9	8
Field	STSTS	Reserved	Reserved	SFDES	AUTES	Reserved	Reserved	Reserved
R/W Attribute	R0,W	R0, W0	R0, W0	R0,W	R0,W	R0, W0	R0, W0	R0, W0
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	TINTES	Reserved	TRGES	Reserved	Reserved	Reserved	Reserved	TMRES
R/W Attribute	R0,W	R0, W0	R0,W	R0, W0	R0, W0	R0, W0	R0, W0	R0,W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

[bit15] STSTS: Setting the Serial Test Bit

Writing "1" to this bit sets SACSR:STST to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit14:13] Reserved: Reserved Bits

[bit12] SFDES: Setting the Sync Field Detection Interrupt Enable Bit

Writing "1" to this bit sets the SACSR:SFDE to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit11] AUTES: Setting the Auto Baud Rate Adjustment Bit

Writing "1" to this bit sets SACSR:AUTE to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit10:8] Reserved: Reserved Bits

[bit7] TINTES: Setting the Timer Interrupt Enable Bit

Writing "1" to this bit sets SACSR:TINTE to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit6] Reserved: Reserved Bit

[bit5] TRGES: Setting the External Trigger Enable Bit

Writing "1" to this bit sets SACSR:TRGE to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit4:1] Reserved Bits

[bit0] TMRES: Setting the Serial Timer Enable Bit

Writing "1" to this bit sets SACSR:TMRE to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

9.42. LIN Assist Mode Control Set Register (LAMCRS)

The LIN assist mode control set register (LAMCRS) can be used to set a bit in the LIN assist mode control register (LAMCR).

Note:

- *For details on the operations performed on this register, also see the description of target register LAMCR.*

Bit Configuration of LIN Assist Mode Control Set Register (LAMCRS)

Figure 9-44 shows the bit configuration of the LIN assist mode control set register (LAMCRS).

Figure 9-44 Bit Configuration of LIN Assist Mode Control Set Register (LAMCRS)

Bit	15 ... 8	7	6	5	4	3	2	1	0
Field	Reserved	Reserved	Reserved	Reserved	Reserved	LTDRCLS	LCSTYPS	LIDENS	LAMENS
R/W Attribute	R0,W0	R0, W0	R0, W0	R0, W0	R0, W0	R0,W	R0,W	R0,W	R0,W
Protection Attribute	-	-	-	-	-	-	-	-	-
Initial Value	00000000	0	0	0	0	0	0	0	0

[bit15:4] Reserved Bits

[bit3] LTDRCLS: Setting the LIN Transmission Data Register Clear Bit

Writing "1" to this bit sets LAMCR:LTDRCL to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit2] LCSTYPS: Setting the LIN Checksum Type Selection Bit

Writing "1" to this bit sets LAMCR:LCSTYP to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit1] LIDENS: Setting the LIN ID Register Use Enable Bit

Writing "1" to this bit sets LAMCR:LIDEN to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit0] LAMENS: Setting the LIN Assist Mode Processing Enable Bit

Writing "1" to this bit sets LAMCR:LAMEN to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

9.43. LIN Assist Mode Interrupt Enable Set Register (LAMIERS)

The LIN assist mode interrupt enable set register (LAMIERS) can be used to set a bit in the LIN assist mode interrupt enable register (LAMIER).

Note:

- *For details on the operations performed on this register, also see the description of target register LAMIER.*

Bit Configuration of LIN Assist Mode Interrupt Enable Set Register (LAMIERS)

Figure 9-45 shows the bit configuration of the LIN assist mode interrupt enable set register (LAMIERS).

Figure 9-45 Bit Configuration of LIN Assist Mode Interrupt Enable Set Register (LAMIERS)

Bit	15	14	13	12	11	10	9	8	7 ... 0
Field	Reserved	LCSERIES	LPTERIES	LSFERIES	LBSERIES	LCSCIES	Reserved	LAHCIES	Reserved
R/W Attribute	R0, W0	R0,W	R0,W	R0,W	R0,W	R0,W	R0, W0	R0,W	R0,W0
Protection Attribute	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	00000000

[bit15] Reserved: Reserved Bit

[bit14] LCSERIES: Setting the LIN Checksum Error Interrupt Enable Bit

Writing "1" to this bit sets LAMIER:LCSERIE to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit13] LPTERIES: Setting the LIN ID Parity Error Interrupt Enable Bit

Writing "1" to this bit sets LAMIER:LPTERIE to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit12] LSFERIES: Setting the LIN Sync Data Error Interrupt Enable Bit

Writing "1" to this bit sets LAMIER:LSFERIE to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit11] LBSERIES: Setting the LIN Bus Error Interrupt Enable Bit

Writing "1" to this bit sets LAMIER:LBSERIE to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit10] LCSCIES: Setting the LIN Checksum Calculation Completion Interrupt Enable Bit

Writing "1" to this bit sets LAMIER:LCSCIE to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit9] Reserved: Reserved Bit

[bit8] LAHCIES: Setting the LIN Auto Header Completion Interrupt Enable Bit

Writing "1" to this bit sets LAMIER:LAHCIE to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit7:bit0] Reserved Bits

9.44. FIFO Control Set Register 1 (FCR1S)

FIFO control set register 1 (FCR1S) can set a bit in FIFO control register 1 (FCR1).

Note:

- *For details on the operations performed on this register, also see the description of target register FCR1.*

Bit Configuration of FIFO Control Set Register 1 (FCR1S)

Figure 9-46 shows the bit configuration of FIFO control set register 1 (FCR1S).

Figure 9-46 Bit Configuration of FIFO Control Set Register 1 (FCR1S)

Bit	15	14	13	12	11	10	9	8	7 ... 0
Field	Reserved	Reserved	Reserved	FLSTES	FRIIES	Reserved	FTIES	FSELS	(FCR0S)
R/W Attribute	R0, W0	R0, W0	R0, W0	R0,W	R0,W	R0, W0	R0,W	R0,W	
Protection Attribute	-	-	-	-	-	-	-	-	
Initial Value	0	0	0	0	0	0	0	0	

[bit15:13] Reserved: Reserved Bits

[bit12] FLSTES: Setting the Retransmission Data Lost Detection Enable Bit

Writing "1" to this bit sets FCR1:FLSTE to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit11] FRIIES: Setting the Reception FIFO Idle Detection Enable Bit

Writing "1" to this bit sets FCR1:FRIIE to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit10] Reserved: Reserved Bit

[bit9] FTIES: Setting the Transmission FIFO Interrupt Enable Bit

Writing "1" to this bit sets FCR1:FTIE to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit8] FSELS: Setting the FIFO Selection Bit

Writing "1" to this bit sets FCR1:FSEL to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

9.45. FIFO Control Set Register 0 (FCR0S)

FIFO control set register 0 (FCR0S) can set a bit in FIFO control register 0 (FCR0).

Note:

- *For details on the operations performed on this register, also see the description of target register FCR0.*

Bit Configuration of FIFO Control Set Register 0 (FCR0S)

Figure 9-47 shows the bit configuration of FIFO control set register 0 (FCR0S).

Figure 9-47 Bit Configuration of FIFO Control Set Register 0 (FCR0S)

Bit	15 ... 8	7	6	5	4	3	2	1	0
Field	(FCR1S)	Reserved	Reserved	FLDS	FSETS	FCL2S	FCL1S	FE2S	FE1S
R/W Attribute		R0, W0	R0, W0	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
Protection Attribute		-	-	-	-	-	-	-	-
Initial Value		0	0	0	0	0	0	0	0

[bit7:6] Reserved: Reserved Bits

[bit5] FLDS: Setting the FIFO Pointer Reload Bit

Writing "1" to this bit sets FCR0:FLD to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit4] FSETS: Setting the FIFO Pointer Saving Bit

Writing "1" to this bit sets FCR0:FSET to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit3] FCL2S: Setting the FIFO2 Reset Bit

Writing "1" to this bit sets FCR0:FCL2 to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit2] FCL1S: Setting the FIFO1 Reset Bit

Writing "1" to this bit sets FCR0:FCL1 to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit1] FE2S: Setting the FIFO2 Operation Enable Bit

Writing "1" to this bit sets FCR0:FE2 to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

[bit0] FE1S: Setting the FIFO1 Operation Enable Bit

Writing "1" to this bit sets FCR0:FE1 to "1".

Writing "0" to this bit is invalid.

"0" is always read from this bit.

CHAPTER 40: Base Timer

This chapter explains the functions and operations of the base timer.

1. Overview of the Base Timer
2. Block Diagrams of the Base Timer
3. Operations of the Base Timer
4. 32-Bit Mode Operation
5. Interrupts from the Base Timer
6. Start of DMA Controller (DMAC)
7. Registers of the Base Timer
8. Notes on Using the Base Timer
9. Base Timer Description by Function Mode

BT-TXXPT03P01R01L08-E1-XX

1. Overview of the Base Timer

Only one of the following timer functions can be selected for the base timer in the FMD2 to FMD0 bit settings in the timer control register (TMCR): reset mode, 16-bit PWM timer, 16-bit PPG timer, 16-/32-bit reload timer, and 16-/32-bit PWC timer. This section provides an overview of the various selectable timers.

Relationship between Mode Settings and Various Timer Functions

FMD2 to FMD0 Bit Settings in the Timer Control Register (TMCR)	Function
000	Reset mode
001	16-bit PWM timer
010	16-bit PPG timer
011	16/32-bit reload timer
100	16/32-bit PWC timer

Reset Mode

Reset mode is the state in which the base timer macros have been reset (to the initial values in each register). To use another timer function or switch the T32 bit setting, first enter this mode and then set the other timer function or the T32 bit. However, after a macro reset, it is possible to set a timer function and the T32 bit without entering this mode.

16-Bit PWM Timer

The 16-bit PWM timer operates with the cycle of 16-bit.
This timer consists of a 17-bit down counter that takes into account the start delay period, a 16-bit data register with a cycle setting buffer, a 16-bit compare register with a duty setting buffer, a start delay value setting register, an ADC trigger value setting register, and a pin controller.

The 16-bit PWM timer is described to be 16-bit down counter where irrelevant to the start delay control.

The registers with buffers store the cycle and duty data, enabling rewriting while the timer is operating.

The counter clock of the 17-bit down counter can be selected from 12 types of internal clocks (internal clock divided by 1/2/4/8/16/32/64/128/256/512/1024/2048) and 3 types of external clocks (rising-edge, falling-edge, and both-edges detection).

One-shot mode and continuous mode can be selected. In one-shot mode, counting stops when an underflow occurs. In continuous mode, counting is repeated following a reload.

The start of the 16-bit PWM timer can be selected from a software trigger and three types of external events (rising-edge, falling-edge, and both-edges detection).

The start delay function can delay the PWM control start time after trigger input.

An ADC trigger signal is output when the ADC trigger value setting register matches the count value of the 16-bit down counter.

16-Bit PPG Timer

This timer consists of a 16-bit down counter, a 16-bit data register for the H width setting, a 16-bit data register for the L width setting, and a pin controller.

The count clock of the 16-bit down counter can be selected from 12 types of internal clocks (internal clock divided by 1/2/4/8/16/32/64/128/256/512/1024/2048) and 3 types of external clocks (rising-edge, falling-edge, and both-edges detection).

One-shot mode and continuous mode can be selected. In one-shot mode, counting stops when an underflow occurs. In continuous mode, counting is repeated following a reload.

The start of the 16-bit PPG timer can be selected from a software trigger and three types of external events (rising-edge, falling-edge, and both-edges detection).

16-/32-Bit Reload Timer

This timer consists of a 16-bit down counter, a 16-bit reload register, and a pin controller.

The count clock of the 16-bit down counter can be selected from 12 types of internal clocks (internal clock divided by 1/2/4/8/16/32/64/128/256/512/1024/2048) and 3 types of external clocks (rising-edge, falling-edge, and both-edges detection).

One-shot mode and continuous mode can be selected. In one-shot mode, counting stops when an underflow occurs. In continuous mode, counting is repeated following a reload.

An external factor pin can be selected as a trigger input function or a gate function when the 16/32-bit reload timer is operating.

If the trigger input function is selected, the start of the 16/32-bit reload timer can be selected from a software trigger and three types of external events (rising-edge, falling-edge, and both-edges detection).

If the gate function is selected, the 16/32-bit reload timer counts only while a valid level is being input to the external factor pin.

16-/32-Bit PWC Timer

This timer consists of a 16-bit up counter, measurement input pins, and a control register.

With the input of external pulses, the timer measures the time between events.

The reference count clock can be selected from 12 types of internal clocks (divided by 1/2/4/8/16/32/64/128/256/512/1024/2048).

Each measurement mode: H pulse width (rising to falling) / L pulse width (falling to rising)
Rising cycle (rising to rising) / falling cycle (falling to falling)
Edge-to-edge measurement (rising or falling to falling or rising)

An interrupt request can be generated at the measurement end time.

The measurement is 1-time only or continuous. Either can be selected.

Interrupt request and DMA request

Interrupt request of this module can be used for purposes below:

1. Interrupt request to CPU (via Interrupt Controller)
2. DMA request to DMA Controller

To use the interrupt as interrupt request to CPU, following configuration is necessary:

- Enable the interrupt request in this module
- Enable the interrupt channel of the interrupt in Interrupt Controller

To use the interrupt as DMA request to DMA Controller, following configuration is necessary:

- Enable the interrupt request in this module
- Select the DMA channel from this module as DMA client in DMA Controller

Note:

- *Interrupt request that supports for DMA request is product specification. Please refer to the product's HWM for detail.*

2. Block Diagrams of the Base Timer

Figure 2-1 to Figure 2-4 are block diagrams of the base timer in each mode.

Figure 2-1 Block Diagram of the 16-Bit PWM Timer

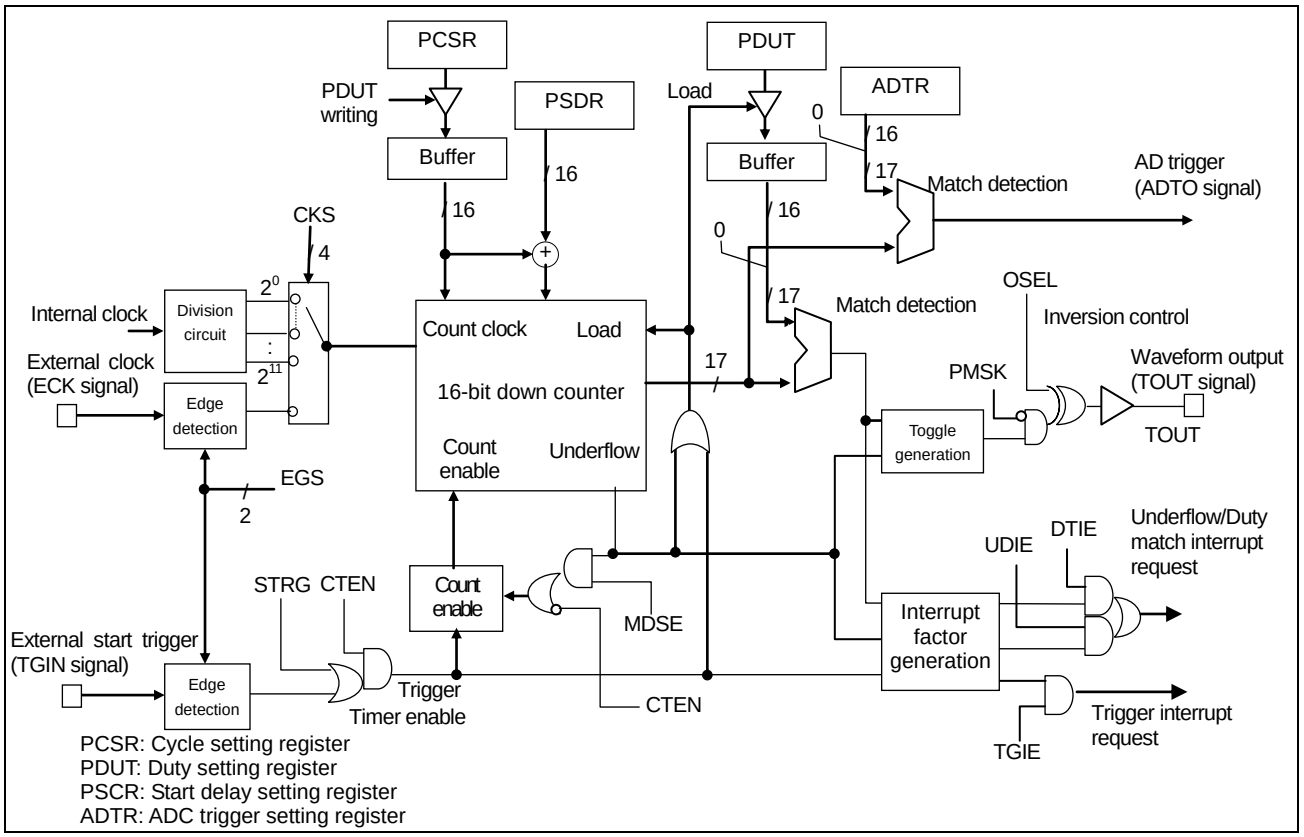


Figure 2-2 Block Diagram of the 16-Bit PPG Timer

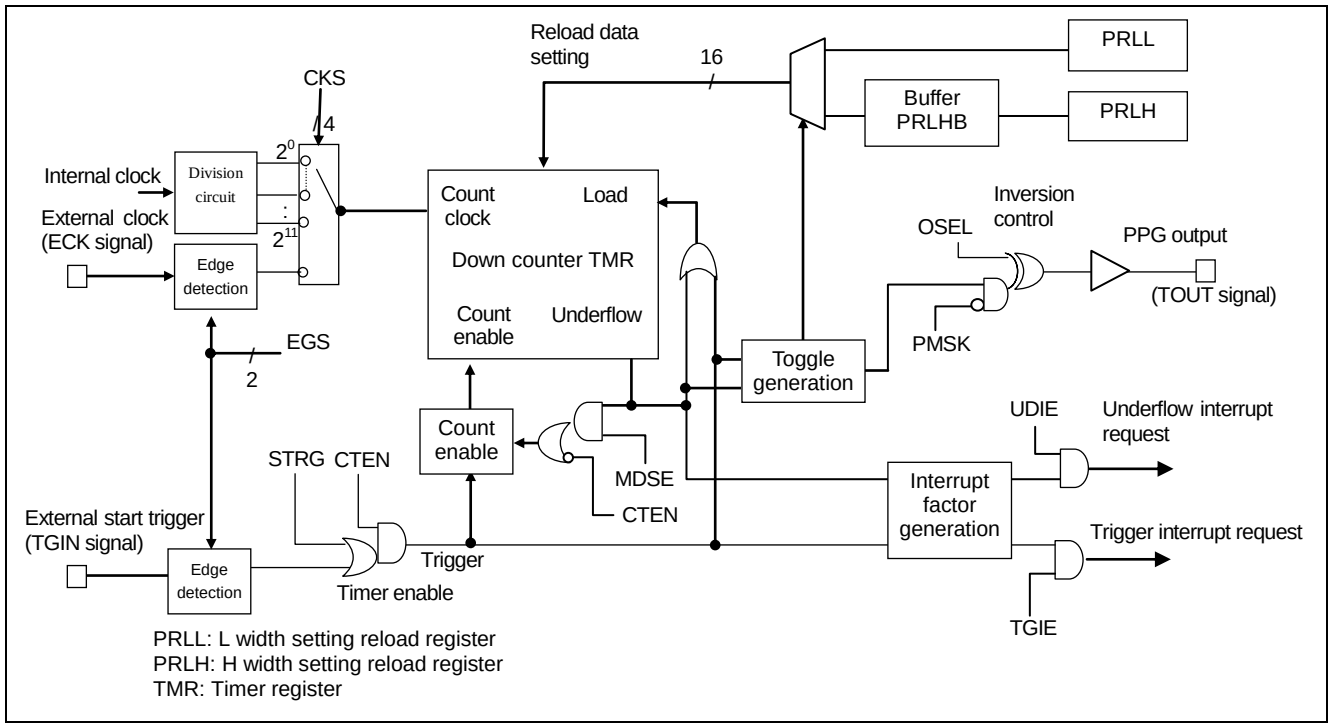


Figure 2-3 Block Diagram of the 16-/32-Bit Reload Timer (ch.1, ch.0)

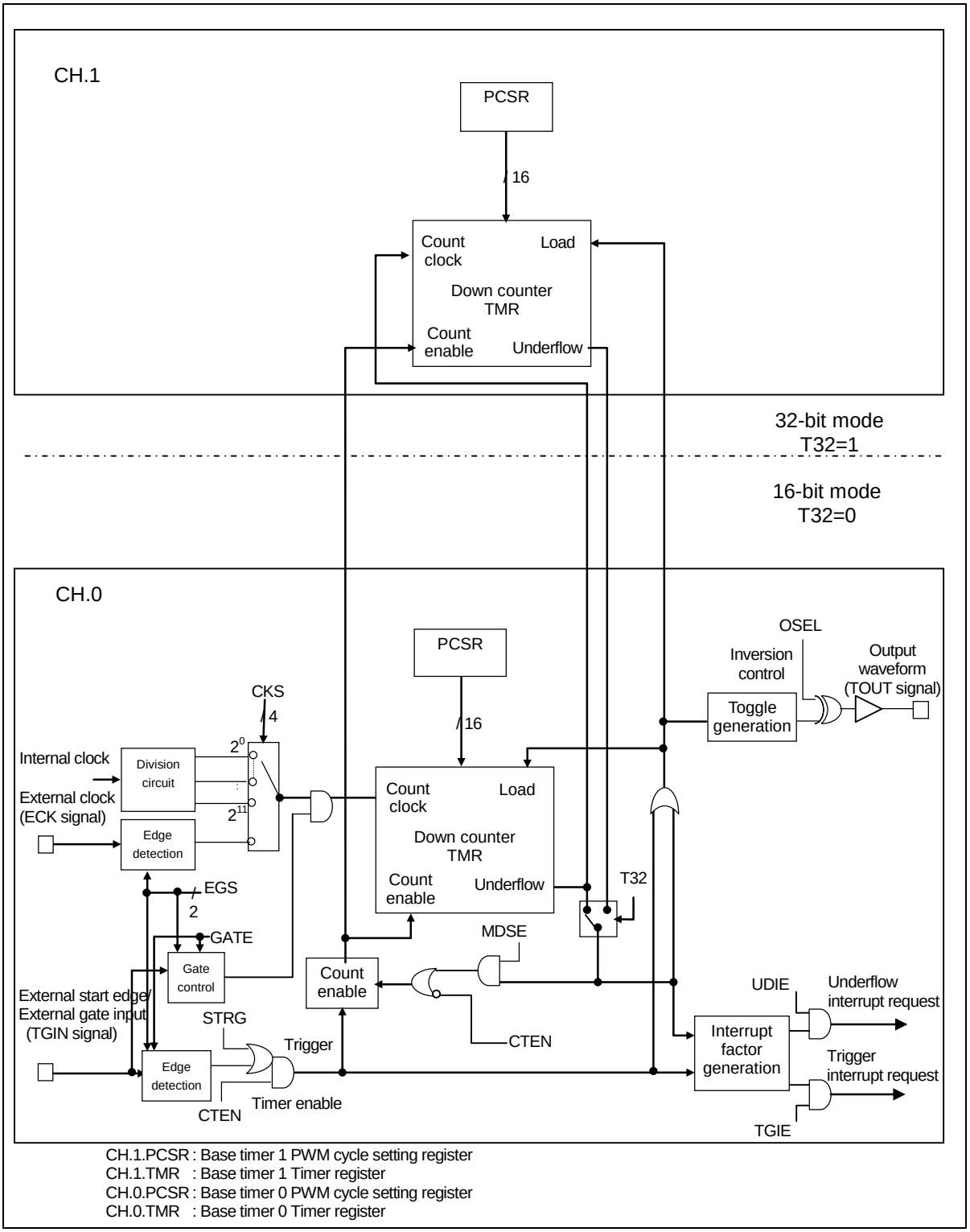
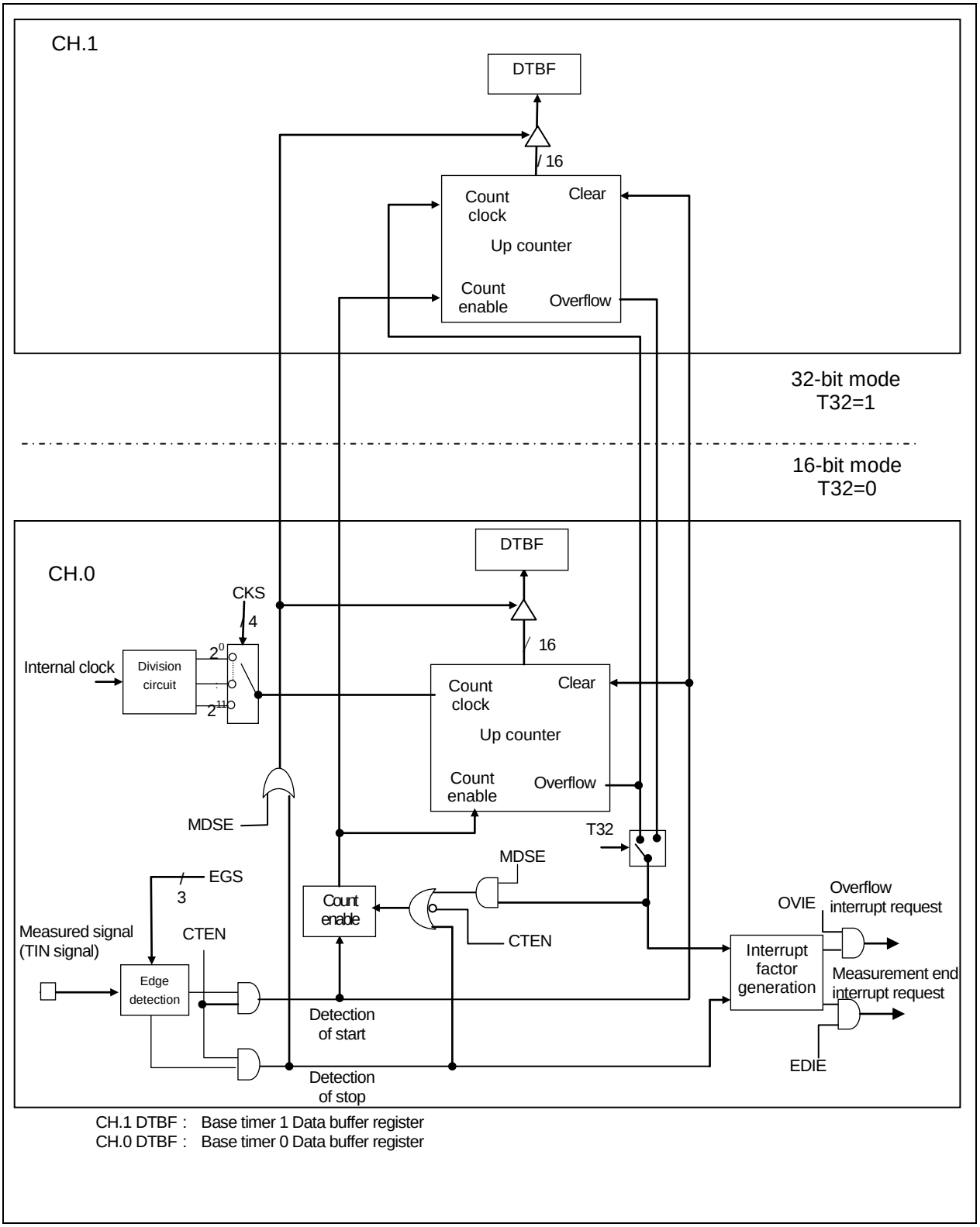


Figure 2-4 Block Diagram of the 16/32-Bit PWC Timer (ch.1, ch.0)



3. Operations of the Base Timer

This section explains the operations of the base timer.

Operations of Base Timer

Reset Mode

Reset mode is the state in which the base timer macros have been reset (to the initial values in each register). To use another timer function or switch the T32 bit setting, first enter this mode and then set the other timer function or the T32 bit. However, after a macro reset, it is possible to set a timer function and the T32 bit without entering this mode. If this mode is set for the even-numbered channel when 32-bit mode is set, the odd-numbered channel is reset at the same time, so reset mode need not be set for the odd-numbered channel.

16-Bit PWM Timer

The 16-bit PWM timer starts counting down from the set cycle value upon trigger activation. The first output at that time is the L level. If the 16-bit down counter matches the set value in the duty setting register, the output is inverted to the H level. Then, the output is inverted to the L level again when the counter underflows. Thus, this timer can generate a waveform with an arbitrary cycle and duty.

16-Bit PPG Timer

The 16-bit PPG timer starts counting down from the set value in the L width setting reload register upon trigger activation. The first output at that time is the L level. The output is inverted to the H level when the counter underflows. Subsequently, the counter starts counting down from the set value in the H width setting reload register. The output is inverted to the L level when the counter underflows. Thus, this timer can generate a waveform with an arbitrary L width and H width.

16-Bit Reload Timer

The 16-bit reload timer starts counting down from the set cycle value upon trigger activation. An interrupt flag is set to "1" when the 16-bit down counter underflows. The output level is either toggle output or pulse output. Toggle output is the output inverted for each underflow by the MDSE bit setting. Pulse output is the output of "H" due to the start of counting or "L" due to an underflow.

32-Bit Reload Timer

With the same basic operation as the 16-bit reload timer, this timer uses two channels, an even-numbered channel and an odd-numbered channel, to operate as a 32-bit reload timer. The even-numbered channel performs the lower 16-bit timer operations, and the odd-numbered channel performs the upper 16-bit timer operations. The interrupt controller and output waveform control conform to the settings of only the even-numbered channel. To set a cycle, write it first to the upper register (odd-numbered channel) and then to the lower register (even-numbered channel).

To read the timer value, read it first from the lower register (even-numbered channel) and then from the upper register (odd-numbered channel).

16-Bit PWC Timer

The PWC timer starts the 16-bit up counter upon input of the set measurement start edge. The timer stops the counter upon the detection of a measurement end edge. The count value at this time is stored as a pulse width in the data buffer register.

32-Bit PWC Timer

With the same basic operation as the 16-bit PWC timer, this timer uses two channels, an even-numbered channel and an odd-numbered channel, to operate as a 32-bit PWC timer. The even-numbered channel performs the lower 16-bit count operations, and the odd-numbered channel performs the upper 16-bit count operations. The interrupt controller conforms to the settings of only the even-numbered channel. To read a measurement value or count value, read it first from the lower register (even-numbered channel) and then from the upper register (odd-numbered channel).

4. 32-Bit Mode Operation

The reload timer and PWC are capable of 32-bit mode operation using two channels. This section shows the basic functions/operations of the 32-bit mode function.

32-Bit Mode Function

This function realizes the operation of the 32-bit data reload timer or 32-bit data PWC timer by using two base timer channels. The value of the timer or counter in operation can also be read when the lower 16-bit timer or counter value of the even-numbered channel is read. This is because the upper 16-bit timer or counter value of the odd-numbered channel is also fetched at the time.

32-Bit Mode Settings

First, reset the state by setting the FMD2 to FMD0 bits in the TMCR register of the even-numbered channel to 0b000 for reset mode. Then, in the same way as in 16-bit mode, make reload timer or PWC timer selection and operation settings. To set 32-bit operation mode at this time, write "1" to the T32 bit in the TMCR register. Keep the T32 bit of the odd-numbered channel at "0". There is no need to set reset mode. For the reload timer, set the upper 16-bit reload value of the 32 bits in the cycle setting register of the odd-numbered channel. Then, set the lower 16-bit reload value in the cycle setting register of the even-numbered channel.

After the writing of the T32 bit, the change to 32-bit operation mode is reflected immediately, so change the settings for both channels in the count stopped state.

To change from 32-bit mode to 16-bit mode, set the FMD2 to FMD0 bits in the TMCR register of the even-numbered channel to 0b000 for reset mode. This resets the states of both the even-numbered and odd-numbered channels. As a result, the settings for 16-bit mode can be made for each of the channels.

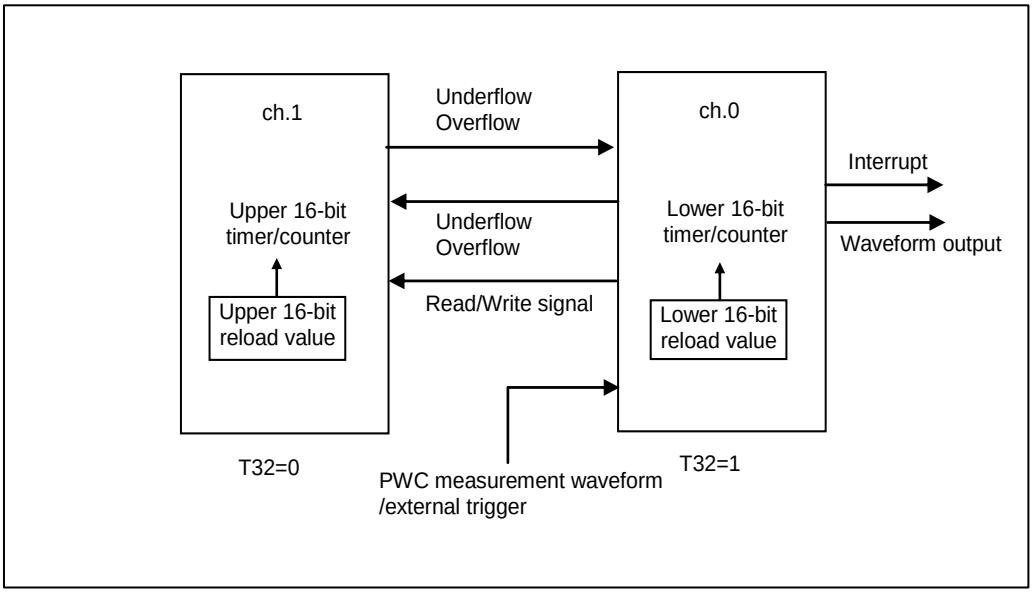
32-Bit Mode Operation

After 32-bit mode is set, if the reload timer or PWC timer is started with even-numbered channel control, the even-numbered channel timer/counter performs the lower 16-bit operations. The odd-numbered channel timer/counter performs the upper 16-bit operations.

Operation in 32-bit mode conforms to the settings made for the even-numbered channel. Thus, the settings made for the odd-numbered channel (excluding those in the cycle setting register at the reload timer time) are ignored. Timer start, waveform output, and interrupt signals for the even-numbered channel are valid. (The odd-numbered channel is fixed at "L" and masked.)

Figure 4-1 shows the configuration of ch.0 and ch.1.

Figure 4-1 Configuration of 32-Bit Mode Operation (for ch.0 and ch.1)



5. Interrupts from the Base Timer

This section shows a list summarizing the interrupt request flags, interrupt enable bits, and interrupt factors in each function of the base timer.

Interrupt Controller Bits and Interrupt Factors of Each Function

Table 5-1 shows the interrupt controller bits and interrupt factors of each function.

Table 5-1 Interrupt Controller Bits and Interrupt Factors in Each Mode

	Status Control Register (STC)			
	Interrupt Request Flag Bit	**Interrupt Request Enable Bit**	**Interrupt Factor**	**Interrupt Factor Output Signal**
PWM Timer Function	UDIR : bit0	UDIE : bit4	Detection of underflow	IRQ0
	DTIR : bit1	DTIE : bit5	Detection of duty match	
	TGIR : bit2	TGIE : bit6	Detection of timer activation trigger	IRQ1
PPG Timer Function	UDIR : bit0	UDIE : bit4	Detection of underflow	IRQ0
	TGIR : bit2	TGIE : bit6	Detection of timer activation trigger	IRQ1
Reload Timer Function	UDIR : bit0	UDIE : bit4	Detection of underflow	IRQ0
	TGIR : bit2	TGIE : bit6	Detection of timer activation trigger	IRQ1
PWC Timer Function	OVIR : bit0	OVIE : bit4	Detection of overflow	IRQ0
	EDIR : bit2	EDIE : bit6	Detection of measurement end	IRQ1

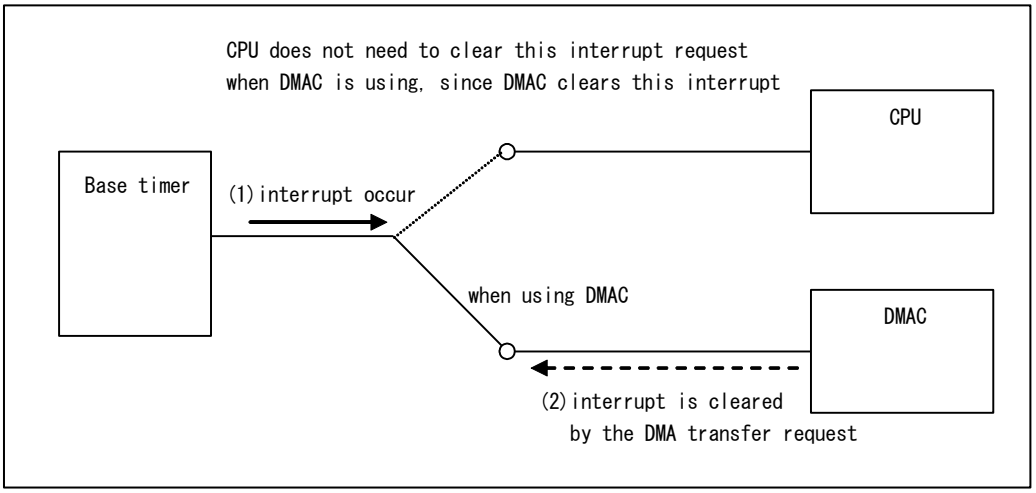
6. Start of DMA Controller (DMAC)

The generation of a base timer interrupt request can be used to start the DMAC.

DMA Transfer Operation Using Base Timer Interrupt Factor

The generation of a base timer interrupt factor can be used to start the DMAC. Figure 6-1 shows an overview of DMAC start with the base timer.

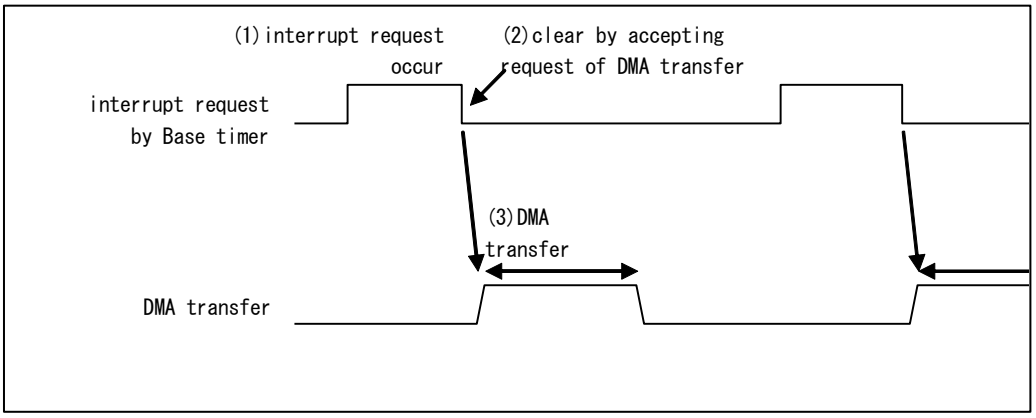
Figure 6-1 Overview of DMAC Start with the Base Timer



Configure the DMAC before starting it with the base timer. For details on DMAC settings, see the following chapter: "DMA Controller".

Figure 6-2 shows an example of a DMA transfer operation with a base timer interrupt request.

Figure 6-2 DMA Transfer Operation Example



7. Registers of the Base Timer

This section lists the registers of each mode of the base timer.

List of Registers When 16-Bit PWM Timer is Selected

Table 7-1 List of Registers When the 16-Bit PWM Timer is Selected

Abbreviated Register Name	Register Name	Reference
TMCR	Timer control register	9.1.8
TMCR2	Timer control register 2	9.1.8
STC	Status control register	9.1.8
STCC	Status Control Clear Register	9.1.8
STCS	Status Control Set Register	9.1.8
PCSR	PWM cycle setting register	9.1.9
PDUT	PWM duty setting register	9.1.10
TMR	Timer register	9.1.11
PSDR	Start delay value setting register	9.1.12
ADTR	ADC trigger value setting register	9.1.13

List of Registers When 16-Bit PPG Timer is Selected

Table 7-2 List of Registers When the 16-Bit PPG Timer is Selected

Abbreviated Register Name	Register Name	Reference
TMCR	Timer control register	9.2.6
TMCR2	Timer control register 2	9.2.6
STC	Status control register	9.2.6
STCC	Status Control Clear Register	9.2.6
STCS	Status Control Set Register	9.2.6
PRLL	L width setting reload register	9.2.7
PRLH	H width setting reload register	9.2.8
TMR	Timer register	9.2.9

List of Registers When Reload Timer is Selected

Table 7-3 List of Registers When the Reload Timer is Selected

Abbreviated Register Name	Register Name	Reference
TMCR	Timer control register	9.3.3
TMCR2	Timer control register 2	9.3.3
STC	Status control register	9.3.3
STCC	Status Control Clear Register	9.3.3
STCS	Status Control Set Register	9.3.3
PCSR	Cycle setting register	9.3.4
TMR	Timer register	9.3.5

List of Registers When PWC Timer is Selected

Table 7-4 List of Registers When the PWC Timer is Selected

Abbreviated Register Name	Register Name	Reference
TMCR	Timer control register	9.4.2
TMCR2	Timer control register 2	9.4.2
STC	Status control register	9.4.2
STCC	Status Control Clear Register	9.4.2
STCS	Status Control Set Register	9.4.2
DTBF	Data buffer register	9.4.3

Table 7-5 Offset Addresses

CH No.	Offset_Address (Common PERI #2)	CH No.	Offset_Address (Common PERI #1)	CH No.	Offset_Address (Common PERI #0)
24	0x0078_0000	12	0x0088_8000	0	0x0080_8000
25	0x0078_0400	13	0x0088_8400	1	0x0080_8400
26	0x0078_0800	14	0x0088_8800	2	0x0080_8800
27	0x0078_0C00	15	0x0088_8C00	3	0x0080_8C00
28	0x0078_1000	16	0x0088_9000	4	0x0080_9000
29	0x0078_1400	17	0x0088_9400	5	0x0080_9400
30	0x0078_1800	18	0x0088_9800	6	0x0080_9800
31	0x0078_1C00	19	0x0088_9C00	7	0x0080_9C00
32	0x0078_2000	20	0x0088_A000	8	0x0080_A000
33	0x0078_2400	21	0x0088_A400	9	0x0080_A400
34	0x0078_2800	22	0x0088_A800	10	0x0080_A800
35	0x0078_2C00	23	0x0088_AC00	11	0x0080_AC00

Tables 7-18, 7-19, 7-20 show the register map of each mode.

The "***/***/***/***" expression in each table corresponds to the reload/PWM/PPG/PWC timer, respectively.

Table 7-6 Register Map (Channel No.: 0) (12) (24)

Offset	Register Name/Initial Value			
	+3	+2	+1	+0
0x0000_0000	Reserved 00000000_00000000		BTxx_PCSR/BTxx_PCSR/BTxx_PRLL /Reserved XXXXXXXX_XXXXXXXX	
0x0000_0004	Reserved 00000000_00000000		Reserved/BTxx_PDUT/ BTxx_PRLH/ BTxx_DTBF XXXXXXXX_XXXXXXXX /00000000_00000000 (BTxx_DTBF)	
0x0000_0008	Reserved 00000000_00000000		BTxx_TMR/ BTxx_TMR/ BTxx_TMR/Reserved 00000000_00000000 XXXXXXXX_XXXXXXXX*	
0x0000_000C	Reserved 00000000_00000000		BTxx_TMCR 00000000_00000000	
0x0000_0010	Reserved 00000000_00000000		BTxx_TMCR2 00000000	BTxx_STC 00000000
0x0000_0014	Reserved 00000000_00000000		Reserved 00000000	BTxx_STCC 00000000
0x0000_0018	Reserved 00000000_00000000		Reserved 00000000	BTxx_STCS 00000000
0x0000_001C	Reserved 00000000_00000000		Reserved/BTxx_PSDR/Reserved/Reserved 00000000_00000000	
0x0000_0020	Reserved 00000000_00000000		Reserved/BTxx_ADTR/Reserved/Reserved 00000000_00000000	
0x0000_0030	Reserved 11111111_11111111		Reserved 11111111	BT_BTSEL01/1213/2425 1111_0000
0x0000_0034	Reserved 11111111_11111111		BT_BTSSSR0/12/24 11111111_11111111	
0x0000_0038	Reserved 11111111_11111111		BT_BTTRR0/12/24 11110000_00000000	

* The initial value is XXXXXXXX_XXXXXXXX only during reload timer operation.

Table 7-7 Register Map (Channel No.: 1, 3, 5, 7, 9, and 11) (13, 15, 17, 19, 21, and 23) (25, 27, 29, 31, 33, and 35)

Offset	Register Name/Initial Value			
	+3	+2	+1	+0
0x0000_0000	Reserved 00000000_00000000		BTxx_PCSR/ BTxx_PCSR/ BTxx_PRLL /Reserved XXXXXXXX_XXXXXXXX	
0x0000_0004	Reserved 00000000_00000000		Reserved/ BTxx_PDUT/ BTxx_PRLH/ BTxx_DTBF XXXXXXXX_XXXXXXXX /00000000_00000000(BTxx_DTBF)	
0x0000_0008	Reserved 00000000_00000000		BTxx_TMR/ BTxx_TMR/ BTxx_TMR/ Reserved 00000000_00000000 /XXXXXXXX_XXXXXXXX*	
0x0000_000C	Reserved 00000000_00000000		BTxx_TMCR 00000000_00000000	
0x0000_0010	Reserved 00000000_00000000		BTxx_TMCR2 00000000	BTxx_STC 00000000
0x0000_0014	Reserved 00000000_00000000		Reserved 00000000	BTxx_STCC 00000000
0x0000_0018	Reserved 00000000_00000000		Reserved 00000000	BTxx_STCS 00000000
0x0000_001C	Reserved 00000000_00000000		Reserved/BTxx_PSDR/Reserved/Reserved 00000000_00000000	
0x0000_0020	Reserved 00000000_00000000		Reserved/BTxx_ADTR/Reserved/Reserved 00000000_00000000	

* The initial value is XXXXXXXX_XXXXXXXX only during reload timer operation.

Table 7-8 Register Map (Channel No.: 2, 4, 6, 8, and 10) (14, 16, 18, 20, and 22) (26, 28, 30, 32, and 34)

Offset	Register Name/Initial Value			
	+3	+2	+1	+0
0x0000_0000	Reserved 00000000_00000000		BTxx_PCSR/ BTxx_PCSR/ BTxx_PRLL /Reserved XXXXXXXX_XXXXXXXX	
0x0000_0004	Reserved 00000000_00000000		Reserved/ BTxx_PDUT/ BTxx_PRLH / BTxx_DTBF XXXXXXXX_XXXXXXXX /00000000_00000000(BTxx_DTBF)	
0x0000_0008	Reserved 00000000_00000000		BTxx_TMR/ BTxx_TMR/ BTxx_TMR /Reserved 00000000_00000000 / XXXXXXXX_XXXXXXXX*	
0x0000_000C	Reserved 00000000_00000000		BTxx_TMCR 00000000_00000000	
0x0000_0010	Reserved 00000000_00000000		BTxx_TMCR2 00000000	BTxx_STC 00000000
0x0000_0014	Reserved 00000000_00000000		Reserved 00000000	BTxx_STCC 00000000
0x0000_0018	Reserved 00000000_00000000		Reserved 00000000	BTxx_STCS 00000000
0x0000_001C	Reserved 00000000_00000000		Reserved/BTxx_PSDR/Reserved/Reserved 00000000_00000000	
0x0000_0020	Reserved 00000000_00000000		Reserved/BTxx_ADTR/Reserved/Reserved 00000000_00000000	
0x0000_0030	Reserved 11111111_11111111		Reserved 11111111	BT_BTSEL23/45/67/89/ 1011/1415/1617/1819/ 2021/2223/2627/2829/ 3031/3233/3435 1111_0000

* The initial value is XXXXXXXX_XXXXXXXX only during reload timer operation.

8. Notes on Using the Base Timer

This section explains precautions on use of the base timer.

Notes to Observe When Accessing Register

Status Control Register (STC) Access

- This register supports writing from the bit-band alias area. For details on the bit-band alias area, see "CHAPTER: BIT-BAND UNIT."
- To clear a specific bit in this register, write "1" to the corresponding bit in the status control clear register (STCC).
- To set a specific bit in this register, write "1" to the corresponding bit in the status control set register (STCS).
- Direct writing to this register is possible only during writing to all bits.

Notes to Observe on Configuration Using Program, Common to Use of Timers

- Rewriting of the following bits in the TMCR2 register and TMCR register during operation is prohibited. Be sure that such rewriting is done either before the timer starts or after it stops.

[TMCR2 bit8], [TMCR bit14,13,12]	CKS3 to CKS0: Clock selection bits
[bit10,9,8]	EGS2, EGS1, EGS0: Measurement edge selection bits
[bit7]	T32: 32-bit timer selection bit (when the reload timer and PWC function are selected)
[bit6,5,4]	FMD2 to FMD0: Timer function selection bits
[bit2]	MDSE: Measurement mode (single/continuous) selection bit

- All the registers of the base timer are initialized when the FMD2 to FMD0 bits in the TMCR register are set to 0b000 for reset mode. For this reason, all the registers must be reconfigured.
- The settings for bits other than the FMD2 to FMD0 bits in the TMCR register are ignored and initialized when the FMD2 to FMD0 bits in the TMCR register are set to 0b000 for reset mode.

Notes on Using the 16-Bit PWM/PPG/Reload Timer

- If the interrupt request flag set timing and clear timing overlap, the flag set has priority, and the clear operation is disabled.
- If the load timing and count timing overlap, the load operation has priority for the down counter.
- After configuring the timer function with the FMD2 to FMD0 bits in the TMCR register, set the cycle, duty, H width, and L width.
- If a restart is detected at the count end time in one-shot mode, the count value is reloaded. Then, the restart begins.

Notes on Using the PWC Timer

- If "1" is written to the count start enable bit (CTEN), the counter is cleared. Thus, any data in the counter before the start is enabled is invalid.
- If the PWC mode setting (FMD2 to FMD0 = 0b100) and the measurement start setting (CTEN= 1) are made from a system reset and reset mode at the same time, the resulting operation may depend on the state of the immediately preceding measurement signal.
- If a measurement start edge is detected at the same time that a restart is set in continuous measurement mode, the timer starts counting immediately from 0x0001.
- If a restart is performed after a count operation begins, operations such as the following may occur, depending on the timing.
  - For a restart at the same time as a measurement end edge in pulse-width single measurement mode

The restart is performed and the measurement start edge wait state begins, but the measurement end flag (EDIR) is set to "1".

- For a restart at the same time as a measurement end edge in pulse-width continuous measurement mode

  The restart is performed and the measurement start edge wait state begins, but the measurement end flag (EDIR) is set to "1" and the measurement results at that point are transferred to DTBF.

Note the operation of the flag when using the interrupt controller, etc. at the restart time during operation as described above.

9. Base Timer Description by Function Mode

This section explains each function of the base timer.

Functions of the Base Timer

1. PWM Timer Function
2. PPG Timer Function
3. Reload Timer Function
4. PWC Timer Function

9.1. PWM Timer Function

Only one of the following timer functions can be selected for the base timer in the FMD2 to FMD0 bit settings in the timer control register (TMCR): 16-bit PWM timer, 16-bit PPG timer, 16/32-bit reload timer, and 16/32-bit PWC timer. This section explains the timer function with the PWM setting.

1. 16-Bit PWM Timer Operation
2. One-Shot Operation
3. Start Delay Function
4. ADC Trigger Output Timing
5. Interrupt Factors and Timing Chart
6. Output Waveform
7. Timer Control Registers (TMCR,TMCR2), Status Control Register (STC) , Status Control Clear Register (STCC), and Status Control Set Register (STCS)
8. PWM Cycle Setting Register (PCSR)
9. PWM Duty Setting Register (PDUT)
10. Timer Register (TMR)
11. Start Delay Value Setting Register (PSDR)
12. ADC Trigger Value Setting Register (ADTR)

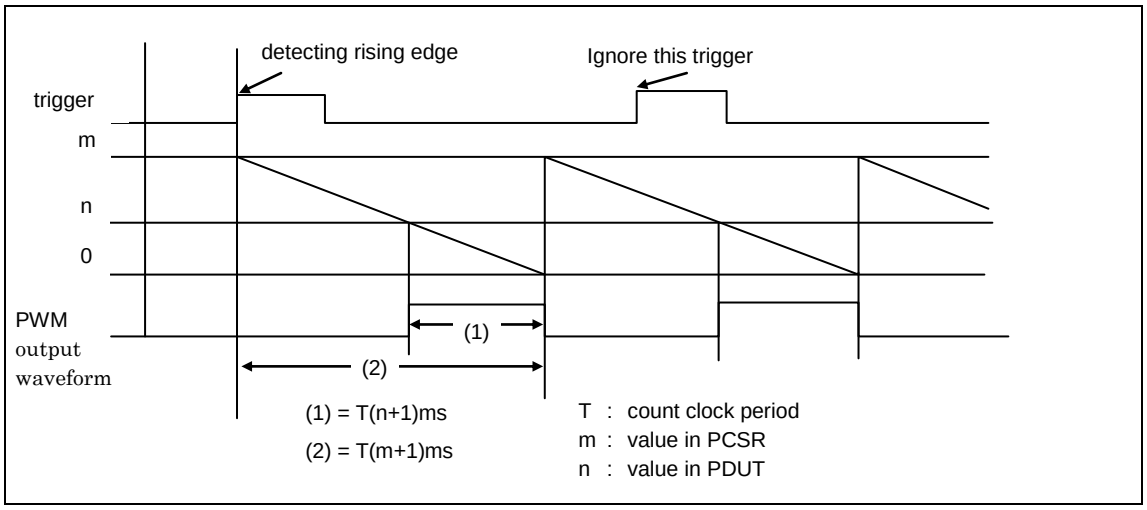
9.1.1. 16-Bit PWM Timer Operation

In PWM operation, waveforms of the set cycle can be output either singly or continuously upon the detection of a trigger. The cycle of the output pulse can be controlled through PCSR value changes. The duty ratio can be controlled through PDUT value changes. After writing data to PCSR, be sure to write to PDUT.

Continuous Operation

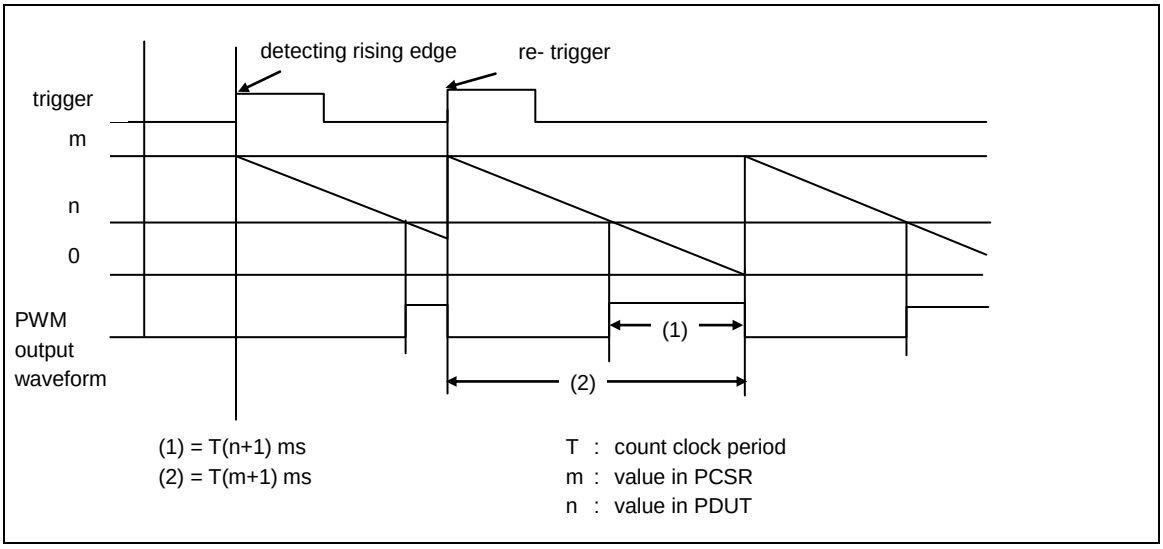
When Restart is Disabled (RTGEN=0)

Figure 9-1 PWM Operation Timing Chart (When Restart is Disabled)



When Restart is Enabled (RTGEN=1)

Figure 9-2 PWM Operation Timing Chart (When Restart is Enabled)



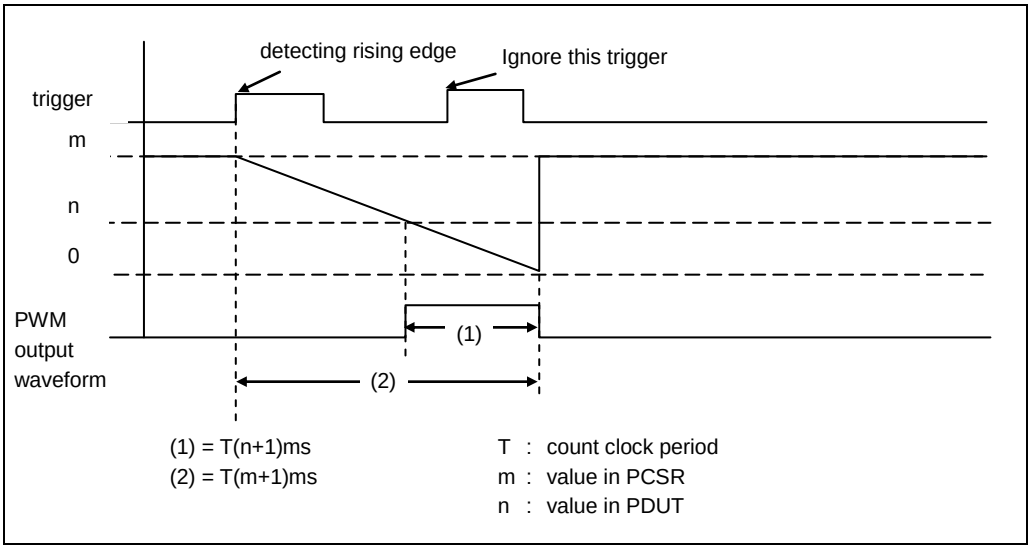
9.1.2. One-Shot Operation

In one-shot operation, a trigger can cause the output of a single pulse of any width. If restart is enabled, the counter is reloaded when an edge is detected during operation.

One-Shot Operation

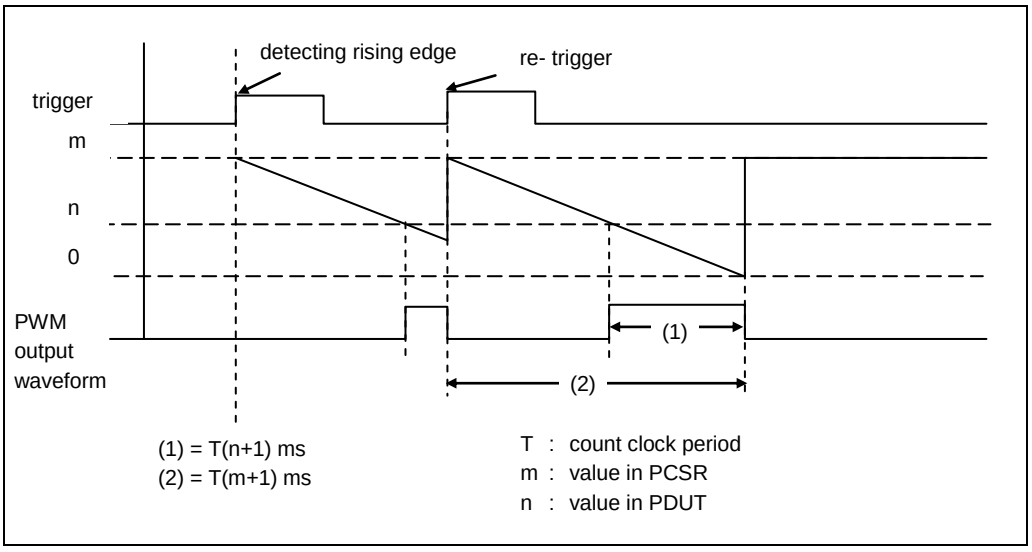
When Restart is Disabled (RTGEN=0)

Figure 9-3 One-Shot Operation Timing Chart (Trigger Restart Disabled)



When Restart is Enabled (RTGEN=1)

Figure 9-4 One-Shot Operation Timing Chart (Trigger Restart Enabled)



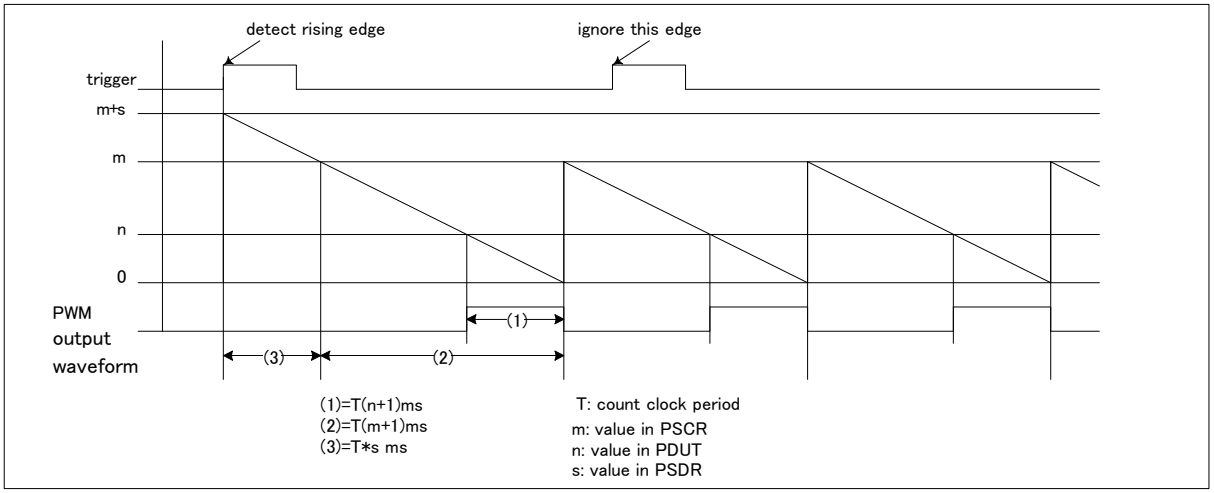
9.1.3. Start Delay Function

The start delay function can delay the time when PWM control starts after a trigger input.

Start Delay Operation in Continuous Operation

When Restart is Disabled (RTGEN=0)

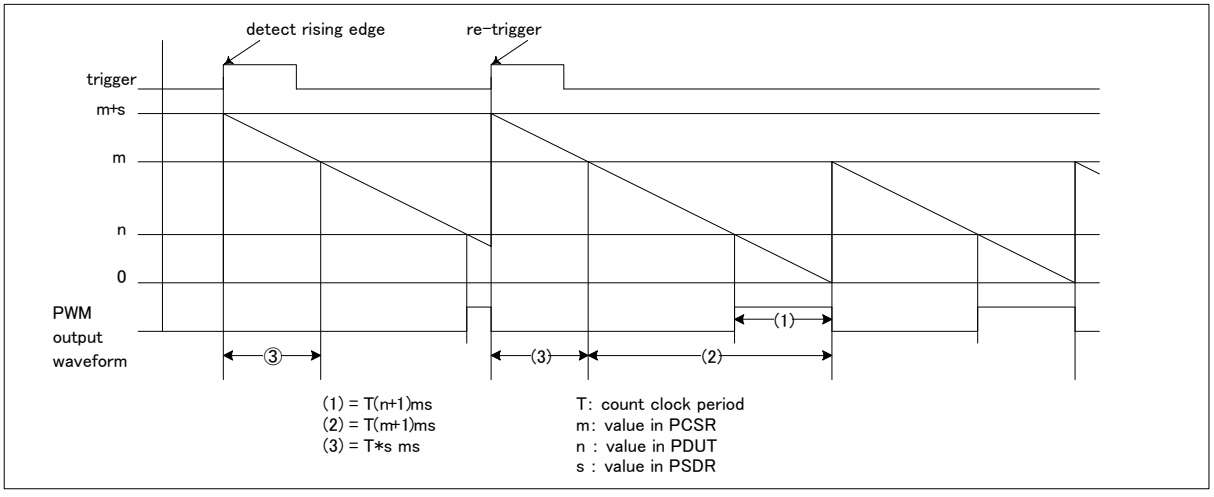
Figure 9-5 Start Delay Operation in Continuous Operation (Trigger Restart Disabled)



No PWM waveform is output during the start delay period ((3) above).

When Restart is Enabled (RTGEN=1)

Figure 9-6 Start Delay Operation in Continuous Operation (Trigger Restart Enabled)

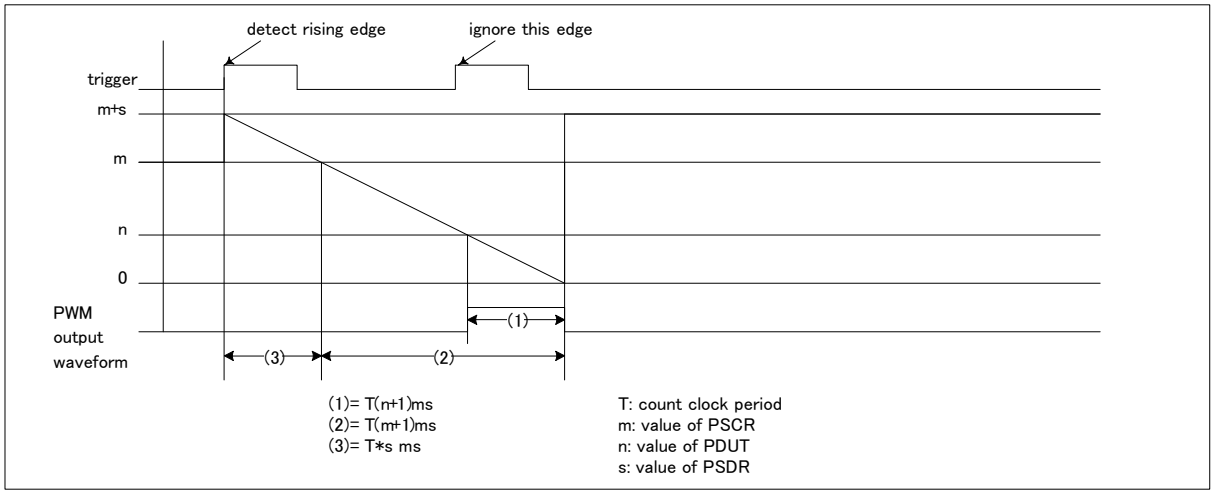


No PWM waveform is output during the start delay period ((3) above).

Start Delay Operation in One-Shot Operation

When Restart is Disabled (RTGEN=0)

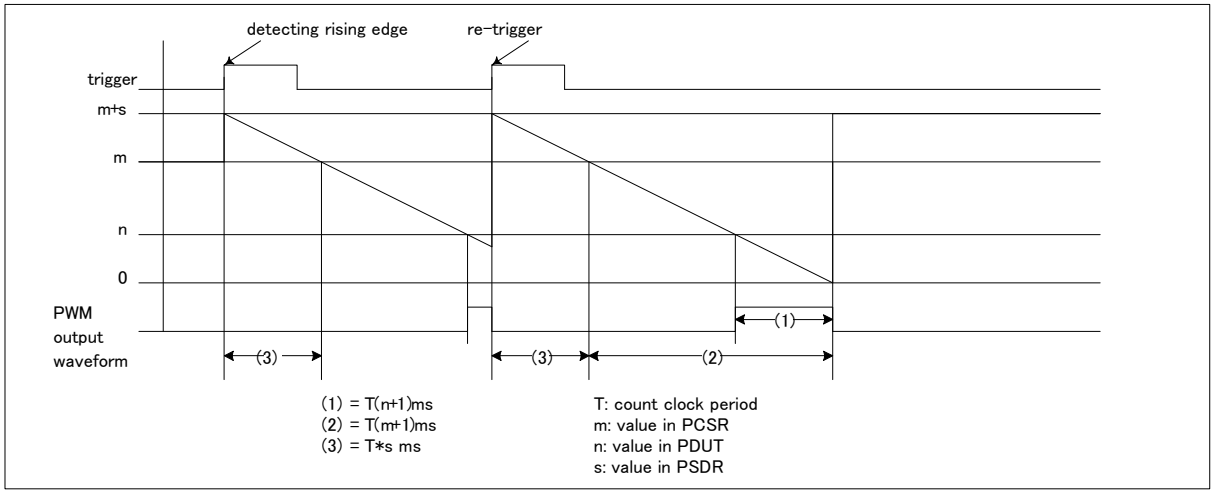
Figure 9-7 Start Delay Operation in One-Shot Operation (Trigger Restart Disabled)



No PWM waveform is output during the start delay period ((3) above).

When Restart is Enabled (RTGEN=1)

Figure 9-8 Start Delay Operation in One-Shot Operation (Trigger Restart Enabled)



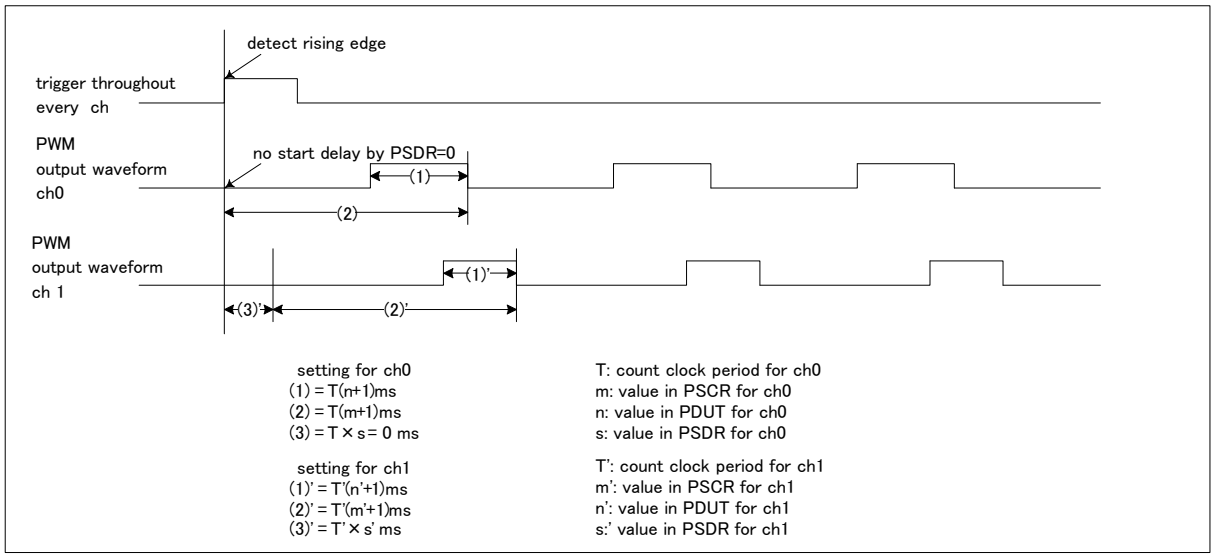
No PWM waveform is output during the start delay period ((3) above).

Start Delay Operation with Multiple Channels Interlocked

Operation with multiple channels interlocked in a certain phase relationship is enabled through control of a trigger as common to the channels.

When Restart is Disabled (RTGEN=0)

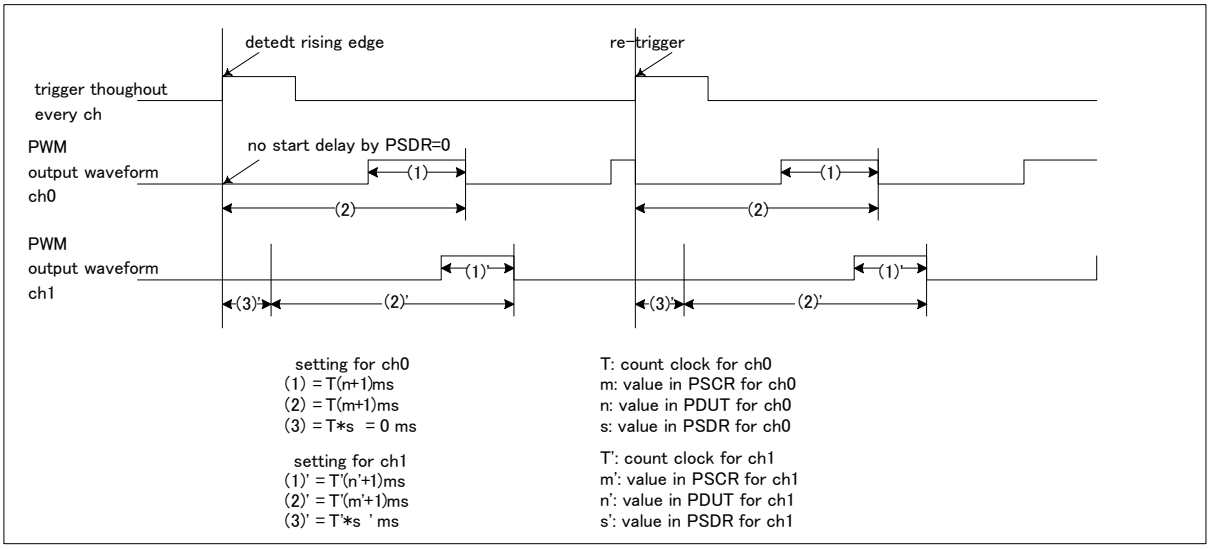
Figure 9-9 Start Delay Operation in Continuous Operation with Multiple Channels Interlocked (Trigger Restart Disabled)



No PWM waveform is output during the start delay period ((3) above).

When Restart is Enabled (RTGEN=1)

Figure 9-10 Start Delay Operation in Continuous Operation with Multiple Channels Interlocked (Trigger Restart Enabled)



No PWM waveform is output during the start delay period ((3) above).

9.1.4. ADC Trigger Output Timing

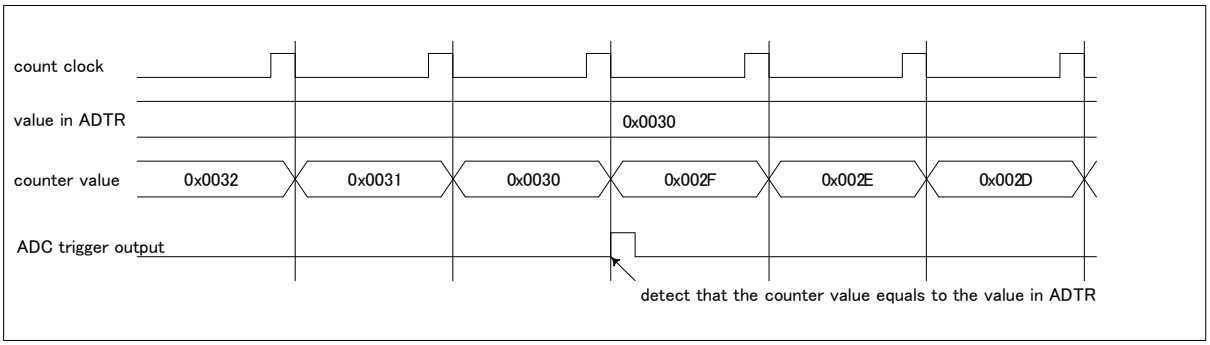
This section explains the ADC trigger output timing.

ADC Trigger Output Timing

ADC trigger is output when the 16-bit down counter is compared with the ADTR register and they match.

Figure 9-11 shows a timing chart of ADC trigger output when ADTR is 0x0030.

Figure 9-11 Timing Chart of ADC Trigger Output



9.1.5. Interrupt Factors and Timing Chart

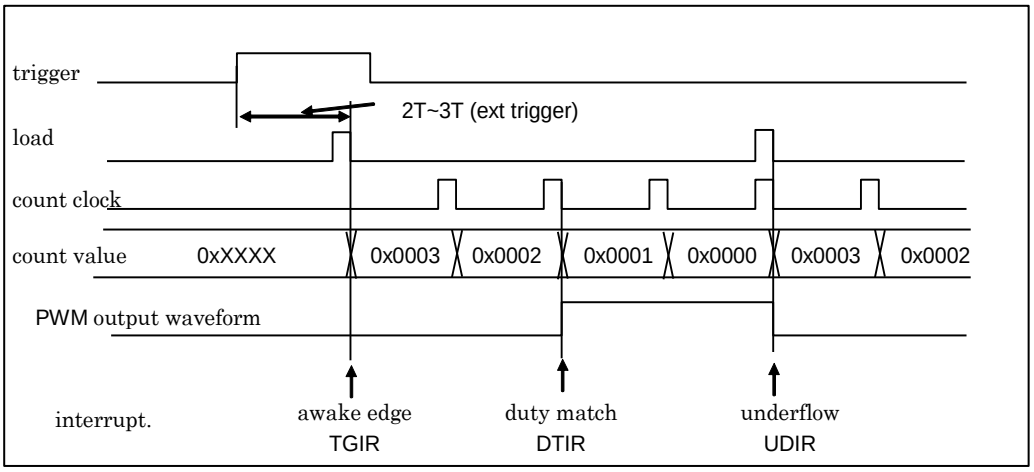
This section shows interrupt factors and a timing chart.

Interrupt Factors and Timing Chart (PWM Output: Normal Polarity)

As the time from the input of a trigger until the loading of a counter value, the required software trigger time is T, and the required external trigger time is 2T to 3T (T: internal clock cycle).

Figure 9-12 shows the interrupt factors and a timing chart where the cycle setting value is 3 and the duty value is 1.

Figure 9-12 PWM Timer Interrupt Factors and Timing Chart



9.1.6. Output Waveform

This section shows PWM output.

Output Methods for All "L" or All "H" in PWM Output

Figure 9-13 shows an output method where the PWM output is all "L". Figure 9-14 shows an output method where it is all "H".

Figure 9-13 Example Where PWM Output is All "L" Level

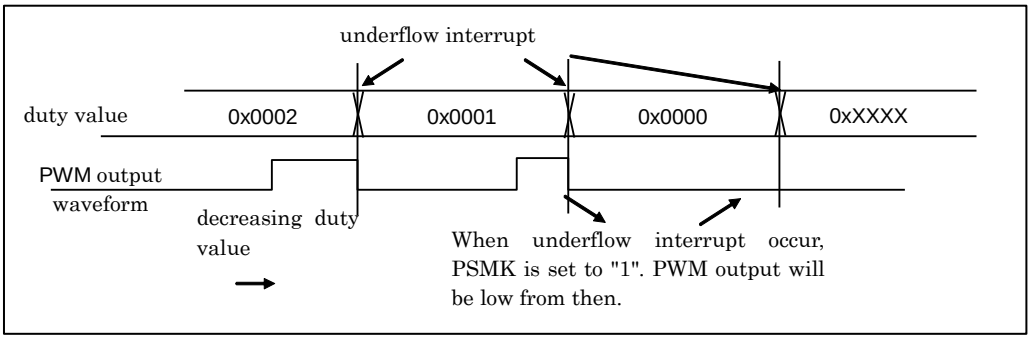
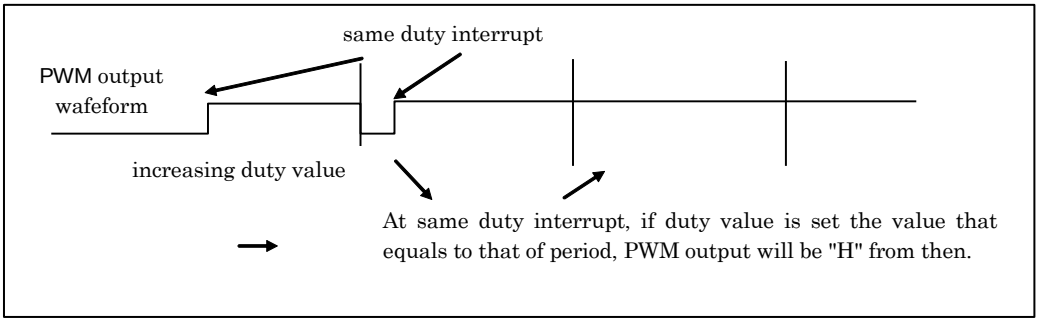


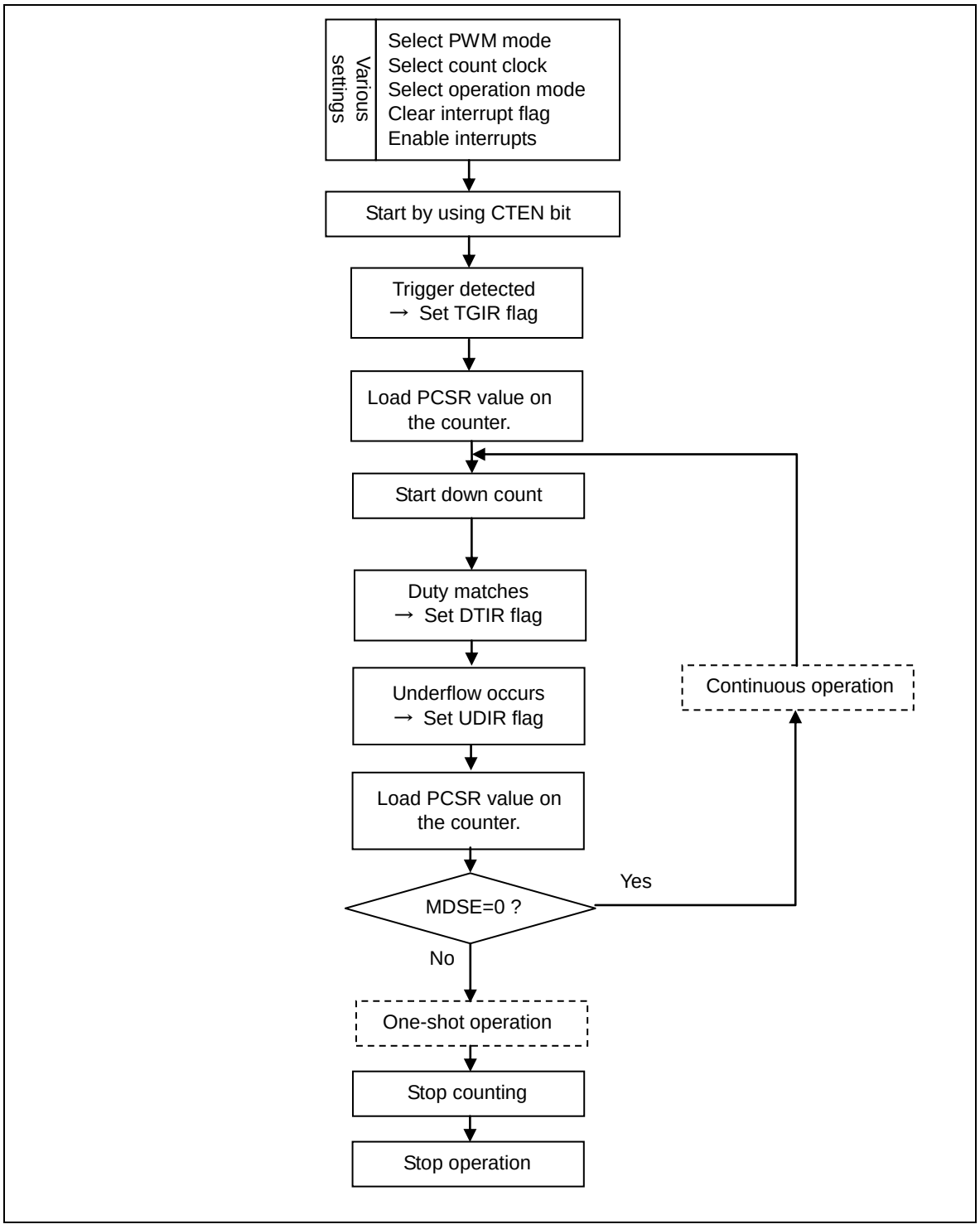
Figure 9-14 Example Where PWM Output is All "H" Level



9.1.7. PWM Timer Operation Flow

This section shows the PWM timer operation flow.

PWM Timer Operation Flow



9.1.8. Timer Control Registers (TMCR, TMCR2), Status Control Register (STC), Status Control Clear Register (STCC), and Status Control Set Register (STCS)

The timer control register (TMCR) controls the PWM timer. Note that there are bits that cannot be rewritten during PWM timer operation.

For details on writing to the status control register (STC), see "CHAPTER 40:8. Notes on Using the Base Timer."

Timer Control Register (Upper Byte of TMCR)

bit	15	14	13	12	11	10	9	8
Field	Reserved	CKS2	CKS1	CKS0	RTGEN	PMSK	EGS1	EGS0
R/W Attribute	R0,W0	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

[bit15] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

[TMCR2:bit8,bit14:12] CKS[3:0]: Count Clock Selection Bits

- These bits select a count clock for the 17-bit down counter.
- Any modification to the count clock is reflected immediately after the setting is changed. Therefore, modify CKS3 to CKS0 while counting is stopped (CTEN= 0). However, note that the bits can be modified at the same time as "1" is written to the count operation enable bit (CTEN).

Bits				Description
0	0	0	0	Internal clock
0	0	0	1	Internal clock divided by 4
0	0	1	0	Internal clock divided by 16
0	0	1	1	Internal clock divided by 128
0	1	0	0	Internal clock divided by 256
0	1	0	1	External clock (rising-edge event)
0	1	1	0	External clock (falling-edge event)
0	1	1	1	External clock (both-edges event)
1	0	0	0	Internal clock divided by 512
1	0	0	1	Internal clock divided by 1024
1	0	1	0	Internal clock divided by 2048
1	0	1	1	Internal clock divided by 2
1	1	0	0	Internal clock divided by 8
1	1	0	1	Internal clock divided by 32
1	1	1	0	Internal clock divided by 64
1	1	1	1	Setting prohibited

[bit11] RTGEN: Restart Enable Bit

This bit enables restart by a software trigger or trigger input.

Bit	Description
0	Disable restart.
1	Enable restart.

[bit10] PMSK: Pulse Output Mask Bit

- This bit controls the output waveform level of the PWM output waveform.
- When the bit is "0", the PWM waveform is output as is.
- When the bit is "1", PWM output is masked to L output regardless of the cycle or duty setting value.

Note:

- *If the output polarity specification bit (OSEL) in the timer control register (lower byte of TMCR) is set for inverted output, setting the PMSK bit to "1" results in masking to H output.*

Bit	Description
0	Normal output
1	Fixed at L output

[bit9:8] EGS[1:0]: Trigger Input Edge Selection Bits

- These bits select the valid edge for an input waveform as an external start factor, and they set trigger conditions.
- For the initial value or the 0b00 setting, no valid edge is selected for an input waveform, so no external waveform causes a start.

Note:

- *If "1" is written to the STRG bit, the software trigger becomes valid regardless of the EGS1 and EGS0 settings.*

- Modify EGS1 and EGS0 while counting is stopped (CTEN= 0). However, note that the bits can be modified at the same time as "1" is written to the CTEN bit.

Bits		Description
0	0	Trigger input invalid
0	1	Rising edge
1	0	Falling edge
1	1	Both edges

Timer Control Register (Lower Byte of TMCR)

bit	7	6	5	4	3	2	1	0
Field	Reserved	FMD2	FMD1	FMD0	OSEL	MDSE	CTEN	STRG
R/W Attribute	R0,W0	R/W	R/W	R/W	R/W	R/W	R/W	R0,W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

[bit7] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

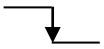
[bit6:4] FMD[2:0]: Timer Function Selection Bits

- These bits select the timer function.
- If 0b001 is set in the FMD2 to FMD0 bits, the PWM function is selected.
- Modify these bits while the timer is stopped (CTEN= 0). However, note that the bits can be modified at the same time as "1" is written to the CTEN bit.

Bits			Description
0	0	0	Reset mode
0	0	1	16-bit PWM timer
0	1	0	16-bit PPG timer
0	1	1	16/32-bit reload timer
1	0	0	16/32-bit PWC timer
1	0	1	Setting prohibited
1	1	0	
1	1	1	

[bit3] OSEL: Output Polarity Specification Bit

- This bit sets the PWM output polarity.

Polarity	After Reset	Duty Match	Underflow
Normal	"L" output		
Inverse	"H" output		

Bit	Description
0	Normal polarity
1	Inverse polarity

[bit2] MDSE: Mode Selection Bit

- This bit selects either operation for continuous pulse output or one-shot operation for single-pulse output.
- Modify the bit while the timer is stopped (CTEN= 0). However, note that the bits can be modified at the same time as "1" is written to the CTEN bit.

Bit	Description
0	Continuous operation
1	One-shot operation

[bit1] CTEN: Count Operation Enable Bit

- This bit enables operation of the down counter.
- If "0" is written to this bit when counter operation is enabled (CTEN bit is "1"), the counter stops.

Bit	Description	
	Read	Write
0	Stop operation.	Set this bit to "0".
1	Enable operation.	Set this bit to "1".

[bit0] STRG: Software Trigger Bit

- If "1" is written to the STRG bit when the CTEN bit is "1", a software trigger is applied.
- The read value of the STRG bit is always "0".

Notes:

- *Even if "1" is written to the CTEN and STRG bits at the same time, a software trigger is applied.*
- *If "1" is written to the STRG bit, the software trigger becomes valid regardless of the EGS1 and EGS0 settings.*

Bit	Description
0	Invalid
1	Startup by software

Timer Control Register 2 (TMCR2)

bit	15	14	13	12	11	10	9	8
Field	Reserved							CKS3
R/W Attribute	R0,W0							R/W
Protection Attribute	-							-
Initial Value	0000000							0

Note: This register is located in the upper byte of the STC register.

[bit15:9] Reserved: Reserved Bits

The read value is "0".

If writing to this bit, write "0".

[bit8] CKS3: Count Clock Selection Bit

See "Count clock selection bits" in See "Timer Control Registers (TMCR, TMCR2), Status Control Register (STC), Status Control Clear Register (STCC), and Status Control Set Register (STCS)."

Status Control Register (STC)

bit	7	6	5	4	3	2	1	0
Field	Reserved	TGIE	DTIE	UDIE	Reserved	TGIR	DTIR	UDIR
R/W Attribute	R0,W0	R/W	R/W	R/W	R0,W0	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

Note: The TMCR2 register is located in the upper byte of this register.

[bit7] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

[bit6] TGIE: Trigger Interrupt Request Enable Bit

- This bit controls interrupt requests of the trigger interrupt request bit (bit2 TGIR).
- If the TGIE bit is enabled and the TGIR bit is set to "1", an interrupt request is issued to the CPU.
- Writing "1" to the STCC:TGIEC bit clears this bit.
- Writing "1" to the STCS:TGIES bit sets this bit.

Bit	Description
0	Disable interrupt requests.
1	Enable interrupt requests.

[bit5] DTIE: Duty Match Interrupt Request Enable Bit

- This bit controls interrupt requests of the duty match interrupt request bit (bit1 DTIR).
- If the DTIE bit is enabled and the DTIR bit is set to "1", an interrupt request is issued to the CPU.
- Writing "1" to the STCC:DTIEC bit clears this bit.
- Writing "1" to the STCS:DTIES bit sets this bit.

Bit	Description
0	Disable interrupt requests.
1	Enable interrupt requests.

[bit4] UDIE: Underflow Interrupt Request Enable Bit

- This bit controls interrupt requests of the underflow interrupt request bit (bit0 UDIR).
- If the UDIE bit is enabled and the UDIR bit is set to "1", an interrupt request is issued to the CPU.
- Writing "1" to the STCC:UDIEC bit clears this bit.
- Writing "1" to the STCS:UDIES bit sets this bit.

Bit	Description
0	Disable interrupt requests.
1	Enable interrupt requests.

[bit3] Reserved: Reserved Bit

The read value is "0".
If writing to this bit, write "0".

[bit2] TGIR: Trigger Interrupt Request Bit

- The TGIR bit is set to "1" when a software trigger or trigger input is detected.
- Writing "1" to the STCC:TGIRC bit clears this bit.
- This bit is read-only. Writing data to this bit has no effect on operation.

Bit	Description
0	The interrupt factor is cleared.
1	Detect the interrupt factor.

[bit1] DTIR: Duty Match Interrupt Request Bit

- The DTIR bit is set to "1" when the count value matches the duty setting value.
- Writing "1" to the STCC:DTIRC bit clears this bit.
- This bit is read-only. Writing data to this bit has no effect on operation.

Bit	Description
0	The interrupt factor is cleared.
1	Detect the interrupt factor.

[bit0] UDIR: Underflow Interrupt Request Bit

- The UDIR bit is set to "1" when the count value underflows from 0x0000 to 0xFFFF.
- Writing "1" to the STCC:UDIRC bit clears this bit.
- This bit is read-only. Writing data to this bit has no effect on operation.

Bit	Description
0	The interrupt factor is cleared.
1	Detect the interrupt factor.

Status Control Clear Register (STCC)

bit	7	6	5	4	3	2	1	0
Field	Reserved	TGIEC	DTIEC	UDIEC	Reserved	TGIRC	DTIRC	UDIRC
R/W Attribute	R0,W0	R0,W	R0,W	R0,W	R0,W0	R0,W	R0,W	R0,W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

Note: Reserved bits are located in the upper byte of this register.

[bit7] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

[bit6] TGIEC: Trigger Interrupt Request Enable Clear Bit

- If "1" is written to this bit, the STC:TGIE bit is cleared to "0".
- The read value of this bit is always "0".

Bit	Description
0	Invalid
1	Clear the TGIE bit.

[bit5] DTIEC: Duty Match Interrupt Request Enable Clear Bit

- If "1" is written to this bit, the STC:DTIE bit is cleared to "0".
- The read value of this bit is always "0".

Bit	Description
0	Invalid
1	Clear the DTIE bit.

[bit4] UDIEC: Underflow Interrupt Request Enable Clear Bit

- If "1" is written to this bit, the STC:UDIE bit is cleared to "0".
- The read value of this bit is always "0".

Bit	Description
0	Invalid
1	Clear the UDIE bit.

[bit3] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

[bit2] TGIRC: Trigger Interrupt Request Clear Bit

- If "1" is written to this bit, the STC:TGIR bit is cleared to "0".
- The read value of this bit is always "0".

Bit	Description
0	Invalid
1	Clear the TGIR bit.

[bit1] DTIRC: Duty Match Interrupt Request Clear Bit

- If "1" is written to this bit, the STC:DTIR bit is cleared to "0".
- The read value of this bit is always "0".

Bit	Description
0	Invalid
1	Clear the DTIR bit.

[bit0] UDIRC: Underflow Interrupt Request Clear Bit

- If "1" is written to this bit, the STC:UDIR bit is cleared to "0".
- The read value of this bit is always "0".

Bit	Description
0	Invalid
1	Clear the UDIR bit.

Status Control Set Register (STCS)

bit	7	6	5	4	3	2	1	0
Field	Reserved	TGIES	DTIES	UDIES	Reserved			
R/W Attribute	R0,W0	R0,W	R0,W	R0,W	R0,W0			
Protection Attribute	-	-	-	-	-			
Initial Value	0	0	0	0	0000			

Note: Reserved bits are located in the upper byte of this register.

[bit7] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

[bit6] TGIES: Trigger Interrupt Request Enable Set Bit

- If "1" is written to this bit, the STC:TGIE bit is set to "1".
- The read value of this bit is always "0".

Bit	Description
0	Invalid
1	Set the TGIE bit.

[bit5] DTIES: Duty Match Interrupt Request Enable Set Bit

- If "1" is written to this bit, the STC:DTIE bit is set to "1".
- The read value of this bit is always "0".

Bit	Description
0	Invalid
1	Set the DTIE bit.

[bit4] UDIES: Underflow Interrupt Request Enable Set Bit

- If "1" is written to this bit, the STC:UDIE bit is set to "1".
- The read value of this bit is always "0".

Bit	Description
0	Invalid
1	Set the UDIE bit.

[bit3-0] Reserved: Reserved Bits

The read value is "0".

If writing to this bit, write "0".

9.1.9. PWM Cycle Setting Register (PCSR)

The PWM cycle setting register (PCSR) is a register with a buffer for setting a cycle. There is a transfer to the timer register at the start time and at the underflow time.

bit	15	0
Field	PCSR[15:0]	
R/W Attribute	R/W	
Protection Attribute	-	
Initial Value	XXXXXXXXXXXXXXXX	

These bits compose the register with a buffer for setting a cycle. There is a transfer to the timer register at the start time and at the underflow time.

When initializing and rewriting the cycle setting register, be sure to write to the duty setting register after writing to the cycle setting register.

- Use 16- or 32-bit data access for the PCSR register.
- After configuring the PWM function with the FMD2 to FMD0 bits in the TMCR register, set a cycle in the PCSR register.

9.1.10. PWM Duty Setting Register (PDUT)

The PWM duty setting register (PDUT) is a register with a buffer for setting a duty. An underflow causes a buffer transfer.

bit	15	0
Field	PDUT[15:0]	
R/W Attribute	R/W	
Protection Attribute	-	
Initial Value	XXXXXXXXXXXXXXXX	

These bits compose the register with a buffer for setting a duty. An underflow causes a transfer from the buffer.

If the set values of the cycle setting register and duty setting register are the same value, the output for normal polarity is all "H", and the output for inverse polarity is all "L".

If PCSR < PDUT in the set values, the output for normal polarity is all "L", and the output for inverse polarity is all "H".

- Use 16- or 32-bit data access for the PDUT register.
- After configuring the PWM function with the FMD2 to FMD0 bits in the TMCR register, set a duty in the PDUT register.

9.1.11. Timer Register (TMR)

The timer register (TMR) can read the value of the 17-bit down counter.

bit	31	17	16
Field	Reserved		TMR[16]
R/W Attribute	R0,W0		R,WX
Protection Attribute	-		-
Initial Value	0x0000		0

bit	15	0
Field	TMR[15:0]	
R/W Attribute	R,WX	
Protection Attribute	-	
Initial Value	0x0000	

[bit31:17] Reserved: Reserved Bits

The read value is "0".

If writing to this bit, write "0".

[bit16:0] TMR16 to TMR0: Timer Value Data Bits

The timer register can read the value of the 17-bit down counter that takes into account the start delay period.

- During trigger input, a value of "start delay period (PSDR) + PWM cycle (PCSR)" is reloaded into the TMR register.
- A value of "PWM cycle (PCSR)" is reloaded into the TMR register when an underflow occurs.
- Use 32-bit data access for the TMR register.

9.1.12. Start Delay Value Setting Register (PSDR)

The start delay value setting register (PSDR) sets a start delay value for the start delay function.

bit	15	0
Field	PSDR[15:0]	
R/W Attribute	R/W	
Protection Attribute	-	
Initial Value	0x0000	

The PSDR[15:0] bits set a start delay value to delay the start of the PWM timer after trigger input.

- To not use the start delay function, set the PSDR[15:0] bits to 0x0000.
- No PWM waveform is output during start delay control period after trigger input.
- After configuring the PWM function with the FMD2 to FMD0 bits in the TMCR register, set a cycle in the PSDR register.
- Use 16- or 32-bit data access for the PSDR register.
- Any modification to the PSDR[15:0] bits is reflected immediately after the setting is changed. Set the PSDR[15:0] bits while counting is stopped (TMCR:CTEN= 0).

9.1.13. ADC Trigger Value Setting Register (ADTR)

The ADC trigger value setting register (ADTR) sets the time for ADC trigger output.

bit	15	0
Field	ADTR[15:0]	
R/W Attribute	R/W	
Protection Attribute	-	
Initial Value	0x0000	

The ADC trigger is output when the 16-bit down counter is compared with the ADTR[15:0] bits and they match.

- For the ADTR register, set a value equal to or less than the PCSR[15:0] bits in the PCSR register.
- Use 16- or 32-bit data access for the ADTR register.
- After configuring the PWM function with the FMD2 to FMD0 bits in the TMCR register, set a cycle in the ADTR register.

9.2. PPG Timer Function

Only one of the following timer functions can be selected for the base timer in the FMD2 to FMD0 bit settings in the timer control register (TMCR): 16-bit PWM timer, 16-bit PPG timer, 16/32-bit reload timer, and 16/32-bit PWC timer. This section explains the timer function with the PPG setting.

1. 16-Bit PPG Timer Operation
2. Continuous Operation
3. One-Shot Operation
4. Interrupt Factors and Timing Chart
5. PPG Timer Operation Flow
6. Timer Control Registers (TMCR, TMCR2), Status Control Register (STC), Status Control Clear Register (STCC), and Status Control Set Register (STCS) When the PPG Timer is Selected
7. L Width Setting Reload Register (PRLL)
8. H Width Setting Reload Register (PRLH)
9. Timer Register (TMR)

9.2.1. 16-Bit PPG Timer Operation

PPG timer operation can control output pulses by using the L width and H width settings for the output pulses in the respective reload registers.

Overview of Operation

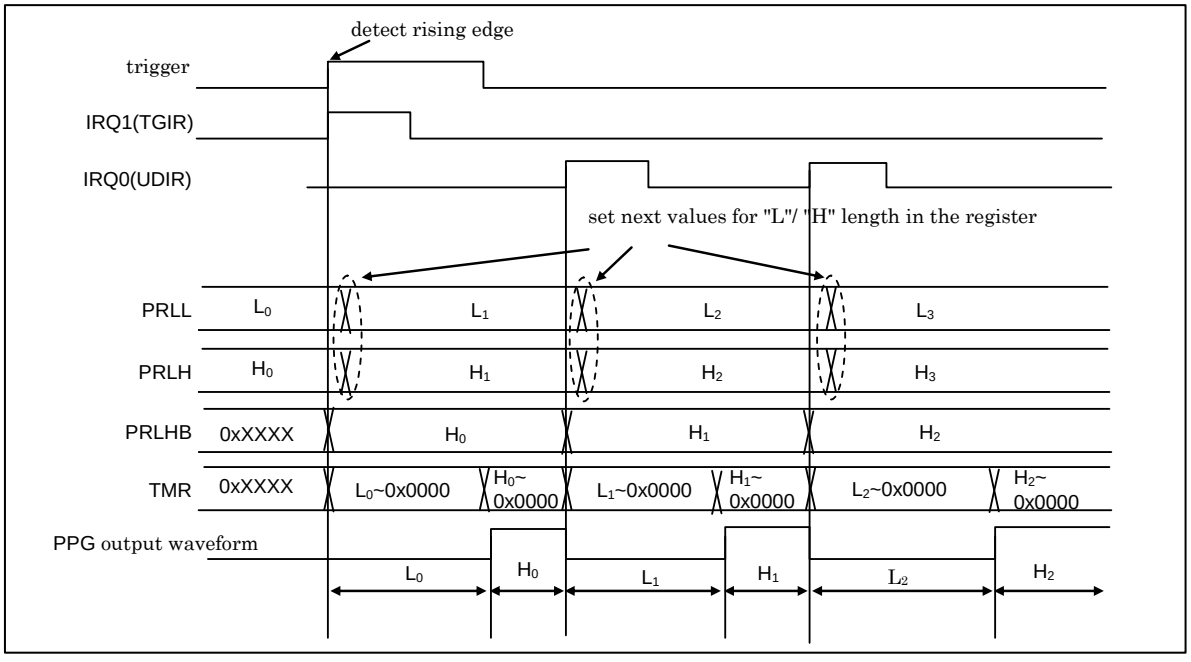
There are two 16-bit reload registers for the L width and H width settings and one buffer for the H width setting (PRLL, PRLH, and PRLHB).

An activation trigger first causes loading of the set value of PRLL into the 16-bit down counter. At the same time, the set value of PRLH is transferred to PRLHB. The PPG output level changes to L, and each count clock counts down. The detection of an underflow causes reloading of the PRLHB value into the counter and inversion of the PPG output waveform. Meanwhile, each count clock counts down. The detection of another underflow causes inversion of the PPG output waveform, reloading of the set value of PRLL into the counter, and transfer of the set value of PRLH to PRLHB.

The pulse output by the output waveform from this operation has an L width and H width corresponding to each reload register value.

Timing of Writing to Reload Registers

Data is written to the reload registers PRLL and PRLH when an activation trigger is detected during the period after the underflow interrupt factor (UDIR) is set to "1" and before the change to the next cycle. The data that is set at this time is the settings for the next cycle. The set data in PRLL and PRLH is automatically transferred to TMR and PRLHB, respectively, when an activation trigger is detected or when an underflow occurs at the H width count end time. The data transferred to PRLHB is automatically reloaded into TMR when an underflow occurs at the L width count end time.



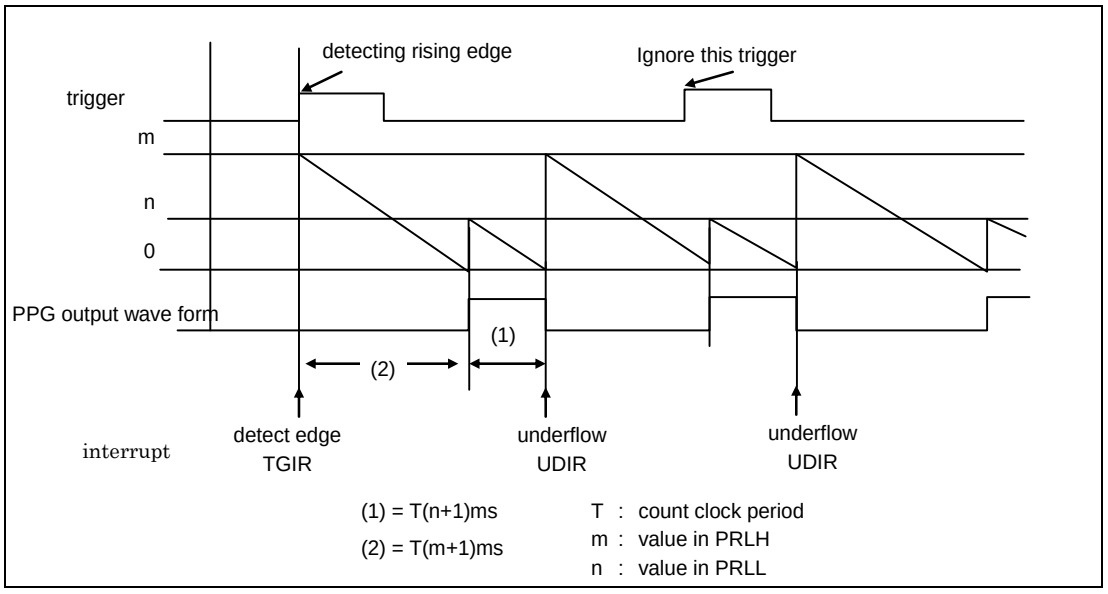
9.2.2. Continuous Operation

In continuous operation, pulses can be output continuously through updating of the L width and H width at the set timing for each interrupt factor. If restart is enabled, the counter is reloaded when an edge is detected during operation.

Continuous Operation

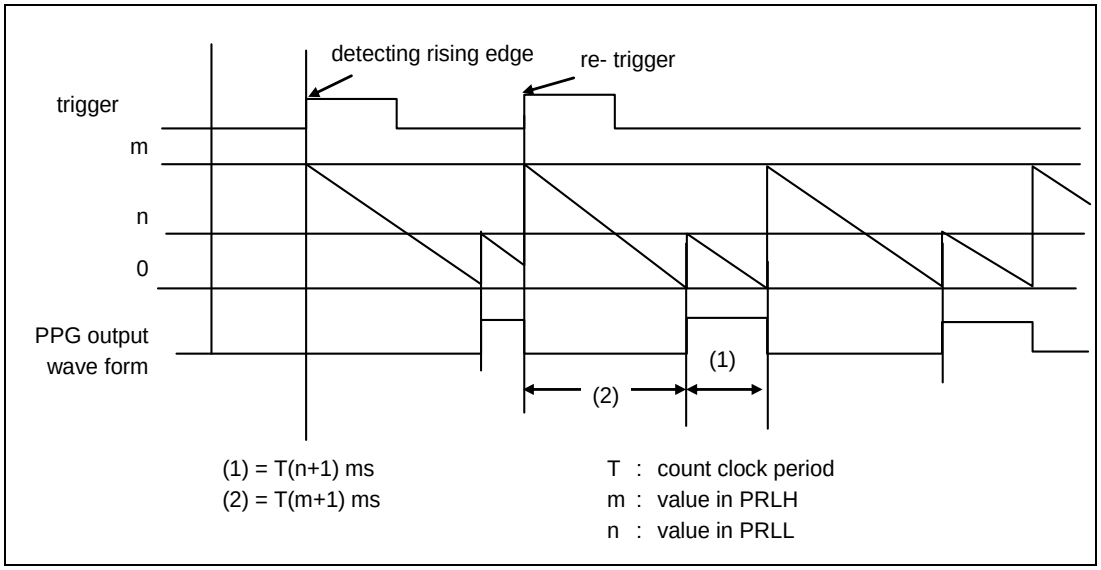
When Restart is Disabled (RTGEN=0)

Figure 9-15 PPG Operation Timing Chart (When Restart is Disabled)



When Restart is Enabled (RTGEN=1)

Figure 9-16 PPG Operation Timing Chart (When Restart is Enabled)



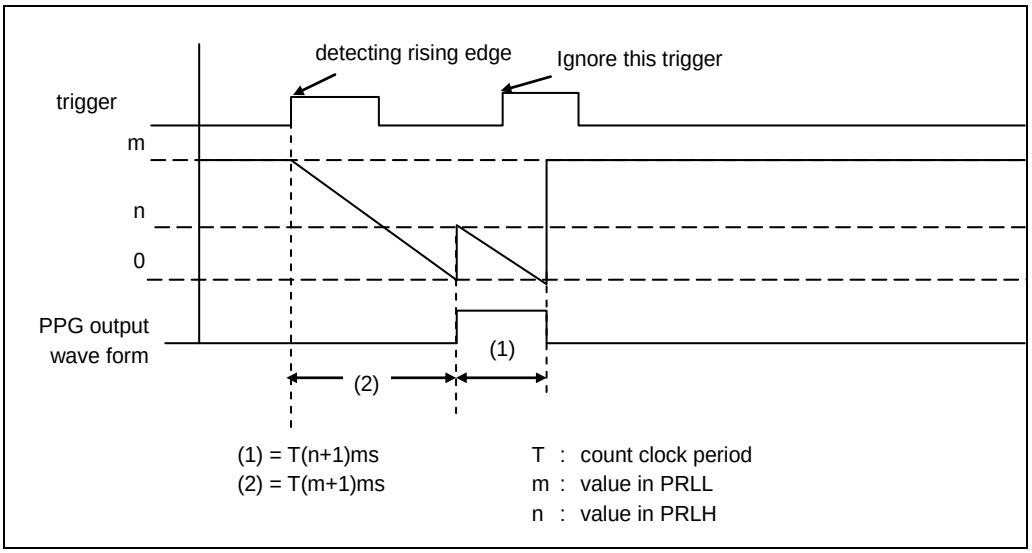
9.2.3. One-Shot Operation

In one-shot operation, a trigger can cause the output of a single pulse of any width. If restart is enabled, the counter is reloaded when an edge is detected during operation.

One-Shot Operation

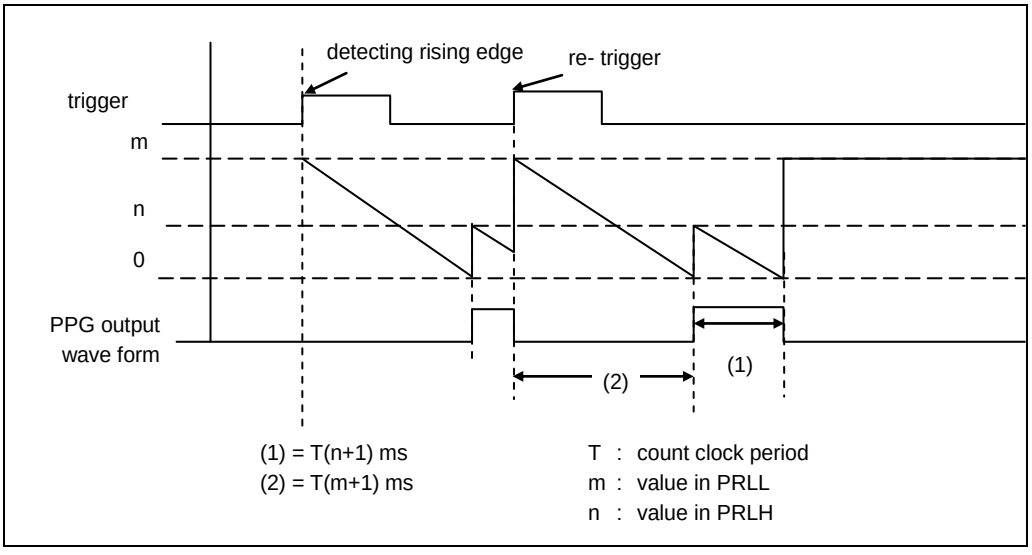
When Restart is Disabled (RTGEN=0)

Figure 9-17 One-Shot Operation Timing Chart (Trigger Restart Disabled)



When Restart is Enabled (RTGEN=1)

Figure 9-18 One-Shot Operation Timing Chart (Trigger Restart Enabled)



Relationship between Reload Value and Pulse Width

The output pulse width is the following value: 1 is added to the value written to the 16-bit reload register, and the sum is multiplied by the cycles of the count clock. Therefore, the pulse width of 1 cycle of the count clock is the value when the reload register value is 0x0000. The pulse width of 65536 cycles of the count clock is the value when the reload register value is 0xFFFF. The pulse width calculation formula is as follows.

PL = T x (L + 1) PL: L pulse width
PH = T x (H + 1) PH: H pulse width

T: Count clock cycle
L: PRLL value
H: PRLH value

9.2.4. Interrupt Factors and Timing Chart

This section shows interrupt factors and a timing chart.

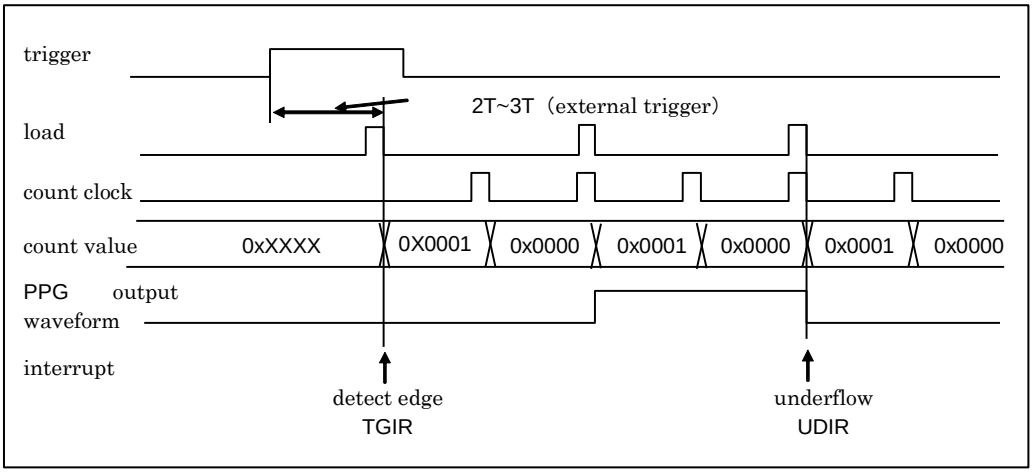
Interrupt Factors and Timing Chart (PPG Output: Normal Polarity)

For the period from trigger application until the loading of a counter value, the required software trigger time is T, and the required external trigger time is 2T to 3T (T: Internal clock cycle).

An interrupt factor is generated when a PPG activation trigger is detected or when an underflow is detected at the H level output time.

Figure 9-19 shows the interrupt factors and a timing chart where the L width setting value is 1 and the H width setting value is 1.

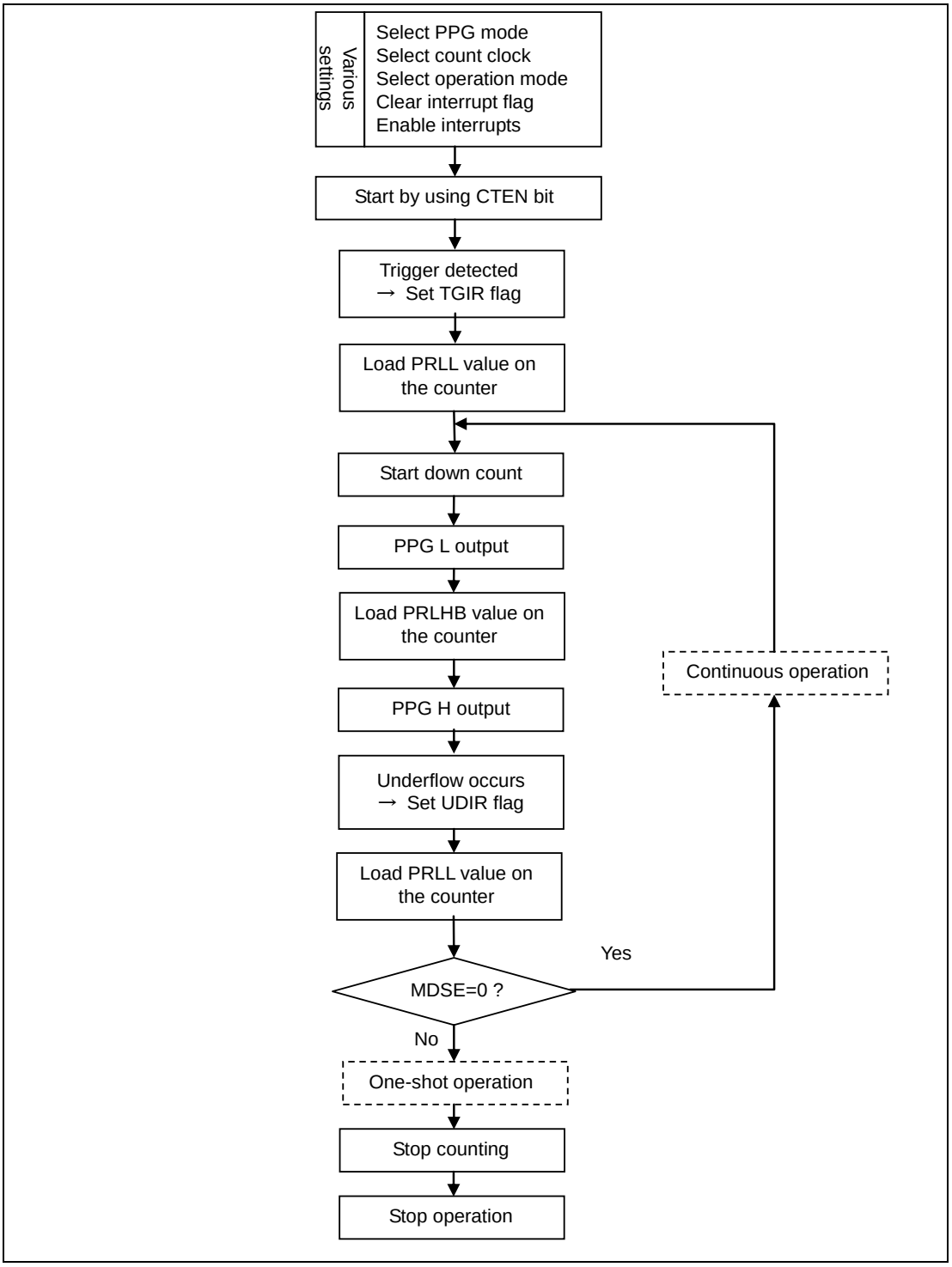
Figure 9-19 PPG Timer Interrupt Factors and Timing Chart



9.2.5. PPG Timer Operation Flow

This section shows the PPG timer operation flow.

PPG Timer Operation Flow



9.2.6. Timer Control Registers (TMCR, TMCR2), Status Control Register (STC), Status Control Clear Register (STCC), and Status Control Set Register (STCS) When the PPG Timer is Selected

For details on writing to the status control register (STC), see "CHAPTER 40:8. Notes on Using the Base Timer."

Timer Control Register (Upper Byte of TMCR)

bit	15	14	13	12	11	10	9	8
Field	Reserved	CKS2	CKS1	CKS0	RTGEN	PMSK	EGS1	EGS0
R/W Attribute	R0,W0	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

[bit15] Reserved: Reserved Bit

The read value is "0".
If writing to this bit, write "0".

[TMCR2:bit8,bit14:12] CKS[3:0]: Count Clock Selection Bits

- These bits select a count clock for the 16-bit down counter.
- Any modification to the count clock is reflected immediately after the setting is changed. Therefore, modify CKS3 to CKS0 while counting is stopped (CTEN= 0). However, note that the bits can be modified at the same time as "1" is written to the CTEN bit.

Use of PWM Start Delay Mode and ADC Trigger Mode

Bits				Description
0	0	0	0	Internal clock
0	0	0	1	Internal clock divided by 4
0	0	1	0	Internal clock divided by 16
0	0	1	1	Internal clock divided by 128
0	1	0	0	Internal clock divided by 256
0	1	0	1	External clock (rising-edge event)
0	1	1	0	External clock (falling-edge event)
0	1	1	1	External clock (both-edges event)
1	0	0	0	Internal clock divided by 512
1	0	0	1	Internal clock divided by 1024
1	0	1	0	Internal clock divided by 2048
1	0	1	1	Internal clock divided by 2
1	1	0	0	Internal clock divided by 8
1	1	0	1	Internal clock divided by 32
1	1	1	0	Internal clock divided by 64
1	1	1	1	Setting prohibited

[bit11] RTGEN: Restart Enable Bit

This bit enables restart by a software trigger or trigger input.

Bit	Description
0	Disable restart.
1	Enable restart.

[bit10] PMSK: Pulse Output Mask Bit

- This bit controls the output waveform level of the PPG output waveform.
- When the bit is "0", the PPG waveform is output as is.
- When the bit is "1", PPG output is masked to L output regardless of the cycle or duty setting value.

Note:

- *If the output polarity specification bit (OSEL) in the timer control register (lower byte of TMCR) is set for inverted output, setting the PMSK bit to "1" results in masking to H output.*

Bit	Description
0	Normal output
1	Fixed at L output

[bit9:8] EGS[1:0]: Trigger Input Edge Selection Bits

- These bits select the valid edge for an input waveform as an external start factor, and they set trigger conditions.
- For the initial value or the 0b00 setting, no valid edge is selected for an input waveform, so no external waveform causes a start.

Note:

- *If "1" is written to the STRG bit, the software trigger becomes valid regardless of the EGS1 and EGS0 settings.*

- Modify EGS1 and EGS0 while counting is stopped (CTEN= 0). However, note that the bits can be modified at the same time as "1" is written to the CTEN bit.

Bits		Description
0	0	Trigger input invalid
0	1	Rising edge
1	0	Falling edge
1	1	Both edges

Timer Control Register (Lower Byte of TMCR)

bit	7	6	5	4	3	2	1	0
Field	Reserved	FMD2	FMD1	FMD0	OSEL	MDSE	CTEN	STRG
R/W Attribute	R0,W0	R/W	R/W	R/W	R/W	R/W	R/W	R0,W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

[bit7] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

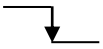
[bit6:4] FMD[2:0]: Timer Function Selection Bits

- These bits select the timer function.
- If 0b010 is set in the FMD2 to FMD0 bits, the PPG function is selected.
- Modify these bits while the timer is stopped (CTEN= 0). However, note that the bits can be modified at the same time as "1" is written to the CTEN bit.

Bits			Description
0	0	0	Reset mode
0	0	1	16-bit PWM timer
0	1	0	16-bit PPG timer
0	1	1	16/32-bit reload timer
1	0	0	16/32-bit PWC timer
1	0	1	Setting prohibited
1	1	0	
1	1	1	

[bit3] OSEL: Output Polarity Specification Bit

- This bit sets the PPG output polarity.

Polarity	After Reset	L Width Count End	H Width Count End
Normal	"L" output		
Inverse	"H" output		

Bit	Description
0	Normal polarity
1	Inverse polarity

[bit2] MDSE: Mode Selection Bit

- This bit selects either operation for continuous pulse output or one-shot operation for single-pulse output.
- Modify the bit while the timer is stopped (CTEN= 0). However, note that the bits can be modified at the same time as "1" is written to the CTEN bit.

Bit	Description
0	Continuous operation
1	One-shot operation

[bit1] CTEN: Count Operation Enable Bit

- This bit enables operation of the down counter.
- If "0" is written to this bit when counter operation is enabled (CTEN bit is "1"), the counter stops.

Bit	Description	
	Read	Write
0	Stop operation.	Set this bit to "0".
1	Enable operation.	Set this bit to "1".

[bit0] STRG: Software Trigger Bit

- If "1" is written to the STRG bit when the CTEN bit is "1", a software trigger is applied.
- The read value of the STRG bit is always "0".

Notes:

- *Even if "1" is written to the CTEN and STRG bits at the same time, a software trigger is applied.*
- *If "1" is written to the STRG bit, the software trigger becomes valid regardless of the EGS1 and EGS0 settings.*

Bit	Description
0	Invalid
1	Startup by software

Timer Control Register 2 (TMCR2)

bit	15	14	13	12	11	10	9	8
Field	Reserved							CKS3
R/W Attribute	R0,W0							R/W
Protection Attribute	-							-
Initial Value	0000000							0

Note: This register is located in the upper byte of the STC register.

[bit15:9] Reserved: Reserved Bits

The read value is "0".

If writing to this bit, write "0".

[bit8] CKS3: Count Clock Selection Bit

See "Count clock selection bits" in "9.2.6Timer Control Registers (TMCR, TMCR2), Status Control Register (STC), Status Control Clear Register (STCC), and Status Control Set Register (STCS) When the PPG Timer is Selected."

Status Control Register (STC)

bit	7	6	5	4	3	2	1	0
Field	Reserved	TGIE	Reserved	UDIE	Reserved	TGIR	Reserved	UDIR
R/W Attribute	R0,W0	R/W	R0,W0	R/W	R0,W0	R,WX	R0,W0	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

Note: The TMCR2 register is located in the upper byte of this register.

[bit7] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

[bit6] TGIE: Trigger Interrupt Request Enable Bit

- This bit controls interrupt requests of the trigger interrupt request bit (bit2 TGIR).
- If the TGIE bit is enabled and the TGIR bit is set to "1", an interrupt request is issued to the CPU.
- Writing "1" to the STCC:TGIEC bit clears this bit.
- Writing "1" to the STCS:TGIES bit sets this bit.

Bit	Description
0	Disable interrupt requests.
1	Enable interrupt requests.

[bit5] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

[bit4] UDIE: Underflow Interrupt Request Enable Bit

- This bit controls interrupt requests of the underflow interrupt request bit (bit0 UDIR).
- If the UDIE bit is enabled and the UDIR bit is set to "1", an interrupt request is issued to the CPU.
- Writing "1" to the STCC:UDIEC bit clears this bit.
- Writing "1" to the STCS:UDIES bit sets this bit.

Bit	Description
0	Disable interrupt requests.
1	Enable interrupt requests.

[bit3] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

[bit2] TGIR: Trigger Interrupt Request Bit

- The TGIR bit is set to "1" when a software trigger or trigger input is detected.
- Writing "1" to the STCC:TGIRC bit clears this bit.
- This bit is read-only. Writing data to this bit has no effect on operation.

Bit	Description
0	The interrupt factor is cleared.
1	Detect the interrupt factor.

[bit1] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

[bit0] UDIR: Underflow Interrupt Request Bit

- During counting from the set H width value, the UDIR bit is set to "1" when the count value underflows and changes from 0x0000 to 0xFFFF.
- Writing "1" to the STCC:UDIRC bit clears this bit.
- This bit is read-only. Writing data to this bit has no effect on operation.

Bit	Description
0	The interrupt factor is cleared.
1	Detect the interrupt factor.

Status Control Clear Register (STCC)

bit	7	6	5	4	3	2	1	0
Field	Reserved	TGIEC	Reserved	UDIEC	Reserved	TGIRC	Reserved	UDIRC
R/W Attribute	R0,W0	R0,W	R0,W0	R0,W	R0,W0	R0,W	R0,W0	R0,W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

Note: Reserved bits are located in the upper byte of this register.

[bit7] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

[bit6] TGIEC: Trigger Interrupt Request Enable Clear Bit

- If "1" is written to this bit, the STC:TGIE bit is cleared to "0".
- The read value of this bit is always "0".

Bit	Description
0	Invalid
1	Clear the TGIE bit.

[bit5] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

[bit4] UDIEC: Underflow Interrupt Request Enable Clear Bit

- If "1" is written to this bit, the STC:UDIE bit is cleared to "0".
- The read value of this bit is always "0".

Bit	Description
0	Invalid
1	Clear the UDIE bit.

[bit3] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

[bit2] TGIRC: Trigger Interrupt Request Clear Bit

- If "1" is written to this bit, the STC:TGIR bit is cleared to "0".
- The read value of this bit is always "0".

Bit	Description
0	Invalid
1	Clear the TGIR bit.

[bit1] Reserved: Reserved Bit

The read value is "0".
If writing to this bit, write "0".

[bit0] UDIRC: Underflow Interrupt Request Clear Bit

- If "1" is written to this bit, the STC:UDIR bit is cleared to "0".
- The read value of this bit is always "0".

Bit	Description
0	Invalid
1	Clear the UDIR bit.

Status Control Set Register (STCS)

bit	7	6	5	4	3	2	1	0
Field	Reserved	TGIES	Reserved	UDIES	Reserved			
R/W Attribute	R0,W0	R0,W	R0,W0	R0,W	R0,W0			
Protection Attribute	-	-	-	-	-			
Initial Value	0	0	0	0	0000			

Note: Reserved bits are located in the upper byte of this register.

[bit7] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

[bit6] TGIES: Trigger Interrupt Request Enable Set Bit

- If "1" is written to this bit, the STC:TGIE bit is set to "1".
- The read value of this bit is always "0".

Bit	Description
0	Invalid
1	Set the TGIE bit.

[bit5] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

[bit4] UDIES: Underflow Interrupt Request Enable Set Bit

- If "1" is written to this bit, the STC:UDIE bit is set to "1".
- The read value of this bit is always "0".

Bit	Description
0	Invalid
1	Set the UDIE bit.

[bit3:0] Reserved: Reserved Bits

The read value is "0".

If writing to this bit, write "0".

9.2.7. L Width Setting Reload Register (PRLL)

The L width setting reload register (PRLL) is the register used to set the L width of the PPG output waveform. An underflow at the activation trigger detection time or after the end of H width counting causes a transfer to the timer register.

bit	15	0
Field	PRLL[15:0]	
R/W Attribute	R/W	
Protection Attribute	-	
Initial Value	XXXXXXXXXXXXXXXX	

These bits compose the register used to set the L width of the PPG output waveform. An underflow at the activation trigger detection time or H width count end time causes a transfer to the timer register.

- Use 16- or 32-bit data access for the PRLL register.
- After configuring the PPG function with the FMD2 to FMD0 bits in the TMCR register, set the L width in the PRLL register.

9.2.8. H Width Setting Reload Register (PRLH)

The H width setting reload register (PRLH) is the register with a buffer used to set the H width of the PPG output waveform. An underflow when an activation trigger is detected or after H width counting ends causes a transfer from PRLH to the buffer register. An underflow at the L width count end time causes a transfer from the buffer register to the timer register.

bit	15	0
Field	PRLH[15:0]	
R/W Attribute	R/W	
Protection Attribute	-	
Initial Value	XXXXXXXXXXXXXXXX	

These bits compose the register used to set the H width of the PPG output waveform. An underflow at the activation trigger detection time or H width count end time causes a transfer from PRLH to the buffer register. An underflow at the L width count end time causes a transfer from the buffer register to the timer register.

- Use 16- or 32-bit data access for the PRLH register.
- After configuring the PPG function with the FMD2 to FMD0 bits in the TMCR register, set the H width in the PRLH register.

9.2.9. Timer Register (TMR)

The timer register (TMR) can read the value of the 16-bit down counter.

bit	15	0
Field	TMR[15:0]	
R/W Attribute	R,WX	
Protection Attribute	-	
Initial Value	0x0000	

The timer register can read the value of the 16-bit down counter.

- Use 16- or 32-bit data access for the TMR register.

9.3. Reload Timer Function

Only one of the following timer functions can be selected for the base timer in the FMD2 to FMD0 bit settings in the timer control register (TMCR): 16-bit PWM timer, 16-bit PPG timer, 16/32-bit reload timer, and 16/32-bit PWC timer. This section explains the timer function with the reload timer setting.

1. Operations of 16-Bit Reload Timer
2. Reload Timer Operation Flow
3. Timer Control Registers (TMCR, TMCR2), Status Control Register (STC), Status Control Clear Register (STCC), and Status Control Set Register (STCS) When the Reload Timer is Selected
4. Cycle Setting Register (PCSR)
5. Timer Register (TMR)

9.3.1. Operations of 16-Bit Reload Timer

The reload timer operates by performing a countdown from the set value in the cycle setting register, in synchronization with the count clock. When the count value becomes 0, the counting ends, or the timer automatically loads the cycle setting to continue operating until the countdown is stopped.

Count Operation When Internal Clock is Selected

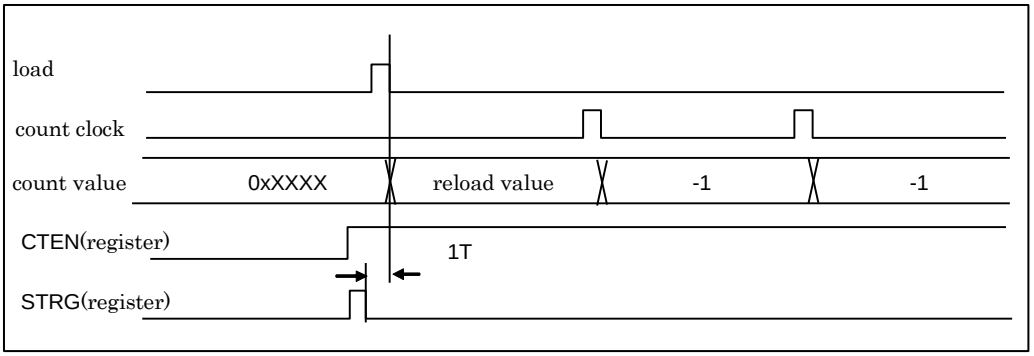
To start a count operation at the same time as counting is enabled, write "1" to both the CTEN bit and STRG bit in the timer control register. In a state where the timer has started (CTEN= 1), trigger input with the STRG bit is always enabled regardless of the operation mode.

With count operations enabled, the start of the timer by a software trigger or external trigger results in the cycle setting register value being loaded into the counter and a countdown being started.

The time required from the generation of a counter start trigger until the loading of cycle setting register data into the counter is 1T (T: Internal clock cycle).

Figure 9-20 shows the start and operation of the counter using a software trigger.

Figure 9-20 Count Operation When the Internal Clock is Selected



Underflow Operation

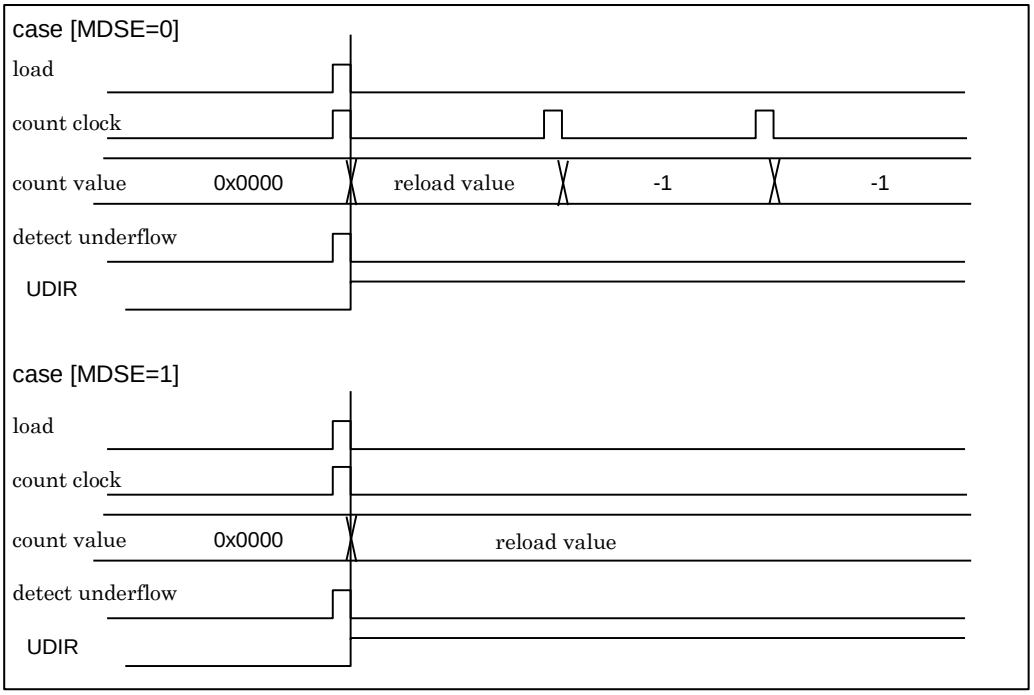
If the counter value changes from 0x0000 to 0xFFFF, an underflow has occurred. Therefore, an underflow occurs at the count of [set value of cycle setting register + 1].

The cycle setting register (PCSR) contents are loaded into the counter when an underflow occurs. If the MDSE bit in the timer control register (TMCR) is "0", the count operation continues. If the MDSE bit is "1", the count operation stops with the loaded counter value unchanged.

The UDIR bit in the status control register (STC) is set to "1" by an underflow. If the UDIE bit is "1" at this time, an interrupt request is generated.

Figure 9-21 shows an underflow operation timing chart.

Figure 9-21 Underflow Operation Timing Chart

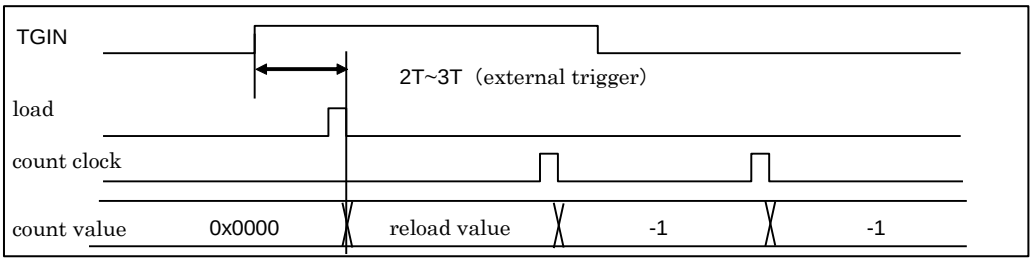


Operation of Input Pin Function

If the trigger input function (TMCR2:GATE= 0) is selected, the TGIN pin can be used as trigger input. When a valid edge is input to the TGIN pin, the cycle setting register contents are loaded into the counter, and a count operation begins. For the period from trigger application until the loading of a counter value, the required time is 2T to 3T (T: internal clock cycle).

Figure 9-22 shows trigger input operation where the valid edge specification is a rising edge.

Figure 9-22 Trigger Input Operation (TMCR2:GATE= 0)



If the gate function (TMCR2:GATE= 1) is selected, the TGIN pin can be used as the gate function.

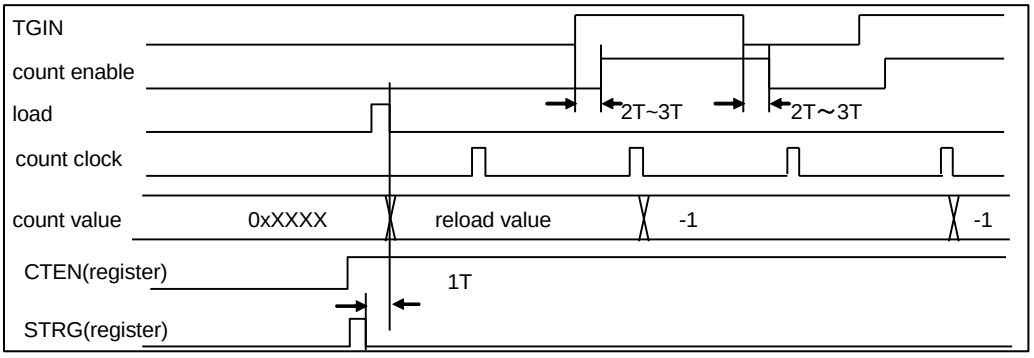
With count operations enabled (TMCR:CTEN= 1), the start of the timer by a software trigger (TMCR:STRG= 1) results in the cycle setting register (PCSR) value being loaded into the counter. The count clock counts down only for a period when a valid level is being input to the TGIN pin.

For synchronization of a signal input from the TGIN pin, the time required from the input of a valid level until count enable becomes effective is 2T to 3T for each of the start and end (T: internal clock cycle).

The gate function and trigger input function are controlled exclusively. Therefore, activation by an external event cannot be used when the gate function is in use.

Figure 9-23 shows gate function operation where the valid level is "H".

Figure 9-23 Gate Function Operation(TMCR2:GATE= 1)

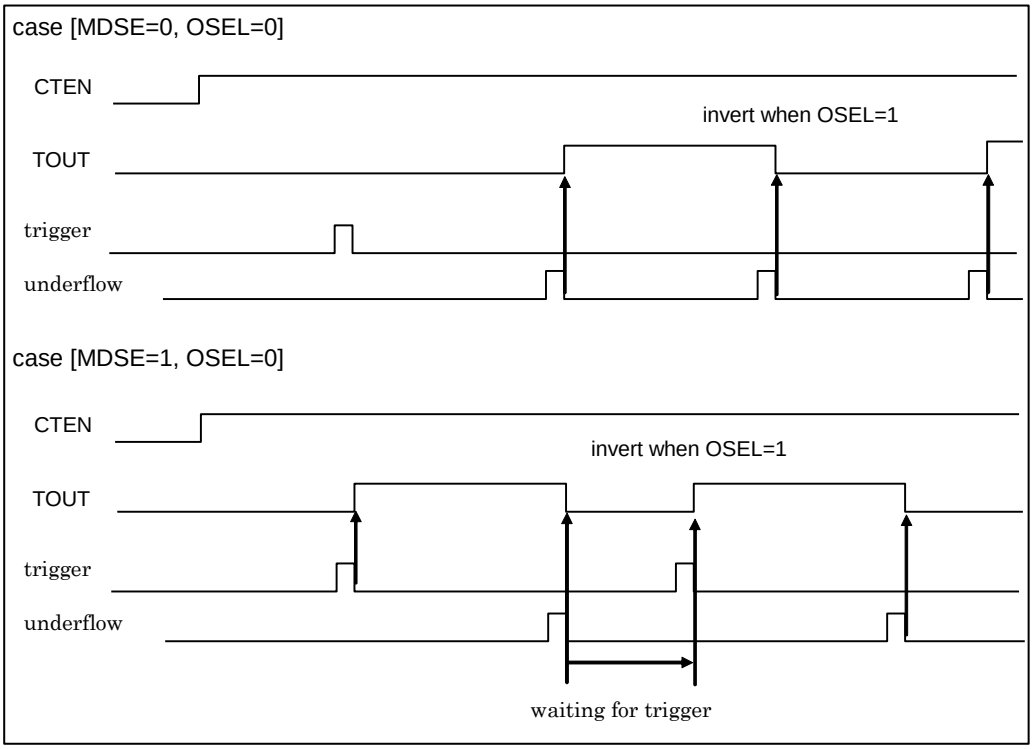


Operation of Output Pin Function

The TOUT output pin functions as toggle output in reload mode and as pulse output in one-shot mode. The toggle output is inverted by an underflow. The pulse output indicates that counting is in progress. The OSEL bit in the timer control register (TMCR) can set the output polarity. If OSEL is "0", the initial value is "0" for toggle output, and "1" is output during counting for one-shot pulse output. If OSEL is "1", the output waveform is inverted.

Figure 9-24 shows an output pin function operation timing chart.

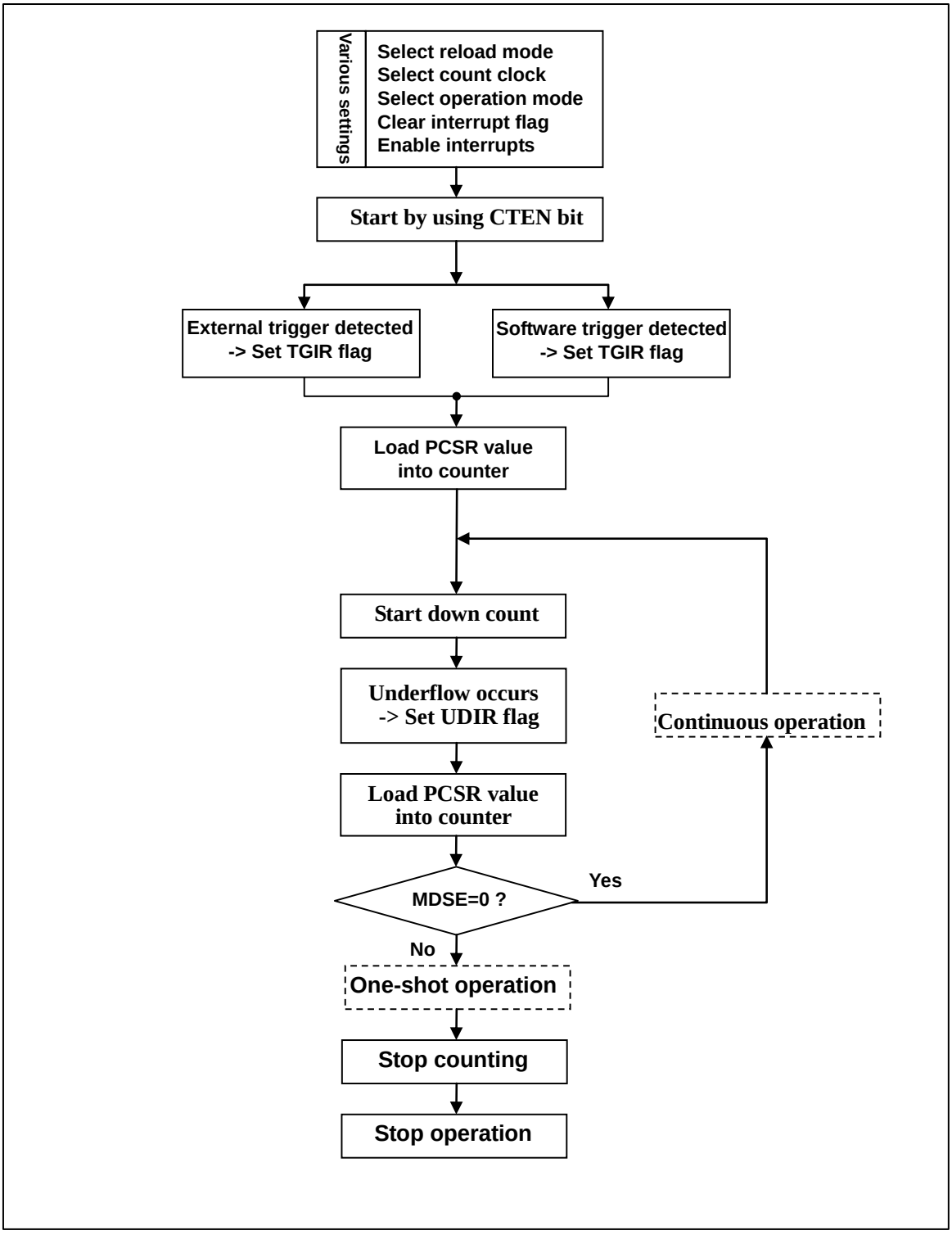
Figure 9-24 Output Pin Function Operation Timing Chart



9.3.2. Reload Timer Operation Flow

This section shows the reload timer operation flow.

Reload Timer Operation Flow



9.3.3. Timer Control Registers (TMCR, TMCR2), Status Control Register (STC), Status Control Clear Register (STCC), and Status Control Set Register (STCS) When the Reload Timer is Selected

The timer control register (TMCR) controls timer operation.

For details on writing to the status control register (STC), see "CHAPTER 40:8. Notes on Using the Base Timer."

Timer Control Register (Upper Byte of TMCR)

bit	15	14	13	12	11	10	9	8
Field	Reserved	CKS2	CKS1	CKS0	Reserved		EGS1	EGS0
R/W Attribute	R0,W0	R/W	R/W	R/W	R0,W0		R/W	R/W
Protection Attribute	-	-	-	-	-		-	-
Initial Value	0	0	0	0	00		0	0

[bit15] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

[TMCR2:bit8,bit14:12] CKS[3:0]: Count Clock Selection Bits

- These bits select a count clock for the 16-bit down counter.
- Any modification to the count clock is reflected immediately after the setting is changed. Therefore, modify CKS3 to CKS0 while counting is stopped (CTEN= 0). However, note that the bits can be modified at the same time as "1" is written to the CTEN bit.

Bits				Description
0	0	0	0	Internal clock
0	0	0	1	Internal clock divided by 4
0	0	1	0	Internal clock divided by 16
0	0	1	1	Internal clock divided by 128
0	1	0	0	Internal clock divided by 256
0	1	0	1	External clock (rising-edge event)
0	1	1	0	External clock (falling-edge event)
0	1	1	1	External clock (both-edges event)
1	0	0	0	Internal clock divided by 512
1	0	0	1	Internal clock divided by 1024
1	0	1	0	Internal clock divided by 2048
1	0	1	1	Internal clock divided by 2
1	1	0	0	Internal clock divided by 8
1	1	0	1	Internal clock divided by 32
1	1	1	0	Internal clock divided by 64
1	1	1	1	Setting prohibited

[bit11:10] Reserved: Reserved Bits

The read value is "0".

If writing to this bit, write "0".

[bit9:8] EGS[1:0]: Trigger Input Edge and Gate Function Level Selection Bits

- Trigger input selected (TMCR2:GATE= 0)
  - These bits select the valid edge for an input waveform as an external start factor, and they set trigger conditions.
  - For the initial value or the 0b00 setting, no valid edge is selected for an input waveform, so no external waveform causes a start.
- Gate function selected (TMCR2:GATE= 1)
  - These bits select a valid level corresponding to an input waveform as an external count factor, and the down counter counts down only while the selected level is valid.

Note:

- *If "1" is written to the STRG bit, the software trigger becomes valid regardless of the EGS1 and EGS0 settings.*

- Modify EGS1 and EGS0 while counting is stopped (CTEN= 0). However, note that the bits can be modified at the same time as "1" is written to the CTEN bit.

Bits		Description	
		Trigger Input Selected (TMCR2:GATE= 0)	**Gate Function Selected (TMCR2:GATE= 1)**
0	0	Trigger input invalid	"L" level
0	1	External trigger(rising edge)	"H" level
1	0	External trigger(falling edge)	"L" level
1	1	External trigger(both edges)	"H" level

Timer Control Register (Lower Byte of TMCR)

bit	7	6	5	4	3	2	1	0
Field	T32	FMD2	FMD1	FMD0	OSEL	MDSE	CTEN	STRG
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R0,W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

[bit7] T32: 32-Bit Timer Selection Bit

- This bit selects the 32-bit timer function.
- If the reload timer function is selected with 0b011 set in the FMD2 to FMD0 bits, setting the T32 bit to "1" results in operation in 32-bit timer mode.
- Modify these bits while the timer is stopped (CTEN= 0). However, note that the bit can be modified at the same time as "1" is written to the CTEN bit. (See 32-Bit Mode Operation.)

Bit	Description
0	16-bit timer mode
1	32-bit timer mode

[bit6:4] FMD[2:0]: Timer Function Selection Bits

- These bits select the timer function.
- If 0b011 is set in the FMD2 to FMD0 bits, the reload timer function is selected.
- Modify these bits while the timer is stopped (CTEN= 0). However, note that the bits can be modified at the same time as "1" is written to the CTEN bit.

Bits			Description
0	0	0	Reset mode
0	0	1	16-bit PWM timer
0	1	0	16-bit PPG timer
0	1	1	16/32-bit reload timer
1	0	0	16/32-bit PWC timer
Other than above			Setting prohibited

[bit3] OSEL: Output Polarity Specification Bit

- This bit selects either normal or inverse output as the timer output level.
- When combined with the mode selection bit (bit2 MDSE), this bit generates an output waveform as follows.

MDSE	OSEL	Output Waveform
0	0	Toggle output of "L" at the count start time
0	1	Toggle output of "H" at the count start time
1	0	Rectangular output of "H" during counting
1	1	Rectangular output of "L" during counting

Bit	Description
0	Normal polarity
1	Inverse polarity

[bit2] MDSE: Mode Selection Bit

- If the MDSE bit is set to "0", reload mode is selected. The cycle setting register (PCSR) value is loaded into the counter at the same time that the count value underflows from 0x0000 to 0xFFFF, and the count operation continues.
- If the MDSE bit is set to "1", one-shot mode is selected. When the count value underflows from 0x0000 to 0xFFFF, operation stops.
- Modify these bits while the timer is stopped (CTEN= 0). However, note that the bits can be modified at the same time as "1" is written to the CTEN bit.

Bit	Description
0	Reload mode
1	One-shot mode

[bit1] CTEN: Timer Enable Bit

- This bit enables operation of the down counter.
- If "0" is written to this bit when counter operation is enabled (CTEN bit is "1"), the counter stops.

Bit	Description	
	Read	Write
0	Stop operation.	Set this bit to "0".
1	Enable operation.	Set this bit to "1".

[bit0] STRG: Software Trigger Bit

- If "1" is written to the STRG bit when the CTEN bit is "1", a software trigger is applied.
- The read value of the STRG bit is always "0".

Notes:

- *Even if "1" is written to the CTEN and STRG bits at the same time, a software trigger is applied.*
- *If "1" is written to the STRG bit, the software trigger becomes valid regardless of the EGS1 and EGS0 settings.*

Bit	Description
0	Invalid
1	Startup by software

Timer Control Register 2 (TMCR2)

bit	15	14	13	12	11	10	9	8
Field	GATE	Reserved						CKS3
R/W Attribute	R/W	R0,W0						R/W
Protection Attribute	-	-						-
Initial Value	0	000000						0

Note: This register is located in the upper byte of the STC register.

[bit15] GATE: Gate Input Enable Bit

This bit selects whether to use the external factor pin for the trigger input function or gate function during reload timer operation.

- Trigger input function: A countdown begins when a valid edge is input to the external factor pin.
- Gate function: A countdown is performed only while a valid level is being input to the external factor pin.

After configuring the reload timer function with the FMD2 to FMD0 bits in the TMCR register, set the GATE bit.

Bit	Description
0	Trigger input function
1	Gate function

[bit14:9] Reserved: Reserved Bits

The read value is "0".
If writing to this bit, write "0".

[bit8] CKS3: Count Clock Selection Bit

See "Count clock selection bits" in "9.3.3Timer Control Registers (TMCR, TMCR2), Status Control Register (STC), Status Control Clear Register (STCC), and Status Control Set Register (STCS) When the Reload Timer is Selected."

Status Control Register (STC)

bit	7	6	5	4	3	2	1	0
Field	Reserved	TGIE	Reserved	UDIE	Reserved	TGIR	Reserved	UDIR
R/W Attribute	R0,W0	R/W	R0,W0	R/W	R0,W0	R,WX	R0,W0	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

Note: The TMCR2 register is located in the upper byte of this register.

[bit7] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

[bit6] TGIE: Trigger Interrupt Request Enable Bit

- This bit controls interrupt requests of the trigger interrupt request bit (bit2 TGIR).
- If the TGIE bit is enabled and the TGIR bit is set to "1", an interrupt request is issued to the CPU.
- Writing "1" to the STCC:TGIEC bit clears this bit.
- Writing "1" to the STCS:TGIES bit sets this bit.

Bit	Description
0	Disable interrupt requests.
1	Enable interrupt requests.

[bit5] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

[bit4] UDIE: Underflow Interrupt Request Enable Bit

- This bit controls interrupt requests of the underflow interrupt request bit (bit0 UDIR).
- If the UDIE bit is enabled and the UDIR bit is set to "1", an interrupt request is issued to the CPU.
- Writing "1" to the STCC:UDIEC bit clears this bit.
- Writing "1" to the STCS:UDIES bit sets this bit.

Bit	Description
0	Disable interrupt requests.
1	Enable interrupt requests.

[bit3] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

[bit2] TGIR: Trigger Interrupt Request Bit

- The TGIR bit is set to "1" when a software trigger or trigger input is detected.
- Writing "1" to the STCC:TGIRC bit clears this bit.
- This bit is read-only. Writing data to this bit has no effect on operation.

Bit	Description
0	The interrupt factor is cleared.
1	Detect the interrupt factor.

[bit1] Reserved: Reserved Bit

The read value is "0".
If writing to this bit, write "0".

[bit0] UDIR: Underflow Interrupt Request Bit

- The UDIR bit is set to "1" when the count value underflows and changes from 0x0000 to 0xFFFF.
- Writing "1" to the STCC:UDIRC bit clears this bit.
- This bit is read-only. Writing data to this bit has no effect on operation.

Bit	Description
0	The interrupt factor is cleared.
1	Detect the interrupt factor.

Status Control Clear Register (STCC)

bit	7	6	5	4	3	2	1	0
Field	Reserved	TGIEC	Reserved	UDIEC	Reserved	TGIRC	Reserved	UDIRC
R/W Attribute	R0,W0	R0,W	R0,W0	R0,W	R0,W0	R0,W	R0,W0	R0,W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

Note: Reserved bits are located in the upper byte of this register.

[bit7] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

[bit6] TGIEC: Trigger Interrupt Request Enable Clear Bit

- If "1" is written to this bit, the STC:TGIE bit is cleared to "0".
- The read value of this bit is always "0".

Bit	Description
0	Invalid
1	Clear the TGIE bit.

[bit5] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

[bit4] UDIEC: Underflow Interrupt Request Enable Clear Bit

- If "1" is written to this bit, the STC:UDIE bit is cleared to "0".
- The read value of this bit is always "0".

Bit	Description
0	Invalid
1	Clear the UDIE bit.

[bit3] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

[bit2] TGIRC: Trigger Interrupt Request Clear Bit

- If "1" is written to this bit, the STC:TGIR bit is cleared to "0".
- The read value of this bit is always "0".

Bit	Description
0	Invalid
1	Clear the TGIR bit.

[bit1] Reserved: Reserved Bit

The read value is "0".
If writing to this bit, write "0".

[bit0] UDIRC: Underflow Interrupt Request Clear Bit

- If "1" is written to this bit, the STC:UDIR bit is cleared to "0".
- The read value of this bit is always "0".

Bit	Description
0	Invalid
1	Clear the UDIR bit.

Status Control Set Register (STCS)

bit	7	6	5	4	3	2	1	0
Field	Reserved	TGIES	Reserved	UDIES	Reserved			
R/W Attribute	R0,W0	R0,W	R0,W0	R0,W	R0,W0			
Protection Attribute	-	-	-	-	-			
Initial Value	0	0	0	0	0000			

Note: Reserved bits are located in the upper byte of this register.

[bit7] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

[bit6] TGIES: Trigger Interrupt Request Enable Set Bit

- If "1" is written to this bit, the STC:TGIE bit is set to "1".
- The read value of this bit is always "0".

Bit	Description
0	Invalid
1	Set the TGIE bit.

[bit5] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

[bit4] UDIES: Underflow Interrupt Request Enable Set Bit

- If "1" is written to this bit, the STC:UDIE bit is set to "1".
- The read value of this bit is always "0".

Bit	Description
0	Invalid
1	Set the UDIE bit.

[bit3:0] Reserved: Reserved Bits

The read value is "0".

If writing to this bit, write "0".

9.3.4. Cycle Setting Register (PCSR)

The cycle setting register (PCSR) is a register that retains the initial count value. In 32-bit mode, this value for the even-numbered channel is the initial value of the lower 16-bit count. This value for the odd-numbered channel is the initial value of the upper 16-bit count. The initial value at the reset time is undefined. Be sure to access this register with 16- or 32-bit data transfer instructions.

bit	15	0
Field	PCSR[15:0]	
R/W Attribute	R/W	
Protection Attribute	-	
Initial Value	XXXXXXXXXXXXXXXX	

These bits compose the register used to set a cycle. An underflow causes a transfer to the timer register.

- Use 16- or 32-bit data access for the PCSR register.
- After configuring the reload timer function with the FMD2 to FMD0 bits in the TMCR register, set a cycle in the PCSR register.
- To write data to the PCSR register in 32-bit mode, access first the upper 16-bit data (data for the odd-numbered channel) and then the lower 16-bit data (data for the even-numbered channel).

9.3.5. Timer Register (TMR)

The timer register (TMR) is a register that can read the count value of a timer. In 32-bit mode, this value for the even-numbered channel is the value of the lower 16-bit count. This value for the odd-numbered channel is the value of the upper 16-bit count. The initial value is undefined.

Be sure to read this register with 16- or 32-bit data transfer instructions.

bit	15	0
Field	TMR[15:0]	
R/W Attribute	R,WX	
Protection Attribute	-	
Initial Value	XXXXXXXXXXXXXXXX	

The timer register can read the value of the 16-bit down counter.

- Use 16- or 32-bit data access for the TMR register.
- To read the TMR register in 32-bit mode, access first the lower 16-bit data (data for the even-numbered channel) and then the upper 16-bit data (data for the odd-numbered channel).

9.4. PWC Timer Function

Only one of the following timer functions can be selected for the base timer in the FMD2 to FMD0 bit settings in the timer control register (TMCR): 16-bit PWM timer, 16-bit PPG timer, 16/32-bit reload timer, and 16/32-bit PWC timer. This section explains the timer function with the PWC setting.

1. Operations of PWC Timer
2. Timer Control Registers (TMCR, TMCR2), Status Control Register (STC), Status Control Clear Register (STCC), and Status Control Set Register (STCS)When the PWC Timer is Selected
3. Data Buffer Register (DTBF)

9.4.1. Operations of PWC Timer

The PWC timer has a pulse width measurement function. The timer can select twelve types of count clocks, and it can measure the time and cycle between any input pulse events by using a counter. This section shows the basic functions/operations of the pulse width measurement function.

Pulse Width Measurement Function

After the start, the function does not perform a count operation until the counter is cleared to 0x0000 and the set measurement start edge is input. The function starts counting up from 0x0001 when the measurement start edge is detected. It stops counting when the measurement end edge is detected. The count value at this time is stored as a pulse width in the register.

An interrupt request can be generated at the measurement end time and at the overflow occurrence time.

After measurement ends, the function operates according to the measurement mode as follows.

- In single measurement mode: It stops operating.
- In continuous measurement mode: It first transfers the counter value to the buffer register and then stops counting until the measurement start edge is input again.

Figure 9-25 Pulse Width Measurement Operation (Single Measurement Mode/"H" Width Measurement)

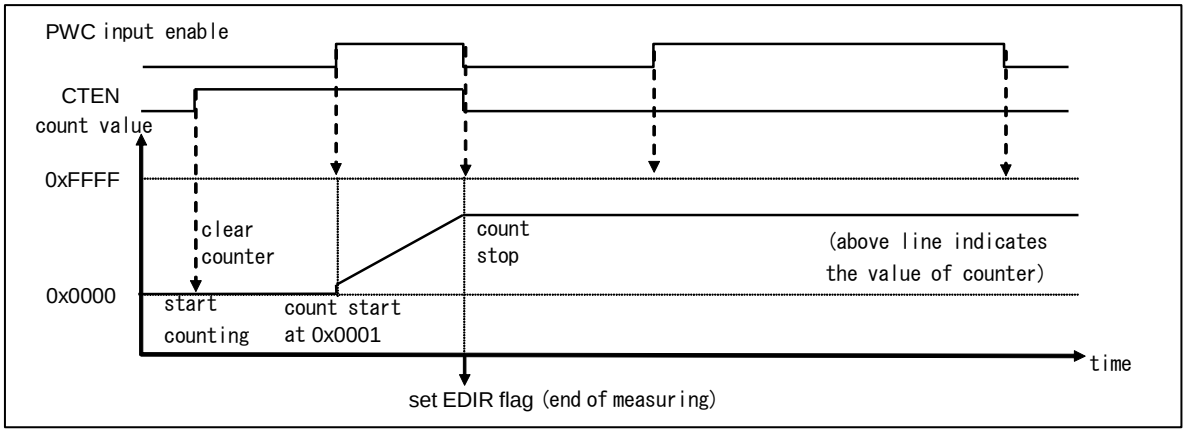
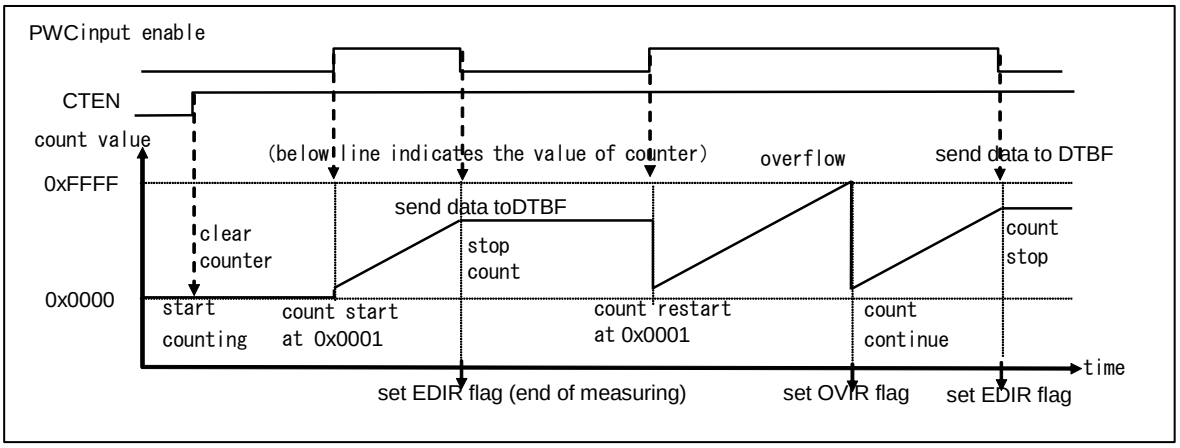


Figure 9-26 Pulse Width Measurement Operation (Continuous Measurement Mode/"H" Width Measurement)



Selection of Count Clock

Twelve types of count clocks can be selected for the counter through the setting of TMCR2 register bit8 (CKS3) and TMCR register bit14 to 12 (CKS2, CKS1, and CKS0).

The following count clocks can be selected.

TMCR2 and TMCR Registers CKS3, CKS2, CKS1, and CKS0 Bits	Internal Count Clock Selected
0000	Internal clock [initial value]
0001	Internal clock divided by 4
0010	Internal clock divided by 16
0011	Internal clock divided by 128
0100	Internal clock divided by 256
0101	Setting prohibited
0110	
0111	
1000	Internal clock divided by 512
1001	Internal clock divided by 1024
1010	Internal clock divided by 2048
1011	Internal clock divided by 2
1100	Internal clock divided by 8
1101	Internal clock divided by 32
1110	Internal clock divided by 64
1111	Setting prohibited

The initial value selected after a reset is the internal clock.

Be sure to select a count clock before starting the counter.

Selection of Operation Mode

To select each operation mode/measurement mode, set TMCR.

Operation mode setting ... TMCR bit10 to 8: EGS2, EGS1, EGS0 (Select a measurement edge.)
Measurement mode setting ... TMCR bit2: MDSE (Select single measurement/continuous measurement.)

The following table lists a selection of operation modes.

Operation Mode		MDSE	EGS2	EGS1	EGS0
Rising to falling H pulse width measurement	Continuous measurement mode: Buffer enabled	0	0	0	0
	Single measurement mode: Buffer disabled	1	0	0	0
Rising to rising Cycle measurement between rising edges	Continuous measurement mode: Buffer enabled	0	0	0	1
	Single measurement mode: Buffer disabled	1	0	0	1
Falling to falling Cycle measurement between falling edges	Continuous measurement mode: Buffer enabled	0	0	1	0
	Single measurement mode: Buffer disabled	1	0	1	0
Measurement between rising or falling and falling or rising	Continuous measurement mode: Buffer enabled	0	0	1	1
	Single measurement mode: Buffer disabled	1	0	1	1
Falling to rising L pulse width measurement	Continuous measurement mode: Buffer enabled	0	1	0	0
	Single measurement mode: Buffer disabled	1	1	0	0
Setting prohibited		0	1	0	1
		1	1	0	1
		0	1	1	0
		1	1	1	0
		0	1	1	1
		1	1	1	1

The initial values selected after a reset are H pulse width measurement and continuous measurement mode.

Be sure to select an operation mode before starting the counter.

Starting and Stopping Pulse Width Measurement

To start/restart/forcibly stop each operation, set bit1 (CTEN bit) in TMCR.

Pulse width measurement is started/restarted by the writing of "1" to the CTEN bit and forcibly stopped by the writing of "0" to the CTEN bit.

CTEN	Function
1	Start/Restart pulse width measurement.
0	Stop pulse width measurement.

Post-Start Operation

The operations after the start of pulse width measurement mode do not include counting until the measurement start edge is input. After the detection of the measurement start edge, the 16-bit up counter starts counting from 0x0001.

Restart

After the start, a repeated start performed during operation (writing "1" again in a state where the CTEN bit is "1") is called a "restart." The operation in any such restart is described below.

- In the measurement start edge wait state:
  There is no effect on operation.
- During measurement:
  The count is cleared to 0x0000. Then, the measurement start edge wait state begins again. If the measurement end edge is detected at the same time as a restart, the measurement end flag (EDIR) is set to "1". Then, if the mode is continuous measurement mode, the measurement results are transferred to DTBF.

About Stopping

In single measurement mode, the count operation is automatically stopped by a counter overflow or the end of measurement, so stopping it does not need to be a concern. In continuous measurement mode, to stop counting before counting is automatically stopped, you need to forcibly stop it.

Counter Clearing and Initial Value

The 16-bit up counter is cleared to 0x0000 in the following cases.

- Upon a reset
- When "1" is written to bit1 (CTEN bit) in TMCR (even including restart times)

The 16-bit up counter is initialized to 0x0001 in the following case.

- When the measurement start edge is detected

Pulse Width Measurement Operation Details

Single Measurement and Continuous Measurement

The modes for pulse width measurement are a mode where measurement is 1-time only and a mode where measurement is continuous. Each mode is selected with the MDSE bit in TMCR. (See "Selection of Operation .") The modes differ as described below.

Single measurement mode:

The first input of the measurement end edge stops the counting by the counter and sets the measurement end flag (EDIR) in STC to "1". No subsequent measurements are made. However, if a restart is performed at the same time, the mode enters the measurement start wait state.

Continuous measurement mode:

The input of the measurement end edge stops the counting by the counter and sets the measurement end flag (EDIR) in STC to "1". The counting remains stopped until the measurement start edge is input again. The counter is initialized to 0x0001 and measurement begins when the measurement start edge is input again. The measurement results of the counter are transferred to DTBF at the measurement end time.

Be sure to select/change the measurement mode while the counter is stopped.

Measurement Result Data

The handling of measurement results and counter values and the function of DTBF differ between single measurement mode and continuous measurement mode. The measurement results from each mode differ as described below.

Single measurement mode:

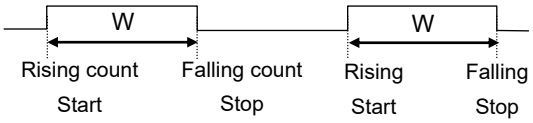
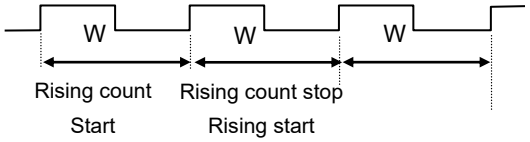
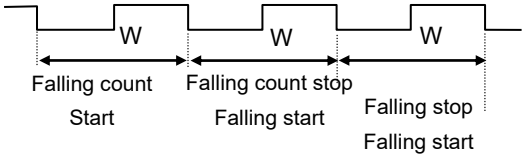
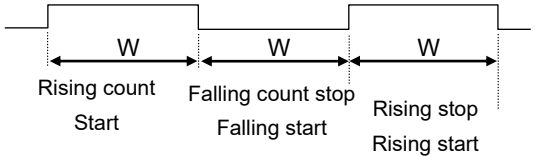
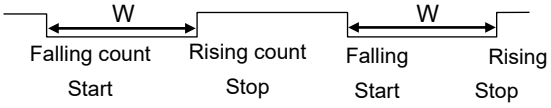
If DTBF is read during operation, the count value being measured is obtained.
If DTBF is read after measurement ends, the measurement result data is obtained.

Continuous measurement mode:

The measurement results of the counter are transferred to DTBF at the measurement end time. The immediately preceding measurement results are obtained when DTBF is read, and the last measurement results are retained even during a measurement operation. The count value being measured cannot be read.
In continuous measurement mode, if the next measurement ends before the measurement results are read, the last measurement results are overwritten by the new measurement results. In such cases, the error flag (ERR) is set to "1" in STC. The error flag (ERR) is automatically cleared when DTBF is read.

Measurement Modes and Count Operations

The measurement mode can be selected from the following five types, depending on where the input pulse is measured. The following table explains them.

Measurement Mode	EGS2, 1, 0	Measurement Description (W: Pulse Width Measured)
H pulse width measurement	000	 The width of the "H" period is measured. Count (measurement)start: Upon detection of a rising edge Count (measurement)end: Upon detection of a falling edge
Cycle measurement between rising edges	001	 The cycle between rising edges is measured. Count (measurement)start: Upon detection of a rising edge Count (measurement) end: Upon detection of a rising edge
Cycle measurement between falling edges	010	 The cycle between falling edges is measured. Count (measurement) start: Upon detection of a falling edge Count (measurement)end: Upon detection of a falling edge
Pulse width measurement between all edges	011	 The width between continuously input edges is measured. Count (measurement) start: Upon detection of an edge Count (measurement) end: Upon detection of an edge
L pulse width measurement	100	 The width of the "L" period is measured. Count (measurement) start: Upon detection of a falling edge Count (measurement) end: Upon detection of a rising edge

In either measurement mode, after the counter is cleared to 0x0000 by the start of measurement, the counter does not do any counting until the measurement start edge is input. From the input of the measurement start edge, the counter continues counting up for each count clock until the measurement end edge is input.

For measurements such as pulse width measurement and cycle measurement between all edges in continuous measurement mode, the end edge is the next measurement start edge.

Pulse Width/Cycle Calculation Method

The method of calculating the pulse width/cycle from the obtained measurement result data in DTBF after measurement ends is shown below.

TW = n x t	TW: Measured pulse width/cycle n: Measurement result data in DTBF t: Count clock cycle

Interrupt Request Generation

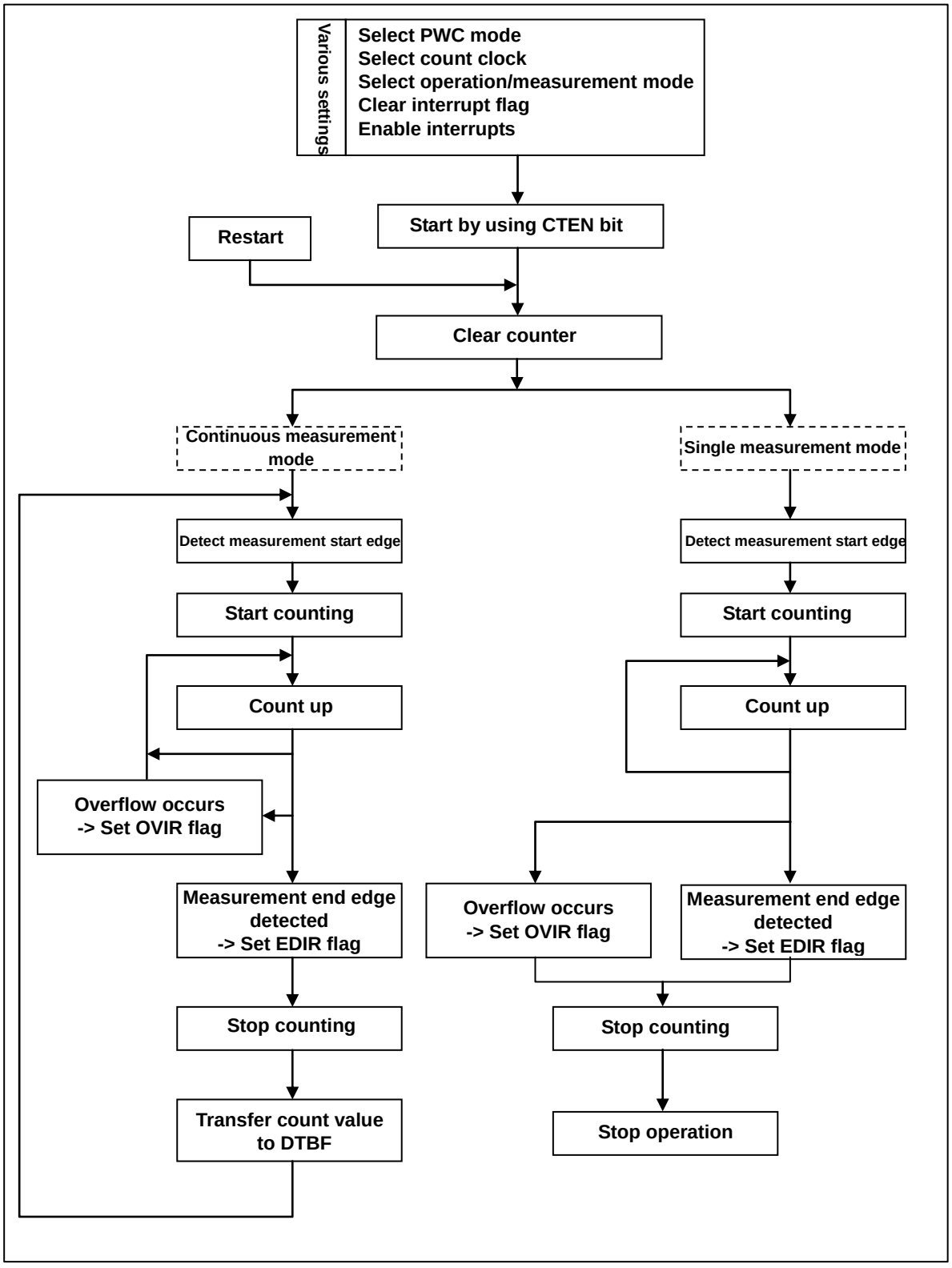
The following two interrupt requests can be generated.

- Interrupt request due to a counter overflow
  If counting up during measurement causes an overflow, the overflow flag (OVIR) is set to "1". Furthermore, if overflow interrupt requests are enabled, an interrupt request is generated.
- Interrupt request due to measurement end
  If the measurement end edge is detected, the measurement end flag (EDIR) in STC is set to "1". Furthermore, if measurement end interrupt requests are enabled, an interrupt request is generated.

  The measurement end flag (EDIR) is automatically cleared by the reading of the measurement results, DTBF.

Pulse Width Measurement Operation Flow

Figure 9-27 Pulse Width Measurement Operation Flow



9.4.2. Timer Control Registers (TMCR, TMCR2), Status Control Register (STC), Status Control Clear Register (STCC), and Status Control Set Register (STCS)When the PWC Timer is Selected

The timer control register (TMCR) controls timer operation.

For details on writing to the status control register (STC), see "CHAPTER 40:8. Notes on Using the Base Timer."

Timer Control Register (Upper Byte of TMCR)

Bit	15	14	13	12	11	10	9	8
Field	Reserved	CKS2	CKS1	CKS0	Reserved	EGS2	EGS1	EGS0
R/W Attribute	R0,W0	R/W	R/W	R/W	R0,W0	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

[bit15] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

[TMCR2:bit8,bit14:12] CKS[3:0]: Count Clock Selection Bits

- These bits select a count clock for the 16-bit down counter.
- Any modification to the count clock is reflected immediately after the setting is changed. Therefore, modify CKS3 to CKS0 while counting is stopped (CTEN= 0). However, note that the bits can be modified at the same time as "1" is written to the CTEN bit.

Bits				Description
0	0	0	0	Internal clock
0	0	0	1	Internal clock divided by 4
0	0	1	0	Internal clock divided by 16
0	0	1	1	Internal clock divided by 128
0	1	0	0	Internal clock divided by 256
0	1	0	1	Setting prohibited
0	1	1	0	
0	1	1	1	
1	0	0	0	Internal clock divided by 512
1	0	0	1	Internal clock divided by 1024
1	0	1	0	Internal clock divided by 2048
1	0	1	1	Internal clock divided by 2
1	1	0	0	Internal clock divided by 8
1	1	0	1	Internal clock divided by 32
1	1	1	0	Internal clock divided by 64
1	1	1	1	Setting prohibited

[bit11] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

[bit10:8] EGS[2:0]: Measurement Edge Selection Bits

- These bits set a measurement edge condition.
- Modify EGS2 to EGS0 while counting is stopped (CTEN= 0). However, note that the bits can be modified at the same time as "1" is written to the CTEN bit.

Bits			Description
0	0	0	"H" pulse width measurement (rising to falling)
0	0	1	Cycle measurement between rising edges (rising to rising)
0	1	0	Cycle measurement between falling edges (falling to falling)
0	1	1	Pulse width measurement between all edges (rising or falling to falling or rising)
1	0	0	"L" pulse width measurement (falling to rising)
1	0	1	Setting prohibited
1	1	0	
1	1	1	

Timer Control Register (Lower Byte of TMCR)

bit	7	6	5	4	3	2	1	0
Field	T32	FMD2	FMD1	FMD0	Reserved	MDSE	CTEN	Reserved
R/W Attribute	R/W	R/W	R/W	R/W	R0,W0	R/W	R,W	R0,W0
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

[bit7] T32: 32-Bit Timer Selection Bit

- This bit selects the 32-bit timer function.
- If the PWC function is selected with 0b100 set in the FMD2 to FMD0 bits, setting the T32 bit to "1" results in operation in 32-bit PWC mode.
- Modify these bits while the timer is stopped (CTEN= 0). However, note that the bit can be modified at the same time as "1" is written to the CTEN bit. (See 32-Bit Mode Operation.)

Bit	Description
0	16-bit timer mode
1	32-bit timer mode

[bit6:4] FMD[2:0]: Timer Function Selection Bits

- These bits select the timer function.
- If 0b100 is set in the FMD2 to FMD0 bits, the PWC timer function is selected.
- Modify these bits while the timer is stopped (CTEN= 0). However, note that the bits can be modified at the same time as "1" is written to the CTEN bit.

Bits			Description
0	0	0	Reset mode
0	0	1	16-bit PWM timer
0	1	0	16-bit PPG timer
0	1	1	16/32-bit reload timer
1	0	0	16/32-bit PWC timer
1	0	1	Setting prohibited
1	1	0	
1	1	1	

[bit3] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

[bit2] MDSE: Mode Selection Bit

- Modify these bits while the timer is stopped (CTEN= 0). However, note that the bits can be modified at the same time as "1" is written to the CTEN bit.

Bit	Description
0	Continuous measurement mode (buffer register enabled)
1	Single measurement mode (stop after 1 measurement)

[bit1] CTEN: Timer Enable Bit

- This bit enables the starting or restarting of the up counter.
- If "1" is written with the counter in the operation enabled state (CTEN bit is "1"), a restart is assumed, the counter is cleared, and operation enters the measurement start edge wait state.
- If "0" is written to this bit when counter operation is enabled (CTEN bit is "1"), the counter stops.
- After measurement ends in single measurement mode, CTEN is cleared.

Bit	Description	
	Read	Write
0	Stop operation.	Set this bit to "0".
1	Enable operation.	Set this bit to "1".

[bit0] Reserved: Reserved Bit

The read value is "0".
If writing to this bit, write "0".

Timer Control Register 2 (TMCR2)

bit	15	14	13	12	11	10	9	8
Field	Reserved							CKS3
R/W Attribute	R0,W0							R/W
Protection Attribute	-							-
Initial Value	0000000							0

Note: This register is located in the upper byte of the STC register.

[bit15:9] Reserved: Reserved Bits

The read value is "0".

If writing to this bit, write "0".

[bit8] CKS3: Count Clock Selection Bit

See "Count clock selection bits" in "9.4.2Timer Control Registers (TMCR, TMCR2), Status Control Register (STC), Status Control Clear Register (STCC), and Status Control Set Register (STCS)When the PWC Timer is Selected."

Status Control Register (STC)

bit	7	6	5	4	3	2	1	0
Field	ERR	EDIE	Reserved	OVIE	Reserved	EDIR	Reserved	OVIR
R/W Attribute	R,WX	R/W	R0,W0	R/W	R0,W0	R,WX	R0,W0	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

Note: The TMCR2 register is located in the upper byte of this register.

[bit7] ERR: Error Flag Bit

- This bit is a flag indicating that, in continuous measurement mode, the next measurement has ended before the measurement results in the DTBF register have been read. In this case, the DTBF register values are updated with the new measurement results, so the immediately preceding measurement results are lost.
- Measurement continues regardless of the ERR bit value.
- The ERR bit is read-only. Writing to the bit has no effect on the bit value.
- Reading the measurement results (DTBF) clears the ERR bit.

Bit	Description
0	Normal state
1	Overwrite any unread measurement results with the next measurement results.

[bit6] EDIE: Measurement End Interrupt Request Enable Bit

- This bit controls interrupt requests of the measurement end interrupt request bit (bit2 EDIR).
- If the EDIE bit is enabled and the EDIR bit is set to "1", an interrupt request is issued to the CPU.
- Writing "1" to the STCC:EDIEC bit clears this bit.
- Writing "1" to the STCS:EDIES bit sets this bit.

Bit	Description
0	Disable interrupt requests.
1	Enable interrupt requests.

[bit5] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

[bit4] OVIE: Overflow Interrupt Request Enable Bit

- This bit controls interrupt requests of the overflow interrupt request bit (bit0 OVIR).
- If the OVIE bit is enabled and the OVIR bit is set to "1", an interrupt request is issued to the CPU.
- Writing "1" to the STCC:OVIEC bit clears this bit.
- Writing "1" to the STCS:OVIES bit sets this bit.

Bit	Description
0	Disable interrupt requests.
1	Enable interrupt requests.

[bit3] Reserved: Reserved Bit

The read value is "0".
If writing to this bit, write "0".

[bit2] EDIR: Measurement End Interrupt Request Bit

- This bit indicates that measurement has ended by setting the flag to "1" at the end time.
- Reading the measurement results (DTBF) clears the EDIR bit. The EDIR bit is read-only. Writing to the bit has no effect on the bit value.

Bit	Description
0	Read measurement results (DTBF).
1	Detect the interrupt factor.

[bit1] Reserved: Reserved Bit

The read value is "0".
If writing to this bit, write "0".

[bit0] OVIR: Overflow Interrupt Request Bit

- The flag is set to "1" when the count value overflows from 0xFFFF to 0x0000.
- Writing "1" to the STCC:OVIRC clears this bit.
- This bit is read-only. Writing data to this bit has no effect on operation.

Bit	Description
0	The interrupt factor is cleared.
1	Detect the interrupt factor.

Status Control Clear Register (STCC)

bit	7	6	5	4	3	2	1	0
Field	Reserved	EDIEC	Reserved	OVIEC	Reserved			OVIRC
R/W Attribute	R0,W0	R0,W	R0,W0	R0,W	R0,W0			R0,W
Protection Attribute	-	-	-	-	-			-
Initial Value	0	0	0	0	000			0

Note: Reserved bits are located in the upper byte of this register.

[bit7] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

[bit6] EDIEC: Measurement End Interrupt Request Enable Clear Bit

- If "1" is written to this bit, the STC:EDIE bit is cleared to "0".
- The read value of this bit is always "0".

Bit	Description
0	Invalid
1	Clear the EDIE bit.

[bit5] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

[bit4] OVIEC: Overflow Interrupt Request Enable Clear Bit

- If "1" is written to this bit, the STC:OVIE bit is cleared to "0".
- The read value of this bit is always "0".

Bit	Description
0	Invalid
1	Clear the OVIE bit.

[bit3:1] Reserved: Reserved Bits

The read value is "0".

If writing to this bit, write "0".

[bit0] OVIRC: Overflow Interrupt Request Clear Bit

- If "1" is written to this bit, the STC:OVIR bit is cleared to "0".
- The read value of this bit is always "0".

Bit	Description
0	Invalid
1	Clear the OVIR bit.

Status Control Set Register (STCS)

bit	7	6	5	4	3	2	1	0
Field	Reserved	EDIES	Reserved	OVIES	Reserved			
R/W Attribute	R0,W0	R0,W	R0,W0	R0,W	R0,W0			
Protection Attribute	-	-	-	-	-			
Initial Value	0	0	0	0	0000			

Note: Reserved bits are located in the upper byte of this register.

[bit7] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

[bit6] EDIES: Measurement End Interrupt Request Enable Set Bit

- If "1" is written to this bit, the STC:EDIE bit is set to "1".
- The read value of this bit is always "0".

Bit	Description
0	Invalid
1	Set the EDIE bit.

[bit5] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

[bit4] OVIES: Overflow Interrupt Request Enable Set Bit

- If "1" is written to this bit, the STC:OVIE bit is set to "1".
- The read value of this bit is always "0".

Bit	Description
0	Invalid
1	Set the OVIE bit.

[bit3:0] Reserved: Reserved Bits

The read value is "0".

If writing to this bit, write "0".

9.4.3. Data Buffer Register (DTBF)

The data buffer register (DTBF) is a register that can read the measurement value or count value of the PWC timer. In 32-bit mode, the value of the lower 16 bits is read for the even-numbered channel, and the value of the upper 16 bits for the odd-numbered channel.

Be sure to read this register with 16- or 32-bit data transfer instructions.

bit	15	0
Field	DTBF[15:0]	
R/W Attribute	R,WX	
Protection Attribute	-	
Initial Value	0x0000	

- These bits compose the DTBF register as a read-only register in either continuous measurement mode or single measurement mode. Writing does not change the register value.
- In continuous measurement mode (TMCR bit2 MDSE = 0), the register is used as a buffer register to store the last measurement results.
- In single measurement mode (TMCR bit2 MDSE = 1), the up counter is directly accessed with the DTBF register. Reading is allowed even during counting so that the count value can be read. After measurement ends, the measurement results are stored as is.
- Use 16- or 32-bit data access for the DTBF register.

CHAPTER 41: Base Timer I/O Selection Function

This chapter explains the base timer I/O selection function.

1. **Overview**
2. **Configuration**
3. **Explanation of Operation**
4. **Registers**

BTSEL-TXXPT03P01R01L05-E1-XX

1. Overview

This section provides an overview of the base timer I/O selection function.

The base timer I/O selection function is a function for selecting a signal I/O method for the base timer by setting an I/O mode.

By switching the timer function, the base timer mounted in a channel can be used as any of the timers described below for each channel. The I/O method for the respective functions can be selected.

Timer outputs can be simultaneously read using TOUT Read Register (BT_BTTRR0, BT_BTTRR12, BT_BTTRR24).

The following 7 patterns can be selected for I/O pin connections.

- 16-bit timer standard mode
- 32-bit timer full-function mode
- PPG trigger 2-channel sharing mode
- Timer start/stop mode
- Simultaneous soft start mode
- Timer start/stop and simultaneous soft start mode
- Timer start mode

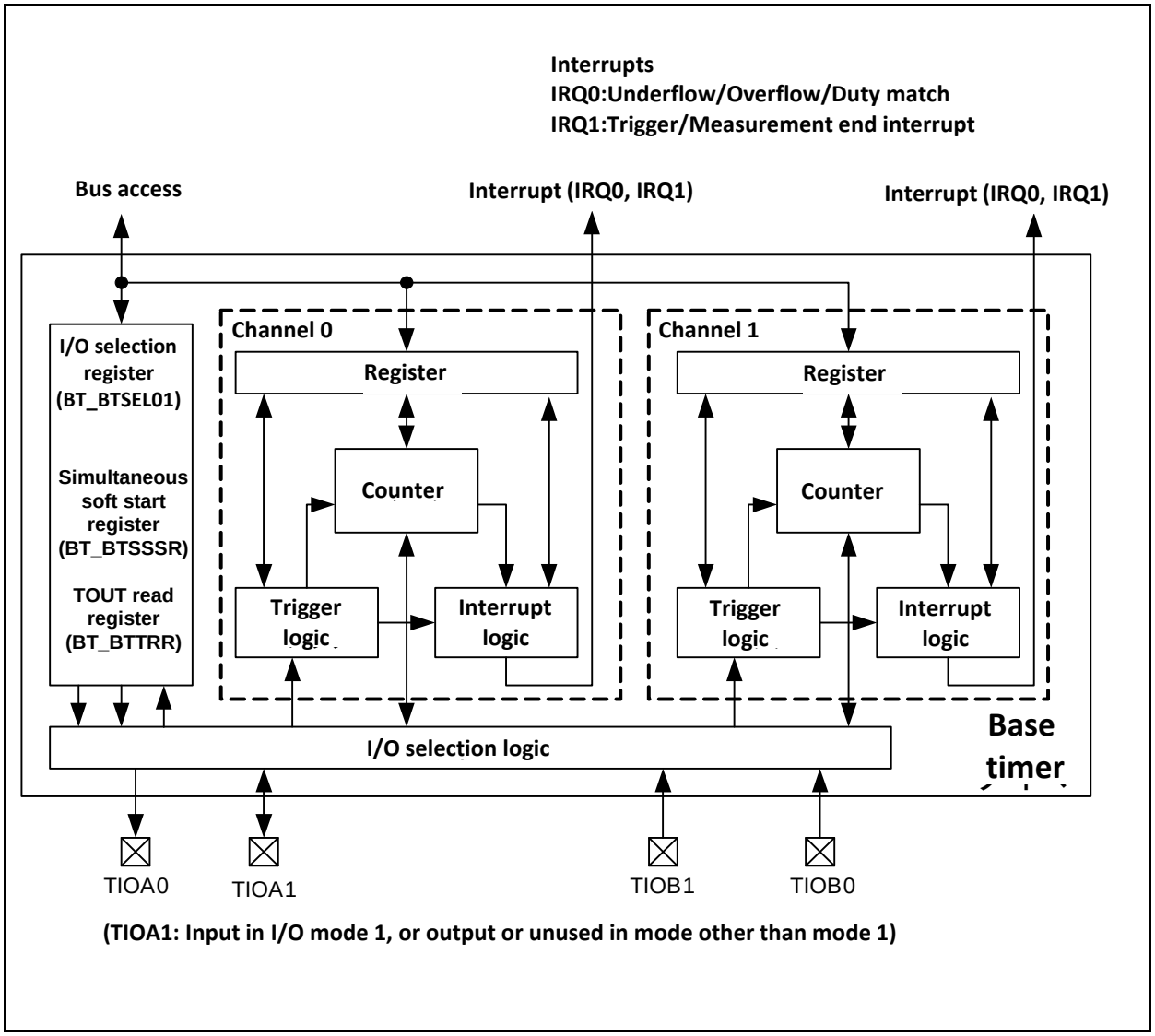
The 32-bit reload timer and 32-bit PWC timer can be implemented by using 2 channels of the mounted base timer. These 2 channels are channel m (m is an even number) and channel n (n = m + 1).

2. Configuration

The 32-bit reload timer and 32-bit PWC timer can be implemented by using 2 channels of the mounted base timer. These 2 channels are channel m (m is an even number) and channel n (n = m + 1).

This section explains the configuration of the base timer and I/O selection function.

Figure 2-1 Block Diagram



Configuration for channel 0 and channel 1

3. Explanation of Operation

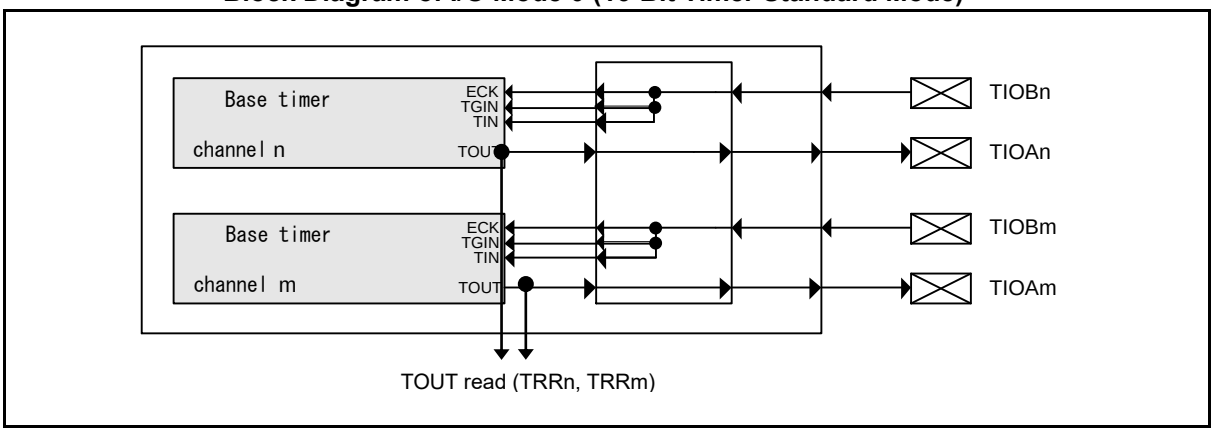
This section explains base timer I/O assignment.

Before using a timer, make an I/O setting for the base timer by using the I/O mode selection bit (BTSEL01). You can select one of the following 7 modes.

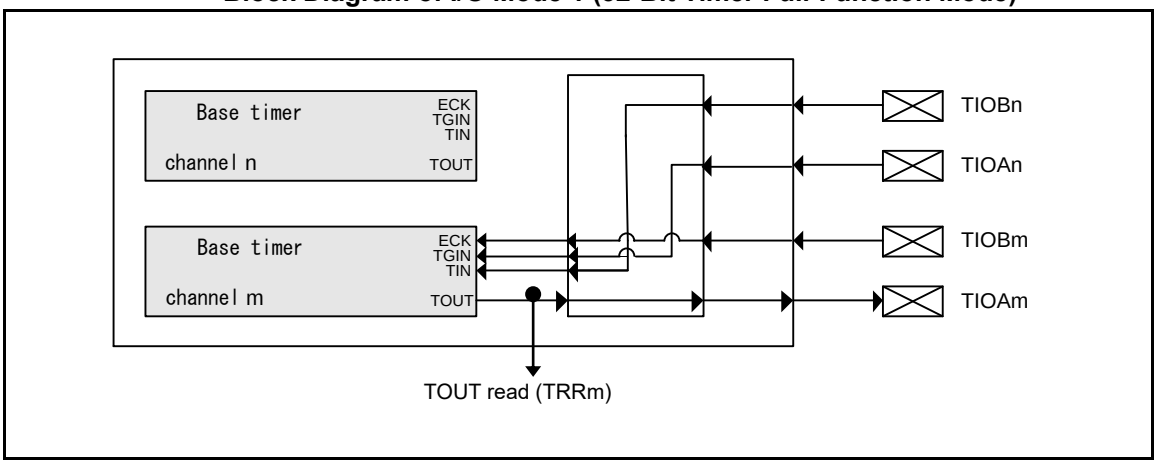
- I/O mode 0: 16-bit timer standard mode
  In this mode, the base timer in 1 channel operates individually and separately from the others.
- I/O mode 1: 32-bit timer full-function mode
  The signals of the even-numbered channels of the base timer are individually assigned to external pins in operation in this mode.
- I/O mode 2: PPG trigger 2-channel sharing mode
  This mode enables simultaneous input of external start triggers to the base timers of 2 channels. The base timers of the 2 channels can be started simultaneously.
- I/O mode 4: Timer start/stop mode
  In this mode, an even-numbered channel controls the start/stop of an odd-numbered channel. The odd-numbered channel is started by the rising edge* of an output signal from the even-numbered channel, and stopped by the falling edge*.
- I/O mode 5: Simultaneous soft start mode
  In this mode, software starts multiple channels simultaneously.
- I/O mode 6: Soft start timer start/stop mode
  In this mode, an even-numbered channel controls the start/stop of an odd-numbered channel. Software starts the even-numbered channel. The odd-numbered channel is started by the rising edge* of an output signal from the even-numbered channel, and stopped by the falling edge*.
- I/O mode 7: Timer start mode
  In this mode, an even-numbered channel controls the start of an odd-numbered channel. The odd-numbered channel is started by the rising edge* of an output signal from the even-numbered channel.

*: Use the trigger input selection bit (BTxx_TMCR:EGS) for the setting.

Block Diagram of I/O Mode 0 (16-Bit Timer Standard Mode)



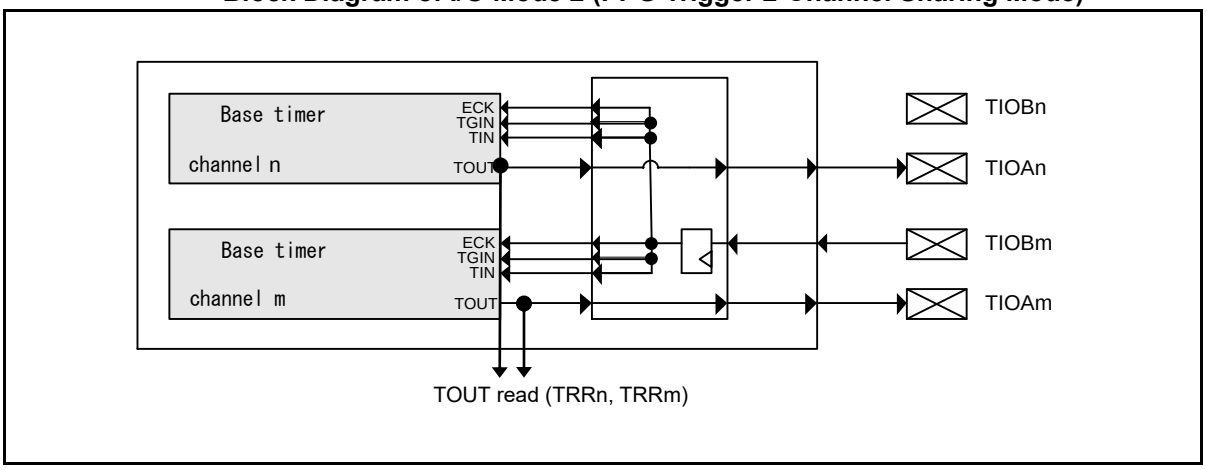
Block Diagram of I/O Mode 1 (32-Bit Timer Full-Function Mode)



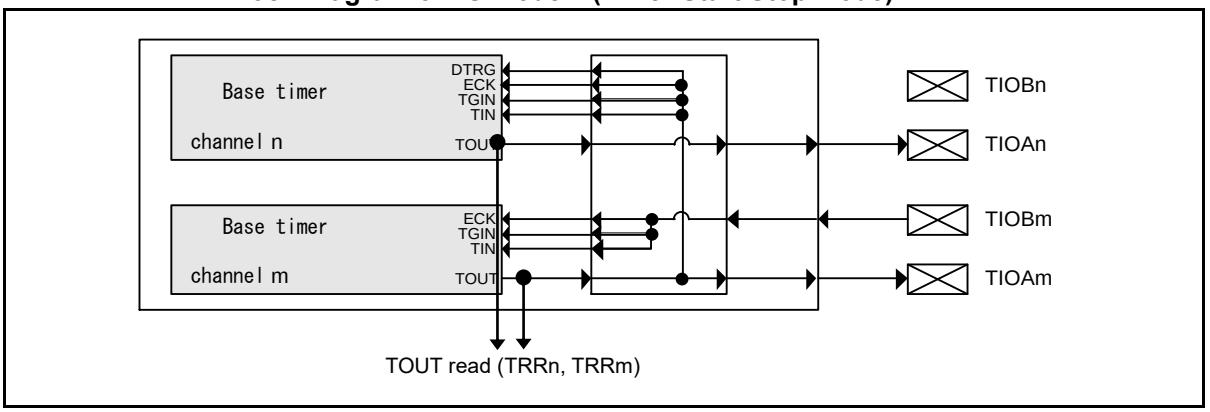
Note:

- *If I/O mode 1 is set, set read value of TRRn is undefined.*

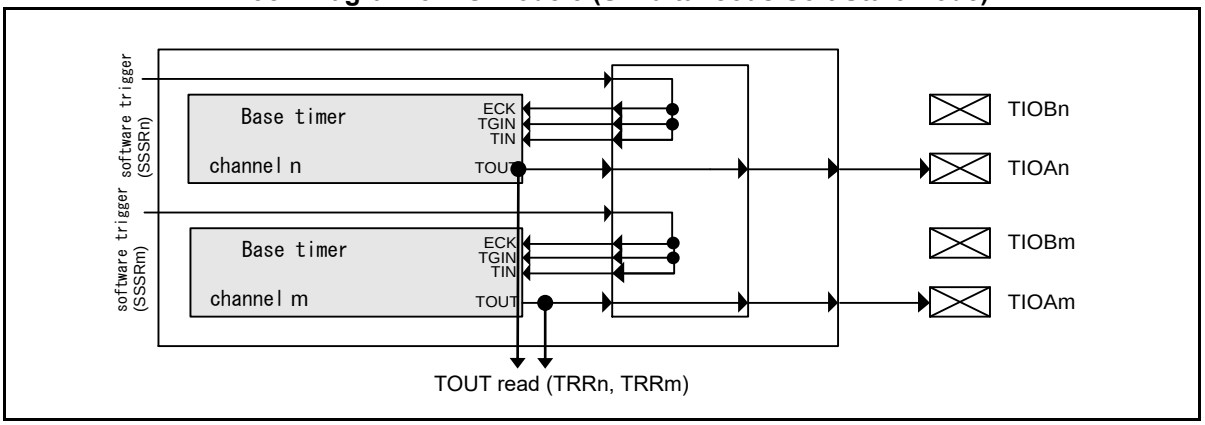
Block Diagram of I/O Mode 2 (PPG Trigger 2-Channel Sharing Mode)



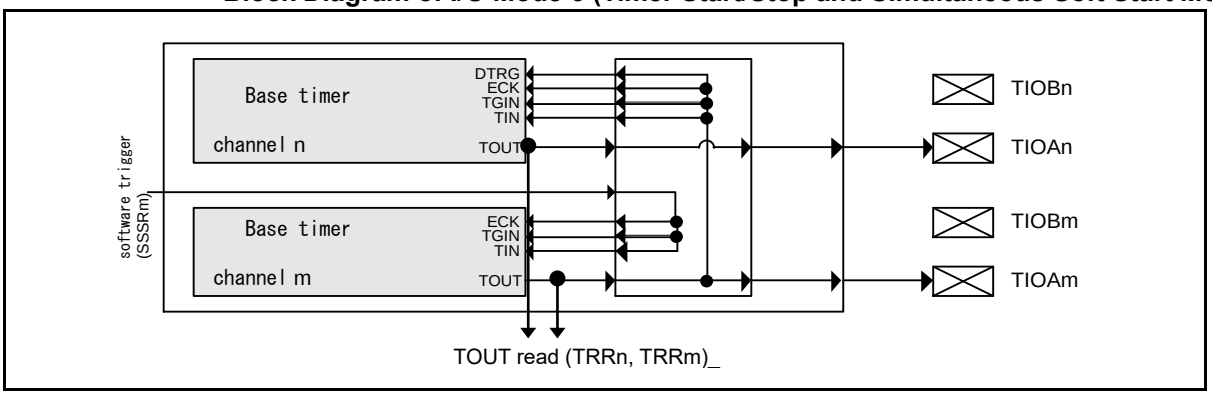
Block Diagram of I/O Mode 4 (Timer Start/Stop Mode)



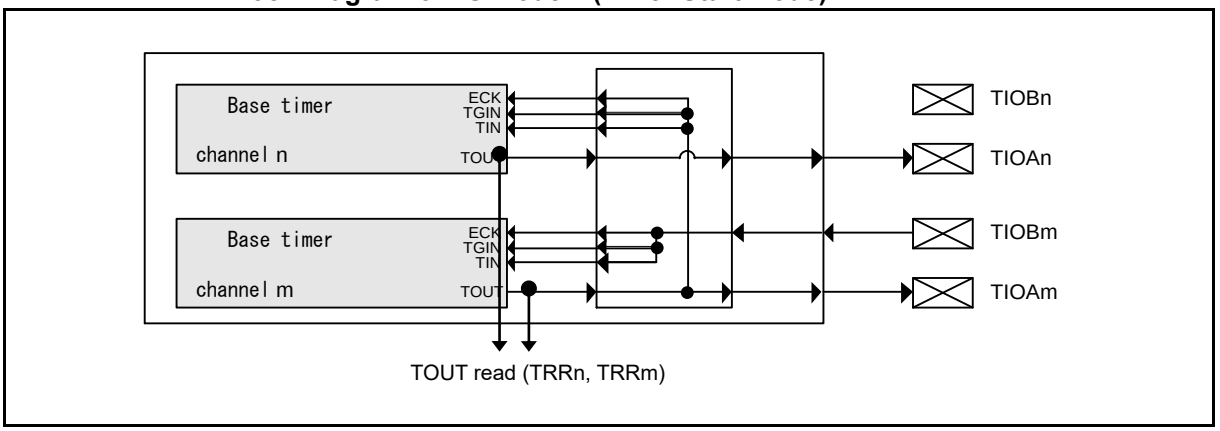
Block Diagram of I/O Mode 5 (Simultaneous Soft Start Mode)



Block Diagram of I/O Mode 6 (Timer Start/Stop and Simultaneous Soft Start Mode)



Block Diagram of I/O Mode 7 (Timer Start Mode)



Note:

- *If I/O mode 1 is set, set TIOAn of the corresponding odd-numbered channel to port input mode by using the GPIO setting.*

4. Registers

This section explains the base timer I/O selection function registers.

Table 4-1 List of Base Timer I/O Selection Registers

Abbreviated Register Name	Register Name	Reference
BT_BTSEL01	I/O selection register (channel 0, 1)	4.1
BT_BTSEL23	I/O selection register (channel 2, 3)	4.1
BT_BTSEL45	I/O selection register (channel 4, 5)	4.1
BT_BTSEL67	I/O selection register (channel 6, 7)	4.1
BT_BTSEL89	I/O selection register (channel 8, 9)	4.1
BT_BTSEL1011	I/O selection register (channel 10, 11)	4.1
BT_BTSEL1213	I/O selection register (channel 12, 13)	4.1
BT_BTSEL1415	I/O selection register (channel 14, 15)	4.1
BT_BTSEL1617	I/O selection register (channel 16, 17)	4.1
BT_BTSEL1819	I/O selection register (channel 18, 19)	4.1
BT_BTSEL2021	I/O selection register (channel 20, 21)	4.1
BT_BTSEL2223	I/O selection register (channel 22, 23)	4.1
BT_BTSEL2425	I/O selection register (channel 24, 25)	4.1
BT_BTSEL2627	I/O selection register (channel 26, 27)	4.1
BT_BTSEL2829	I/O selection register (channel 28, 29)	4.1
BT_BTSEL3031	I/O selection register (channel 30, 31)	4.1
BT_BTSEL3233	I/O selection register (channel 32, 33)	4.1
BT_BTSEL3435	I/O selection register (channel 34, 35)	4.1
BT_BTSSSR0	Simultaneous soft start register (channel 0)	4.2
BT_BTSSSR12	Simultaneous soft start register (channel 12)	4.2
BT_BTSSSR24	Simultaneous soft start register (channel 24)	4.2
BT_BTTRR0	TOUT read register (channel 0)	4.3
BT_BTTRR12	TOUT read register (channel 12)	4.3
BT_BTTRR24	TOUT read register (channel 24)	4.3

Tables 4.2, 4.3, and 4.4 show the register map of each mode.

The "***/***/***/***" expression in each table corresponds to the reload/PWM/PPG/PWC timer, respectively.

Table 4-2 Register Map (Channel No.: 0) (12) (24)

Offset	Register Name / Initial Value			
	+3	+2	+1	+0
0x0000_0000	Reserved 00000000_00000000		BTxx_PCSR/BTxx_PCSR/BTxx_PRLL /Reserved XXXXXXXX_XXXXXXXX	
0x0000_0004	Reserved 00000000_00000000		Reserved/BTxx_PDUT/ BTxx_PRLH/ BTxx_DTBF XXXXXXXX_XXXXXXXX /00000000_00000000 (BTxx_DTBF)	
0x0000_0008	Reserved 00000000_00000000		BTxx_TMR/ BTxx_TMR/ BTxx_TMR/Reserved 00000000_00000000 XXXXXXXX_XXXXXXXX*	
0x0000_000C	Reserved 00000000_00000000		BTxx_TMCR 00000000_00000000	
0x0000_0010	Reserved 00000000_00000000		BTxx_TMCR2 00000000	BTxx_STC 00000000
0x0000_0014	Reserved 00000000_00000000		Reserved 00000000	BTxx_STCC 00000000
0x0000_0018	Reserved 00000000_00000000		Reserved 00000000	BTxx_STCS 00000000
0x0000_001C	Reserved 00000000_00000000		Reserved/BTxx_PSDR/Reserved/Reserved 00000000_00000000	
0x0000_0020	Reserved 00000000_00000000		Reserved/BTxx_ADTR/Reserved/Reserved 00000000_00000000	
0x0000_0030	Reserved 11111111_11111111		Reserved 11111111	BT_BTSEL01/1213/ 2425 1111_0000
0x0000_0034	Reserved 11111111_11111111		BT_BTSSSR0/12/24 11111111_11111111	
0x0000_0038	Reserved 11111111_11111111		BT_BTTRR0/12/24 11110000_00000000	

*The initial value is XXXXXXXX_XXXXXXXX only during reload timer operation.

Table 4-3 Register Map (Channel No.: 1, 3, 5, 7, 9, and 11) (13, 15, 17, 19, 21, and 23) (25, 27, 29, 31, 33, and 35)

Offset	Register Name / Initial Value			
	+3	+2	+1	+0
0x0000_0000	Reserved 00000000_00000000		BTxx_PCSR/ BTxx_PCSR/ BTxx_PRLL /Reserved XXXXXXXX_XXXXXXXX	
0x0000_0004	Reserved 00000000_00000000		Reserved/ BTxx_PDUT/ BTxx_PRLH/ BTxx_DTBF XXXXXXXX_XXXXXXXX /00000000_00000000 (BTxx_DTBF)	
0x0000_0008	Reserved 00000000_00000000		BTxx_TMR/ BTxx_TMR/ BTxx_TMR/ Reserved 00000000_00000000 /XXXXXXXX_XXXXXXXX*	
0x0000_000C	Reserved 00000000_00000000		BTxx_TMCR 00000000_00000000	
0x0000_0010	Reserved 00000000_00000000		BTxx_TMCR2 00000000	BTxx_STC 00000000
0x0000_0014	Reserved 00000000_00000000		Reserved 00000000	BTxx_STCC 00000000
0x0000_0018	Reserved 00000000_00000000		Reserved 00000000	BTxx_STCS 00000000
0x0000_001C	Reserved 00000000_00000000		Reserved/BTxx_PSDR/Reserved/Reserved 00000000_00000000	
0x0000_0020	Reserved 00000000_00000000		Reserved/BTxx_ADTR/Reserved/Reserved 00000000_00000000	

*The initial value is XXXXXXXX_XXXXXXXX only during reload timer operation.

Table 4-4 Register Map (Channel No.: 2, 4, 6, 8, and 10) (14, 16, 18, 20, and 22) (26, 28, 30, 32, and 34)

Offset	Register Name / Initial Value			
	+3	+2	+1	+0
0x0000_0000	Reserved 00000000_00000000		BTxx_PCSR/ BTxx_PCSR/ BTxx_PRLL /Reserved XXXXXXXX_XXXXXXXX	
0x0000_0004	Reserved 00000000_00000000		Reserved/ BTxx_PDUT/ BTxx_PRLH / BTxx_DTBF XXXXXXXX_XXXXXXXX /00000000_00000000 (BTxx_DTBF)	
0x0000_0008	Reserved 00000000_00000000		BTxx_TMR/ BTxx_TMR/ BTxx_TMR /Reserved 00000000_00000000 / XXXXXXXX_XXXXXXXX*	
0x0000_000C	Reserved 00000000_00000000		BTxx_TMCR 00000000_00000000	
0x0000_0010	Reserved 00000000_00000000		BTxx_TMCR2 00000000	BTxx_STC 00000000
0x0000_0014	Reserved 00000000_00000000		Reserved 00000000	BTxx_STCC 00000000
0x0000_0018	Reserved 00000000_00000000		Reserved 00000000	BTxx_STCS 00000000
0x0000_001C	Reserved 00000000_00000000		Reserved/BTxx_PSDR/Reserved/Reserved 00000000_00000000	
0x0000_0020	Reserved 00000000_00000000		Reserved/BTxx_ADTR/Reserved/Reserved 00000000_00000000	
0x0000_0030	Reserved 11111111_11111111		Reserved 11111111	BT_BTSEL23/45/67/89/ 1011/1415/1617/1819/ 2021/2223/2627/2829/ 3031/3233/3435 1111_0000

*The initial value is XXXXXXXX_XXXXXXXX only during reload timer operation.

4.1. I/O Selection Registers (BT_BTSELmn)

This section shows the bit configuration of the I/O selection registers.

These bits set the I/O modes of the 2 channels of base timer channel m (m is 0 or an even number) and channel n (n = m +1: odd number) for the following connection.

Bit	31	8
Field	Reserved	
R/W Attribute	R1,WX	
Protection Attribute	-	
Initial Value	11111111_11111111_11111111	

Bit	7	6	5	4	3	2	1	0
Field	Reserved				BTSELmn			
R/W Attribute	R1,WX				R/W0	R/W		
Protection Attribute	-							
Initial Value	1111				0000			

[bit31:4] Reserved: Reserved Bits

[bit3:0] BTSELmn: I/O Mode Selection Bits

These bits set the I/O modes of the 2 channels of base timer channel m and channel n for the following connection.

Bits				Description
0	0	0	0	I/O mode 0: 16-bit timer standard mode
0	0	0	1	I/O mode 1: 32-bit timer full-function mode
0	0	1	0	I/O mode 2: PPG trigger 2-channel sharing mode
0	0	1	1	Setting prohibited
0	1	0	0	I/O mode 4: Timer start/stop mode
0	1	0	1	I/O mode 5: Simultaneous soft start mode
0	1	1	0	I/O mode 6: Timer start/stop and simultaneous soft start mode
0	1	1	1	I/O mode 7: Timer start mode
1	x	x	x	Setting prohibited

Note:

- *Do not change this register by setting reset mode (TMCR:FMD2 to 0="0b000"). After setting reset mode, rewrite this register.*

4.2. Simultaneous Soft Start Register (BT_BTSSSR0, BT_BTSSSR12, BT_BTSSSR24)

This section shows the bit configuration of the simultaneous soft start register.

These bits represent input signals in I/O modes 5 and 6. This register can be used to generate triggers for all channels simultaneously.

Bit	31	16
Field	Reserved	
R/W Attribute	R1,WX	
Protection Attribute	-	
Initial Value	11111111_11111111	

Bit	15	14	13	12	11	10	9	8
Field	Reserved				SSSR11	SSSR10	SSSR9	SSSR8
R/W Attribute	R1, WX				R1,W	R1,W	R1,W	R1,W
Protection Attribute	-							
Initial Value	1111				1	1	1	1

Bit	7	6	5	4	3	2	1	0
Field	SSSR7	SSSR6	SSSR5	SSSR4	SSSR3	SSSR2	SSSR1	SSSR0
R/W Attribute	R1,W	R1,W	R1,W	R1,W	R1,W	R1,W	R1,W	R1,W
Protection Attribute	-							
Initial Value	1	1	1	1	1	1	1	1

[bit31:12] Reserved: Reserved Bits

[bit11:0] SSSR[11:0]: Simultaneous Soft Start Bits

These bits represent input signals in I/O modes 5 and 6. For details on connections, see the block diagram of each I/O mode in "3. Explanation of Operation."

Writing "1" starts the corresponding channel, and writing "0" has no effect. Up to 12 channels with channel numbers 0 to 11 can be started simultaneously.

Correspondence of the channel number is below.

BT_BTSSSR0 : SSSR0-11 = ch0-11 (cperi0)

BT_BTSSSR12 : SSSR0-11 = ch12-23 (cperi1)

BT_BTSSSR24 : SSSR0-11 = ch24-35 (cperi2)

SSSR [X]	Description
0	No operation
1	Assigns the "1" pulse to input and starts the corresponding channel.

[X] represents the channel number of the base timer. It is a value from 0 to 11.

4.3. TOUT Read Register (BT_BTTRR0, BT_BTTRR12, BT_BTTRR24)

This section shows the bit configuration of the TOUT read register.

By using these bits, software can read TOUT status from the Base Timer channels.

Bit	31															16
Field	Reserved															
R/W Attribute	R1,WX															
Protection Attribute	-															
Initial Value	11111111_11111111															

Bit	15	14	13	12	11	10	9	8
Field	Reserved				TRR11	TRR10	TRR9	TRR8
R/W Attribute	R1, WX				R,WX	R,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	1111				0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	TRR7	TRR6	TRR5	TRR4	TRR3	TRR2	TRR1	TRR0
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

[bit31:12] Reserved: Reserved Bits

[bit11:0] TRR[11:0]: Simultaneous TOUT Read Bits

These bits represent TOUT status of the Base Timer channels. For details on connections, see the block diagram of each I/O mode in 3. Explanation of Operation.

TRR[X]	Description
0	TOUT status is '0'
1	TOUT status is '1'

[X] represents the channel number of the base timer. It is a value from 0 to 11.

CHAPTER 42: 32-bit Free-run Timer

This chapter describes the functions of the 32-bit free-run timer.

1. Overview of the 32-Bit Free-Run Timer
2. Configuration and Block Diagram
3. Explanation of the 32-Bit Free-Run Timer Operation
4. Registers of the 32-Bit Free-Run Timer
5. Precautions for Using This Device

FRT-TXXPT03P01R01L06-E1-XX

1. Overview of the 32-Bit Free-Run Timer

The 32-bit free-run timer supports the 32-bit up count mode or up/down count mode. This timer can be used with the 32-bit input capture and the 32-bit output compare. This timer can measure the input pulse width and external clock cycle.

Functions of the 32-Bit Free-Run Timer

- The 32-bit free-run timer is composed of the 32-bit up/down counter, control register, 32-bit compare clear register, 32-bit compare clear buffer register, and prescaler.
- The 9 types of counter operation clocks (φ, φ/2, φ/4, φ/8, φ/16, φ/32, φ/64, φ/128, and φ/256) (φ: peripheral clock) can be selected.
- A compare clear interrupt is generated when the values of the compare clear register and the 32-bit counter are compared and their values match. A 0 detection interrupt is generated when the 32-bit counter detects the count value 0x00000000.
- The compare clear register has a buffer register (the data written to this buffer register is transferred to the compare clear register). When the 32-bit counter is stopped, the data is transferred as soon as data is written to the buffer. When the 32-bit counter is operating, data is transferred from the buffer upon detecting the count value 0x00000000.
- The count value is reset to 0x00000000 when a hardware reset occurs, software clears the timer, or the value of the compare clear register and the count value match in up count mode.
- The output value of the 32-bit counter can be used as a clock count of the output compare and the input capture.

Interrupt request and DMA request

- Interrupt request of this module can be used for purposes below:
  1. Interrupt request to CPU (via Interrupt Controller)
  2. DMA request to DMA Controller
- To use the interrupt as interrupt request to CPU, following configuration is necessary:
  - Enable the interrupt request in this module
  - Enable the interrupt channel of the interrupt in Interrupt Controller
- To use the interrupt as DMA request to DMA Controller, following configuration is necessary:
  - Enable the interrupt request in this module
  - Select the DMA channel from this module as DMA client in DMA Controller

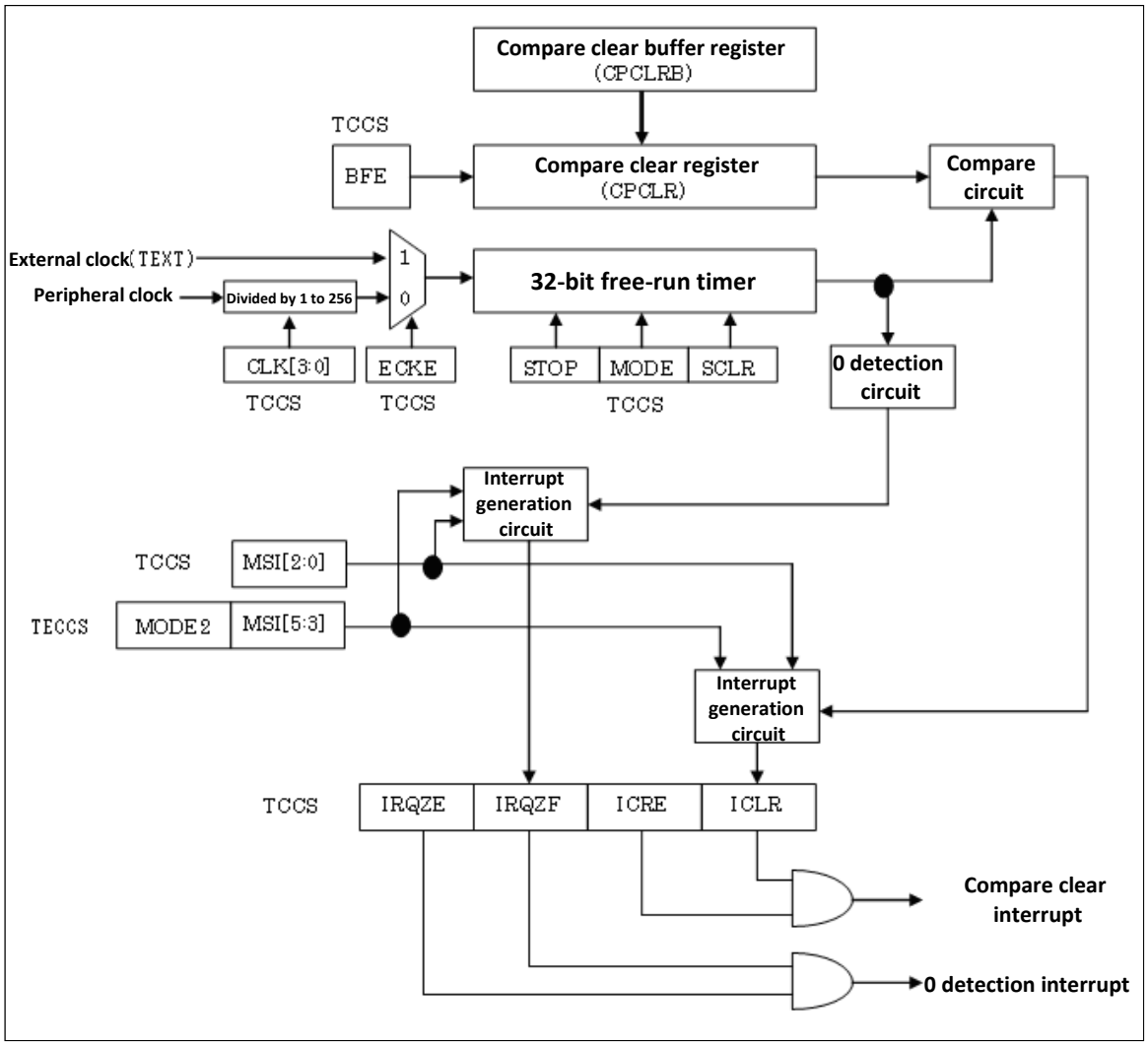
Note:

- *Interrupt request that supports for DMA request is product specification. Please refer to the product's HWM for detail.*

2. Configuration and Block Diagram

Figure 2-1 is a configuration diagram of the 32-bit free-run timer.

Figure 2-1 Configuration Diagram of 32-Bit Free-Run Timer



3. Explanation of the 32-Bit Free-Run Timer Operation

This section provides a summary of the operation of the 32-bit free-run timer.

Operation of the 32-Bit Free-Run Timer

The 32-bit free-run timer starts counting from the value set in the timer data register (TCDT) after the timer is set to enabled (TCCS:STOP). When the 32-bit output compare and the 32-bit input capture are connected to the free-run timer, then the timer count value is used as base time of the 32-bit output compare and the 32-bit input capture.

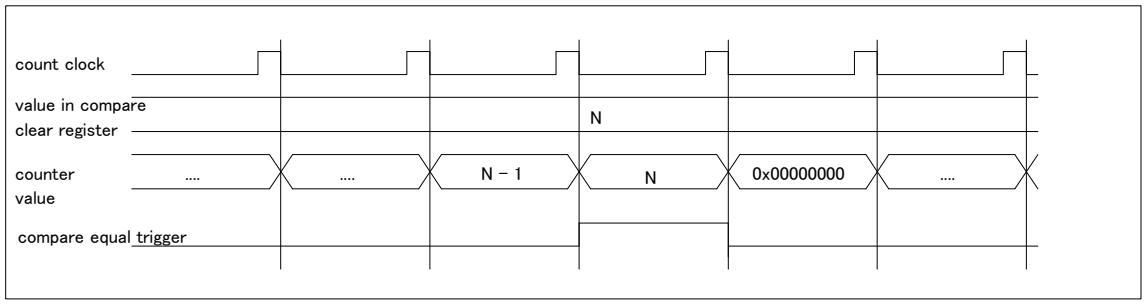
Counter Clear

The count value of the free-run timer is cleared to 0 when any of the following conditions is met.

- When the count value matches the value of the compare clear register (CPCLR) in the up count mode (MODE: bit5=0 of the timer state control register (TCCS))
- When "1" is written to the timer clear bit (SCLR: bit4) of the timer state control register (TCCS) while the free-run timer is operating (STOP: bit6=0 of the timer state control register (TCCS))
- When 0x00000000 is written to the timer data register (TCDT) while the free-run timer is stopped (STOP: bit6=1 of the timer state control register (TCCS))
- When the hardware is reset. Upon reset, the counter is cleared immediately.

When "1" is written to the timer clear bit (SCLR: bit4) of the timer state control register (TCCS) or when the count value matches the value of the compare clear register, the counter is cleared synchronously with the count timing.

Figure 3-1 Clear Timing of 32-Bit Free-Run Timer



Note:

- *The count value of the free-run timer is not cleared if "1" is written to the timer clear bit (SCLR: bit4) of the timer state control register (TCCS) while the timer is stopped.*

Timer Mode

Either of the following modes can be selected for the free-run timer.

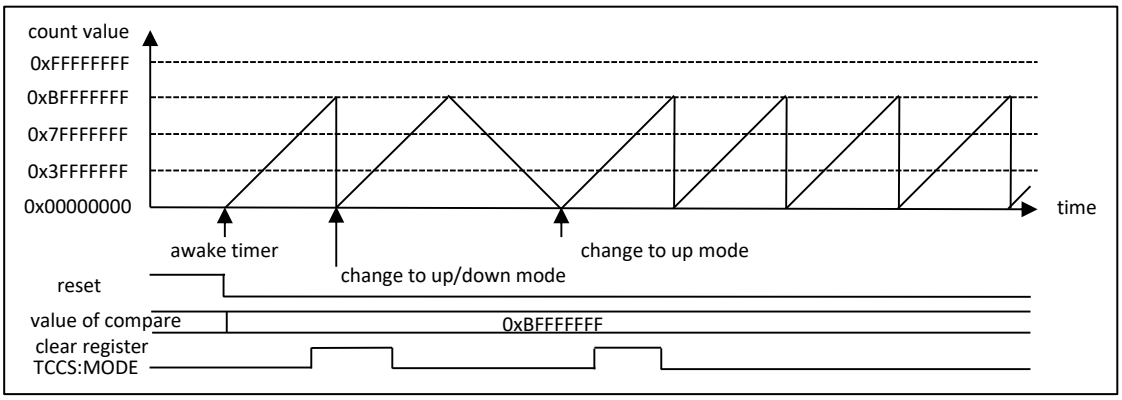
- Up count mode (MODE: bit5=0 of the timer state control register (TCCS))
- Up/down count mode (MODE: bit5=1 of the timer state control register (TCCS))

In the up count mode, the counter starts counting from the timer data register (TCDT) that is set in advance. The counter continues to count up until the count value matches the value of the compare clear register (CPCLR). Then, the counter is cleared to 0x00000000 and starts counting up again.

In the up/down count mode, the counter starts counting from the timer data register (TCDT) that is set in advance. The counter continues to count up until the count value matches the value of the compare clear register (CPCLR). Then, the counter changes the counting direction from up count to down count. It continues to count down until the counter value reaches 0x00000000 and starts counting up again.

A value can be written to the timer count mode bit (MODE: bit5) of the timer state control register (TCCS) at any time, even when the timer is operating or stopped. The value written to this bit during the timer operation is stored in a buffer. The count mode changes when the count value becomes 0x00000000.

Figure 3-2 Change of Timer Mode (While Timer Is Operating)



Compare Clear Buffer

The compare clear register (CPCLR) has a buffer function that can be enabled or disabled. When the buffer function is enabled (BFE: bit7=1 of the timer state control register (TCCS)), the data written to the compare clear buffer register (CPCLRB) is transferred to the CPCLR register upon detecting the count value 0. When the buffer function is disabled (BFE: bit7=0 of the timer state control register (TCCS)), the data can be directly written to the compare clear register (CPCLR).

Figure 3-3 Up Count Mode Operation When Compare Clear Buffer Is Disabled (BFE: bit7=0 of Timer State Control Register (TCCS))

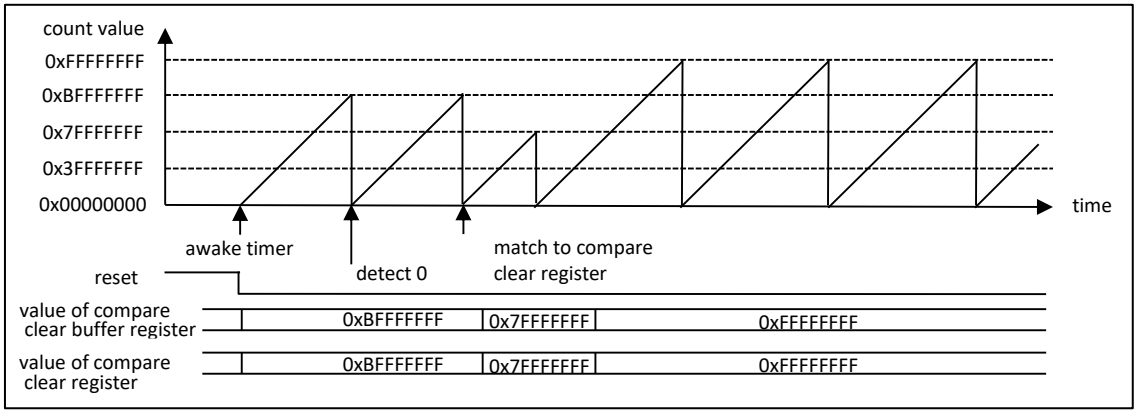


Figure 3-4 Up Count Mode Operation When Compare Clear Buffer Is Enabled (BFE: bit7=1 of Timer State Control Register (TCCS))

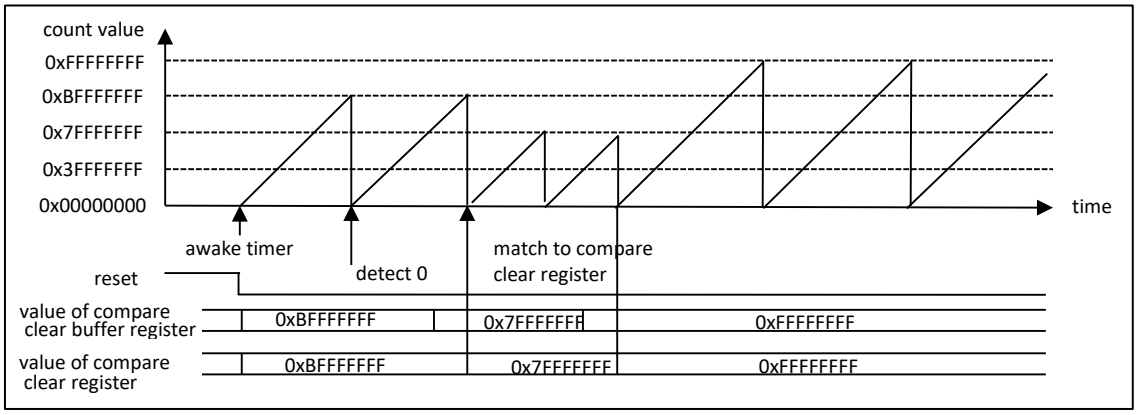
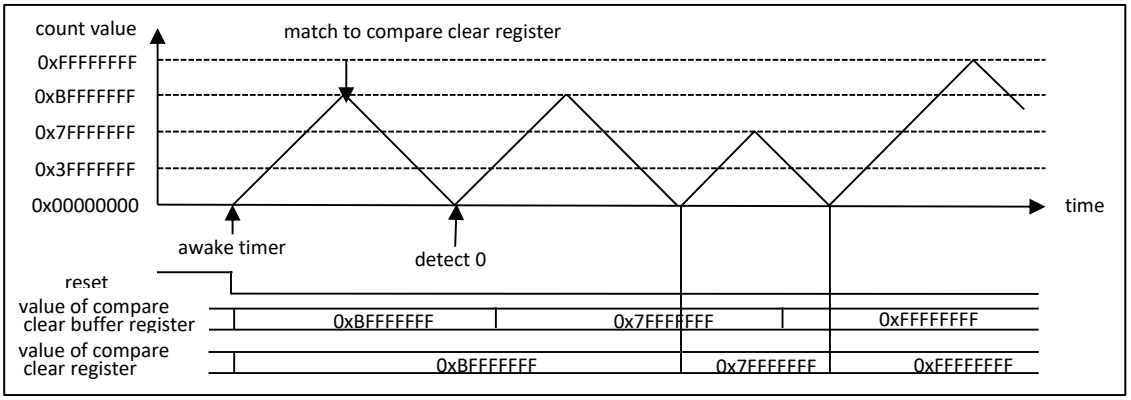


Figure 3-5 Up/Down Count Mode Operation When Compare Clear Buffer Is Enabled (BFE: bit7=1 of Timer State Control Register (TCCS))



Timer Interrupt

The 32-bit free-run timer can generate the following 2 interrupts.

- Compare clear interrupt
- 0 detection interrupt

The compare clear interrupt is generated when the count value matches the value of the compare clear register. The 0 detection interrupt is generated when the count value reaches 0x00000000H.

Note:

- *The 0 detection interrupt is not generated when the timer is cleared (SCLR: bit4=1 of the timer state control register (TCCS)).*

Figure 3-6 Interrupt Generated in Up Count Mode (MODE: bit5=0 of the Timer State Control Register (TCCS))

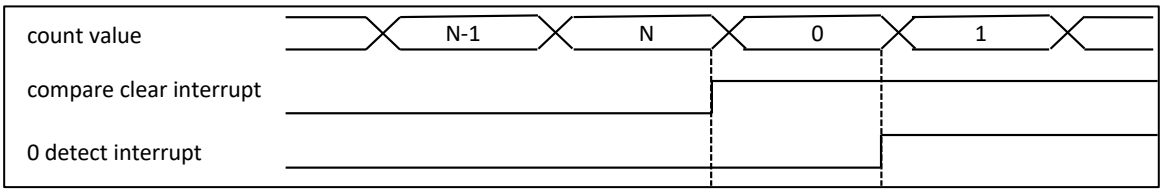
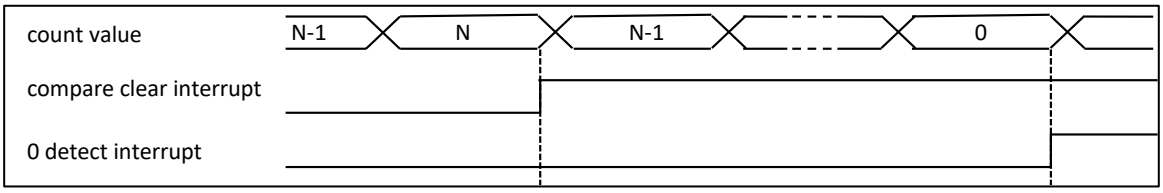


Figure 3-7 Interrupt Generated in Up/Down Count Mode (MODE: bit5=1 of the Timer State Control Register (TCCS))



Interrupt Mask Function

Either the 0 detection interrupt or the compare clear interrupt, or both interrupts can be masked.

The following describes the case when either interrupt is masked.

- An interrupt request flag can be masked by setting the interrupt mask selection bits (MSI2 to MSI0: bit12 to bit10) of the timer state control register (TCCS). The interrupt mask selection bits (MSI2 to MSI0: bit12 to bit10) are the 3-bit reload down register that reloads the value when the mask count value reaches 0x000.
  The mask count value can also be loaded by directly writing data to the interrupt mask selection bits (MSI2 to MSI0: bit12 to bit10). The number of mask counts is the value set in the interrupt mask selection bits (MSI2 to MSI0: bit12 to bit10). An interrupt request flag is not masked when the mask count value (MSI2 to MSI0: bit12 to bit10) becomes 0x000.
- The mask control of an interrupt request varies depending on the count mode (MODE: bit5 of the timer state control register (TCCS)). In the up count mode (MODE: bit5=0), only the compare clear interrupt request flag can be masked and the 0 detection interrupt is generated every time a timer counter value of 0 is detected. In the up/down count mode (MODE: bit5=1), only the 0 detection interrupt request flag can be masked.

The following describes the case when both interrupt requests are masked.

- Both interrupts can be masked only when the free-run timer is in the up/down count mode (MODE: bit5=1) and when the timer extended state control register (TECCS) is set to MODE2: bit11=1 and the timer state control register (TCCS) is set to MODE: bit5=1.
- MSI2 to MSI0 bits of the timer state control register (TCCS) are used to mask the 0 detection interrupt. MSI5 to MSI3 bits of the timer extended state control register (TECCS) are used to mask the compare clear interrupt.

Notes:

- *The 0 detection interrupt is not generated when the timer is cleared (SCLR: bit4=1 of the timer state control register (TCCS)).*
- *If Interrupt mask function "by setting TCCS.MSI[2:0] Register" is used, please write TCCS[15:8], TCCSC[15:8] and TCCSS[15:8] after 32bit free-run timer stopped.*

Figure 3-8 Compare Clear Interrupts to Be Masked in Up Count Mode

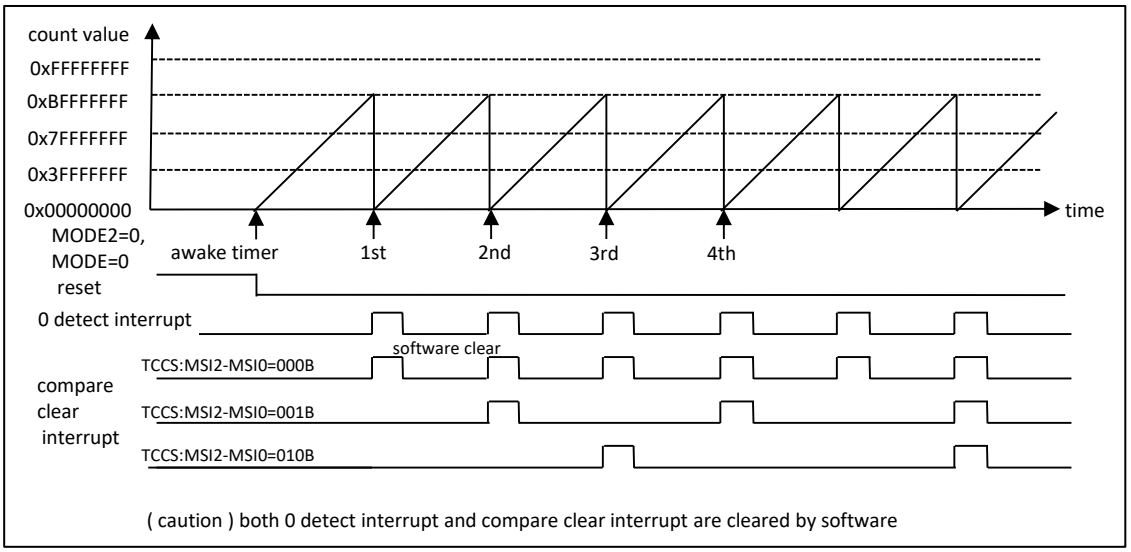


Figure 3-9 0 Detection Interrupts to Be Masked in Up/Down Count Mode

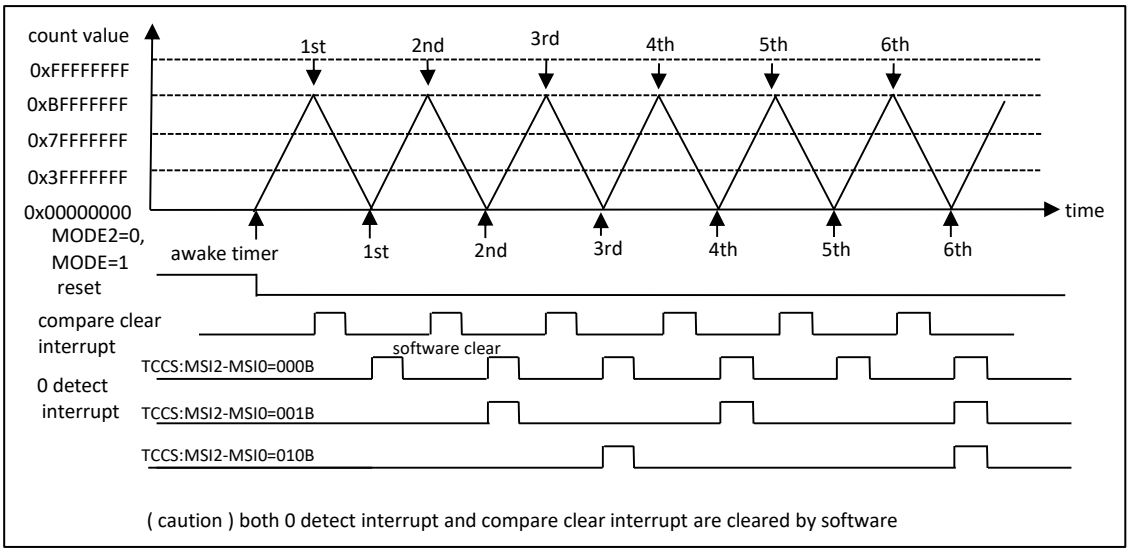
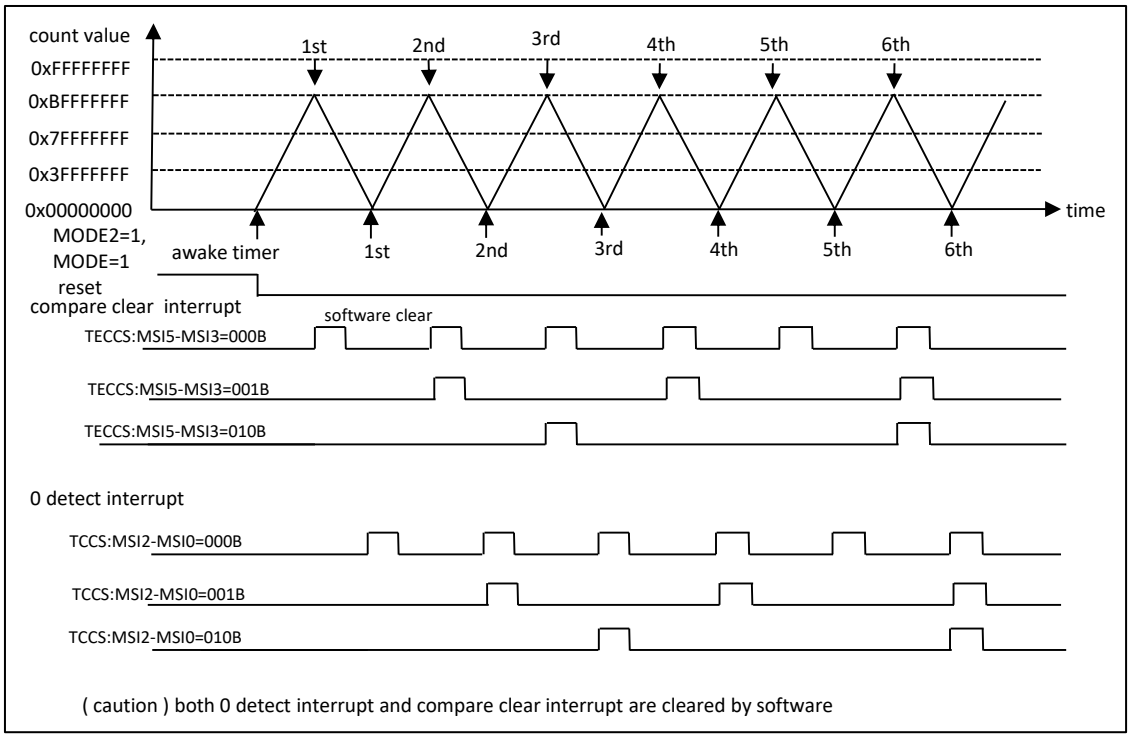


Figure 3-10 0 Detection Interrupts and Compare Clear Interrupts to Be Masked in Up/Down Count Mode

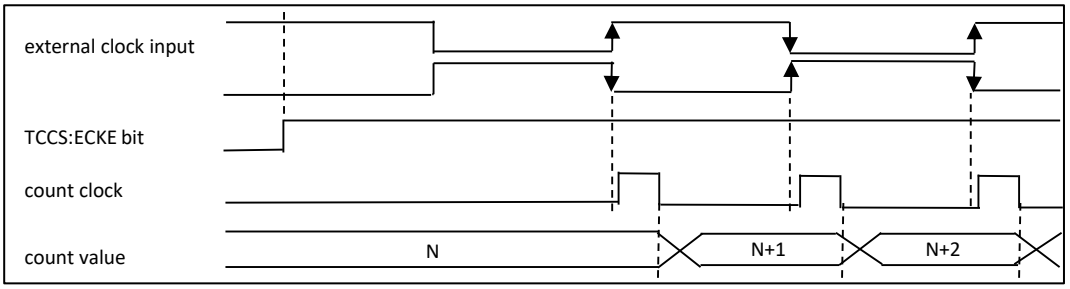


Selected External Count Clock

The free-run timer counts based on an input clock (peripheral clock or external clock).

If the external clock mode (ECKE: bit 15=1 of the timer state control register (TCCS)) is selected, the free-run timer starts counting at a rising edge of the external input when the initial value of the external clock input (TEXT) is "H". After that, the free-run timer counts at both edges. When the initial value of the external input is "L", the free-run timer starts counting at a falling edge of the external input. After that, the free-run timer counts up at both edges.

Figure 3-11 Count Timing of Free-Run Timer (in Up Count Mode)



Note:

- *When the external clock input is used, the timer counts at both edges of the external clock.*

3.1. Interrupts of the 32-Bit Free-Run Timer

There are the 2 types of interrupts as the interrupts of the 32-bit free-run timer: compare interrupt and 0 detection interrupt.

Free-Run Timer Interrupt

Table 2-1 shows the interrupt control bits and the interrupt factor of the free-run timer.

Table 3-1 Interrupt Control Bits and Interrupt Factor of Free-Run Timer

	Free-run Timer	
	Compare Clear	**0 Detection**
Interrupt request flag bit	Compare clear interrupt flag bit (ICLR:bit9) of the timer state control register (TCCS)	0 detection interrupt flag bit (IRQZF:bit14) of the timer state control register (TCCS)
Interrupt request enable bit	Compare clear interrupt request enable bit (ICRE:bit8) of the timer state control register (TCCS)	0 detection interrupt request enable bit (IRQZE:bit13) of the timer state control register (TCCS)
Interrupt Factor	The value of the free-run timer matches the value of the compare clear register (CPCLR).	The free-run timer value becomes "0".

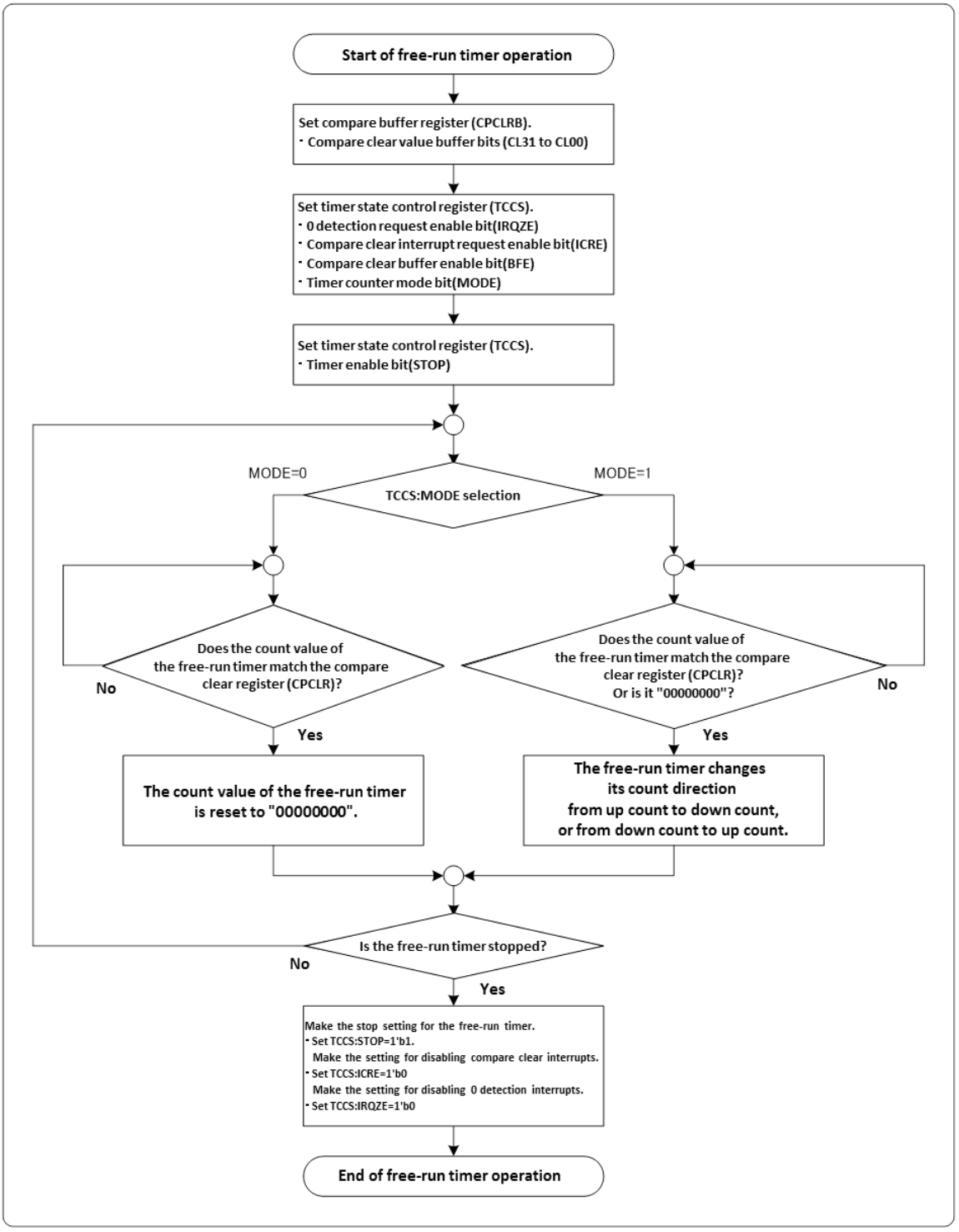
"1" is set to the compare clear interrupt flag (ICLR: bit9) of the timer state control register (TCCS) when the value of the free-run timer matches the value of the compare clear register (CPCLR). When the interrupt request is set to enabled (ICRE: bit8=1 of the timer state control register (TCCS)) in this state, then the interrupt request signal (MIRQ) becomes "H".

"1" is set to the 0 detection interrupt flag (IRQZF: bit14) of the timer state control register (TCCS) when the value of the free-run timer becomes 0x00000000. When the interrupt request is set to enabled (IRQZE: bit13=1 of the timer state control register (TCCS)) in this state, then the interrupt request signal (ZIRQ) becomes "H".

3.2. Setting Procedure Example of the 32-Bit Free-Run Timer

This section provides a setting procedure example of the free-run timer.

Figure 3-12 Setting Procedure Example of Free-Run Timer Operation



4. Registers of the 32-Bit Free-Run Timer

This section provides the register list of the 32-bit Free-run Timer.

Table 4-1 Register Map

Common Peripheral #0 Group (Channel No.:0, 1, 2, 3, 4, 5, 6, and 7)

Offset	Address Offset Value / Register Name				Block Name
	+3	+2	+1	+0	
0000_0000	FRTxx_CPCLRB/FRTxx_CPCLR 11111111_11111111_11111111_11111111				Common Peripheral #0
0000_0004	FRTxx_TCDT 00000000_00000000_00000000_00000000				
0000_0008	FRTxx_TCCS 11111111_11111111_00000000_01000000				
0000_000C	FRTxx_TECCS 11111111_11111111_00000000_11111111				
0000_0010	FRTxx_TCCSC 00000000_00000000_00000000_00000000				
0000_0014	FRTxx_TCCSS 00000000_00000000_00000000_00000000				
0000_0018 | 0000_03FC	-				

Note:

- *xx is the channel number (0 to 7).*

Common Peripheral #1 Group (Channel No.:8, 9, 10, 11, 12, 13, 14, and 15)

Offset	Address Offset Value / Register Name				Block Name
	+3	+2	+1	+0	
0000_0000	FRTxx_CPCLRB/FRTxx_CPCLR 11111111_11111111_11111111_11111111				Common Peripheral #1
0000_0004	FRTxx_TCDT 00000000_00000000_00000000_00000000				
0000_0008	FRTxx_TCCS 11111111_11111111_00000000_01000000				
0000_000C	FRTxx_TECCS 11111111_11111111_00000000_11111111				
0000_0010	FRTxx_TCCSC 00000000_00000000_00000000_00000000				
0000_0014	FRTxx_TCCSS 00000000_00000000_00000000_00000000				
0000_0018 | 0000_03FC	-				

Note:

- *xx is the channel number (8 to 15).*

Registers of the 32-Bit Free-Run Timer

Table 4-2 Register List of 32-Bit Free-Run Timer (Case to Use the APB I/F)

Abbreviated Register Name	Register Name	Reference
CPCLRB/CPCLR	Compare clear buffer register, compare clear register	4.1
TCDT	Timer data register	4.2
TCCS	Timer state control register	4.3
TECCS	Timer extended state control register	4.4
TCCSC	Timer state control clear register	4.5
TCCSS	Timer state control set register	4.6

Register Bit Locations of the 32-Bit Free-Run Timer

Table 4-3 Register Bit Locations of the 32-Bit Free-Run Timer

	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
CPCLRB/CPCLR	CL31	CL30	CL29	CL28	CL27	CL26	CL25	CL24
	CL23	CL22	CL21	CL20	CL19	CL18	CL17	CL16
	CL15	CL14	CL13	CL12	CL11	CL10	CL09	CL08
	CL07	CL06	CL05	CL04	CL03	CL02	CL01	CL00
TCDT	T31	T30	T29	T28	T27	T26	T25	T24
	T23	T22	T21	T20	T19	T18	T17	T16
	T15	T14	T13	T12	T11	T10	T09	T08
	T07	T06	T05	T04	T03	T02	T01	T00
TCCS	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
	ECKE	IRQZF	IRQZE	MSI2	MSI1	MSI0	ICLR	ICRE
	BFE	STOP	MODE	SCLR	CLK3	CLK2	CLK1	CLK0
TECCS	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
	Reserved	Reserved	Reserved	Reserved	MODE2	MSI5	MSI4	MSI3
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
TCCSC	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
	ECKEC	IRQZFC	IRQZEC	Reserved	Reserved	Reserved	ICLRC	ICREC
	BFEC	STOPC	MODEC	Reserved	Reserved	Reserved	Reserved	Reserved
TCCSS	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
	ECKES	Reserved	IRQZES	Reserved	Reserved	Reserved	Reserved	ICRES
	BFES	STOPS	MODES	SCLRS	Reserved	Reserved	Reserved	Reserved

4.1. Compare Clear Buffer Register (CPCLRB)/ Compare Clear Register (CPCLR)

The compare clear buffer register (CPCLRB) is the buffer register of the compare clear register (CPCLR). Both registers are located in the same address.

Compare Clear Buffer Register (CPCLRB)

bit	31	30	29	28	27	26	25	24
Field	CL31	CL30	CL29	CL28	CL27	CL26	CL25	CL24
R/W Attribute	W	W	W	W	W	W	W	W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	1	1	1	1	1	1	1	1

bit	23	22	21	20	19	18	17	16
Field	CL23	CL22	CL21	CL20	CL19	CL18	CL17	CL16
R/W Attribute	W	W	W	W	W	W	W	W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	1	1	1	1	1	1	1	1

bit	15	14	13	12	11	10	9	8
Field	CL15	CL14	CL13	CL12	CL11	CL10	CL09	CL08
R/W Attribute	W	W	W	W	W	W	W	W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	1	1	1	1	1	1	1	1

bit	7	6	5	4	3	2	1	0
Field	CL07	CL06	CL05	CL04	CL03	CL02	CL01	CL00
R/W Attribute	W	W	W	W	W	W	W	W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	1	1	1	1	1	1	1	1

[bit31:0] CL31 to CL00: Compare Clear Value Buffer Bits

The compare clear buffer register (CPCLRB) is a buffer register that is located in the same address as the compare clear register (CPCLR).

The value of the compare clear buffer register (CPCLRB) is immediately transferred to the compare clear register (CPCLR) when the buffer function is set to disabled (BFE: bit7=0 of the timer state control register (TCCS)) or the free-run timer is stopped.

The value of the compare clear buffer register (CPCLRB) is transferred to the compare clear register (CPCLR) when the buffer function is set to enabled (BFE: bit7=1 of the timer state control register (TCCS)) and 0 is detected as the count value of the free-run timer.

Notes:

- *Do not set 0x00000000 in the compare clear buffer register (CPCLRB).*
- *For access to this register, use word access instructions.*

Compare Clear Register (CPCLR)

bit	31	30	29	28	27	26	25	24
Field	CL31	CL30	CL29	CL28	CL27	CL26	CL25	CL24
R/W Attribute	R	R	R	R	R	R	R	R
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	1	1	1	1	1	1	1	1

bit	23	22	21	20	19	18	17	16
Field	CL23	CL22	CL21	CL20	CL19	CL18	CL17	CL16
R/W Attribute	R	R	R	R	R	R	R	R
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	1	1	1	1	1	1	1	1

bit	15	14	13	12	11	10	9	8
Field	CL15	CL14	CL13	CL12	CL11	CL10	CL09	CL08
R/W Attribute	R	R	R	R	R	R	R	R
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	1	1	1	1	1	1	1	1

bit	7	6	5	4	3	2	1	0
Field	CL07	CL06	CL05	CL04	CL03	CL02	CL01	CL00
R/W Attribute	R	R	R	R	R	R	R	R
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	1	1	1	1	1	1	1	1

[bit31:0] CL31 to CL00: Compare Clear Value Bits

The compare clear register (CPCLR) is used to compare the count value of the free-run timer.

In the up count mode (MODE: bit5=0 of the timer state control register (TCCS)), the count value of the free-run timer is reset to 0x00000000 when the value of this register matches the count value of the free-run timer.

In the up/down count mode (MODE: bit5=1 of the timer state control register (TCCS)), the free-run timer changes the counting direction from up count to down count when the value of the compare clear register (CPCLR) matches the count value of the free-run timer. The counting direction is changed from down count to up count when 0 is detected.

Note:

- *For access to this register, use word access instructions.*

4.2. Timer Data Register (TCDT)

The timer data register (TCDT) reads the count value of the free-run timer. This register can also be used to set the count value of the free-run timer.

bit	31	30	29	28	27	26	25	24
Field	T31	T30	T29	T28	T27	T26	T25	T24
R/W Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	T23	T22	T21	T20	T19	T18	T17	T16
R/W Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	T15	T14	T13	T12	T11	T10	T09	T08
R/W Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	T07	T06	T05	T04	T03	T02	T01	T00
R/W Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

[bit31:0] T31 to T00: Timer Data Value Bits

The timer data register (TCDT) is used to read the count value of the free-run timer.

The count value can be set by writing a value to this register. However, a value needs to be written while the free-run timer is stopped (STOP: bit 6=1 of the timer state control register (TCCS)).

The counter is cleared to the count value 0x00000000 as soon as any of the following factors occurs.

- Hardware reset
- The timer clear bit (SCLR: bit4) of the timer state control register (TCCS) is "1" while the free-run timer is operating (STOP: bit6=0 of the timer state control register (TCCS)).
- The value of the compare clear register (CPCLR) matches the timer count value in the up count mode (MODE: bit5=0 of the timer state control register (TCCS)).

Notes:

- *The free-run timer is not cleared to 0x00000000 even if the timer clear bit (SCLR: bit4) of the timer state control register (TCCS) is set to "1" while the free-run timer is stopped (STOP: bit6=1 of the timer state control register (TCCS)).*
- *For access to this register, use word access instructions.*

4.3. Timer State Control Register (TCCS)

The timer state control register (TCCS) is a register that is used to control the operation of the free-run timer.

For details on writing to this register, see "5.Precautions for Using This Device."

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R1,WX							
Protection Attribute	-							
Initial Value	11111111							

bit	23	22	21	20	19	18	17	16
Field	Reserved							
R/W Attribute	R1,WX							
Protection Attribute	-							
Initial Value	11111111							

bit	15	14	13	12	11	10	9	8
Field	ECKE	IRQZF	IRQZE	MSI2	MSI1	MSI0	ICLR	ICRE
R/W Attribute	R/W	R,WX	R/W	R,W	R,W	R,W	R,WX	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	BFE	STOP	MODE	SCLR	CLK3	CLK2	CLK1	CLK0
R/W Attribute	R/W	R/W	R/W	R0,W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	1	0	0	0	0	0	0

[bit31:16] Reserved Bits

"1" is always read from this bit.

Writing data to these bits has no effect on the operation.

[bit15] ECKE: Clock Selection Bit

This bit is used to select a peripheral clock or an external clock as the count clock of the free-run timer.
When this bit is set to "0", a peripheral clock is selected.
To select the count clock frequency, the clock frequency selection bits (CLK3 to CLK0: bit3 to bit0) must also be selected.
When this bit is set to "1", an external clock is selected and the clock is input from the "TEXT" pin.

This bit is cleared to "0" by writing "1" to the ECKEC bit of the TCCSC register.
This bit is set to "1" by writing "1" to the ECKES bit of the TCCSS register.

Bit	Description
0	Peripheral clock
1	External clock

Note:

- *The count clock is changed immediately when this bit is changed. Therefore, change this bit when the output compare and the input capture are stopped if the free-run timer is connected to them.*

[bit14] IRQZF: 0 Detection Interrupt Flag Bit

This bit is set to "1" when the count value of the free-run timer is 0.

This bit is a read-only bit. Writing data to these bits has no effect on the operation.

This bit is cleared by writing "1" to the IRQZFC bit of the TCCSC register.

Bit	Description
0	0 is not detected.
1	0 is detected.

Notes:

- *This bit is not set to "1" when the timer is cleared ("1" is written to the SCLR: bit4) while the free-run timer is operating (STOP: bit6=0 of the timer enable bit).*
- *In the up/down count mode (MODE: bit5=1), this bit is set to "1" when an interrupt that is set by the interrupt mask selection bits (MSI2 to MSI0: bit12 to bit10) occurs. This bit is not set to "1" when no interrupt occurs.*
- *In the up count mode (MODE: bit5=0), this bit is set to "1" every time 0 is detected regardless of the value of the interrupt mask selection bits (MSI2 to MSI0: bit12 to bit10).*

[bit13] IRQZE: 0 Detection Request Enable Bit

When this bit is set to "1" and the 0 detection interrupt flag bit (IRQZF: bit14) is set to "1", an interrupt request to the CPU is generated.

This bit is cleared to "0" by writing "1" to the IRQZEC bit of the TCCSC register. This bit is set to "1" by writing "1" to the IRQZES bit of the TCCSS register.

Bit	Description
0	Disable interrupt requests.
1	Enable interrupt requests.

[bit12:10] MSI[2:0]: Interrupt Mask Selection Bits

For MODE2: bit11=0 of the timer extended state control register (TECCS)

- In the up count mode (MODE: bit5=0), these bits are used to set the number of times to mask the compare clear interrupt flag. In the up/down count mode bit (MODE: bit5=1), these bits are used to set the number of times to mask the 0 detection interrupt flag.
- When these bits are set to 0b000, the interrupt flags are not masked.

For MODE2: bit11=1 of the timer extended state control register (TECCS)

- In the up/down count mode (MODE: bit5=1), these bits are used to set the number of times to mask the 0 detection interrupt flag.
- In the up count mode (MODE:bit5=0), it is prohibited to use these bits to make this setting.

Bits			Description
0	0	0	Generate an interrupt flag at the first match occurrence.
0	0	1	Generate an interrupt flag at the second match occurrence.
0	1	0	Generate an interrupt flag at the third match occurrence.
0	1	1	Generate an interrupt flag at the fourth match occurrence.
1	0	0	Generate an interrupt flag at the fifth match occurrence.
1	0	1	Generate an interrupt flag at the sixth match occurrence.
1	1	0	Generate an interrupt flag at the seventh match occurrence.
1	1	1	Generate an interrupt flag at the eighth match occurrence.

Notes:

The value read is a mask counter value. The mask counter is a decrement counter.

- *The written data is written to the mask register.*
- *If Interrupt mask function "by setting TCCS.MSI[2:0] Register" is used, please write TCCS[15:8], TCCSC[15:8] and TCCSS[15:8] after 32bit free-run timer stopped.*
-

[bit9] ICLR: Compare Clear Interrupt Flag Bit

This bit is set to "1" when the value of the compare clear register (CPCLR) and the value of the free-run timer match.

This bit is a read-only bit. Writing data to these bits has no effect on the operation.

This bit is cleared by writing "1" to the ICLRC bit of the TCCSC register.

Bit	Description
0	No compare clear match.
1	Compare clear match.

Notes:

- *In the up count mode (MODE: bit5=0), this bit is set to "1" when the interrupt flag set by the interrupt mask selection bits (MSI2 to MSI0) occurs. This bit is not set to "1" when no interrupt occurs.*
- *In the up/down count mode (MODE: bit5=1), regardless of the value of the interrupt mask selection bits (MSI2 to MSI0), this bit is set to "1" every time the compare clear occurs.*

[bit8] ICRE: Compare Clear Interrupt Request Enable Bit

When this bit is set to "1" and the compare clear interrupt flag bit (ICLR: bit9) is set to "1", an interrupt request to the CPU is generated.

This bit is cleared to "0" by writing "1" to the ICREC bit of the TCCSC register.
This bit is set to "1" by writing "1" to the ICRES bit of the TCCSS register.

Bit	Description
0	Disable interrupt requests.
1	Enable interrupt requests.

[bit7] BFE: Compare Clear Buffer Enable Bit

This bit is used to enable the compare clear buffer register (CPCLRB).
When this bit is set to "0":
The compare clear buffer register (CPCLRB) becomes invalid. Thus, data can be written directly to the compare clear register (CPCLR).
When this bit is set to "1":
The compare clear buffer register (CPCLRB) becomes valid. The data written and stored in the compare clear buffer register (CPCLRB) is transferred to the compare clear register (CPCLR) when "0" is detected as the count value of the free-run timer.

This bit is cleared to "0" by writing "1" to the BFEC bit of the TCCSC register.
This bit is set to "1" by writing "1" to the BFES bit of the TCCSS register.

Bit	Description
0	Disable the compare clear buffer.
1	Enable the compare clear buffer.

[bit6] STOP: Timer Enable Bit

This bit is used to stop/start counting of the free-run timer.
When this bit is set to "0":
The free-run timer starts counting.
When this bit is set to "1":
The free-run timer stops counting.

This bit is cleared to "0" by writing "1" to the STOPC bit of the TCCSC register.
This bit is set to "1" by writing "1" to the STOPS bit of the TCCSS register.

Bit	Description
0	Enable counting (start counting).
1	Disable counting (stop counting).

[bit5] MODE: Timer Count Mode Bit

This bit is used to select a count mode of the free-run timer.
When this bit is set to "0":
The up count mode is selected. The timer continues to count up until the count value matches the value of the compare clear register (CPCLR) and is reset to 0. After that, it starts counting up again.
When this bit is set to "1":
The up/down count mode is selected. The timer continues to count up until the count value matches the value of the compare clear register (CPCLR). After that, the count direction changes to down count. When the count value reaches 0, the count direction changes to up count again.

A value can be written to this bit regardless of whether the timer is operating (STOP: bit6=0 of the timer enable bit) or stopped (STOP: bit6=1). When the timer is operating, the value written to this bit is stored in a buffer. After that, when the timer value becomes 0, the count mode is set based on the value stored in the buffer.

This bit is cleared to "0" by writing "1" to the MODEC bit of the TCCSC register.
This bit is set to "1" by writing "1" to the MODES bit of the TCCSS register.

Bit	Description
0	Up count mode
1	Up/down count mode

[bit4] SCLR: Timer Clear Bit

This bit is used to initialize the free-run timer.

Initializing the value of the free-run timer:
The free-run timer is initialized to 0 at the next count clock when "1" is written to this bit while the free-run timer is operating (STOP: bit6=0 of the timer enable bit).

The free-run timer is not initialized if "1" is written to this bit when the free-run timer is stopped (STOP: bit 6=1 of the timer enable bit).

Initializing the count direction of the free-run timer:
The free-run timer always starts the operation from up count when its operation is started again (STOP: bit6=0 of the timer enable bit) after "1" is written to this bit.
The free-run timer starts the operation from up count even if it stops the operation when counting down (STOP: bit6=1 of the timer enable bit), and then starts operating again (STOP: bit6=0 of the timer enable bit) after "1" is written to this bit.

This bit is set to "1" by writing "1" to the SCLRS bit of the TCCSS register.

The read value is always "0".

Bit	Description	
	During Read Operation	During Write Operation
0	"0" always read	Do not initialize the counter.
1		Initialize the counter to 0x00000000.

Notes:

- *The 0 detection interrupt is not generated even if this bit is set to "1".*
  The timer is not cleared when "0" is written to this bit before the next count clock after "1" is written to this bit.
- *In the up/down count mode, when "0" is written to the SCLR bit before updating the timer count after "1" is written to the SCLR bit while the timer is counting down, the count value is not updated and the count direction is changed to up count.*

[bit3:0] CLK[3:0]: Clock Frequency Selection Bits

These bits are used to select the count clock frequency of the free-run timer.
The clock frequency is changed immediately upon setting these bits.

Bits				Description					
				Count Clock	ϕ = 40 MHz	ϕ = 20 MHz	ϕ = 10 MHz	ϕ = 5 MHz	ϕ = 2.5 MHz
0	0	0	0	Φ	25 ns	50 ns	100 ns	200 ns	400 ns
0	0	0	1	Φ/2	50 ns	100 ns	200 ns	400 ns	800 ns
0	0	1	0	Φ/4	100 ns	200 ns	400 ns	800 ns	1.6 us
0	0	1	1	Φ/8	200 ns	400 ns	800 ns	1.6 us	3.2 us
0	1	0	0	Φ/16	400 ns	800 ns	1.6 us	3.2 us	6.4 us
0	1	0	1	Φ/32	800 ns	1.6 us	3.2 us	6.4 us	12.8 us
0	1	1	0	Φ/64	1.6 us	3.2 us	6.4 us	12.8 us	25.6 us
0	1	1	1	Φ/128	3.2 us	6.4 us	12.8 us	25.6 us	51.2 us
1	0	0	0	Φ/256	6.4 us	12.8 us	25.6 us	51.2 us	102.4 us
Other settings are prohibited.				-	-	-	-	-	-

ϕ: Peripheral clock.

4.4. Timer Extended State Control Register (TECCS)

The timer extended state control register (TECCS) is an extended control register that controls the operation of the free-run timer.

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R1,WX							
Protection Attribute	-							
Initial Value	11111111							

bit	23	22	21	20	19	18	17	16
Field	Reserved							
R/W Attribute	R1,WX							
Protection Attribute	-							
Initial Value	11111111							

bit	15	14	13	12	11	10	9	8
Field	Reserved				MODE2	MSI5	MSI4	MSI3
R/W Attribute	R0,WX				R/W	R,W	R,W	R,W
Protection Attribute	-				-	-	-	-
Initial Value	0000				0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	Reserved							
R/W Attribute	R1,WX							
Protection Attribute	-							
Initial Value	11111111							

[bit31:16] Reserved: Reserved Bits

"1" is always read from this bit.

Writing data to these bits has no effect on the operation.

[bit15:12] Reserved: Reserved Bits

"0" is always read from this bit.

Writing data to these bits has no effect on the operation.

[bit11] MODE2: Interrupt Mask Mode Bit 2

This bit is used to independently mask the 0 detection interrupt and the compare clear interrupt when the free-run timer is in the up/down count mode (MODE: bit5=1 of the timer state control register (TCCS)).

If "1" is written to this bit when the free-run timer is in the up/down count mode (MODE: bit5=1 of the timer state control register (TCCS)), the value set to the compare clear interrupt mask selection bits (MSI5 to MSI3: bit10 to bit8) of this register is enabled and the compare clear interrupt flag is masked the specified number of times. The value set to the interrupt mask selection bits (MSI2 to MSI0: bit12 to bit10) of the timer state control register (TCCS) becomes valid as the number of times to mask the 0 detection interrupt flag.

Bits		Description
MODE2	MODE*	
0	0	The setting values of MSI5 to MSI3 are invalid.
0	1	The setting values of MSI5 to MSI3 are invalid.
1	0	The setting is prohibited (the operation is not guaranteed).
1	1	The setting values of MSI5 to MSI3 are valid.

*: bit5 of the timer state control register (TCCS)

Note:

- *The operation when "1" is written to this bit is not guaranteed when the free-run timer is in the up count mode.*

[bit10:8] MSI[5:3]: Compare Clear Interrupt Mask Selection Bits

- These bits are valid only when the interrupt mask mode bit 2 is "1" (MODE2: bit11=1) and the free-run timer is in the up/down count mode (MODE: bit5=1 of the timer state control register (TCCS)). These bits are used to set the number of times to mask the compare clear interrupt flag.
  The number of times to mask the 0 detection interrupt flag is set by MSI2 to MSI0: bit12 to bit10 of the timer state control register (TCCS).
- The compare clear interrupt flag is not masked when these bits are set to 0b000.

Bits			Description
0	0	0	Generate an interrupt flag at the first match occurrence.
0	0	1	Generate an interrupt flag at the second match occurrence.
0	1	0	Generate an interrupt flag at the third match occurrence.
0	1	1	Generate an interrupt flag at the fourth match occurrence.
1	0	0	Generate an interrupt flag at the fifth match occurrence.
1	0	1	Generate an interrupt flag at the sixth match occurrence.
1	1	0	Generate an interrupt flag at the seventh match occurrence.
1	1	1	Generate an interrupt flag at the eighth match occurrence.

Notes:

- *The value read is a mask counter value. The mask counter is a decrement counter.*
- *The written data is written to the mask register.*

- *While the free-run timer is operating (the timer enable bit STOP: bit6=0 of the timer state control register (TCCS)), the value written to the mask register is reloaded to the counter only when the mask counter becomes 0.*
- *When the free-run timer is stopped (the timer enable bit STOP: bit6=1 of the timer state control register (TCCS)), the value written to the mask register is immediately reloaded to the mask counter.*

[bit7:0] Reserved: Reserved Bits

"1" is always read from this bit.

Writing data to these bits has no effect on the operation.

4.5. Timer State Control Clear Register (TCCSC)

The timer state control clear register (TCCSC) is a register that is used to clear bits of the timer state control register (TCCS).

For details on writing to this register, see "5. Precautions for Using This Device "

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial Value	00000000							

bit	23	22	21	20	19	18	17	16
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial Value	00000000							

bit	15	14	13	12	11	10	9	8
Field	ECKEC	IRQZFC	IRQZEC	Reserved			ICLRC	ICREC
R/W Attribute	R0,W	R0,W	R0,W	R0,W0			R0,W	R0,W
Protection Attribute	-	-	-	-			-	-
Initial Value	0	0	0	000			0	0

bit	7	6	5	4	3	2	1	0
Field	BFEC	STOPC	MODEC	Reserved				
R/W Attribute	R0,W	R0,W	R0,W	R0,W0				
Protection Attribute	-	-	-	-				
Initial Value	0	0	0	00000				

[bit31:16] Reserved: Reserved Bits

"0" is always read from this bit.
Always write "0" to this bit.

[bit15] ECKEC: ECKE Clear Bit

When "1" is written to this bit, the clock selection bit (ECKE) of the timer state control register (TCCS) is cleared. "0" is always read from this bit.

Bit	Description
0	Affect neither this bit nor the clock selection bit (ECKE) of the timer state control register (TCCS).
1	Clear the clock selection bit (ECKE) of the timer state control register (TCCS).

[bit14] IRQZFC: IRQZF Clear Bit

When "1" is written to this bit, the 0 detection interrupt flag bit (IRQZF) of the timer state control register (TCCS) is cleared. "0" is always read from this bit.

Bit	Description
0	Affect neither this bit nor the 0 detection interrupt flag bit (IRQZF) of the timer state control register (TCCS).
1	Clear the 0 detection interrupt flag bit (IRQZF) of the timer state control register (TCCS).

[bit13] IRQZEC: IRQZE Clear Bit

When "1" is written to this bit, the 0 detection request enable bit (IRQZE) of the timer state control register (TCCS) is cleared. "0" is always read from this bit.

Bit	Description
0	Affect neither this bit nor the 0 detection request enable bit (IRQZE) of the timer state control register (TCCS).
1	Clear the 0 detection request enable bit (IRQZE) of the timer state control register (TCCS).

[bit12:10] Reserved: Reserved Bits

"0" is always read from this bit.
Always write "0" to this bit.

[bit9] ICLRC: ICLR Clear Bit

When "1" is written to this bit, the compare clear interrupt flag bit (ICLR) of the timer state control register (TCCS) is cleared. "0" is always read from this bit.

Bit	Description
0	Affect neither this bit nor the compare clear interrupt flag bit (ICLR) of the timer state control register (TCCS).
1	Clear the compare clear interrupt flag bit (ICLR) of the timer state control register (TCCS).

[bit8] ICREC: ICRE Clear Bit

When "1" is written to this bit, the compare clear interrupt request enable bit (ICRE) of the timer state control register (TCCS) is cleared. "0" is always read from this bit.

Bit	Description
0	Affect neither this bit nor the compare clear interrupt request enable bit (ICRE) of the timer state control register (TCCS).
1	Clear the compare clear interrupt request enable bit (ICRE) of the timer state control register (TCCS).

[bit7] BFEC: BFE Clear Bit

When "1" is written to this bit, the compare clear buffer enable bit (BFE) of the timer state control register (TCCS) is cleared. "0" is always read from this bit.

Bit	Description
0	Affect neither this bit nor the compare clear buffer enable bit (BFE) of the timer state control register (TCCS).
1	Clear the compare clear buffer enable bit (BFE) of the timer state control register (TCCS).

[bit6] STOPC: STOP Clear Bit

When "1" is written to this bit, the timer enable bit (STOP) of the timer state control register (TCCS) is cleared. "0" is always read from this bit.

Bit	Description
0	Affect neither this bit nor the timer enable bit (STOP) of the timer state control register (TCCS).
1	Clear the timer enable bit (STOP) of the timer state control register (TCCS).

[bit5] MODEC: MODE Clear Bit

When "1" is written to this bit, the timer count mode bit (MODE) of the timer state control register (TCCS) is cleared. "0" is always read from this bit.

Bit	Description
0	Affect neither this bit nor the timer count mode bit (MODE) of the timer state control register (TCCS).
1	Clear the timer count mode bit (MODE) of the timer state control register (TCCS).

[bit4:0] Reserved: Reserved Bits

"0" is always read from this bit. Always write "0" to this bit.

4.6. Timer State Control Set Register (TCCSS)

The timer state control set register (TCCSC) is a register that is used to set bits of the timer state control register (TCCS).

For details on writing to this register, see "5. Precautions for Using This Device "

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial Value	00000000							

bit	23	22	21	20	19	18	17	16
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial Value	00000000							

bit	15	14	13	12	11	10	9	8
Field	ECKES	Reserved	IRQZES	Reserved				ICRES
R/W Attribute	R0,W	R0,W0	R0,W	R0,W0				R0,W
Protection Attribute	-	-	-	-				-
Initial Value	0	0	0	0000				0

bit	7	6	5	4	3	2	1	0
Field	BFES	STOPS	MODES	SCLRS	Reserved			
R/W Attribute	R0,W	R0,W	R0,W	R0,W	R0,W0			
Protection Attribute	-	-	-	-	-			
Initial Value	0	0	0	0	0000			

[bit31:16] Reserved: Reserved Bits

"0" is always read from this bit.
Always write "0" to this bit.

[bit15] ECKES: ECKE Set Bit

When "1" is written to this bit, the clock selection bit (ECKE) of the timer state control register (TCCS) is set to "1". "0" is always read from this bit.

Bit	Description
0	Affect neither this bit nor the clock selection bit (ECKE) of the timer state control register (TCCS).
1	Set the clock selection bit (ECKE) of the timer state control register (TCCS).

[bit14] Reserved: Reserved Bits

"0" is always read from this bit.
Always write "0" to this bit.

[bit13] IRQZES: IRQZE Set Bit

When "1" is written to this bit, the 0 detection request enable bit (IRQZE) of the timer state control register (TCCS) is set to "1". "0" is always read from this bit.

Bit	Description
0	Affect neither this bit nor the 0 detection request enable bit (IRQZE) of the timer state control register (TCCS).
1	Set the 0 detection request enable bit (IRQZE) of the timer state control register (TCCS).

[bit12:9] Reserved: Reserved Bits

"0" is always read from this bit.
Always write "0" to this bit.

[bit8] ICRES: ICRE Set Bit

When "1" is written to this bit, the compare clear interrupt request enable bit (ICRE) of the timer state control register (TCCS) is set to "1". "0" is always read from this bit.

Bit	Description
0	Affect neither this bit nor the compare clear interrupt request enable bit (ICRE) of the timer state control register (TCCS).
1	Set the compare clear interrupt request enable bit (ICRE) of the timer state control register (TCCS).

[bit7] BFES: BFE Set Bit

When "1" is written to this bit, the compare clear buffer enable bit (BFE) of the timer state control register (TCCS) is set to "1". "0" is always read from this bit.

Bit	Description
0	Affect neither this bit nor the compare clear buffer enable bit (BFE) of the timer state control register (TCCS).
1	Set the compare clear buffer enable bit (BFE) of the timer state control register (TCCS).

[bit6] STOPS: STOP Set Bit

When "1" is written to this bit, the timer enable bit (STOP) of the timer state control register (TCCS) is set to "1". "0" is always read from this bit.

Bit	Description
0	Affect neither this bit nor the timer enable bit (STOP) of the timer state control register (TCCS).
1	Set the timer enable bit (STOP) of the timer state control register (TCCS).

[bit5] MODES: MODE Set Bit

When "1" is written to this bit, the timer count mode bit (MODE) of the timer state control register (TCCS) is set to "1". "0" is always read from this bit.

Bit	Description
0	Affect neither this bit nor the timer count mode bit (MODE) of the timer state control register (TCCS).
1	Set the timer count mode bit (MODE) of the timer state control register (TCCS).

[bit4] SCLRS: SCLR Set Bit

When "1" is written to this bit, the timer clear bit (SCLR) of the timer state control register (TCCS) is set to "1". "0" is always read from this bit.

Bit	Description
0	Affect neither this bit nor the timer clear bit (SCLR) of the timer state control register (TCCS).
1	Set the timer clear bit (SCLR) of the timer state control register (TCCS).

[bit3:0] Reserved: Reserved Bits

"0" is always read from this bit.
Always write "0" to this bit.

5. Precautions for Using This Device

The following shows the notes when using the 32-bit free-run timer.

Notes to Observe When Accessing a Register

When Accessing the Compare Clear Register (CPCLR) or the Compare Clear Buffer Register (CPCLRB)

Use word access instructions to the compare clear register (CPCLR) and the compare clear buffer register (CPCLRB).

When Accessing the Timer State Control Register (TCCS)

- This register doesn't supports writing from the bit-band alias area. For the bit-band alias area, see "CHAPTER: BIT-BAND UNIT."
- To clear a specified bit of this register, clear the bit by writing "1" to the applicable bit of the timer state control clear register (TCCSC).
- To set a specified bit of this register, set the bit by writing "1" to the applicable bit of the timer state control set register (TCCSS).
- Data can be written directly to this register only when writing to all bits.
- In the normal reading mode, the interrupt mask counter value is read from Interrupt Mask Selection bits(MSI[2:0]) in the Timer State Control Register (TCCS).
- If Interrupt mask function "by setting TCCS.MSI[2:0] Register" is used, please write TCCS[15:8], TCCSC[15:8] and TCCSS[15:8] after 32bit free-run timer stopped.

When Accessing the Timer Extended State Control Register (TECCS)

- In the normal reading mode, the interrupt mask counter value is read from MSI5 to MSI3.

Notes When Operating the Free-Run Timer

When Setting Using a Program

- When the hardware is reset, the count value becomes 0x00000000, but the 0 detection interrupt flag is not set.
- Because the timer mode bit (MODE: bit5 of the timer state control register (TCCS)) has a buffer, a specified timer mode is enabled after 0 is detected.
- The timer clear bit (SCLR: bit4=1 of the timer state control register (TCCS)) initializes the timer. However, this bit does not generate the 0 detection interrupt.
- The compare clear flag is not set if the timer starts counting when the value of the compare clear register (CPCLR) matches the count value.
- Set any values other than 0 to the compare clear register (CPCLR). Note that the following operations occur if 0 is set.
  - When the timer mode bit (MODE: bit5 of the timer state control register (TCCS)) is in the up count mode (MODE: bit5=0), the count value is updated to 0 and fixed to 0. Then, the 0 detection interrupt flag and the compare clear flag are set at every count clock.
  - When the timer count mode bit (MODE: bit5 of the timer state control register (TCCS)) is in the up/down count mode (MODE: bit5=1), the count value is counted up from 0x00000000 to 0xFFFFFFFF, and this up count operation is repeated. The 0 detection interrupt flag and the compare clear flag are set to "1" when the count value becomes 0.

CHAPTER 43: 32-bit Input Capture

This chapter describes the functions of the 32-bit input capture.

1. Overview of the 32-Bit Input Capture
2. Configuration and Block Diagram
3. Explanation of 32-Bit Input Capture Operation
4. Registers of the 32-Bit Input Capture
5. Precautions for Using This Device

ICU-TXXPT03P01R01L04-E1-XX

1. Overview of the 32-Bit Input Capture

The 32-bit input capture can measure the input pulse width and external clock cycle based on the value of the 32-bit free-run timer.

Functions of the 32-Bit Input Capture

- The 32-bit input capture is composed of 2 input captures. Each channel can be operated individually.
- Each input capture is composed of 2 independent external input pins (IN0 and IN1), capture registers that correspond to the pins, and the capture control register. When an edge signal is detected on an external input pin, the value of the free-run timer can be stored in the capture register. At the same time, an interrupt is generated.
- 3 types of trigger edges (rising edge, falling edge, and both edges) of an external input signal can be selected. There is a register that indicates whether a trigger edge is a rising or falling edge.
- An interrupt is generated when a valid edge is detected from an external input signal.

Interrupt request and DMA request

- Interrupt request of this module can be used for purposes below:
  1. Interrupt request to CPU (via Interrupt Controller)
  2. DMA request to DMA Controller
- To use the interrupt as interrupt request to CPU, following configuration is necessary:
  - Enable the interrupt request in this module
  - Enable the interrupt channel of the interrupt in Interrupt Controller
- To use the interrupt as DMA request to DMA Controller, following configuration is necessary:
  - Enable the interrupt request in this module
  - Select the DMA channel from this module as DMA client in DMA Controller

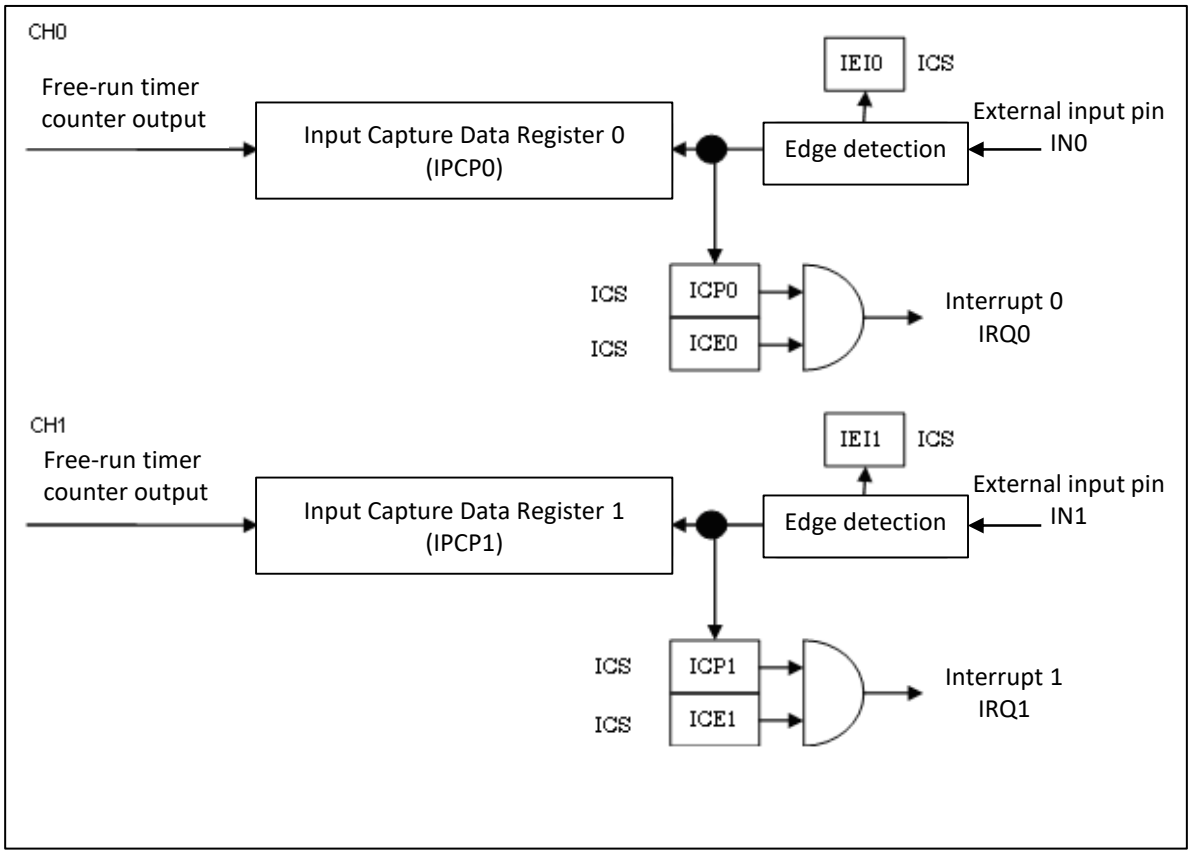
Note:

- *Interrupt request that supports for DMA request is product specification. Please refer to the product's HWM for detail.*

2. Configuration and Block Diagram

Figure 2-1 is a configuration diagram of the 32-bit input capture.

Figure 2-1 Configuration Diagram of 32-Bit Input Capture



3. Explanation of 32-Bit Input Capture Operation

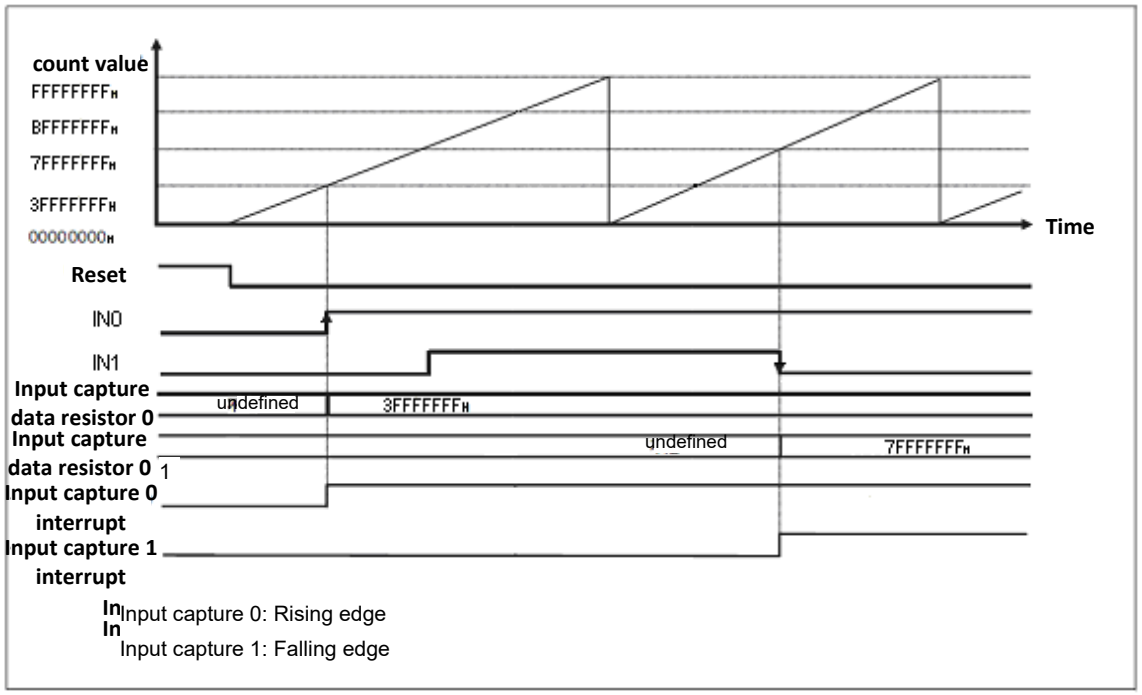
This section provides an overview of 32-bit input capture operation.

Operation of the 32-Bit Input Capture

The 32-bit input capture is used to detect a specified valid edge. When a valid edge is detected, an interrupt flag is set. Then the value of the 32-bit free-run timer is loaded to the input capture register.

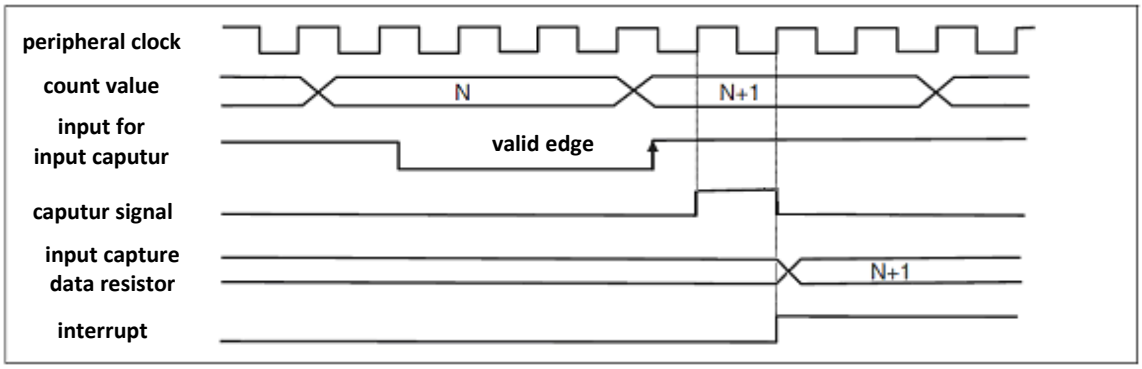
Input Capture Operation

Figure 3-1 Example of Input Capture Timing



Input Capture Input Timing

Figure 3-2 Example of Input Capture Timing for Input Signals



3.1. Interrupt of the 32-Bit Input Capture

As the interrupt of the 32-bit input capture, there is an input capture interrupt triggered by an external input signal.

Input Capture Interrupt

Table 2-1 shows the interrupt control bits and interrupt factor of the input capture.

Table 3-1 Interrupt Control Bits and Interrupt Factor of Input Capture 1, 0

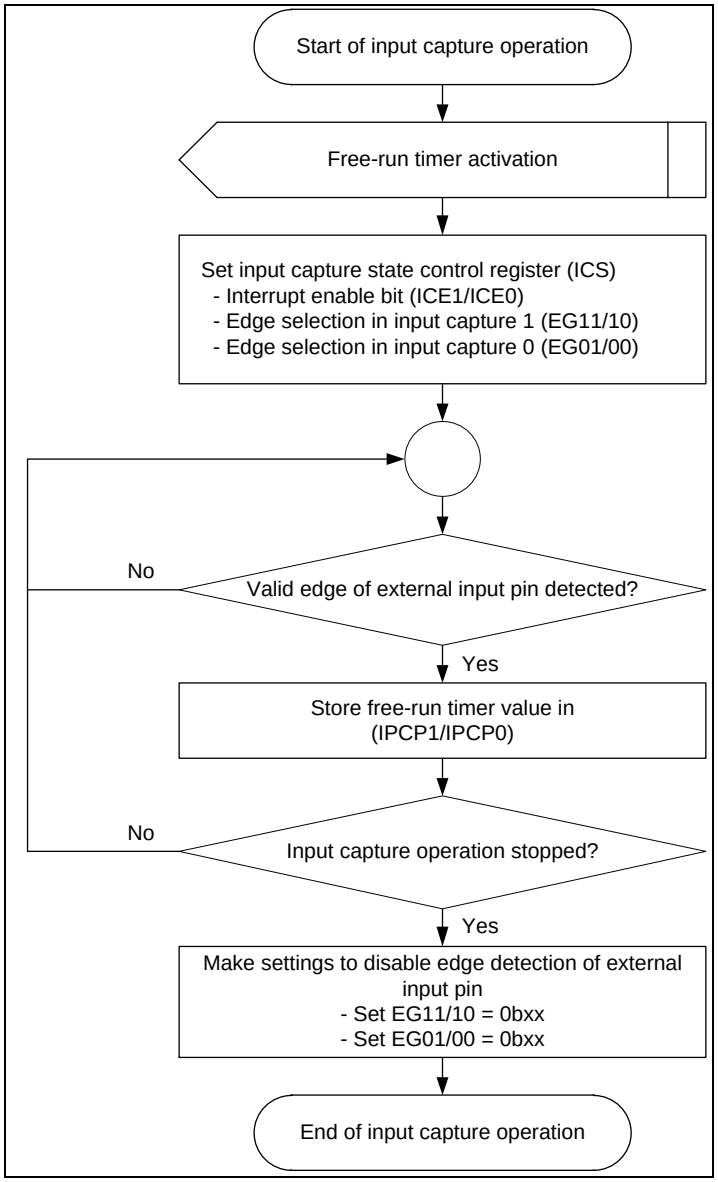
	Input Capture 1, 0
Interrupt Request Flag Bit	The interrupt request flag bits ICP1, 0 (bit7, bit6) in the input capture state control register (ICS)
Interrupt Request Enable Bit	The interrupt request enable bits ICE1, 0 (bit5, bit4) in the input capture state control register (ICS)
Interrupt Factor	A valid edge is detected on the external input pin (IN1, IN0).

For the interrupt capture, when a valid edge is detected on the external input pin (IN1, IN0), the interrupt request flag (ICP1, ICP0: bit7, bit6) in the input capture state control register (ICS) is set to "1". When an interrupt request is set to enabled (ICE1, ICE0: bit5, bit4 = 0b11 in the input capture state control register (ICS)) in this state, then the interrupt request is output to the interrupt controller.

3.2. Setting Procedure Example

This section provides a setting procedure example of the input capture.

Figure 3-3 Procedure Example of Setting Input Capture Operation



4. Registers of the 32-Bit Input Capture

This section provides the register list of the 32-bit input capture.

Table 4-1 Register Map

Common Peripheral Group #0 (Channel No.:0, 1, 2, 3, 4, 5, 6, and 7)

Offset	Address Offset Value / Register Name				Block Name
	+3	+2	+1	+0	
0000_8000	ICUxx_IPCP0 XXXXXXXX_XXXXXXXX_XXXXXXXX_XXXXXXXX				Common Peripheral #0
0000_8004	ICUxx_IPCP1 XXXXXXXX_XXXXXXXX_XXXXXXXX_XXXXXXXX				
0000_8008	ICUxx_ICS 11111111_11111111_11111100_00000000				
0000_800C	ICUxx_ICSC 00000000_00000000_00000000_00000000				
0000_8010	ICUxx_ICSS 00000000_00000000_00000000_00000000				
0000_8014 0000_83FC	-				

Notes:

- "X": Initial value undefined
- "xx": Channel number (0 to 7)

Common Peripheral Group #1 (Channel No.:8, 9, 10, 11, 12, 13, 14, and 15)

Offset	Address Offset Value / Register Name				Block Name
	+3	+2	+1	+0	
0000_8000	ICUxx_IPCP0 XXXXXXXX_XXXXXXXX_XXXXXXXX_XXXXXXXX				Common Peripheral #1
0000_8004	ICUxx_IPCP1 XXXXXXXX_XXXXXXXX_XXXXXXXX_XXXXXXXX				
0000_8008	ICUxx_ICS 11111111_11111111_11111100_00000000				
0000_800C	ICUxx_ICSC 00000000_00000000_00000000_00000000				
0000_8010	ICUxx_ICSS 00000000_00000000_00000000_00000000				
0000_8014 0000_83FC	-				

Notes:

- "X": Initial value undefined
- "xx": Channel number (8 to 15)

Register List of the 32-Bit Input Capture

Table 4-1 Register List of the 32-Bit Input Capture

Abbreviated Register Name	Register Name	Reference
IPCP0/IPCP1	Input capture data registers 0, 1	4.1
ICS	Input capture state control register	4.2
ICSC	Input capture state control clear register	4.3
ICSS	Input capture state control set register	4.4

Register Bit Locations of 32-Bit Input Capture

Table 4-2 Register Bit Locations of 32-Bit Input Capture

	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
IPCP0	CP31	CP30	CP29	CP28	CP27	CP26	CP25	CP24
	CP23	CP22	CP21	CP20	CP19	CP18	CP17	CP16
	CP15	CP14	CP13	CP12	CP11	CP10	CP09	CP08
	CP07	CP06	CP05	CP04	CP03	CP02	CP01	CP00
IPCP1	CP31	CP30	CP29	CP28	CP27	CP26	CP25	CP24
	CP23	CP22	CP21	CP20	CP19	CP18	CP17	CP16
	CP15	CP14	CP13	CP12	CP11	CP10	CP09	CP08
	CP07	CP06	CP05	CP04	CP03	CP02	CP01	CP00
ICS	Reserved							
	Reserved							
	Reserved						IEI1	IEI0
	ICP1	ICP0	ICE1	ICE0	EG11	EG10	EG01	EG00
ICSC	Reserved							
	Reserved							
	Reserved							
	ICP1C	ICP0C	ICE1C	ICE0C	Reserved			
ICSS	Reserved							
	Reserved							
	Reserved							
	Reserved		ICE1S	ICE0S	Reserved			

4.1. Input Capture Data Registers 0, 1 (IPCP0, IPCP1)

Input capture data registers 0, 1 (IPCP0, IPCP1) are used to store the count value of the free-run timer when a valid edge of an external input signal is detected.

Input Capture Data Register (IPCP0)

bit	31	30	29	28	27	26	25	24
Field	CP31	CP30	CP29	CP28	CP27	CP26	CP25	CP24
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	X	X	X	X	X	X	X	X

bit	23	22	21	20	19	18	17	16
Field	CP23	CP22	CP21	CP20	CP19	CP18	CP17	CP16
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	X	X	X	X	X	X	X	X

bit	15	14	13	12	11	10	9	8
Field	CP15	CP14	CP13	CP12	CP11	CP10	CP09	CP08
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	X	X	X	X	X	X	X	X

bit	7	6	5	4	3	2	1	0
Field	CP07	CP06	CP05	CP04	CP03	CP02	CP01	CP00
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	X	X	X	X	X	X	X	X

[bit31:0] CP[31:00]: Input Capture Data Value Bit

The input capture data register 0 (IPCP0) is used to store the value of the free-run timer when a valid edge of the external input pin IN0 signal is detected.

The free-run timer that is mentioned here indicates the operating state of the free-run timer that is connected to the input capture.

Note:

- *For access to this register, use word access instructions.*

Input Capture Data Register (IPCP1)

bit	31	30	29	28	27	26	25	24
Field	CP31	CP30	CP29	CP28	CP27	CP26	CP25	CP24
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	X	X	X	X	X	X	X	X

bit	23	22	21	20	19	18	17	16
Field	CP23	CP22	CP21	CP20	CP19	CP18	CP17	CP16
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	X	X	X	X	X	X	X	X

bit	15	14	13	12	11	10	9	8
Field	CP15	CP14	CP13	CP12	CP11	CP10	CP09	CP08
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	X	X	X	X	X	X	X	X

bit	7	6	5	4	3	2	1	0
Field	CP07	CP06	CP05	CP04	CP03	CP02	CP01	CP00
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	X	X	X	X	X	X	X	X

[bit31:0] CP[31:00]: Input Capture Data Value Bit

Input capture data register 1 (IPCP1) is used to store the value of the free-run timer when a valid edge of the external input pin IN1 signal is detected.

The free-run timer that is mentioned here indicates the operating state of the free-run timer that is connected to the input capture.

Note:

- *For access to this register, use word access instructions.*

4.2. Input Capture State Control Register (ICS)

The input capture state control register (ICS) is used to select an edge, enable an interrupt request, and control an interrupt request flag. This register is also used to indicate a valid edge detected in the input captures 0, 1.

For details on writing to this register, see "5. Precautions for Using This Device."

Flag bits of the input capture state control registers are only read if the APB-IF is valid.

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R1,WX							
Protection Attribute	-							
Initial Value	11111111							

bit	23	22	21	20	19	18	17	16
Field	Reserved							
R/W Attribute	R1,WX							
Protection Attribute	-							
Initial Value	11111111							

bit	15	14	13	12	11	10	9	8
Field	Reserved						IEI1	IEI0
R/W Attribute	R1,WX						R,WX	R,WX
Protection Attribute	-						-	-
Initial Value	111111						0	0

bit	7	6	5	4	3	2	1	0
Field	ICP1	ICP0	ICE1	ICE0	EG11	EG10	EG01	EG00
R/W Attribute	R,WX	R,WX	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

[bit31:10] Reserved: Reserved Bits

The read value is "1".

Writing data to these bits has no effect on the operation.

[bit9] IEI1: Valid Edge Indication Bit (Input Capture 1)

- This bit is a valid edge indication bit in input capture data register 1. This bit indicates that a rising or falling edge is detected.
- When a falling edge is detected, this bit is set to "0".
- When a rising edge is detected, this bit is set to "1".
- This bit is read-only.

Bit	Description
0	A falling edge is detected.
1	A rising edge is detected.

Notes:

- *The value read is meaningless if the edge selection bit (EG11, EG10: bit3, bit2) is set to "00_B".*
- *If the edge selection bit (EG11, EG10: bit3, bit2) is set to any value other than "00_B", then the value is updated when the interrupt request flag (ICP1) is set.*

[bit8] IEI0: Valid Edge Indication Bit (Input Capture 0)

- This bit is a valid edge indication bit in input capture data register 0. This bit indicates that a rising or falling edge is detected.
- When a falling edge is detected, this bit is set to "0".
- When a rising edge is detected, this bit is set to "1".
- This bit is a read-only bit.

Bit	Description
0	A falling edge is detected.
1	A rising edge is detected.

Notes:

- *The value read is meaningless if the edge selection bit (EG01, EG00: bit1, bit0) is set to "00_B".*
- *If the edge selection bit (EG01, EG00: bit1, bit0) is set to any value other than "00_B", then the value is updated when the interrupt request flag (ICP0) is set.*

[bit7] ICP1: Interrupt Request Flag Bit (Input Capture 1)

- This bit is used as an interrupt request flag of input capture 1.
- This bit is set to "1" immediately upon detecting a valid edge on the external input pin (IN1).
- An interrupt is generated as soon as this bit is set to "1" when the interrupt request enable bit (ICE1: bit5) is "1".
- This bit is a read-only bit. Writing data to these bits has no effect on the operation.

- This bit is cleared to "0" by writing "1" to the ICP1C bit in the ICSC register.

Bit	Description
0	No valid edge is detected.
1	A valid edge is detected.

[bit6] ICP0: Interrupt Request Flag Bit (Input Capture 0)

- This bit is used as an interrupt request flag of input capture 0.
- This bit is set to "1" immediately upon detecting a valid edge on the external input pin (IN0).
- An interrupt is generated as soon as this bit is set to "1" when the interrupt request enable bit (ICE0: bit4) is "1".
- This bit is a read-only bit. Writing data to these bits has no effect on the operation.
- This bit is cleared to "0" by writing "1" to the ICP0C bit in the ICSC register.

Bit	Description
0	No valid edge is detected.
1	A valid edge is detected.

[bit5] ICE1: Interrupt Request Enable Bit (Input Capture 1)

- This bit is used to enable an interrupt request of input capture1.
- An interrupt of input capture 1 is generated when the interrupt request flag bit (ICP1: bit7) is set while this bit is "1".
- This bit is cleared to "0" by writing "1" to the ICE1C bit in the ICSC register.
- This bit is set to "1" by writing "1" to the ICE1S bit in the ICSS register.

Bit	Description
0	Disable interrupt requests.
1	Enable interrupt requests.

[bit4] ICE0: Interrupt Request Enable Bit (Input Capture 0)

- This bit is used to enable an interrupt request of input capture 0.
- An interrupt of input capture 0 is generated when the interrupt request flag bit (ICP0: bit6) is set while this bit is "1".
- This bit is cleared to "0" by writing "1" to the ICE0C bit in the ICSC register.
- This bit is set to "1" by writing "1" to the ICE0S bit in the ICSS register.

Bit	Description
0	Disable interrupt requests.
1	Enable interrupt requests.

[bit3:2] EG11, EG10: Edge Selection Bits (Input Capture 1)

- These bits are used to enable the operation of input capture 1. These bits specify a valid edge of the external input (IN1).

Bits		Description
EG11	EG10	
0	0	No edge is detected (stop).
0	1	A rising edge is detected.
1	0	A falling edge is detected.
1	1	Both edges are detected.

[bit1:0] EG01, EG00: Edge Selection Bits (Input Capture 0)

■ These bits are used to enable the operation of input capture 0. These bits specify a valid edge of the external input (IN0).

Bits		Description
EG01	EG00	
0	0	No edge is detected (stop).
0	1	A rising edge is detected.
1	0	A falling edge is detected.
1	1	Both edges are detected.

4.3. Input Capture State Control Clear Register (ICSC)

The input capture state control clear register (ICSC) is a register to clear a bit in the input capture state control register (ICS).

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial Value	00000000							

bit	23	22	21	20	19	18	17	16
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial Value	00000000							

bit	15	14	13	12	11	10	9	8
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial Value	00000000							

bit	7	6	5	4	3	2	1	0
Field	ICP1C	ICP0C	ICE1C	ICE0C	Reserved			
R/W Attribute	R0,W	R0,W	R0,W	R0,W	R0,W0			
Protection Attribute	-	-	-	-	-			
Initial Value	0	0	0	0	0000			

[bit31:8] Reserved: Reserved Bits

The read value is "0".

Always write "0" to this bit.

[bit7] ICP1C: ICP1 Clear Bit

- Writing "1" to this bit clears the ICP1 bit in the ICS register.
- "0" is always read from this bit.

Bit	Description
0	Do not affect this bit and the interrupt request flag bit (input capture1) (ICP1) in the input capture state control register (ICS).
1	Clear the interrupt request flag bit (input capture1) (ICP1) in the input capture state control register (ICS).

[bit6] ICP0C: ICP0 Clear Bit

- Writing "1" to this bit clears the ICP0 bit in the ICS register.
- "0" is always read from this bit.

Bit	Description
0	Do not affect this bit and the interrupt request flag bit (input capture0) (ICP0) in the input capture state control register (ICS).
1	Clear the interrupt request flag bit (input capture0) (ICP0) in the input capture state control register (ICS).

[bit5] ICE1C: ICE1 Clear Bit

- Writing "1" to this bit clears the ICE1 bit in the ICS register.
- "0" is always read from this bit.

Bit	Description
0	Do not affect this bit and the interrupt request enable bit (input capture 1) (ICE1) in the input capture state control register (ICS).
1	Clear the interrupt request enable bit (input capture 1) (ICE1) in the input capture state control register (ICS).

[bit4] ICE0C: ICE0 Clear Bit

- Writing "1" to this bit clears the ICE0 bit in the ICS register.
- "0" is always read from this bit.

Bit	Description
0	Do not affect this bit and the interrupt request enable bit (input capture 0) (ICE0) in the input capture state control register (ICS).
1	Clear the interrupt request enable bit (input capture 0) (ICE0) in the input capture state control register (ICS).

[bit3:0] Reserved: Reserved Bits

The read value is "0".
Always write "0" to this bit.

4.4. Input Capture State Control Set Register (ICSS)

The input capture state control set register (ICSS) is a register that sets the bit in the input capture state control register (ICS).

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial Value	00000000							

bit	23	22	21	20	19	18	17	16
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial Value	00000000							

bit	15	14	13	12	11	10	9	8
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial Value	00000000							

bit	7	6	5	4	3	2	1	0
Field	Reserved		ICE1S	ICE0S	Reserved			
R/W Attribute	R0,W0		R0,W	R0,W	R0,W0			
Protection Attribute	-		-	-	-			
Initial Value	00		0	0	0000			

[bit31:6] Reserved: Reserved Bits

The read value is "0".

Always write "0" to this bit.

[bit5] ICE1S: ICE1 Set Bit

- Writing "1" to this bit sets the ICE1 bit in the ICS register to "1".
- "0" is always read from this bit.

Bit	Description
0	Do not affect this bit and the interrupt request enable bit (input capture 1) (ICE1) in the input capture state control register (ICS).
1	Set the interrupt request enable bit (input capture 1) (ICE1) in the input capture state control register (ICS).

[bit4] ICE0S: ICE0 Set Bit

- Writing "1" to this bit sets the ICE0 bit in the ICS register to "1".
- "0" is always read from this bit.

Bit	Description
0	Do not affect this bit and the interrupt request enable bit (input capture 0) (ICE0) in the input capture state control register (ICS).
1	Set the interrupt request enable bit (input capture 0) (ICE0) in the input capture state control register (ICS).

[bit3:0] Reserved: Reserved Bit

The read value is "0".
Always write "0" to this bit.

5. Precautions for Using This Device

The following shows the notes when using the 32-bit input capture.

Notes to Observe when Accessing a Register

When Accessing the Input Capture Data Registers 0, 1 (IPCP0, 1)

Use word access instructions for the input capture data registers 0, 1 (IPCP0, 1).

When Accessing the Input Capture State Control Register (ICS)

- This register supports writing from the bit-band alias area. For the bit-band alias area, see "CHAPTER: BIT-BAND UNIT."
- To clear a specified bit in this register, clear the bit by writing "1" to the applicable bit in the input capture state control clear register (ICSC).
- To set a specified bit in this register, set the bit by writing "1" to the applicable bit in the input capture state control set register (ICSS).
- Data can be written directly to this register only when writing to all bits.

Notes on Interrupt Processing

- The valid edge indication bit (IEI1, IEI0: bit9, bit8) in the input capture state control register (ICS) indicates a detected latest edge when the level of the external input pin (IN0, IN1) switches while an interrupt routine is processing after the interrupt request flag (ICP1, ICP0) in the input capture state control register (ICS) is set to "1".

Notes on Input Capture Operation

About Capture Timing

- Capture resolution is 1 peripheral clock because the input capture operates according to the peripheral clock timing. The resolution is 1 timer count because the capture data is the timer counter value of the free-run timer.

CHAPTER 44: 32-bit Output Compare

This chapter describes the 32-bit output compare function.

1. **Overview**
2. **Configuration**
3. **Explanation of Operation**
4. **Setting Procedure Example**
5. **Registers**
6. **Precautions for Using This Device**

OCU-TXXPT03P01R01L05-E1-XX

1. Overview

The 32-bit output compare consists of the 32-bit compare register, compare output latch, and compare control register. It can output a waveform by using the timer value of the 32-bit free-run timer.

Functions of the 32-Bit Output Compare

- It consists of 2 output compares. You can set an initial value for each output pin (OUT0 and OUT1).
- The 32-bit output compare consists of the 32-bit compare register, compare output latch, and compare control register.
- You can use the 2 output compares in combination to control the output pins. The output pins are inverted by the use of the 2 compare registers.
- An interrupt is generated when the output compare register value matches the free-run timer value.

Interrupt request and DMA request

- Interrupt request of this module can be used for purposes below:
  1. Interrupt request to CPU (via Interrupt Controller)
  2. DMA request to DMA Controller
- To use the interrupt as interrupt request to CPU, following configuration is necessary:
  - Enable the interrupt request in this module
  - Enable the interrupt channel of the interrupt in Interrupt Controller
- To use the interrupt as DMA request to DMA Controller, following configuration is necessary:
  - Enable the interrupt request in this module
  - Select the DMA channel from this module as DMA client in DMA Controller

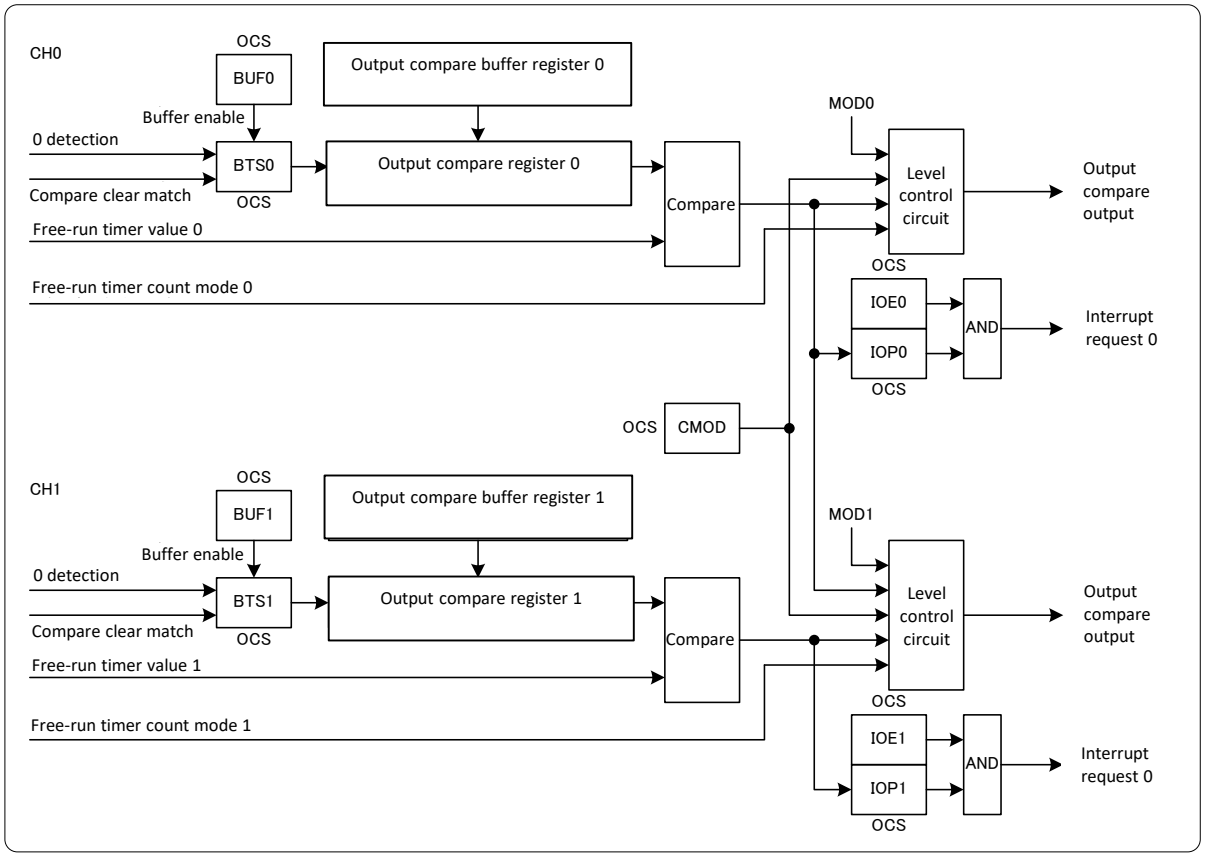
Note:

- *Interrupt request that supports for DMA request is product specification. Please refer to the product's HWM for detail.*

2. Configuration

Figure 2-1 shows the configuration of the 32-bit output compare.

Figure 2-1 Configuration of the 32-Bit Output Compare



3. Explanation of Operation

3.1. Interrupt

There is the compare match interrupt as a 32-bit output compare interrupt.

Output Compare Interrupt

Table 3-1 Interrupt Control Bits and Interrupt Factors of Output Compares (CH.0, CH.1)

	Output Compare (CH.0)	Output Compare (CH.1)
Interrupt Request Flag Bit	Compare match interrupt flag bit in the compare control register (OCS) (IOP0: bit6)	Compare match interrupt flag bit in the compare control register (OCS) (IOP1: bit7)
Interrupt Request Enable Bit	Compare match interrupt enable bit in the compare control register (OCS) (IOE0: bit4)	Compare match interrupt enable bit in the compare control register (OCS) (IOE1: bit5)
Interrupt Factor	The free-run timer value matches the value in the output compare register (OCCP0).	The free-run timer value matches the value in the output compare register (OCCP1).

When the free-run timer value matches the value in the output compare register (OCCP0, 1), the compare match interrupt flag in the compare control register (OCS) (IOP1, IOP0) is set to "1". In this state, when the interrupt request is set to be enabled (OCS:IOE1 = 1, OCS:IOE0 = 1), the interrupt request signal (IRQ0, IRQ1) outputs "H".

3.2. 32-Bit Output Compare Operation

The 32-bit output compare is used to compare the "value set in the specified compare clear register" and "value in the 32-bit free-run timer." When a match is detected, the interrupt flag is set and a compare output level is set according to the control signal.

When the free-run timer is in up-down count mode and the count peak and compare register value match, the match signal is ignored.

Output Compare Operation (for MODn Pin = "L" (Invert Mode))

Each Channel can Perform the Compare Operation (CMOD:bit12 in the Compare Control Register (OCS) = 0).

- The output compare 0 output (OUT0) is immediately inverted when the free-run timer and compare register 0 (OCCP0) match.
- The output compare 1 output (OUT1) is immediately inverted when the free-run timer and compare register 1 (OCCP1) match.

Figure 3-1 Output Waveform Example for When Compare Registers 0 and 1 are Separately Used with Initial Output Value = 0 (Free-Run Timer is in Up Count Mode.)

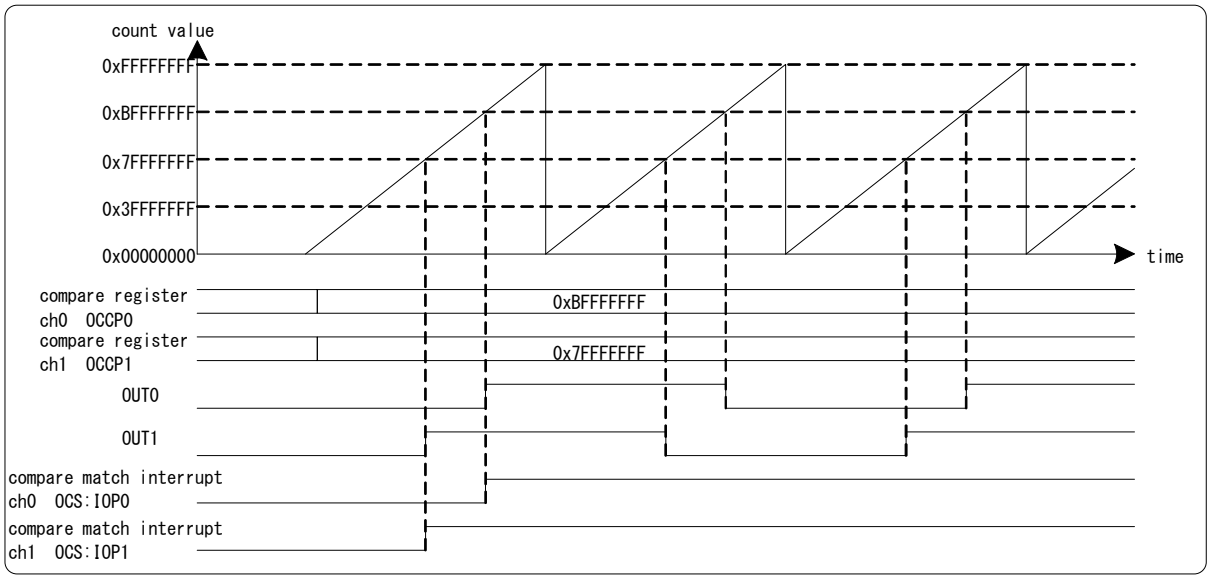
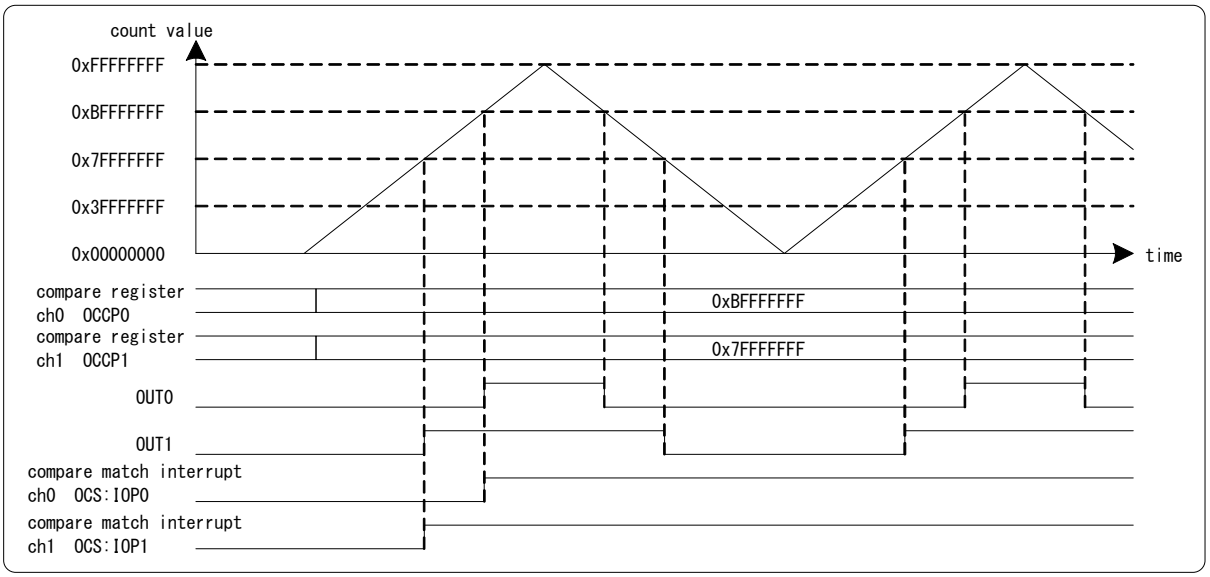


Figure 3-2 Output Waveform Example for When Compare Registers 0 and 1 are Separately Used with Initial Output Value = 0 (Free-Run Timer is in Up-Down Count Mode.)



The Channels can Perform the Compare Operation as a Pair of Channels (CMOD:bit12 in the Compare Control Register (OCS) = 1).

- The output compare 0 output (OUT0) is immediately inverted when the free-run timer and compare register 0 (OCCP0) match.
- The output compare 1 output (OUT1) is immediately inverted when the free-run timer and compare register 0 (OCCP0) or compare register 1 (OCCP1) match.

Figure 3-3 Output Waveform Example for When Compare Registers 0 and 1 are Used as a Pair with Initial Output Value = 0 (Free-Run Timer is in Up Count Mode.)

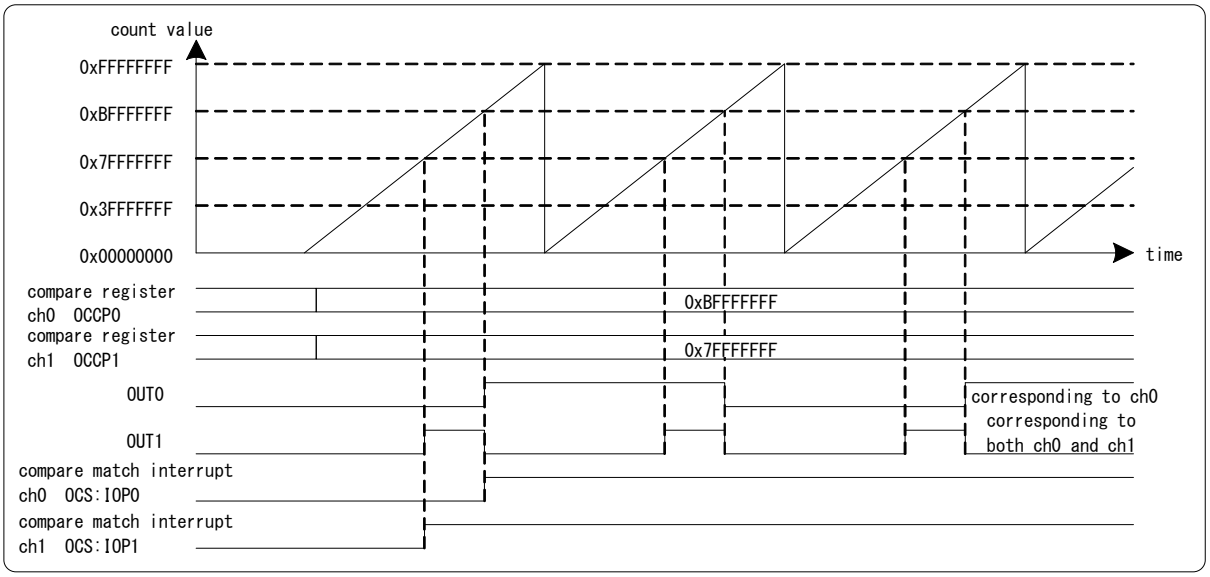
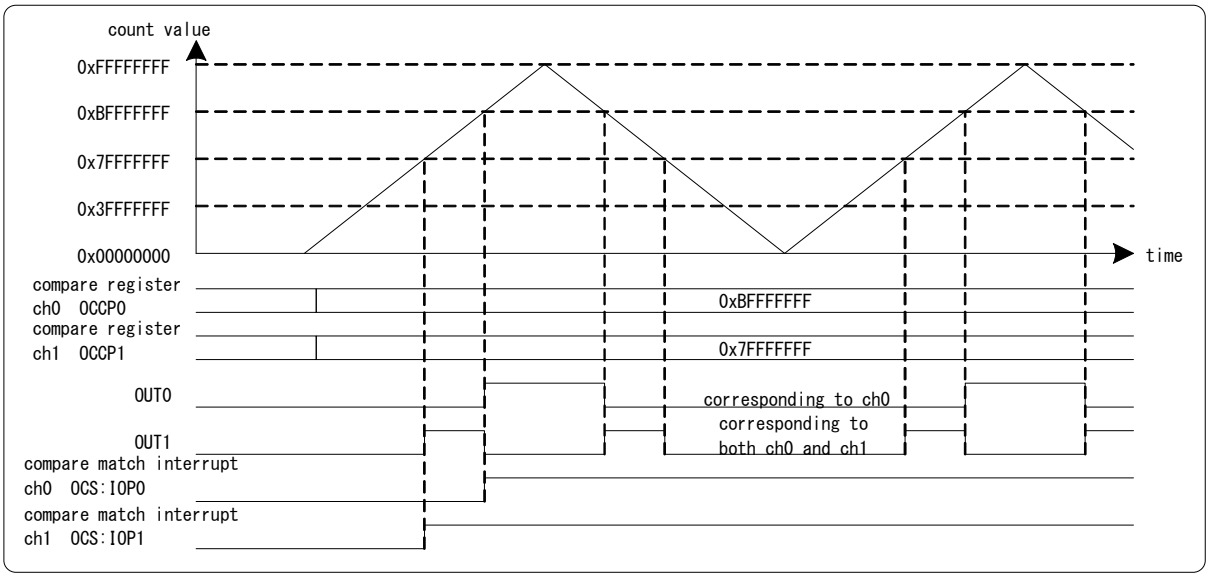
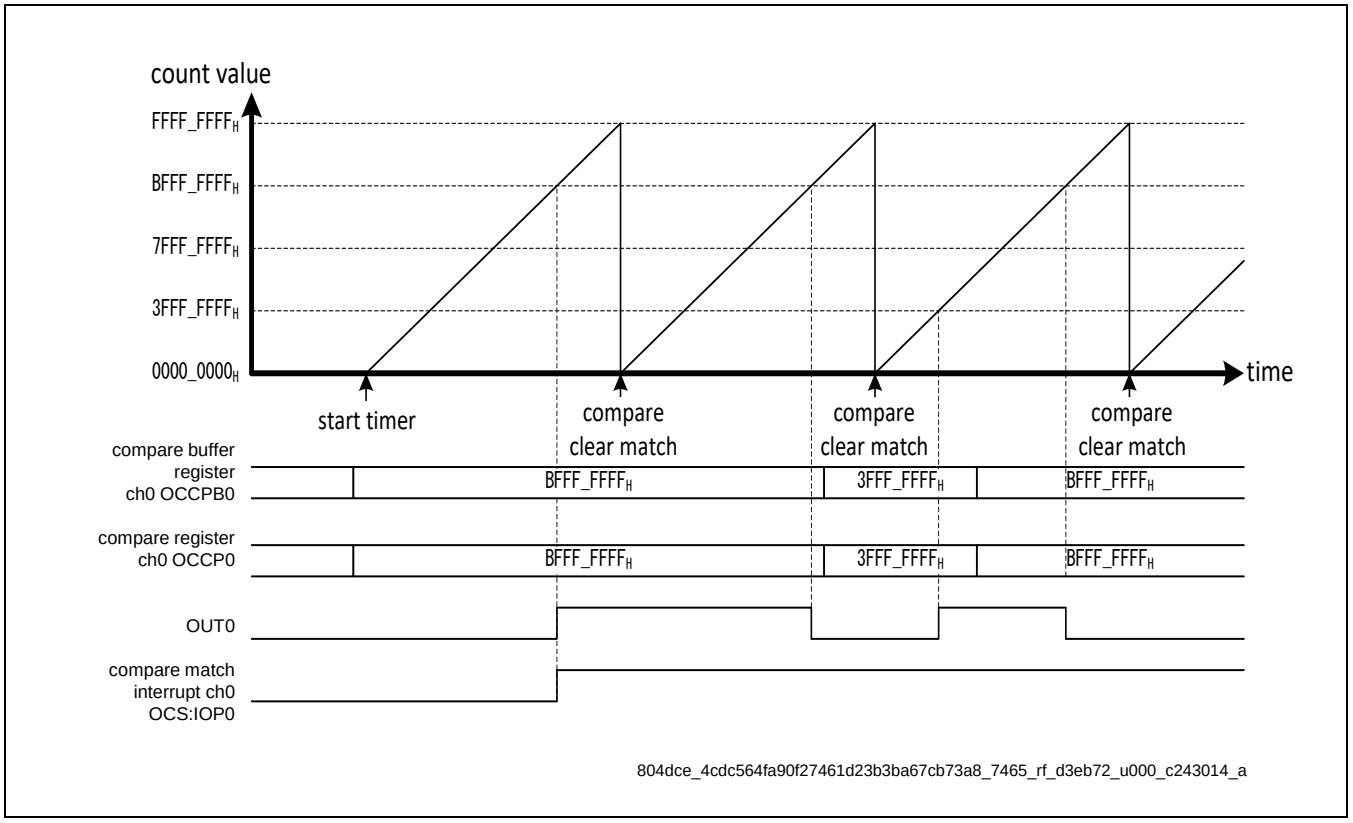


Figure 3-4 Output Waveform Example for When Compare Registers 0 and 1 are Used as a Pair with Initial Output Value = 0 (Free-Run Timer is in Up-Down Count Mode.)



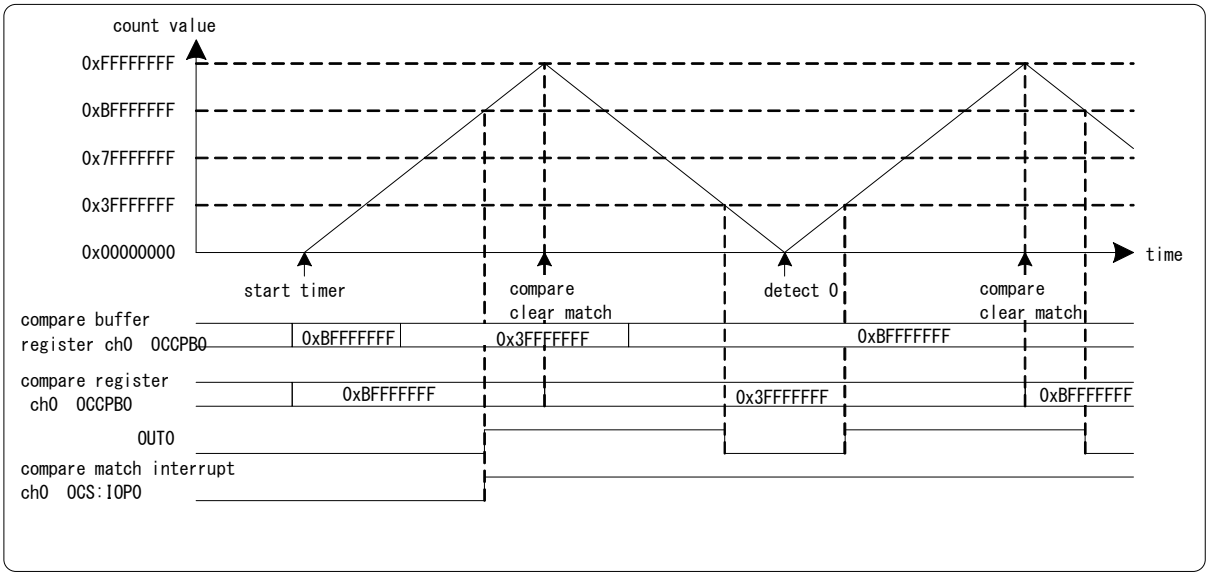
Output Waveform When the Compare Buffer is Disabled

Figure 3-5 Output Waveform Example for When the Compare Buffer is Disabled (Free-Run Timer is in Up Count Mode.)



Output Waveform When the Compare Buffer is Enabled

Figure 3-6 Output Waveform Example for When Compare Buffer Contents are Transferred at Occurrence of Compare Clear Match (Free-Run Timer is in Up-Down Count Mode.)



Output compare operation (for MODn Pin = "H" (Set/Reset Mode))

- When CMOD: bit12 in the compare control register (OCS) is "0"
  - Output compare outputs (OUT0, OUT1) are set to "H" when the count direction of the free-run timer is up count and the free-run timer matches the corresponding compare register (OCCP0, OCCP1).
  - Output compare outputs (OUT0, OUT1) are set to "L" when the count direction of the free-run timer is down count and the free-run timer matches the corresponding compare register (OCCP0, OCCP1).
- When CMOD: bit12 in the compare control register (OCS) is "1"
  - Output compare outputs (OUT0, OUT1) are set to "L" when the count direction of the free-run timer is up count and the free-run timer matches the corresponding compare register (OCCP0, OCCP1).
  - Output compare outputs (OUT0, OUT1) are set to "H" when the count direction of the free-run timer is down count and the free-run timer matches the corresponding compare register (OCCP0, OCCP1).

Figure 3-7 Output Waveform for When the Output Pin is Set to "1" at Occurrence of Output Compare Match (Free-Run Timer is in Up Count Mode.)

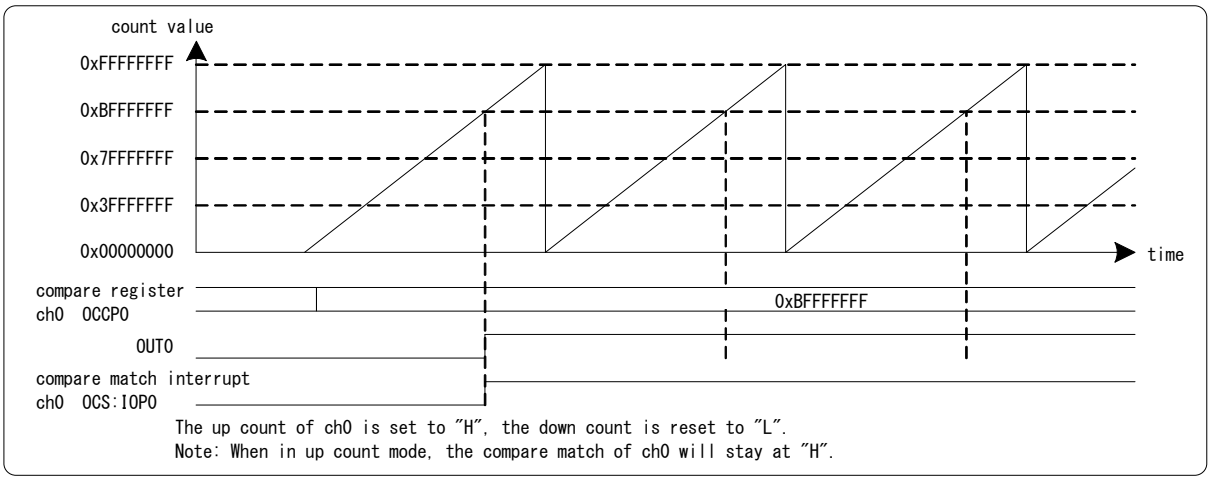
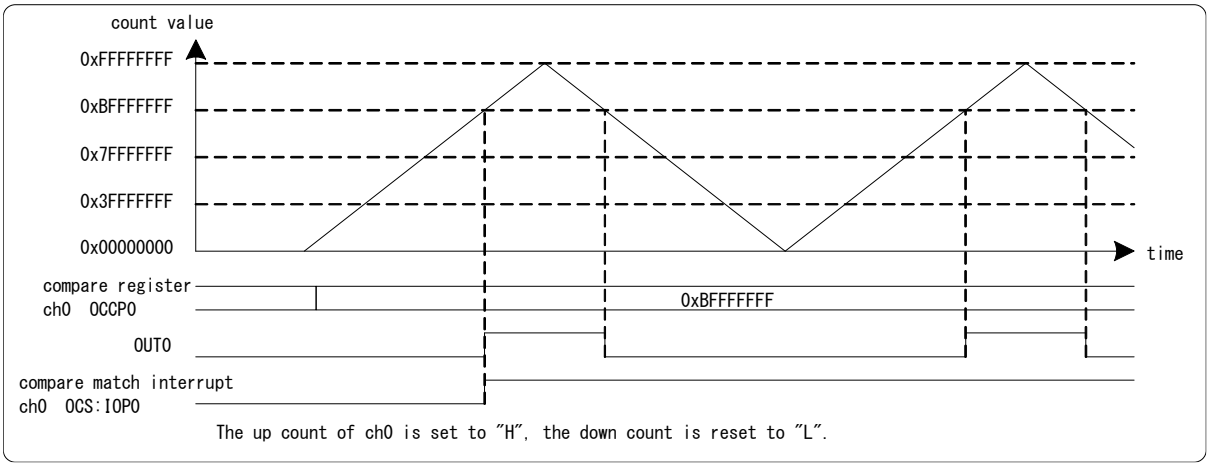


Figure 3-8 Output Waveform for When the Output Pin is Set/Reset at Occurrence of Output Compare Match (Free-Run Timer is in Up-Down Count Mode.)



Output Compare Timing

When the free-run timer value matches the output compare register value, the output compare generates the compare match signal, and controls the OUT pin output according to the mode to generate an interrupt flag.

When a compare match occurs, the OUT pin output is controlled in sync with the count timing of the free-run timer.

Figure 3-9 Timing of Output Compare Register Interrupt

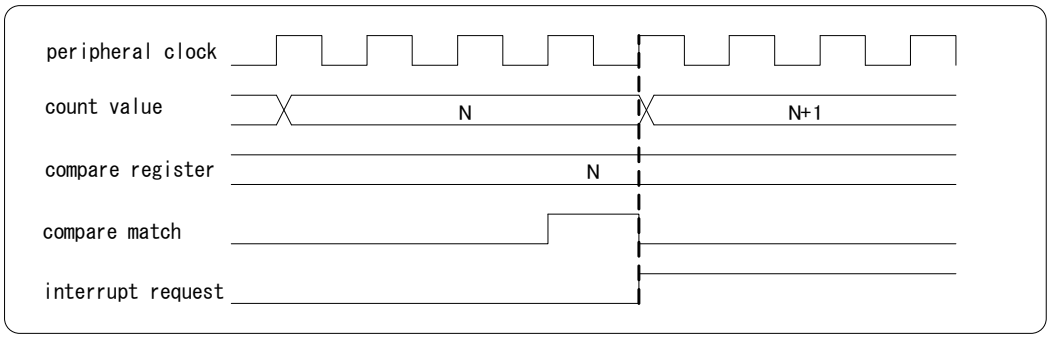
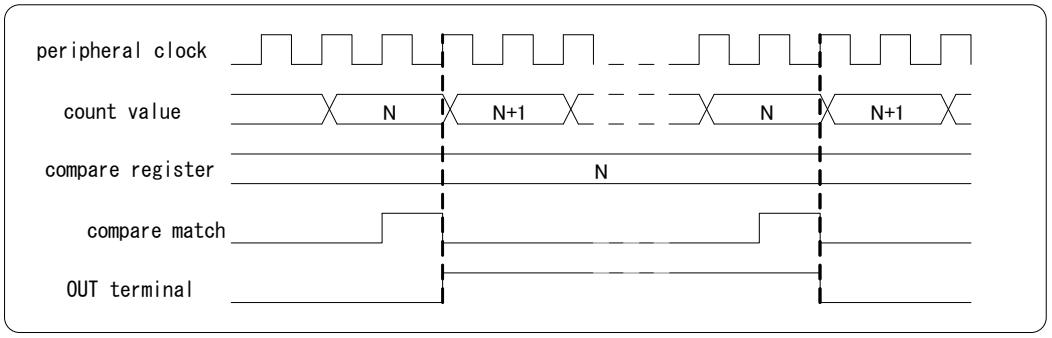


Figure 3-10 Timing of Change in OUT Pin Output



When the compare operation is enabled (OCS.CSTn= 1) or disabled (OCS.CSTn= 0) after free-run timer is stopped or started, the output compare operations are as below.

Free-Run Timer Status	Compare Operation		Compare Match
Stop	Disable		Not occur
	Enable	When the free-run timer value matches the output compare register value	Occur continuously during stop. (only up-count mode)
		When the free-run timer value does not match the output compare register value	Not occur

Free-Run Timer Status	Compare Operation		Compare Match
Run	Disable		Not occur
	Enable	When the free-run timer value matches the output compare register value	Occur
		When the free-run timer value does not match the output compare register value	Not occur

When free-run timer is stopped or started after compare operation is enabled (OCS.CSTn= 1) or disabled (OCS.CSTn= 0), the output compare operations are as below.

Compare Operation Status	Free-Run Timer		Compare Match
Disable	Stop		Not occur
	Start		Not occur
Enable	Stop	When the free-run timer value matches the output compare register value	Occur continuously during stop. (only up-count mode)
		When the free-run timer value does not match the output compare register value	Not occur
	Start	When the free-run timer value matches the output compare register value	Occur (See Note)
		When the free-run timer value does not match the output compare register value	Not occur

Note:

- *When in starting the free-run timer or at the compare match timing in Figure 3-9, the compare match signal occurs.*

If the output compare is up count mode, stop the free-run timer after disabling compare operation.

Please enable compare operation after the free-run timer is cleared. The reason is that the compare match occurs continuously if the compare operation status is enabled under the condition that the free-run timer value matches the output compare register value.

Operation of the Output Compare and Free-Run Timer

Transfer Timing of the Output Compare Buffer Register (for When the Free-Run Timer is in Up Count Mode.)

Figure 3-11 For Transfer When a Compare Clear Match is Found

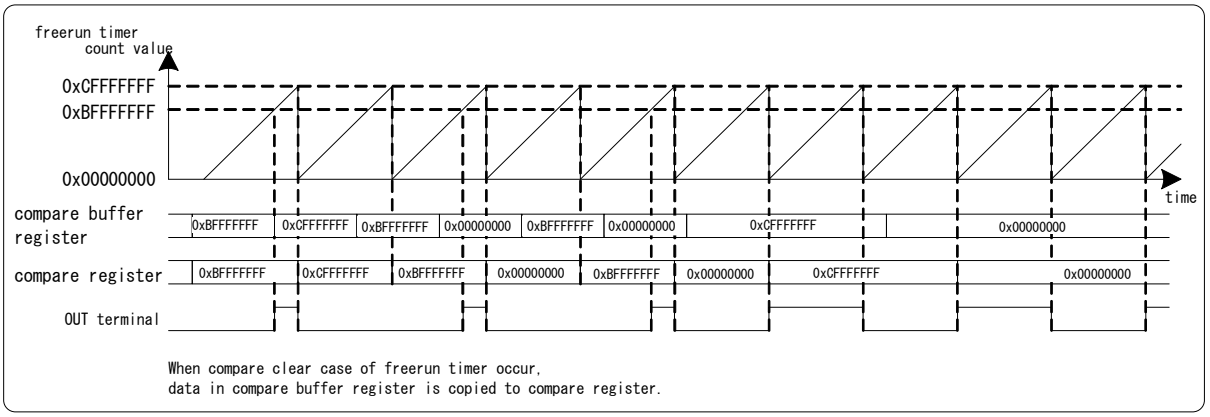
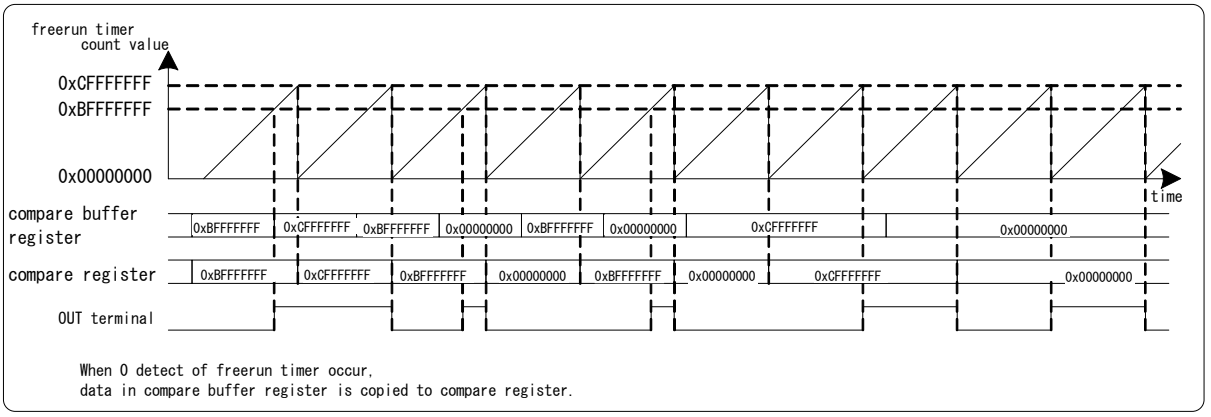


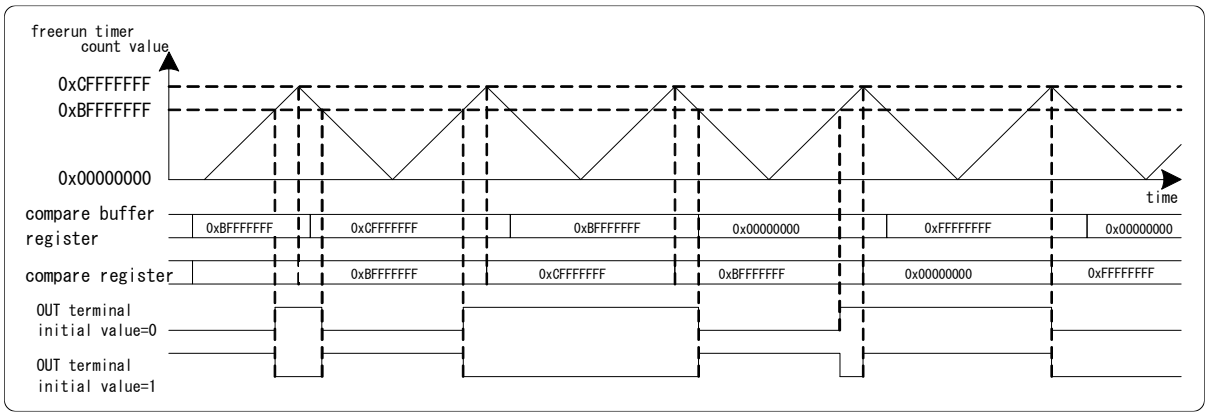
Figure 3-12 For Transfer When 0 is Detected



Output Compare Operation Example 1 (Free-Run Timer Up/Down Counting)

- The compare buffer data for output compare is transferred when a compare clear match is found for the free-run timer.
- In the case where the output is inverted when the output compare output matches (OCS:CMOD = 0, MOD pin = "L")

Figure 3-13 Output Compare Operation Example 1 (Free-Run Timer Up/Down Counting)



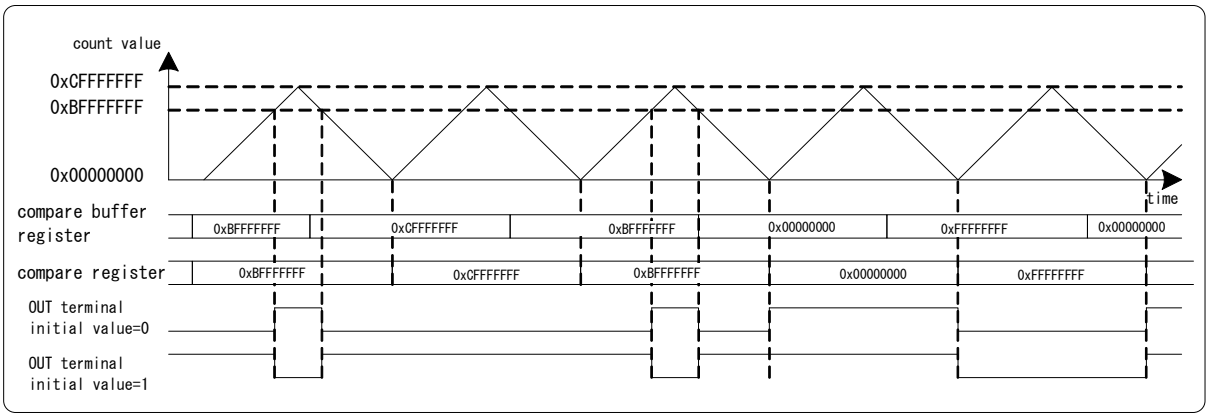
Notes:

- *The OUT pin output is set to "H" when the compare register value is set to 0x00000000, regardless of the free-run timer count value. (It is reset to "L" when OCS:CMOD is "1".)*
- *The OUT pin output is reset to "L" when the compare register value is set to 0xFFFFFFFF, regardless of the free-run timer count value. (It is set to "H" when OCS:CMOD is "1".)*
- *If the compare clear register value of the free-run timer and the compare register value of output compare are the same, no comparison is performed. However, if the initial value of the free-run timer and the compare register value of output compare are the same, a comparison is performed first. At this time, if both the compare clear register value and the compare register value are set to 0xFFFFFFFF, the OUT pin is reset to "L" regardless of the free-run timer count value.*

Output Compare Operation Example 2 (Free-Run Timer Up/Down Counting)

- The compare buffer data for output compare is transferred when 0 is detected in the free-run timer.
- In the case where the output is inverted when the output compare output matches (OCS:CMOD = 0, MOD pin = "L")

Figure 3-14 Output Compare Operation Example 2 (Free-Run Timer Up/Down Counting)



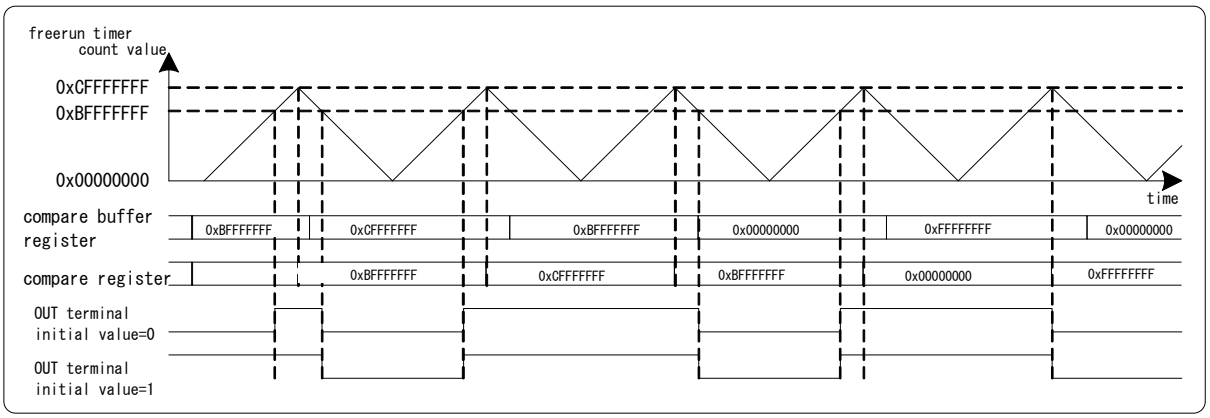
Notes:

- *The OUT pin output is set to "H" when the compare register value is set to 0x00000000, regardless of the free-run timer count value. (It is reset to "L" when OCS:CMOD is "1".)*
- *The OUT pin output is reset to "L" when the compare register value is set to 0xFFFFFFFF, regardless of the free-run timer count value. (It is set to "H" when OCS:CMOD is "1".)*
- *If the compare clear register value of the free-run timer and the compare register value of output compare are the same, no comparison is performed. However, if the initial value of the free-run timer and the compare register value of output compare are the same, a comparison is performed first. At this time, if both the compare clear register value and the compare register value are set to 0xFFFFFFFF, the OUT pin is reset to "L" regardless of the free-run timer count value.*

Output Compare Operation Example 3 (Free-Run Timer Up/Down Counting)

- The compare buffer data for output compare is transferred when a compare clear match is found for the free-run timer.
- In the case where the output compare output is set to "H" at a match in up-counting, and reset to "L" at a match in down-counting (OCS:CMOD = 0, MOD pin = "H")

Figure 3-15 Output Compare Operation Example 3 (Free-Run Timer Up/Down Counting)



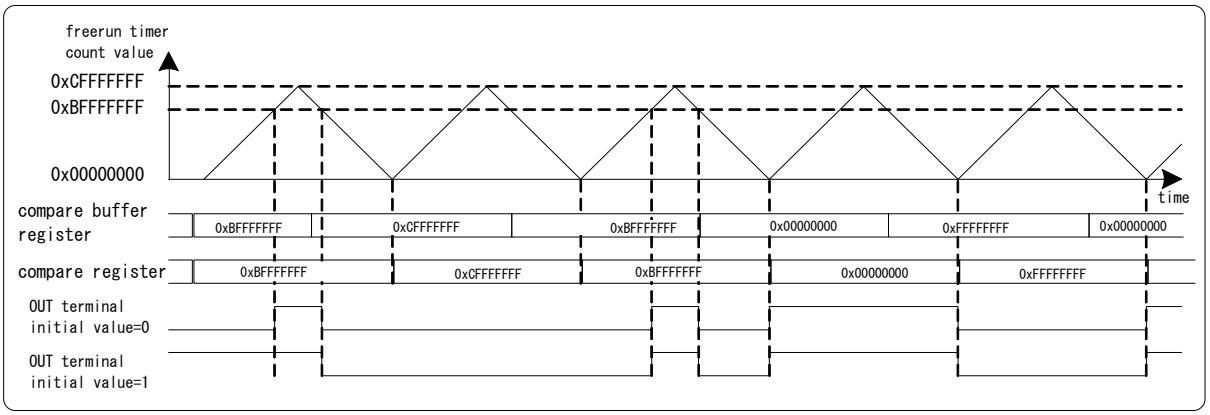
Notes:

- *The OUT pin output is set to "H" when the compare register value is set to 0x00000000, regardless of the free-run timer count value. (It is reset to "L" when OCS:CMOD is "1".)*
- *The OUT pin output is reset to "L" when the compare register value is set to 0xFFFFFFFF, regardless of the free-run timer count value. (It is set to "H" when OCS:CMOD is "1".)*
- *If the compare clear register value of the free-run timer and the compare register value of output compare are the same, no comparison is performed. However, if the initial value of the free-run timer and the compare register value of output compare are the same, a comparison is performed first. At this time, if both the compare clear register value and the compare register value are set to 0xFFFFFFFF, the OUT pin is reset to "L" regardless of the free-run timer count value.*

Output Compare Operation Example 4 (Free-Run Timer Up/Down Counting)

- The compare buffer data for output compare is transferred when 0 is detected in the free-run timer.
- In the case where the output compare output is set to "H" at a match in up-counting, and reset to "L" at a match in down-counting (OCS:CMOD = 0, MOD pin = "H")

Figure 3-16 Output Compare Operation Example 4 (Free-Run Timer Up/Down Counting)



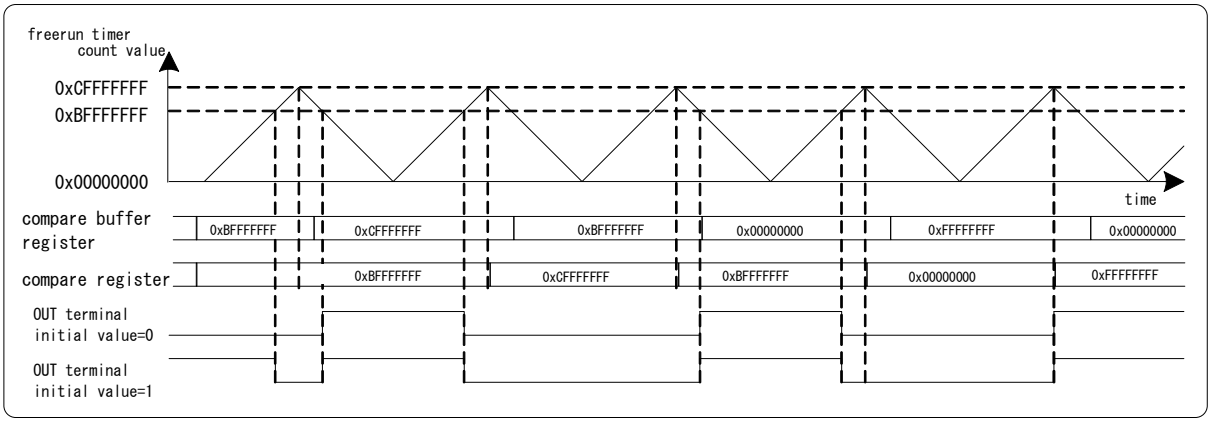
Notes:

- *The OUT pin output is set to "H" when the compare register value is set to 0x00000000, regardless of the free-run timer count value. (It is reset to "L" when OCS:CMOD is "1".)*
- *The OUT pin output is reset to "L" when the compare register value is set to 0xFFFFFFFF, regardless of the free-run timer count value. (It is set to "H" when OCS:CMOD is "1".)*
- *If the compare clear register value of the free-run timer and the compare register value of output compare are the same, no comparison is performed. However, if the initial value of the free-run timer and the compare register value of output compare are the same, a comparison is performed first. At this time, if both the compare clear register value and the compare register value are set to 0xFFFFFFFF, the OUT pin is reset to "L" regardless of the free-run timer count value.*

Output Compare Operation Example 5 (Free-Run Timer Up/Down Counting)

- The compare buffer data for output compare is transferred when a compare clear match is found for the free-run timer.
- In the case where the output compare output is reset to "L" at a match in up-counting, and set to "H" at a match in down-counting (OCS:CMOD = 1, MOD pin = "H")

Figure 3-17 Output Compare Operation Example 5 (Free-Run Timer Up/Down Counting)



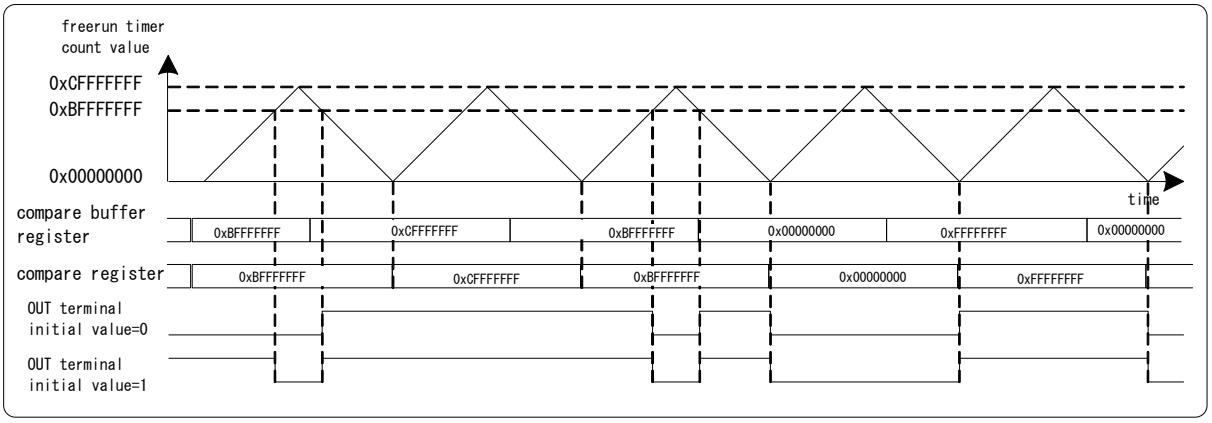
Notes:

- *The OUT pin output is reset to "L" when the compare register value is set to 0x00000000, regardless of the free-run timer count value. (It is set to "H" when OCS:CMOD is "0".)*
- *The OUT pin output is set to "H" when the compare register value is set to 0xFFFFFFFF, regardless of the free-run timer count value. (It is reset to "L" when OCS:CMOD is "0".)*
- *If the compare clear register value of the free-run timer and the compare register value of output compare are the same, no comparison is performed. However, if the initial value of the free-run timer and the compare register value of output compare are the same, a comparison is performed first. At this time, if both the compare clear register value and the compare register value are set to 0xFFFFFFFF, the OUT pin is reset to "L" regardless of the free-run timer count value.*

Output Compare Operation Example 6 (Free-Run Timer Up/Down Counting)

- The compare buffer data for output compare is transferred when 0 is detected in the compare timer of the free-run timer.
- In the case where the output compare output is reset to "L" at a match in up-counting, and set to "H" at a match in down-counting (OCS:CMOD = 1, MOD pin = "H")

Figure 3-18 Output Compare Operation Example 6 (Free-Run Timer Up/Down Counting)

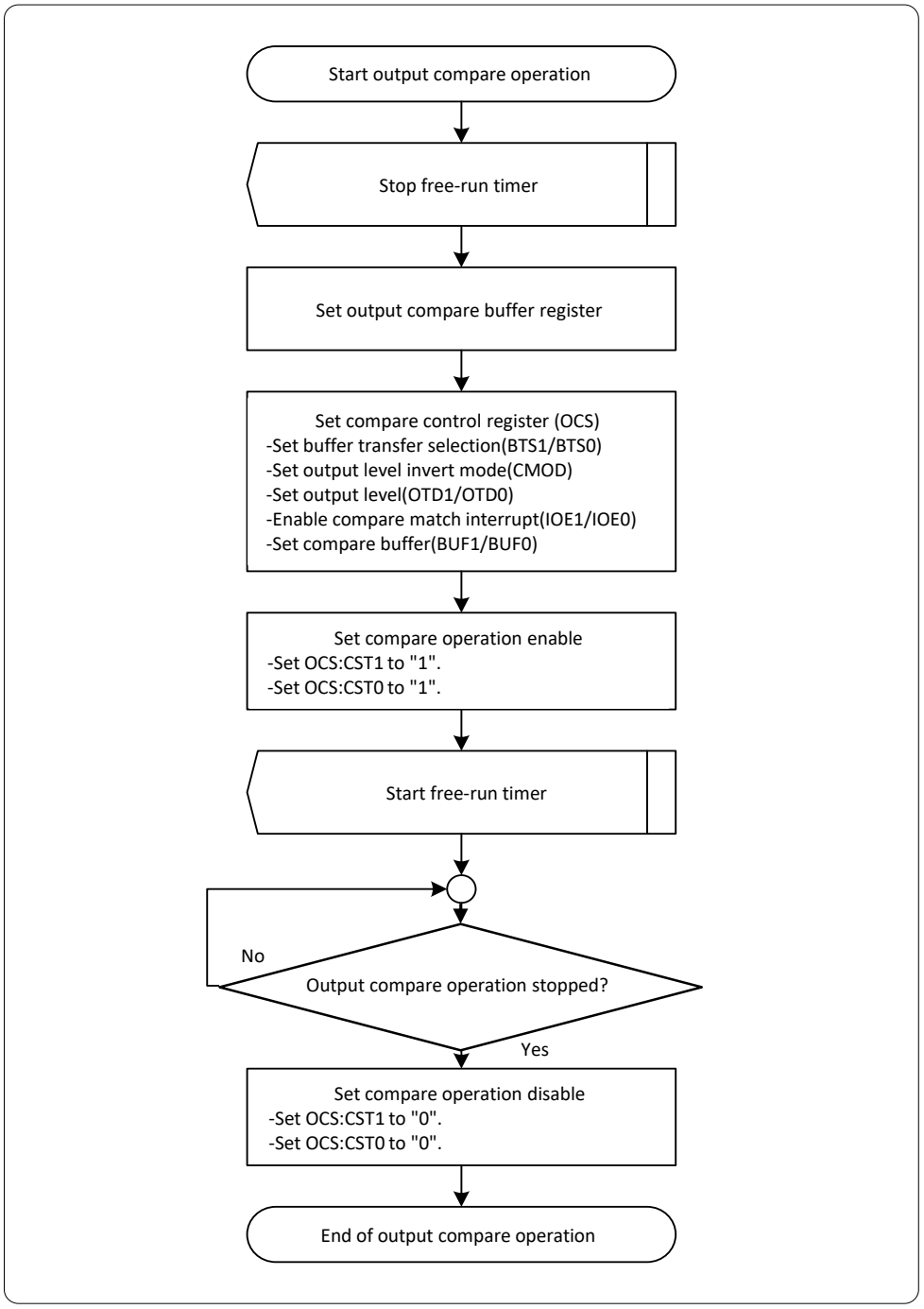


Notes:

- *The OUT pin output is reset to "L" when the compare register value is set to 0x00000000, regardless of the free-run timer count value. (It is set to "H" when OCS:CMOD is "0".)*
- *The OUT pin output is set to "H" when the compare register value is set to 0xFFFFFFFF, regardless of the free-run timer count value. (It is reset to "L" when OCS:CMOD is "0".)*
- *If the compare clear register value of the free-run timer and the compare register value of output compare are the same, no comparison is performed. However, if the initial value of the free-run timer and the compare register value of output compare are the same, a comparison is performed first. At this time, if both the compare clear register value and the compare register value are set to 0xFFFFFFFF, the OUT pin is reset to "L" regardless of the free-run timer count value.*

4. Setting Procedure Example

Figure 4-1 Procedure Example of Setting Output Capture Operation



5. Registers

This section provides the register list of the 32-bit output compare.

Table 5-1 Register Map

Common Peripheral Group #0 (Channel No.:0, 1, 2, 3, 4, 5, 6, and 7)

Offset	Register Name / Initial Value				Block Name
	+3	+2	+1	+0	
0000_0000	OCUxx_OCCPB0/OCUxx_OCCP0 00000000_00000000_00000000_00000000				Common Peripheral #0
0000_0004	OCUxx_OCCPB1/OCUxx_OCCP1 00000000_00000000_00000000_00000000				
0000_0008	OCUxx_OCS 11111111_11111111_11100000_00001100				
0000_000C	OCUxx_OCSC 00000000_00000000_00000000_00000000				
0000_0010	OCUxx_OCSS 00000000_00000000_00000000_00000000				
0000_0014 | 0000_03FC					

Note:

- *xx is the channel number (0 to 7).*

Common Peripheral Group #1 (Channel No.:8, 9, 10, 11, 12, 13, 14, and 15)

Offset	Register Name / Initial Value				Block Name
	+3	+2	+1	+0	
0000_0000	OCUxx_OCCPB0/OCUxx_OCCP0 00000000_00000000_00000000_00000000				Common Peripheral #1
0000_0004	OCUxx_OCCPB1/OCUxx_OCCP1 00000000_00000000_00000000_00000000				
0000_0008	OCUxx_OCS 11111111_11111111_11100000_00001100				
0000_000C	OCUxx_OCSC 00000000_00000000_00000000_00000000				
0000_0010	OCUxx_OCSS 00000000_00000000_00000000_00000000				
0000_0014 | 0000_03FC					

Note:

- *xx is the channel number (8 to 15).*

Table 5-2 List of Register Mirror Area

Address of Register Mirror Area	Mirrored Peripheral	Mirror Area Behavior	PPU
0000_0020 to 0000_003F 0000_0040 to 0000_005F ..... 0000_03E0 to 0000_03FF	OCU pair ch.0	Accessing this region results to the access to OCU pair ch.0 register area with following offset. Offset: (addr & 0000_001F)	This area is covered by PPU #240 as well as the mirrored peripheral.
0000_0420 to 0000_043F 0000_0440 to 0000_045F ..... 0000_07E0 to 0000_07FF	OCU pair ch.1	Accessing this region results to the access to OCU pair ch.1 register area with following offset. Offset: (addr & 0000_001F)	This area is covered by PPU #241 as well as the mirrored peripheral.
0000_0820 to 0000_083F 0000_0840 to 0000_085F ..... 0000_0BE0 to 0000_0BFF	OCU pair ch.2	Accessing this region results to the access to OCU pair ch.2 register area with following offset. Offset: (addr & 0000_001F)	This area is covered by PPU #242 as well as the mirrored peripheral.
0000_0C20 to 0000_0C3F 0000_0C40 to 0000_0C5F ..... 0000_0FE0 to 0000_0FFF	OCU pair ch.3	Accessing this region results to the access to OCU pair ch.3 register area with following offset. Offset: (addr & 0000_001F)	This area is covered by PPU #243 as well as the mirrored peripheral.
0000_1020 to 0000_103F 0000_1040 to 0000_105F ..... 0000_13E0 to 0000_13FF	OCU pair ch.4	Accessing this region results to the access to OCU pair ch.4 register area with following offset. Offset: (addr & 0000_001F)	This area is covered by PPU #244 as well as the mirrored peripheral.
0000_1420 to 0000_143F 0000_1440 to 0000_145F ..... 0000_17E0 to 0000_17FF	OCU pair ch.5	Accessing this region results to the access to OCU pair ch.5 register area with following offset. Offset: (addr & 0000_001F)	This area is covered by PPU #245 as well as the mirrored peripheral.
0000_1820 to 0000_183F 0000_1840 to 0000_185F ..... 0000_1BE0 to 0000_1BFF	OCU pair ch.6	Accessing this region results to the access to OCU pair ch.6 register area with following offset. Offset: (addr & 0000_001F)	This area is covered by PPU #246 as well as the mirrored peripheral.
0000_1C20 to 0000_1C3F 0000_1C40 to 0000_1C5F ..... 0000_1FE0 to 0000_1FFF	OCU pair ch.7	Accessing this region results to the access to OCU pair ch.7 register area with following offset. Offset: (addr & 0000_001F)	This area is covered by PPU #247 as well as the mirrored peripheral.

Address of Register Mirror Area	Mirrored Peripheral	Mirror Area Behavior	PPU
0000_0020 to 0000_003F 0000_0040 to 0000_004F ..... 0000_03E0 to 0000_03FF	OCU pair ch.8	Accessing this region results to the access to OCU pair ch.8 register area with following offset. Offset: (addr & 0000_001F)	This area is covered by PPU #248 as well as the mirrored peripheral.
0000_0420 to 0000_043F 0000_0440 to 0000_043F ..... 0000_07E0 to 0000_07FF	OCU pair ch.9	Accessing this region results to the access to OCU pair ch.9 register area with following offset. Offset: (addr & 0000_001F)	This area is covered by PPU #249 as well as the mirrored peripheral.
0000_0820 to 0000_083F 0000_0840 to 0000_085F ..... 0000_0BE0 to 0000_0BFF	OCU pair ch.10	Accessing this region results to the access to OCU pair ch.10 register area with following offset. Offset: (addr & 0000_001F)	This area is covered by PPU #250 as well as the mirrored peripheral.
0000_0C20 to 0000_0C3F 0000_0C40 to 0000_0C5F ..... 0000_0FE0 to 0000_0FFF	OCU pair ch.11	Accessing this region results to the access to OCU pair ch.11 register area with following offset. Offset: (addr & 0000_001F)	This area is covered by PPU #251 as well as the mirrored peripheral.
0000_1020 to 0000_103F 0000_1040 to 0000_105F ..... 0000_13E0 to 0000_13FF	OCU pair ch.12	Accessing this region results to the access to OCU pair ch.12 register area with following offset. Offset: (addr & 0000_001F)	This area is covered by PPU #252 as well as the mirrored peripheral.
0000_1420 to 0000_143F 0000_1440 to 0000_145F ..... 0000_17E0 to 0000_17FF	OCU pair ch.13	Accessing this region results to the access to OCU pair ch.13 register area with following offset. Offset: (addr & 0000_001F)	This area is covered by PPU #253 as well as the mirrored peripheral.
0000_1820 to 0000_183F 0000_1840 to 0000_185F ..... 0000_1BE0 to 0000_1BFF	OCU pair ch.14	Accessing this region results to the access to OCU pair ch.614 register area with following offset. Offset: (addr & 0000_001F)	This area is covered by PPU #254 as well as the mirrored peripheral.
0000_1C20 to 0000_1C3F 0000_1C40 to 0000_1C5F ..... 0000_1FE0 to 0000_1FFF	OCU pair ch.15	Accessing this region results to the access to OCU pair ch.15 register area with following offset. Offset: (addr & 0000_001F)	This area is covered by PPU #255 as well as the mirrored peripheral.

Register List of 32-Bit Output Compare

Table 5-1 Register List of 32-Bit Output Compare

Abbreviated Register Name	Register Name	Reference
OCCPB0, 1/ OCCP0, 1	Output compare buffer register 0, 1/ Output compare register 0, 1	5.1
OCS	Compare control register	5.2
OCSC	Compare control clear register	5.3
OCSS	Compare control set register	5.4

Register Bit Locations of 32-Bit Output Compare

	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
OCCPB0/ OCCP0	OP31	OP30	OP29	OP28	OP27	OP26	OP25	OP24
	OP23	OP22	OP21	OP20	OP19	OP18	OP17	OP16
	OP15	OP14	OP13	OP12	OP11	OP10	OP9	OP8
	OP7	OP6	OP5	OP4	OP3	OP2	OP1	OP0
OCCPB1/ OCCP1	OP31	OP30	OP29	OP28	OP27	OP26	OP25	OP24
	OP23	OP22	OP21	OP20	OP19	OP18	OP17	OP16
	OP15	OP14	OP13	OP12	OP11	OP10	OP9	OP8
	OP7	OP6	OP5	OP4	OP3	OP2	OP1	OP0
OCS	Reserved							
	Reserved							
	Reserved	BTS1	BTS0	CMOD	Reserved		OTD1	OTD0
	IOP1	IOP0	IOE1	IOE0	BUF1	BUF0	CST1	CST0
OCSC	Reserved							
	Reserved							
	Reserved	BTS1C	BTS0C	CMODC	Reserved		OTD1C	OTD0C
	IOP1C	IOP0C	IOE1C	IOE0C	BUF1C	BUF0C	CST1C	CST0C
OCSS	Reserved							
	Reserved							
	Reserved	BTS1S	BTS0S	CMODS	Reserved		OTD1S	OTD0S
	Reserved		IOE1S	IOE0S	BUF1S	BUF0S	CST1S	CST0S

5.1. Output Compare Buffer Register (OCCPB0, 1)/Output Compare Register (OCCP0, 1)

The output compare buffer register (OCCPB) is the buffer register for the output compare register (OCCP). The output compare buffer register (OCCPB) and the output compare register (OCCP) exist at the same address.

Output Compare Buffer Register (OCCPB0, OCCPB1)

bit	31	30	29	28	27	26	25	24
Field	OP31	OP30	OP29	OP28	OP27	OP26	OP25	OP24
Attribute	W	W	W	W	W	W	W	W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	OP23	OP22	OP21	OP20	OP19	OP18	OP17	OP16
Attribute	W	W	W	W	W	W	W	W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	OP15	OP14	OP13	OP12	OP11	OP10	OP09	OP08
Attribute	W	W	W	W	W	W	W	W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	OP07	OP06	OP05	OP04	OP03	OP02	OP01	OP00
Attribute	W	W	W	W	W	W	W	W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

[bit31:0] OP31 to OP00: Output Compare Buffer Value Bits

- The output compare buffer register (OCCPB) is the buffer register for the output compare register (OCCP).
- The value in the output compare buffer register (OCCPB) is immediately transferred to the output compare register (OCCP) on the following condition: The buffer function is disabled (BUF1, BUF0: bit3, bit2 in the compare control register (OCS) = 0b11) or the free-run timer stops.
- When the buffer function is enabled (BUF1, BUF0: bit3, bit2 in the compare control register (OCS) = 0b00), the value in the output compare buffer register (OCCPB) is transferred to the output compare register (OCCP) on the following condition: a compare clear match or detection of 0 according to the transfer selection (BTS1, BTS0: bit14, bit13) in the compare control register (OCS).

Note:

- *For access to the OCCPB0 and OCCPB1 registers, use word access instructions.*

Output Compare Register (OCCP0, OCCP1)

bit	31	30	29	28	27	26	25	24
Field	OP31	OP30	OP29	OP28	OP27	OP26	OP25	OP24
Attribute	R	R	R	R	R	R	R	R
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	OP23	OP22	OP21	OP20	OP19	OP18	OP17	OP16
Attribute	R	R	R	R	R	R	R	R
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	OP15	OP14	OP13	OP12	OP11	OP10	OP09	OP08
Attribute	R	R	R	R	R	R	R	R
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	OP07	OP06	OP05	OP04	OP03	OP02	OP01	OP00
Attribute	R	R	R	R	R	R	R	R
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

[bit31:0] OP31 to OP00: Output Compare Value Bits

- The output compare register (OCCP) is used for comparison of its value with the count value of the free-run timer. Before enabling free-run timer operation, set a value in the output compare buffer register (OCCPB).
- When the value in output compare register (OCCP) matches the count value in the free-run timer, the compare signal is generated. Then, the output compare interrupt flag (IOP1, IOP0: bit7, bit6 in the compare control register (OCS)) is set to "1".

Notes:

- *For access to the OCCP0 and OCCP1 registers, use word access instructions.*
- *If the free-run timer is operating in up-down count mode and OCS:CMOD is 0, when the compare register value is set to 0x00000000, the OUT pin output is set to "H" regardless of the free-run timer count value. (It is reset to "L" when OCS:CMOD is "1".) Also, when the compare register value is set to 0xFFFFFFFF and OCS:CMOD is 0, the OUT pin output is reset to "L" regardless of the free-run timer count value. (It is set to "H" when OCS:CMOD is "1".)*
- *If the compare clear register value of the free-run timer and the compare register value of output compare are the same, no comparison is performed. However, if the initial value of the free-run timer and the compare register value of output compare are the same, a comparison is performed first. At this time, if both the compare clear register value and the compare register*

value are set to 0xFFFFFFFF, the OUT pin is reset to "L" regardless of the free-run timer count value.

*: In the explanation of the free-run timer, the operating state of the free-run timer connected to the output compare is described.

5.2. Compare Control Register (OCS)

The compare control register (OCS) controls the compare output (OUT pin) level, output enable, output level invert mode, compare operation enable, compare match interrupt enable, and compare match interrupt flag.
For details on writing to this register, see "6 Precautions for Using This Device."

Compare Control Register (OCS)

bit	31	30	29	28	27	26	25	24
Field	Reserved							
Attribute	R1,W1							
Protection Attribute	-							
Initial value	11111111							

bit	23	22	21	20	19	18	17	16
Field	Reserved							
Attribute	R1,W1							
Protection Attribute	-							
Initial Value	11111111							

bit	15	14	13	12	11	10	9	8
Field	Reserved	BTS1	BTS0	CMOD	Reserved		OTD1	OTD0
Attribute	R1,W1	R/W	R/W	R/W	R0,W0		R/W	R/W
Protection Attribute	-	-	-	-	-		-	-
Initial Value	1	1	1	0	00		0	0

bit	7	6	5	4	3	2	1	0
Field	IOP1	IOP0	IOE1	IOE0	BUF1	BUF0	CST1	CST0
Attribute	R,WX	R,WX	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	1	1	0	0

[bit31:16] Reserved: Reserved Bit

- When writing, always write "1".
- When reading, "1" is always read.

[bit15] Reserved: Reserved Bit

- When writing, always write "1".
- When reading, "1" is always read.

[bit14] BTS1: Buffer Transfer Selection Bit (CH.1)

Bit	Description
0	Start output compare (CH.1) transfer when 0 is detected.
1	Start output compare (CH.1) transfer when a compare clear match occurs.

- The BTS1 bit is used to select the timing of data transfer from output compare buffer register 1 (OCCPB1) to output compare register 1 (OCCP1).
- When the BTS1 bit is set to "0":
  Data transfer starts when 0 is detected in the free-run timer.
- When the BTS1 bit is set to "1":
  Data transfer starts when a compare clear match occurs in the free-run timer.
- The BTS1 bit is cleared to "0" by writing "1" to the BTS1C bit in the OCSC register.
- The BTS1 bit is set to "1" by writing "1" to the BTS1S bit in the OCSS register.

[bit13] BTS0: Buffer Transfer Selection Bit (CH.0)

Bit	Description
0	Start output compare (CH.0) transfer when 0 is detected.
1	Start output compare (CH.0) transfer when a compare clear match occurs.

- The BTS0 bit is used to select the timing of data transfer from output compare buffer register 0 (OCCPB0) to output compare register 0 (OCCP0).
- When the BTS0 bit is set to "0":
  Data transfer starts when 0 is detected in the free-run timer.
- When the BTS0 bit is set to "1":
  Data transfer starts when a compare clear match occurs in the free-run timer.
- The BTS0 bit is cleared to "0" by writing "1" to the BTS0C bit in the OCSC register.
- The BTS0 bit is set to "1" by writing "1" to the BTS0S bit in the OCSS register.

[bit12] CMOD: Output Level Invert Mode Bit

Bit	Description		
	MODn Pin* = "L" (Invert Mode)		MODn Pin* = "H" (Set/Reset Mode)
	OUT0	OUT1	OUT0, OUT1
0	The OUT0 level is immediately inverted when a match with compare register 0 occurs.	The OUT1 level is immediately inverted when a match with compare register 1 occurs.	Set to "H" for a match in up count mode. Set to "L" for a match in down count mode.
1	(Same as above)	The OUT1 level is immediately inverted when a match with compare register 0 or 1 occurs.	Set to "L" for a match in up count mode. Set to "H" for a match in down count mode.

*: The MODn (n = 0, 1) pin signal is an external input signal to control the OUTn output pin signal.

- The CMOD bit is used to switch the output level of the OUT pin when a match occurs.
- When "0" is set in the CMOD bit:
  - For MODn pin (n = 0, 1) = "L"

- The output level of the OUT0 pin is immediately inverted when the free-run timer and compare register 0 match.
- The output level of the OUT1 pin is immediately inverted when the free-run timer and compare register 1 match.

- For MODn pin* (n = 0, 1) = "H"
  - The output level of the OUTx pin is set to "H" when the free-run timer and compare register x match in up count mode. (x = 0, 1)
  - The output level of the OUTx pin is reset to "L" when the free-run timer and compare register x match in down count mode. (x = 0, 1)

- When "1" is set in the CMOD bit:
  - For MODn pin (n = 0, 1) = "L"
    - The output level of the OUT0 pin is immediately inverted when the free-run timer and compare register 0 match.
    - The output level of the OUT1 pin is immediately inverted when the free-run timer and compare register 0 or 1 match.
      (If compare register 0 and 1 contain the same value, the behavior is the same as when 1 compare register is used.)
  - For MODn pin* (n = 0, 1) = "H"
    - The output level of the OUTx pin is reset to "L" when the free-run timer and compare register x match in up count mode. (x = 0, 1)
    - The output level of the OUTx pin is set to "H" when the free-run timer and compare register x match in down count mode. (x = 0, 1)
- The CMOD bit is cleared to "0" by writing "1" to the CMODC bit in the OCSC register.
- The CMOD bit is set to "1" by writing "1" to the CMODS bit in the OCSS register.

[bit11:10] Reserved: Reserved Bits

- When writing, always write "0".
- When reading, "0" is always read.

[bit9] OTD1: OUT1 Output Level Bit

Bit	Description	
	During Read Operation	During Write Operation
0	Output value of OUT1	Output "L" to OUT1.
1		Output "H" to OUT1.

- The OTD1 bit is used for setting an OUT1 pin output value of output compare 1.
- The initial value of OUT1 pin output is "L".
- When writing a value, be sure to stop the compare operation beforehand (compare operation enable bit CST1: bit1 = 0).
- The OTD1 bit is writable when the compare operation is stopped (compare operation enable bit CST1: bit1 = 0).
- The value read from the OTD1 bit indicates the OUT1 pin output value of output compare 1.
- The OTD1 bit is cleared to "0" by writing "1" to the OTD1C bit in the OCSC register when the compare operation is stopped (compare operation enable bit CST1: bit1 = 0).

- The OTD1 bit is set to "1" by writing "1" to the OTD1S bit in the OCSS register when the compare operation is stopped (compare operation enable bit CST1: bit1 = 1).

[bit8] OTD0: OUT0 Output Level Bit

Bit	Description	
	During Read Operation	During Write Operation
0	Output value of OUT0	Output "L" to OUT0.
1		Output "H" to OUT0.

- The OTD0 bit is used for setting an OUT0 pin output value of output compare 0.
- The initial value of OUT0 pin output is "L".
- When writing a value, be sure to stop the compare operation beforehand (compare operation enable bit CST0: bit0 = 0).
- The OTD0 bit is writable when the compare operation is stopped (compare operation enable bit CST0: bit0 = 0).
- The value read from the OTD0 bit indicates the OUT0 pin output value of output compare 0.
- The OTD0 bit is cleared to "0" by writing "1" to the OTD0C bit in the OCSC register when the compare operation is stopped (compare operation enable bit CST0: bit0 = 0).
- The OTD0 bit is set to "1" by writing "1" to the OTD0S bit in the OCSS register when the compare operation is stopped (compare operation enable bit CST0: bit0 = 0).

[bit7] IOP1: Compare Match Interrupt Flag Bit (CH.1)

Bit	Description
0	The compare match interrupt for compare register 1 does not occur.
1	The compare match interrupt for compare register 1 occurs.

- The IOP1 bit is an interrupt flag indicating that compare register 1 has matched the free-run timer value.
- The IOP1 bit is set to "1" when the compare register value matches the free-run timer value.
- When the IOP1 bit is set with the compare match interrupt enable bit (IOE1: bit5) set to "1", the output compare interrupt is generated.
- The IOP1 bit is cleared to "0" by writing "1" to the IOP1C bit in the OCSC register.

[bit6] IOP0: Compare Match Interrupt Flag Bit (CH.0)

Bit	Description
0	The compare match interrupt for compare register 0 does not occur.
1	The compare match interrupt for compare register 0 occurs.

- The IOP0 is an interrupt flag indicating that compare register 0 has matched the free-run timer value.
- The IOP0 bit is set to "1" when the compare register value matches the free-run timer value.
- When the IOP0 bit is set with the compare match interrupt enable bit (IOE0: bit4) set to "1", the output compare interrupt is generated.
- The IOP0 bit is cleared to "0" by writing "1" to the IOP0C bit in the OCSC register.

[bit5] IOE1: Compare Match Interrupt Enable Bit (CH.1)

Bit	Description
0	Disable the compare match interrupt for compare register 1.
1	Enable the compare match interrupt for compare register 1.

- The IOE1 bit is used to enable the output compare interrupt of compare register 1.
- When the compare match interrupt flag bit (IOP1: bit7) is set to "1" with the IOE1 bit set to "1", the output compare 1 interrupt is generated.
- The IOE1 bit is cleared to "0" by writing "1" to the IOE1C bit in the OCSC register.
- The IOE1 bit is set to "1" by writing "1" to the IOE1S bit in the OCSS register.

[bit4] IOE0: Compare Match Interrupt Enable Bit (CH.0)

Bit	Description
0	Disable the compare match interrupt for compare register 0.
1	Enable the compare match interrupt for compare register 0.

- The IOE0 bit is used to enable the output compare interrupt of compare register 0.
- When the compare match interrupt flag bit (IOP0: bit6) is set to "1" with the IOE0 bit set to "1", the output compare 0 interrupt is generated.
- The IOE0 bit is cleared to "0" by writing "1" to the IOE0C bit in the OCSC register.
- The IOE0 bit is set to "1" by writing "1" to the IOE0S bit in the OCSS register.

[bit3] BUF1: Compare Buffer Disable Bit (CH.1)

Bit	Description
0	Enable the compare buffer for compare register 1.
1	Disable the compare buffer for compare register 1.

- The BUF1 bit is used to disable the buffer function of output compare register 1.
- When the BUF1 bit is set to "0", the buffer function is enabled.
- When the BUF1 bit is set to "1", the buffer function is disabled.
- The BUF1 bit is cleared to "0" by writing "1" to the BUF1C bit in the OCSC register.
- The BUF1 bit is set to "1" by writing "1" to the BUF1S bit in the OCSS register.

[bit2] BUF0: Compare Buffer Disable Bit (CH.0)

Bit	Description
0	Enable the compare buffer for compare register 0.
1	Disable the compare buffer for compare register 0.

- The BUF0 bit is used to disable the buffer function of output compare register 0.
- When the BUF0 bit is set to "0", the buffer function is enabled.
- When the BUF0 bit is set to "1", the buffer function is disabled.
- The BUF0 bit is cleared to "0" by writing "1" to the BUF0C bit in the OCSC register.
- The BUF0 bit is set to "1" by writing "1" to the BUF0S bit in the OCSS register.

[bit1] CST1: Compare Operation Enable Bit (CH.1)

Bit	Description
0	Disable the compare operation for compare register 1.
1	Enable the compare operation for compare register 1.

- The CST1 bit is used to enable the compare operation between the free-run timer and compare register 1.
- When enabling the compare operation, be sure to write values in compare buffer register 1 (OCCPB1) and the timer data register of the free-run timer (TCDT) beforehand.
- The CST1 bit is cleared to "0" by writing "1" to the CST1C bit in the OCSC register.
- The CST1 bit is set to "1" by writing "1" to the CST1S bit in the OCSS register.

[bit0] CST0: Compare Operation Enable Bit (CH.0)

Bit	Description
0	Disable the compare operation for compare register 0.
1	Enable the compare operation for compare register 0.

- The CST0 bit is used to enable the compare operation between the free-run timer and compare register 0.
- When enabling the compare operation, be sure to write values in compare buffer register 0 (OCCPB0) and the timer data register of the free-run timer (TCDT) beforehand.
- The CST0 bit is cleared to "0" by writing "1" to the CST0C bit in the OCSC register.
- The CST0 bit is set to "1" by writing "1" to the CST0S bit in the OCSS register.

5.3. Compare Control Clear Register (OCSC)

The compare control clear register (OCSC) is a register to clear bits in the compare control register (OCS).

Compare Control Clear Register (OCSC)

bit	31	30	29	28	27	26	25	24
Field	Reserved							
Attribute	R0,W0							
Protection Attribute	-							
Initial value	00000000							

bit	23	22	21	20	19	18	17	16
Field	Reserved							
Attribute	R0,W0							
Protection Attribute	-							
Initial Value	00000000							

bit	15	14	13	12	11	10	9	8
Field	Reserved	BTS1C	BTS0C	CMODC	Reserved		OTD1C	OTD0C
Attribute	R0,W0	R0,W	R0,W	R0,W	R0,W0		R0,W	R0,W
Protection Attribute	-	-	-	-	-		-	-
Initial Value	0	0	0	0	00		0	0

bit	7	6	5	4	3	2	1	0
Field	IOP1C	IOP0C	IOE1C	IOE0C	BUF1C	BUF0C	CST1C	CST0C
Attribute	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

[bit31:16] Reserved: Reserved Bit

- When writing, always write "0".
- When reading, "0" is always read.

[bit15] Reserved: Reserved Bit

- When writing, always write "0".
- When reading, "0" is always read.

[bit14] BTS1C: BTS1 Clear Bit

Bit	Description
0	Do not affect the BTS1C bit and the buffer transfer selection bit (CH.1) (BTS1) in the compare control register (OCS).
1	Clear the buffer transfer selection bit (CH.1) (BTS1) in the compare control register (OCS).

- When "1" is written, the buffer transfer selection bit (CH.1) (BTS1) in the compare control register (OCS) is cleared to "0".
- "0" is always read from the BTS1C bit.

[bit13] BTS0C: BTS0 Clear Bit

Bit	Description
0	Do not affect the BTS0C bit and the buffer transfer selection bit (CH.0) (BTS0) in the compare control register (OCS).
1	Clear the buffer transfer selection bit (CH.0) (BTS0) in the compare control register (OCS).

- When "1" is written, the buffer transfer selection bit (CH.0) (BTS0) in the compare control register (OCS) is cleared to "0".
- "0" is always read from the BTS0C bit.

[bit12] CMODC: CMOD Clear Bit

Bit	Description
0	Do not affect the CMODC bit and the output level invert mode bit (CMOD) in the compare control register (OCS).
1	Clear the output level invert mode bit (CMOD) in the compare control register (OCS).

- When "1" is written, the output level invert mode bit (CMOD) in the compare control register (OCS) is cleared to "0".
- "0" is always read from the CMODC bit.

[bit11:10] Reserved: Reserved bits

- When writing, always write "0".
- When reading, "0" is always read.

[bit9] OTD1C: OTD1 Clear Bit

Bit	Description
0	Do not affect the OTD1C bit and the OUT1 output level bit (OTD1) in the compare control register (OCS).
1	Clear the OUT1 output level bit (OTD1) in the compare control register (OCS).

- The OUT1 output level bit (OTD1) in the compare control register (OCS) can be cleared when the compare operation is stopped (the compare operation enable bit CST1: bit1 in the compare control register (OCS) = 0).
- When "1" is written, the OUT1 output level bit (OTD1) in the compare control register (OCS) is cleared to "0".
- "0" is always read from the OTD1C bit.

[bit8] OTD0C: OTD0 Clear Bit

Bit	Description
0	Do not affect the OTD0C bit and the OUT0 output level bit (OTD0) in the compare control register (OCS).
1	Clear the OUT0 output level bit (OTD0) in the compare control register (OCS).

- The OUT0 output level bit (OTD0) in the compare control register (OCS) can be cleared when the compare operation is stopped (compare operation enable bit CST0: bit0 in the compare control register (OCS) = 0).
- When "1" is written, the OUT0 output level bit (OTD0) in the compare control register (OCS) is cleared to "0".
- "0" is always read from the OTD0C bit.

[bit7] IOP1C: IOP1 Clear Bit

Bit	Description
0	Do not affect the IOP1C bit and the compare match interrupt flag bit (CH.1) (IOP1) in the compare control register (OCS).
1	Clear the compare match interrupt flag bit (CH.1) (IOP1) in the compare control register (OCS).

- When "1" is written, the compare match interrupt flag bit (CH.1) (IOP1) in the compare control register (OCS) is cleared to "0".
- "0" is always read from the IOP1C bit.

[bit6] IOP0C: IOP0 Clear Bit

Bit	Description
0	Do not affect the IOP0C bit and the compare match interrupt flag bit (CH.0) (IOP0) in the compare control register (OCS).
1	Clear the compare match interrupt flag bit (CH.0) (IOP0) in the compare control register (OCS).

- When "1" is written, the compare match interrupt flag bit (CH.0) (IOP0) in the compare control register (OCS) is cleared to "0".
- "0" is always read from the IOP0C bit.

[bit5] IOE1C: IOE1 Clear Bit

Bit	Description
0	Do not affect the IOE1C bit and the compare match interrupt enable bit (CH.1) (IOE1) in the compare control register (OCS).
1	Clear the compare match interrupt enable bit (CH.1) (IOE1) in the compare control register (OCS).

- When "1" is written, the compare match interrupt enable bit (CH.1) (IOE1) in the compare control register (OCS) is cleared to "0".
- "0" is always read from the IOE1C bit.

[bit4] IOE0C: IOE0 Clear Bit

Bit	Description
0	Do not affect the IOE0C bit and the compare match interrupt enable bit (CH.0) (IOE0) in the compare control register (OCS).
1	Clear the compare match interrupt enable bit (CH.0) (IOE0) in the compare control register (OCS).

- When "1" is written, the compare match interrupt enable bit (CH.0) (IOE0) in the compare control register (OCS) is cleared to "0".
- "0" is always read from the IOE0C bit.

[bit3] BUF1C: BUF1 Clear Bit

Bit	Description
0	Do not affect the BUF1C bit and the compare buffer disable bit (CH.1) (BUF1) in the compare control register (OCS).
1	Clear the compare buffer disable bit (CH.1) (BUF1) in the compare control register (OCS).

- When "1" is written, the compare buffer disable bit (CH.1) (BUF1) in the compare control register (OCS) is cleared to "0".
- "0" is always read from the BUF1C bit.

[bit2] BUF0C: BUF0 Clear Bit

Bit	Description
0	Do not affect the BUF0C bit and the compare buffer disable bit (CH.0) (BUF0) in the compare control register (OCS).
1	Clear the compare buffer disable bit (CH.0) (BUF0) in the compare control register (OCS).

- When "1" is written, the compare buffer disable bit (CH.0) (BUF0) in the compare control register (OCS) is cleared to "0".
- "0" is always read from the BUF0C bit.

[bit1] CST1C: CST1 Clear Bit

Bit	Description
0	Do not affect the CST1C bit and the compare operation enable bit (CH.1) (CST1) in the compare control register (OCS).
1	Clear the compare operation enable bit (CH.1) (CST1) in the compare control register (OCS).

- When "1" is written, the compare operation enable bit (CH.1) (CST1) in the compare control register (OCS) is cleared to "0".
- "0" is always read from the CST1C bit.

[bit0] CST0C: CST0 Clear Bit

Bit	Description
0	Do not affect the CST0C bit and the compare operation enable bit (CH.0) (CST0) in the compare control register (OCS).
1	Clear the compare operation enable bit (CH.0) (CST0) in the compare control register (OCS).

- When "1" is written, the compare operation enable bit (CH.0) (CST0) in the compare control register (OCS) is cleared to "0".
- "0" is always read from the CST0C bit.

5.4. Compare Control Set Register (OCSS)

The compare control set register (OCSS) is a register to set bits in the compare control register (OCS).

Compare Control Register Set Register (OCSS)

bit	31	30	29	28	27	26	25	24
Field	Reserved							
Attribute	R0,W0							
Protection Attribute	-							
Initial value	00000000							

bit	23	22	21	20	19	18	17	16
Field	Reserved							
Attribute	R0,W0							
Protection Attribute	-							
Initial Value	00000000							

bit	15	14	13	12	11	10	9	8
Field	Reserved	BTS1S	BTS0S	CMODS	Reserved		OTD1S	OTD0S
Attribute	R0,W0	R0,W	R0,W	R0,W	R0,W0		R0,W	R0,W
Protection Attribute	-	-	-	-	-		-	-
Initial Value	0	0	0	0	00		0	0

bit	7	6	5	4	3	2	1	0
Field	Reserved		IOE1S	IOE0S	BUF1S	BUF0S	CST1S	CST0S
Attribute	R0,W0		R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
Protection Attribute	-		-	-	-	-	-	-
Initial Value	00		0	0	0	0	0	0

[bit31:16] Reserved: Reserved Bit

- When writing, always write "0".
- When reading, "0" is always read.

[bit15] Reserved: Reserved Bit

- When writing, always write "0".
- When reading, "0" is always read.

[bit14] BTS1S: BTS1 Set Bit

Bit	Description
0	Do not affect the BTS1S bit and the buffer transfer selection bit (CH.1) (BTS1) in the compare control register (OCS).
1	Set the buffer transfer selection bit (CH.1) (BTS1) in the compare control register (OCS).

- When "1" is written, the buffer transfer selection bit (CH.1) (BTS1) in the compare control register (OCS) is set to "1".
- "0" is always read from the BTS1S bit.

[bit13] BTS0S: BTS0 Set Bit

Bit	Description
0	Do not affect the BTS0S bit and the buffer transfer selection bit (CH.0) (BTS0) in the compare control register (OCS).
1	Set the buffer transfer selection bit (CH.0) (BTS0) in the compare control register (OCS).

- When "1" is written, the buffer transfer selection bit (CH.0) (BTS0) in the compare control register (OCS) is set to "1".
- "0" is always read from the BTS0S bit.

[bit12] CMODS: CMOD Set Bit

Bit	Description
0	Do not affect the CMODS bit and the output level invert mode bit (CMOD) in the compare control register (OCS).
1	Set the output level invert mode bit (CMOD) in the compare control register (OCS).

- When "1" is written, the output level invert mode bit (CMOD) in the compare control register (OCS) is set to "1".
- "0" is always read from the CMODS bit.

[bit11:10] Reserved: Reserved Bits

- When writing, always write "0".
- When reading, "0" is always read.

[bit9] OTD1S: OTD1 Set Bit

Bit	Description
0	Do not affect the OTD1S bit and the OUT1 output level bit (OTD1) in the compare control register (OCS).
1	Set the OUT1 output level bit (OTD1) in the compare control register (OCS).

- The OUT1 output level bit (OTD1) in the compare control register (OCS) can be set when the compare operation is stopped (the compare operation enable bit CST1: bit1 in the compare control register (OCS) = 0).
- When "1" is written, the OUT1 output level bit (OTD1) in the compare control register (OCS) is set to "1".
- "0" is always read from the OTD1S bit.

[bit8] OTD0S: OTD0 Set Bit

Bit	Description
0	Do not affect the OTD0S bit and the OUT0 output level bit (OTD0) in the compare control register (OCS).
1	Set the OUT0 output level bit (OTD0) in the compare control register (OCS).

- The OUT0 output level bit (OTD0) in the compare control register (OCS) can be set when the compare operation is stopped (the compare operation enable bit CST0: bit0 in the compare control register (OCS) = 0).
- When "1" is written, the OUT0 output level bit (OTD0) in the compare control register (OCS) is set to "1".
- "0" is always read from the OTD0S bit.

[bit7:6] Reserved: Reserved Bits

- When writing, always write "0".
- When reading, "0" is always read.

[bit5] IOE1S: IOE1 Set Bit

Bit	Description
0	Do not affect the IOE1S bit and the compare match interrupt enable bit (CH.1) (IOE1) in the compare control register (OCS).
1	Set the compare match interrupt enable bit (CH.1) (IOE1) in the compare control register (OCS).

- When "1" is written, the compare match interrupt enable bit (CH.1) (IOE1) in the compare control register (OCS) is set to "1".
- "0" is always read from the IOE1S bit.

[bit4] IOE0S: IOE0 Set Bit

Bit	Description
0	Do not affect the IOE0S bit and the compare match interrupt enable bit (CH.0) (IOE0) in the compare control register (OCS).
1	Set the compare match interrupt enable bit (CH.0) (IOE0) in the compare control register (OCS).

- When "1" is written, the compare match interrupt enable bit (CH.0) (IOE0) in the compare control register (OCS) is set to "1".
- "0" is always read from the IOE0S bit.

[bit3] BUF1S: BUF1 Set Bit

Bit	Description
0	Do not affect the BUF1S bit and the compare buffer disable bit (CH.1) (BUF1) in the compare control register (OCS).
1	Set the compare buffer disable bit (CH.1) (BUF1) in the compare control register (OCS).

- When "1" is written, the compare buffer disable bit (CH.1) (BUF1) in the compare control register (OCS) is set to "1".
- "0" is always read from the BUF1S bit.

[bit2] BUF0S: BUF0 Set Bit

Bit	Description
0	Do not affect the BUF0S bit and the compare buffer disable bit (CH.0) (BUF0) in the compare control register (OCS).
1	Set the compare buffer disable bit (CH.0) (BUF0) in the compare control register (OCS).

- When "1" is written, the compare buffer disable bit (CH.0) (BUF0) in the compare control register (OCS) is set to "1".
- "0" is always read from the BUF0S bit.

[bit1] CST1S: CST1 Set Bit

Bit	Description
0	Do not affect the CST1S bit and the compare operation enable bit (CH.1) (CST1) in the compare control register (OCS).
1	Set the compare operation enable bit (CH.1) (CST1) in the compare control register (OCS).

- When "1" is written, the compare operation enable bit (CH.1) (CST1) in the compare control register (OCS) is set to "1".
- "0" is always read from the CST1S bit.

[bit0] CST0S: CST0 Set Bit

Bit	Description
0	Do not affect the CST0S bit and the compare operation enable bit (CH.0) (CST0) in the compare control register (OCS).
1	Set the compare operation enable bit (CH.0) (CST0) in the compare control register (OCS).

- When "1" is written, the compare operation enable bit (CH.0) (CST0) in the compare control register (OCS) is set to "1".
- "0" is always read from the CST0S bit.

6. Precautions for Using This Device

The following are precautions for using the 32-bit output compare.

Notes on Accessing the Output Compare Buffer Register (OCCPB) and Output Compare Register (OCCP)

- For access to the output compare buffer register (OCCPB) and output compare register (OCCP), use word access instructions.
- Also, do not use read-modify-write (RMW) instructions for access.

Notes on Accessing the Compare Control Register (OCS)

- This register supports writing from the bit-band alias area. For the bit-band alias area, see the chapter of "BIT-BAND UNIT."
- To clear a specified bit of this register, clear the bit by writing "1" to the applicable bit in the compare control clear register (OCSC).
- To set a specified bit of this register, set the bit by writing "1" to the applicable bit in the compare control set register (OCSS).
- Data can be written directly to this register only when writing to all bits.

Precautions for Using the Output Compare

Compare Output (OUT Pin) Level

- When the value in the output compare register (OCCP) matches the count value in the free-run timer, the compare signal is generated. Then, the output compare interrupt flag bit (IOP1, IOP0: bit7, bit6 in the compare control register OCS) is set to "1". However, if the peak value of the free-run timer matches the value in the output compare register (OCCP) when the free-run timer is in up-down count mode, the compare signal will not be generated.
- When the compare signal is generated, the output of the OUT pins (OUT0, OUT1) corresponding to output compare registers 0 and 1 is controlled by the output level invert mode bit (CMOD: bit12 in the compare control register OCS) and external input signal (MODn). Then, the output is set to the inverted level or "H/L" level, accordingly.

Compare Output (OUT Pin) When the Free-Run Timer is in Up-Down Count Mode

- When the output level invert mode bit (CMOD) of the compare control register (OCS) is 0
  When 0xFFFFFFFF is set in the output compare register (OCCP), the OUT pin outputs "L" regardless of the free-run timer value or invert mode. If 0x00000000 is set, it outputs "H".
- When the output level invert mode bit (CMOD) of the compare control register (OCS) is 1
  When 0xFFFFFFFF is set in the output compare register, the OUT pin outputs "H" regardless of the free-run timer value or invert mode. If 0x00000000 is set, it outputs "L".

CHAPTER 45: 32-bit Reload Timer



This chapter explains 32-bit Reload Timer

1. Overview
2. Configuration and Block Diagram
3. Operation of the 32-Bit Reload Timer
4. Registers

RLT-TXXPT03P01R01L02-E1-XX

1. Overview

This section gives a brief overview of the 32-bit Reload Timer.

Features of 32-Bit Reload Timer

The 32-bit Reload Timer consists of a 32-bit down counter, a 32-bit Reload register, one input(TIN), one output (TOT), and control registers. The 32-bit Reload Timer has the following features:

- External and internal clock/event source
- Trigger signal programmable as rising/falling edge or both
- Gated count function
- One-shot or reload counter mode
- Counter state can be made visible at external pin
- Prescaler with six different settings for the internal clock and two different settings for the external clock
- Several Reload Timers can be cascaded to form a longer Reload Timer
- Support for MCU debug mode

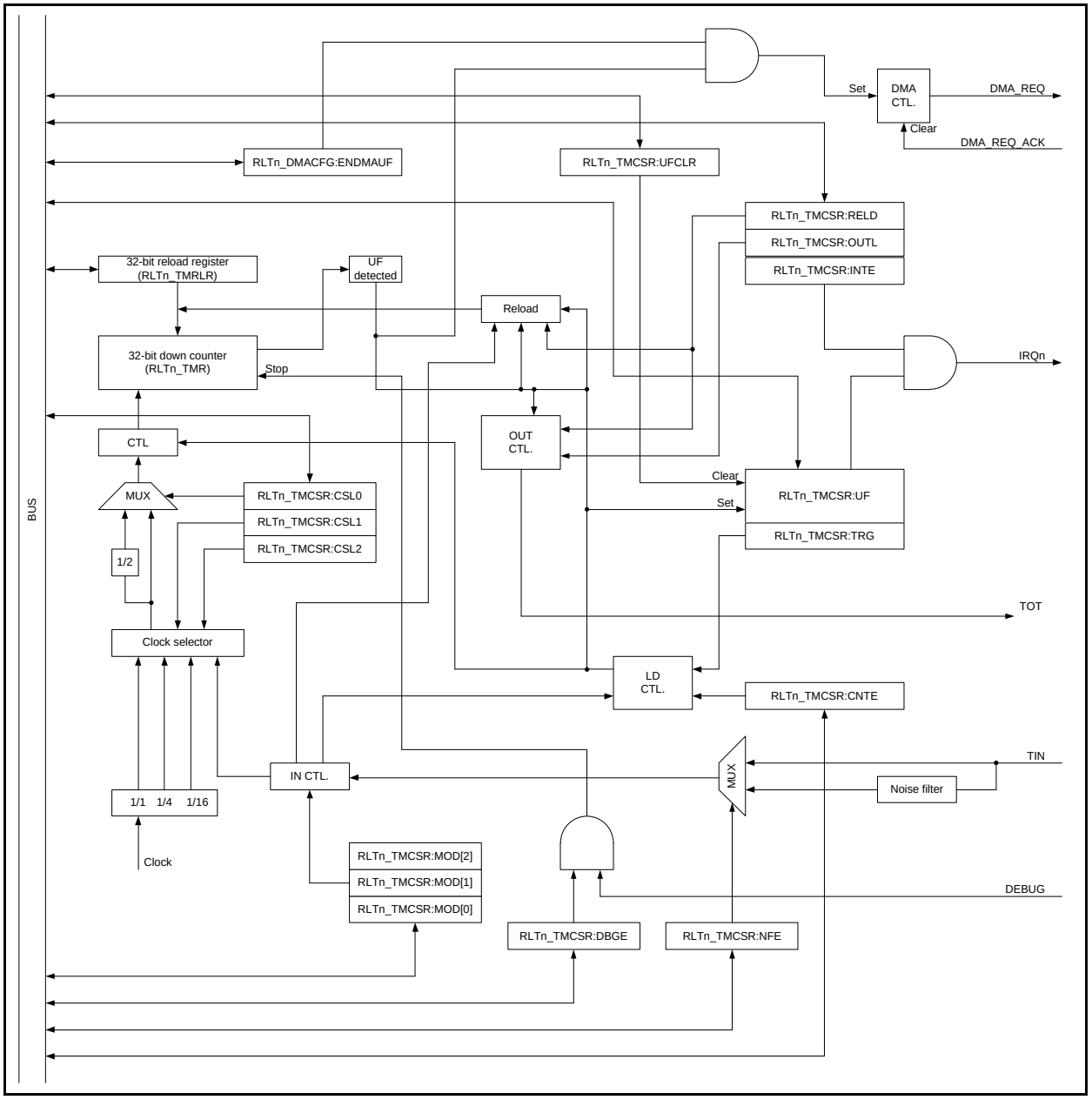
DMA and Interrupts

The 32-bit Reload Timer can generate DMA request which can be used to start DMA transfers.
The 32-bit Reload Timer can generate interrupt in case of underflow.

2. Configuration and Block Diagram

This section shows a block diagram of the 32-bit Reload Timer.

Figure 2-1 Block Diagram of 32-Bit Reload Timer



3. Operation of the 32-Bit Reload Timer

This section describes the operation of the 32-bit Reload Timer.

3.1 Internal Clock and External Event Counter Operations of 32-Bit Reload Timer
3.2 Underflow Operation of 32-Bit Reload Timer
3.3 Output Functions of 32-Bit Reload Timer
3.4 Counter Operation State
3.5 DMA Operation

3.1. Internal Clock and External Event Counter Operations of 32-Bit Reload Timer

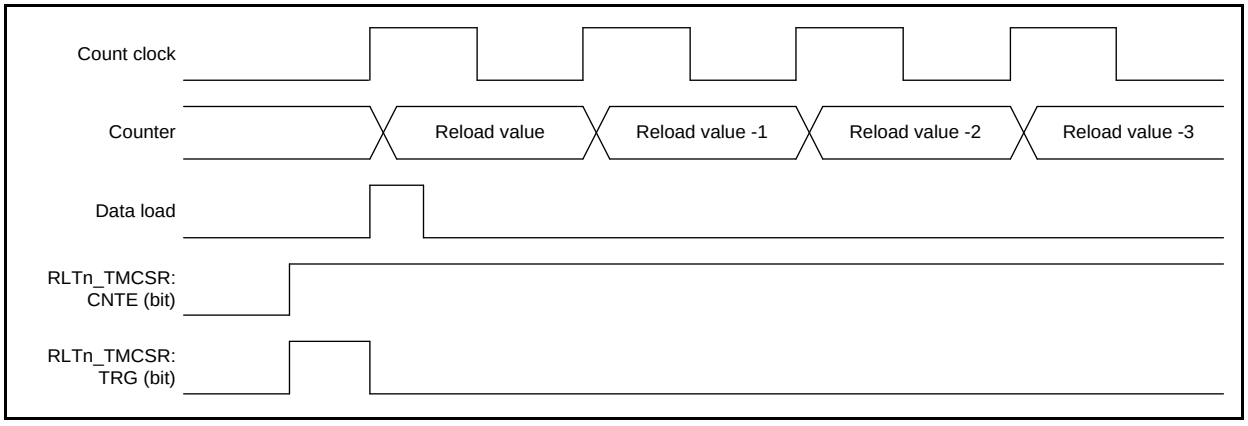
In internal clock mode, the peripheral clock with different divider settings can be selected as the clock source for operating the Reload Timer. The external input TIN can be selected as either a trigger input, or as a gate input by a register setting.
In event counter mode, the TIN is used as an external event input. Each active edge on this input (rising, falling, or both) decrements the counter.
When RLTn_TMCSR:CSL0 = 1, then each second event is counted.

Internal Clock Operation of 32-Bit Reload Timer

Writing "1" to both RLTn_TMCSR:CNTE and RLTn_TMCSR:TRG bits enables and starts counting at the same time. Using the RLTn_TMCSR:TRG bit as a trigger input is always available when the timer is enabled (RLTn_TMCSR:CNTE = 1), regardless of the operation mode.

Figure 3-1 shows counter activation and counter operation.

Figure 3-1 Activation and Operation of 32-Bit Reload Timer Counter



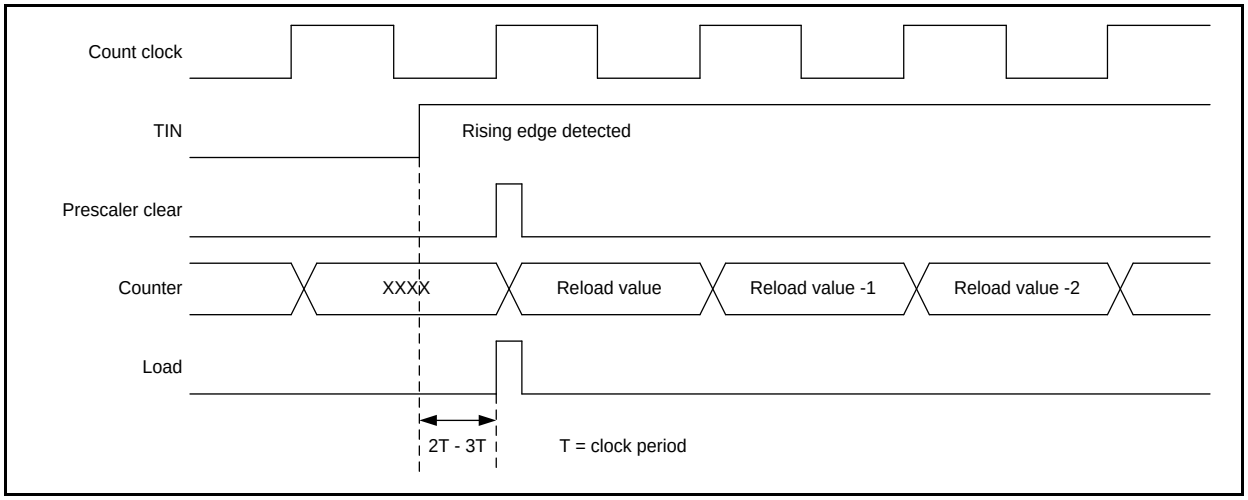
Input Functions of 32-Bit Reload Timer (In Internal Clock Mode)

The TIN input can be used as either a trigger input, or as a gate input, when an internal clock is selected as the clock source.

Trigger Input

When used as a trigger input, an active edge causes the timer to load the reload register contents and resets the internal prescaler. Then, count operation starts. For the minimum pulse width length of TIN refer to device specific datasheet.

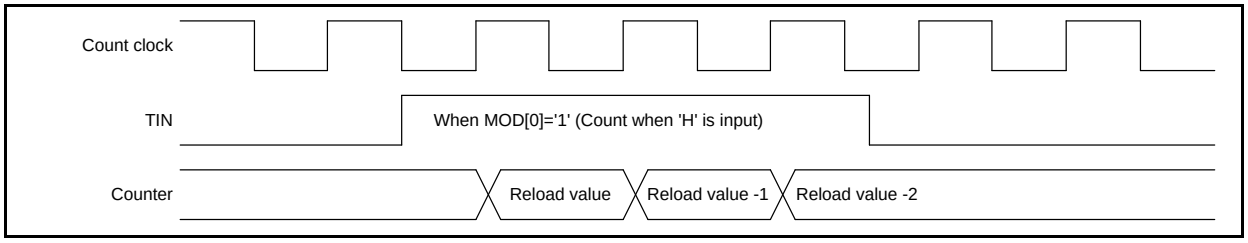
Figure 3-2 Trigger Input Operation of 32-Bit Reload Timer



Gate Input

When used as a gate input, the counter only counts while the active level specified by the RLTn_TMCSR:MOD[0] bit is input to the TIN. In this case, the count clock continues to operate unless stopped. The software trigger can be used in gate mode, regardless of the gate level. For the minimum pulse width length of TIN refer to device specific datasheet.

Figure 3-3 Gate Input Operation of 32-Bit Reload Timer



External Event Counter

When external event count mode is selected, the TIN is used as an external event input. The counter counts on the active edge specified in the RLTn_TMCSR. For the minimum pulse width length of TIN refer to device specific datasheet.

3.2. Underflow Operation of 32-Bit Reload Timer

An underflow is defined for this timer as the time when the counter value changes from 0x00000000 to 0xFFFFFFFF or reload occurs (RLTn_TMCSR:RELD= 1). Therefore, an underflow occurs after (reload register setting + 1) counts.

Underflow Operation of 32-Bit Reload Timer

If the RLTn_TMCSR:RELD bit is "1" and an underflow occurs, the contents of the reload register is loaded into the counter and counting continues.

If the RLTn_TMCSR:RELD bit is "0", counting stops when counter reaches 0xFFFFFFFF.

The RLTn_TMCSR:UF bit is set when the underflow occurs. If the RLTn_TMCSR:INTE bit is "1" at this time, an interrupt request is generated.

Figure 3-4 shows the operation when an underflow occurs with various values of RLTn_TMCSR:RELD. Figure 3-5 shows the clearing operation of underflow flag.

Figure 3-4 Underflow Operation of 32-Bit Reload Timer

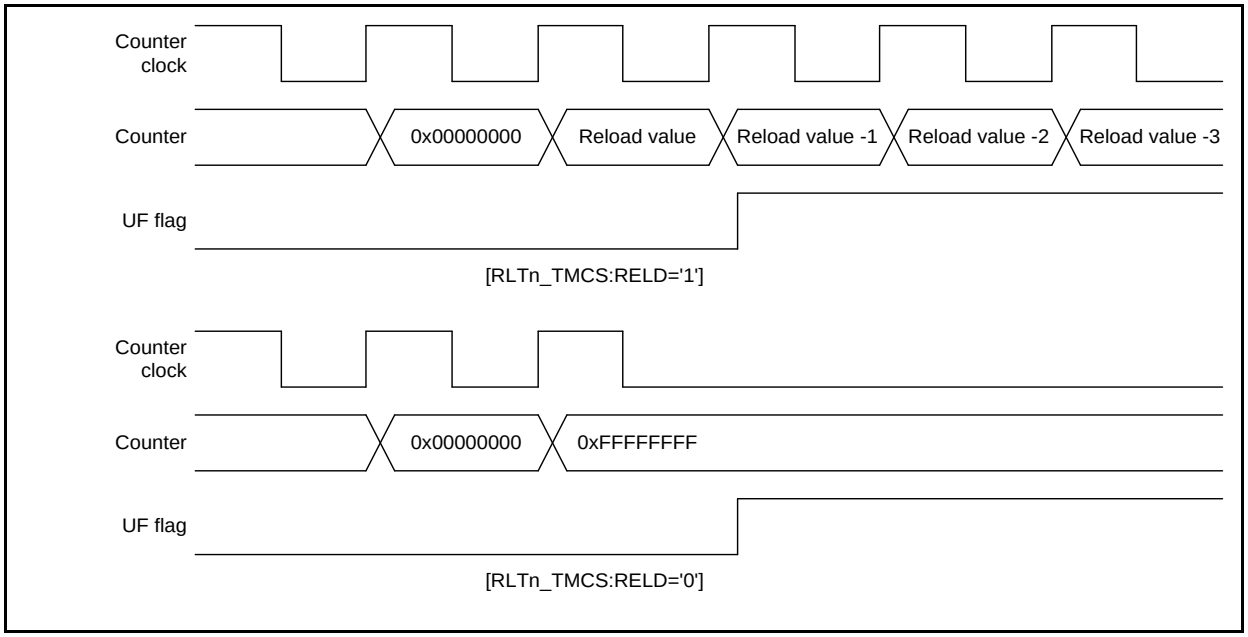
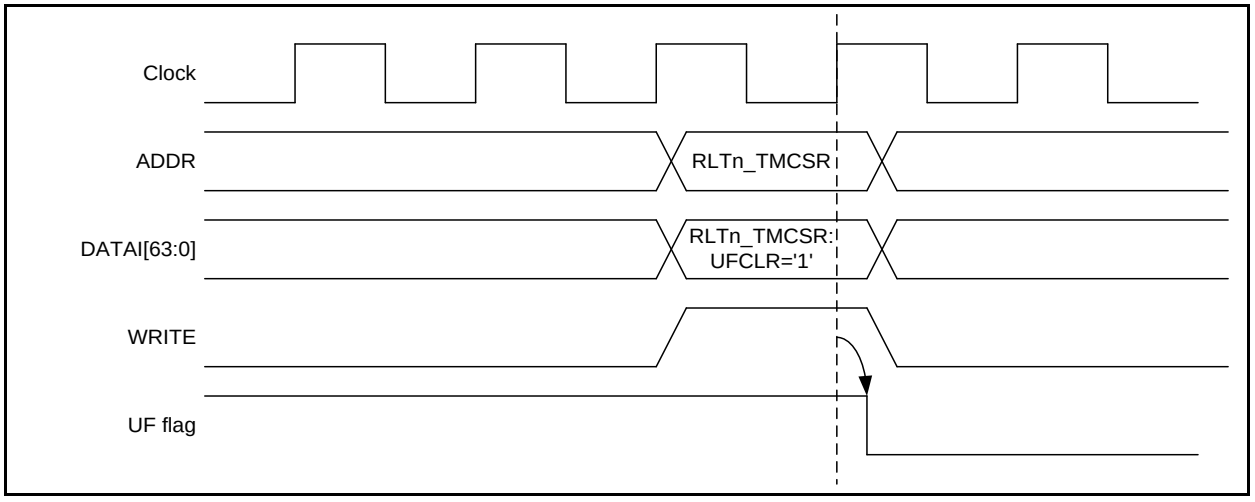


Figure 3-5 Clearing of Underflow Bit



3.3. Output Functions of 32-Bit Reload Timer

In reload mode, the TOT output performs toggle output (inverts at each underflow). In one-shot mode, the TOT output is used as a pulse output that shows the configured level while the counting is in progress.

Output Signal Functions of 32-Bit Reload Timer

The RLTn_TMCSR:OUTL bit sets the output polarity.

When RLTn_TMCSR:OUTL = 0, the initial value for toggle output is "L" and the one-shot pulse output is "H" while the count is in progress.

When RLTn_TMCSR:OUTL = 1, the output waveforms are opposite.

Figure 3-6 and Figure 3-7 show the output signal functions.

Figure 3-6 Output Signal Function of 32-Bit Reload Timer in Reload Mode

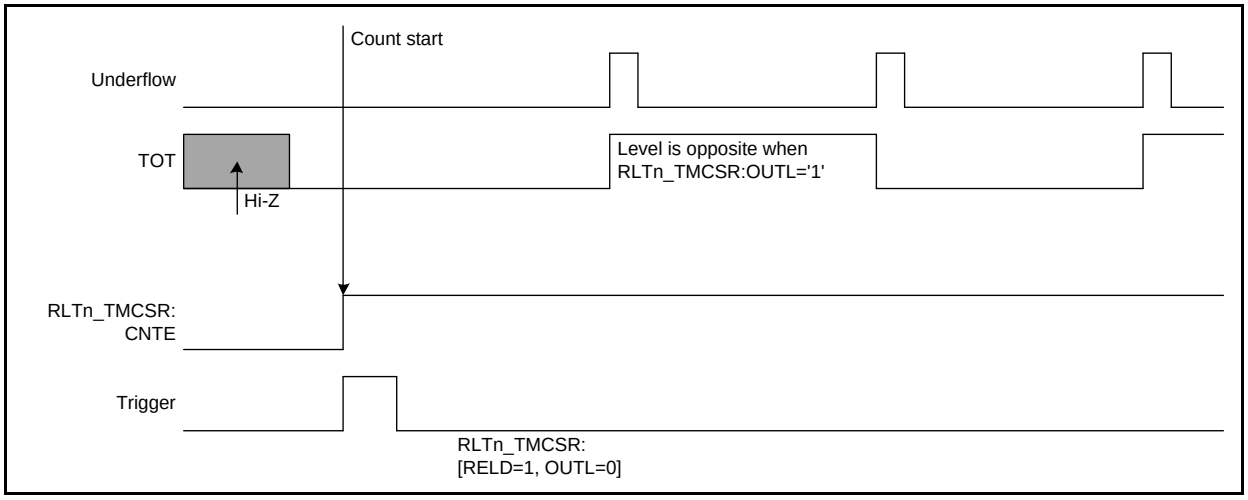
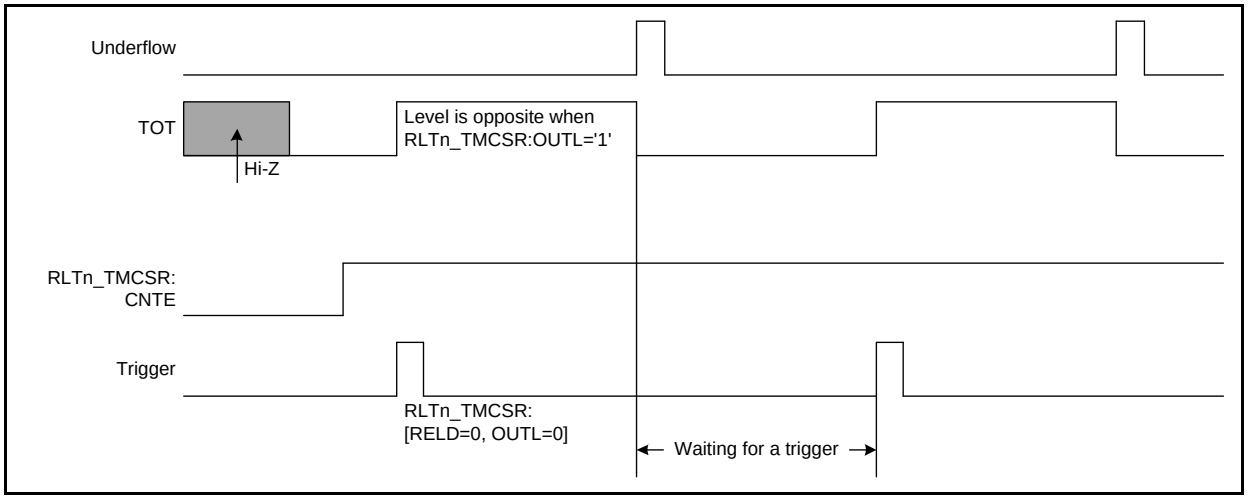


Figure 3-7 Output Signal Function of 32-Bit Reload Timer in One-Shot Mode



3.4. Counter Operation State

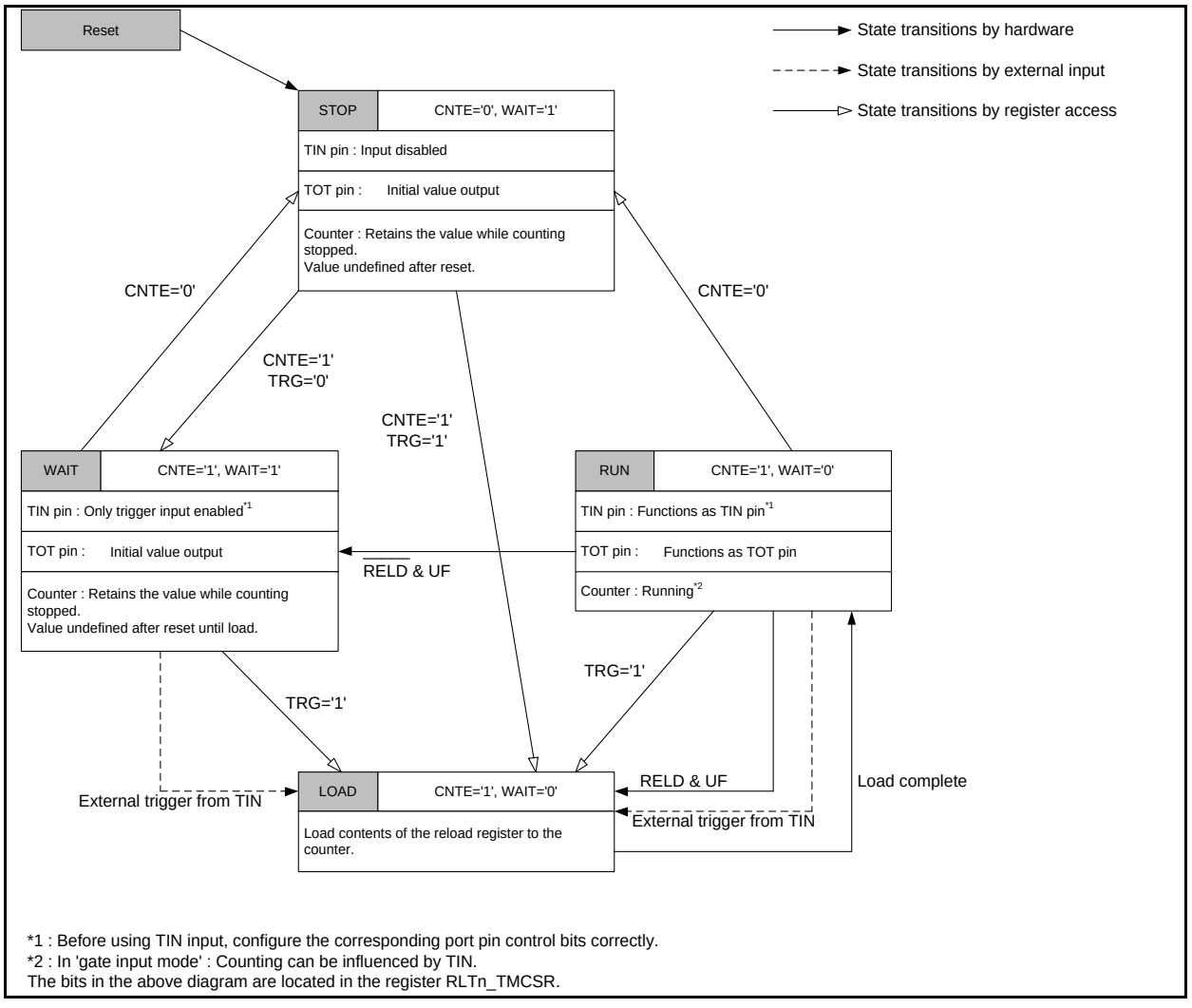
The counter state is determined by RLTn_TMCSR:CNTE bit in the Timer Control Status Register and the internal WAIT signal.

Available States for Reload Timer Are:

Stop state: RLTn_TMCSR:CNTE = 0 and WAIT = 1
Wait state: RLTn_TMCSR:CNTE = 1 and WAIT = 1
Run state: RLTn_TMCSR:CNTE = 1 and WAIT = 0

Counter Operation States

Figure 3-8 Counter Operation States



*1 : Before using TIN input, configure the corresponding port pin control bits correctly.
*2 : In 'gate input mode' : Counting can be influenced by TIN.
The bits in the above diagram are located in the register RLTn_TMCSR.

3.5. DMA Operation

The DMA support is determined by the RLTn_DMACFG:ENDMAUF bit. Setting this bit enables DMA request generation for DMA. Assertion of DMA_REQ_ACK signal acknowledges request and hence DMA_REQ signal gets de-asserted.

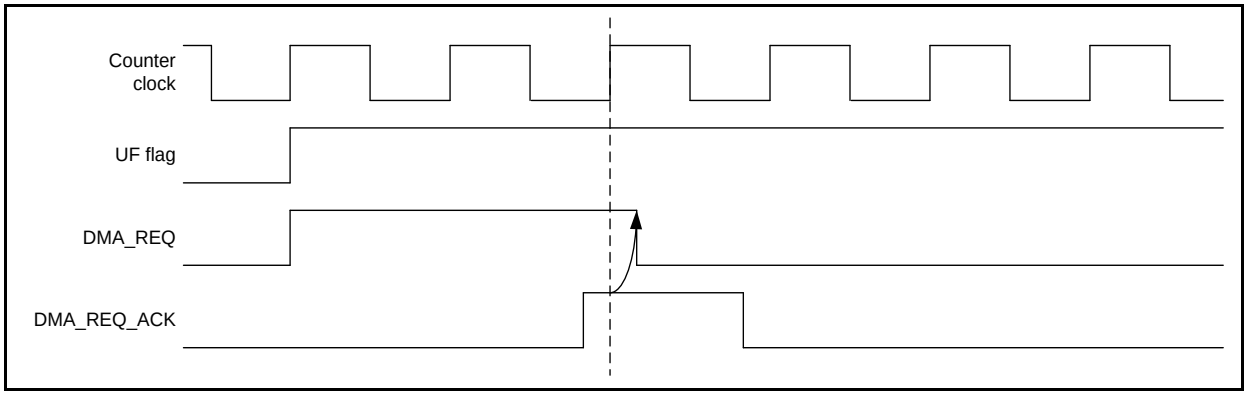
Enabling DMA Support

Writing "1" to RLTn_DMACFG:ENDMAUF enables DMA request generation for DMA when RLTn_TMCSR:UF bit sets. However, writing "0" to RLTn_DMACFG:ENDMAUF disables DMA request generation even if RLTn_TMCSR:UF bit sets.

When DMA_REQ_ACK is asserted the DMA_REQ signal gets de-asserted by acknowledging the DMA request.

Figure 3-9 shows behavior of DMA_REQ_ACK when asserted.

Figure 3-9 De-Asserting DMA_REQ Signal



4. Registers

This section describes the registers of 32-bit Reload Timer.

Table 4-1 Memory Layout of 32-Bit Reload Timer Registers

Offset	+3	+2	+1	+0
0x00000000	RLTn_DMACFG 00000000_00000000_00000000_00000000			
0x00000004	Reserved			
0x00000008	RLTn_TMCSR 00000000_00000000_00000000_00000000			
0x0000000C	Reserved			
0x00000010	RLTn_TMRLR XXXXXXXX_XXXXXXXX_XXXXXXXX_XXXXXXXX			
0x00000014	RLTn_TMR XXXXXXXX_XXXXXXXX_XXXXXXXX_XXXXXXXX			

Note:

- *The initial register value after reset indicates as follows:*
  - "1": Initial value "1"
  - "0": Initial value "0"
  - “X”: Initial value undefined
  - “-“: Reserved bit/Undefined bit
  - “*”: Initial value "0" or "1" according to the setting

4.1. DMA Configuration Register（RLTn_DMACFG）

The DMA Configuration Register controls the DMA request generation for underflow condition.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	ENDMAUF
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:1] Reserved

[bit0] ENDMAUF : DMA enable for underflow (ENDMAUF)

Bit	Description
0	No DMA request is generated
1	A DMA request is generated when the counter, RLTn_TMR, underflows

4.2. Timer Control Status Register（RLTn_TMCSR）

The Timer Control Status Register controls the operation mode and interrupt of the 32-bit Reload Timer.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	CNTE
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	TRG	UFCLR	UF
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,W	R0,W	R,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	MOD[2]	MOD[1]	MOD[0]	CSL2	CSL1	CSL0	Reserved	NFE
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R0,WX	R/W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	DBGE	Reserved	OUTL	RELD	INTE	Reserved	Reserved	Reserved
ACCESS_TYPE	R/W	R0,W0	R/W	R/W	R/W	R0,WX	R0,WX	R0,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:25] Reserved

[bit24] CNTE: Count enable

The Count Enable (CNTE) bit is a timer count enable bit.

Bit	Description
0	Stops count operation
1	Sets the timer to wait for a trigger

[bit23:19] Reserved

[bit18] TRG: Trigger

Software trigger bit.

Bit	Description
0	No effect
1	Applies a software trigger, causing the timer to load the reload register content to the counter and starts counting

Notes:

- *Applying a trigger using this register is only valid when RLTn_TMCSR:CNTE = 1. If RLTn_TMCSR:CNTE = 0, writing "1" to TRG has no effect.*
- *Set this bit in 'gate input mode' and in 'event count mode' to load the reload register content before counting starts.*

[bit17] UFCLR: Underflow interrupt clear

Bit	Description
0	No effect
1	Clears RLTn_TMCSR:UF bit

[bit16] UF: Underflow

The Underflow (UF) is timer interrupt request flag.

Bit	Description
0	No underflow occurred
1	When an underflow occurred

Note:

- *UF bit is cleared by writing "1" to RLTn_TMCSR:UFCLR bit.*

[bit15:13] MOD: Operation mode

The Operation Mode (MOD[2:0]) bits set the operation mode and input (TIN) functions.
Table 4-2 and Table 4-3 list the MOD[2:0] bit settings.

[bit12] CSL2: Clock select 2

The Clock Select 2 (CSL2) bit specifies the clock/event source and the clock division ratio.
Table 4-4 lists the selected clock sources for different CSL0/1/2 settings.

[bit11] CSL1: Clock select 1

The Clock Select 1 (CSL1) bit specifies the clock/event source and the clock division ratio.
Table 4-4 lists the selected clock sources for different CSL0/1/2 settings.

[bit10] CSL0: Clock select 0

The Clock Select 0 (CSL0) bit specifies the count clock division ratio.
Table 4-4 lists the selected clock sources for different CSL0/1/2 settings.

[bit9] Reserved

[bit8] NFE: Noise filter enable

This bit is used to enable/disable the noise filter for the TIN input.

Bit	Description
0	Noise filter for TIN is disabled
1	Noise filter for TIN is enabled

[bit7] DBGE: Debug mode enable

This bit is used to enable/disable debug mode for RLT.

Bit	Description
0	Debug mode disabled
1	Debug mode enabled

Notes:

- *When DBGE is set to "1" and the processor is in debug state, the timer counter operation is paused, and writing to the RLTn_TMRLR register directly updates the timer counter (RLTn_TMR register). When the processor leaves debug state or DBGE is set to "0", the timer counter operation is resumed.*
- *This bit is enable at*
  - *from ch. 0 to ch. 3 of CPERI#2*
  - *from ch.0 to ch. 1 of MCU_CONFIG_GROUP.*

[bit6] Reserved

[bit5] OUTL: Output level

This Output Level (OUTL) bit sets the output level for the TOT.
Refer to Table 4-5.

[bit4] RELD: Reload

This Reload (RELD) bit enables reload operations.
Refer to Table 4-5.

Bit	Description
0	The timer operates in one-shot mode. In this mode, the count operation stops when an underflow occurs due to the counter value changing from 0x00000000 to 0xFFFFFFFF
1	The timer operates in reload mode. In this mode, the timer loads the reload register contents into the counter and continues counting whenever an underflow occurs

[bit3] INTE: Interrupt enable

Timer interrupt request enable bit.

Bit	Description
0	No interrupt request is generated even when the RLTn_TMCSR:UF bit changes to "1"
1	An interrupt request is generated when the RLTn_TMCSR:UF bit changes to "1"

[bit2:0] Reserved

Table 4-2 RLTn_TMCSR:MOD[2:0] Bit Settings for Internal Clock Mode (RLTn_TMCSR:CSL1 / RLTn_TMCSR:CSL2 = 0b00, 0b01, or 0b10)

MOD[2]	MOD[1]	MOD[0]	Input Function	Active Edge or Level
0	0	0	Trigger disabled	-
0	0	1	Trigger input	Rising edge
0	1	0		Falling edge
0	1	1		Both edge
1	x	0	Gate input	"L" level
1	x	1		"H" level

Table 4-3 RLTn_TMCSR:MOD[2:0] Bit Settings for Event Counter Mode (RLTn_TMCSR:CSL1 / RLTn_TMCSR:CSL2 = 0b11)

MOD[2]	MOD[1]	MOD[0]	Input Function	Active Edge or Level
x	0	0	-	-
	0	1	Event input	Rising edge
	1	0		Falling edge
	1	1		Both edge

Table 4-4 Clock Sources for RLTn_TMCSR:CSL0 / RLTn_TMCSR:CSL1 / RLTn_TMCSR:CSL2 Bit Settings

CSL2	CSL1	CSL0	Count Clock (Time for Peripheral Clock)
0	0	0	1/1
0	0	1	1/2
0	1	0	1/4
0	1	1	1/8
1	0	0	1/16
1	0	1	1/32
1	1	0	External event count mode, each event on TIN
1	1	1	External event count mode, each second event on TIN

Table 4-5 RLTn_TMCSR: RLTn_TMCSR:OUTL, and RLTn_TMCSR:RELD Settings

OUTL	RELD	Output Waveform
0	0	Output an "H" level pulse during counting.
1	0	Output an "L" level pulse during counting.
0	1	Toggle output. Starts with "L" level output. Changes level on timer reload.
1	1	Toggle output. Starts with "H" level output. Changes level on timer reload.

Note:

- *Bits marked 'x' in the table can be set to any value.*

4.3. 32-Bit Reload Register (RLTn_TMRLR)

The Timer Reload Register holds the reload value of the 32-bit Reload Timer.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	TMRLR[31:24]							
ACCESS_TYPE	R/W							
PROT_TYPE	-							
INITIAL_VALUE	XXXXXXXX							

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	TMRLR[23:16]							
ACCESS_TYPE	R/W							
PROT_TYPE	-							
INITIAL_VALUE	XXXXXXXX							

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	TMRLR[15:8]							
ACCESS_TYPE	R/W							
PROT_TYPE	-							
INITIAL_VALUE	XXXXXXXX							

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	TMRLR[7:0]							
ACCESS_TYPE	R/W							
PROT_TYPE	-							
INITIAL_VALUE	XXXXXXXX							

[bit31:0] TMRLR: Timer reload register

The Timer Reload Register (RLTn_TMRLR) is a 32-bit reload register holds the reload value. Initial value is undefined.

This register can be accessed only in 32-bit/64-bit mode.

When RLTn_TMCSR:DBGE is set to "1" and the processor is in debug state, writing to this register updates the timer counter immediately.

Note:

- *This register is not initialized by Hard Reset (i.e. “X”).*

4.4. 32-Bit Timer Register (RLTn_TMR)

Reading this register returns the count value of the 32-bit Reload Timer. The initial value is undefined.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	TMR[31:24]							
ACCESS_TYPE	R,WX							
PROT_TYPE	-							
INITIAL_VALUE	XXXXXXXX							

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	TMR[23:16]							
ACCESS_TYPE	R,WX							
PROT_TYPE	-							
INITIAL_VALUE	XXXXXXXX							

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	TMR[15:8]							
ACCESS_TYPE	R,WX							
PROT_TYPE	-							
INITIAL_VALUE	XXXXXXXX							

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	TMR[7:0]							
ACCESS_TYPE	R,WX							
PROT_TYPE	-							
INITIAL_VALUE	XXXXXXXX							

[bit31:0] TMR: Timer register

Reading this Timer Register (RLTn_TMR) returns the count value of the 32-bit Reload Timer. Initial value is undefined.

This register can be accessed only in 32-bit/64-bit mode.

Note:

- *This register is not initialized by Hard Reset (i.e. "X").*

CHAPTER 46: Reload Timer Simultaneous Soft Start

This chapter describes the reload timer simultaneous soft start.

1. **Overview**
2. **Configuration**
3. **Explanation of Operation**
4. **Registers**

RLTSSSR-TXXPT03P01R01L05-E1-XX

1. Overview

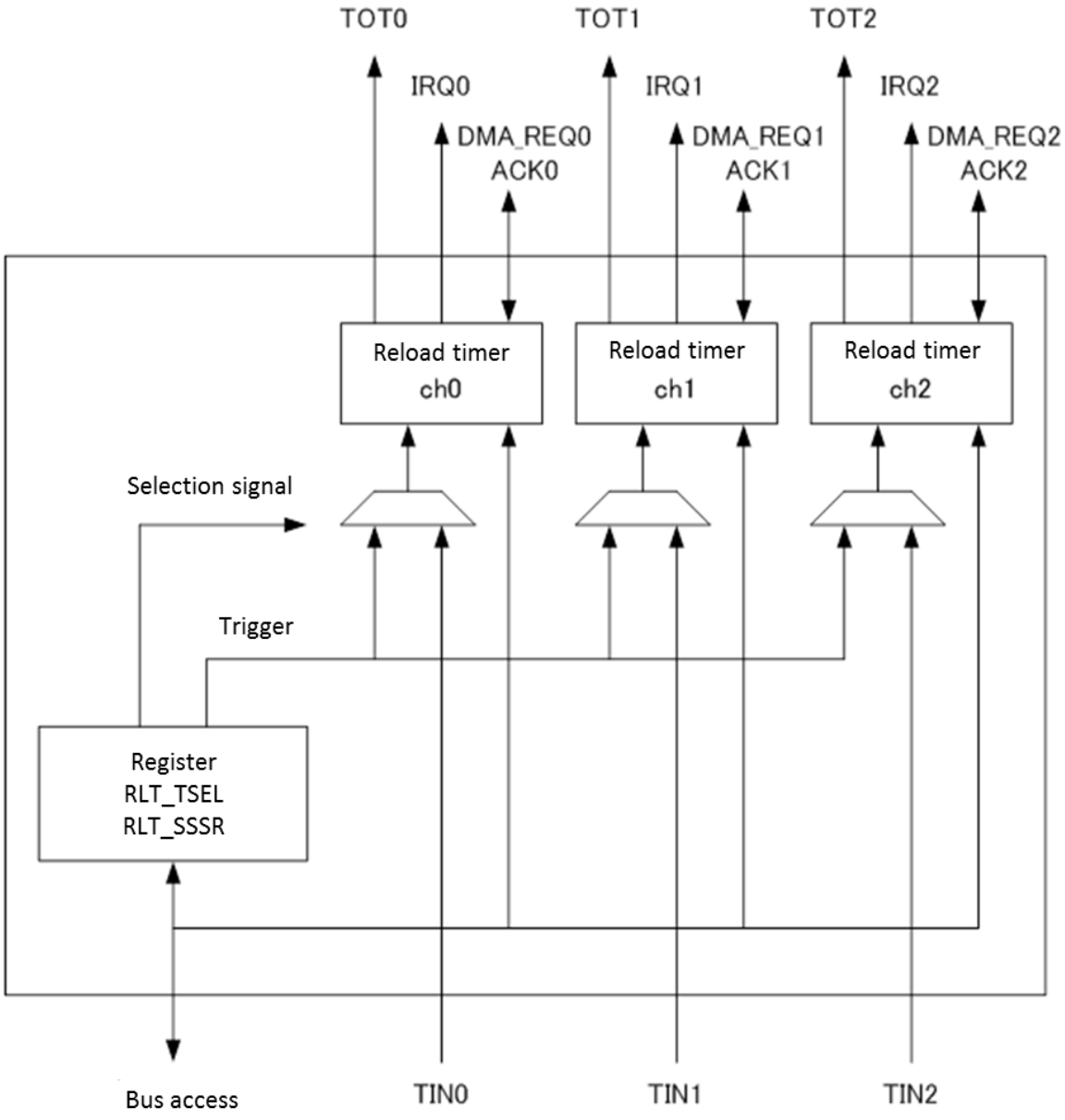
This section provides the overview of the reload timer simultaneous soft start.

The reload timer simultaneous soft start is a function to use a register write access as a trigger to start the reload timers of multiple channels simultaneously.

2. Configuration

This section describes the configuration of the reload timer simultaneous soft start.

Figure 2-1 Block Diagram



For Channel 0, Channel 1, and Channel 2

3. Explanation of Operation

This section describes the operation of the reload timer simultaneous soft start.

Trigger Selection Function

The RLTx_TSEL(x=0, 1, 2) register enables selections of the external trigger and software trigger for each channel.

Multiple Channel Simultaneous Start

A write access to the RLTx_SSSR(x=0, 1, 2) register can activate a software trigger to start multiple reload timers simultaneously. The software trigger is "0" in normal, and it changes from '0' to '1' to '0'.

Notes:

- *The reload timer simultaneous soft start activates a trigger that changes '0' to '1' to '0'. Set the register of the reload timer according to your needs.*
- *When the simultaneous soft start operates, set to disable the Noise Filter of Reload Timer.*

4. Registers

This section describes the registers of the reload timer simultaneous soft start.

Table 4-1 List of Register Area

Offset	Register name / Initial value				Block name
	+3	+2	+1	+0	
0000_0000	RLT0_TSEL 00000000 00000000 00000000 00000000				Common Peri #0
0000_0004	RLT0_SSSR 00000000 00000000 00000000 00000000				

Offset	Register name / Initial value				Block name
	+3	+2	+1	+0	
0000_0000	RLT1_TSEL 00000000 00000000 00000000 00000000				Common Peri #1
0000_0004	RLT1_SSSR 00000000 00000000 00000000 00000000				

Offset	Register name / Initial value				Block name
	+3	+2	+1	+0	
0000_0000	RLT2_TSEL 00000000 00000000 00000000 00000000				Common Peri #2
0000_0004	RLT2_SSSR 00000000 00000000 00000000 00000000				

Table 4-2 List of Register Mirror Area

Address of Register Mirror Area	Mirrored Peripheral	Mirror Area Behavior	PPU
0003_BC00 to 0003_BFFF	ReloadTimer SSSR CommonPERI#0	Accessing this region results to the access to ReloadTimer SSSR CommonPERI#0 register area with following offset. Offset: (addr & 0000_03FF)	This area is covered by PPU #80 as well as the mirrored peripheral.
000B_BC00 to 000B_BFFF	ReloadTimer SSSR CommonPERI#1	Accessing this region results to the access to ReloadTimer SSSR CommonPERI#1 register area with following offset. Offset: (addr & 0000_03FF)	This area is covered by PPU #81 as well as the mirrored peripheral.

Note:

- *The register mirror area does not exist in Common Peri #2.*

Table 4-3 List of Registers of Reload Timer Simultaneous Soft Start

Abbreviated Register Name	Register Name	Reference
RLT0_TSEL	Reload timer trigger selection register (channel 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15)	4.1
RLT1_TSEL	Reload timer trigger selection register (channel 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31)	4.1
RLT2_TSEL	Reload timer trigger selection register (channel 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47)	4.1
RLT0_SSSR	Reload timer simultaneous soft start register (channel 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15)	4.2
RLT1_SSSR	Reload timer simultaneous soft start register (channel 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31)	4.2
RLT2_SSSR	Reload timer simultaneous soft start register (channel 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47)	4.2

4.1. Reload Timer Trigger Selection Register (RLT0_TSEL, RLT1_TSEL, RLT2_TSEL)

This register enables selections of the external trigger and software trigger for each channel.

Bit	31							16
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	-							
Initial Value	00000000_00000000							

Bit	15	14	13	12	11	10	9	8
Field	TSEL15	TSEL14	TSEL13	TSEL12	TSEL11	TSEL10	TSEL9	TSEL8
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	TSEL7	TSEL6	TSEL5	TSEL4	TSEL3	TSEL2	TSEL1	TSEL0
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

[bit31:16] Reserved: Reserved Bits

[bit15:0] TSEL15 to TSEL0: Trigger Selection Bits

These bits can select triggers for up to 16 channels.

TSEL15 to TSEL0 correspond to Channel 15 to Channel 0.

The channel that does not contain a reload timer can access registers, but it does not have a trigger selection function.

Bit	Description
0	Selects an external trigger (TIN).
1	Selects a software trigger.

4.2. Reload Timer Simultaneous Soft Start Register (RLT0_SSSR, RLT1_SSSR, RLT2_SSSR)

A write access can activate a software trigger to start multiple reload timers simultaneously.

Bit	31	16
Field	Reserved	
R/W Attribute	R0,WX	
Protection Attribute	-	
Initial Value	00000000_00000000	

Bit	15	14	13	12	11	10	9	8
Field	SSSR15	SSSR14	SSSR13	SSSR12	SSSR11	SSSR10	SSSR9	SSSR8
R/W Attribute	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	SSSR7	SSSR6	SSSR5	SSSR4	SSSR3	SSSR2	SSSR1	SSSR0
R/W Attribute	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

[bit31:16] Reserved: Reserved Bits

[bit15:0] SSSR15 to SSSR0: Simultaneous Soft Start Bits

These bits activate software triggers for up to 16 channels simultaneously.
SSSR15 to SSSR0 correspond to Channel 15 to Channel 0.
The channel that does not contain a reload timer can access registers, but it does not have a software trigger function.

Bit	Description
0	Do not activate a software trigger.
1	Activate a software trigger.

CHAPTER 47: QPRC (Quadrature Position/Revolution Counter)

This chapter explains the functions and operations of the QPRC(Quadrature Position/Revolution Counter).

1. Overview
2. Configuration
3. Operations
4. Registers

QPRC-TXXPT03P01R02L02-E1-XX

1. Overview

The QPRC(Quadrature Position/Revolution Counter) is used to measure the position of Position Encoder. Also, it can be used as an up/down counter by its setting. The QPRC(Quadrature Position/Revolution Counter) contains a 16-bit position counter, a 16-bit revolution counter, two 16-bit compare registers, a control register, and its control circuit.

Features of QPRC(Quadrature Position/Revolution Counter)

The Position Counter can be Operated in One of the Following 3 Counting Modes

- PC_Mode1 : Up/down count mode
- PC_Mode2 : Phase difference count mode (supporting the 2-time and 4-time frequency multiplication)
- PC_Mode3 : Count mode with direction

The Revolution Counter can be Operated in One of the Following 3 Counting Modes

- RC_Mode1 : The revolution counter can count up or down at a ZIN active edge only.
- RC_Mode2 : The revolution counter can count up or down with an output value of position counter only.
- RC_Mode3 : The revolution counter can count up or down both with an output value of position counter and a signal at ZIN active edge.

A Signal Edge Detection can be Set for Detecting an Input Event from Three AIN, BIN and ZIN External Pins

- Detection of falling edge
- Detection of rising edge
- Detection of both rising and falling edges

The Following Two Functions can be Selected for Input in ZIN Pin

- Counter clear function
- Gate function

An Interrupt Request can be Generated If

- The position counter value matches the Position Compare Register,
- The position counter value matches the Position and Revolution Compare Register value, or the revolution counter value matches the Position and Revolution Compare Register value,
- The position counter underflows,
- The position counter overflows(that is, the position counter value matches the value of the QPRC Maximum Position Register),
- The position counter is reset at a ZIN active edge,
- The counting of position counter is inverted,
- The position counter matches the Position Compare Register value, and the revolution counter matches the Position and Revolution Compare Register value, or
- An outrange revolution counter value is detected.

The Following Useful Functions are Provided for Counting

- Swap function of AIN and BIN external pins
- Mask reset function of the position counter

- Count direction check function during position counter operation or during overflow/underflow occurrence

Interrupt request and DMA request

Interrupt request of this module can be used for purposes below:

1. Interrupt request to CPU (via Interrupt Controller)
2. DMA request to DMA Controller

To use the interrupt as interrupt request to CPU, following configuration is necessary:

- Enable the interrupt request in this module
- Enable the interrupt channel of the interrupt in Interrupt Controller

To use the interrupt as DMA request to DMA Controller, following configuration is necessary:

- Enable the interrupt request in this module
- Select the DMA channel from this module as DMA client in DMA Controller

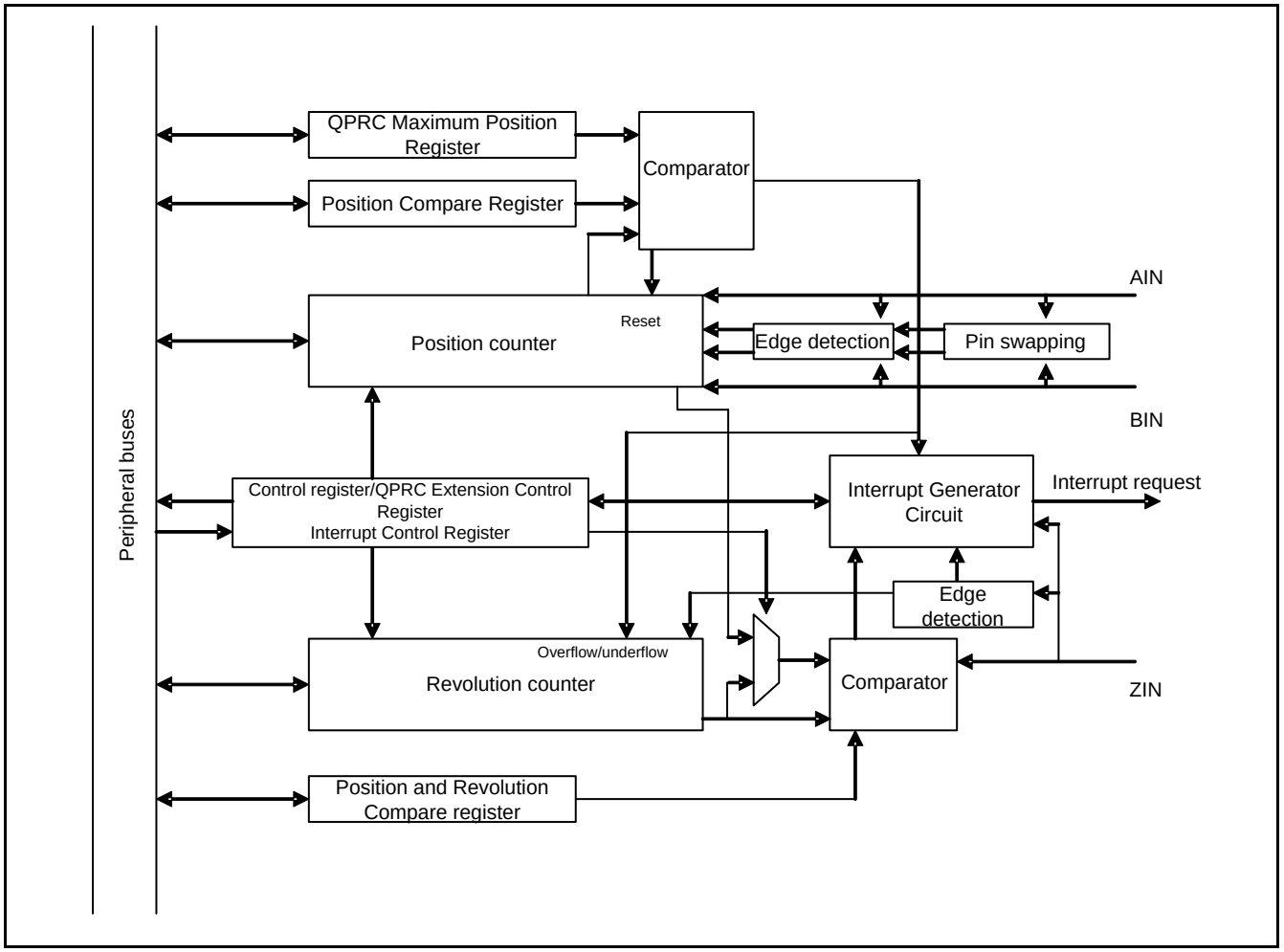
Note:

- *Interrupt request that supports for DMA request is product specification. Please refer to the product's HWM for detail.*

2. Configuration

The following shows the configuration of QPRC(Quadrature Position/Revolution Counter).

Figure 2-1 Block Diagram of QPRC(Quadrature Position/Revolution Counter)



3. Operations

This section explains the operation of QPRC(Quadrature Position/Revolution Counter).

3.1. Operation of Position Counter

The position counter receives an input signal from AIN or BIN external pin as an event of count clock, and increments or decrements the counter. As listed in Table 3-1, the position counter can select a counting mode by setting of the position counter mode bits(QCR.PCM[1:0]) of a control register. The counting conditions depend on the selected count mode.

The position counter is counted up or down in the following ZIN conditions only.

- If the ZIN function is set to the count clear function(QCR.CGSC= 0)
- If the ZIN function is set to the Gate function(QCR.CGSC= 1), the ZIN "L" level detection(QCR.CGE[1:0]= 0b01) is set, and the ZIN is "L"
- If the ZIN function is set to the Gate function(QCR.CGSC= 1), the ZIN "H" level detection(QCR.CGE[1:0]= 0b10) is set, and the ZIN is "H"

If the ZIN function is set to the Gate function(QCR.CGSC= 1) and if a level other than ZIN "H" or "L" level detection(QCR.CGE[1:0]= 0b00 or 0b11) is set, the position counter is not counted up or down.

Also, if the AIN and BIN configuration is swapped by SWAP bits of a control register, the AIN and BIN pins are swapped and the position counter is counted up or down.
For example, if PC_Mode1(QCR.PCM[1:0]= 0b01) and QCR.AES[1:0]= 0b10(rising edge) and QCR.BES[1:0]= 0b01(falling edge) are set, the following occurs.

- If QCR.SWAP= 0 and when a rising edge of AIN signal is detected, the position counter is counted up. When a falling edge of BIN signal is detected, the position counter is counted down.
- If QCR.SWAP= 1, the position counter is counted down at a falling edge of AIN signal but it is counted up at a rising edge of BIN signal.

Table 3-1 Counting Conditions of AIN and BIN Pin Position Counter

Position Count Mode(PC_Mode)	AIN Counting Conditions	BIN Counting Conditions
Count disable PC_Mode0:QCR.PCM[1:0]= 0b00	Position counter disable	Position counter disable
Up/down counting PC_Mode1:QCR.PCM[1:0]= 0b01	AIN Active edge	BIN Active edge
Phase difference count PC_Mode2:QCR.PCM[1:0]= 0b10	AIN Active edge or "H"/"L" level	"H"/"L" level or BIN active edge
Counting with direction PC_Mode3:QCR.PCM[1:0]= 0b11	"H"/"L" level	BIN Active edge

Note:

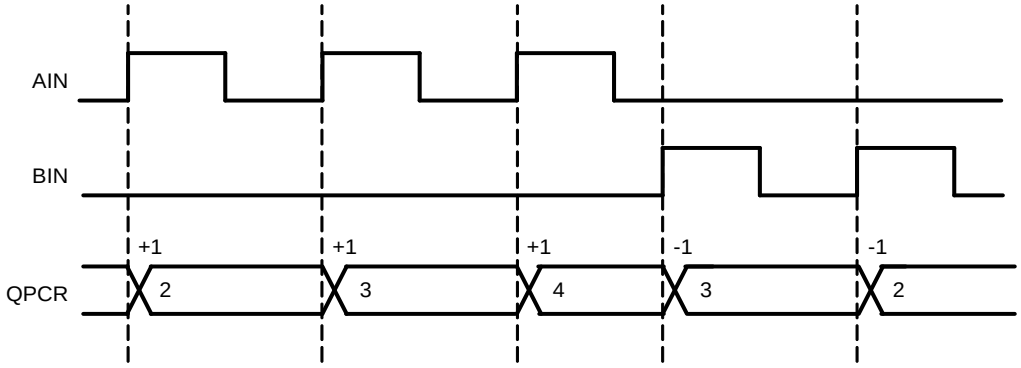
- *The active edge of AIN signal and the active edge of BIN signal mean a rising edge, a falling edge, or both of the respective signal if they are set by the AIN Detection Edge Select bits(QCR.AES[1:0]= 0b01 or 0b10 or 0b11) or by the BIN Detection Edge Select bits(QCR.BES[1:0]= 0b01 or 0b10 or 0b11).*

PC_Mode1: Up/Down Count Mode

An external signal entered from AIN or BIN external pin is received as the counting clock, and the position counter is counted up or down.

In this mode, the position counter is counted up when an active edge of AIN signal is detected. When an active edge of BIN signal is detected, the position counter is counted down.

Figure 3-1 Operations in Up/Down Count Mode
(QCR.AES[1:0]=0b10, QCR.BES[1:0]=0b10, QCR.SWAP= 0)



PC_Mode2: Phase Difference Count Mode(Supporting the 2-Time and 4-Time Frequency Multiplication)

- This mode is useful for counting the difference between phases A and B of "encoder output signal." If the phase-A and phase-B outputs are respectively connected to the AIN and BIN pins and if phase A is leading phase B, the counter is counted up. If delayed, the counter is counted down.
- In this mode, when an active edge of AIN signal is detected, the BIN signal level is checked and the position counter counts it. In the opposite case, the position counter also counts it.
- Counting in the 4-time or 2-time frequency multiplication can be made by setting the AES and BES bits of QPRC Control Register(QCR). The counting in these frequency multiplication modes allows more accurate position measurement as its counting resolution is very high.

Table 3-2 AES and BES Bit Settings in Frequency Multiplication Mode

Frequency Multiplication Mode	QCR.AES[1:0]	QCR.BES[1:0]
2-time frequency multiplication mode	11	00
	00	11
4-time frequency multiplication mode	11	11

Table 3-3 Counting in 2-Time Frequency Multiplication Mode (QCR.AES[1:0]=0b00, QCR.BES[1:0]=0b11, QCR.SWAP= 0)

Edge Detection Pin	Detection Edge	Level Check Pin	Input Level	Counting Direction	Figure 3-2 Timing
BIN	Rising edge	AIN	"H"	Up	(1)
	Rising edge		"L"	Down	(2)
	Falling edge		"H"	Down	(3)
	Falling edge		"L"	Up	(4)

Figure 3-2 Operation in 2-Time Frequency Multiplication Mode (QCR.AES[1:0]=0b00, QCR.BES[1:0]=0b11, QCR.SWAP= 0)

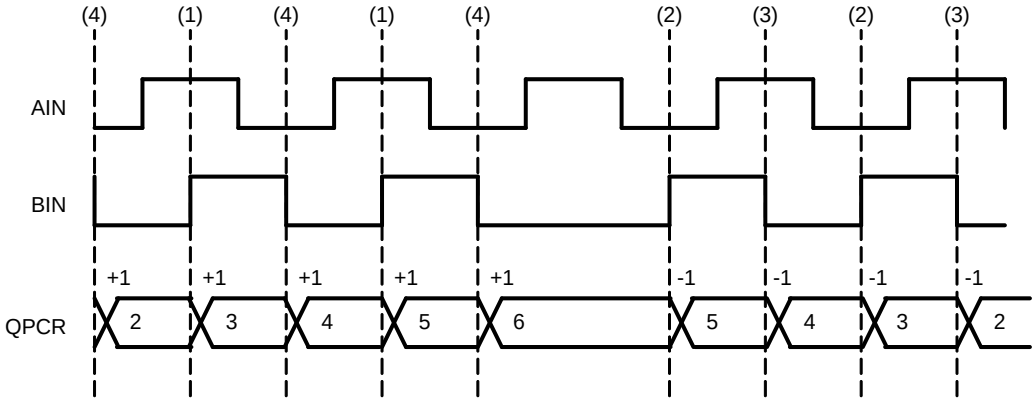
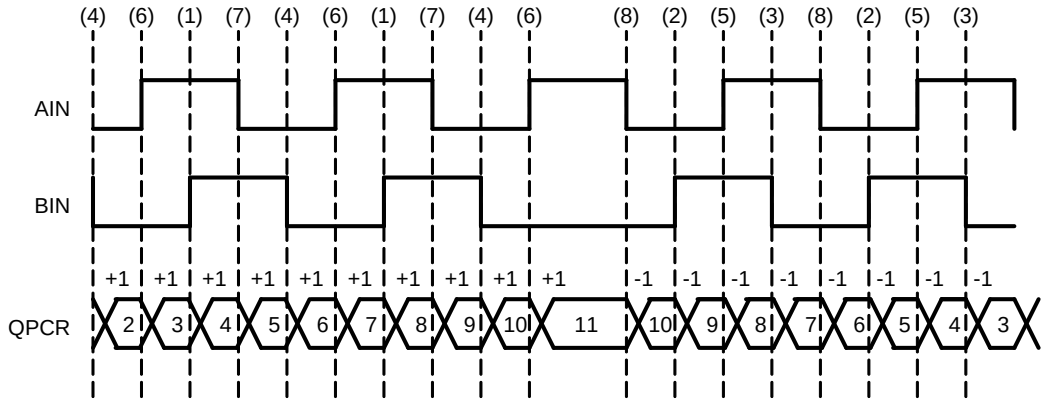


Table 3-4 Counting in 4-Time Frequency Multiplication Mode (QCR.AES[1:0]=0b11, QCR.BES[1:0]=0b11)

Edge Detection Pin	Detection Edge	Level Check Pin	Input Level	Counting Direction	Figure 3-3 Timing
BIN	Rising edge	AIN	"H"	Up	(1)
	Rising edge		"L"	Down	(2)
	Falling edge		"H"	Down	(3)
	Falling edge		"L"	Up	(4)
AIN	Rising edge	BIN	"H"	Down	(5)
	Rising edge		"L"	Up	(6)
	Falling edge		"H"	Up	(7)
	Falling edge		"L"	Down	(8)

Figure 3-3 Operation in 4-Time Frequency Multiplication Mode (QCR.AES[1:0]=0b11, QCR.BES[1:0]=0b11, QCR.SWAP= 0)



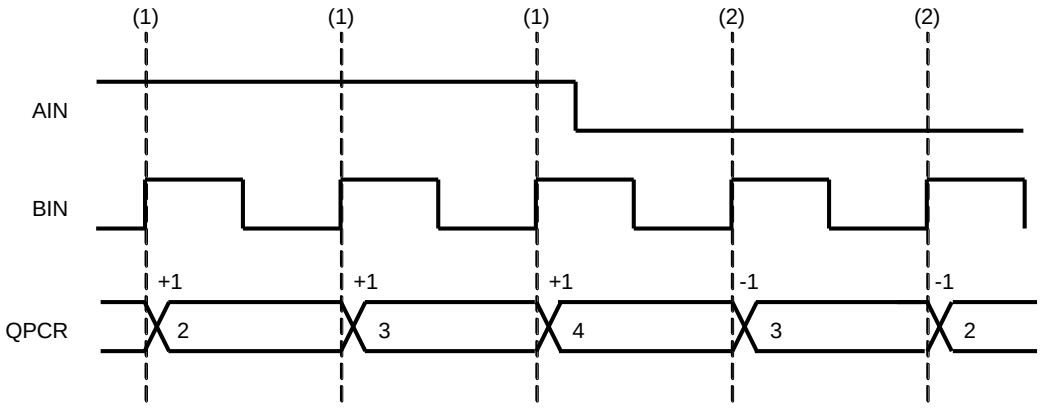
PC_Mode3: Count Mode with Direction

- A signal entered from the BIN external pin is received as the counting clock, and an input level of the signal entered form the AIN external pin is used for count direction control for counter up/down counting.
- In this mode, when an active edge of BIN signal is detected, the AIN signal level is checked and the position counter counted up or down. A rising edge, a falling edge, or both can be set as the active edge.

Table 3-5 Counting in the Direction Control Counting Mode

Edge Detection Pin	Detection Edge	Level Check Pin	Input Level	Counting Direction	Figure 3-4 Timing
BIN	Active edge	AIN	"H"	Up	(1)
	Active edge		"L"	Down	(2)

Figure 3-4 Operation in the Direction Control Counting Mode (QCR.AES[1:0]=0b00, QCR.BES[1:0]=0b10, QCR.SWAP= 0)



3.2. Operation of Revolution Counter

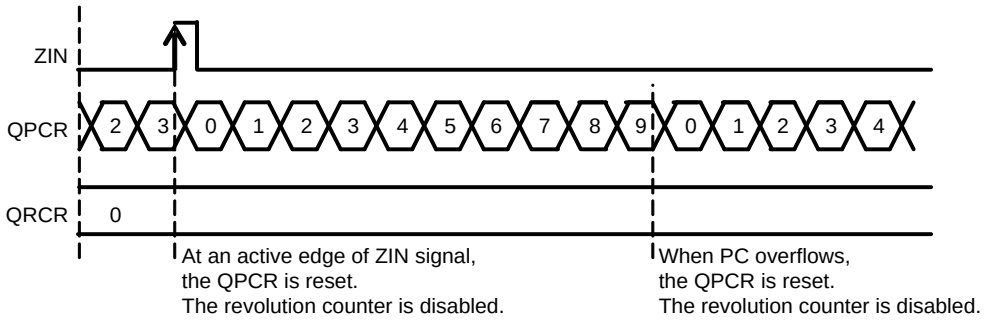
When the revolution counter receives an input from the ZIN pin(having the counter clear function) or an output of position counter(underflow or overflow), it is counted up or down. A rising edge, a falling edge, or both can be set as the active edge of ZIN signal.

The counting conditions of revolution counter depend on the selected mode as follows.

RC_Mode0(QCR.RCM[1:0]= 0b00)

- The revolution counter is disabled.
- If the ZIN signal is used for counter clear function(QCR.CGSC= 0), the position counter is reset at an active edge of the ZIN signal. Also, the position counter is reset when this counter overflows.

Figure 3-5 RC_Mode0 Operation(QPRC Maximum Position Register QMPR=9, QCR.CGSC= 0)



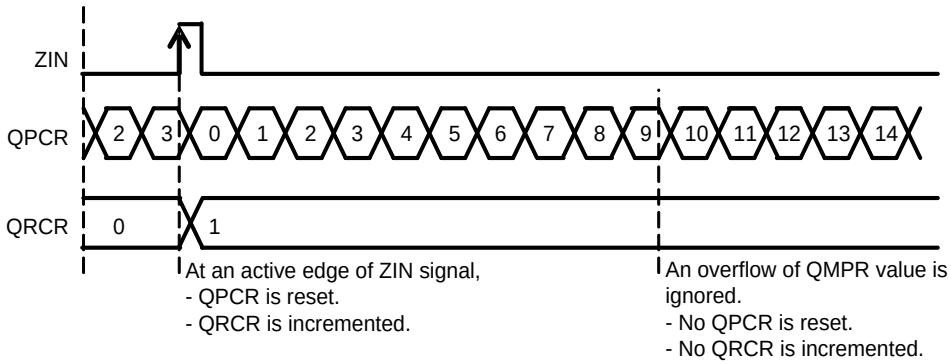
QPCR: QPRC Position Count Register

QRCR: QPRC Revolution Count Register

RC_Mode1(QCR.RCM[1:0]= 0b01)

- When ZIN signal is used for the counter clear function(QCR.CGSC= 0), the revolution counter is operated only an active edge of ZIN signal(but an input from the position counter is ignored).
- When an active edge of ZIN signal is detected during incrementing of position counter(QICR.DIRPC= 0), the revolution counter is counted up. When an active edge of ZIN is detected during decrementing of position counter(QICR.DIRPC= 1), it is counted down.
- When the ZIN signal is used for counter clear function(QCR.CGSC= 0), the position counter is reset only at an active edge of ZIN signal.
- The position counter is not reset even when an overflow of position counter is detected. When an overflow of position counter is detected, the position counter is counted up and the overflow flag(QICR.OFDF) is set to "1".

Figure 3-6 RC_Mode1 Operation(QPRC Maximum Position Register QMPR=9, QCR.CGSC= 0)



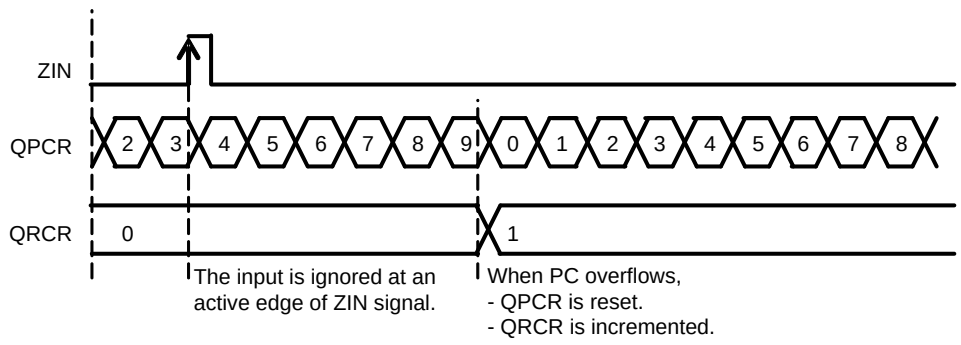
Notes:

- *When an active edge of ZIN signal and an active edge which counts down position counter are detected at the same time during incrementing of position counter(QICR.DIRPC= 0), the revolution counter is counted down.*
- *When an active edge of ZIN signal and an active edge which counts up position counter are detected at the same time during decrementing of position counter(QICR.DIRPC= 1), the revolution counter is counted up.*
- *When an active edge of ZIN signal, an active edge of AIN signal, and an active edge of BIN signal are detected at the same time, the revolution counter is counted up or down in accordance with the last position counter direction bit(QICR.DIRPC).*

RC_Mode2(QCR.RCM[1:0]= 0b10)

- The revolution counter is counted up or down only by the output value of position counter.
- The position counter is reset only when an overflow of position counter is detected(but an event of ZIN signal is ignored).
- When an overflow of position counter is detected in any of 3 position counter modes(PC_Mode1, PC_Mode2 and PC_Mode3), the position counter is counted up. When an underflow of it is detected, the position counter is counted down.

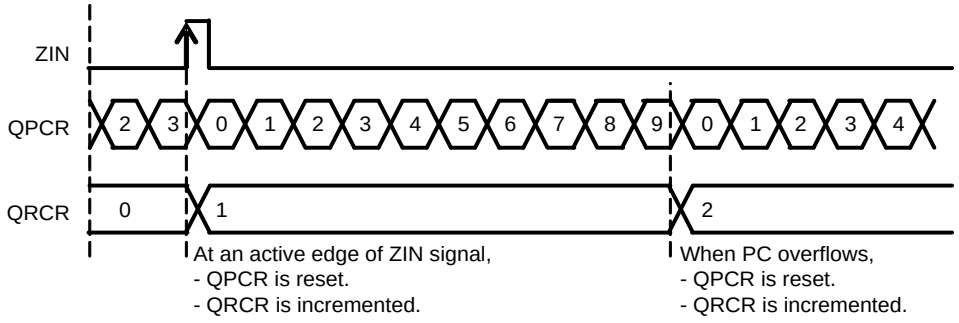
Figure 3-7 RC_Mode2 Operation(QPRC Maximum Position Register QMPR=9)



RC_Mode3(QCR.RCM[1:0]= 0b11)

- In this mode, the revolution counter is operated with an output of position counter and when the ZIN signal is used for the counter clear function(QCR.CGSC= 0), the revolution counter is also counted up or down at an active edge of ZIN signal.
- When an active edge of ZIN signal is detected during incrementing of position counter(QICR.DIRPC= 0) or when an overflow of position counter is detected, the revolution counter is counted up.
- When an active edge of ZIN signal is detected during decrementing of position counter(QICR.DIRPC= 1) or when an underflow of position counter is detected, the revolution counter is counted down.
- When the ZIN signal is used for the counter clear function(QCR.CGSC= 0), the position counter is reset at an active edge of ZIN signal or at detection of position counter overflow.

Figure 3-8 RC_Mode3 Operation(QPRC Maximum Position Register QMPR=9, QCR.CGSC= 0)



Notes:

- *When an active edge of ZIN signal and an active edge which counts down position counter are detected at the same time during incrementing of position counter(QICR.DIRPC= 0), the revolution counter is counted down.*
- *When an active edge of ZIN signal and an active edge which counts up position counter are detected at the same time during decrementing of position counter(QICR.DIRPC= 1), the revolution counter is counted up.*
- *When an active edge of ZIN signal, an active edge of AIN signal, and an active edge of BIN signal are detected at the same time, the revolution counter is counted up or down in accordance with the last position counter direction bit(QICR.DIRPC).*

3.3. Absolute Value of Positions

In RC_Mode2 and 3 mode(when the revolution counter operates with an output of position counter), each position has the following absolute value.

QPRC Position Count Register(QPCR) + QPRC Revolution Count Register(QRCR) × (QPRC Maximum Position Register(QMPR) +1)

Example: Time measurement

The revolution counter counts the "hours", and the position counter counts the "minutes".
If QMPR=59, QPCR=20, and QRCR=5

Time = 20 + 5 × (59 + 1)

= 320 minutes. This is the absolute value in position counter units(minutes).

3.4. QPRC(Quadrature Position/Revolution Counter) Interrupts

The following table defines the conditions where an interrupt request of QPRC(Quadrature Position/Revolution Counter) can generate.

Table 3-6 Generation Conditions of QPRC(Quadrature Position/Revolution Counter) Interrupt Requests

Interrupt Request	Interrupt Request Flag	Interrupt Request is Enabled If	Interrupt Request is Cleared If
Count inversion interrupt request	QICR.CDCF= 1	QICR.CDCIE= 1	"0" is written to QICR.CDCF
Zero index interrupt request	QICR.ZIIF= 1	QICR.OUZIE= 1	"0" is written to QICR.ZIIF
Overflow interrupt request	QICR.OFDF= 1		"0" is written to QICR.OFDF
Underflow interrupt request	QICR.UFDF= 1		"0" is written to QICR.UFDF
PC and RC match interrupt request	QICR.QPRCMF= 1	QICR.QPRCMIE= 1	"0" is written to QICR.QPRCMF
PC match interrupt request	QICR.QPCMF= 1	QICR.QPCMIE= 1	"0" is written to QICR.QPCMF
PC match and RC match interrupt request	QICR.QPCNRCMF= 1	QICR.QPCNRCMIE= 1	"0" is written to QICR.QPCNRCMF
Outrange interrupt request	QECR.ORNGF= 1	QICR.ORNGIE= 1	"0" is written to QECR.ORNGF

QICR : QPRC Interrupt Control Register

QECR : QPRC Extension Control Register

3.5. Operation Example of QPRC Maximum Position Register(QMPR) Interrupt

The QPRC Maximum Position Register(QMPR) value is used as the reload data to the position counter when an overflow or underflow of position counter is detected.

When the position counter value matches the QPRC Maximum Position Register(QMPR) value, the operation of the revolution counter depends on the selected mode as follows:

- When the position counter is counted up in RC_Mode0(QCR.RCM[1:0]= 0b00), RC_Mode2(QCR.RCM[1:0]= 0b10) or RC_Mode3(QCR.RCM[1:0]= 0b11), the overflow flag(QICR.OFDF) is set to "1" and the position counter is reset.
- When the position counter is counted up in RC_Mode1(QCR.RCM[1:0]= 0b01), the overflow flag(QICR.OFDF) is set to "1". During this time, the position counter is not reset but is counted up.

The following gives an operation example where the QPRC Maximum Position Register(QMPR) is used in RC_Mode2(QCR.RCM[1:0]= 0b10).

During counting up:
When the position counter maximum value overflows to 0x0000, the revolution counter is counted up. During this time, the overflow flag(QICR.OFDF) is set to "1".

Example: If the QPRC Maximum Position Register(QMPR) is set to 18

Position counter	15	16	17	18	0	1	2
Revolution counter	1	1	1	1	2	2	2

During counting down:
When an underflow is detected with 0x0000 and when the value of Quad Counter Maximum Position Counter Register(QMPR) is reloaded to the position counter, the revolution counter is counted down. During this time, the underflow flag(QICR.UFDF) is set to "1".

Example: If the QPRC Maximum Position Register(QMPR) is set to 5

Position counter	4	3	2	1	0	5	4	3	2	1	0	5
Revolution counter	1	1	1	1	1	0	0	0	0	0	0	0xFFFF

Note:

- *The counting direction of position counter depends on the AIN and BIN external input signals only.*

3.6. Position Counter Reset Mask Function

The position counter reset mask function can be used only when RC_Mode0(QCR.RCM[1:0]= -b00) or RC_Mode3(QCR.RCM[1:0]= 0b11) is selected. This function operates regardless of setting of the position counter mode(PC_Mode1, PC_Mode2 or PC_Mode3).

The position counter reset mask function is executed in the following sequence.

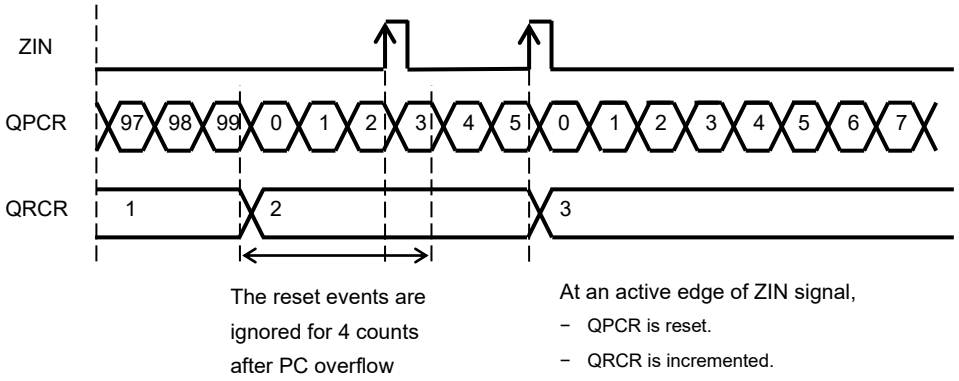
1. When an active event of ZIN signal, an overflow of position counter, or an underflow of position counter are detected, a value being set by the position counter reset mask bits(QCR.PCRM[1:0]) is set to the mask counter*1.
2. When the position counter is counted up or down in the same counting direction, the mask counter*1 is counted down.
   The position counter is reset only when the mask counter*1 is set to 0x0. Also, the revolution counter is not counted up or down.
   When a count inversion of the position counter is detected, the mask counter*1 is set to 0x0.
3. If the mask counter*1 is set to 0x0, the position counter is set to 0x0000 when an active edge of ZIN signal or an overflow of position counter is detected.

*1 : The number of times to mask both the reset of position counter and the counting up/down of revolution counter is counted. The masking continues until this counter value reaches 0x0.

The following gives an operation example where the position counter reset mask function is used in RC_Mode3(QCR.RCM[1:0]=0b11).

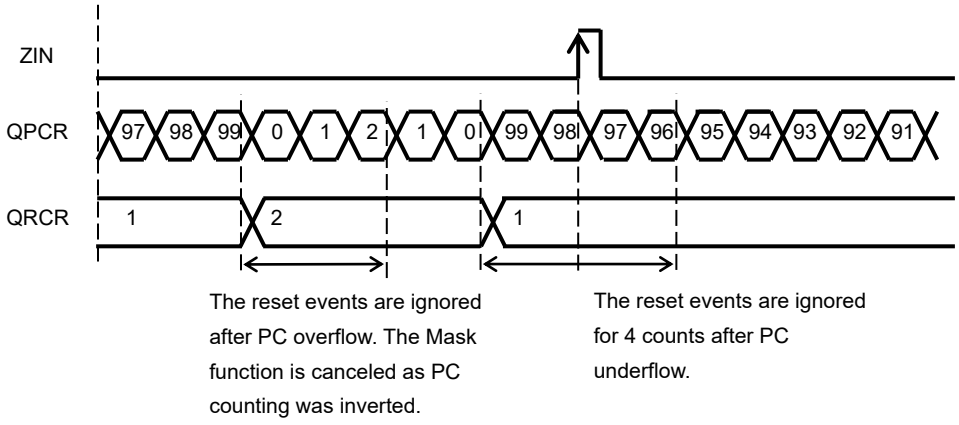
Example 1:
An active edge of ZIN signal is ignored for four(4) counts(QPRC=0 to 3) of position counter after occurrence of position counter overflow.

Figure 3-9 Position Counter Reset Mask Operation Example 1 (QMPR=99, QCR.PCRM[1:0]= 0b10, QCR.CGSC= 0)



Example 2:
An active edge of ZIN signal is ignored for four(4) counts(QPRC=99 to 96) of position counter after occurrence of position counter underflow following count inversion of position counter.

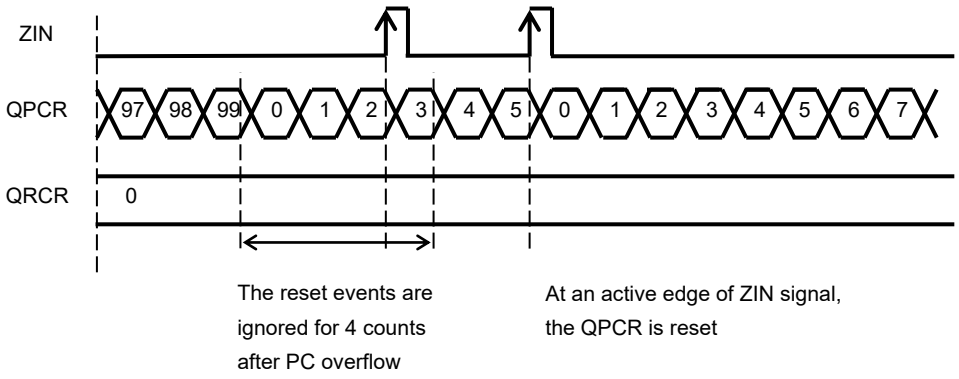
Figure 3-10 Position Counter Reset Mask Operation Example 2 (QMPR=99, QCR.PCRM[1:0]=0b10, QCR.CGSC= 0)



The following gives an operation example where the position counter reset mask function is used in RC_Mode0(QCR.RCM[1:0]= 0b00).

Example 3:
An active edge of ZIN signal is ignored for four(4) counts(QPRC=0 to 3) of position counter after occurrence of position counter overflow if the revolution counter is disabled.

Figure 3-11 Position Counter Reset Mask Operation Example 3 (QMPR=99, QCR.PCRM[1:0]= 0b10, QCR.CGSC= 0)



Notes:

- *While the position counter reset mask function is operating, the mask function is released and the position counter can be reset in the following conditions.*
  - The position counter mode bit(QCR.PCM[1:0]) is changed, or
  - The revolution counter mode bit(QCR.RCM[1:0]) is changed, or
  - The direction of the position counter is changed.
- *Even if an overflow or underflow of the position counter occurs without inversion of the position counter while the position counter reset mask function is operating in RC_Mode0(QCR.RCM[1:0]= 0b00) or RC_Mode3(QCR.RCM[1:0]= 0b11), the revolution counter is not counted up or down. However, if an overflow occurs, the position counter becomes 0x0000. If an underflow occurs, the QMPR is reloaded to the position counter. The overflow interrupt request flag bit(QICR.OFDF) or the underflow interrupt request flag bit(QICR.UFDF) is set to "1".*

4. Registers

This section explains the configuration and functions of the registers used for the QPRC(Quadrature Position/Revolution Counter).

Table 4-1 List of QPRC(Quadrature Position/Revolution Counter) Register

Abbreviation	Register Name	See
QPCR	QPRC Position Count Register	4.1
QRCR	QPRC Revolution Count Register	4.2
QPCCR	QPRC Position Counter Compare Register	4.3
QPRCR	QPRC Position and Revolution Counter Compare Register	4.4
QCR	QPRC Control Register	4.5
QECR	QPRC Extension Control Register	4.6
QICRL	Low-Order Bytes of QPRC Interrupt Control Register	4.7
QICRH	High-Order Bytes of QPRC Interrupt Control Register	4.8
QMPR	QPRC Maximum Position Register	4.9

Quad Counter

Offset Address	Register			
	+3	+2	+1	+0
0x000	QCxx_QRCR[H,W]		QCxx_QPCR[H,W]	
0x004	QCxx_QPRCR[H,W]		QCxx_QPCCR[H,W]	
0x008	QCxx_QICRH [B,H,W]	QCxx_QICRL [B,H,W]	QCxx_QMPR[H,W]	
0x00C	QCxx_QECR[B,H,W]		QCxx_QCR[B,H,W]	
0x010	-		-	
0x014	-		-	
0x018	-		-	
0x01C	-		-	
0x020 to 0x3FC	-			

4.1. Position Count Register(QPCR)

The QPRC Position Count Register(QPCR) indicates the position counter

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	QPCR[15:0]															
R/W Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W
Protection Attribute	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[bit15:0] QPCR[15:0]:
Reading this register reads out the current value of the position counter. While the position counter stops counting(QCR.PSTP= 1), the count value can be written to this register.

This register is set to 0x0000 in the following one of conditions.

- Reset
- A ZIN active edge is detected in the following conditions.
  - The ZIN function is set to the counter clear function(QCR.CGSC= 0) in RC_Mode1(QCR.RCM[1:0]= 0b01).
  - After the position counter has been incremented or decremented by the mask set value when the count inversion of the position counter is not detected where the ZIN function is set to the counter clear function(QCR.CGSC= 0) and the reset mask function of the position counter is valid(QCR.PCRM[1:0]= 0b01 or 0b10 or 0b11) in RC_Mode0(QCR.RCM[1:0]= 0b00) or RC_Mode3(QCR.RCM[1:0]= 0b11)
  - The ZIN function is set to the counter clear function(QCR.CGSC= 0) and the reset mask function of the position counter is invalid(QCR.PCRM[1:0]= 0b00) in RC_Mode0(QCR.RCM[1:0]= 0b00) or RC_Mode3(QCR.RCM[1:0]= 0b11).
- A position counter overflow is detected in the following conditions.
  - RC_Mode2(QCR.RCM[1:0]= 0b10)
  - After the position counter has been incremented or decremented by the mask set value when the count inversion of the position counter is not detected where the ZIN function is set to the counter clear function(QCR.CGSC= 0) and the reset mask function of the position counter is valid(QCR.PCRM[1:0]= 0b01 or 0b10 or 0b11) in RC_Mode0(QCR.RCM[1:0]= 0b00) or RC_Mode3(QCR.RCM[1:0]= 0b11)
  - The ZIN function is set to the counter clear function(QCR.CGSC= 0) and the reset mask function of the position counter is invalid(QCR.PCRM[1:0]= 0b00) in RC_Mode0(QCR.RCM[1:0]= 0b00) or RC_Mode3(QCR.RCM[1:0]= 0b11).
- 0x0000 is written to this QPCR while the position counter is under suspension(QCR.PSTP= 1).

The value of the QPRC Maximum Position Register(QMPR) is set to this register in the following condition.

- A position counter underflow is detected.

Notes:

- *Do not access the QPRC Position Count Register(QPCR) with a byte access instruction.*
- *After the count value was written to the QPRC Position Count Register(QPCR) while the position counter was under suspension(QCR.PSTP= 1) in RC_Mode0(QCR.RCM[1:0]= 0b00), RC_Mode1(QCR.RCM[1:0]= 0b01), or RC_Mode3(QCR.RCM[1:0]= 0b11), if a ZIN active edge is detected with the count function(QCR.CGSC= 0), the QPRC Position Count Register(QPCR) will be set to 0x0000.*
  To write the count value to the QPRC Position Count Register(QPCR), make the ZIN detection edge invalid(QCR.CGE[1:0]= 0b00) before writing it to the QPCR.

4.2. QPRC Revolution Count Register(QRCR)

The QPRC Revolution Count Register(QRCR) indicates the revolution counter.

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	QRCR[15:0]															
R/W Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W
Protection Attribute	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[bit15:0] QRCR[15:0] :

Reading this register reads out the current value of the revolution counter. While the revolution counter stops counting(QCR.RCM[1:0]= 0b00), the count value can be written to this register.

This register is set to 0x0000 in the following one of conditions.

- Reset
- 0x0000 is written to this register while the revolution counter is under suspension(QCR.RCM[1:0]= 0b00).

Notes:

- *Do not access the QPRC Revolution Count Register(QRCR) with a byte access instruction.*
- *As the direction of the position counter is not detected in PC_Mode0(QCR.PCM[1:0]= 0b00), the last position counter direction bit(QICR.DIRPC) becomes indefinite. Therefore, if the mode is changed from PC_Mode0(QCR.PCM[1:0]= 0b00) to another mode, when a ZIN active edge is detected before an AIN/BIN active edge is detected, the following operations apply.*
  - The position counter is reset if the mode is RC_Mode0(QCR.RCM[1:0]= 0b00),RC_Mode1(QCR.RCM[1:0]= 0b01), or RC_Mode3(QCR.RCM[1:0]= 0b11)
  - The revolution counter is not counted up or down.

4.3. QPRC Position Counter Compare Register(QPCCR)

The QPRC Position Counter Compare Register(QPCCR) is used to compare with the count value of the position counter.

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	QPCCR[15:0]															
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[bit15:0] QPCCR[15:0] : QPRC Position Counter Compare

If the value of this register matches that of the position counter, the QPRC position counter comparison match flag(QICR.QPCMF) is set to "1". This Compare Register can be used only to compare with the count value of the position counter.

Note:

- *Do not access the QPRC Position Counter Compare Register(QPCCR) with a byte access instruction.*

4.4. QPRC Position and Revolution Counter Compare Register(QPRCR)

The QPRC Position and Revolution Counter Compare Register(QPRCR) is used to compare with the selected count value of the position or revolution counter.

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	QPRCR[15:0]															
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[bit15:0] QPRCR[15:0] : QPRC Position and Revolution Counter Compare

Select whether to compare with the count value of the position or revolution counter using the RSEL bit of the QPRC Control Register(QCR). If the value of this register matches that of the position or revolution counter, the QPRC position and revolution counter comparison match flag(QICR.QPRCMF) is set to "1".

Note:

- *Do not access the QPRC Position and Revolution Counter Compare Register(QPRCR) with a byte access instruction.*

4.5. QPRC Control Register(QCR)

The QPRC Control Register(QCR) is used to specify the operation mode of the position counter or 16-bit revolution counter. It is also used to start or stop each counter.

Low-Order Bytes of QPRC Control Register(QCRL)

bit	7	6	5	4	3	2	1	0
Field	SWAP	RSEL	CGSC	PSTP	RCM[1:0]		PCM[1:0]	
R/W Attribute	R/W	R/W	R/W	R/W	R/W		R/W	
Protection Attribute	-	-	-	-	-		-	
Initial Value	0	0	0	0	00		00	

[bit7] SWAP : Swap Bit

This bit is used to swap the connections of the AIN input and BIN input to the position counter.

When this bit is set to "0", the AIN pin is used for the AIN input of the position counter, and the BIN pin is used for the BIN input of the position counter. When this bit is set to "1", the AIN pin is used for the BIN input of the position counter, and the BIN pin is used for the AIN input of the position counter.

Bit	Description
0	No swap
1	Swaps AIN and BIN inputs.

Note:

- *Change the swap bit(SWAP) when the position counter is disabled(PCM[1:0]= 0b00).*

[bit6] RSEL : Register Function Selection Bit

This bit is used to select whether to compare the value of the QPRC Position and Revolution Counter Compare Register(QPRCR) with that of the position or revolution counter.

Bit	Description
0	Compares the value of the QPRC Position and Revolution Counter Compare Register(QPRCR) with that of the position counter.
1	Compares the value of the QPRC Position and Revolution Counter Compare Register(QPRCR) with that of the revolution counter.

Note:

- *When the value of the position counter matches that of the QPRC Position Counter Compare Register(QPCCR) and also the value of the revolution counter matches that of the QPRC Position and Revolution Counter Compare Register(QPRCR), the PC match and RC match interrupt request flag bit(QICR: QPCNRCMF) is set to "1" regardless of the setting of this bit.*

[bit5] CGSC : Count Clear or Gate Selection Bit

This bit is used to select the function of the ZIN external pin.

When the counter clear function is enabled(QGSC= 0), the ZIN pin clears the position counter if the revolution count mode is set to RC_Mode0(RCM[1:0]= 0b00), RC_Mode1(RCM[1:0]= 0b01), or RC_Mode3(RCM[1:0]= 0b11). The CGE[1] and CGE[0] bits of the QCR register clear the position counter by selecting a valid edge of the ZIN pin and detecting the selected edge.

When the gate function is enabled(QGSC= 1), the ZIN pin controls the count operation of the position counter. The CGE[1] and CGE[0] bits of the QCR register count the position counter at the valid level of the ZIN pin.

Bit	Description
0	Counter clear function
1	Gate function

[bit4] PSTP : Position Counter Stop Bit

This bit is used to stop the position counter.

Bit	Description
0	Enables count operation.
1	Stops count operation.

[bit3:2] RCM[1:0] : Revolution Counter Mode Bits

These bits are used to select the count mode of the revolution counter and the reset mode of the position counter. For the effect on the position counter, see "3.2 Operation of Revolution Counter".

Bits	Description
00	Disables the revolution counter(RC_Mode0).
01	The revolution counter is counted up or down only with a ZIN active edge(RC_Mode1).
10	The revolution counter is counted up or down only when overflow or underflow is detected in the position counter that matches QMPR(RC_Mode2).
11	The revolution counter is counted up or down in two cases: a position counter overflow or underflow is detected and a ZIN active edge is detected(RC_Mode3).

[bit1:0] PCM[1:0] : Position Counter Mode Bits

These bits are used to select the count mode of the position counter.

Bits	Description
00	Disables the position counter(PC_Mode0) to stop it.
01	Up/down count mode(PC_Mode1) Increments the value with an AIN active edge and decrements it with a BIN active edge.
10	Phase difference count mode(PC_Mode2) Counts up if AIN is leading BIN and down if BIN is leading AIN.
11	Directional count mode(PC_Mode3) Counts up or down with the BIN active edge and AIN level.

Note:

- *As the direction of the position counter is not detected in PC_Mode0(PCM[1:0]= 0b00), the last position counter direction bit(QICR.DIRPC) becomes indefinite. Therefore, if the mode is changed from PC_Mode0(PCM[1:0]= 0b00) to another mode, when a ZIN active edge is detected before an AIN/BIN active edge is detected, the following operations apply.*
  - The position counter is reset if the mode is RC_Mode0(RCM[1:0]= 0b00), RC_Mode1(RCM[1:0]= 0b01), or RC_Mode3(RCM[1:0]= 0b11)
  - The revolution counter is not counted up or down

High-Order Bytes of QPRC Control Register(QCRH)

bit	15 14	13 12	11 10	9 8
Field	CGE[1:0]	BES[1:0]	AES[1:0]	PCRM[1:0]
R/W Attribute	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-
Initial Value	00	00	00	00

[bit15:14] CGE[1:0] : Detection Edge Selection Bits

These bits are used to select the detection edge when the ZIN external pin is used for the counter clear function(CGSC= 0). They are also used to select the detection level when the ZIN external pin is used for the gate function(CGSC= 1).

Bits	ZIN Used for Counter Clear Function (CGSC= 0)	ZIN Used for Gate Function (CGSC="1")
00	Disables edge detection.	Disables level detection.
01	Detects a falling edge.	Detects level "L".
10	Detects a rising edge.	Detects level "H".
11	Detects both rising and falling edges.	Disables level detection.

[bit13:12] BES[1:0] : BIN Detection Edge Selection Bits

These bits are used to select the detection edge of the BIN external pin.

Bits	Description
00	Disables edge detection.
01	Detects a falling edge.
10	Detects a rising edge.
11	Detects both rising and falling edges.

[bit11:10] AES[1:0] : AIN Detection Edge Selection Bits

These bits are used to select the detection edge of the AIN external pin.

Bits	Description
00	Disables edge detection.
01	Detects a falling edge.
10	Detects a rising edge.
11	Detects both rising and falling edges.

[bit9:8] PCRM[1:0] : Position Counter Reset Mask Bits

These bits are used to specify the period(mask time) to ignore the events shown below after detecting a position counter overflow or underflow or detecting a ZIN active edge.

- Position counter resetting
- Revolution counter increment or decrement

This mask function is released when the count direction of the position counter is changed, and restarts when a position counter overflow or underflow is detected or a ZIN active edge is detected.

Bits	Description
00	No reset mask
01	The position counter reset or a revolution counter count-up or -down events are ignored until the position counter changes twice.
10	The position counter reset or a revolution counter count-up or -down events are ignored until the position counter changes four times.
11	The position counter reset or a revolution counter count-up or -down events are ignored until the position counter changes eight times.

Notes:

- *The position counter reset mask function is available only in RC_Mode0(RCM[1:0]= 0b00) and RC_Mode3(RCM[1:0]= 0b11). This function operates regardless of the setting of the position counter mode(PC_Mode1, PC_Mode2, or PC_Mode3).*
- *While the position counter reset mask function is operating, the mask function is released and the position counter can be reset in the following conditions.*
  - When the position counter mode bit(PCM[1:0]) is changed
  - When the revolution counter mode bit(RCM[1:0]) is changed
  - When the direction of the position counter is changed

4.6. QPRC Extension Control Register(QECR)

The QPRC Extension Control Register(QECR) is used to select that the revolution counter is inside the count range, indicate that the revolution counter is outside the count range, or control whether or not to generate an interrupt when the revolution counter gets out of the range.

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved													ORNGIE	ORNGF	ORNGMD
R/W Attribute	R 0, W 0	R 0, W 0	R 0, W 0	R 0, W 0	R 0, W 0	R 0, W 0	R 0, W 0	R 0, W 0	R 0, W 0	R 0, W 0	R 0, W 0	R 0, W 0	R 0, W 0	R/W	R,W	R/W
Protection Attribute	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[bit15:3] Reserved : Reserved Bits

These bits must always be written to "0".

The read value is "0".

[bit2] ORNGIE : Outrange Interrupt Enable Bit

This bit is used to control whether or not to issue an interrupt notification to the MCU when the outrange interrupt request flag(ORNGF) is set to "1". When this bit is set to "1", an interrupt is generated if the value of the revolution counter gets out of the range(ORNGF= 1).

Bit	Description
0	Interrupt disabled
1	Interrupt enabled

[bit1] ORNGF : Outrange Interrupt Request Flag Bit

This flag indicates that the revolution counter is outside the count range.

If a positive number is selected as the outrange mode of the revolution counter(ORNGMD= 0), this flag is set to "1" when the revolution counter changes from 0x0001 to 0x0000 after counting down or when it changes from 0xFFFE to 0xFFFF after counting up.

If the 8K value is selected as the outrange mode of the revolution counter(ORNGMD= 1), this flag is set to "1" when the revolution counter changes from 0x8001 to 0x8000 after counting down or when it changes from 0x7FFE to 0x7FFF after counting up.

This flag can be only cleared to "0" in write mode. Setting "1" has no effect.

"1" is read by the read-modify-write access operation.

Bit	Description	
	Read	Write
0	Out of range is not detected.	Clears this bit.
1	Out of range is detected.	No effect.

[bit0] ORNGMD : Outrange Mode Selection Bit

This bit defines the outrange mode of the revolution counter.

Bit	Description
0	Selects a positive number(in the range from 0x0000 to 0xFFFF).
1	Selects the 8K value(in the range from 0x0000 to 0x7FFF).

4.7. Low-Order Bytes of QPRC Interrupt Control Register(QICRL)

The Low-Order Bytes of QPRC Interrupt Control Register(QICRL) are used to control a position counter overflow or underflow interrupt, zero index interrupt, QPRC position counter comparison match interrupt, or QPRC position and revolution counter comparison match interrupt.

bit	7	6	5	4	3	2	1	0
Field	ZIIF	OFDF	UFDF	OUZIE	QPRCMF	QPRCMIE	QPCMF	QPCMIE
R/W Attribute	R,W	R,W	R,W	R/W	R,W	R/W	R,W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

[bit7] ZIIF : Zero Index Interrupt Request Flag Bit

This flag is set to "1" when the position counter is reset by the ZIN input.

This flag can be only cleared to "0" in write mode. Setting "1" has no effect.

"1" is read by the read-modify-write access operation.

Bit	Description	
	Read	Write
0	Detects no zero index.	Clears this bit.
1	Detects zero index.	No effect.

Note:

- *The zero index interrupt request flag bit(ZIIF) is not set to "1" even if ZIN is used as the gate function(QCR.CGSC= 1) or the position counter is reset in RC_Mode2(QCR.RCM[1:0]= 0b10).*

[bit6] OFDF : Overflow Interrupt Request Flag Bit

This flag indicates that a position counter overflow occurs. When the set value of the QPRC Maximum Position Register(QMPR) matches the value of the position counter and the position counter is counted up, this bit is set to "1".

This flag can be only cleared to "0" in write mode. Setting "1" has no effect.

"1" is read by the read-modify-write access operation.

Bit	Description	
	Read	Write
0	Detects no overflow.	Clears this bit.
1	Detects overflow.	No effect.

[bit5] UFDF : Underflow Interrupt Request Flag Bit

This flag indicates that a position counter underflow occurs. When the position counter is 0x0000 and the position counter is counted down, this bit is set to "1".

This flag can be only cleared to "0" in write mode. Setting "1" has no effect.

"1" is read by the read-modify-write access operation.

Bit	Description	
	Read	Write
0	Detects no underflow.	Clears this bit.
1	Detects underflow.	No effect.

[bit4] OUZIE : Overflow, Underflow, or Zero Index Interrupt Enable Bit

This bit is used to control whether or not to issue an interrupt notification to the MCU when the overflow interrupt request flag bit(OFDF), underflow interrupt request flag bit(UFDF), or zero index interrupt request flag bit(ZIIF) is set to "1". When this bit is set to "1", an interrupt is generated if overflow is detected(OFDF= 1), underflow is detected(UFDF= 1), or zero index is detected(ZIIF= 1).

Bit	Description
0	Interrupt disabled
1	Interrupt enabled

[bit3] QPRCMF : PC and RC Match Interrupt Request Flag Bit

This flag indicates whether the value of the position counter matches that of the QPRC Position and Revolution Counter Compare Register(QPRCR) or the value of the revolution counter(QRCR) matches that of the QPRC Position and Revolution Counter Compare Register(QPRCR).

When the comparison between the position counter and QPRC Position and Revolution Counter Compare Register(QPRCR) is selected(QCR.RSEL= 0), this flag is set to "1" if the value of the position counter matches that of the QPRC Position and Revolution Counter Compare Register(QPRCR).

When the comparison between the revolution counter and QPRC Position and Revolution Counter Compare Register(QPRCR) is selected(QCR.RSEL= 1), this flag is set to "1" if the value of the revolution counter matches that of the QPRC Position and Revolution Counter Compare Register(QPRCR).

This flag can be only cleared to "0" in write mode. Setting "1" has no effect.

"1" is read by the read-modify-write access operation.

Bit	Description	
	Read	Write
0	Detects no comparison match with the QPRCR value.	Clears this bit.
1	Detects a comparison match with the QPRCR value.	No effect.

Notes:

- *If the register function selection bit(QCR.RSEL) is set to "0", the PC and RC match interrupt request flag bit(QPRCMF) is set to "1" immediately when the following one of conditions is satisfied.*
  - The mode is changed to PC_Mode1(QCR.PCM[1:0]= 0b01), PC_Mode2(QCR.PCM[1:0]= 0b10), or PC_Mode3(QCR.PCM[1:0]= 0b11) when the position counter is disabled(QCR.PCM[1:0]= 0b00) and the value of the position counter matches that of the QPRC Position and Revolution Counter Compare Register(QPRCR).
  - The value of the position counter matches that of the QPRC Position and Revolution Counter Compare Register(QPRCR) when data is written to the QPRC Position Count Register(QPCR) or QPRC Position and Revolution Counter Compare Register(QPRCR) in PC_Mode1(QCR.PCM[1:0]= 0b01), PC_Mode2(QCR.PCM[1:0]= 0b10), or PC_Mode3(QCR.PCM[1:0]= 0b11)

- *If the register function selection bit(QCR.RSEL) is set to "1", the PC and RC match interrupt request flag bit(QPRCMF) is set to "1" immediately when the following condition is satisfied.*
  - The value of the revolution counter matches that of the QPRC Position and Revolution Counter Compare Register(QPRCR) by writing data to the QPRC Position and Revolution Counter Compare Register(QPRCR) when the mode is RC_Mode1(QCR.RCM[1:0]= 0b01), RC_Mode2(QCR.RCM[1:0]= 0b10), or RC_Mode3(QCR.RCM[1:0]= 0b11).
  - The value of the revolution counter matches that of the QPRC Position and Revolution Counter Compare Register(QPRCR) by changing the mode from RC_Mode0(QCR.RCM[1:0]= 0b00) to another mode.

- *When the register function selection bit(QCR.RSEL) is changed, the PC and RC match interrupt request flag bit(QPRCMF) is set to "1" immediately if the following one of conditions is satisfied.*
  - The value of the revolution counter matches that of the QPRC Position and Revolution Counter Compare Register(QPRCR) when the register function selection bit(QCR.RSEL) is changed from "0" to "1" in the mode other than RC_Mode0(QCR.RCM[1:0]= 0b00).
  - The value of the position counter matches that of the QPRC Position and Revolution Counter Compare Register(QPRCR) when the register function selection bit(QCR.RSEL) is changed from "1" to "0" in the mode other than RC_Mode0(QCR.RCM[1:0]="00")

[bit2] QPRCMIE : PC and RC Match Interrupt Enable Bit

This bit is used to control whether or not to issue an interrupt notification to the MCU when the PC and RC match interrupt request flag(QPRCMF) is set to "1". When this bit is set to "1", an interrupt is generated if the value of the position or revolution counter matches that of the QPRC Position and Revolution Counter Compare Register(QPRCR)(QPRCMF= 1).

Bit	Description
0	Interrupt disabled
1	Interrupt enabled

[bit1] QPCMF : PC Match Interrupt Request Flag Bit

This flag indicates whether or not the value of the position counter matches that of the QPRC Position Counter Compare Register(QPCCR).

This flag is set to "1" if the value of the position counter matches that of the QPRC Position Counter Compare Register(QPCCR).

This flag can be only cleared to "0" in write mode. Setting "1" has no effect.

"1" is read by the read-modify-write access operation.

Bit	Description	
	Read	Write
0	Detects no comparison match with the QPCCR value.	Clears this bit.
1	Detects a comparison match with the QPCCR value.	No effect.

Note:

- *The PC match interrupt request flag bit(QPCMF) is set to "1" immediately when the following one of conditions is satisfied.*
  - The mode is changed to PC_Mode1(QCR.PCM[1:0]= 0b01), PC_Mode2(QCR.PCM[1:0]= 0b10), or PC_Mode3(QCR.PCM[1:0]= 0b11) when the position counter is disabled(QCR.PCM[1:0]= 0b00) and the value of the position counter matches that of the QPRC Position Counter Compare Register(QPCCR).
  - The value of the position counter matches that of the QPRC Position Counter Compare Register(QPCCR) by writing to the QPRC Position Count Register(QPCR) when the position counter stop bit(QCR.PSTP) is "1" and when the mode is PC_Mode1(QCR.PCM[1:0]= 0b01), PC_Mode2(QCR.PCM[1:0]= 0b10), or PC_Mode3(QCR.PCM[1:0]= 0b11).
  - The value of the position counter matches that of the QPRC Position Counter Compare Register(QPCCR) by writing to the QPRC Position Counter Compare Register(QPCCR) when the mode is PC_Mode1(QCR.PCM[1:0]= 0b01), PC_Mode2(QCR.PCM[1:0]= 0b10), or PC_Mode3(QCR.PCM[1:0]= 0b11).

[bit0] QPCMIE : PC Match Interrupt Enable Bit

This bit is used to control whether or not to issue an interrupt notification to the MCU when the PC match interrupt request flag(QPCMF) is set to "1".

When this bit is set to "1", an interrupt is generated if the value of the position counter matches that of the QPRC Position Counter Compare Register(QPCCR)(QPCMF= 1).

Bit	Description
0	Interrupt disabled
1	Interrupt enabled

4.8. High-Order Bytes of QPRC Interrupt Control Register(QICRH)

The High-Order Bytes of QPRC Interrupt Control Register(QICRH) are used to control a match between the position counter and QPCCR, a match between the revolution counter and QPRCR, and a count inversion interrupt. They are also used to indicate the direction of the position counter when the last underflow or overflow interrupt was detected or the last value of the position counter was changed.

bit	15	14	13	12	11	10	9	8
Field	Reserved		QPCNRCMF	QPCNRCMIE	DIROU	DIRPC	CDCF	CDCIE
R/W Attribute	R0,W0		R,W	R/W	R,WX	R,WX	R,W	R/W
Protection Attribute	-		-	-	-	-	-	-
Initial Value	00		0	0	0	0	0	0

[bit15:14] Reserved : Reserved Bits

These bits must always be written to "0".

The read value is "0".

[bit13] QPCNRCMF : PC Match and RC Match Interrupt Request Flag Bit

This flag indicates whether or not the value of the position counter matches that of the QPRC Position Counter Compare Register(QPCCR) and the value of the revolution counter matches that of the QPRC Position and Revolution Counter Compare Register(QPRCR).

This flag is set to "1" when the value of the position counter matches that of the QPRC Position Counter Compare Register(QPCCR)(QPCMF= 1) and the value of the revolution counter matches that of the QPRC Position and Revolution Counter Compare Register(QPRCR).

This flag can be only cleared to "0" in write mode. Setting "1" has no effect.

"1" is read by the read-modify-write access operation.

Bit	Description	
	Read	Write
0	Detects no match.	Clears this bit.
1	Detects a match.	No effect.

Note:

- *The PC match and RC match interrupt request flag bit(QPCNRCMF) is set to "1" immediately when the following one of conditions is satisfied.*
  - The mode is changed to PC_Mode1(QCR.PCM[1:0]= 0b01), PC_Mode2(QCR.PCM[1:0]= 0b10), or PC_Mode3(QCR.PCM[1:0]= 0b11) when the position counter is disabled(QCR.PCM[1:0]= 0b00) and the revolution counter is in the mode other than RC_Mode0(QCR.RCM[1:0]= 0b00) while the value of the position counter matches that of the QPRC Position Counter Compare Register(QPCCR) and the value of the revolution counter matches that of the QPRC Position and Revolution Counter Compare Register(QPRCR).
  - The value of the position counter matches that of the QPRC Position Counter Compare Register(QPCCR) when data is written to the QPRC Position Count Register(QPCR) or QPRC Position Counter Compare Register(QPCCR) where the value of the revolution counter matches that of the QPRC Position & Revolution Counter Compare Register(QPRCR) when the mode is PC_Mode1(QCR.PCM[1:0]= 0b01), PC_Mode2(QCR.PCM[1:0]= 0b10), or PC_Mode3(QCR.PCM[1:0]= 0b11) and the revolution counter is in the mode other than RC_Mode0(QCR.RCM[1:0]= 0b00).
  - The value of the revolution counter matches that of the QPRC Position and Revolution Counter Compare Register(QPRCR) when the data is written to the QPRC Position and Revolution Counter Compare Register(QPRCR) in the mode other than RC_Mode0(QCR.RCM[1:0]= 0b00) where the specified value matches that of the QPRC Position Count Register(QPCR) or QPRC Position Counter Compare Register(QPCCR) in PC_Mode1(QCR.PCM[1:0]= 0b01), PC_Mode2(QCR.PCM[1:0]= 0b10), or PC_Mode3(QCR.PCM[1:0]= 0b11).
  - The value of the revolution counter matches that of the QPRC Position and Revolution Counter Compare Register(QPRCR) when the mode is changed from RC_Mode0(QCR.RCM[1:0]= 0b00) to another mode where the specified value matches that of the QPRC Position Count Register(QPCR) or QPRC Position Counter Compare Register(QPCCR) in PC_Mode1(QCR.PCM[1:0]= 0b01), PC_Mode2(QCR.PCM[1:0]= 0b10), or PC_Mode3(QCR.PCM[1:0]= 0b11).
  - This bit is set to "1" when the value of the position counter matches that of the QPRC Position Counter Compare Register(QPCCR) and the value of the revolution counter matches that of the QPRC Position and Revolution Counter Compare Register(QPRCR) regardless of the setting of the register function selection bit(QCR.RSEL).

[bit12] QPCNRCMIE : PC Match and RC Match Interrupt Enable Bit

This bit is used to control whether or not to issue an interrupt notification to the MCU when the PC match and RC match interrupt request flag(QPCNRCMF) is set to "1".

When this bit is set to "1", an interrupt is generated if the value of the position counter matches that of the QPRC Position Counter Compare Register(QPCCR) and the value of the revolution counter matches that of the QPRC Position and Revolution Counter Compare Register(QPRCR)(QPCNRCMF= 1).

Bit	Description
0	Interrupt disabled
1	Interrupt enabled

[bit11] DIROU : Last Position Counter Flow Direction Bit

This bit indicates the direction of the position counter when the last position counter overflow or underflow was detected.

Bit	Description
0	The position counter was incremented.
1	The position counter was decremented.

[bit10] DIRPC : Last Position Counter Direction Bit

This bit indicates the count direction when the position counter was last changed.

Bit	Description
0	The position counter was incremented.
1	The position counter was decremented.

Note:

- *As the direction of the position counter is not detected in PC_Mode0(QCR.PCM[1:0]= 0b00), the last position counter direction bit(QICR.DIRPC) becomes indefinite. Therefore, if the mode is changed from PC_Mode0(QCR.PCM[1:0]= 0b00) to another mode, when a ZIN active edge is detected before an AIN/BIN active edge is detected, the following operations apply.*
  - The position counter is reset if the mode is RC_Mode0(QCR.RCM[1:0]= 0b00), RC_Mode1(QCR.RCM[1:0]= 0b01), or RC_Mode3(QCR.RCM[1:0]= 0b11)
  - The revolution counter is not counted up or down

[bit9] CDCF : Count Inversion Interrupt Request Flag Bit

This bit indicates whether or not the position counter inverted the count direction.

This bit is set to "1" when the position counter inverts the count direction. Inverting the count direction means that the counter counts down at the next counting after counting up, or the counter counts up at the next counting after counting down.

This flag can be only cleared to "0" in write mode. Setting "1" has no effect.

"1" is read by the read-modify-write access operation.

Bit	Description	
	Read	Write
0	Does not invert the count direction of the position counter.	Clears this bit.
1	Inverts the count direction of the position counter at least once.	No effect.

Note:

- *As the direction of the position counter is not detected in PC_Mode0(QCR.PCM[1:0]= 0b00), the last position counter direction bit(QICR.DIRPC) becomes indefinite. Therefore, after the mode is changed from PC_Mode0(QCR.PCM[1:0]= 0b00) to another mode, even if an AIN/BIN active edge is detected and the direction of the position counter is inverted, the count inversion interrupt request flag bit(QICR.CDCF) is not set to "1".*

[bit8] CDCIE : Count Inversion Interrupt Enable Bit

This bit is used to control whether or not to issue an interrupt notification to the MCU when the count inversion interrupt request flag(CDCF) is set to "1".

When this bit is set to "1", an interrupt is generated if the count direction of the position counter is inverted(CDCF= 1).

Bit	Description
0	Interrupt disabled
1	Interrupt enabled

4.9. QPRC Maximum Position Register(QMPR)

The QPRC Maximum Position Register(QMPR) is used to specify the maximum value of the position counter.

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	QMPR[15:0]															
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

[bit15:0] QMPR[15:0] :

When the set value of the QPRC Maximum Position Register(QMPR) matches the value of the position counter and the position counter is counted up, a position counter overflow is detected(QICR.OFDF= 1). The set value of the QPRC Maximum Position Register(QMPR) is reloaded to the position counter if a position counter underflow is detected(QICR.UFDF= 1).

Note:

- *Do not access the QPRC Maximum Position Register(QMPR) with a byte access instruction.*

CHAPTER 48: External Interrupt Capture

This chapter describes functions and operations of the EICU module.

1. **Overview**
2. **Configuration**
3. **Explanation of Operation**
4. **Setting Procedure Examples**
5. **Registers**

EICU-TXXPT03P01R01L06-E1-XX

1. Overview

This section describes the features and the block diagram of EICU module.

Features of External Interrupt Capture Unit (EICU)

The EICU module samples the interrupt pin level in isochronous steps after being triggered by an External Interrupt event on that pin. It can be used to record the bit stream that causes the External Interrupt event like CAN wakeup. Features of the EICU are:

- It supports the capture of events among 32 External Interrupt Pins (Pin INTk)
- The capture on any interrupt pin is maskable
- It has a 6-bit linearly programmable prescaler to provide a sample frequency range of 500 Hz to 16 MHz
- Software can read the sample registers and enable for the next observation
- Software can also read the channel number being observed
- It supports the Peripheral Protection Unit (PPU)

DMA and Interrupts

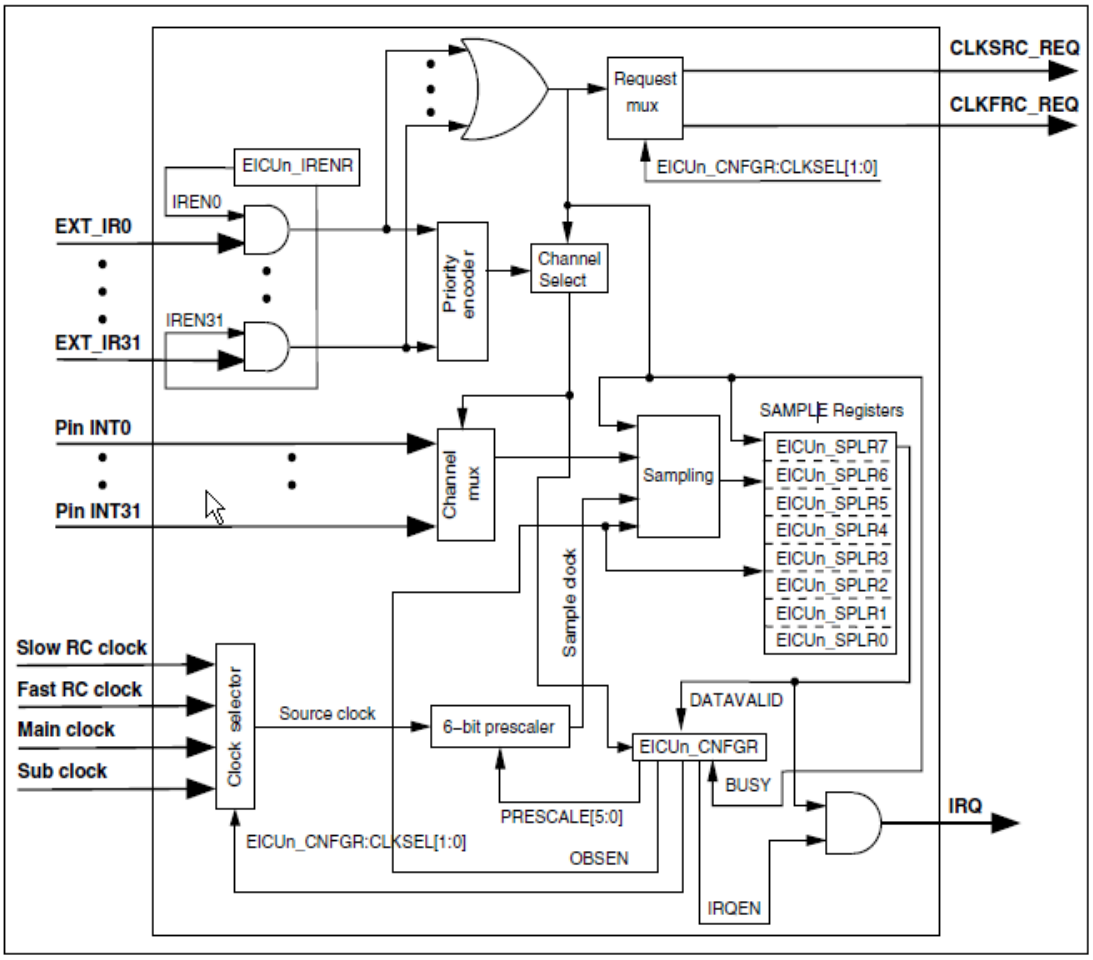
This module does not generate any DMA requests.

An interrupt is generated when valid data is available in the sample registers.

2. Configuration

Shown below is a block diagram of the EICU module.

Figure 2-1 Block Diagram



3. Explanation of Operation

This section describes operation of EICU.

The following section explains the top level block diagram of EICU and its operation. The block diagram of the EICU is shown in Figure 2-1.

Priority Encoder and Channel Select

EXT_IRk inputs come from the External Interrupts module and are the latch output of the first event in the corresponding Pin INTk. The channel select stores the channel (pin) number of the enabled channel (specified by EICUn_IRENR register) which had the first event. It is stored in EICUn_CNFGR:OBSCH[4:0]. In the event of a simultaneous event, the channel number is stored based on priority. Channel 0 has the highest priority and channel 31 has the lowest.

Clock Request

With the first event on any of the enabled interrupt pins, Pin INTk (EXT_IRk = 1), a clock request is sent to the clock controller block only if EICUn_CNFGR:CLKSEL[1:0] is programmed for Slow or Fast CR clock. If the EICUn_CNFGR:CLKSEL[1:0] is programmed for Main or Sub clock, it is assumed that the clocks are already available.

Note:

- *During power saving modes (which disables the Main and/or Sub clock), only the CR clock shall be used with the EICU. Between the first event in any of the channels and the arrival of the selected clock(EICUn_CNFGR:CLKSEL[1:0] = Slow CR clock/Fast CR clock), there can be a possible loss of levels on the interrupt pins (Pin INTk). The first activated interrupt pin (enabled pin) will be selected and sampled. For proper sampling, four times oversampling is suggested. This means, after recognizing an External Interrupt event, the fast CR clock must be started.*

Clock Settings

This block selects between the following four source clocks based on EICUn_CNFGR:CLKSEL[1:0] programming:

- Slow CR clock
- Fast CR clock
- Main clock and
- Sub clock

Prescaler bits scale the selected source clock based on EICUn_CNFGR:PRESCALE[5:0] programming and generate a sample clock.

Signal Sampling and Storing

This module captures 256 samples of the selected channel (Pin INTk) based on the channel select. The sampling is done with respect to the sample clock and the data is stored in eight 32-bit Sample Registers (SPLR). During the sampling period, the EICUn_CNFGR:BUSY bit is set and it is reset with EICUn_CNFGR:DATAVALID = 1. The EICUn_CNFGR:DATAVALID bit is set when 256 samples are collected in the SPLR. When EICUn_CNFGR:DATAVALID = 1, the interrupt (IRQ) is also sent to the host in case EICUn_CNFGR:IRQEN is set.

Enabling/Disabling of the Capture Function

Using the EICUn_CNFGR:OBSEN bit, the sampling function of the External Interrupt Pins can be enabled or disabled. If EICUn_CNFGR:OBSEN = 1, sampling starts with an event on any one of the

enabled External Interrupt channels. If EICUn_CNFGR:OBSEN = 0, the SPLR are flushed, and the EICUn_CNFGR:DATAVALID and EICUn_CNFGR:BUSY bits are reset.

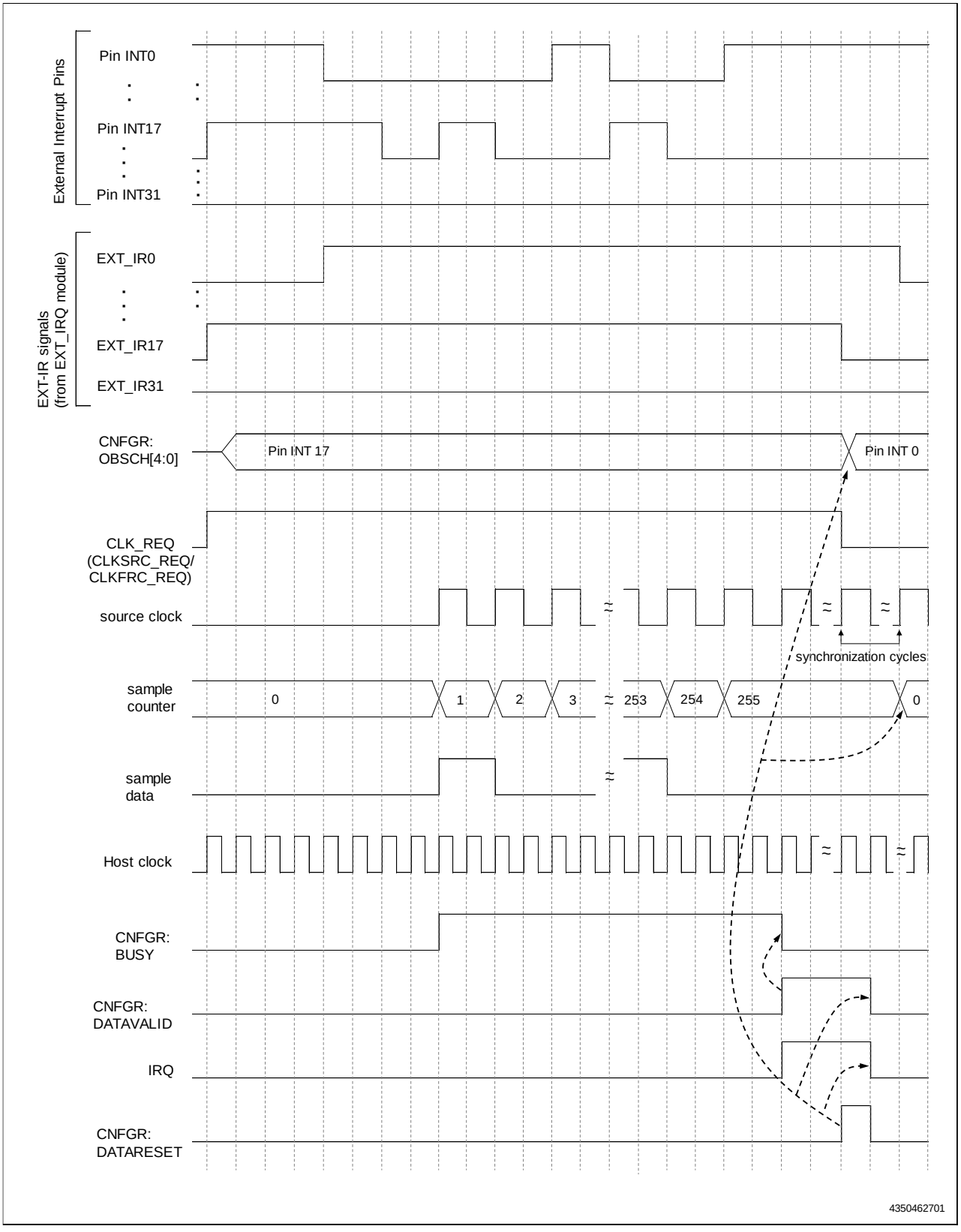
Example Operation of EICU

The timing diagram in Figure 3-1 typically shows the operation of the EICU. Here the assumption is that all the External Interrupt Pins are enabled for event monitoring and EICUn_CNFGR:OBSEN is set for the entire period. Pin INT17 has the first event followed by Pin INT0. All other interrupt pins are inactive. With the event on Pin INT17, a clock request is generated (EICUn_CNFGR:CLKSEL[1:0] = Slow CR/Fast CR clock) and channel select is updated to Pin INT17. Even in the event of another event on Pin INT0, channel select is not updated with Pin INT0. EICUn_CNFGR:BUSY is set when sampling starts and is reset once EICUn_CNFGR:DATAVALID is set. EICUn_CNFGR:DATAVALID is set after 256 samples have been collected and stored in the Sample Registers. An interrupt is sent to the host once EICUn_CNFGR:DATAVALID is set (assuming EICUn_CNFGR:IRQEN = 1) and it is de-asserted with EICUn_CNFGR:DATAVALID = 0. With EICUn_CNFGR:DATARESET = 1, EICUn_CNFGR:DATAVALID is de-asserted.

Note:

- *External Interrupt generation and clear logic is independent of the EICU.*

Figure 3-1 Timing Chart

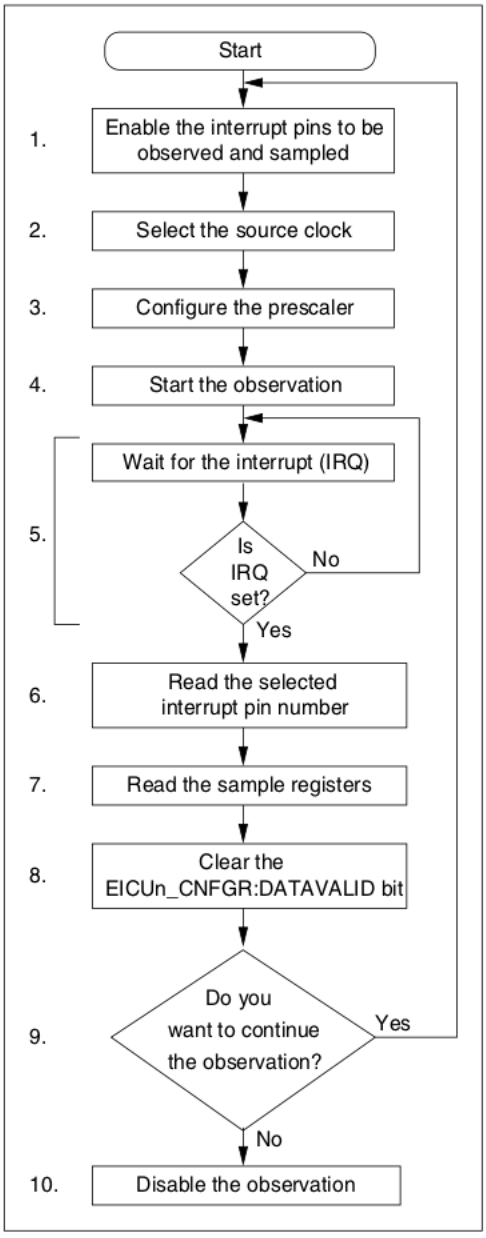


4. Setting Procedure Examples

This section describes how to use the EICU.

Shown below are setting procedure examples of EICU:

Figure 4-1 Procedure Example Flow



General steps a programmer shall follow while using the EICU:

- After the system reset, set the corresponding bits in EICUn_IRENR register for the interrupt pins to be observed and sampled. By default all the interrupt pins are disabled for observation. To sample any interrupt pin, the corresponding enable for that interrupt should also be enabled in the External Interrupt block. Event detection (signal detection) on the interrupt pin depends on the External Interrupt block settings (falling/rising edge, high/low level). For more details, refer to External Interrupt chapter.

- Program the EICUn_CNFGR:CLKSEL[1:0] bits to select the desired source clock from Slow CR clock, Fast CR clock, Main clock, and Sub clock. Note that at reset the Slow CR clock is selected, but by default Slow RC Osc is disabled. Also requirement of glitch free clock switching is that both (old and new) clock sources to be enabled at the time of EICUn_CNFGR:CLKSEL[1:0] change. So, before switching from Slow RC Osc clock to Main clock, the user needs to ensure that both oscillators are enabled. Once switching is done, the old clock (Slow RC Osc) can be disabled. However, the Main Osc still needs to be enabled during Power Saving State (PSS) so that it can sample the data at wakeup.

- Set the EICUn_CNFGR:PRESCALE[5:0] bits to select the appropriate scaling factor for the source clock.
  - Sampling frequency = source frequency/(EICUn_CNFGR:PRESCALE[5:0] +1)
  - Sampling frequency should be preferably four times faster than the fastest baud rate available amongst the enabled External Interrupt Pins. Source clock and prescaler values should be programmed accordingly.
- Set the EICUn_CNFGR:OBSEN to start the observation.

- With the first event in any of the enabled interrupt pins (EXT_IRk = 1), the clock is requested (CLKSRC_REQ/CLKFRC_REQ), if the EICUn_CNFGR:CLKSEL[1:0] is programmed for Slow CR clock or Fast CR clock. After the arrival of the source clock (refer to the datasheet for the time required for the Slow RC and Fast CR clock), the sampling frequency is calculated based on the prescaler setting. The External Interrupt Pin number with the first event is stored in EICUn_CNFGR:OBSCH[4:0] (channel select). The Pin INTk corresponding to the selected channel is sampled for 256 samples at the sampling frequency and the samples are stored in the SPLR. After capturing 256 samples, EICUn_CNFGR:DATAVALID bit is set. Simultaneously the interrupt (IRQ) is also set if EICUn_CNFGR:IRQEN = 1. The user should wait for EICUn_CNFGR:DATAVALID = 1 to read the SPLR. The clock request is de-asserted after the EICUn_CNFGR:DATAVALID bit is set.

- The selected channel number can be read from EICUn_CNFGR:OBSCH[4:0]. In the event of simultaneous events in more than one enabled External Interrupt Pin, priority is always given to the lower channel number. On other words Pin INT0 has the highest priority and Pin INT31 has the lowest priority.

- After the arrival of the interrupt, the user can read the data from the 32-bit EICUn_SPLR0 to 7 registers.

- The user should set EICUn_CNFGR:DATARESET to clear EICUn_CNFGR:DATAVALID and subsequently the interrupt (IRQ) after reading the EICUn_SPLR0 to 7 registers. To start a new observation, EICUn_CNFGR:DATAVALID should be always cleared. EICUn_CNFGR:DATAVALID is cleared only when the user sets the EICUn_CNFGR:DATARESET bit or EICUn_CNFGR:OBSEN is cleared

- To continue the observation with the same set of enabled Pin INTk channels, the source clock and prescaler settings, go directly to step 5.

- If you want to change any programmed values or want to discontinue observation, disable the EICUn_CNFGR:OBSEN bit. The observation and sampling process is inactive when the EICUn_CNFGR:OBSEN bit is "0" or EICUn_CNFGR:DATAVALID is "1"

Example Usage of EICU for CAN

In the event that wake-up via CAN is required and the wake-up message needs to be detected, the EICU can be used. Based on the assumption that the MCU is in low-power mode, the steps below are needed to configure the EICU for wake-up using CAN and must be carried out before the MCU goes to low-power mode:

- Enable the CAN Rx interrupt in the External Interrupt block for interrupt generation according to the CAN bus dominant bit setting.
- Enable observation of the interrupt pin (EICUn_IRENR) connected to the CAN RX pin.
- Select the appropriate capture frequency depending on CAN bit rate.: For example, if CAN is working at 1 Mbps, the sampling frequency should be configured for 4 MHz.
- Enable observation by setting the EICUn_CNFGR:OBSEN bit.

Operation of the EICU

- The device enters low-power state.
- In the event that a wake-up message is received, the External Interrupt block will detect an event on the CAN RX interrupt pin.
- This triggers a wake up sequence of the MCU and EICU.
- After the CR clock is operable (refer to the datasheet for time required for Slow RC and Fast CR clock), capturing the CAN bit stream starts.
- The interrupt pin is sampled and the 256 samples are stored in the SPLR (EICUn_SPLR0 to 7).
- After the 256 samples have been collected, the EICUn_CNFGR:DATAVALID bit will be set and an interrupt is generated if enabled.
- After the device and application has finished the wake-up procedure, the application can check the capture values and recalculate the received message.

5. Registers

A list of registers for EICU is shown:

The EICU module has various registers that enable the channels for observation, configure the prescaler and source clock, and store the sample data. It also has a status register to update the data validity status and the channel number under observation. All registers in the EICU module are explained in this section.

The suffix 'n' in the register name indicates that the register is in instance 'n' of the module.

The following registers are available for each instance of EICU:

- Configuration Register (EICUn_CNFGR)
- External Interrupt Pin Enable Register (EICUn_IRENR)
- Sample Registers 0 to 7 (EICUn_SPLR0 to 7)

Table 5-1 EXTERNAL INTERRUPT Capture Unit Registers

Abbreviated Register Name	Register Name	Reference
EICUn_CNFGR	Configuration Register	5.1
EICUn_IRENR	External Interrupt Pin Enable Register	5.2
EICUn_SPLR0 to 7	Sample Registers 0 to 7	5.3

Table 5-2 Register Map

Offset	Register Name/Initial Value
0x0000_0000	EICUn_CNFGR 00000000_00000000_00000000_00000000
0x0000_0004	EICUn_IRENR 00000000_00000000_00000000_00000000
0x0000_0008	EICUn_SPLR0 00000000_00000000_00000000_00000000
0x0000_000C	EICUn_SPLR1 00000000_00000000_00000000_00000000
0x0000_0010	EICUn_SPLR2 00000000_00000000_00000000_00000000
0x0000_0014	EICUn_SPLR3 00000000_00000000_00000000_00000000
0x0000_0018	EICUn_SPLR4 00000000_00000000_00000000_00000000
0x0000_001C	EICUn_SPLR5 00000000_00000000_00000000_00000000
0x0000_0020	EICUn_SPLR6 00000000_00000000_00000000_00000000
0x0000_0024	EICUn_SPLR7 00000000_00000000_00000000_00000000

5.1. Configuration Register (EICUn_CNFGR)

This register is used to configure the module for its prescaler value and source clock. It also gives information concerning the sampling status, data validity in the Sampling Registers (SPLR) and the current interrupt pin being sampled. An interrupt (IRQ) can also be set along with the EICUn_CNFGR: DATAVALID bit once 256 samples are stored in the SPLR registers. EICUn_CNFGR:OBSCH[4:0] shows the current interrupt pin being sampled. Writing "1" to EICUn_CNFGR:DATARESET clears the EICUn_CNFGR:DATAVALID.

bit	31	30	29	28	27	26	25	24
Field	Reserved					IRQEN	OBSEN	DATARESET
R/W Attribute	R0,WX					R/W	R/W	R0,W
Protection Attribute	-							
Initial Value	00000					0	0	0

bit	23	22	21	20	19	18	17	16
Field	DATAVALID	BUSY	Reserved	OBSCH[4:0]				
R/W Attribute	R,WX	R,WX	R0,WX	R,WX				
Protection Attribute	-							
Initial Value	0	0	0	00000				

bit	15	14	13	12	11	10	9	8
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	-							
Initial Value	00000000							

bit	7	6	5	4	3	2	1	0
Field	PRESCALE[5:0]						CLKSEL[1:0]	
R/W Attribute	R/W						R/W	
Protection Attribute	-							
Initial Value	000000						00	

[bit31:27] Reserved: Reserved Bits

[bit26] IRQEN: Interrupt Enable

This bit enables the interrupt from the module.

The interrupt is set when both the EICUn_CNFGR:DATAVALID and IRQEN bits are high.

"0": Interrupt is disabled

"1": Interrupt is enabled

[bit25] OBSEN: Observation Enable

This bit is used to enable the observation and sampling of the available 32 External Interrupt Pins. When the OBSEN bit is low, ongoing sampling is stopped immediately. With this bit going high, new sampling is started from 0 to 255.

"0": Observation and sampling is disabled

"1": Observation and sampling is enabled

[bit24] DATARESET: Data Reset

This bit is used to reset the EICUn_CNFGR:DATAVALID bit.

"0": No effect

"1": EICUn_CNFGR:DATAVALID bit is reset

Reading this bit always returns "0".

[bit23] DATAVALID: Data Valid

This bit is set when the data in the EICUn_SPLR is valid (256 samples are collected).

"0": Data is not valid

"1": Data is valid,

As long as DATAVALID is "1", sampling of External Interrupt Pin (Pin INTk) is disabled (no further sample request is processed).

[bit22] BUSY: Sampling Status

This bit shows the sampling status of EICUn_CNFGR:OBSCH (Pin INTk). It is reset with EICUn_CNFGR:DATAVALID bit = 1 or EICUn_CNFGR:OBSEN bit = 0.

"0": Sampling of the observed channel is not ongoing

"1": Sampling of observed channel is ongoing,

[bit21] Reserved: Reserved Bits

[bit20:16] OBSCH: Observed Channel

These bits specify the External Interrupt Pin (Pin INTk) number being observed and sampled (out of the 32 available interrupt pins). Observed channel should be read only when

EICUn_CNFGR:DATAVALID bit is "1".

[bit15:8] Reserved: Reserved Bits

[bit7:2] PRESCALE: Prescale

These bits decide the prescaler value for deriving sample clock from source clock. It can be programmed from 0 to 63. The sample clock calculation is shown in the Section ' Setting Procedure Examples '.

prescaler value = PRESCALE + 1

sample clock = source clock/(prescaler value)

[bit1:0] CLKSEL: Clock Select

These bits select the source clock for sampling.

"00": Slow RC oscillator clock

"01": Fast RC oscillator clock

"10": Main oscillator clock

"11": Sub oscillator clock

5.2. External Interrupt Pin Enable Register (EICUn_IRENR)

The External Interrupt Pin Enable Register bits enable/disable the respective External Interrupt pins (Pin INTk) to be observed. In the event of simultaneous events in more than one enabled interrupt pin, priority is always given to the lower pin number, i.e. Pin INT0 has the highest priority and Pin INT31 has the lowest priority. All 256 samples are taken from one single interrupt pin. By default all the interrupt pins are disabled for observation.

bit	31							0
Field	IREN[31:0]							
R/W Attribute	R/W							
Protection Attribute	-							
Initial Value	00000000_00000000_00000000_00000000							

[bit31:0] IREN: External Interrupt Pin Observe Enable

These bits enable/disable the observation of the corresponding External Interrupt Pin.

"0": Corresponding Pin INTk is disabled for observation and sampling

"1": Corresponding Pin INTk is enabled for observation and sampling

5.3. Sample Registers 0 to 7 (EICUn_SPLR0 to 7)

These registers contain the sampled data of the interrupt pin (Pin INTk) captured. There are eight such 32-bit registers, storing a total of 256 samples. These registers act as a 256 bit shift register and the data in these registers are valid only when EICUn_CNFGR:DATAVALID bit is set (after collection of the 256 samples). Only EICUn_SPLR0 is described here. Other registers (i.e. EICUn_SPLR1, EICUn_SPLR2, EICUn_SPLR3, EICUn_SPLR4, EICUn_SPLR5, EICUn_SPLR6, and EICUn_SPLR7) have similar bit fields.

bit	31							0
Field	SPL[31:0]							
R/W Attribute	R,WX							
Protection Attribute	-							
Initial Value	00000000_00000000_00000000_00000000							

[bit31:0] SPL: Sample 0 Register

These bits store the level of the interrupt pin being sampled at isochronous steps.

Note:

- *Sample 0 to Sample 7 registers store 256 samples of the levels of the enabled interrupt channel in isochronous steps. The first sample will be stored in EICUn_SPLR0[0] and the last sample will be stored in EICUn_SPLR7[31].*

CHAPTER 49: CRC (Cyclic Redundancy Check)

This chapter explains the functions and operations of the CRC.

1. **Overview**
2. **Configuration and Diagram**
3. **Operations**
4. **Registers**

1. Overview

The CRC (Cyclic Redundancy Check) is an error detection system. The CRC code is a remainder after an input data string is divided by the pre-defined generator polynomial, assuming the input data string is a high order polynomial. Ordinarily, a data string is suffixed by a CRC code when being sent, and the received data is divided by a generator polynomial as described above. If the received data is dividable, it is judged to be correct.

CRC Functions

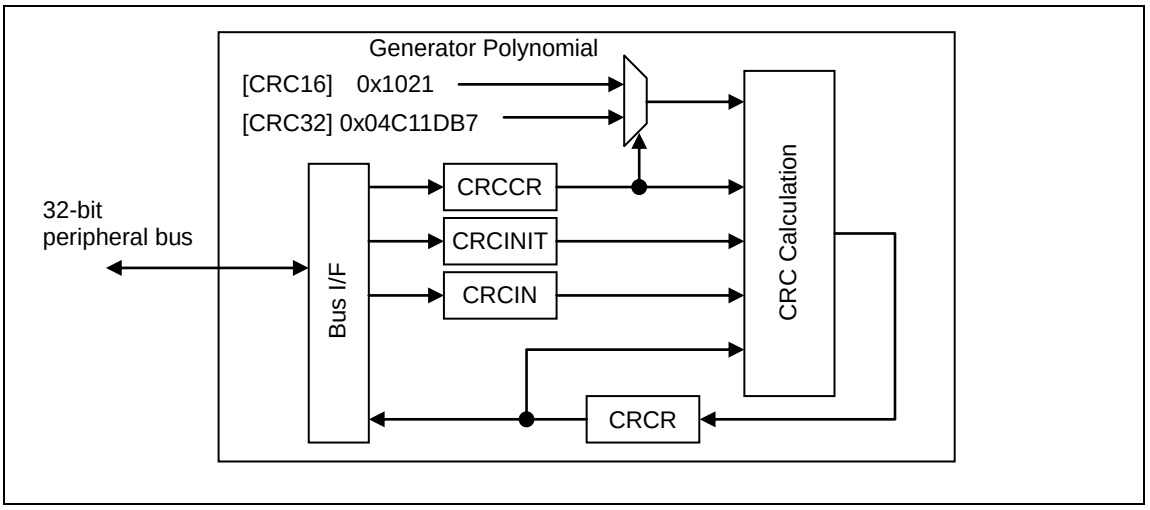
This module enables the calculation in both CCITT CRC16 and IEEE-802.3 CRC32. In this module, the generator polynomial is fixed to the numeric values for those two modes; therefore, the CRC value based on other generator polynomials cannot be calculated.

- CCITT CRC16 generator polynomial: 0x1021
- IEEE-802.3 CRC32 generator polynomial: 0x04C11DB7

2. Configuration and Diagram

Figure 2-1 shows the CRC configuration.

Figure 2-1 CRC Configuration



- CRCCR (CRC Control Register)
  Used to control CRC calculation.
- CRCINIT (Initial Value Register)
  Used to specify the initial values for CRC calculation.
- CRCIN (Input Data Register)
  Used to set input data for CRC calculation.
- CRCR (CRC Register)
  Used to output the CRC calculation result.
- CRC Calculation
  A circuit to perform CRC calculation.

3. Operations

This section provides an overview of CRC operations.

CRC Definition

[CCITT CRC16 Standard]

Generator polynomial	0x1021	(CRCCR.CRC32= 0)
Initial value	0xFFFF	
Final XOR value	0x0000	(CRCCR.FXOR= 0)
Bit order	MSB First	(CRCCR.LSBFST= 0)
Output bit order	MSB First	(CRCCR.CRCLSF= 0)

(The input-output byte order can be specified arbitrarily.)

[IEEE-802.3 CRC32 Ethernet Standard]

Generator polynomial	0x04C11DB7	(CRCCR.CRC32= 1)
Initial value	0xFFFFFFFF	
Final XOR value	0xFFFFFFFF	(CRCCR.FXOR= 1)
Bit order	LSB First	(CRCCR.LSBFST= 1)
Output bit order	LSB First	(CRCCR.CRCLSF= 1)

(The input-output byte order can be specified arbitrarily.)

Reset Operations

When resetting, the Initial Value Register (CRCINIT) and CRC Register (CRCR) are set to 0xFFFFFFFF. Other registers are cleared to "0".

Initialization

Initializing with the initialization bit (CRCCR.INIT) loads the value of the Initial Value Register to the CRC Register (CRCR).

Processing Byte and Bit Orders

The following shows how to process byte and bit orders, using examples.

Input the following one word to the CRC computing unit.

133.82.171.1 = "10000101" "01010010" "10101011" "00000001"

If the byte order is set to big endian (CRCCR.LTLEND= 0), the sending sequence in bytes is configured as shown below.

"10000101" "01010010" "10101011" "00000001"
(1st) (2nd) (3rd) (4th)

If the bit order is set to LSB First (CRCCR.LSBFST= 1), the sending sequence in bits is configured as shown below.

"10100001" "01001010" "11010101" "10000000"
(Head) (End)

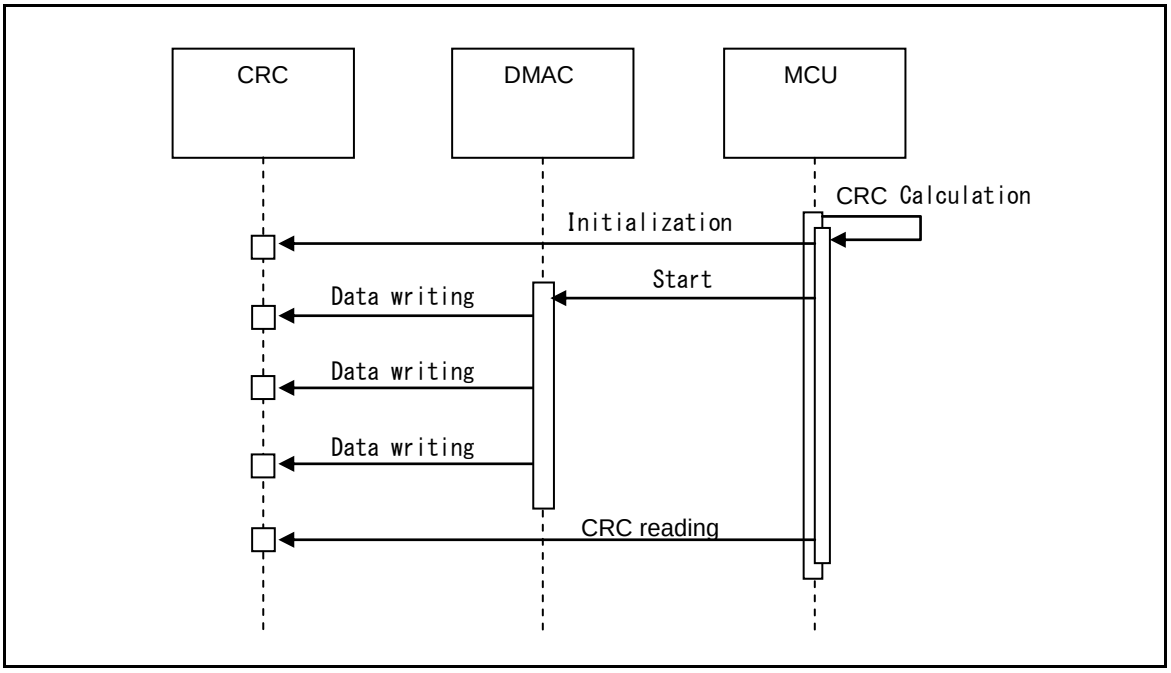
Note:

- *At CRCCR.CRCLTE= 1, the CRC result is rearranged in bytes with the 32-bit width in both CRC16 and CRC32.*
  In particular, in CRC16 mode, note that data is output to the D[31:16] of CRC Register (CRCR).

3.1. Calculation Sequence

Figure 3-1 shows the CRC calculation sequence. In this section, it is assumed that the Initial Value Register (CRCINIT) setting, CRC16 or CRC32 mode selection (CRCCR.CRC32), and byte- or bit- order setting (CRCCR.LTLEND, CRCCR.LSBFST) have already been configured.
If the initial value can be set to 0xFFFFFFFF, the Initial Value Register (CRCINIT) setting can be omitted.

Figure 3-1 CRC Calculation Sequence



- To perform initialization, write "1" to the initialization bit (CRCCR.INIT). The value of the Initial Value Register (CRCINIT) is loaded to the CRC Register (CRCR).

- To write input data, write to the Input Data Register (CRCIN). This then starts CRC calculation. If necessary, input data can be written continuously. Furthermore, different bit widths can be used in a sequence to write input data.

- To obtain a CRC code, read the CRC Register (CRCR).

3.2. Use Examples

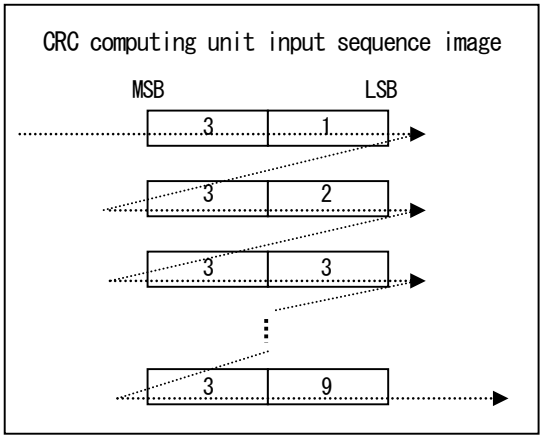
Figure 2-2 to Figure 2-5 show CRC use examples.

Use Example 1 CRC16, Byte Input Fixed

Figure 3-2 Use Example 1 (CRC16, Byte Input Fixed, Core Byte Order: Big Endian)

```
//*****************************************
//  CRC16 (CRC_ITU-T)
//     polynomial:      0x1021
//     initial value:   0xFFFF
//     CRCCR.CRC32:     0 //CRC16
//     CRCCR.LTLEND:    0 //big endian
//     CRCCR.LSBFST:    0 //MSB First
//     CRCCR.CRCLTE:    0 //CRC big endian
//     CRCCR.CRCLSF:    0 //CRC MSB First
//     CRCCR.FXOR:      0 //CRC Final XOR off
//*****************************************

//
// Example 1-1 byte-base writing
//

// Initialization
B_WRITE (CRCCR, 0x01);

// data write 0x313233343536373839
B_WRITE (CRCIN, 0x31);
B_WRITE (CRCIN, 0x32);
B_WRITE (CRCIN, 0x33);
B_WRITE (CRCIN, 0x34);
B_WRITE (CRCIN, 0x35);
B_WRITE (CRCIN, 0x36);
B_WRITE (CRCIN, 0x37);
B_WRITE (CRCIN, 0x38);
B_WRITE (CRCIN, 0x39);

// read result
H_READ (CRCR+2, data);

// check result
assert (data == 0x29B1);

//
// Example 1-2 CRC check
//

// Initialization
B_WRITE (CRCCR, 0x01);

// data write 0x313233343536373839 + CRC
B_WRITE (CRCIN, 0x31);
B_WRITE (CRCIN, 0x32);
B_WRITE (CRCIN, 0x33);
B_WRITE (CRCIN, 0x34);
B_WRITE (CRCIN, 0x35);
B_WRITE (CRCIN, 0x36);
B_WRITE (CRCIN, 0x37);
B_WRITE (CRCIN, 0x38);
B_WRITE (CRCIN, 0x39);
B_WRITE (CRCIN, 0x29); // <-- CRC
B_WRITE (CRCIN, 0xB1); // <-- CRC

// read result
H_READ (CRCR+2, data);

// check result
assert (data == 0x0000);
```

```
(Assumed as follows)

B_WRITE    — Byte writing
H_WRITE    — Half-word writing
W_WRITE    — Word writing

B_READ     — Byte reading
H_READ     — Half-word reading
W_READ     — Word reading

CRCCR      — CRC Control Register address
CRCINIT    — Initial Value Register address
CRCIN      — Input Data Register address
CRCR       — CRC Register address
```



The byte and half-word writing positions are arbitrary. In this example, data is written continuously at position +0.

Table 3-1 shows the CRC result byte order, CRCR (CRC Register) output position, and read address in CRC16 mode.

Table 3-1 CRC Result Byte Order, and CRCR Read Address

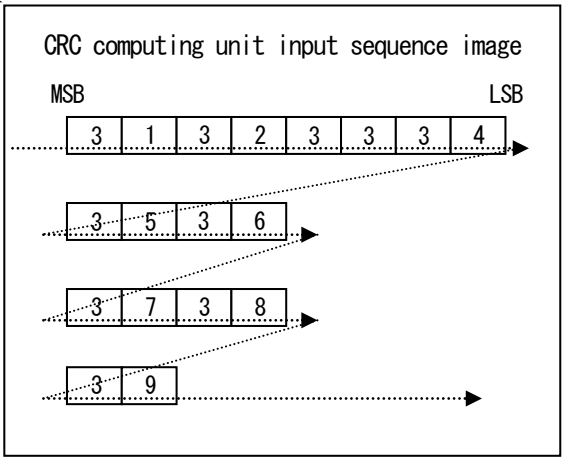
Core Byte Order	CRC Result Byte Order	Output Position to CRCR	CRCR H_READ Address
Big endian	Big endian	D[15:0]	CRCR +2
Big endian	Little endian	D[31:16]	CRCR +0
Little endian	Big endian	D[15:0]	CRCR +0
Little endian	Little endian	D[31:16]	CRCR +2

Use Example 2 CRC16, Different Input Bit Widths Mixed

Figure 3-3 Use Example 2 (CRC16, Different Input Bit Widths Mixed, Core Byte Order: Big Endian)

```
//*****************************************
//  CRC16 (CRC ITU-T)
//     polynomial:      0x1021
//     initial value:   0xFFFF
//     CRCCR.CRC32:     0 //CRC16
//     CRCCR.LTLEND:    0 //big endian
//     CRCCR.LSBFST:    0 //MSB First
//     CRCCR.CRCLTE:    0 //CRC big endian
//     CRCCR.CRCLSF:    0 //CRC MSB First
//     CRCCR.FXOR:      0 //CRC Final XOR off
//*****************************************

//
// Example 2-1 Writing widths mixed
//

// Initialization
B_WRITE (CRCCR, 0x01);

// data write 0x313233343536373839
W_WRITE (CRCIN, 0x31323334);
H_WRITE (CRCIN, 0x3536);
H_WRITE (CRCIN+2, 0x3738);
B_WRITE (CRCIN+3, 0x39);

// read result
H_READ (CRCR+2, data);

// check result
assert (data == 0x29B1);

//
// Example 2-2 CRC check
//

// Initialization
B_WRITE (CRCCR, 0x01);

// data write 0x313233343536373839 + CRC
W_WRITE (CRCIN, 0x31313334);
W_WRITE (CRCIN, 0x35363738);
H_WRITE (CRCIN, 0x3929);      // <— CRC(0x29)
B_WRITE (CRCIN, 0xB1);        // <— CRC(0xB1)

// read result
H_READ (CRCR+2, data);

// check result
assert (data == 0x0000);
```

```
(Assumed as follows)

B_WRITE    —— Byte writing
H_WRITE    —— Half-word writing
W_WRITE    —— Word writing

B_READ     —— Byte reading
H_READ     —— Half-word reading
W_READ     —— Word reading

CRCCR      —— CRC Control Register address
CRCINIT    —— Initial Value Register address
CRCIN      —— Input Data Register address
CRCR       —— CRC Register address
```

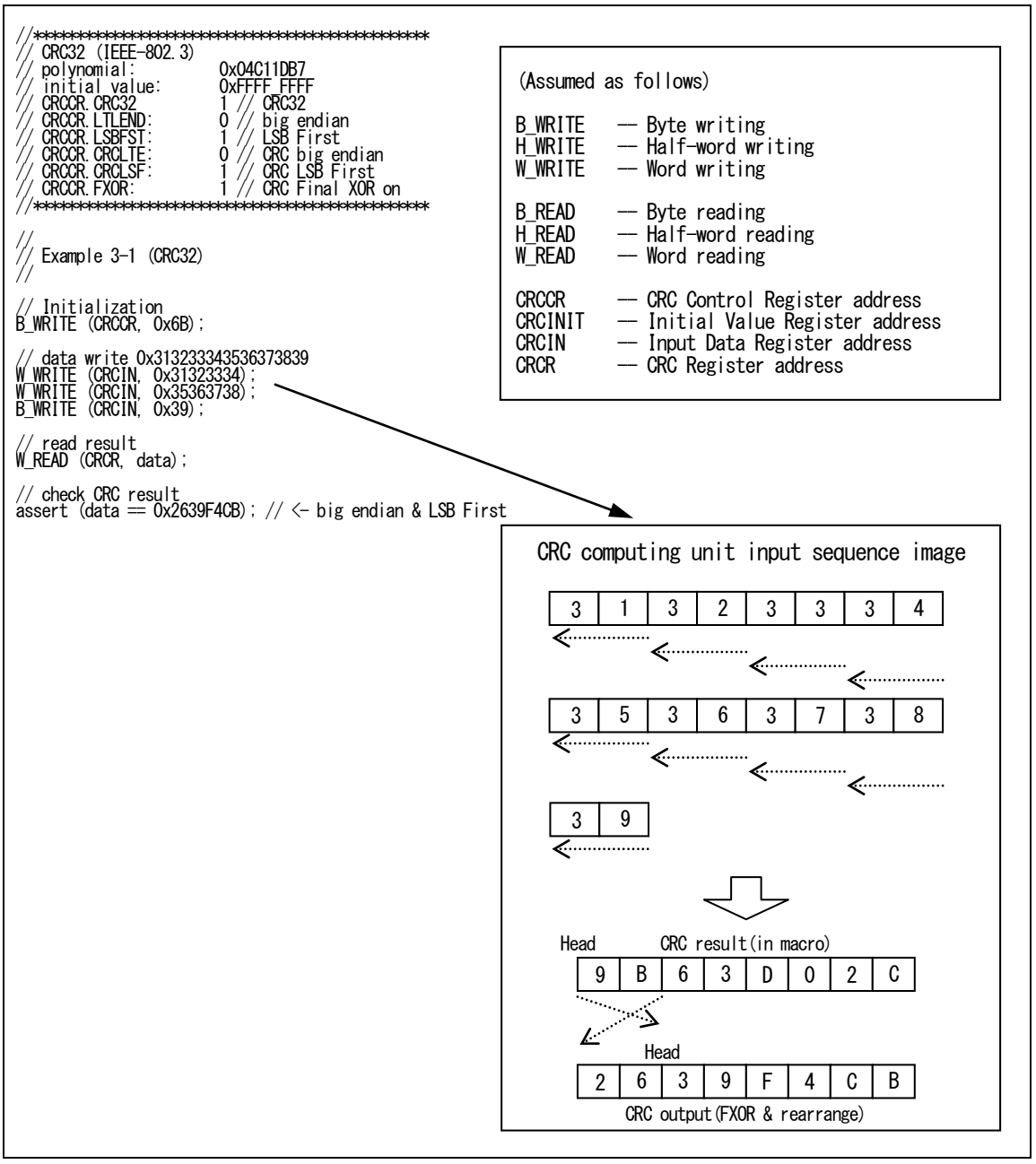


If the byte or bit order setting is correct and the bit input sequence to the CRC computing unit is the same, the writing width can be specified arbitrarily.

For example, if a 1-, 2-, or 3-byte fraction is obtained in the word-base writing mode, both byte and half-word writings may be enabled.

Use Example 3 CRC32, Byte Order: Big Endian

Figure 3-4 Use Example 3 (CRC32, Byte Order: Big Endian)



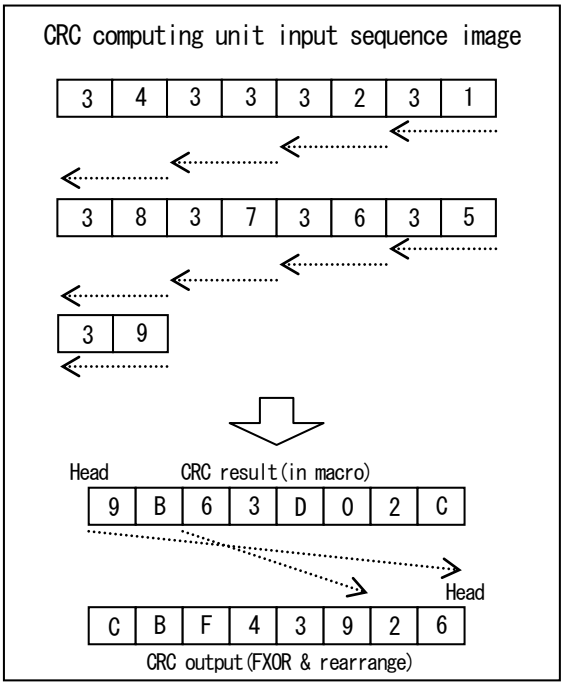
In CRC32 (IEEE-802.3) mode, the bit order is set to LSB First. This CRC computing unit supports both the big endian and little endian as the byte order. The figure above shows an example for big endian.

Use Example 4 CRC32, Byte Order: Little Endian

Figure 3-5 Use Example 4 (CRC32, Byte Order: Little Endian)

```
//*****************************************
// CRC32 (IEEE-802.3)
// polynomial:          0x04C11DB7
// initial value:       0xFFFF FFFF
// CRCCR.CRC32          1 // CRC32
// CRCCR.LTLEND:        1 // little endian
// CRCCR.LSBFST:        1 // LSB First
// CRCCR.CRCLTE:        1 // CRC little endian
// CRCCR.CRCLSF:        1 // CRC LSB First
// CRCCR.FXOR:          1 // CRC Final XOR on
//*****************************************

//
// Example 4-1 (CRC32)
//

// Initialization
B_WRITE (CRCCR, 0x7F);

// data write 0x3132333435363738 39
W_WRITE (CRCIN, 0x34333231);
W_WRITE (CRCIN, 0x38373635);
B_WRITE (CRCIN, 0x39);

// read result
W_READ (CRCR, data);

// check result
assert (data == 0xCBF43926); // <- little endian & LSB First
```

```
(Assumed as follows)

B_WRITE    — Byte writing
H_WRITE    — Half-word writing
W_WRITE    — Word writing

B_READ     — Byte reading
H_READ     — Half-word reading
W_READ     — Word reading

CRCCR      — CRC Control Register address
CRCINIT    — Initial Value Register address
CRCIN      — Input Data Register address
CRCR       — CRC Register address
```



In CRC32 (IEEE-802.3) mode, the bit order is set to LSB First. This CRC computing unit supports both the big endian and little endian as the byte order. The figure above shows an example for little endian.

If bit inversion is not required for the CRC result, perform either one of the following processes to release the bit inversion for the current result.

- Initialize with 0x3F before calculation.
- After data was input, set the CRCCR.FXOR bit to "0" (for example, CRCCR=0x3E).

4. Registers

This section provides a list of CRC registers.

The prefix "CRCxx_" is added to every register name (abbreviation).
xx is the channel number (00 to 03).

Table 4-1 List of CRC Register List

Abbreviation	Register Name	Reference
CRCCR	CRC Control Register	4.1
CRCINIT	Initial Value Register	4.2
CRCIN	Input Data Register	4.3
CRCR	CRC Register	4.4

Table 4-2 shows the register map.

Table 4-2 Register Map

Offset	Register Name			
	+3	+2	+1	+0
0x0000_0000	Reserved			CRCxx_CRCCR 00000000
0x0000_0004	CRCxx_CRCINIT 11111111_11111111_11111111_11111111			
0x0000_0008	CRCxx_CRCIN 00000000_00000000_00000000_00000000			
0x0000_000C	CRCxx_CRCR 11111111_11111111_11111111_11111111			

4.1. CRC Control Register (CRC00_CRCCR, CRC01_CRCCR, CRC02_CRCCR, CRC03_CRCCR)

The CRC Control Register (CRCCR) is used to control CRC calculation.

Bit	7	6	5	4	3	2	1	0
Field	Reserved	FXOR	CRCLSF	CRCLTE	LSBFST	LTLEND	CRC32	INIT
R/W Attribute	R0,W0	R/W	R/W	R/W	R/W	R/W	R/W	R0,W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

[bit7] Reserved: Reserved Bit

This bit must always be written to "0".

The read value is "0".

[bit6] FXOR: Final XOR Control Bit

This bit is used to output the CRC result as the XOR value or XOR.
The XOR value is set to 0xFFFFFFFF. The CRC result value is inverted at FXOR="1".
This processing is performed in the latter part of the CRC Register (CRCR) processing. The CRC result is therefore reflected on the read value immediately after this bit was set.

Bit	Description
0	None
1	Yes

[bit5] CRCLSF: CRC Result Bit-Order Setting Bit

This is a bit-order setting bit for CRC result.
This bit is used to rearrange bits within each byte. Set "0" to specify MSB First and set "1" to specify LSB First.

This processing is performed in the latter part of the CRC Register (CRCR) processing. The CRC result is therefore reflected on the read value immediately after this bit was set.

Bit	Description
0	MSB First
1	LSB First

[bit4] CRCLTE: CRC Result Byte-Order Setting Bit

This is a byte-order setting bit for CRC result.
This bit is used to rearrange the byte order in each word. Set "0" to specify big endian and set "1" to specify little endian.

This processing is performed in the latter part of the CRC Register (CRCR) processing. The CRC result is therefore reflected on the read value immediately after this bit was set.

If this bit is set to "1" in CRC16 mode, data is output to the D[31:16] of CRC Register (CRCR).

Bit	Description
0	Big endian
1	Little endian

[bit3] LSBFST: Bit-Order Setting Bit

This is a bit-order setting bit

This bit is used to specify the head bit of a byte (8 bits). Set "0" to specify MSB First and set "1" to specify LSB First.
Four types of processing orders can be specified when this bit is combined with the LTLEND setting.

Bit	Description
0	MSB First
1	LSB First

[bit2] LTLEND: Byte-Order Setting Bit

This is a byte-order setting bit.

This bit is used to specify the byte order with the write width. Set "0" to specify big endian and set "1" to specify little endian.

Bit	Description
0	Big endian
1	Little endian

[bit1] CRC32: CRC Mode Selection Bit

This bit is used to select the CRC16 or CRC32 mode.

Bit	Description
0	CRC16
1	CRC32

[bit0] INIT: Initialization Bit

This is an initialization bit. Writing "1" initializes data. This bit does not have a value, and always returns "0" at reading.
At initialization, the value of the Initial Value Register (CRCINIT) is loaded to the CRC Register (CRCR). Initialization must be performed once at the start of CRC calculation.

Bit	Description	
	Write	Read
0	Invalid	Always reads "0".
1	Initialization	

4.2. Initial Value Register (CRC00_CRCINIT, CRC01_CRCINIT, CRC02_CRCINIT, CRC03_CRCINIT)

The initial value Register (CRCINIT) is used to save the initial values for CRC calculation.

Bit	31	0
Field	D	
R/W Attribute	R/W	
Protection Attribute	-	
Initial Value	11111111_11111111_11111111_11111111	

[bit31:0] D[31:0]: Initial Value Bit

These bits are used to save the initial values for CRC calculation.

Write the initial values for CRC calculation to this register.

In CRC16 mode, D[15:0] are used while D[31:16] are ignored.

4.3. Input Data Register (CRC00_CRCIN, CRC01_CRCIN, CRC02_CRCIN, CRC03_CRCIN)

The Input Data Register (CRCIN) is used to set input data for CRC calculation.

Bit	31	0
Field	D	
R/W Attribute	R/W	
Protection Attribute	-	
Initial Value	00000000_00000000_00000000_00000000	

[bit31:0] D[31:0]: Input Data bit

These bits are used to set input data for CRC calculation.

Write input data for CRC calculation to this register. There are three types of bit widths: 8, 16, and 32, which can be specified together.

The byte and half-word writing positions are arbitrary. The available address positions are as follows.

- Byte writing : +0, +1, +2, +3
- Half word writing: +0, +2

4.4. CRC Register (CRC00_CRCR, CRC01_CRCR, CRC02_CRCR, CRC03_CRCR)

The CRC Register (CRCR) is used to output the CRC calculation result. This register must be initialized before start calculating.

Bit	31	0
Field	D	
R/W Attribute	R,WX	
Protection Attribute	-	
Initial Value	11111111_11111111_11111111_11111111	

[bit31:0] D[31:0]: CRC bit

These bits are used to read the CRC calculation result. If "1" is written to the initialization bit (CRCCR.INIT), the value of the Initial Value Register (CRCINIT) is loaded to this register.
If input data for CRC calculation is written to the Input Data Register (CRCIN), the CRC calculation result is set to this register after one peripheral clock cycle has elapsed. When all input data writing has been completed, this register holds the final CRC code.
In CRC16 mode, when the byte order is set to big endian (CRCCR.CRCLTE= 0), the result is output to D[15:0]. When the byte order is set to little endian (CRCCR.CRCLTE= 1), the result is output to D[31:16].

CHAPTER 50: I/O Port

This chapter describes I/O port.

1. **Overview**
2. **Configuration and Block Diagrams**
3. **Setting Procedure Examples**
4. **Register List**
5. **Precautions for Using This Device**

GPIO-TXXPT03P01R01L11-E1-XX

1. Overview

This section provides an overview of I/O port.

I/O port has a general-purpose I/O module. The external pins can be used as I/O ports. Assignment of an external pin and input to an internal resource can be set.

General-Purpose I/O Port Module (GPIO)

General-purpose I/O module enables using external pins as I/O ports. The general-purpose I/O module is composed of (product specification) GPIO ports. Each GPIO port has 32 channels, which correspond to external pins. For example, the external pin P216 corresponds to the channel 16 setting of the GPIO port 2.
Key code settings are required for writing to registers.

Port Configuration Module (PPC)

- The port configuration module sets I/O from/to an external pin. This setting can be set per external pin.
  - Output enable display
  - I/O status display
  - Pull-up/Pull-down setting
  - Input level setting
  - Output drive capacity setting
  - Input disable setting
- Output (GPIO, resource) function selection
- Key code settings are required for writing to registers.

Resource Input Configuration Module (RIC)

The resource input configuration module selects input from an external pin or output from another internal resource as resource input.

2. Configuration and Block Diagrams

This section describes the block diagram of I/O port.

Figure 2-1 and Figure 2-2 show the I/O port configuration diagrams.

Figure 2-1 Configuration Diagram of GPIO and PPC

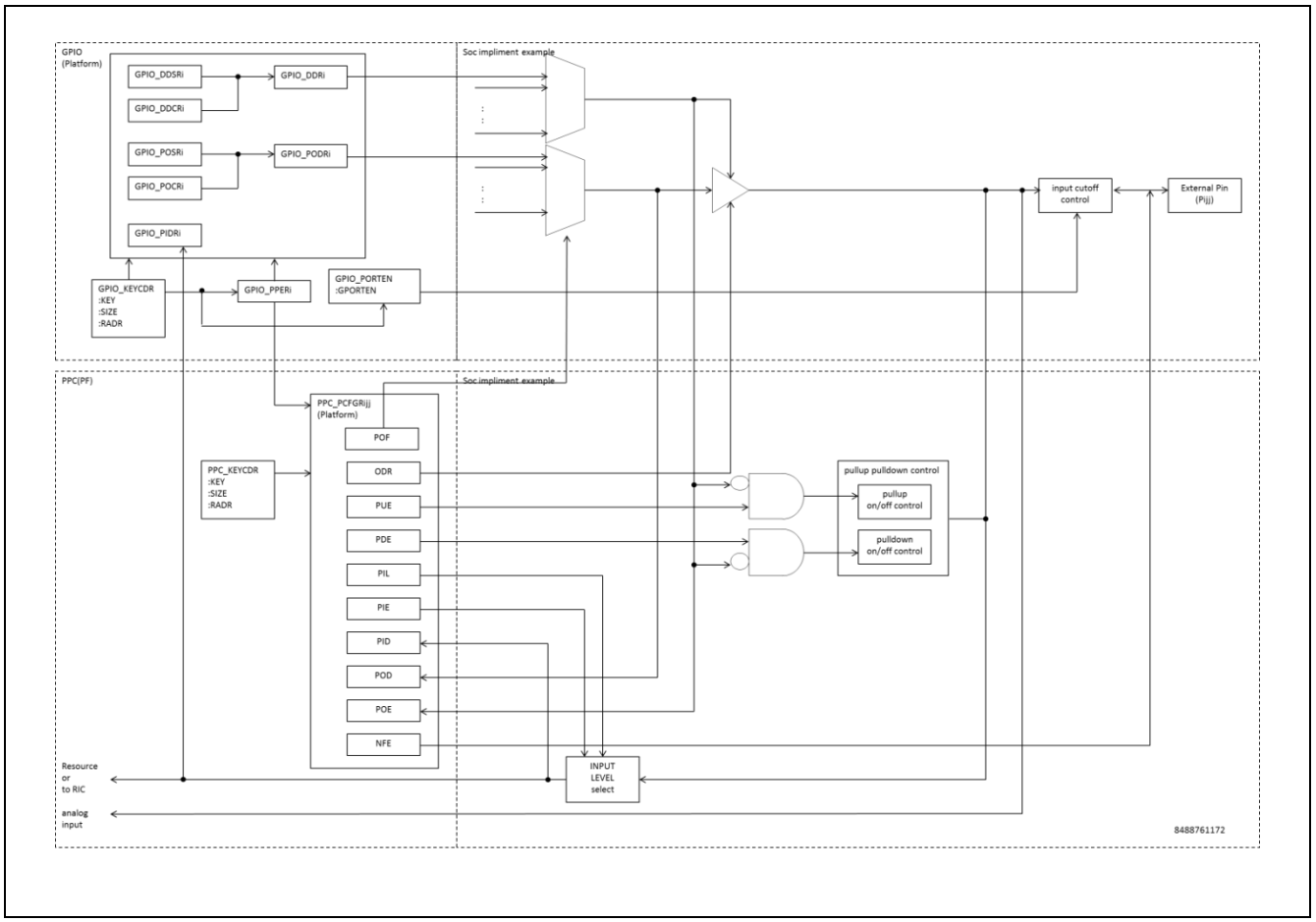
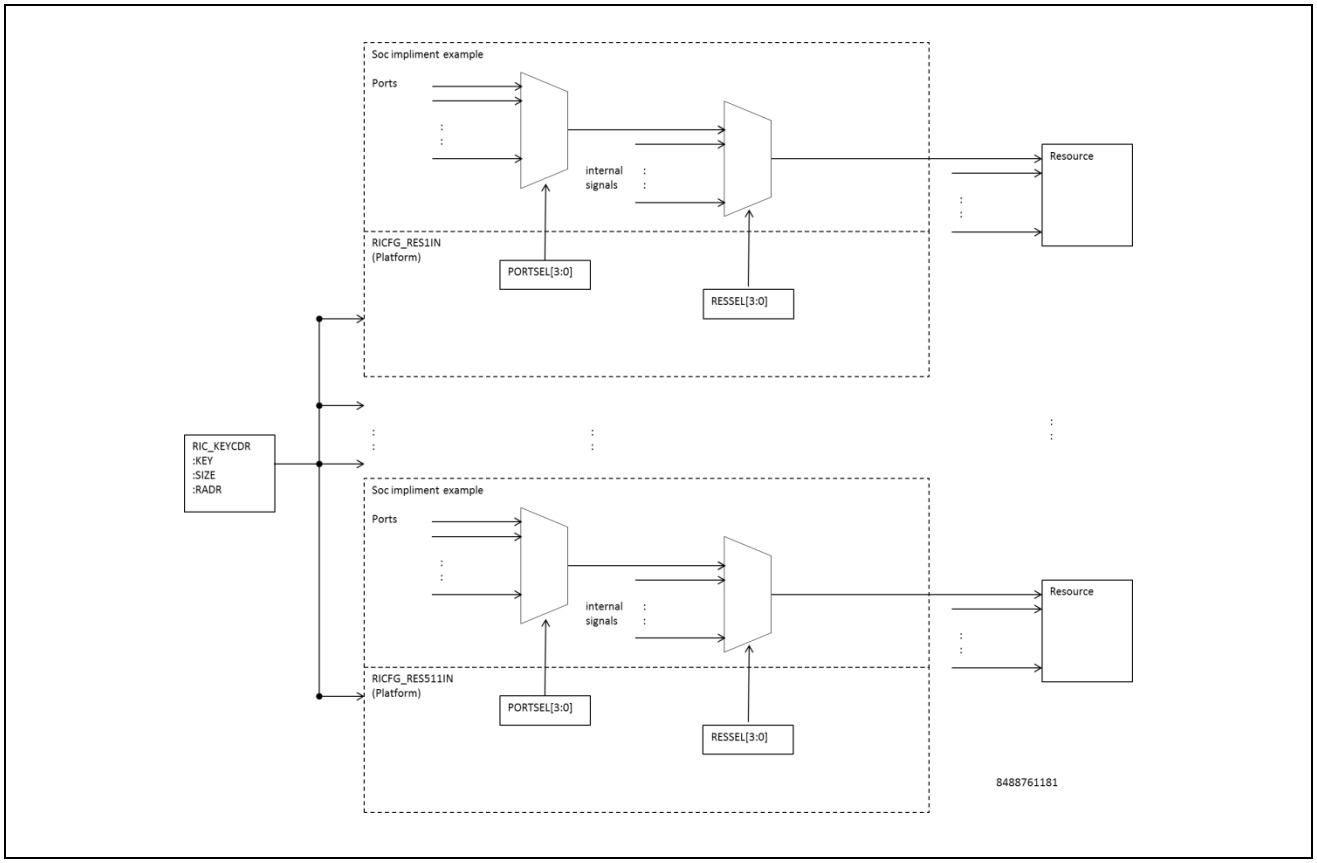


Figure 2-2 Configuration Diagram of RIC

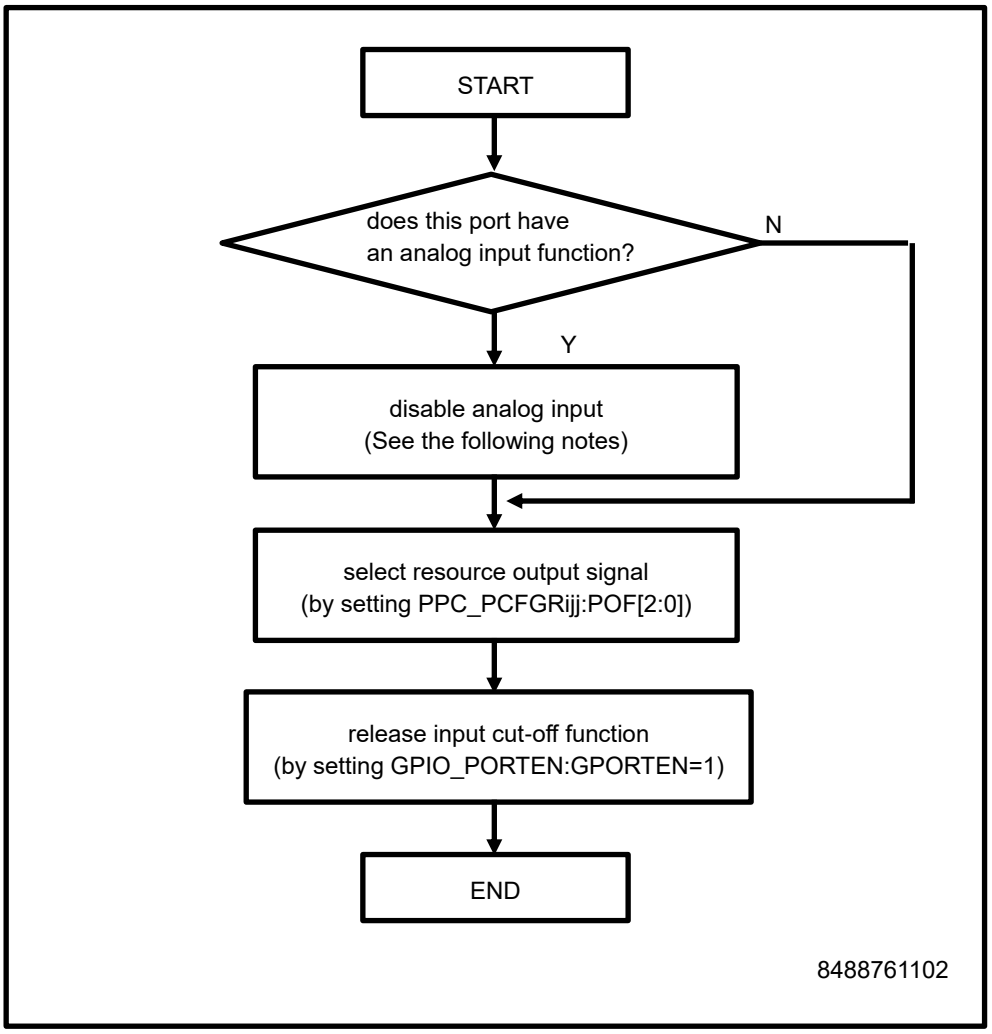


3. Setting Procedure Examples

This section describes the setting procedure examples of I/O port.

Assignment of resource I/O Pin (both Directions)

Figure 3-1 Setting Procedure

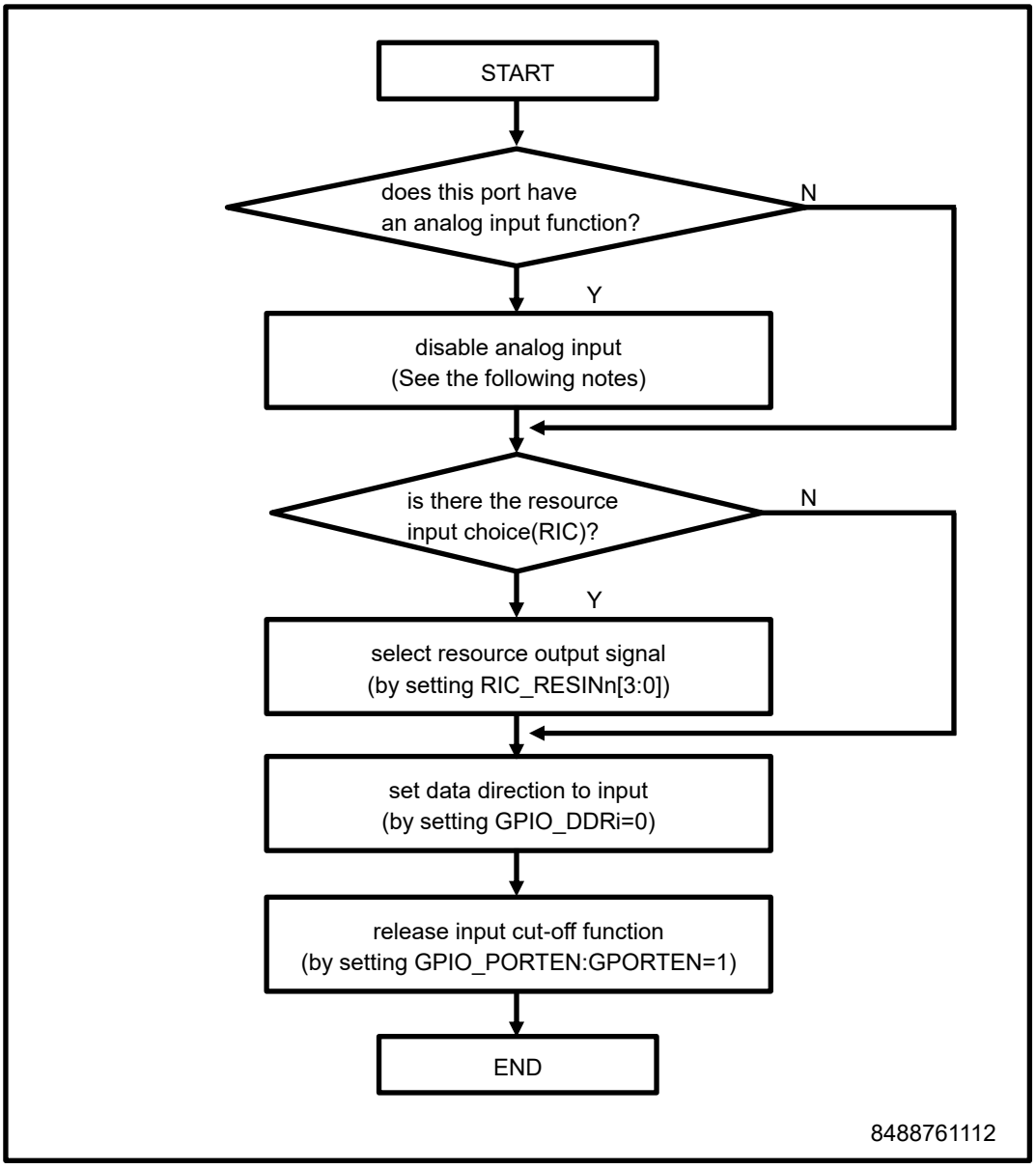


Note:

- *The description of the analog input by A/D converter function should be referred in the product hardware manual.*
- *The following registers are the applicable key code registers.*
  - Port enable register (GPIO_PORTEN)

Assignment of Resource Input Pin (Selecting the Resource Input of an External Pin)

Figure 3-2 Setting Procedure

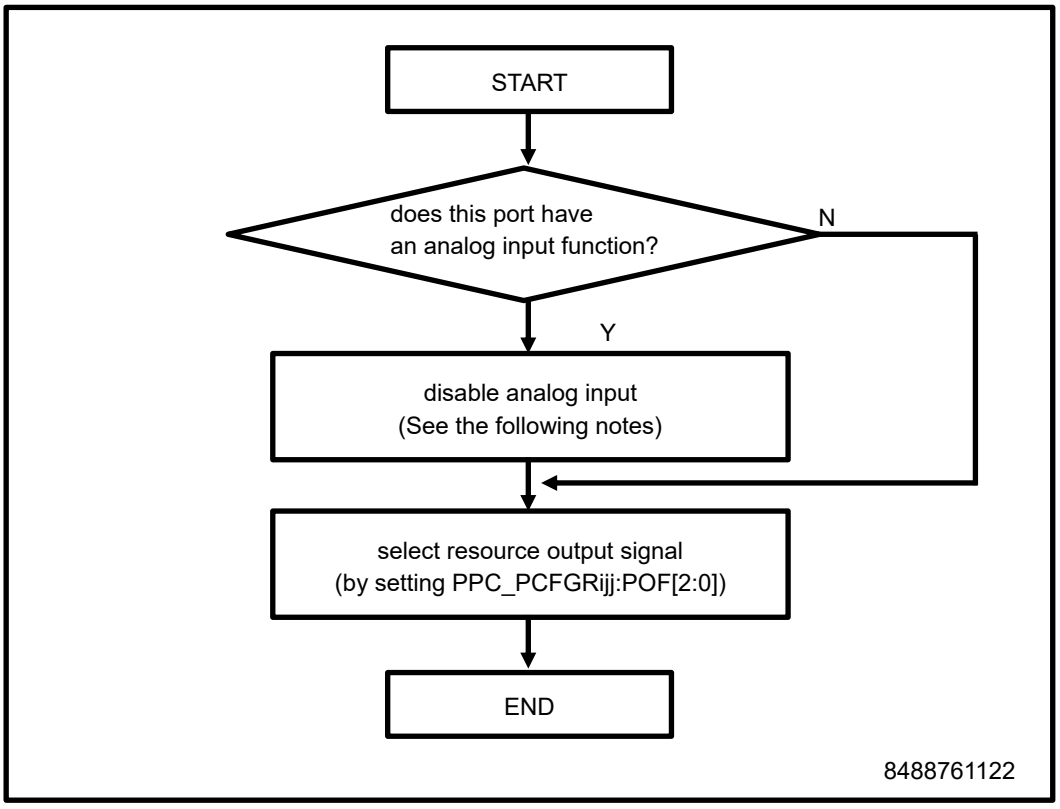


Note:

- *The description of the analog input by A/D converter function should be referred in the product hardware manual.*
- *The following registers are the applicable key code registers.*
  - Data direction register (GPIO_DDRi)
  - Port enable register (GPIO_PORTEN)

Assignment of Resource Output Pin

Figure 3-3 Setting Procedure

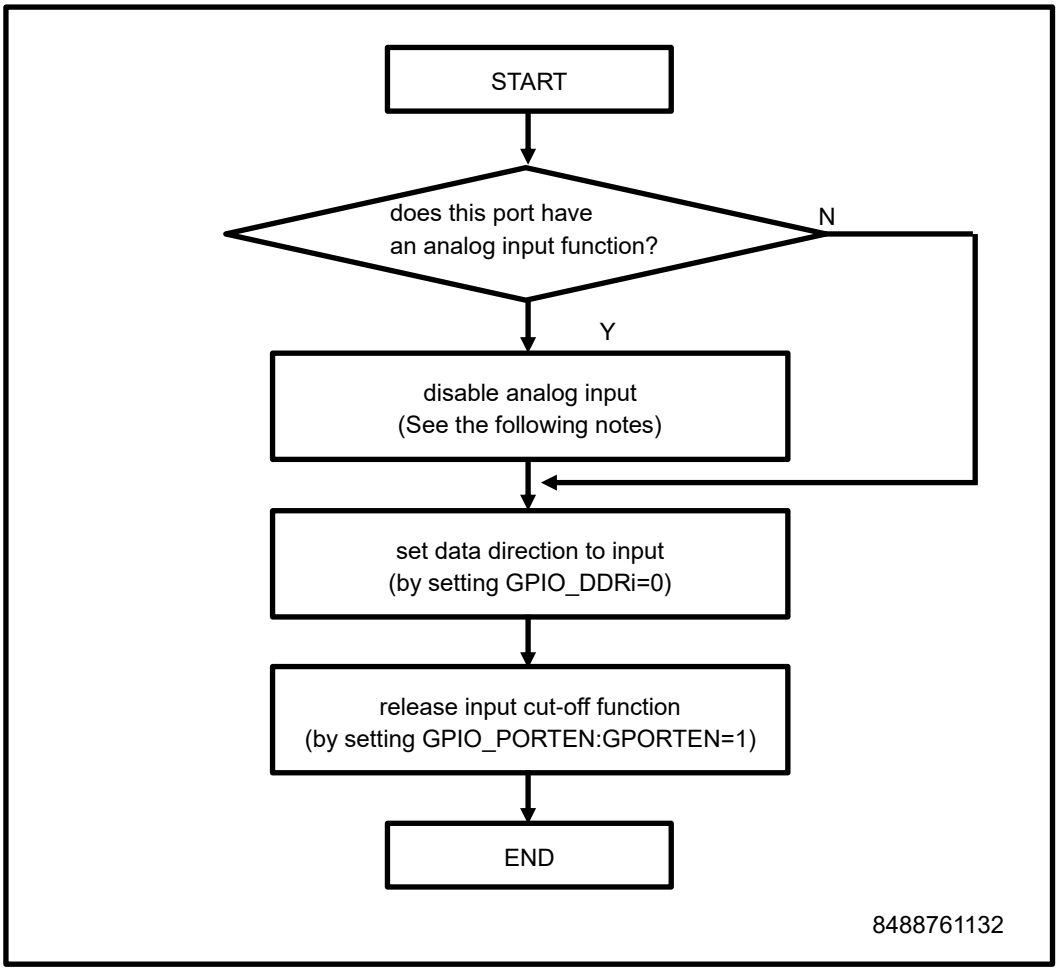


Note:

- *The description of the analog input by A/D converter function should be referred in the product hardware manual.*
- *The following registers are the applicable key code registers.*
  - Port setting register (PPC_PCFGRijj)

Assignment of Port Function (Input)

Figure 3-4 Setting Procedure

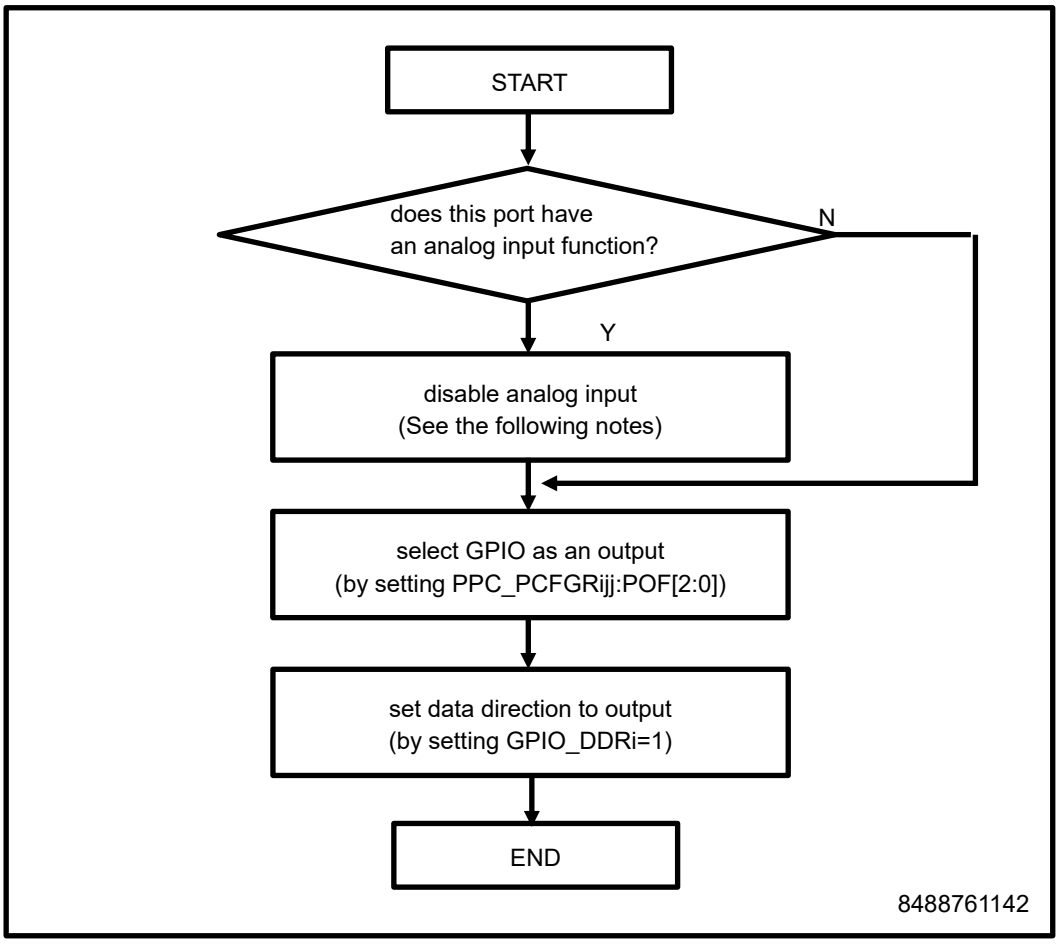


Note:

- *The description of the analog input by A/D converter function should be referred in the product hardware manual.*
- *The following registers are the applicable key code registers.*
  - Data direction register (GPIO_DDRi)
  - Port enable register (GPIO_PORTEN)

Assignment of Port Function (Output)

Figure 3-5 Setting Procedure

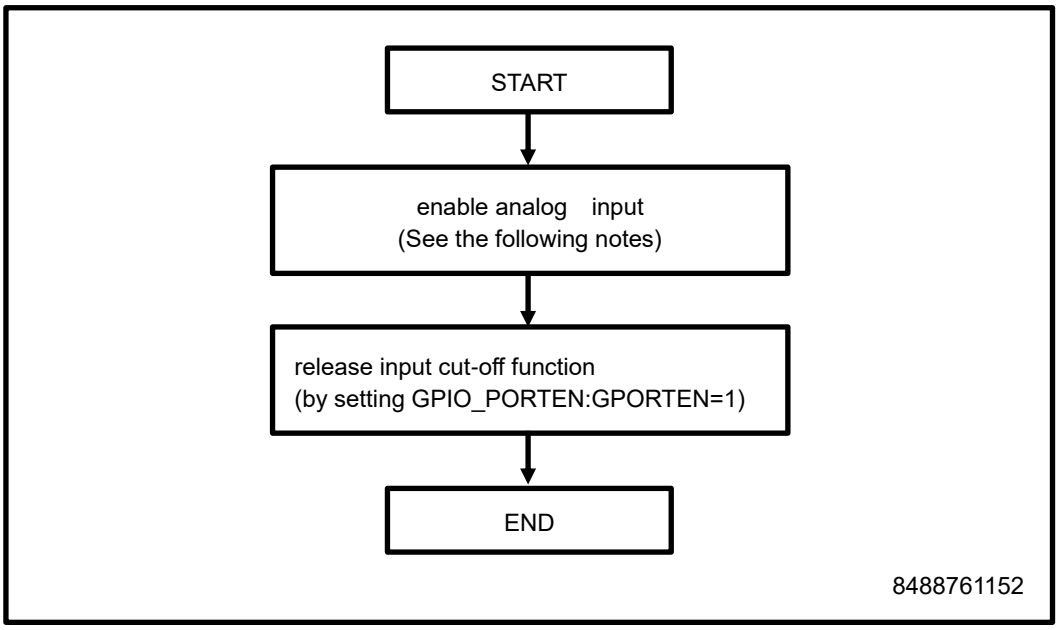


Note:

- *The description of the analog input by A/D converter function should be referred in the product hardware manual.*
- *The following registers are the applicable key code registers.*
  - Port setting register (PPC_PCFGRijj)

Assignment of AD Converter Input

Figure 3-6 Setting Procedure

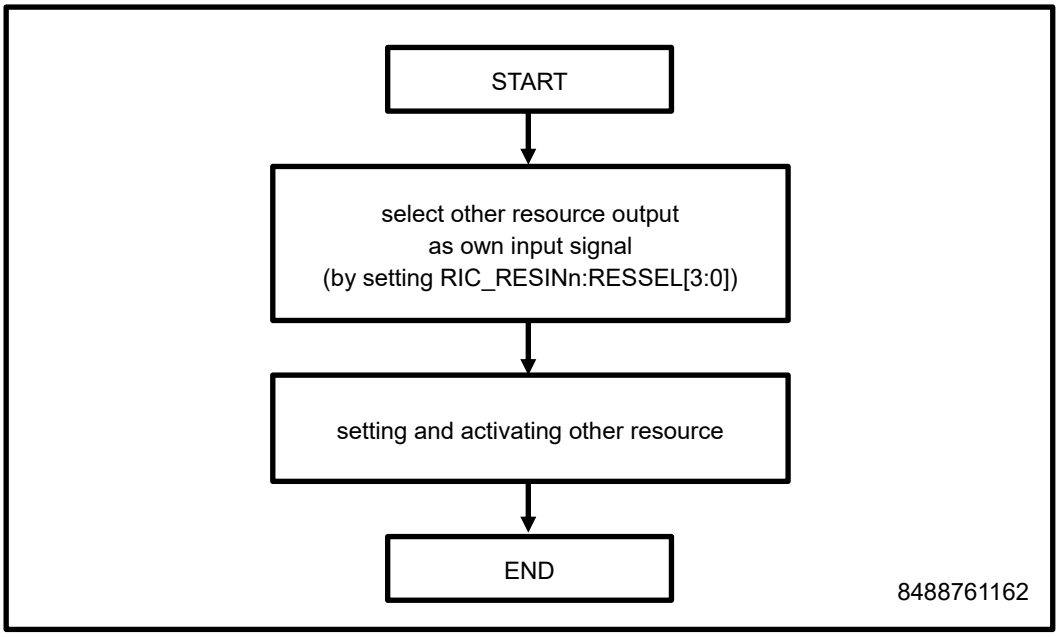


Note:

- *The description of the analog input by A/D converter function should be referred in the product hardware manual.*
- *The following registers are the applicable key code registers.*
  - Port enable register (GPIO_PORTEN)

Resource Input Selection (Selecting Other Resource Output)

Figure 3-7 Setting Procedure



Note:

- *For details on resource input selection, see product specification.*

4. Register List

This section describes the register list of the I/O port.

Table 4-1 I/O Port Register List

Abbreviated Register Name	Register Name	Reference
GPIO_DDRi	Data direction register	4.1
GPIO_DDSRi	Data direction set register	4.2
GPIO_DDCRi	Data direction clear register	4.3
GPIO_PODRi	Port output data register	4.4
GPIO_POSRi	Port output set register	4.5
GPIO_POCRi	Port output clear register	4.6
GPIO_PORTEN	Port input enable register	4.7
GPIO_PIDRi	Port input data register	4.8
GPIO_KEYCDR	GPIO key code register	4.9
PPC_PCFGRijj	Port setting register	4.10
PPC_KEYCDR	PPC key code register	4.11
RIC_RESINn	Resource input setting register	4.12
RIC_KEYCDR	RIC key code register	4.13

i: GPIO port number (i = 0 to (product specification))

jj: GPIO channel number (jj = 00 to 31)

n: Number of selectable resource inputs (n = 0 to (product specification))

Table 4-2 Register Map of GPIO

Address Offset	Register Name / Initial Value			
	+3	+2	+1	+0
0x00000000	GPIO_POSR0 00000000_00000000_00000000_00000000			
0x00000004	GPIO_POCR0 00000000_00000000_00000000_00000000			
0x00000008	GPIO_DDSR0 00000000_00000000_00000000_00000000			
0x0000000C	GPIO_DDCR0 00000000_00000000_00000000_00000000			
0x00000010	GPIO_POSR1 00000000_00000000_00000000_00000000			
0x00000014	GPIO_POCR1 00000000_00000000_00000000_00000000			
0x00000018	GPIO_DDSR1 00000000_00000000_00000000_00000000			
0x0000001C	GPIO_DDCR1 00000000_00000000_00000000_00000000			
0x00000020	GPIO_POSR2 00000000_00000000_00000000_00000000			
0x00000024	GPIO_POCR2 00000000_00000000_00000000_00000000			
0x00000028	GPIO_DDSR2 00000000_00000000_00000000_00000000			
0x0000002C	GPIO_DDCR2 00000000_00000000_00000000_00000000			
0x00000030	GPIO_POSR3 00000000_00000000_00000000_00000000			
0x00000034	GPIO_POCR3 00000000_00000000_00000000_00000000			
0x00000038	GPIO_DDSR3 00000000_00000000_00000000_00000000			
0x0000003C	GPIO_DDCR3 00000000_00000000_00000000_00000000			
0x00000040	GPIO_POSR4 00000000_00000000_00000000_00000000			
0x00000044	GPIO_POCR4 00000000_00000000_00000000_00000000			
0x00000048	GPIO_DDSR4 00000000_00000000_00000000_00000000			
0x0000004C	GPIO_DDCR4 00000000_00000000_00000000_00000000			
0x00000050	GPIO_POSR5 00000000_00000000_00000000_00000000			
0x00000054	GPIO_POCR5 00000000_00000000_00000000_00000000			

Address Offset	Register Name / Initial Value			
	+3	+2	+1	+0
0x00000058	GPIO_DDSR5 00000000_00000000_00000000_00000000			
0x0000005C	GPIO_DDCR5 00000000_00000000_00000000_00000000			
0x00000060	GPIO_POSR6 00000000_00000000_00000000_00000000			
0x00000064	GPIO_POCR6 00000000_00000000_00000000_00000000			
0x00000068	GPIO_DDSR6 00000000_00000000_00000000_00000000			
0x0000006C	GPIO_DDCR6 00000000_00000000_00000000_00000000			
0x00000070	GPIO_POSR7 00000000_00000000_00000000_00000000			
0x00000074	GPIO_POCR7 00000000_00000000_00000000_00000000			
0x00000078	GPIO_DDSR7 00000000_00000000_00000000_00000000			
0x0000007C	GPIO_DDCR7 00000000_00000000_00000000_00000000			
0x00000080	GPIO_POSR8 00000000_00000000_00000000_00000000			
0x00000084	GPIO_POCR8 00000000_00000000_00000000_00000000			
0x00000088	GPIO_DDSR8 00000000_00000000_00000000_00000000			
0x0000008C	GPIO_DDCR8 00000000_00000000_00000000_00000000			
0x00000090	GPIO_POSR9 00000000_00000000_00000000_00000000			
0x00000094	GPIO_POCR9 00000000_00000000_00000000_00000000			
0x00000098	GPIO_DDSR9 00000000_00000000_00000000_00000000			
0x0000009C	GPIO_DDCR9 00000000_00000000_00000000_00000000			
0x000000A0	GPIO_POSR10 00000000_00000000_00000000_00000000			
0x000000A4	GPIO_POCR10 00000000_00000000_00000000_00000000			
0x000000A8	GPIO_DDSR10 00000000_00000000_00000000_00000000			
0x000000AC	GPIO_DDCR10 00000000_00000000_00000000_00000000			

Address Offset	Register Name / Initial Value			
	+3	+2	+1	+0
0x000000B0	GPIO_POSR11 00000000_00000000_00000000_00000000			
0x000000B4	GPIO_POCR11 00000000_00000000_00000000_00000000			
0x000000B8	GPIO_DDSR11 00000000_00000000_00000000_00000000			
0x000000BC	GPIO_DDCR11 00000000_00000000_00000000_00000000			
0x000000C0	GPIO_POSR12 00000000_00000000_00000000_00000000			
0x000000C4	GPIO_POCR12 00000000_00000000_00000000_00000000			
0x000000C8	GPIO_DDSR12 00000000_00000000_00000000_00000000			
0x000000CC	GPIO_DDCR12 00000000_00000000_00000000_00000000			
0x000000D0	GPIO_POSR13 00000000_00000000_00000000_00000000			
0x000000D4	GPIO_POCR13 00000000_00000000_00000000_00000000			
0x000000D8	GPIO_DDSR13 00000000_00000000_00000000_00000000			
0x000000DC	GPIO_DDCR13 00000000_00000000_00000000_00000000			
0x000000E0	GPIO_POSR14 00000000_00000000_00000000_00000000			
0x000000E4	GPIO_POCR14 00000000_00000000_00000000_00000000			
0x000000E8	GPIO_DDSR14 00000000_00000000_00000000_00000000			
0x000000EC	GPIO_DDCR14 00000000_00000000_00000000_00000000			
0x000000F0	GPIO_POSR15 00000000_00000000_00000000_00000000			
0x000000F4	GPIO_POCR15 00000000_00000000_00000000_00000000			
0x000000F8	GPIO_DDSR15 00000000_00000000_00000000_00000000			
0x000000FC	GPIO_DDCR15 00000000_00000000_00000000_00000000			
0x00000100 to 000001FC	Reserved			
0x00000200	GPIO_PODR0 00000000_00000000_00000000_00000000			
0x00000204	GPIO_DDR0 00000000_00000000_00000000_00000000			

Address Offset	Register Name / Initial Value			
	+3	+2	+1	+0
0x00000208	GPIO_PODR1 00000000_00000000_00000000_00000000			
0x0000020C	GPIO_DDR1 00000000_00000000_00000000_00000000			
0x00000210	GPIO_PODR2 00000000_00000000_00000000_00000000			
0x00000214	GPIO_DDR2 00000000_00000000_00000000_00000000			
0x00000218	GPIO_PODR3 00000000_00000000_00000000_00000000			
0x0000021C	GPIO_DDR3 00000000_00000000_00000000_00000000			
0x00000220	GPIO_PODR4 00000000_00000000_00000000_00000000			
0x00000224	GPIO_DDR4 00000000_00000000_00000000_00000000			
0x00000228	GPIO_PODR5 00000000_00000000_00000000_00000000			
0x0000022C	GPIO_DDR5 00000000_00000000_00000000_00000000			
0x00000230	GPIO_PODR6 00000000_00000000_00000000_00000000			
0x00000234	GPIO_DDR6 00000000_00000000_00000000_00000000			
0x00000238	GPIO_PODR7 00000000_00000000_00000000_00000000			
0x0000023C	GPIO_DDR7 00000000_00000000_00000000_00000000			
0x00000240	GPIO_PODR8 00000000_00000000_00000000_00000000			
0x00000244	GPIO_DDR8 00000000_00000000_00000000_00000000			
0x00000248	GPIO_PODR9 00000000_00000000_00000000_00000000			
0x0000024C	GPIO_DDR9 00000000_00000000_00000000_00000000			
0x00000250	GPIO_PODR10 00000000_00000000_00000000_00000000			
0x00000254	GPIO_DDR10 00000000_00000000_00000000_00000000			
0x00000258	GPIO_PODR11 00000000_00000000_00000000_00000000			
0x0000025C	GPIO_DDR11 00000000_00000000_00000000_00000000			

Address Offset	Register Name / Initial Value			
	+3	+2	+1	+0
0x00000260	GPIO_PODR12 00000000_00000000_00000000_00000000			
0x00000264	GPIO_DDR12 00000000_00000000_00000000_00000000			
0x00000268	GPIO_PODR13 00000000_00000000_00000000_00000000			
0x0000026C	GPIO_DDR13 00000000_00000000_00000000_00000000			
0x00000270	GPIO_PODR14 00000000_00000000_00000000_00000000			
0x00000274	GPIO_DDR14 00000000_00000000_00000000_00000000			
0x00000278	GPIO_PODR15 00000000_00000000_00000000_00000000			
0x0000027C	GPIO_DDR15 00000000_00000000_00000000_00000000			
0x00000280 to 000002FC	Reserved			
0x00000300	GPIO_PIDR0 XXXXXXXX_XXXXXXXX_XXXXXXXX_XXXXXXXX			
0x00000304	GPIO_PIDR1 XXXXXXXX_XXXXXXXX_XXXXXXXX_XXXXXXXX			
0x00000308	GPIO_PIDR2 XXXXXXXX_XXXXXXXX_XXXXXXXX_XXXXXXXX			
0x0000030C	GPIO_PIDR3 XXXXXXXX_XXXXXXXX_XXXXXXXX_XXXXXXXX			
0x00000310	GPIO_PIDR4 XXXXXXXX_XXXXXXXX_XXXXXXXX_XXXXXXXX			
0x00000314	GPIO_PIDR5 XXXXXXXX_XXXXXXXX_XXXXXXXX_XXXXXXXX			
0x00000318	GPIO_PIDR6 XXXXXXXX_XXXXXXXX_XXXXXXXX_XXXXXXXX			
0x0000031C	GPIO_PIDR7 XXXXXXXX_XXXXXXXX_XXXXXXXX_XXXXXXXX			
0x00000320	GPIO_PIDR8 XXXXXXXX_XXXXXXXX_XXXXXXXX_XXXXXXXX			
0x00000324	GPIO_PIDR9 XXXXXXXX_XXXXXXXX_XXXXXXXX_XXXXXXXX			
0x00000328	GPIO_PIDR10 XXXXXXXX_XXXXXXXX_XXXXXXXX_XXXXXXXX			
0x0000032C	GPIO_PIDR11 XXXXXXXX_XXXXXXXX_XXXXXXXX_XXXXXXXX			
0x00000330	GPIO_PIDR12 XXXXXXXX_XXXXXXXX_XXXXXXXX_XXXXXXXX			
0x00000334	GPIO_PIDR13 XXXXXXXX_XXXXXXXX_XXXXXXXX_XXXXXXXX			

Address Offset	Register Name / Initial Value			
	+3	+2	+1	+0
0x00000338	GPIO_PIDR14 XXXXXXXX_XXXXXXXX_XXXXXXXX_XXXXXXXX			
0x0000033C	GPIO_PIDR15 XXXXXXXX_XXXXXXXX_XXXXXXXX_XXXXXXXX			
0x00000340 to 37C	Reserved			
0x00000380	Reserved			
0x00000384	Reserved			
0x00000388	Reserved			
0x0000038C	Reserved			
0x00000390	Reserved			
0x00000394	Reserved			
0x00000398	Reserved			
0x0000039C	Reserved			
0x000003A0	Reserved			
0x000003A4	Reserved			
0x000003A8	Reserved			
0x000003AC	Reserved			
0x000003B0	Reserved			
0x000003B4	Reserved			
0x000003B8	Reserved			
0x000003BC	Reserved			
0x000003C0 to 3FC	Reserved			
0x00000400	GPIO_PORTEN 00000000_00000000_00000000_00000000			
0x00000404	GPIO_KEYCDR 00000000_00000000_00000000_00000000			

Table 4-3 Register Map of PPC

Offset	Register Name / Initial Value	
	+1	+0
0x00000000 to 0x0000003F	PPC_PCFGR0000 to 0031 00000000_00000000	
0x00000040 to 0x0000007F	PPC_PCFGR0100 to 0131 00000000_00000000	
0x00000080 to 0x000000BF	PPC_PCFGR0200 to 0231 00000000_00000000	
0x000000C0 to 0x000000FF	PPC_PCFGR0300 to 0331 00000000_00000000	
0x00000100 to 0x0000013F	PPC_PCFGR0400 to 0431 00000000_00000000	
0x00000140 to 0x0000017F	PPC_PCFGR0500 to 0531 00000000_00000000	
0x00000180 to 0x000001BF	PPC_PCFGR0600 to 0631 00000000_00000000	
0x000001C0 to 0x000001FF	PPC_PCFGR0700 to 0731 00000000_00000000	
0x00000200 to 0x0000023F	PPC_PCFGR0800 to 0831 00000000_00000000	
0x00000240 to 0x0000027F	PPC_PCFGR0900 to 0931 00000000_00000000	
0x00000280 to 0x000002BF	PPC_PCFGR1000 to 1031 00000000_00000000	
0x000002C0 to 0x000002FF	PPC_PCFGR1100 to 1131 00000000_00000000	
0x00000300 to 0x0000033F	PPC_PCFGR1200 to 1231 00000000_00000000	
0x00000340 to 0x0000037F	PPC_PCFGR1300 to 1331 00000000_00000000	
0x00000380 to 0x000003BF	PPC_PCFGR1400 to 1431 00000000_00000000	
0x000003C0 to 0x000003FF	PPC_PCFGR1500 to 1531 00000000_00000000	

Offset	Register Name			
	+3	+2	+1	+0
0x00000400	PPC_KEYCDR 00000000_00000000_00000000_00000000			

Table 4-4 Register Map of RIC

Offset	Register Name	
	+2	+1
0x00000000 to 0x00000FFF	RIC_RESINn 00000000_00000000	
Offset	**Register Name**	
	+3	**+2**	**+1**	**+0**
0x00001400	RIC_KEYCDR 00000000_00000000_00000000_00000000			

4.1. Data Direction Register (GPIO_DDRi) (i = 0 to (product specification))

This register sets the I/O direction of a pin. The direction is set when a GPIO is selected by the port output function selection bit (POF[2:0]) of the port setting register (PPC_PCFGRijj). This register can be directly written or can be set/cleared using the data direction set register (GPIO_DDSRi) and the data direction clear register (GPIO_DDCRi).

Bit	31	0
Field	DD	
R/W Attribute	R/W	
Protection Attribute	WS	
Initial Value	00000000_00000000_00000000_00000000	

[bit31:0] DD[n]: Data Direction Selection Bit (n = 0 to 31)

DD[n] n=0 to 31	Description
0	Set data direction to input.
1	Set data direction to output.

Notes:

- *This is the applicable key code register. To write to this register, the GPIO key code register (GPIO_KEYCDR) must be set.*
- *If this register is written before the key code is released, a bus error response is returned.*
- *After key code release, writing to a different address or using a different access size than specified in the key code sequence will not generate an error response, and it can't write the data correctly.*
- *Please be sure to read and check a preset value after writing to a GPIO register.*

4.2. Data Direction Set Register (GPIO_DDSRi) (i = 0 to (product specification))

This register is used to set the data direction selection bit (GPIO_DDRi: DD31 to DD0).

Bit	31	0
Field	DDS	
R/W Attribute	R0,W	
Protection Attribute	WS	
Initial Value	00000000_00000000_00000000_00000000	

[bit31:0] DDS[n]: Data Direction Set Bit (n = 0 to 31)

DDS[n] n=0 to 31	Description
0	No effect
1	Set the data direction selection bit (GPIO_DDRi: DD31 to DD0) to "1".

Notes:

- *This is the applicable key code register. To write to this register, the GPIO key code register (GPIO_KEYCDR) must be set.*
- *If this register is written before the key code is released, a bus error response is returned.*
- *After key code release, writing to a different address or using a different access size than specified in the key code sequence will not generate an error response.*
- *Please be sure to read and check a preset value after writing to a GPIO register.*

4.3. Data Direction Clear Register (GPIO_DDCRi) (i = 0 to (product specification))

The data direction clear (DDC) register is used to clear the corresponding GPIO_DDRi bit.

Bit	31	0
Field	DDC	
R/W Attribute	R0,W	
Protection Attribute	WS	
Initial Value	00000000_00000000_00000000_00000000	

[bit31:0] DDC[n]: Data Direction Clear Register (DDC[n]) (n=0 to 31)

DDC[n] n=0 to 31	Description
0	No effect
1	Clear the data direction selection bit (GPIO_DDRi: DD31 to DD0) to "0".

Notes:

- *This is the applicable key code register. To write to this register, the GPIO key code register (GPIO_KEYCDR) must be set.*
- *If this register is written before the key code is released, a bus error response is returned.*
- *After key code release, writing to a different address or using a different access size than specified in the key code sequence will not generate an error response.*
- *Please be sure to read and check a preset value after writing to a GPIO register.*

4.4. Port Output Data Register (GPIO_PODRi) (i = 0 to (product specification))

This register is used to set a port output value. The setting value is enabled when a port output is selected (GPIO_DDRi: DD31 to DD0 = 1). This register can be directly written or can be set/cleared using the port output set register (GPIO_POSRi) and the port output clear register (GPIO_POCRi).

Bit	31	0
Field	POD	
R/W Attribute	R/W	
Protection Attribute	WS	
Initial Value	00000000_00000000_00000000_00000000	

[bit31:0] POD[n]: Port Output Data Bit (n = 0 to 31)

POD[n] n=0 to 31	Description
0	Output "L".
1	Output "H".

Notes:

- *This is the applicable key code register. To write to this register, the GPIO key code register (GPIO_KEYCDR) must be set.*
- *If this register is written before the key code is released, a bus error response is returned.*
- *After key code release, writing to a different address or using a different access size than specified in the key code sequence will not generate an error response.*
- *Please be sure to read and check a preset value after writing to a GPIO register.*

4.5. Port Output Set Register (GPIO_POSRi) (i = 0 to (product specification))

This register is used to set the port output data bit (GPIO_PODRi: POD31 to POD0).

Bit	31	0
Field	POS	
R/W Attribute	R0,W	
Protection Attribute	WS	
Initial Value	00000000_00000000_00000000_00000000	

[bit31:0] POS[n]: Port Output Set Bit (n = 0 to 31)

POS[n] n=0 to 31	Description
0	No effect
1	Set the port output data bit (GPIO_PODRi: POD31 to POD0) to "1".

Notes:

- *This is the applicable key code register. To write to this register, the GPIO key code register (GPIO_KEYCDR) must be set.*
- *If this register is written before the key code is released, a bus error response is returned.*
- *After key code release, writing to a different address or using a different access size than specified in the key code sequence will not generate an error response.*
- *Please be sure to read and check a preset value after writing to a GPIO register.*

4.6. Port Output Clear Register (GPIO_POCRi) (i = 0 to (product specification))

This register is used to clear the port output data bit (GPIO_PODRi: POD31 to POD0).

Bit	31	0
Field	POC	
R/W Attribute	R0,W	
Protection Attribute	WS	
Initial Value	00000000_00000000_00000000_00000000	

[bit31:0] POC[n]: Port Output Clear Bit (n = 0 to 31)

POC[n] n=0 to 31	Description
0	No effect
1	Clear the port output data bit (GPIO_PODRi: POD31 to POD0) to "0".

Notes:

- *This is the applicable key code register. To write to this register, the GPIO key code register (GPIO_KEYCDR) must be set.*
- *If this register is written before the key code is released, a bus error response is returned.*
- *After key code release, writing to a different address or using a different access size than specified in the key code sequence will not generate an error response.*
- *Please be sure to read and check a preset value after writing to a GPIO register.*

4.7. Port Input Enable Register (GPIO_PORTEN)

This register sets the input rejection of a port.

Bit	31	8
Field	Reserved	
R/W Attribute	R0,WX	
Protection Attribute	WS	
Initial Value	00000000_00000000_00000000	

Bit	7	6	5	4	3	2	1	0
Field	Reserved						Reserved	GPORTEN
R/W Attribute	R0,WX						R/W0	R/W
Protection Attribute	WS							
Initial Value	000000						0	0

[bit31:1] Reserved: Reserved Bits

[bit0] GPORTEN: Global Input Enable Bit

Bit	Description
0	Disable input to the pin.
1	Enable input to the pin

Notes:

- *This is the applicable key code register. To write to this register, the GPIO key code register (GPIO_KEYCDR) must be set.*
- *If this register is written before the key code is released, a bus error response is returned.*
- *After key code release, writing to a different address or using a different access size than specified in the key code sequence will not generate an error response.*
- *Please be sure to read and check a preset value after writing to a GPIO register.*

4.8. Port Input Data Register (GPIO_PIDRi) (i = 0 to (product specification))

This register indicates an input data value. This register indicates input data according to the setting of the input level selection bit (PPC_PCFGRijj:PIL[1:0]) when the global input enable bit (GPIO_PORTEN:GPORTEN) is "1". An indefinite value is read from the port input data register when the global input enable bit (GPIO_PORTEN:GPORTEN) is "0".

Bit	31	0
Field	PID	
R/W Attribute	R,WX	
Protection Attribute	-	
Initial Value	XXXXXXXX_XXXXXXXX_XXXXXXXX_XXXXXXXX	

[bit31:0] PID[n]: Port Input Data Register (n = 0 to 31)

PID[n] n=0 to 31	Description
0	"L" is input.
1	"H" is input.

Note:

- *This register is not initialized by Hard Reset (i.e. “X”)*

4.9. GPIO Key Code Register (GPIO_KEYCDR)

This register sets register writing with a function for protection against erroneous writing. If writing to this register is not done using the prescribed method, writing to the relevant register is invalid.A lock target register can write only once in after unlocking.When writing continuously, please unlock once again.

Bit	31	30	29	28	27	26	25	24
Field	KEY		SIZE		Reserved			
R/W Attribute	R0,W		R0,W		R0,WX			
Protection Attribute	-							
Initial Value	00		00		0000			

Bit	23	22	21	20	19	18	17	16
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	-							
Initial Value	00000000							

Bit	15	14	13	12	11	10	9	8
Field	Reserved	RADR[14:8]						
R/W Attribute	R0,WX	R0,W						
Protection Attribute	-							
Initial Value	0	0000000						

Bit	7	6	5	4	3	2	1	0
Field	RADR[7:0]							
R/W Attribute	R0,W							
Protection Attribute	-							
Initial Value	00000000							

[bit31:30] KEY: Key Code Bits

These are key code setting bits. Write 0b00, 0b01, 0b10, and 0b11 continuously to these bits in this order. The key code setting becomes invalid immediately upon any writing to these bits in a different order. In such cases, set the key code again from the beginning.

Bits		Description
0	0	1st key code
0	1	2nd key code
1	0	3rd key code
1	1	4th key code

[bit29:28] SIZE: Access Size Bits

These bits set the access size for writing to the applicable key code register. Write the same data to these bits when writing key codes 0b00, 0b01, 0b10, and 0b11 in this order.

Bits		Description
0	0	Set byte access.
0	1	Set half-word access.
1	0	Set word access.
1	1	Reserved

[bit27:15] Reserved: Reserved Bits

[bit14:0] RADR: Port Address Bits

These bits set the lower 15 bits of the address of the applicable key code register. Write the same data to these bits when writing key codes 0b00, 0b01, 0b10, and 0b11 in this order.

RADR	Description
-	Set the lower 15 bits of the address of the applicable key code register.

Notes:

- *The key code setting becomes invalid upon any writing to the KEY[1:0] in any order than 0b00, 0b01, 0b10, and 0b11. In such case, set the key code again from the beginning.*
- *When different data is written to the SIZE[1:0] or the RADR[14:0] while writing the key code 0b00, 0b01, 0b10, and 0b11, the key code setting will become invalid. In such case, set it again from the beginning.*
- *The applicable key code registers are the following registers.*
  - Data direction register (GPIO_DDRi)
  - Data direction set register (GPIO_DDSRi)
  - Data direction clear register (GPIO_DDCRi)
  - Port Output Data Register (GPIO_PODRi)
  - Port Output Set Register (GPIO_POSRi)
  - Port Output Clear Register (GPIO_POCRi)
  - Port input enable register (GPIO_PORTEN)
- *This register is valid only for word access.*
- *There is no effect on the key code protection release process even if registers other than this register and the applicable key code register that is in the release processing (the register set by the RADR) are accessed while writing the key code 0b00, 0b01, 0b10, and 0b11.*

4.10. Port Setting Register (PPC_PCFGRijj) (i = 0 to (product specification), jj = 00 to 31)

This register sets and displays I/O from/to an external pin.

Bit	15	14	13	12	11	10	9	8
Field	POE	POD	PID	PIE	PIL		PUE	PDE
R/W Attribute	R,WX	R,WX	R,WX	R/W	R/W		R/W	R/W
Protection Attribute	WS							
Initial Value	0	0	X	0	00		0	0

Bit	7	6	5	4	3	2	1	0
Field	ODR		NFE	Reserved		POF		
R/W Attribute	R/W		R/W	R0,WX		R/W		
Protection Attribute	WS							
Initial Value	00		0	00		000		

[bit15] POE: Port Output Enable Bit

This bit indicates whether pin output is enabled.

Bit	Description
0	Pin output is high-impedance.
1	Pin output is enabled.

[bit14] POD: Port Output Data Bit

This bit indicates the value outputted to a pin.

Bit	Description
0	Output "L".
1	Output "H".

[bit13] PID: Port Input Data Bit

This bit indicates the value inputted to the pin selected by the input level bit (PIL[1:0]).

This bit is indefinite when the global input enable bit (GPORTEN) is "0".

Bit	Description
0	The pin input is "L".
1	The pin input is "H".

[bit12] PIE: Port Input Enable Bit

This bit selects a pin input.

Bit	Description
0	Pin input is disabled or analog input.
1	Pin input is enabled.

[bit11:10] PIL: Input Level Bit

This bit selects a pin input level.

Bits		Description
0	0	Type A
0	1	Type B
1	0	Type C
1	1	Type D

For details, see product specification.

[bit9] PUE: Pull Up Enable Bit

This bit set to enable pull-up resistor, when the port is configured as input.

Bit	Description
0	No pull up.
1	Pull up.

[bit8] PDE: Pull Down Enable Bit

This bit set to enable pull-down resistor, when the port is configured as input.

Pull up (PUE) has priority over pull down, if pull down and pull up (PUE) are competing.

Bit	Description
0	No pull down.
1	Pull down.

[bit7:6] ODR: Port Output Drive Selection Bit

This bit selects an output drive capacity of a port.

Bits		Description
0	0	Type A
0	1	Type B
1	0	Type C
1	1	Type D

For details, see product specification.

[bit5] NFE: Port Noise Filter Enable/Disable Select Bit

This bit selects whether to enable or disable the noise filter for the port.

Bit	Description
0	Disable the noise filter.
1	Enable the noise filter.

[bit4:3] Reserved: Reserved Bits

[bit2:0] POF: Port Output Function Selection Bit

This bit selects a function to output to a port.

Bits			Description
0	0	0	Resource A output
0	0	1	Resource B output
0	1	0	Resource C output
0	1	1	Resource D output
1	0	0	Resource E output
1	0	1	Resource F output
1	1	0	Resource G output
1	1	1	Resource H output

For details, see product specification.

Notes:

- *This is the applicable key code register. To write to this register, the PPC key code register (PPC_KEYCDR) must be set.*
- *If this register is written before the key code is released, a bus error response is returned.*
- *After key code release, writing to a different address or using a different access size than specified in the key code sequence will not generate an error response.*
- *Please be sure to read and check a preset value after writing to a GPIO register.*
- *This register is not initialized by Hard Reset (i.e. "X").*

4.11. PPC Key Code Register (PPC_KEYCDR)

This register sets register writing with a function for protection against erroneous writing. If writing to this register is not done using the prescribed method, writing to the relevant register is invalid.

Bit	31	30	29	28	27	26	25	24
Field	KEY		SIZE		Reserved			
R/W Attribute	R0,W		R0,W		R0,WX			
Protection Attribute	-							
Initial Value	00		00		0000			

Bit	23	22	21	20	19	18	17	16
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	-							
Initial Value	00000000							

Bit	15	14	13	12	11	10	9	8
Field	Reserved	RADR[14:8]						
R/W Attribute	R0,WX	R0,W						
Protection Attribute	-							
Initial Value	0	0000000						

Bit	7	6	5	4	3	2	1	0
Field	RADR[7:0]							
R/W Attribute	R0,W							
Protection Attribute	-							
Initial Value	00000000							

[bit31:30] KEY: Key Code Bits

These are key code setting bits. Write 0b00, 0b01, 0b10, and 0b11 continuously to these bits in this order. The key code setting becomes invalid immediately upon any writing to these bits in a different order. In such cases, set the key code again from the beginning.

Bits		Description
0	0	1st key code
0	1	2nd key code
1	0	3rd key code
1	1	4th key code

[bit29:28] SIZE: Access Size Bits

These bits set the access size for writing to the applicable key code register. Write the same data to these bits when writing key codes 0b00, 0b01, 0b10, and 0b11 in this order.

Bits		Description
0	0	Set byte access.
0	1	Set half-word access.
1	0	Set word access.
1	1	Reserved

[bit27:15] Reserved: Reserved Bits

[bit14:0] RADR: Port Address Bits

These bits set the lower 15 bits of the address of the applicable key code register. Write the same data to these bits when writing key codes 0b00, 0b01, 0b10, and 0b11 in this order.

RADR	Description
-	Set the lower 15 bits of the address of the applicable key code register.

Notes:

- *The key code setting becomes invalid upon any writing to the KEY[1:0] in any order than 0b00, 0b01, 0b10, and 0b11. In such case, set the key code again from the beginning.*
- *When different data is written to the SIZE[1:0] or the RADR[14:0] while writing the key code 0b00, 0b01, 0b10, and 0b11, the key code setting will become invalid. In such case, set it again from the beginning.*
- *The applicable key code register is the port setting register (PPC_PCFGRijj).*
- *This register is valid only for word access.*
- *There is no effect on the key code protection release process even if registers other than this register and the applicable key code register that is in the release processing (the register set by the RADR) are accessed while writing the key code 0b00, 0b01, 0b10, and 0b11.*

4.12. Resource Input Setting Register (RIC_RESINn) (n = 0 to (product specification))

This register selects an external pin input or an output from another internal resource as resource input.

Bit	15	14	13	12	11	10	9	8
Field	Reserved				PORTSEL			
R/W Attribute	R0,WX				R/W			
Protection Attribute	WS							
Initial Value	0000				0000			

Bit	7	6	5	4	3	2	1	0
Field	Reserved				RESSEL			
R/W Attribute	R0,WX				R/W			
Protection Attribute	WS							
Initial Value	0000				0000			

[bit15:12] Reserved: Reserved Bits

[bit11:8] PORTSEL: Resource Selection Bit

This bit selects an input to a corresponding resource.

Bits				Description
0	0	0	0	Source A
0	0	0	1	Source B
0	0	1	0	Source C
0	0	1	1	Source D
0	1	0	0	Source E
0	1	0	1	Source F
0	1	1	0	Source G
0	1	1	1	Source H
1	0	0	0	Source I
1	0	0	1	Source J
1	0	1	0	Source K
1	0	1	1	Source L
1	1	0	0	Source M
1	1	0	1	Source N
1	1	1	0	Source O
1	1	1	1	Source P

For details, see product specification.

[bit7:4] Reserved: Reserved Bits

[bit3:0] RESSEL: Resource Selection Bit

This bit selects an input to a resource.

Bits				Description
0	0	0	0	Source A
0	0	0	1	Source B
0	0	1	0	Source C
0	0	1	1	Source D
0	1	0	0	Source E
0	1	0	1	Source F
0	1	1	0	Source G
0	1	1	1	Source H
1	0	0	0	Source I
1	0	0	1	Source J
1	0	1	0	Source K
1	0	1	1	Source L
1	1	0	0	Source M
1	1	0	1	Source N
1	1	1	0	Source O
1	1	1	1	Source P

For assignment of source, see product specification.

Notes:

- *The PORTSEL is the applicable key code bit. To write to this bit, the RIC key code register (RIC_KEYCDR) must be set.*
- *If this bit is written before the key code is released, a bus error response is returned.*
- *The RESSEL is the applicable key code bit. To write to this bit, the RIC key code register (RIC_KEYCDR) must be set.*
- *If this bit is written before the key code is released, a bus error response is returned.*
- *After key code release, writing to a different address or using a different access size than specified in the key code sequence will not generate an error response.*
- *Please be sure to read and check a preset value after writing to a GPIO register.*

4.13. RIC Key Code Register (RIC_KEYCDR)

This register sets register writing with a function for protection against erroneous writing. If writing to this register is not done using the prescribed method, writing to the relevant register is invalid.

Bit	31	30	29	28	27	26	25	24
Field	KEY		SIZE		Reserved			
R/W Attribute	R0,W		R0,W		R0,WX			
Protection Attribute	-							
Initial Value	00		00		0000			

Bit	23	22	21	20	19	18	17	16
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute	-							
Initial Value	00000000							

Bit	15	14	13	12	11	10	9	8
Field	Reserved	RADR[14:8]						
R/W Attribute	R0,WX	R0,W						
Protection Attribute	-							
Initial Value	0	0000000						

Bit	7	6	5	4	3	2	1	0
Field	RADR[7:0]							
R/W Attribute	R0,W							
Protection Attribute	-							
Initial Value	00000000							

[bit31:30] KEY: Key Code Bits

These are key code setting bits. Write 0b00, 0b01, 0b10, and 0b11 continuously to these bits in this order. The key code setting becomes invalid immediately upon any writing to these bits in a different order. In such cases, set the key code again from the beginning.

Bits		Description
0	0	1st key code
0	1	2nd key code
1	0	3rd key code
1	1	4th key code

[bit29:28] SIZE: Access Size Bits

These bits set the access size for writing to the applicable key code register. Write the same data to these bits when writing key codes 0b00, 0b01, 0b10, and 0b11 in this order.

Bits		Description
0	0	Set byte access.
0	1	Set half-word access.
1	0	Set word access.
1	1	Reserved

[bit27:15] Reserved: Reserved Bits

[bit14:0] RADR: Port Address Bits

These bits set the lower 15 bits of the address of the applicable key code register. Write the same data to these bits when writing key codes 0b00, 0b01, 0b10, and 0b11 in this order.

RADR	Description
-	Set the lower 15 bits of the address of the applicable key code register.

Notes:

- *The key code setting becomes invalid upon any writing to the KEY[1:0] in any order than 0b00, 0b01, 0b10, and 0b11. In such case, set the key code again from the beginning.*
- *When different data is written to the SIZE[1:0] or the RADR[14:0] while writing the key code 0b00, 0b01, 0b10, and 0b11, the key code setting will become invalid. In such case, set it again from the beginning.*
- *The applicable key code register is the resource input setting register (RIC_RESINn).*
- *This register is valid only for word access.*
- *There is no effect on the key code protection release process even if registers other than this register and the applicable key code register that is in the release processing (the register set by the RADR) are accessed while writing the key code 0b00, 0b01, 0b10, and 0b11.*
  When key code protection is released, register of protection target can be written once.
  If several registers are to be written, it is necessary to release the protection for each register writing.

5. Precautions for Using This Device

Notes on switching the I/O port function are shown below.

A glitch may occur for a brief second (2 or 3 ns) when a general-purpose I/O port is switched (from input to output or from output to input or from port function to resource or from resource to port function).

If this glitch may cause a problem for the system, please write a value to port output data register (GPIO_PODRi) in advance at a level that will not cause a problem.

When a resource input is valid and its assignment is moved to another pin, a trigger which causes resource operation may occur if the pin levels before switching and after switching are different.

Therefore, please switch input pins when the resource input function is stopped.

CHAPTER 51: Peripheral Protection Unit

This chapter explains Peripheral Protection Unit (PPU).

1. Overview
2. Configuration and Block Diagram
3. Operation of the PPU
4. Registers
5. Setup Steps of PPU Register

PPU-TXXPT03P01R01L06-E1-XX

1. Overview

This section gives a brief overview of Peripheral Protection Unit.

The Peripheral Protection Unit (PPU) controls and monitors unauthorized access from the master to the peripherals against PPU configurations.

- Used to protect peripherals from unauthorized access.
- Provide read and write access attribute for peripherals.
- The attributes can be set independently in user mode and privileged mode.
- Support two types of attribute
  - Read/Access mode: Write access permission always includes read access permission.
  - Read/Write mode: Read access permission can be given independently to the write access permission.
- Return bus error response for the violated accesses.
- Have status registers to give information for the violations.
- The PPU supports privilege mode forced change function and it changes to privilege mode a privilege level of a peripheral side interface.

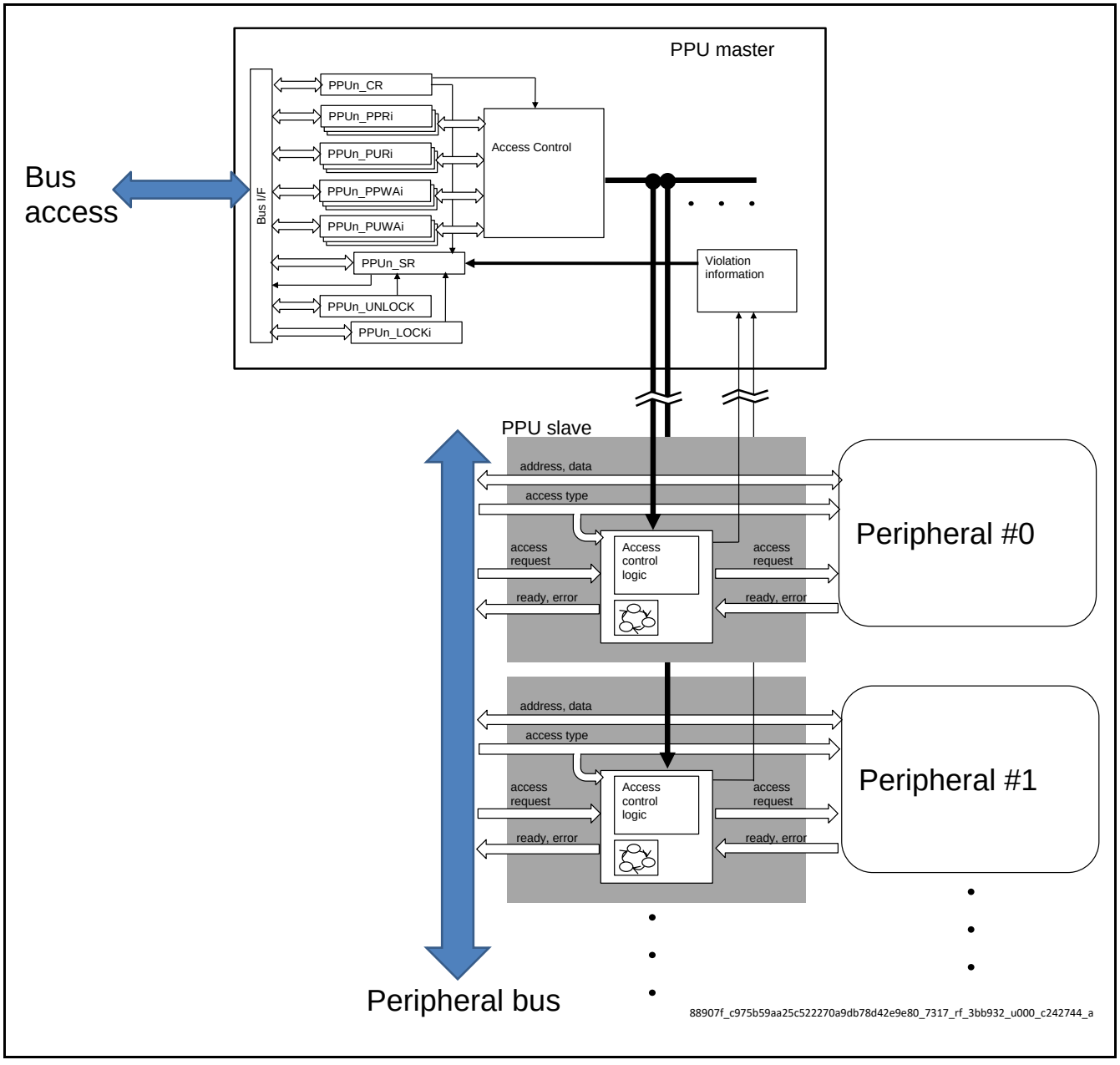
Only CPU is the bus master which is attached by PPU.

Refer PLATFORM OVERVIEW chapter to confirm which peripherals are covered by the PPU.

2. Configuration and Block Diagram

This section shows a block diagram of PPU.

Figure 2-1 Block Diagram



3. Operation of the PPU

This section describes the operation of PPU.

3.1. Access Type and PPU Operation

In this section, PPU protection behavior in each access type is tabulated.

The PPU can identify access type (read or write) and privilege level (privilege mode or user mode). The PPU permits or forbids access to the peripheral according to the PPU configuration.

The permission is configured according to PPU operation mode which is configured by PPU0_CR:MODE bit and PPU0_PPRi/PURi/PPWAi/PUWAi registers. Following table describes the access permissions in each configuration.

Table 3-1 Access Permission at Read/Write Mode in Privilege Mode Access

Read Attribute (PPU0_PPRi)	Write Attribute (PPU0_PPWAi)	Read Access in Privilege Mode	Write Access in Privilege Mode
0	0	Forbidden	Forbidden
0	1	Forbidden	Permitted
1	0	Permitted	Forbidden
1	1	Permitted	Permitted

Table 3-2 Access Permission at Read/Access Mode in Privilege Mode Access

Read Attribute (PPU0_PPRi)	Access Attribute (PPU0_PPWAi)	Read Access in Privilege Mode	Write Access in Privilege Mode
0	0	Forbidden	Forbidden
1	0	Permitted	Forbidden
0 or 1	1	Permitted	Permitted

Table 3-3 Access Permission at Read/Write Mode in User Mode Access

Read Attribute (PPU0_PURi)	Write Attribute (PPU0_PUWAi)	Read Access in User Mode	Write Access in User Mode
0	0	Forbidden	Forbidden
0	1	Forbidden	Permitted
1	0	Permitted	Forbidden
1	1	Permitted	Permitted

Table 3-4 Access Permission at Read/Access Mode in User Mode Access

Read Attribute (PPU0_PURi)	Access Attribute (PPU0_PUWAi)	Read Access in User Mode	Write Access in User Mode
0	0	Forbidden	Forbidden
1	0	Permitted	Forbidden
0 or 1	1	Permitted	Permitted

Accessing to peripheral in forbidden type, the violation will be occurred. For the access violation, refer 3.3 Access Violation for more details.

3.2. Access Protection to PPU Register

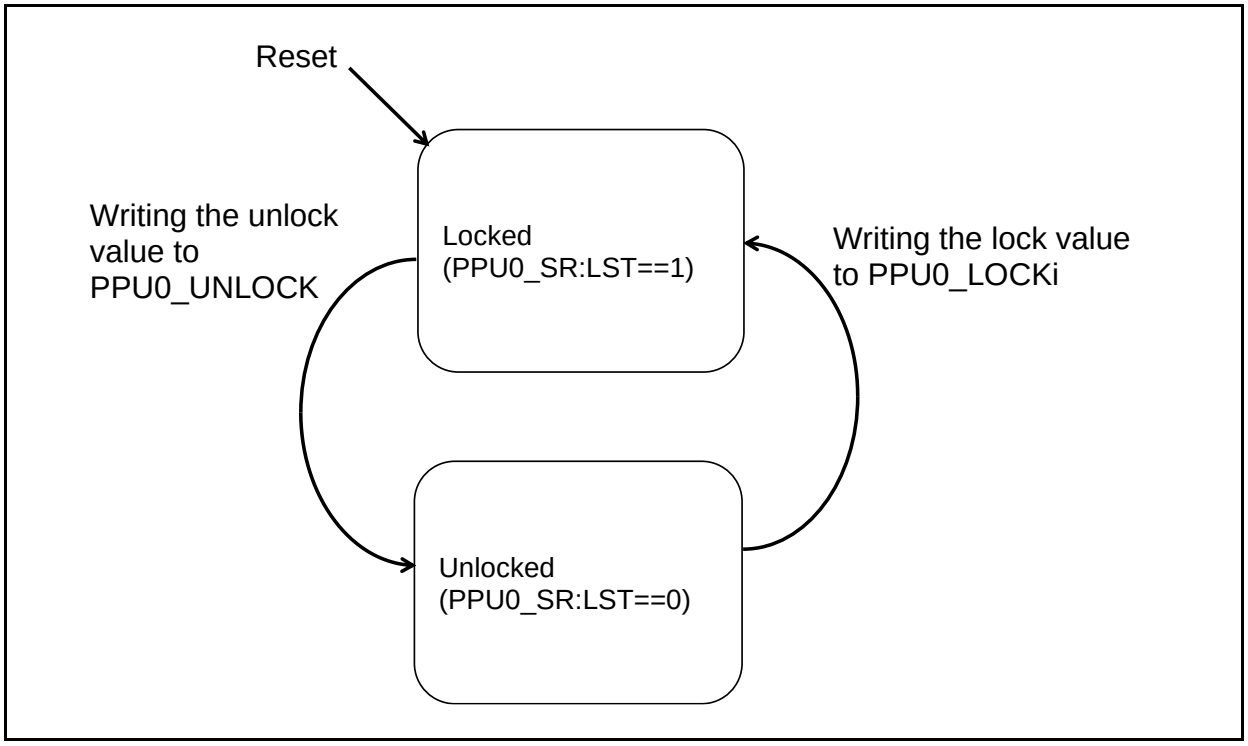
In this section explains access protection scheme to PPU registers.

The PPU has lock/unlock mechanism to writing its registers. To write the registers, it is needed to unlock the PPU by writing unlock value. PPU can be locked using PPU0_LOCKi register. To reconfigure PPU again, you need to unlock the PPU by writing PPU0_UNLOCK register again.

The lock status is always shown in PPU0_CR:LST bit.

There are several lock registers in PPU(Substance is one.). PPU can be locked by writing the lock value in either of the lock registers. They are prepared for a convenience for locking PPU by a programming by DMA transfer(Please refer from Table 3-5 to Table 3-8.).

Figure 3-1 Lock and Unlock



The writing procedure to the PPU0_PPRi register by the DMA forwarding is shown(Please refer to Table 3-5.).

Table 3-5 Write to PPU0_PPRi by DMA Transfer

No.	Offset	Register Name	Value
1	0x008	PPU0_UNLOCK	Unlock value（0x35AB16CC）
2	0x00C	PPU0_WPQCLR	Setting which does not trigger Quick Clear (WPQCLR = 0)
3	0x010	PPU0_WUQCLR	Setting which does not trigger Quick Clear (WUQCLR = 0)
4	0x014	PPU0_RPQCLR	Setting which does not trigger Quick Clear (RPQCLR = 0)
5	0x018	PPU0_RUQCLR	Setting which does not trigger Quick Clear (RUQCLR = 0)
6	0x01C to 0x07F	Reserved	Dummy value
7	0x080 to 0x0F4	PPU0_PPRi	Set value
8	0x0F8	PPU0_LOCK0	Lock value（0x30004000）

The writing procedure to PPU0_PPRi and the PPU0_PURi register by the DMA forwarding is shown(Please refer to Table 3-6).

Table 3-6 Write to PPU0_PPRi, PPU0_PURi by DMA Transfer

No.	Offset	Register Name	Value
1	0x008	PPU0_UNLOCK	Unlock value（0x35AB16CC）
2	0x00C	PPU0_WPQCLR	Setting which does not trigger Quick Clear (WPQCLR = 0)
3	0x010	PPU0_WUQCLR	Setting which does not trigger Quick Clear (WUQCLR = 0)
4	0x014	PPU0_RPQCLR	Setting which does not trigger Quick Clear (RPQCLR = 0)
5	0x018	PPU0_RUQCLR	Setting which does not trigger Quick Clear (RUQCLR = 0)
6	0x01C to 0x07F	Reserved	Dummy value
7	0x080 to 0x0F4	PPU0_PPRi	Set value
8	0x0F8	PPU0_LOCK0	Dummy value（Excluding　0x30004000）
9	0x0FC	Reserved	Dummy value
10	0x100 to 0x174	PPU0_PURi	Set value
11	0x178	PPU0_LOCK1	Lock value（0x30004000）

The writing procedure to PPU0_PPRi, PPU0_PURi, and the PPU0_PPWAi register by the DMA forwarding is shown(Please refer to Table 3-7).

Table 3-7 Write to PPU0_PPRi, PPU0_PURi, PPU0_PPWAi by DMA Transfer

No.	Offset	Register Name	Value
1	0x008	PPU0_UNLOCK	Unlock value（0x35AB16CC）
2	0x00C	PPU0_WPQCLR	Setting which does not trigger Quick Clear (WPQCLR = 0)
3	0x010	PPU0_WUQCLR	Setting which does not trigger Quick Clear (WUQCLR = 0)
4	0x014	PPU0_RPQCLR	Setting which does not trigger Quick Clear (RPQCLR = 0)
5	0x018	PPU0_RUQCLR	Setting which does not trigger Quick Clear (RUQCLR = 0)
6	0x01C to 0x07F	Reserved	Dummy value
7	0x080 to 0x0F4	PPU0_PPRi	Set value
8	0x0F8	PPU0_LOCK0	Dummy value（Excluding　0x30004000）
9	0x0FC	Reserved	Dummy value
10	0x100 to 0x174	PPU0_PURi	Set value
11	0x178	PPU0_LOCK1	Dummy value（Excluding　0x30004000）
12	0x17C	Reserved	Dummy value
13	0x180 to 0x1F4	PPU0_PPWAi	Set value
14	0x1F8	PPU0_LOCK2	Lock value（0x30004000）

The writing procedure to PPU0_PPRi, PPU0_PURi, PPU0_PPWAi, and the PPU0_PUWAi register by the DMA forwarding is shown(Please refer to Table 3-8).

Table 3-8 Write to PPU0_PPRi, PPU0_PURi, PPU0_PPWAi, PPU0_PUWAi by DMA Transfer

No.	Offset	Register Name	Value
1	0x008	PPU0_UNLOCK	Unlock value（0x35AB16CC）
2	0x00C	PPU0_WPQCLR	Setting which does not trigger Quick Clear (WPQCLR = 0)
3	0x010	PPU0_WUQCLR	Setting which does not trigger Quick Clear (WUQCLR = 0)
4	0x014	PPU0_RPQCLR	Setting which does not trigger Quick Clear (RPQCLR = 0)
5	0x018	PPU0_RUQCLR	Setting which does not trigger Quick Clear (RUQCLR = 0)
6	0x01C to 0x07F	Reserved	Dummy value
7	0x080 to 0x0F4	PPU0_PPRi	Set value
8	0x0F8	PPU0_LOCK0	Dummy value（Excluding　0x30004000）
9	0x0FC	Reserved	Dummy value
10	0x100 to 0x174	PPU0_PURi	Set value
11	0x178	PPU0_LOCK1	Dummy value（Excluding　0x30004000）
12	0x17C	Reserved	Dummy value
13	0x180 to 0x1F4	PPU0_PPWAi	Set value
14	0x1F8	PPU0_LOCK2	Dummy value（Excluding　0x30004000）
15	0x1FC	Reserved	Dummy value
16	0x200 to 0x274	PPU0_PUWAi	Set value
17	0x278	PPU0_LOCK3	Lock value（0x30004000）

The writing procedure to PPU0_PPRi, PPU0_PURi, PPU0_PPWAi, PPU0_PUWAi and the PPU0_PFENi register by the DMA forwarding is shown(Please refer to Table 3-9).

Table 3-9 Write to PPU0_PPRi, PPU0_PURi, PPU0_PPWAi, PPU0_PUWAi, PPU0_PFENi by DMA Transfer

No.	Offset	Register Name	Value
1	0x008	PPU0_UNLOCK	Unlock value（0x35AB16CC）
2	0x00C	PPU0_WPQCLR	Setting which does not trigger Quick Clear (WPQCLR = 0)
3	0x010	PPU0_WUQCLR	Setting which does not trigger Quick Clear (WUQCLR = 0)
4	0x014	PPU0_RPQCLR	Setting which does not trigger Quick Clear (RPQCLR = 0)
5	0x018	PPU0_RUQCLR	Setting which does not trigger Quick Clear (RUQCLR = 0)
6	0x01C to 0x07F	Reserved	Dummy value
7	0x080 to 0x0F4	PPU0_PPRi	Set value
8	0x0F8	PPU0_LOCK0	Dummy value（Excluding　0x30004000）
9	0x0FC	Reserved	Dummy value
10	0x100 to 0x174	PPU0_PURi	Set value
11	0x178	PPU0_LOCK1	Dummy value（Excluding　0x30004000）
12	0x17C	Reserved	Dummy value
13	0x180 to 0x1F4	PPU0_PPWAi	Set value
14	0x1F8	PPU0_LOCK2	Dummy value（Excluding　0x30004000）
15	0x1FC	Reserved	Dummy value
16	0x200 to 0x274	PPU0_PUWAi	Set value
17	0x278	PPU0_LOCK3	Dummy value（Excluding　0x30004000）
18	0x27C	Reserved	Dummy value
19	0x280 to 0x2F4	PPU0_PFENi	Set value
20	0x2F8	PPU0_LOCK4	Lock value（0x30004000）

3.3. Access Violation

In this section explains behavior at access violation.

When violated access to the peripheral, PPU, corresponding peripheral, and the bus master (i.e. CPU) behaves as following.

- The access will cause a bus error response. The bus error response in CPU will trigger an exception.
- The access to the peripheral is blocked by the PPU. So the peripheral is not affected by the access.
- The error information is set to PPU0_SR register.

To obtain the violation information, read PPU0_SR register in the ISR for the exception at bus error response.

Figure 3-2 and Figure 3-3 are comparing behaviors for each permission.

Figure 3-2 Access Operation When the Access is Forbidden

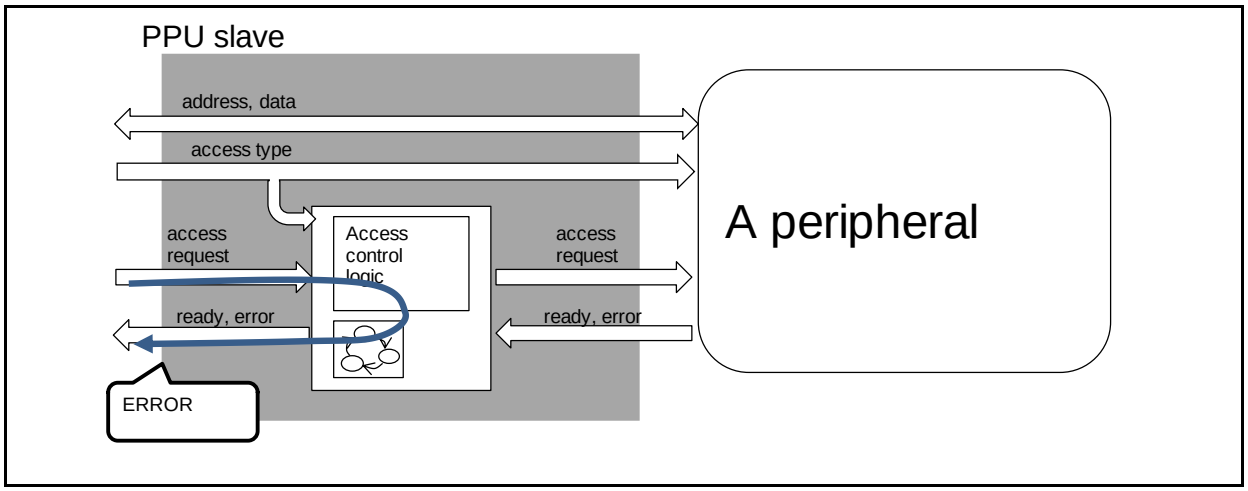
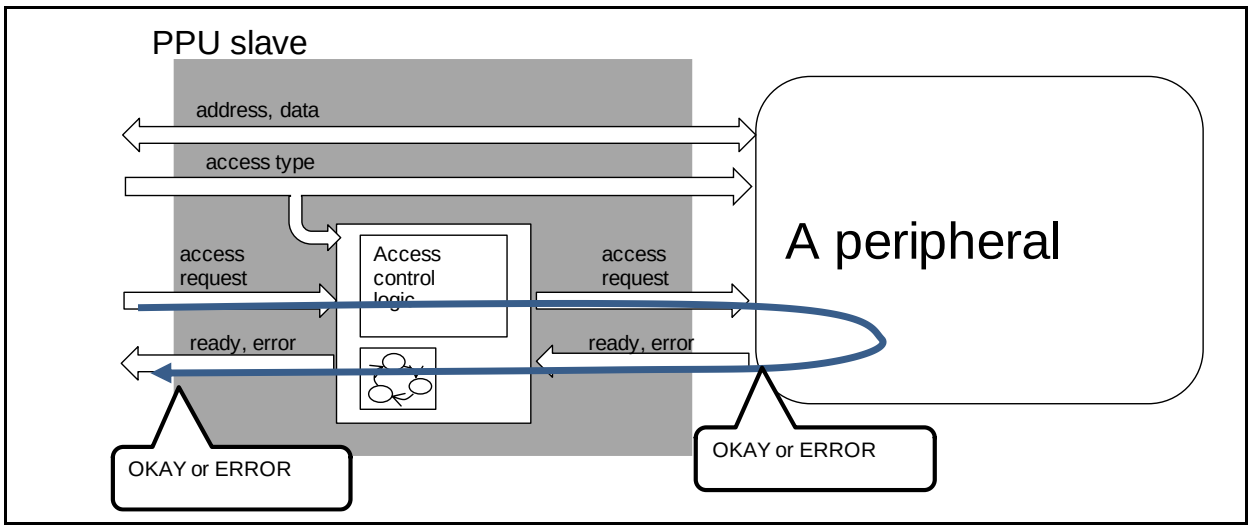


Figure 3-3 Access Operation When the Access is Permitted



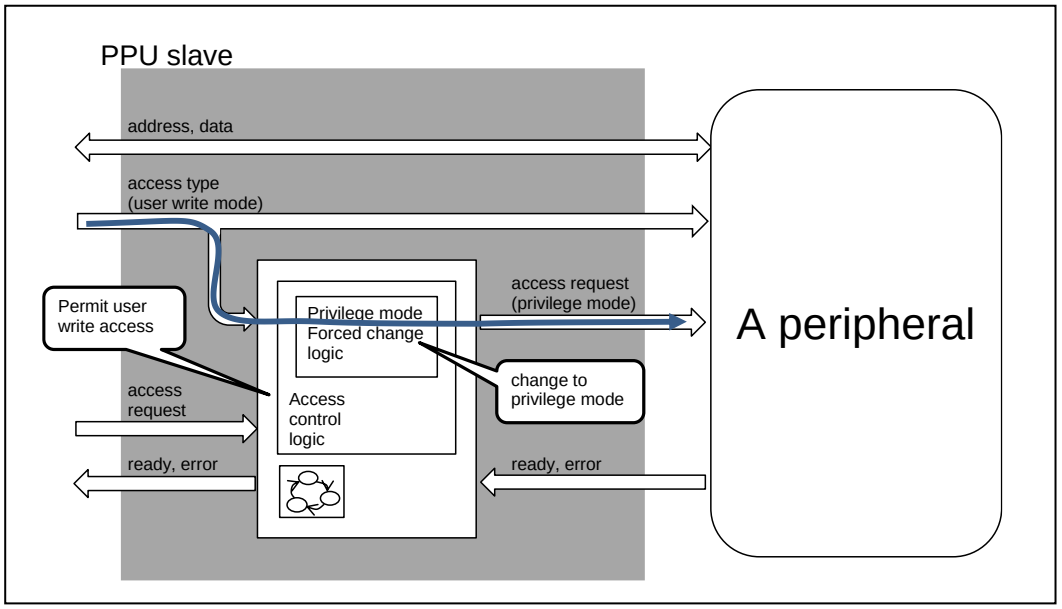
3.4. Privilege Mode Forced Change Function

In this section explains privilege mode forced change function.

The PPU supports privilege mode forced change function.

When this mode is enabled (PPU0_PFENi:PFEN[x] = 1), privilege level on a peripheral side interface is set to privilege mode as explained in Section 4.13 “PPU Privilege Mode Forced change function Enable Register (PPU0_PFENi)”. With this function, even if it is prohibited write access in user mode from bus master (i.e. CPU) to the peripherals, write access in user mode is enabled (Please refer to Figure 3-4).

Figure 3-4 Privilege Mode Forced Change Function



Note:

- *When privilege mode forced change function is enabled, the PPU controls the accesses to a peripheral using the privilege level before forced change (Please refer to following table).*

Access mode from the bus master	PPU0_PFENi:PFEN[x]	Access mode to the peripheral	Access mode to be used for PPU access control
User mode	0	User mode	User mode
Privilege mode	Don’t care	Privilege mode	Privilege mode
User mode	1	Privilege mode	User mode

4. Registers

This section describes the registers of PPU.

Table 4-1 Register Map

Offset	Register Name / Initial Value			
	+3	+2	+1	+0
0x000	PPU0_CR 00000000_00000000_00000000_00000000			
0x004	PPU0_SR 00000000_00000000_00000000_00000001			
0x008	PPU0_UNLOCK 00000000_00000000_00000000_00000000			
0x00C	PPU0_WPQCLR 00000000_00000000_00000000_00000000			
0x010	PPU0_WUQCLR 00000000_00000000_00000000_00000000			
0x014	PPU0_RPQCLR 00000000_00000000_00000000_00000000			
0x018	PPU0_RUQCLR 00000000_00000000_00000000_00000000			
0x01C 0x07F	Reserved			
0x080 0x0F4	PPU0_PPR0 - PPU0_PPR29 11111111_11111111_11111111_11111111			
0x0F8	PPU0_LOCK0 00000000_00000000_00000000_00000000			
0x0FC	Reserved			
0x100 0x174	PPU0_PUR0 - PPU0_PUR29 11111111_11111111_11111111_11111111			
0x178	PPU0_LOCK1 00000000_00000000_00000000_00000000			
0x17C	Reserved			
0x180 0x1F4	PPU0_PPWA0 - PPU0_PPWA29 11111111_11111111_11111111_11111111			
0x1F8	PPU0_LOCK2 00000000_00000000_00000000_00000000			
0x1FC	Reserved			
0x200 0x274	PPU0_PUWA0 - PPU0_PUWA29 00000000_00000000_00000000_00000000			
0x278	PPU0_LOCK3 00000000_00000000_00000000_00000000			

Offset	Register Name / Initial Value			
	+3	+2	+1	+0
0x27C	Reserved			
0x280 0x2F4	PPU0_PFEN0 - PPU0_PFEN29 00000000_00000000_00000000_00000000			
0x2F8	PPU0_LOCK4 00000000_00000000_00000000_00000000			
0x2FC	Reserved			

Accessing reserved area has no effects.

4.1. PPU Control Register (PPU0_CR)

The register controls overall of the PPU operation. To write this register, you have to unlock the PPU by writing the unlock value into PPU Unlock Register.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	MODE	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0, WX	R0, WX	R/W	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX
PROT_TYPE	WPS							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	VCLR	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0, W	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX
PROT_TYPE	WPS							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	FPQ SET	FPQ CLR
ACCESS_TYPE	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX	R0, W	R0, W
PROT_TYPE	WPS							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX
PROT_TYPE	WPS							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit29] MODE : PPU Mode Bit

This bit specifies the operation mode of PPU.

Bit	Description
0	Read/Write mode Read access permission can be given independently to the write access permission. Please refer to 3.1 Access Type and PPU Operation for more details.
1	Read/Access mode Write access permission always includes read access permission. Please refer to 3.1 Access Type and PPU Operation for more details.

[bit23] VCLR : PPU Violation Information Clear

Writing "1" to this bit clears VD, VP, VW and VL[4:0] bits in PPU0_SR register. Writing "0" has no effect.

Bit	Description
0	No effect.
1	Each bit of VD, VP, VW, and VL [4:0] of the PPU0_SR register is cleared.

[bit9] FPQSET : PPU Privilege Mode Forced Change Function Quick Set

Writing "1" to this bit sets PPU0_PFENi (i=0 to 29) registers. Writing "0" has no effect.

Bit	Description
0	No effect.
1	Each bit of the PPU0_PFENi (i=0 to 29) registers is set.

[bit8] FPQCLR : PPU Privilege Mode Forced Change Function Quick Clear

Writing "1" to this bit clears PPU0_PFENi (i=0 to 29) registers. Writing "0" has no effect.

Bit	Description
0	No effect.
1	Each bit of the PPU0_PFENi (i=0 to 29) registers is cleared.

Notes:

- *The access from the CPU to the corresponding peripheral, please go after writing to the MODE bit is completed.*
- *Writing to FPQSET/FPQCLR bit, please go to when access to the corresponding peripheral from the CPU does not occur.*
- *In case of writing "1" to FPQSET bit and FPQCLR bit at the same time, FPQCLR bit is set.*

4.2. PPU Status Register (PPU0_SR)

The register shows the status of PPU.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	VD	VP	VW	VL[4]	VL[3]	VL[2]	VL[1]	VL[0]
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	LST
ACCESS_TYPE	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX	R, WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	1

[bit23] VD : Violation Detection Bit

Then violated access to the peripheral against corresponding PPU configuration is found, the bit shall be set to "1". The bit is cleared by writing "1" to PPU0_CR:VCLR.

Bit	Description
0	The violation access was not generated.
1	The violation access was generated.

[bit22] VP : Violation Privileged Level Bit

This register contains information for privilege level of the latest violated access. The bit is cleared by writing "1" to PPU0_CR:VCLR.

Bit	Description
0	The access was done in user mode.
1	The access was done in privilege mode.

[bit21] VW : Violation Write Access Bit

This register contains information for access type (read or write) of the latest violated access. The bit is cleared by writing "1" to PPU0_CR:VCLR.

Bit	Description
0	The access was read access.
1	The access was write access.

[bit20:16] VL[4:0] : Violation Location

This register contains information for access destination of the latest violated access. As application specific PPU protection area, areas of up to seven are available.

Bits	Description
00000	Reserved
00001	Violated access was done to a register in MCU Config Group.
00010	Violated access was done to a register in SYSC1.
00011	Violated access was done to a register in Memory Config Group orSystem SRAM.
00100	Violated access was done to a register in CPERI#0.
00101	Violated access was done to a register in CPERI#1.
00110	Violated access was done to a register in CPERI#2.
00111	Violated access was done to a register in EBI Group.
01000	Violated access was done to a register in SHE Group.
01001	Violated access was done to a register in DDRHSSPI Group.
01010	Violated access was done to a register in Application specific area #0.
01011	Violated access was done to a register in Application specific area #1.
01100	Violated access was done to a register in Application specific area #2.
01101	Violated access was done to a register in Application specific area #3.
01110	Violated access was done to a register in Application specific area #4.
01111	Violated access was done to a register in Application specific area #5.
10000	Violated access was done to a register in Application specific area #6.
Else	Reserved

[bit0] LST : Lock Status Bit

The bit has to be "0" to program the PPU. Please refer to 3.2 Access Protection to PPU Register for more details.

Bit	Description
0	Unlocked
1	Locked

Note:

- *Priority of settings to PPU0_SR in the case where the access violation occurs simultaneously in several groups is as follows.*

Group Name	Priorities
MCU Config Group	Highest level
SYSC1 Group	↑ (Higher)
Memory Config Group	
System SRAM	
CPERI#0	
CPERI#1	
CPERI#2	
EBI Group	
SHE Group	
DDRHSSPI Group	
Application specific area #0	
Application specific area #1	
Application specific area #2	
Application specific area #3	
Application specific area #4	
Application specific area #5	↓ (Lower)
Application specific area #6	Lowest level

4.3. PPU Unlock Register (PPU0_UNLOCK)

The register unlocks the access to PPU register. Without locking, write access to specific PPU registers results a bus error response. By writing the unlock value (0x35AB16CC), the PPU is unlocked.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	UL[31]	UL[30]	UL[29]	UL[28]	UL[27]	UL[26]	UL[25]	UL[24]
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE	WP							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	UL[23]	UL[22]	UL[21]	UL[20]	UL[19]	UL[18]	UL[17]	UL[16]
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE	WP							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	UL[15]	UL[14]	UL[13]	UL[12]	UL[11]	UL[10]	UL[9]	UL[8]
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE	WP							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	UL[7]	UL[6]	UL[5]	UL[4]	UL[3]	UL[2]	UL[1]	UL[0]
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE	WP							
INITIAL_VALUE	0	0	0	0	0	0	0	0

Note:

- *Writing by byte or half-word access to this register results a bus error response.*

[bit31:0] UL : Unlock

By writing 0x35AB16CC, the PPU is unlocked and PPU0_SR:LST is set to "0". Writing any other value has no effect.

4.4. PPU Privileged Write Attribute Quick Clear Register (PPU0_WPQCLR)

The register clears collectively write attribute (in Read/Write mode) or access attribute (in Read/Access mode) of the corresponding peripherals in privileged mode access. To write this register, you have to unlock the PPU by writing the unlock value into PPU Unlock Register.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX
PROT_TYPE	WPS							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX
PROT_TYPE	WPS							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX
PROT_TYPE	WPS							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	WPQCLR
ACCESS_TYPE	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX	R0, W
PROT_TYPE	WPS							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit0] WPQCLR : PPU Privileged Write Attribute Quick Clear

Writing "1" to this bit clears PPU0_PPWAi (i=0 to 29) registers. Writing "0" has no effect.

Bit	Description
0	No effect.
1	Each bit of the PPU0_PPWAi (i=0 to 29) registers is cleared.

Notes:

- *The access from the CPU to the corresponding peripheral, please go after writing to the MODE bit is completed.*
- *Writing to WPQCLR bit, please go to when access to the corresponding peripheral from the CPU does not occur.*

4.5. PPU User Write Attribute Quick Clear Register (PPU0_WUQCLR)

The register clears collectively write attribute (in Read/Write mode) or access attribute (in Read/Access mode) of the corresponding peripherals in user mode access. To write this register, you have to unlock the PPU by writing the unlock value into PPU Unlock Register.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX
PROT_TYPE	WPS							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX
PROT_TYPE	WPS							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX
PROT_TYPE	WPS							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	WUQCLR
ACCESS_TYPE	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX	R0, W
PROT_TYPE	WPS							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit0] WUQCLR : PPU User Write Attribute Quick Clear

Writing "1" to this bit clears PPU0_PUWAi (i=0 to 29) registers. Writing "0" has no effect.

Bit	Description
0	No effect.
1	Each bit of the PPU0_PUWAi (i=0 to 29) registers is cleared.

Notes:

- *The access from the CPU to the corresponding peripheral, please go after writing to the MODE bit is completed.*
- *Writing to WUQCLR bit, please go to when access to the corresponding peripheral from the CPU does not occur.*

4.6. PPU Privileged Read Attribute Quick Clear Register (PPU0_RPQCLR)

The register clears collectively read attribute of the corresponding peripherals in privileged mode access. To write this register, you have to unlock the PPU by writing the unlock value into PPU Unlock Register.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX
PROT_TYPE	WPS							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX
PROT_TYPE	WPS							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX
PROT_TYPE	WPS							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	RPQCLR
ACCESS_TYPE	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX	R0, W
PROT_TYPE	WPS							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit0] RPQCLR : PPU Privileged Read Attribute Quick Clear

Writing "1" to this bit clears PPU0_PPRi (i=0 to 29) registers. Writing "0" has no effect.

Bit	Description
0	No effect.
1	Each bit of the PPU0_PPRi (i=0 to 29) registers is cleared.

Notes:

- *The access from the CPU to the corresponding peripheral, please go after writing to the MODE bit is completed.*
- *Writing to RPQCLR bit, please go to when access to the corresponding peripheral from the CPU does not occur.*

4.7. PPU User Read Attribute Quick Clear Register (PPU0_RUQCLR)

The register clears collectively read attribute of the corresponding peripherals in user mode access. To write this register, you have to unlock the PPU by writing the unlock value into PPU Unlock Register.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX
PROT_TYPE	WPS							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX
PROT_TYPE	WPS							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX
PROT_TYPE	WPS							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	RUQCLR
ACCESS_TYPE	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX	R0, WX	R0, W
PROT_TYPE	WPS							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit0] RUQCLR : PPU User Read Attribute Quick Clear

Writing "1" to this bit clears PPU0_PURi (i=0 to 29) registers. Writing "0" has no effect.

Bit	Description
0	No effect.
1	Each bit of the PPU0_PURi (i=0 to 29) registers is cleared.

Notes:

- *The access from the CPU to the corresponding peripheral, please go after writing to the MODE bit is completed.*
- *Writing to RUQCLR bit, please go to when access to the corresponding peripheral from the CPU does not occur.*

4.8. PPU Privileged Read Attribute Register (PPU0_PPRi)

The register sets read attribute of the corresponding peripherals in privileged mode access. To write this register, you have to unlock the PPU by writing the unlock value into PPU Unlock Register.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	PPR[31]	PPR[30]	PPR[29]	PPR[28]	PPR[27]	PPR[26]	PPR[25]	PPR[24]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	WPS							
INITIAL_VALUE	1	1	1	1	1	1	1	1

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	PPR[23]	PPR[22]	PPR[21]	PPR[20]	PPR[19]	PPR[18]	PPR[17]	PPR[16]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	WPS							
INITIAL_VALUE	1	1	1	1	1	1	1	1

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	PPR[15]	PPR[14]	PPR[13]	PPR[12]	PPR[11]	PPR[10]	PPR[9]	PPR[8]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	WPS							
INITIAL_VALUE	1	1	1	1	1	1	1	1

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	PPR[7]	PPR[6]	PPR[5]	PPR[4]	PPR[3]	PPR[2]	PPR[1]	PPR[0]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	WPS							
INITIAL_VALUE	1	1	1	1	1	1	1	1

[bit31:0] PPR[x] : PPU Privileged Mode Read Attribute Bit (x=31 to 0)

Read attribute for Privileged mode access of the peripheral #(i*32+x).

PPR [x] x=31 to 0	Description
0	Read attribute in privileged mode is disabled.
1	Read attribute in privilege mode is enabled.

Note:

- *The access from the CPU to the corresponding peripheral, please go after writing to the PPR[x] bit is completed.*

4.9. PPU User Read Attribute Register (PPU0_PURi)

The register sets read attribute of the corresponding peripherals in user mode access. To write this register, you have to unlock the PPU by writing the unlock value into PPU Unlock Register.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	PUR[31]	PUR[30]	PUR[29]	PUR[28]	PUR[27]	PUR[26]	PUR[25]	PUR[24]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	WPS							
INITIAL_VALUE	1	1	1	1	1	1	1	1

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	PUR[23]	PUR[22]	PUR[21]	PUR[20]	PUR[19]	PUR[18]	PUR[17]	PUR[16]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	WPS							
INITIAL_VALUE	1	1	1	1	1	1	1	1

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	PUR[15]	PUR[14]	PUR[13]	PUR[12]	PUR[11]	PUR[10]	PUR[9]	PUR[8]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	WPS							
INITIAL_VALUE	1	1	1	1	1	1	1	1

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	PUR[7]	PUR[6]	PUR[5]	PUR[4]	PUR[3]	PUR[2]	PUR[1]	PUR[0]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	WPS							
INITIAL_VALUE	1	1	1	1	1	1	1	1

[bit31:0] PUR[x] : PPU User Mode Read Attribute Bit (x=31 to 0)

Read attribute for user mode access of the peripheral #(i*32+x).

PUR [x] x=31 to 0	Description
0	Read attribute in user mode is disabled.
1	Read attribute in user mode is enabled.

Note:

- *The access from the CPU to the corresponding peripheral, please go after writing to the PUR[x] bit is completed.*

4.10. PPU Privileged Write or Access Attribute Register (PPU0_PPWAi)

The register sets write attribute (in Read/Write mode) or access attribute (in Read/Access mode) of the corresponding peripherals in privileged mode access. To write this register, you have to unlock the PPU by writing the unlock value into PPU Unlock Register.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	PPWA[31]	PPWA[30]	PPWA[29]	PPWA[28]	PPWA[27]	PPWA[26]	PPWA[25]	PPWA[24]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	WPS							
INITIAL_VALUE	1	1	1	1	1	1	1	1

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	PPWA[23]	PPWA[22]	PPWA[21]	PPWA[20]	PPWA[19]	PPWA[18]	PPWA[17]	PPWA[16]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	WPS							
INITIAL_VALUE	1	1	1	1	1	1	1	1

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	PPWA[15]	PPWA[14]	PPWA[13]	PPWA[12]	PPWA[11]	PPWA[10]	PPWA[9]	PPWA[8]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	WPS							
INITIAL_VALUE	1	1	1	1	1	1	1	1

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	PPWA[7]	PPWA[6]	PPWA[5]	PPWA[4]	PPWA[3]	PPWA[2]	PPWA[1]	PPWA[0]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	WPS							
INITIAL_VALUE	1	1	1	1	1	1	1	1

[bit31:0] PPWA[x] : PPU Privileged Mode Write or Access Attribute Bit (x=31 to 0)

Write or access attribute for Privileged mode access of the peripheral #(i*32+x).

PPWA [x] x=31 to 0	Description
0	[Read/Write mode] Write attribute in privileged mode is disabled. [Read/Access mode] Access attribute in privileged mode is disabled.
1	[Read/Write mode] Write attribute in privileged mode is enabled. [Read/Access mode] Access attribute in privileged mode is enabled.

Note:

- *The access from the CPU to the corresponding peripheral, please go after writing to the PPWA[x] bit is completed.*

4.11. PPU User Write or Access Attribute Register (PPU0_PUWAi)

The register sets write attribute (in Read/Write mode) or access attribute (in Read/Access mode) of the corresponding peripherals in user mode access. To write this register, you have to unlock the PPU by writing the unlock value into PPU Unlock Register.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	PUWA[31]	PUWA[30]	PUWA[29]	PUWA[28]	PUWA[27]	PUWA[26]	PUWA[25]	PUWA[24]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	WPS							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	PUWA[23]	PUWA[22]	PUWA[21]	PUWA[20]	PUWA[19]	PUWA[18]	PUWA[17]	PUWA[16]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	WPS							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	PUWA[15]	PUWA[14]	PUWA[13]	PUWA[12]	PUWA[11]	PUWA[10]	PUWA[9]	PUWA[8]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	WPS							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	PUWA[7]	PUWA[6]	PUWA[5]	PUWA[4]	PUWA[3]	PUWA[2]	PUWA[1]	PUWA[0]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	WPS							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:0] PUWA[x] : PPU User Mode Write or Access Attribute Bit (x=31 to 0)

Write or access attribute for User mode access of the peripheral #(i*32+x).

PUWA[x] x=31 to 0	Description
0	[Read/Write mode] Write attribute in user mode is disabled. [Read/Access mode] Access attribute in user mode is disabled.
1	[Read/Write mode] Write attribute in user mode is enabled. [Read/Access mode] Access attribute in user mode is enabled.

Note:

- *The access from the CPU to the corresponding peripheral, please go after writing to the PUWA[x] bit is completed.*

4.12. PPU Lock Register (PPU0_LOCKi)

The registers lock the access to PPU register by writing the lock value (0x30004000). There are several Lock register instances in PPU but all of them work in the same manner.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	L[31]	L[30]	L[29]	L[28]	L[27]	L[26]	L[25]	L[24]
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE	WP							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	L[23]	L[22]	L[21]	L[20]	L[19]	L[18]	L[17]	L[16]
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE	WP							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	L[15]	L[14]	L[13]	L[12]	L[11]	L[10]	L[9]	L[8]
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE	WP							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	L[7]	L[6]	L[5]	L[4]	L[3]	L[2]	L[1]	L[0]
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE	WP							
INITIAL_VALUE	0	0	0	0	0	0	0	0

Note:

- *Writing by byte or half-word access to this register results a bus error response.*

[bit31:0] L : Lock

By writing 0x30004000, the PPU is locked and PPU0_SR:LST is set to "1". Writing any other values has no effect.

4.13. PPU Privilege Mode Forced Change Function Enable Register (PPU0_PFENi)

The register controls privilege mode forced change operation of the corresponding peripherals. When this mode is enabled (PPU0_PFENi= 1), privilege level to the corresponding peripherals forcibly changes to privilege mode. To write this register, you have to unlock the PPU by writing the unlock value into PPU Unlock Register.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	PFEN[31]	PFEN[30]	PFEN[29]	PFEN[28]	PFEN[27]	PFEN[26]	PFEN[25]	PFEN[24]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	WPS							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	PFEN[23]	PFEN[22]	PFEN[21]	PFEN[20]	PFEN[19]	PFEN[18]	PFEN[17]	PFEN[16]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	WPS							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	PFEN[15]	PFEN[14]	PFEN[13]	PFEN[12]	PFEN[11]	PFEN[10]	PFEN[9]	PFEN[8]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	WPS							
INITIAL_VALUE	0	0	0	0	0	0	0	0

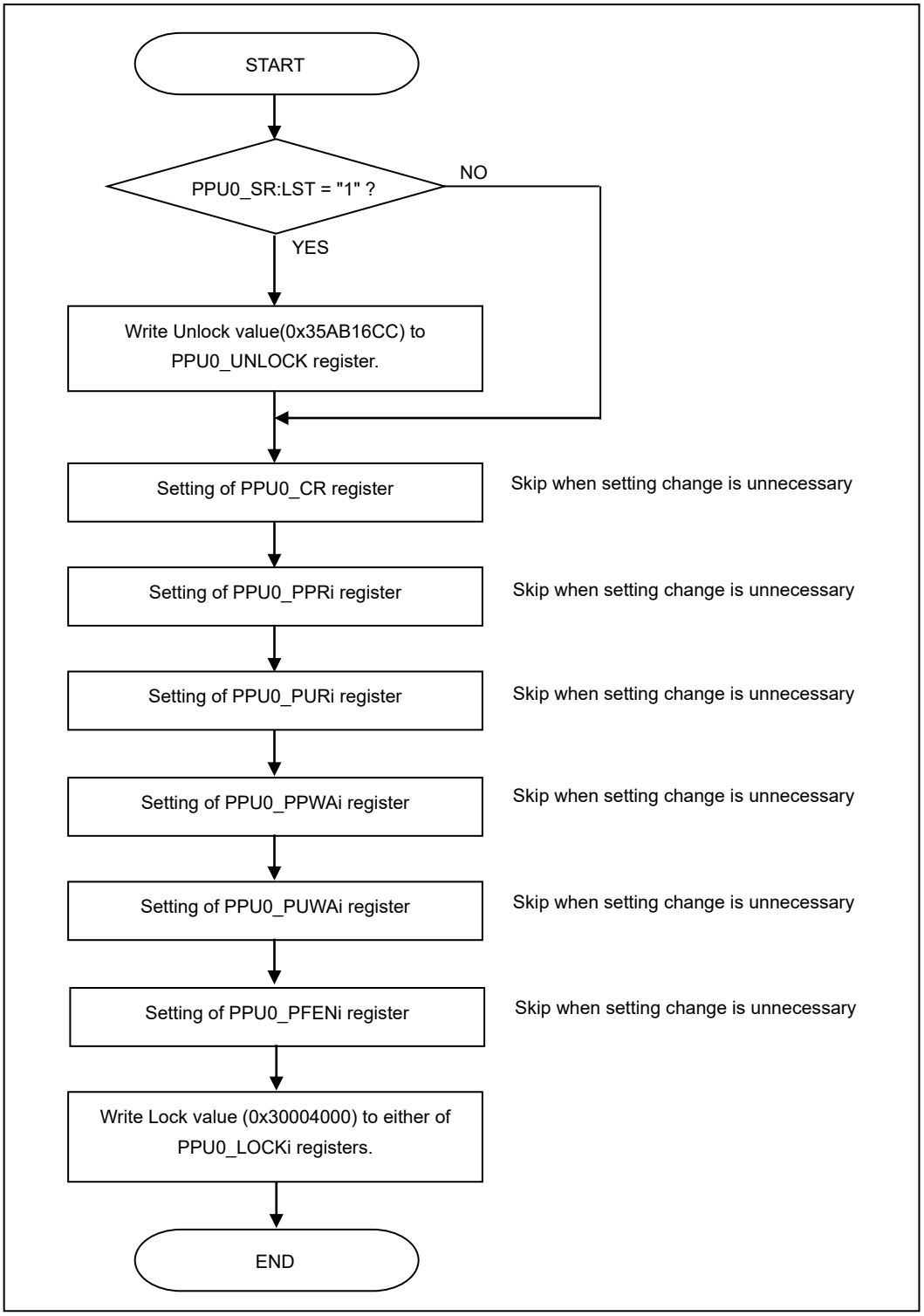
BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	PFEN[7]	PFEN[6]	PFEN[5]	PFEN[4]	PFEN[3]	PFEN[2]	PFEN[1]	PFEN[0]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	WPS							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:0] PFEN[x] : PPU Privilege Mode Forced Change Function Enable Bit (x=31 to 0)
Control of privilege mode Forced change operation of the peripheral #(i*32+x).

5. Setup Steps of PPU Register

Setup steps of the PPU register are shown below.

Figure 5-1 Setup Steps of PPU Register



Note:

- *If the setting of PPU registers is made by DMAC, before starting the access to the peripheral, CPU needs to confirm the lock value to PPU0_LOCKi register was written by DMAC (i.e. PPU0_SR:LST= 1).*

CHAPTER 52: DDR High Speed SPI Controller

This chapter explains the functionality and operation of the DDR High Speed SPI Controller (DDRHSSPI).

1. Overview
2. Configuration
3. Serial Interface
4. Operations of the DDRHSSPI
5. Registers

DDRHSSPI-TXXPT03P01R01L05-E1-XX

1. Overview

The DDRHSSPI provides various operating modes for interfacing to serial peripheral devices that use the de-facto standard SPI protocol. It supports also the Quad and Octal transfer modes to access multi-bit Serial Flash Memories. The DDRHSSPI's features are described in this section.

Features of the DDR High Speed SPI Controller

- In case of DDRHSSPIn_MID.MID=0x00000001
  - Supports Legacy Mode and Quad Mode
  - Allows the use of up to 4 Serial Flash Memories
- In case of DDRHSSPIn_MID.MID = 0x00000100 or 0x00000300
  - Supports Legacy Mode, Quad Mode and Octal Mode (Dual Quad, Dual Legacy)
  - Allows the use of up to 8 Serial Flash Memories
- Features a programmable transfer rate of the serial clock
- External Serial Flash Memories can be memory-mapped to the address-space of the MCU, in Command Sequencer Mode
- In Command Sequencer Mode, memory accesses initiated by the MCU and other masters are automatically converted to Serial Flash Memory read commands by the DDRHSSPI.
- In Command Sequencer Mode, access to the SPI Flash Memory is done through the Prefetch Buffer (PB)
- Direct Mode allows the DDRHSSPI to be used as a standard SPI master through a FIFO interface

Abbreviations

This section lists the terms and abbreviations used in this chapter.

Table 1-1 Terms and Abbreviations

Term	Meaning
Byte Time	Byte Time is the time required for transmission of 8 bits of data, over the SPI interface In SDR Mode, one byte time is: 1 cycle of SCLK in Octal Mode, 2 cycles of SCLK in Quad Mode and 8 cycles of SCLK in Legacy Mode. In DDR Mode one byte time is: 1/2 cycle of SCLK in Octal Mode, 1 cycle of SCLK in Quad Mode and 4 cycles of SCLK in Legacy Mode.
CSR	Control and Status Registers of DDRHSSPI.
DAP	Debug Access Port
DDR	Double Data Rate
DLP	Data Learning Pattern
DMA	Direct Memory Access
FIFO	First-In, First-Out
Half Word	16-bits of data
MCU	Micro Control Unit
on-the-fly	Switching modes of serial interface, during transferring data there
PB	Prefetch Buffer
RX	Receive
SDR	Standard Data Rate
SPI	Serial Peripheral Interface
SCLK	Serial Clock on the external port of the chip
SSEL	Serial Slave Select on the external port of the chip
SDATA	Serial Data on the external port of the chip
SW	Software
TX	Transmit
Word	32-bits of data

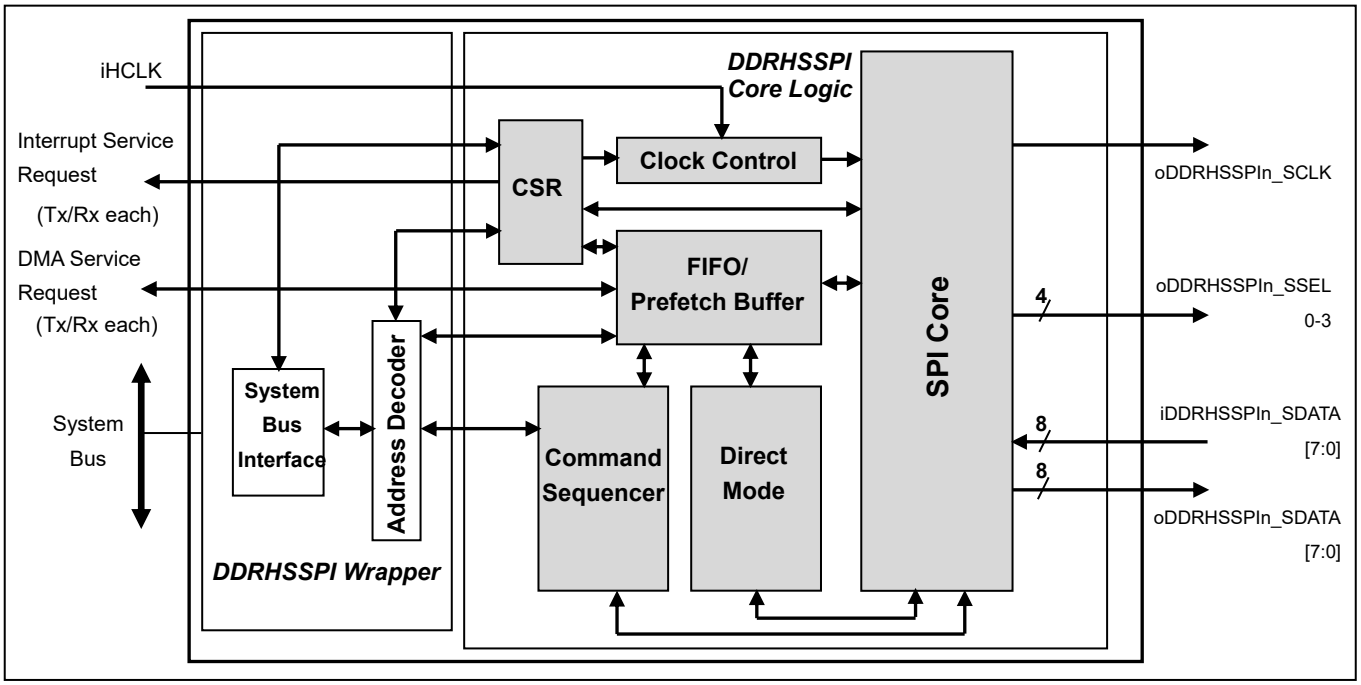
2. Configuration

This section describes a block diagram of DDRHSSPI.

Block Diagram

Figure 2-1 shows the block diagram of the DDRHSSPI, followed by the explanations of internal blocks.

Figure 2-1 DDRHSSPI Block Diagram



CSR

The CSR block holds the configuration and status registers, which are used to control and monitor the status of the DDRHSSPI.

Please refer to section 4.3 Address Map of DDRHSSPI for more detailed information about the CSR.

SPI Core and Clock Control

The SPI Core block contains the SPI protocol engine, which handles the read and write operations from/to the external serial Flash Memory. The SPI communication-related attributes like Serial Clock Frequency and Serial Clock Mode, etc., are configured via the CSR. Depending on whether the DDRHSSPI is operating in Direct Mode or in Command Sequencer Mode, the SPI Core connects with the FIFOs. The DDRHSSPI can initiate serial transfers with Serial Flash Memories, which are connected to the four Slave Select lines: oDDRHSSPIn_SSEL0-3. An internal clock divider is used to derive the serial clock output (oDDRHSSPIn_SCLK).

System Bus and Address Decoding

These two blocks are used as the interface to the System Bus. The masters such as MCUs are able to access the DDRHSSPI through the System Bus Interface and Address Decoder.

The Address Decoder performs the decoding of the address bus. If the System Bus access is targeted to a command or a status register, the request is sent to the CSR block. If the access is for the Command Sequencer or the Direct Mode blocks, then also access relevant info is routed to Command Sequencer or Direct Mode. When in Command Sequencer Mode, if the System Bus access is for a Serial Flash Memory which is mapped onto one of the four Slave Select lines, then the memory address is passed to the Command Sequencer block.

FIFO / Prefetch Buffer

This block can be used in two different modes based on DDRHSSPI configured mode, Command Sequencer Mode or Direct Mode. In Direct Mode, this block is used as two separate FIFOs: One for TX and one for RX. Each FIFO is 24-locations deep and has a data-width of 32 bits.

In Command Sequencer Mode this block is used as Prefetch Buffer for only receiving data from Flash Memory. The Prefetch Buffer is 48-locations deep and has data width of 32 bits.

Direct Mode

In Direct Mode DDRHSSPI internally uses FIFO/Prefetch buffer block configured as two FIFOs for temporary storage: One holds the data to be transmitted (TX-FIFO) and one stores the data to be received (RX-FIFO).

TX-FIFO and RX-FIFO are used by DDRHSSPI only in Direct Mode.

The Direct Mode block has following functions:

- Transmits data onto the Serial Interface via TX-FIFO.
- Receives data from the Serial Interface via RX-FIFO.
- DMA is available both for transmission and reception.
- Controls the data format to transmit.
  - Tri-state with variable length
  - SPI protocol on each data
  - Data Rate Mode (SDR Mode or DDR Mode) on each data

Command Sequencer

The Command Sequencer maps the external Serial Flash Memories on the address space of MCU.

In this mode the FIFO/Prefetch Buffer block is configured as Prefetch Buffer (PB) available for temporary storage and prefetch accesses, and only read operations of the Flash Memory are possible. The read data are output to the System Bus via internal Prefetch Buffer.

The Command Sequencer block has following functions:

- Watches continuity of memory access on the System Bus and determines whether it should keep current serial transfer or start a new one.
- Generates a serial transaction with
  - Decoding a Command Sequence table in CSR to output a series of data to the SPI Core.
  - Generating a serial address: it concatenates address bits from System Bus and a register value in CSR.
- Controls Prefetch Buffer to flush.
- Runs an Idle Timer (ITIMER) to control whether to close current serial transaction.

3. Serial Interface

This section describes the Serial Interface of DDRHSSPI.

3.1. Serial Clock Modes and Data Rate Modes

DDRHSSPI supports two Data Rate Modes for Serial Flash Interface, namely Single Data Rate (SDR) and Double Data Rate (DDR). In Direct Mode, the DDR is allowed only at TX-Only Mode.

Maximum output serial clock frequency in both SDR and DDR Modes, is up to 1/2 peripheral bus frequency.

Based on the programmed values of the DDRHSSPIn_PCC0-3.ACES bits (valid only in SDR Mode), each peripheral can have up to 2 clock modes in SDR Mode and only 1 clock mode in DDR Mode. DDRHSSPIn_PCC0-3.ACES bit setting in DDR Mode is not allowed. These bits decide the serial data input and output timings of DDRHSSPI, with respect to the serial SPI clock. This is explained in Table 3-1 and Table 3-2.

Table 3-1 Clock Modes in SDR Mode

MODE	ACES (Active Clock Edges are Same on Peripheral)	Description
Mode 0	0	Output data (oDDRHSSPIn_SDATA) from DDRHSSPI are driven one half-cycle before the first positive edge of serial clock (oDDRHSSPIn_SCLK) and on the subsequent **negative** edges of oDDRHSSPIn_SCLK.
		Input data (iDDRHSSPIn_SDATA) on the side of DDRHSSPI are sampled on the **positive** edges of oDDRHSSPIn_SCLK.
Mode 4	1	Output data (oDDRHSSPIn_SDATA) from DDRHSSPI are driven one half-cycle before the first positive edge of serial clock (oDDRHSSPIn_SCLK) and on the subsequent **negative** edges of oDDRHSSPIn_SCLK.
		Input data (iDDRHSSPIn_SDATA) on the side of DDRHSSPI are sampled on the **negative** edges of oDDRHSSPIn_SCLK.

Figure 3-1 SDR Clock Mode 0

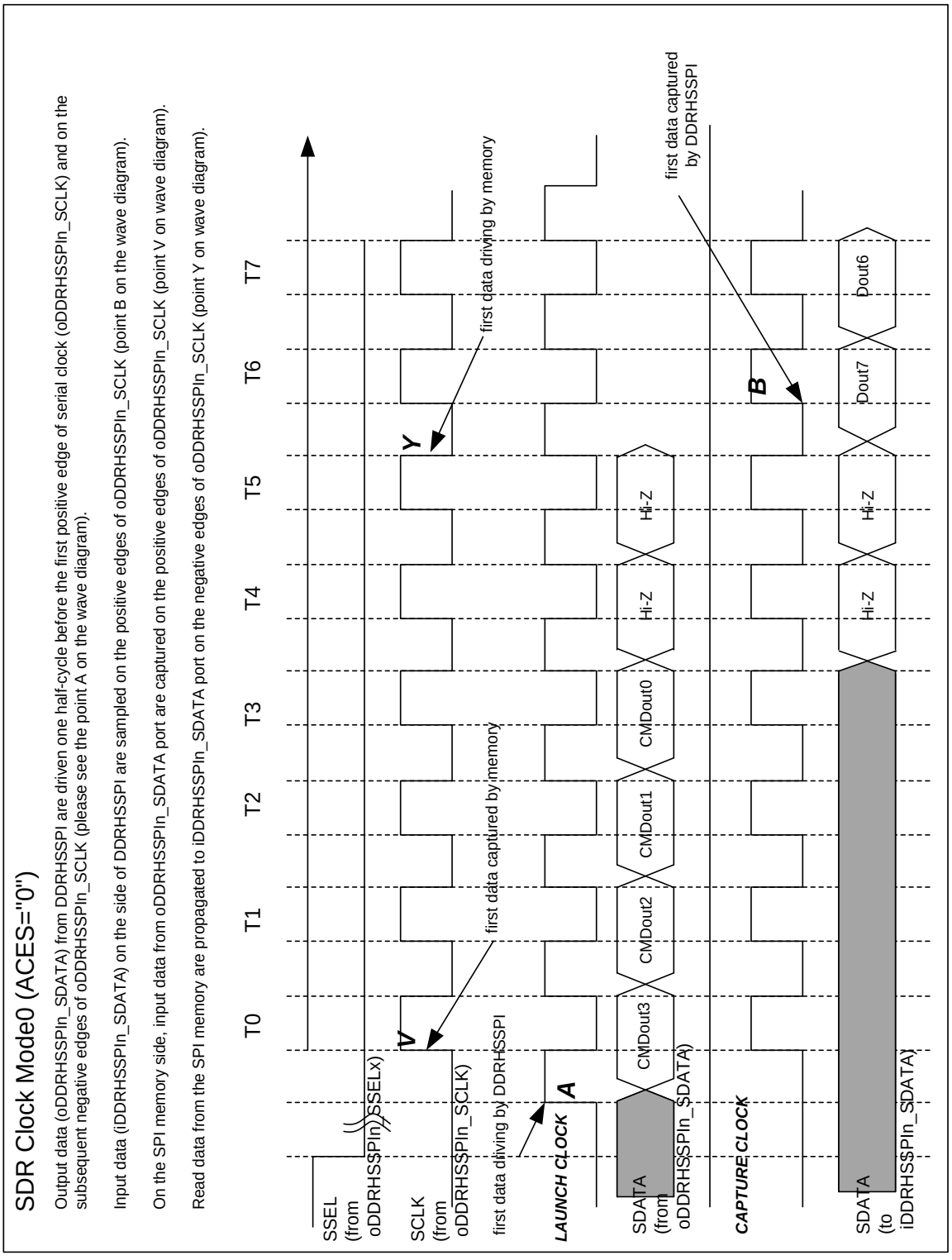


Figure 3-2 SDR Clock Mode 4

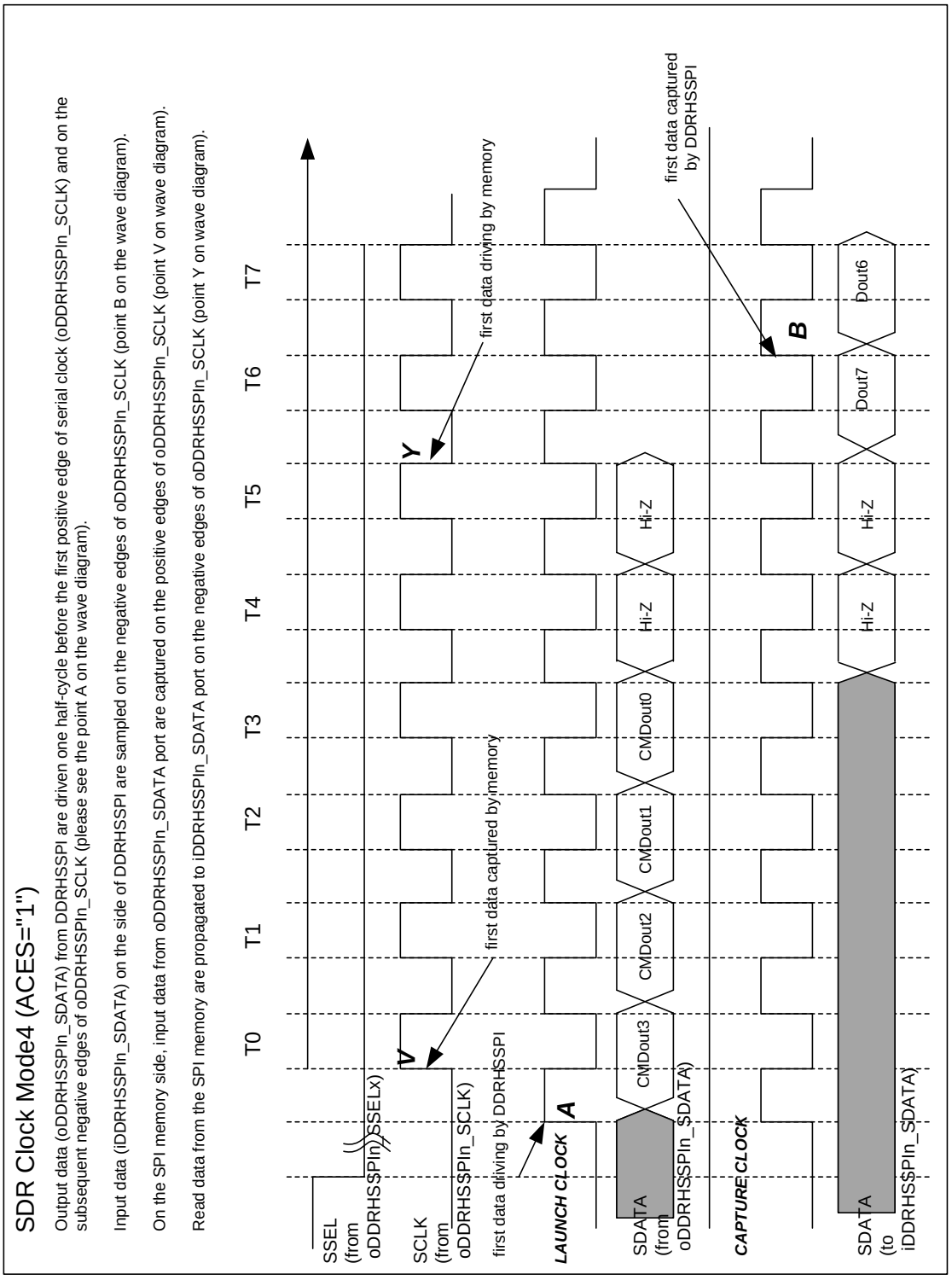


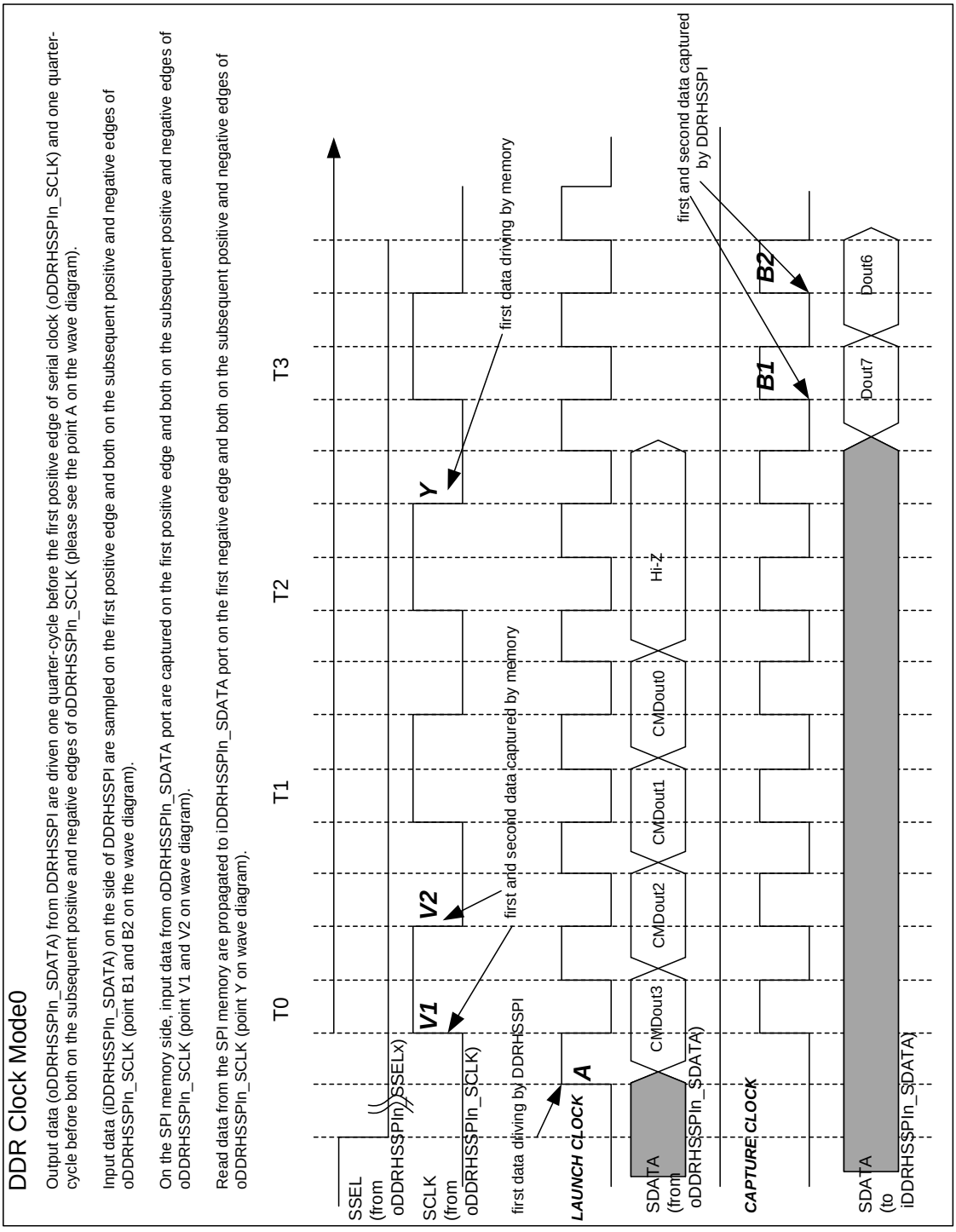
Table 3-2 Clock Modes in DDR Mode

MODE	Description
Mode 0	Output data (oDDRHSSPIn_SDATA) from DDRHSSPI are driven one quarter-cycle before the **first positive edge** of serial clock (oDDRHSSPIn_SCLK) and one quarter-cycle before both on the **subsequent positive and negative edges** of oDDRHSSPIn_SCLK.
	Input data (iDDRHSSPIn_SDATA) on the side of DDRHSSPI are sampled on the **first positive edge,** and both on the **subsequent positive and negative edges** of oDDRHSSPIn_SCLK.

Note:

- *In Direct Mode, the DDR is allowed only at TX-Only Mode.*

Figure 3-3 DDR Clock Mode 0



3.2. SPI Clock Frequency

oDDRHSSPIn_SCLK clock is internally generated by dividing the System Bus Clock (iHCLK). The clock division ratio of the resulting internal clock-divider can be programmed in the DDRHSSPIn_PCC0-3 Registers. The DDRHSSPIn_PCC0-3.CDRS (4-bit wide) decides the clock division ratio for output serial clock frequency. Maximum clock division rate supported is 32.

For CDRS value 0 output serial clock frequency SCLK is 1/2 of iHCLK

For CDRS value 1 output serial clock frequency SCLK is 1/4 of iHCLK.

.....

For CDRS value 15 output serial clock frequency SCLK is 1/32 of iHCLK.

Output clock frequency set by CDRS is independent of Data Rate Mode (SDR Mode or DDR Mode) settings.

3.3. SPI Input Data Sampling Point

This feature is available only when DDRHSSPIn_MID.MID is 0x00000100 or 0x00000300.

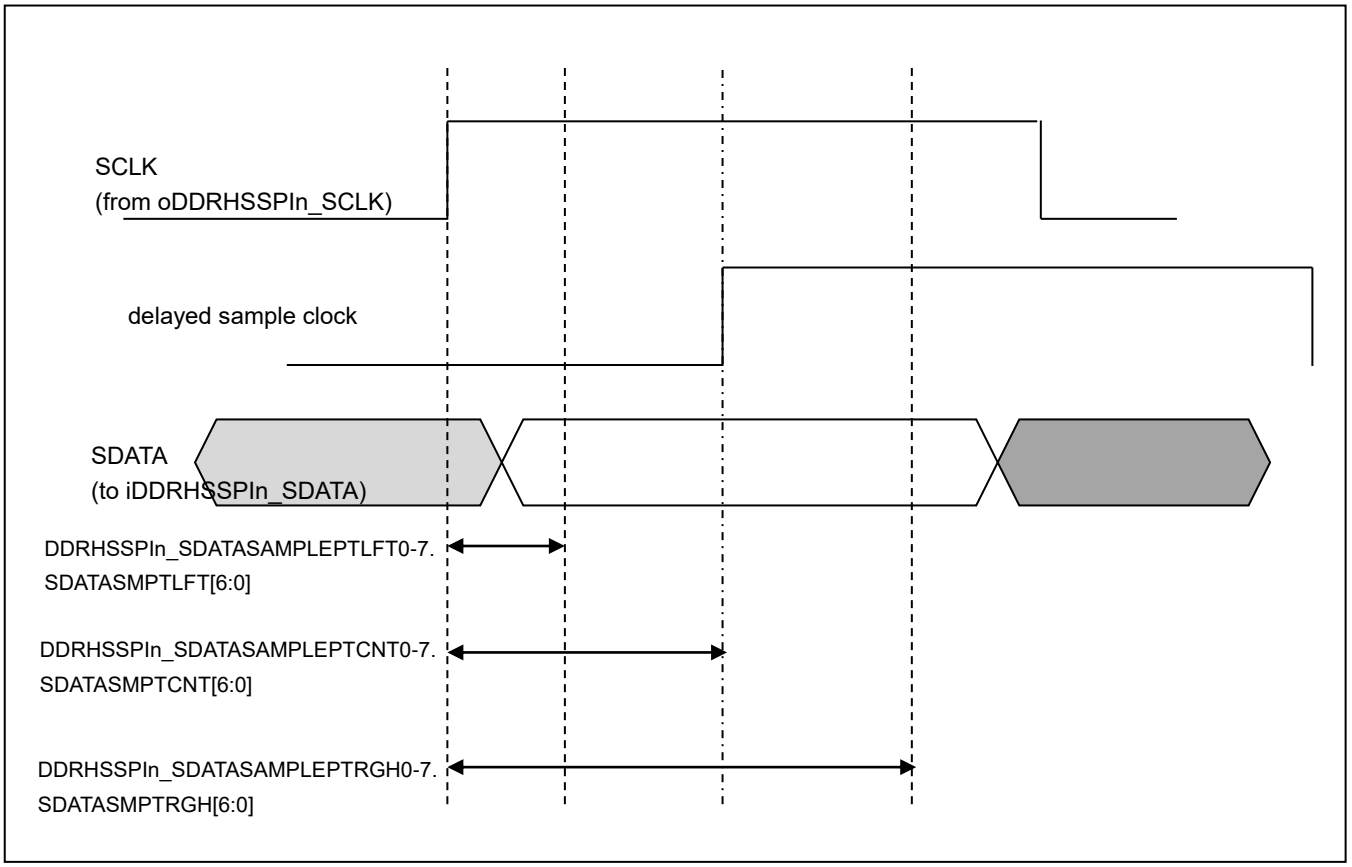
The delay of internal clock which samples iDDRHSSPIn_SDATA is adjustable, comparing to oDDRHSSPIn_SCLK clock. By this feature, the sampling timing of input data from Flash Memory can be finely tuned. The SPI clock sampling point per each bit line of iDDRHSSPIn_SDATA port can be programmed in the DDRHSSPIn_SDATASAMPLEPTCNT0-7 Registers.

Sampling clock fine tuning is implemented for each iDDRHSSPIn_SDATA[7:0] input line with buffers forming delay chain (on the internal clock) with programmable delay selection. It is up to SW to choose appropriate value of iDDRHSSPIn_SDATA clock sample delay point in order to provide consistency and compensate for input delay per data lines.

When in using DLP, it is possible to set additional two clock sampling points per each iDDRHSSPIn_SDATA input (so called left and right sampling point) for checking purpose only. These values are set in DDRHSSPIn_SDATASAMPLEPTLFT0-7 and DDRHSSPIn_SDATASAMPLEPTRGH0-7 Registers. Any of DLP errors at CNT/LFT/RGH will cause an error response on the System Bus. For more details, please refer to Notes of these registers before using DLP.

Please refer to Figure 3-4 explaining this in wave diagrams.

Figure 3-4 SDATA Sampling Clock Settings



Please note that setting delay is always done comparing to the rising edge of oDDRHSSPIn_SCLK clock and it is not possible to set this to be "negative" i.e. to be before oDDRHSSPIn_SCLK rising edge.

It is recommended to set values of corresponding registers such that SDATASMPTLFT ≤ SDATASMPTCNT≤ SDATASMPTRGH.

Note:

- *When in DDRHSSPIn_MID.MID is 0x00000001, all of SDATASMPTLFT, SDATASMPTCNT and SDATASMPTRGH must be fixed to 0.*

3.4. SPI Data Learning Pattern

This feature is available only when DDRHSSPIn_MID.MID is 0x00000100 or 0x00000300.

In case DDRHSSPI is connected with Flash Memory that has a feature outputting a Data Learning Pattern (DLP) during the dummy cycles, DDRHSSPI can check if the sampling point setting is correct.

This feature can be used only in Command Sequencer Mode and DDR Mode.

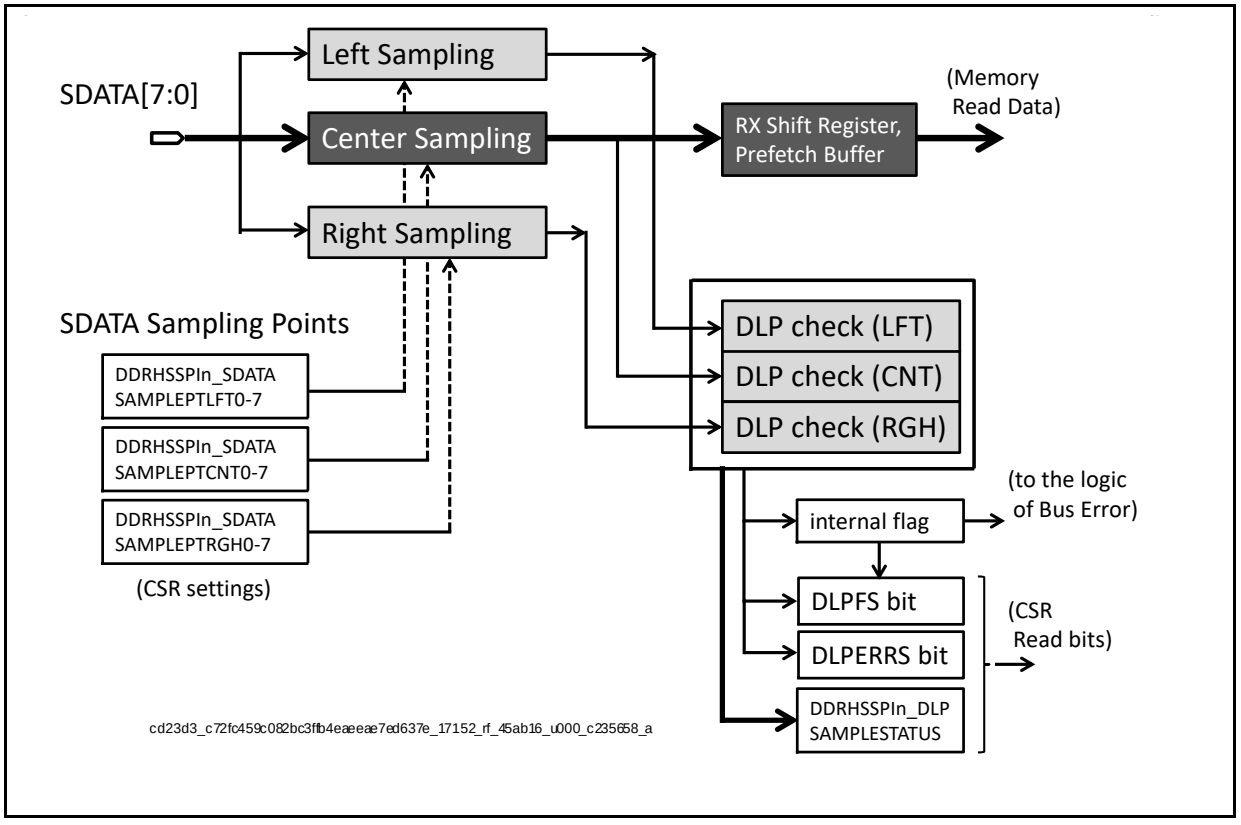
This feature is enable, when DDRHSSPIn_MCTRL.DLPEN bit is "1". This feature must be also enabled on the Flash Memory side. Please program the same value of DLP to both registers, in the Flash Memory and in DDRHSSPI (DDRHSSPIn_DLP Register). Once this feature is enabled, Flash Memory will output a Data Learning Pattern during dummy cycles, and DDRHSSPI will receive data on iDDRHSSPIn_SDATA at the same time.

Since DDRHSSPI has individual sampling points on each line of iDDRHSSPIn_SDATA, during the DLP pattern, DDRHSSPI can test neighboring (left-center-right) sampling points by registers DDRHSSPIn_SDATASAMPLEPTCNT0-7, DDRHSSPIn_SDATASAMPLEPTLFT0-7 and DDRHSSPIn_SDATASAMPLEPTRGH0-7.

All checks in iDDRHSSPIn_SDATA sampling clock settings (left-center-right) are comparing to the value of DDRHSSPIn_DLP Register as reference.

The logic related to SDATA Sampling Points and DLP is illustrated as below. The internal flag is directly invisible from the CPU, however, its status is reflected to the Bus Error and the system can detect it.

Figure 3-5 SDATA Sampling Points and Related Logics



Notes:

- *Once having detected a DLP error, the following Memory Read access is affected as below.*
  (1)When a DLP error occurs, it is held on the internal flag along DDRHSSPIn_FAULTF.DLPFS bit.
  (2)The software clears DLP-related flags (DDRHSSPIn_FAULTF.DLPFS and DDRHSSPIn_RXF.DLPERR) and adjusts the SDATA Sampling Points to proper values.
  (3)This internal flag affects the next Memory Read access depending on the address.
  (a)Same word-line addresses: "Error"
  If ("new address" >> 2) = ("previous address" >> 2), the result is as below.
  - Bus Error on the System Bus
  - Internal flag = "1"
  - DDRHSSPIn_FAULTF.DLPFS = "1"
  - DDRHSSPIn_RXF.DLPERR = "0"
  - DDRHSSPIn_DLPSAMPLESTATUS = 0x00000000
  (b)Different word-line addresses: "Success"
  If ("new address" >> 2) ≠ ("previous address" >> 2), the result is as below.
  - No Bus Error on the System Bus
  - Internal flag = "0"
  - DDRHSSPIn_FAULTF.DLPFS = "0"
  - DDRHSSPIn_RXF.DLPERR = "0"
  - DDRHSSPIn_DLPSAMPLESTATUS 0x00000000
  Figure 3-6 and Figure 3-7 illustrate these cases (a) and (b). Each diagram depicts two consecutive Memory Read accesses - the first one failing to sample SDATA, and the second one with successful sampling. The address of the second access determines the result.
- *The proposed Calibration sequence is described in the section 4.4.4 "Using the DDRHSSPI in Command Sequencer Mode of Operation".*

Figure 3-6 Memory Read accesses from "FAIL" to "PASS" ((a) Same Word-line)

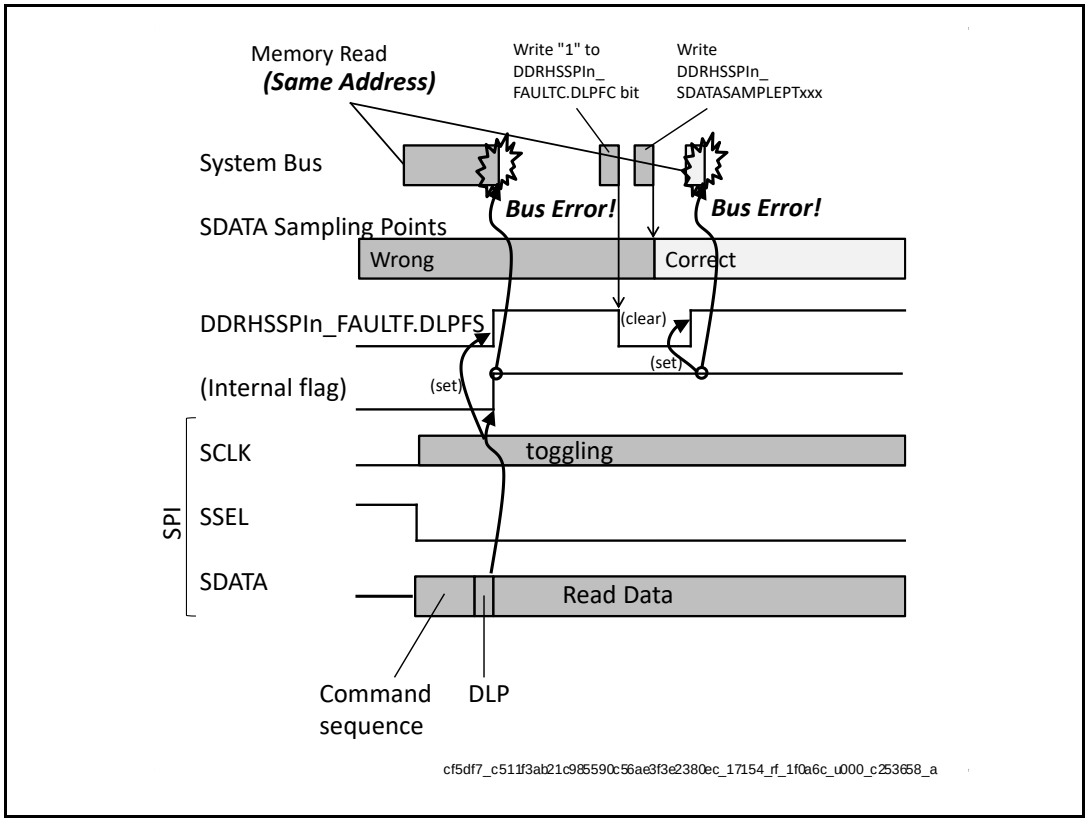
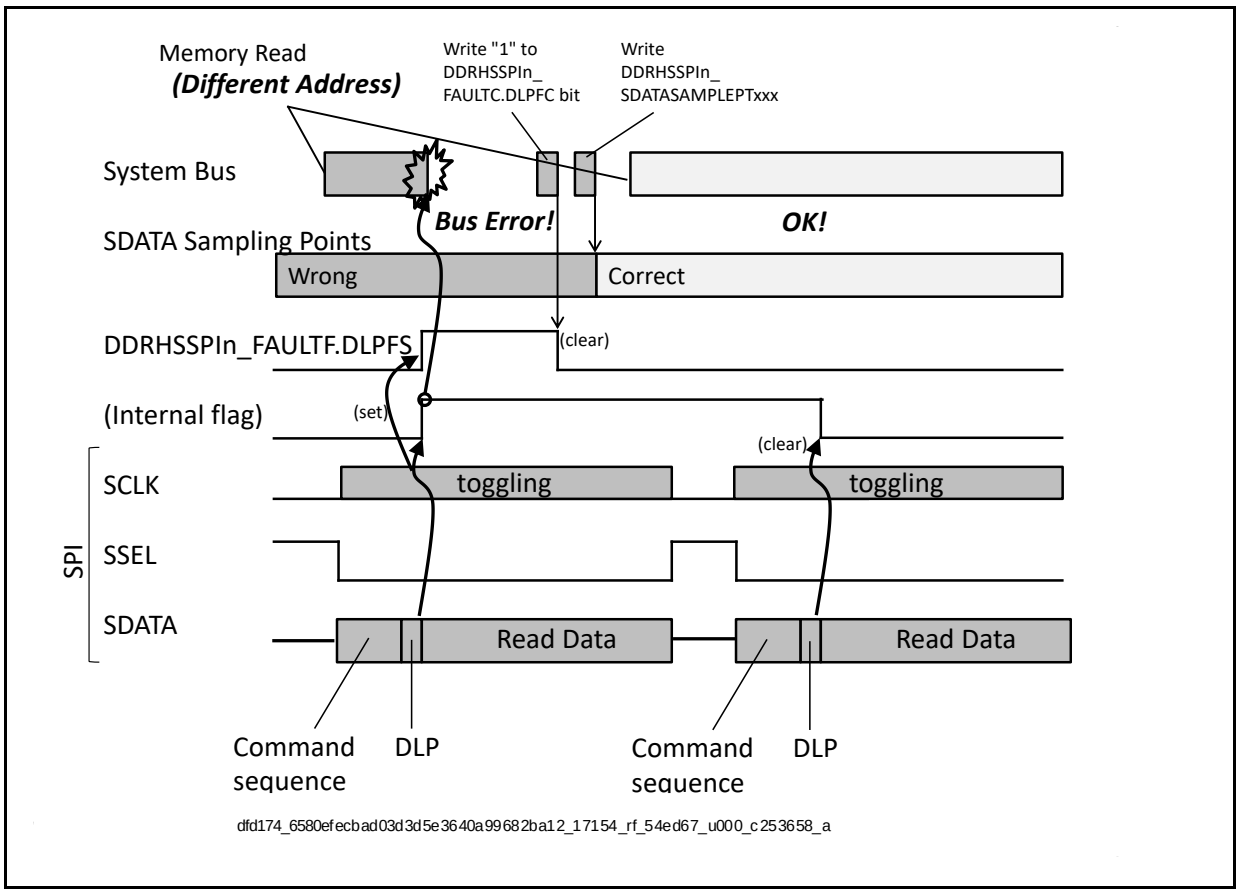


Figure 3-7 Memory Read accesses from "FAIL" to "PASS" ((b) Different Word-line)



3.5. SPI Data Protocol

DDRHSSPI supports Legacy (1-bit), Quad(4-bit) and Octal (8-bit) SPI protocols. The Quad and Octal Protocols are used for interfacing with Serial Flash Memories like Cypress FL-S series.

Legacy Protocol

The Legacy Protocol is a full-duplex protocol. When DDRHSSPI is configured with Legacy Protocol, the data can be received on a single input port (i.e. iDDRHSSPIn_SDATA[1]) and simultaneously, the data can also be transmitted on a single output port (i.e. oDDRHSSPIn_SDATA[0]). While Legacy Protocol is being used, the unused data lines (i.e. oDDRHSSPIn_SDATA[7:2]) are tri-stated by DDRHSSPI.

In Direct Mode, when DDRHSSPIn_DMTRP.TRP is configured for TX-and-RX Mode, it is allowed to use only the Legacy Mode.

In Command Sequencer Mode, the Legacy Protocol is not supported.

Quad Protocol

In Quad Protocol, four serial data lines (i.e. SDATA[3:0]) are used and SDATA[7:4] are tri-stated, in a half-duplex manner for accessing single device. Data transmission and reception cannot happen simultaneously.

In Direct Mode, when DDRHSSPIn_DMTRP.TRP is configured for "TX-Only in Quad Mode", the Quad Protocol is used.

In Command Sequencer Mode, the Quad Protocol is supported.

Octal Protocol

This feature is available only when DDRHSSPIn_MID.MID is 0x00000100 or 0x00000300.

In Octal Protocol, eight serial data lines (i.e. SDATA[7:0]) are used, in a half-duplex manner for accessing two Serial Flash Memories. Data transmission and reception cannot happen simultaneously.

In this mode DDRHSSPI is attached to two identical Flash Memories with 4 data bits each. Both of these memories also must be configured with same configuration parameters. In this mode, one memory will be attached to SDATA[3:0] and the other to SDATA[7:4]. These memories will be connected to the same SSEL signal.

The data transfer of Octal Protocol is classified as below.

- Dual Legacy

  This transfer type is available only in Direct Mode (TX-Only Mode and TX-and-RX Mode). The output ports oDDRHSSPIn_SDATA[0] and oDDRHSSPIn_SDATA[4] are used at Legacy Protocol each. And the input ports iDDRHSSPIn_SDATA[1] and iDDRHSSPIn_SDATA[5] are used at Legacy Protocol each, as well. Please refer to Table 4-1 for more details.

- Dual Quad

  This transfer type is available both in Direct Mode (TX-Only Mode) and Command Sequencer Mode. In Direct Mode, this transfer type is not available in TX-and-RX Mode.
  The output ports oDDRHSSPIn_SDATA[3:0] and oDDRHSSPIn_SDATA[7:4] are used at Quad Protocol each. And the input ports iDDRHSSPIn_SDATA[3:0] and iDDRHSSPIn_SDATA[7:4] are used at Quad Protocol each, as well. Please refer to Table 4-1 for more details.
  In Command Sequencer Mode, the value of address field is divided in half, since the data are located in each Serial Flash Memory at every 2 bytes.

3.6. Shift Direction

Most Significant Bit in the Shift Register is transmitted first and the first received data is shifted into the Least Significant Bit in the Shift Register. i.e. the Shift Register is shifted left. The read or write accesses to the data-registers always have Least Significant Bit of the data in bit 0.

The following figures Figure 4-7 depict the direction in which the data in the Shift Register is shifted to/from the serial data lines. The waveforms assume DDRHSSPIn_DMFIFOCFG.FWIDTH = 0b00 in Direct Mode. The Figures depict that the transmit data is loaded into the Shift Register from the TX-FIFO. However, the source of transmit data can be other registers, such as DDRHSSPIn_RDCSDC0-11.RDCSDATA.

When DDRHSSPIn_MID.MID is 0x00000100 or 0x00000300, DDRHSSPI is available for Dual Legacy Mode and Dual Quad Mode. Figure 3-14 and Figure 3-15 depict the direction in which the data in the Shift Register is shifted to/from the serial data lines, under Dual Legacy Mode and Dual Quad Mode.

In Command Sequencer Mode, byte ordering is always LSB first.

In Direct Mode, byte ordering is always MSB first.

Figure 3-8 Quad Mode / byte ordering - Command Sequencer Mode

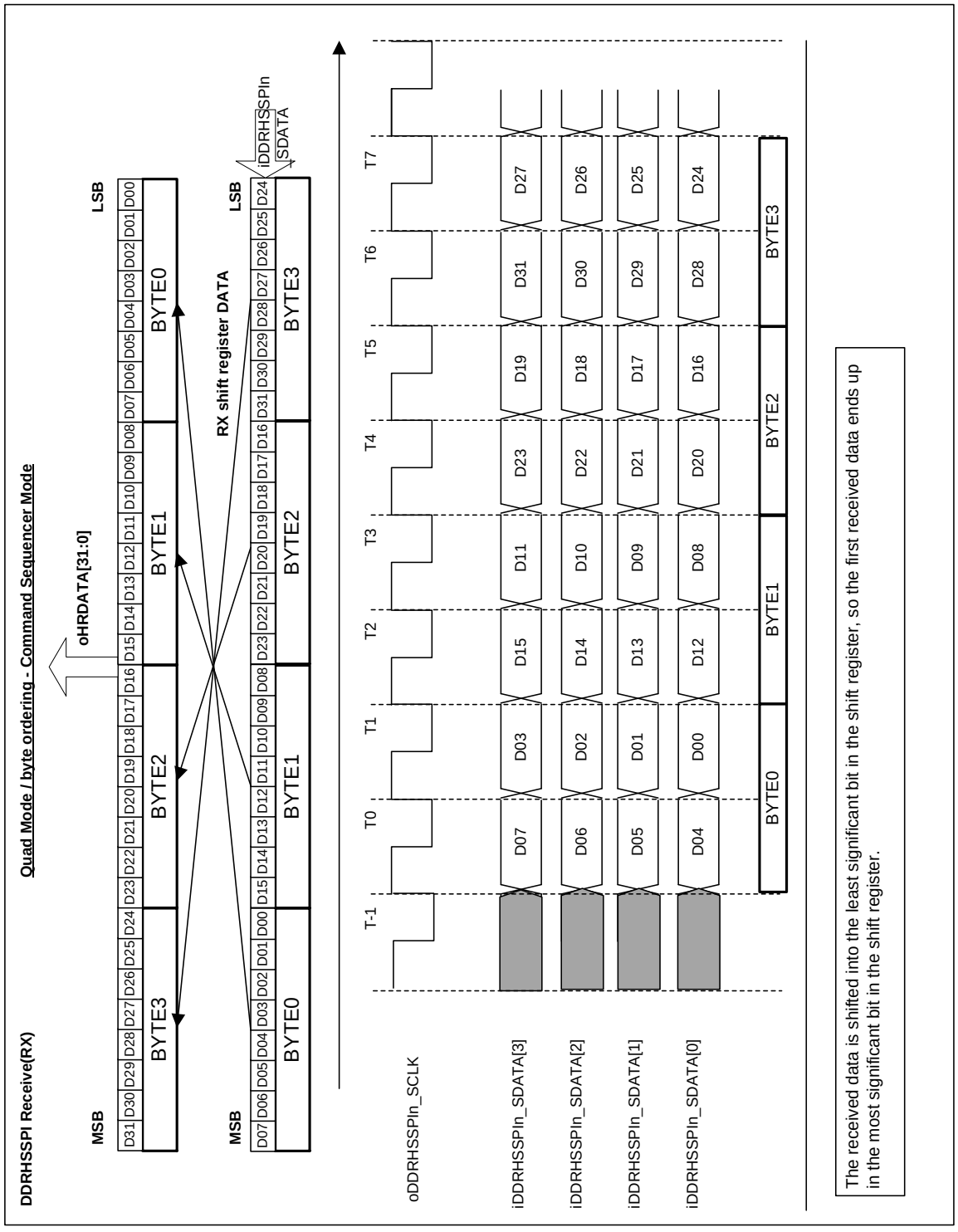


Figure 3-9 Quad Mode / byte ordering - Command Sequencer Mode (RX Register - In Time)

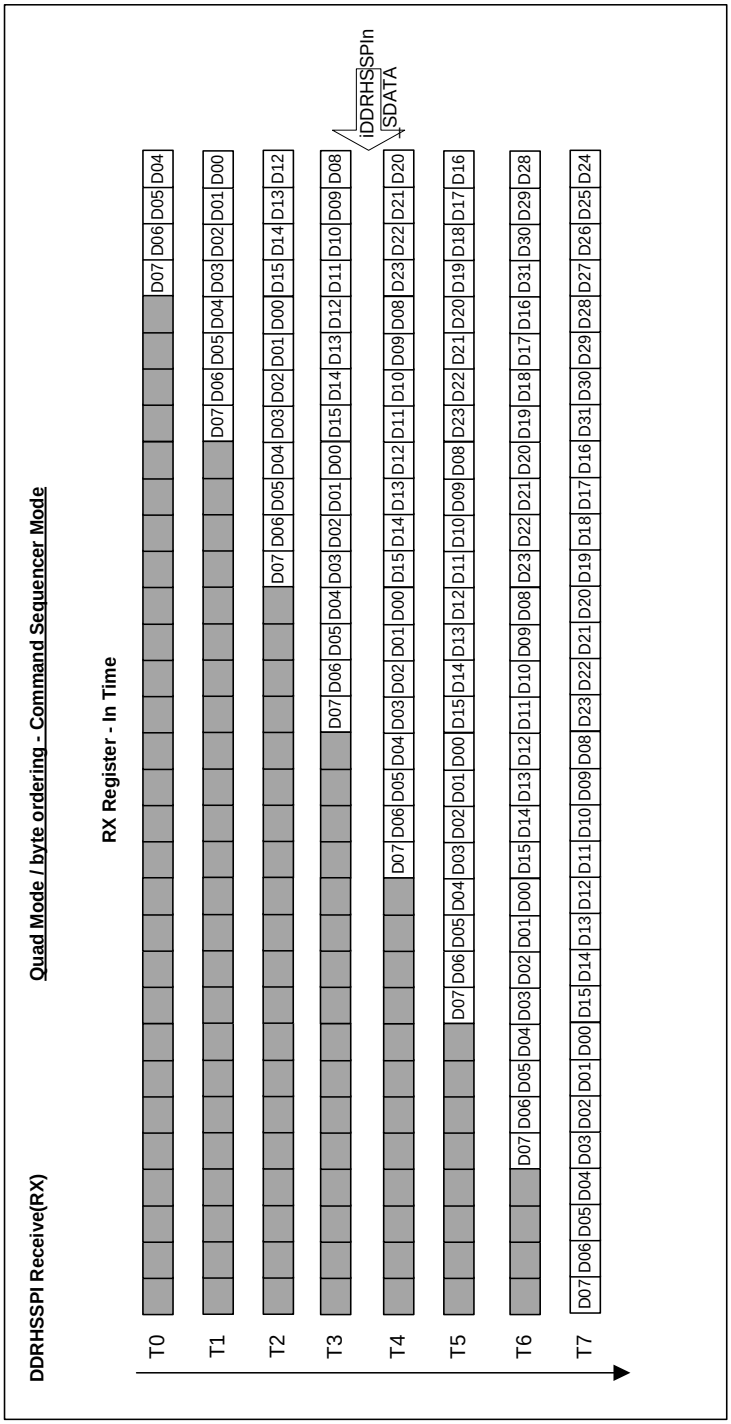


Figure 3-10 Quad Mode / byte ordering - Direct Mode

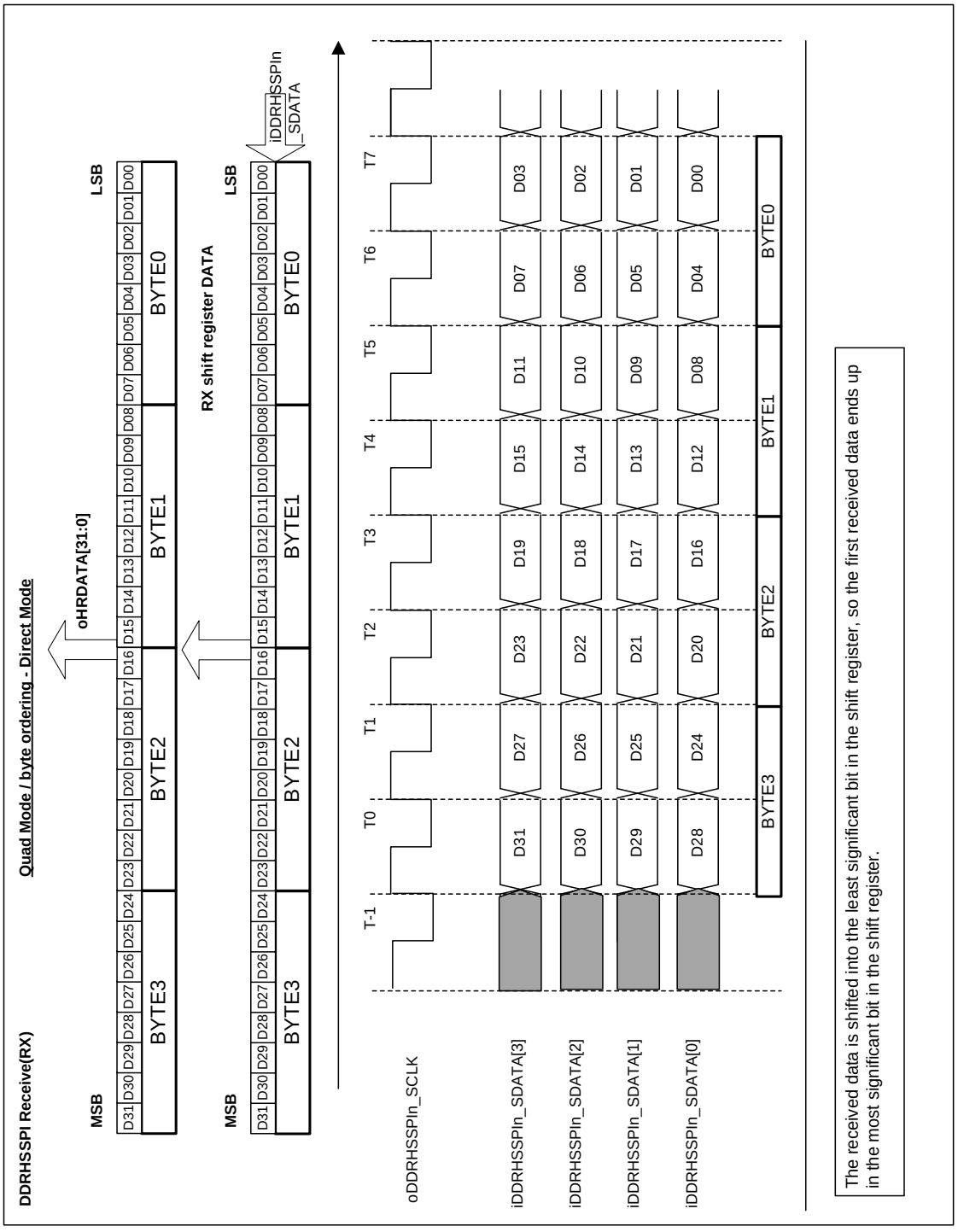


Figure 3-11 Quad Mode / byte ordering - Direct Mode (RX Register - In Time)

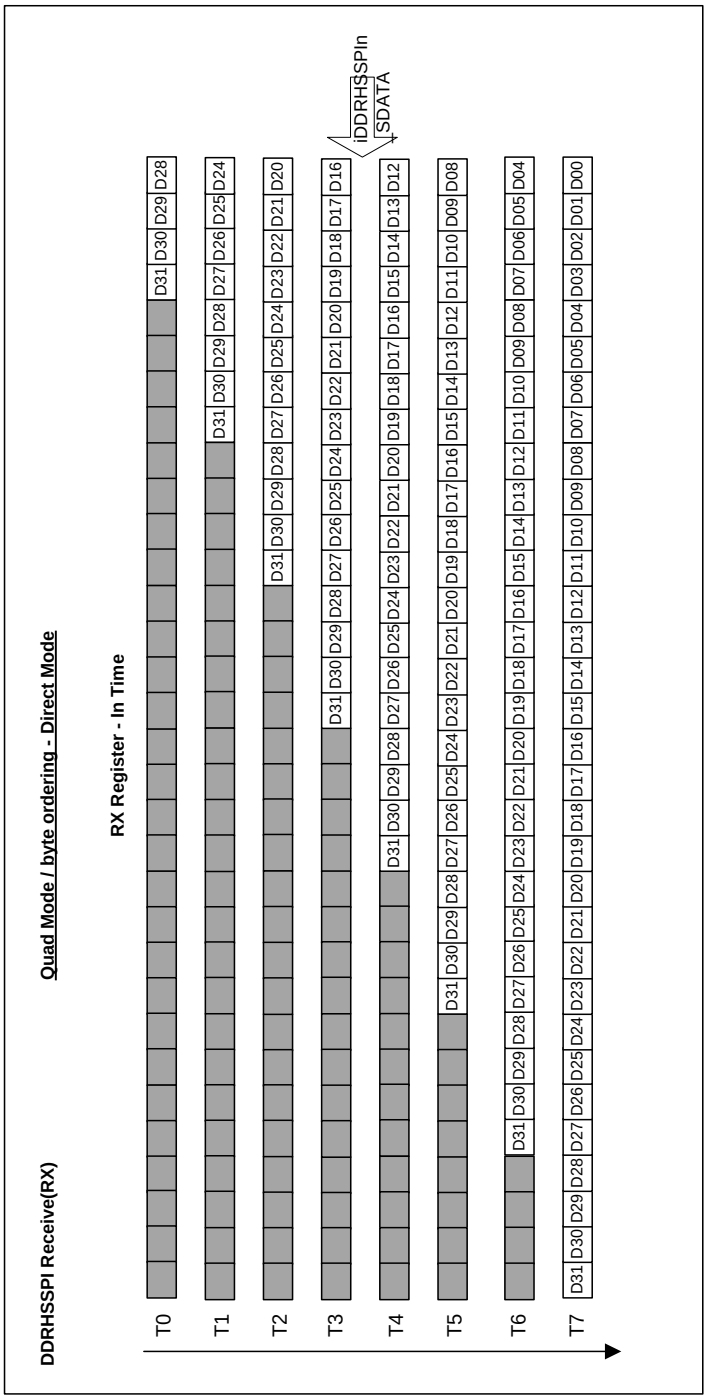


Figure 3-12 Shift Direction (Legacy Mode and Quad Mode in Direct Mode) (Assumptions: DDRHSSPIn_DMFIFOCFG.FWIDTH= 0b00)

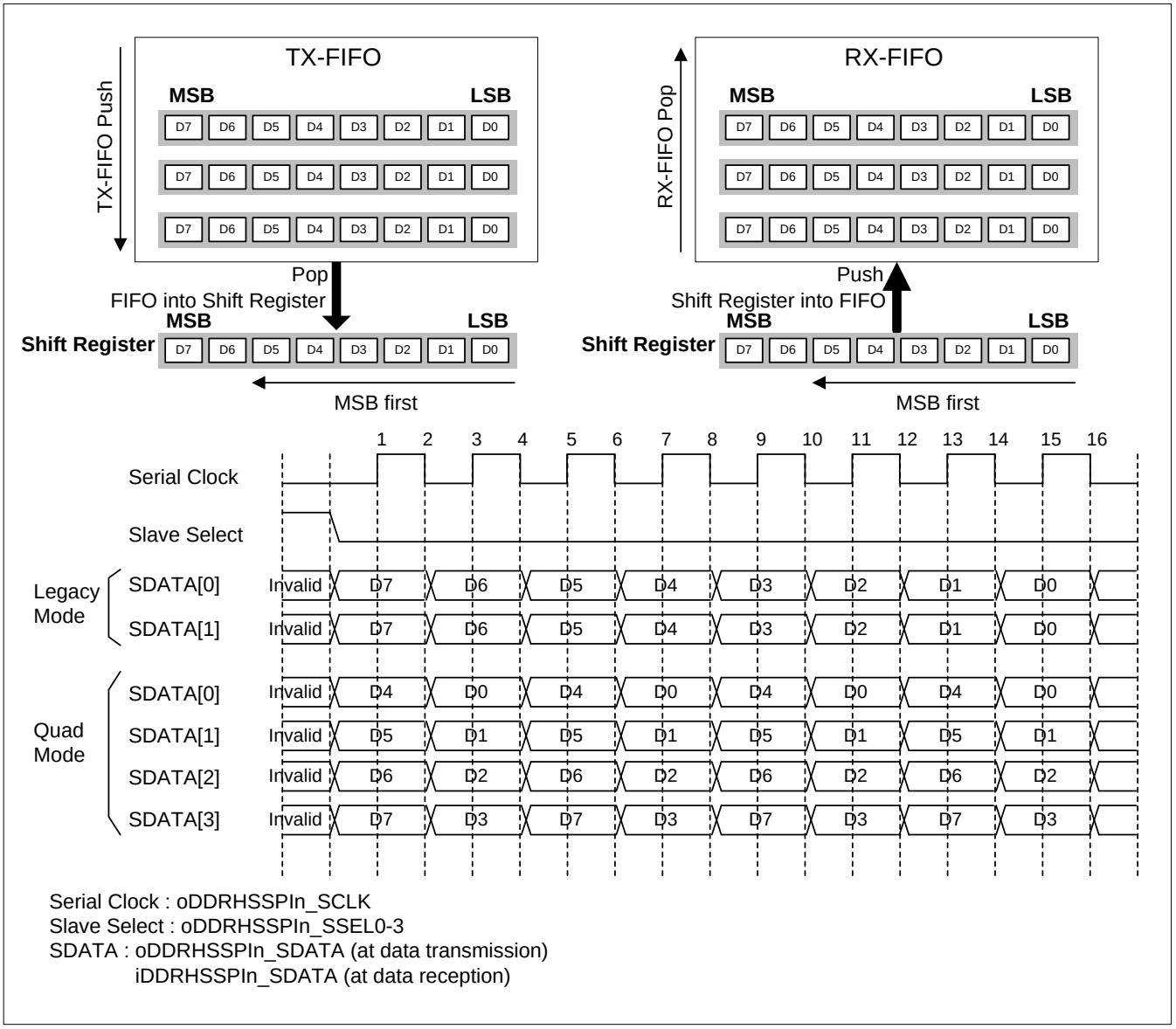
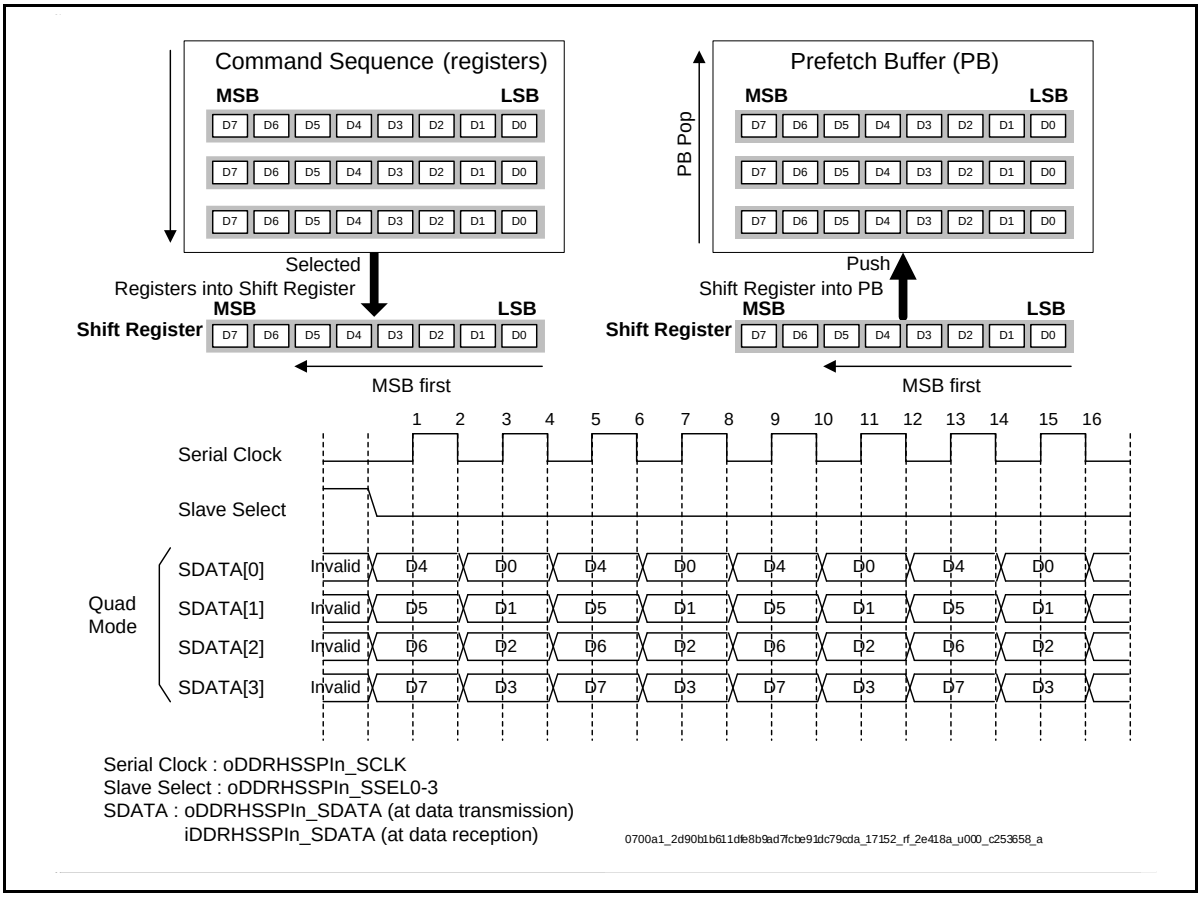
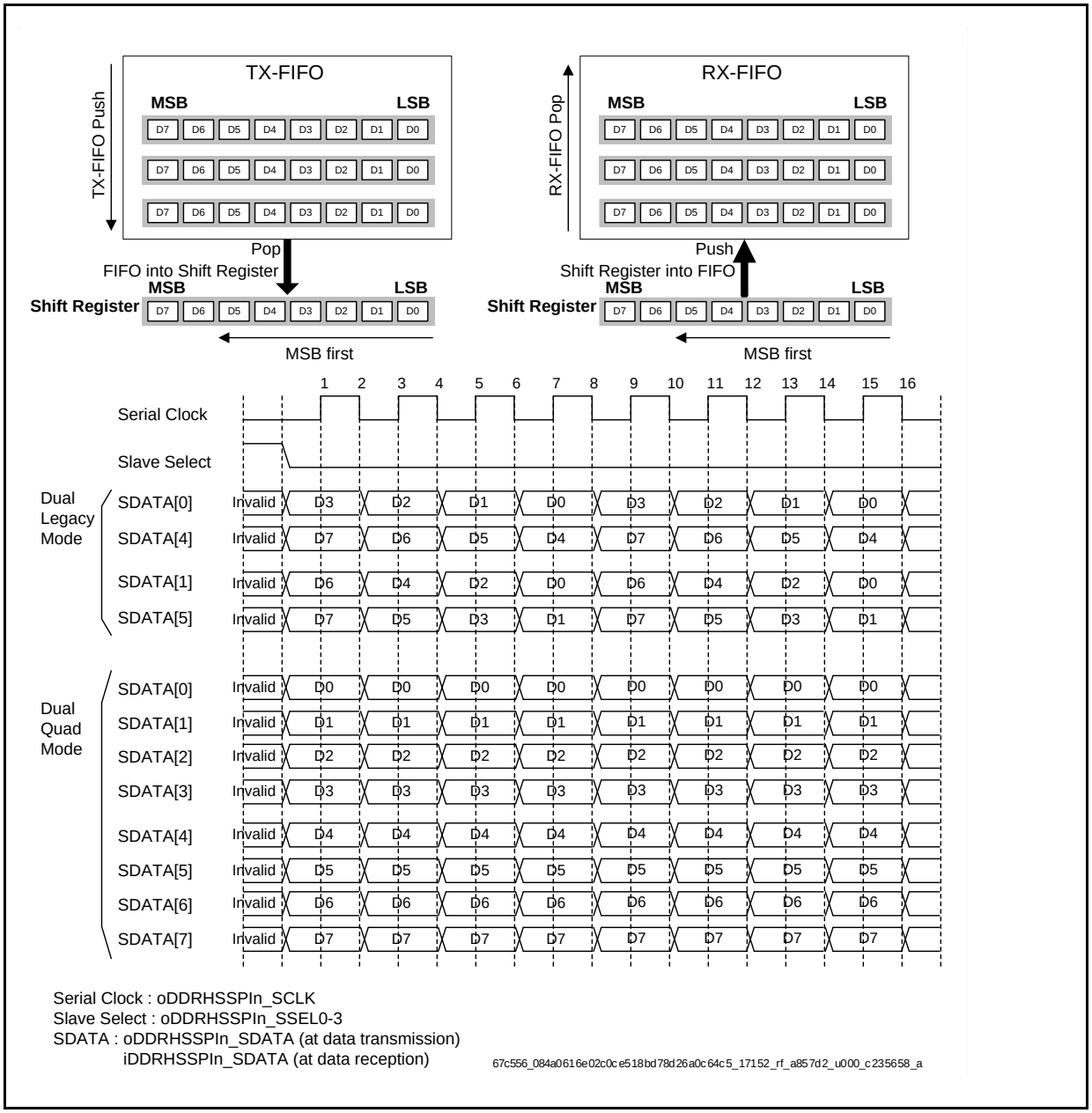


Figure 3-13 Shift Direction (Quad Mode in Command Sequencer Mode)



(Each line in the PB has 32-bit width, and this diagram is simplified by showing the lower 8 bits.)

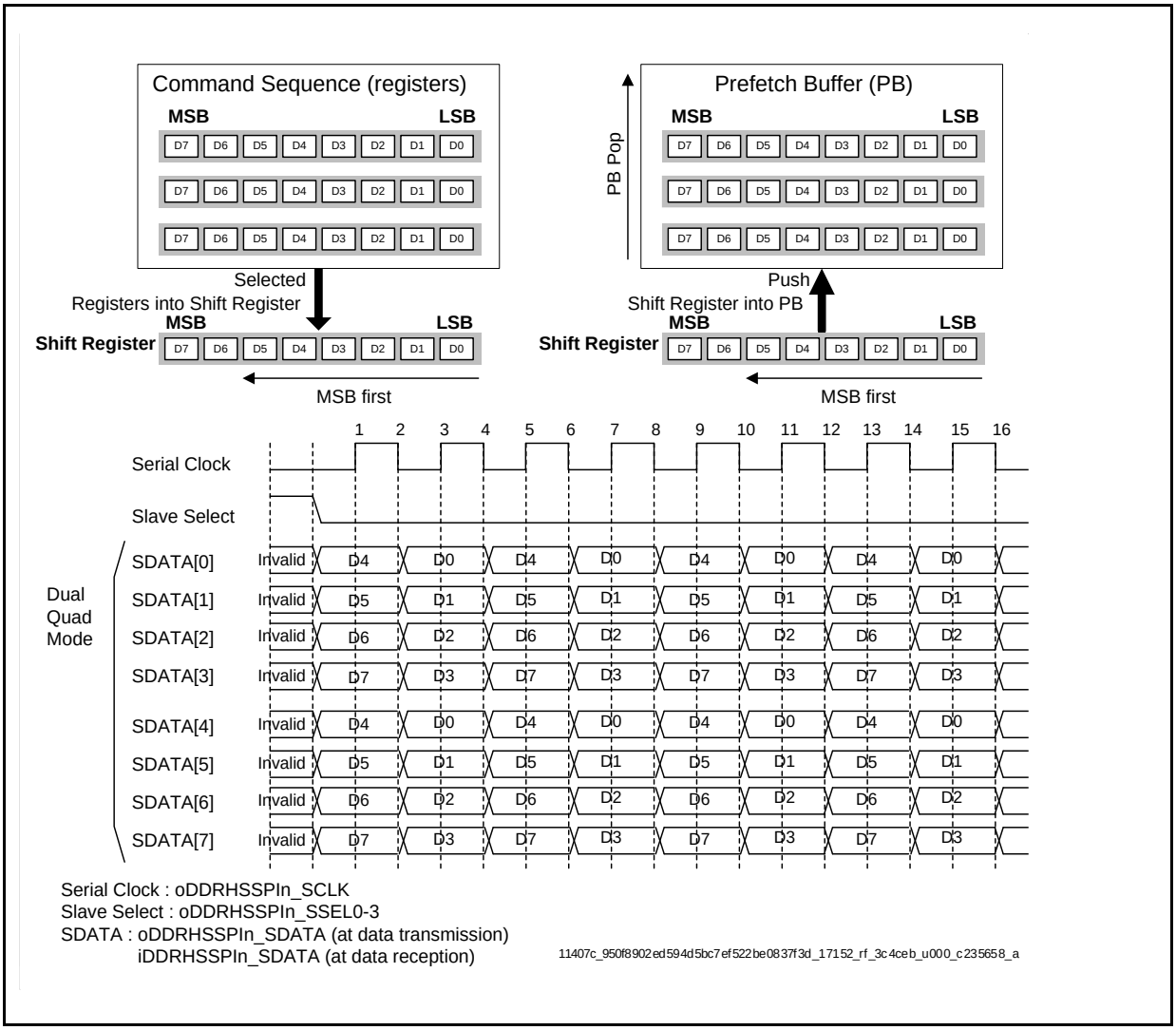
Figure 3-14 Shift Direction (Dual Legacy Mode and Dual Quad Mode in Direct Mode) (Assumptions: DDRHSSPIn_DMFIFOCFG.FWIDTH= 0b00)



Note:

- *The bit alignment is different between TX and RX in Dual Legacy Mode. (Please refer to Table 4-1, and its TXCTRL="1" & TXDATA[12]="1" condition -> SDR mode.)*

Figure 3-15 Shift Direction (Dual Quad Mode in Command Sequencer Mode)



(Each line in the PB has 32-bit width, and this diagram is simplified by showing the lower 8 bits.)

4. Operations of the DDRHSSPI

This section describes the operation of DDRHSSPI.

DDRHSSPI can be configured in one of the two operating modes: Direct Mode and Command Sequencer Mode.

In Direct Mode, the MCU can directly write data into the TX-FIFO to be transmitted to the Serial Flash Memory. Similarly, the MCU can directly read the data received from Serial Flash Memory, over the Serial Interface - from the RX-FIFO.
The SPI Core transfers the data to/from the FIFOs over the Serial Interface. The Direct Mode is described in section 4.1 Direct Mode. In Command Sequencer Mode, DDRHSSPI maps the external Serial Flash Memory onto the address space of the MCU.

In Command Sequencer Mode, only Flash Memory reads are supported. If the MCU (or any other master) initiates a System Bus transfer to access any of the mapped Serial Flash Memories, the DDRHSSPI initiates serial transfer for the corresponding memory read operation. Till the time DDRHSSPI accesses the external device, the System Bus transfer is stalled. The Command Sequencer Mode is described in section 4.2 Command Sequencer Mode.

If DDRHSSPIn_MID.MID is 0x00000100 or 0x00000300, DDRHSSPI can be connected with up to 8 Serial Flash Memories. At the connection with 8 Serial Flash Memories, each of the 4 Slave Select outputs is connected with 2 Serial Flash Memories.

4.1. Direct Mode

In Direct Mode, the MCU (or the DMA controller) is responsible to directly control the serial transfer on the Serial Interface. Direct Mode of transfer can be enabled using the DDRHSSPIn_MCTRL.CSEN bit.

Direct Mode supports the following modes.

- TX-Only Mode

  This is only to transmit data to serial interface, and supports Legacy Mode and Quad Mode. If DDRHSSPIn_MID.MID is 0x00000100 or 0x00000300, Dual Legacy Mode and Dual Quad Mode are added. This mode supports both SDR and DDR, and is used in the following cases.
  (1)Mode setting of Serial Flash Memory
  (2)Page programming of Serial Flash Memory
  (3)Initialing Memory Read access to Serial Flash Memory

- TX-and-RX Mode

  This is to receive data from serial interface with transmitting data (full-duplex), and supports Legacy Mode and Dual Legacy Mode. The transmission of one TX-FIFO level will involve receiving one RX-FIFO level. This mode does not support DDR, and is used in the following case.
  (1)Status reading from Serial Flash Memory

In Direct Mode, DDRHSSPI uses its internal TX-FIFO and RX-FIFO, for temporary storage of the data to be transmitted and the data received over the Serial Interface.

4.1.1. Internal FIFOs

In Direct Mode, DDRHSSPI internally has two FIFOs for temporary storage: One for the data to be transmitted (TX-FIFO) and one for the data to be received (RX-FIFO).

Based on whether the serial transfers in DDRHSSPI are configured as TX-Only Mode or TX-and-RX Mode in the DDRHSSPIn_DMTRP.TRP, only one or both FIFOs are used by DDRHSSPI. If DDRHSSPI

is configured for TX-Only Mode, the TX-FIFO is used. If DDRHSSPI is configured for TX-and-RX Mode, then both TX-FIFO and RX-FIFO are used.

4.1.2. FIFO Size

Each FIFO is 24-locations deep and has a data-width of 32 bits. However, the software can configure the valid data-width of the TX-FIFO and the RX-FIFO in DDRHSSPIn_DMFIFOCFG.FWIDTH.

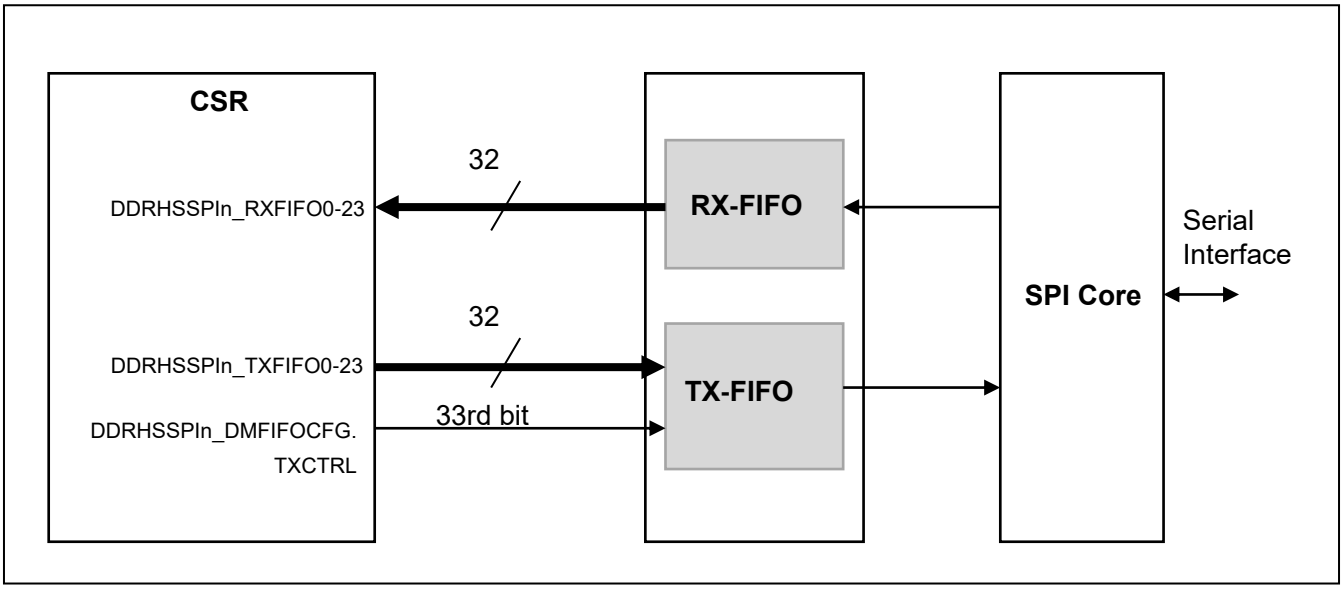
The Shift Register in the SPI Core is 32-bit wide. When the width of the FIFO is changed in the DDRHSSPIn_DMFIFOCFG.FWIDTH, the usable width of the Shift Register also changes accordingly.

Please refer to Figure 4-1, for details.

Note:

- *When the value of DDRHSSPIn_DMFIFOCFG.FWIDTH is greater than 0, the value of DDRHSSPIn_DMBCC.BCC should be programmed along the boundary of bytes according to DDRHSSPIn_DMFIFOCFG.FWIDTH, e.g. when FWIDTH is 0b11, the BCC should be a multiple of 4.*

Figure 4-1 DDRHSSPI FIFO's in Direct Mode



In addition to the 32-bit of data-width, each location in TX-FIFO has a 33rd control bit (known as DDRHSSPIn_DMFIFOCFG.TXCTRL bit), which decides whether

- the data from the TX-FIFO is to be transmitted by the SPI Core, OR
- the serial data lines are to be tri-stated.

4.1.3. TX-FIFO Control

If the DDRHSSPIn_DMFIFOCFG.TXCTRL bit is set to "1", the DDRHSSPI further decodes the bits[12:11] of the data in the corresponding TX-FIFO location. All possible value combinations of the DDRHSSPIn_DMFIFOCFG.TXCTRL bit and the bits[12:11] of TX-FIFO data (DDRHSSPIn_TXFIFO0-23.TXDATA) are shown in Table 4-1.

Table 4-1 Serial Data Output Control

DDRHSSPIn_DMFIFOCFG.TXCTRL	Bit 12 of TX-FIFO Data (TXDATA[12])	Bit 11 of TX-FIFO Data (TXDATA[11])	Description
0	Don't Care	Don't Care	Serial data output lines are neither tri-stated nor on-the-fly while transmitting the corresponding data. Data Rate Mode (SDR Mode or DDR Mode) and SPI data width (Legacy Mode, Quad Mode or Octal Mode) are as set in DDRHSSPIn_DMTRP Register. Plus, DDRHSSPIn_DMFIFOCFG.FWIDTH affects the valid bit field of TXDATA as below. (1) DDRHSSPIn_DMTRP.TRP[1:0]≠"11" FWIDTH TRP[1:0] SDATA 0b00 0b00 TXDATA[7:0] will be sent on SDATA[0]. 0b10 TXDATA[7:0] will be sent on SDATA[3:0]. 0b01 0b00 TXDATA[15:8][7:0] will be sent on SDATA[0]. 0b10 TXDATA[15:8][7:0] will be sent on SDATA[3:0]. (in the order of TXDATA[15:8]->[7:0], along the SCLK edge) 0b11 0b00 TXDATA[31:24][23:16][15:8][7:0] will be sent on SDATA[0]. 0b10 TXDATA[31:24][23:16][15:8][7:0] will be sent on SDATA[3:0]. (in the order of TXDATA[31:24]->[23:16]->[15:8]->[7:0] along the SCLK edge) In TX-and-RX Mode, it supports only Legacy Mode. So, RX-FIFO receives data from SDATA[1]. (2) DDRHSSPIn_DMTRP.TRP[1:0]= 0b11 (Only when DDRHSSPIn_MID.MID is 0x00000100 or 0x00000300) (2-1) SDR Mode: FWIDTH SDATA 0b00 TXDATA[7:4] will be sent on SDATA[7:4]. TXDATA[3:0] will be sent on SDATA[3:0]. 0b01 TXDATA[15:12][7:4] will be sent on SDATA[7:4]. TXDATA[11: 8][3: 0] will be sent on SDATA[3:0]. (in the order of TXDATA[15:8]->[7:0], along the SCLK edge) 0b11 TXDATA[31:28][23:20][15:12][7:4] will be sent on SDATA[7:4]. TXDATA[27:24][19:16][11: 8][3:0] will be sent on SDATA[3:0]. (in the order of TXDATA[31:24]->[23:16]->[15:8]->[7:0] along the SCLK edge). (2-2) DDR Mode: FWIDTH SDATA 0b00 Not allowed. 0b01 TXDATA[31:28][23:20] will be sent on SDATA[7:4].

DDRHSSPIn_ DMFIFOCFG. TXCTRL	Bit 12 of TX-FIFO Data (TXDATA[12])	Bit 11 of TX-FIFO Data (TXDATA[11])	Description
			TXDATA[27:24][19:16] will be sent on SDATA[3:0]. (in the order of TXDATA[31:24]->[23:16], along the SCLK edge) 0b11 TXDATA[31:28][23:20][15:12][7:4] will be sent on SDATA[7:4]. TXDATA[27:24][19:16][11: 8][3:0] will be sent on SDATA[3:0]. (in the order of TXDATA[31:24]->[23:16]->[15:8]->[7:0] along the SCLK edge) When in TXCTRL= 0, DDRHSSPI cannot transfer data at Dual Legacy Mode, since the SPI protocol is controlled just by DDRHSSPIn_DMTRP.TRP. Therefore, TX-and-RX Mode is not available for Dual Legacy Mode under this condition. **Common in (1) and (2):** The actual valid byte lane of TXDATA varies according to the lower address bits for TX-FIFO, e.g. if FWIDTH= 0b00 AND HSIZE=0x0 AND HADDR[1:0]= 0b01, TXDATA[15:8] is valid instead of TXDATA[7:0].
1	0	0	Serial data output lines are tri-stated for defined number of SCLK cycles . The number of dummy cycles is defined in TXDATA[7:4] bits in following way: 0b0000: Serial data output lines are tri-stated for 1 SCLK cycle 0b0001: Serial data output lines are tri-stated for 2 SCLK cycles .... 0b1111: Serial data output lines are tri-stated for 16 SCLK cycles Please note that maximum number of dummy cycles programmed per one TX-FIFO entry, is limited to be equivalent to sending one TX-FIFO level (with defined DDRHSSPIn_DMFIFOCFG.FWIDTH). The length of dummy cycles is limited as below. [Max Dummy Cycles] = [Bits per 1 FIFO entry] / [Bits per 1 SCLK cycle] For example, under the following conditions: − Quad Mode (by DDRHSSPIn_DMTRP.TRP[1:0]) − SDR data rate − number of dummy cycles is 4 − DDRHSSPIn_DMFIFOCFG.FWIDTH is 0b01 (16 bit-wide). In this example, the number of dummy cycle is within the limit. If the number of dummy cycles is larger than 4 in this example, the output serial data will be tri-stated for only 4 SCLK cycles and the rest will be ignored.

DDRHSSPIn_DMFIFOCFG.TXCTRL	Bit 12 of TX-FIFO Data (TXDATA[12])	Bit 11 of TX-FIFO Data (TXDATA[11])	Description
1	1	Don't Care	**SDR Mode** In SDR Mode (DDRHSSPIn_DMTRP.DDRM = 0), TXDATA[10] is ignored, and TXDATA[7:0] will be sent in SDR Mode, depending on TXDATA[9:8]: 0b00: Legacy Mode – If DDRHSSPIn_MID.MID is 0x00000100 or 0x00000300 and DDRHSSPIn_DMTRP. TRP is configured to Octal Mode (DDRHSSPIn_DMTRP.TRP[1:0]= 0b11), this works as Dual Legacy Mode, and a byte data will be sent as below (4 SCLK cycles): – TXDATA[7:4] will be sent on SDATA[4]. – TXDATA[3:0] will be sent on SDATA[0]. On the RX side of TX-and-RX Mode, a byte data will be received as below (4 SCLK cycles): – RXDATA[7:6] is received from {SDATA[5], SDATA[1]}. (1st cycle) – RXDATA[5:4] is received from {SDATA[5], SDATA[1]}. (2nd cycle) – RXDATA[3:2] is received from {SDATA[5], SDATA[1]}. (3rd cycle) – RXDATA[1:0] is received from {SDATA[5], SDATA[1]}. (4th cycle) The actual valid byte lane of RXDATA varies according to the lower address bits for RX-FIFO, e.g. if FWIDTH= 0b00 AND HSIZE=0x0 AND HADDR[1:0]= 0b01, RXDATA[15:8] is valid instead of RXDATA[7:0]. – If DDRHSSPIn_DMTRP.TRP is not configured to Octal Mode (DDRHSSPIn_DMTRP.TRP[1:0]≠ 0b11), then TXDATA[7:0] will be sent in Legacy Mode (8 SCLK cycles) on SDATA[0]. On the RX side of TX-and-RX Mode, a byte data will be received from SDATA[1] (8 SCLK cycles). 0b01: Not allowed. 0b10: Quad Mode – If DDRHSSPIn_DMTRP.TRP is not configured to Octal Mode (DDRHSSPIn_DMTRP.TRP[1:0]≠ 0b11), then TXDATA[7:0] will be sent in Quad Mode (2 SCLK cycles) on SDATA[3:0]. – If DDRHSSPIn_DMTRP.TRP is configured to Octal Mode (DDRHSSPIn_DMTRP.TRP[1:0]= 0b11), then it is not allowed to set TXDATA[9:8] to 0b10.

DDRHSSPIn_DMFIFOCFG.TXCTRL	Bit 12 of TX-FIFO Data (TXDATA[12])	Bit 11 of TX-FIFO Data (TXDATA[11])	Description
1	1	Don't Care	(Continued) 0b11: Dual Quad Mode (Only when DDRHSSPIn_MID.MID is 0x00000100 or 0x00000300) – If DDRHSSPIn_DMTRP.TRP is configured to Octal Mode (DDRHSSPIn_DMTRP.TRP[1:0]= 0b11), this works as Dual Quad Mode, and a byte data will be sent as below (1 SCLK cycles): – TXDATA[7:4] will be sent on SDATA[7:4]. – TXDATA[3:0] will be sent on SDATA[3:0]. – If DDRHSSPIn_DMTRP.TRP is not configured to Octal Mode (DDRHSSPIn_DMTRP.TRP[1:0]≠ 0b11), then it is not allowed to set TXDATA[9:8] to 0b11. **DDR Mode** In DDR Mode (DDRHSSPIn_DMTRP.DDRM = 1), then the behavior depends on TXDATA[10]: (1)SDR on-the-fly If TXDATA[10] = 0, then TXDATA[7:0] will be sent same as SDR Mode above. (2)DDR on-the-fly If TXDATA[10] = 1, then the behavior depends on bits TXDATA[9:8]: 0b00: Legacy Mode – If DDRHSSPIn_DMTRP.TRP is configured to Octal Mode (DDRHSSPIn_DMTRP.TRP[1:0]= 0b11), then it is not allowed to set TXDATA[9:8] to 0b00. – If DDRHSSPIn_DMTRP.TRP is not configured to Octal Mode (DDRHSSPIn_DMTRP.TRP[1:0]≠ 0b11), then TXDATA[7:0] will be sent in Legacy Mode (4 SCLK cycles) on SDATA[0]. 0b01: Not allowed. 0b10: Quad Mode – If DDRHSSPIn_DMTRP.TRP is not configured to Octal Mode (DDRHSSPIn_DMTRP.TRP[1:0]≠ 0b11), then TXDATA[7:0] will be sent in Quad Mode (1 SCLK cycles) on SDATA[3:0]. – If DDRHSSPIn_DMTRP.TRP is configured to Octal Mode (DDRHSSPIn_DMTRP.TRP[1:0]= 0b11), then it is not allowed to set TXDATA[9:8] to 0b10. 0b11: Dual Quad Mode (Only when DDRHSSPIn_MID.MID is 0x00000100 or 0x00000300) – If DDRHSSPIn_DMTRP.TRP is configured to Octal Mode (DDRHSSPIn_DMTRP.TRP[1:0]= 0b11), this works as Dual Quad Mode, and two bytes of data will be sent as below (1 SCLK cycles): – TXDATA[31:28][23:20] will be sent on SDATA[7:4]. – TXDATA[27:24][19:16] will be sent on SDATA[3:0]. (in the order of TXDATA[31:24]->[23:16] along the SCLK edge) – If DDRHSSPIn_DMTRP.TRP is not configured to Octal Mode (DDRHSSPIn_DMTRP.TRP[1:0]≠ 0b11), then it is not allowed to set TXDATA[9:8] to 0b11.

4.1.4. Available Command Formats of Serial Flash Memory

In order to apply to the command format of Serial Flash Memory, DDRHSSPI has the following features.

- Dynamic control of SPI Protocol
- Dynamic control of Data Rate Mode (SDR Mode or DDR Mode)
- Inserting dummy cycles

To realize these features, DDRHSSPI uses partial bits of TXDATA to control them. When DDRHSSPIn_DMFIFOCFG.TXCTRL bit is "1", these features are available by decoding some bits of TXDATA. By this control, DDRHSSPI is capable of Serial Flash command formats shown in Table 4-2 and Table 4-3.

Note:

- *When in writing TXDATA[12:8] for SPI control on-the-fly, it is allowed regardless the setting of DDRHSSPIn_DMFIFOCFG.FWIDTH.*

The available command formats of Serial Flash are different according to the value of DDRHSSPIn_MID. MID bits.

(1)SDR (DDRHSSPIn_DMTRP.DDRM= 0)

Table 4-2 The Available Serial Flash Command Format (SDR)

DDRHSSPIn_MID		Command	Address	Mode (*1)	Dummy
0x0000 0001	0x0000 0100 or 0x00000 300				
Available	Available	Legacy (1byte)	none	none	none
Available	Available	Legacy (1byte)	Legacy (3 / 4bytes)	none / 1byte	none / greater than 0
Available	Available	Legacy (1byte)	Quad (3 / 4bytes)	none / 1byte	none / greater than 0
N/A	Available	Dual Legacy (1byte each)	none	none	none
N/A	Available	Dual Legacy (1byte each)	Dual Legacy (3 / 4bytes each)	none / 1byte each	none
N/A	Available	Dual Legacy (1byte each)	Dual Quad (3 / 4bytes each)	none / 1byte each	none / greater than 0

(*1)The "Mode" field is a control code following Address field, and the actual code of Mode field depends on the Serial Flash products.

(2)DDR (DDRHSSPIn_DMTRP.DDRM= 1)

Table 4-3 The Available Serial Flash Command Format (DDR)

DDRHSSPIn_MID		Command	Address	Mode (*1)	Dummy
0x0000 0001	0x0000 0100 or 0x00000 300				
Available	Available	Legacy (1byte), SDR	Quad (3 / 4bytes), DDR	1byte, DDR	greater than 0
N/A	Available	Dual Legacy (1byte each), SDR	Dual Quad (3 / 4bytes each), DDR	1byte each, DDR	greater than 0

(*1)The "Mode" field is a control code following Address field, and the actual code of Mode field depends on the Serial Flash products.

For above Serial Flash command formats, DDRHSSPI supports the following SPI protocols on-the-fly.

Table 4-4 The Available Switching SPI Protocols On-the-Fly

DDRHSSPIn_MID		DDRHSSPIn_ DMTRP. TRP[1:0]	TXDATA[9:8] *1	Rx	Tx
0x0000 0001	0x0000 0100 or 0x00000 300				
Available	Available	0b00 (Legacy)	0b00 (Legacy)	SDATA[1]	SDATA[0]
N/A	N/A	0b00 (Legacy)	0b10 (Quad)	Not allowed	Not allowed
Available	Available	0b10 (Quad)	0b00 (Legacy)	SDATA[1]	SDATA[0]
Available	Available	0b10 (Quad)	0b10 (Quad)	Not allowed	SDATA[3:0]
N/A	Available	0b11 (Dual Quad)	0b00 (Dual Legacy)	SDATA[1], SDATA[5]	SDATA[0], SDATA[4]
N/A	Available	0b11 (Dual Quad)	0b10 (Quad)	Not allowed	Not allowed
N/A	Available	0b11 (Dual Quad)	0b11 (Dual Quad)	Not allowed	SDATA[7:0]

*1 With conditions of DDRHSSPIn_DMFIFOCFG.TXCTRL= 1 and TXDATA[12]= 1

4.1.5. FIFO Accesses in Direct Mode

Depending on the configured width of the FIFOs, 8-bit, 16-bit or 32-bit accesses are allowed to the DDRHSSPIn_RXFIFO0-23 and DDRHSSPIn_TXFIFO0-23 Registers.

If DDRHSSPIn_DMFIFOCFG.FWIDTH is set to 0b11 (32-bit wide mode), then only 32-bit access to the FIFO is allowed.

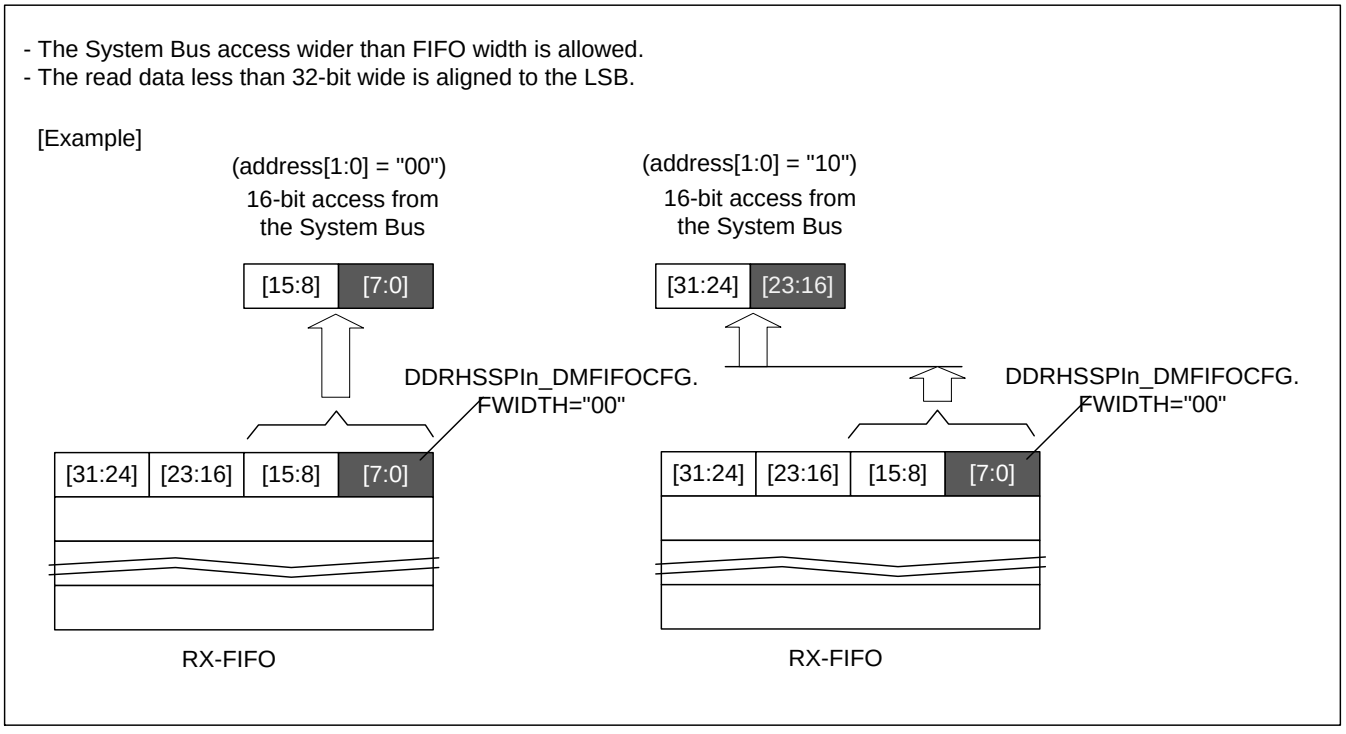
If DDRHSSPIn_DMFIFOCFG.FWIDTH is set to 0b01 (16-bit wide mode), then 16-bit and 32-bit accesses to the FIFO are allowed. A 32-bit access covers a whole word line for one FIFO entry. A 16-bit access with any alignment covers lower 16 bits of one FIFO entry.

If DDRHSSPIn_DMFIFOCFG.FWIDTH is set to 0b00 (8-bit wide mode), then 8-bit, 16-bit and 32-bit accesses to the FIFO are allowed. A 32-bit access covers a whole word line for one FIFO entry. A 16-bit access with any alignment covers lower 16 bits of one FIFO entry. An 8-bit access with any alignment covers the lower 8 bits of one FIFO entry.

When in writing data into TX-FIFO (to any of DDRHSSPIn_TXFIFO0-23), the written data go along with DDRHSSPIn_DMFIFOCFG.TXCTRL bit, and they are written into TX-FIFO together. And when the data width is not a word size (32 bits), the data is aligned to the LSB, before written into TX-FIFO.

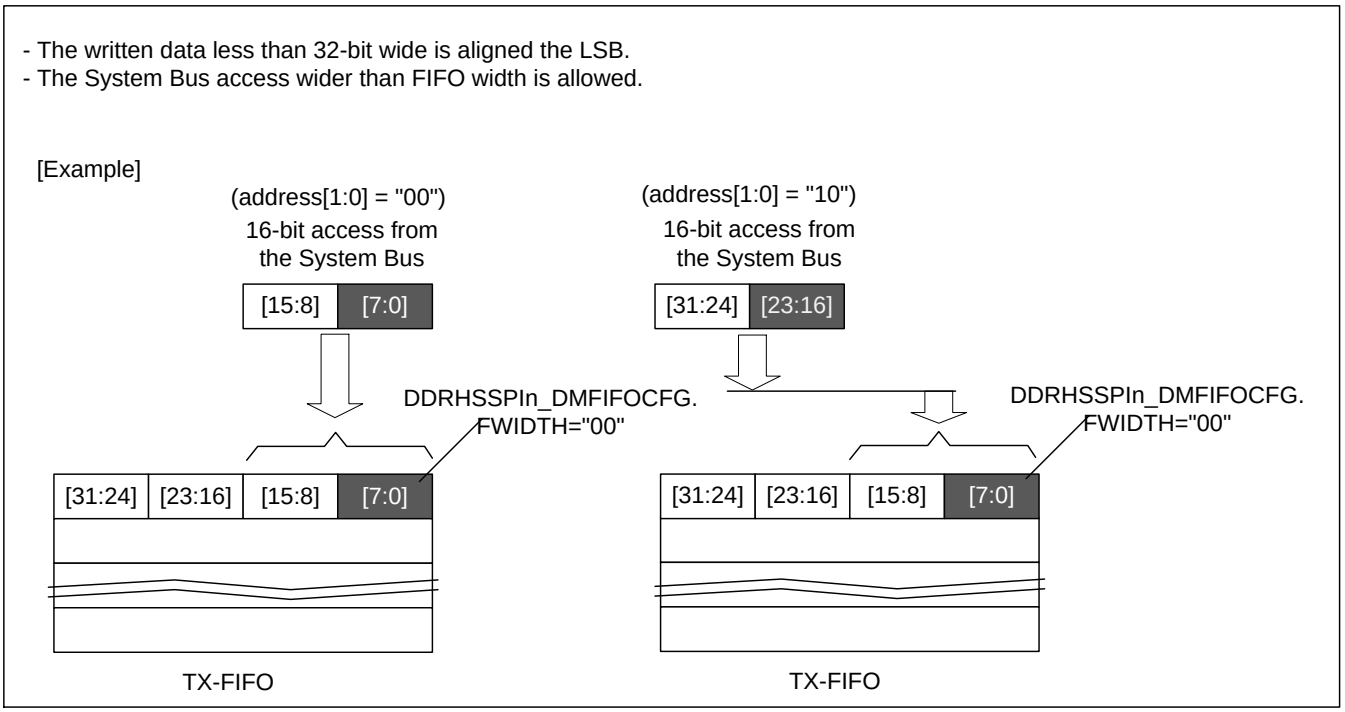
When the System Bus access is wider than RX-FIFO width (configured by DDRHSSPIn_DMFIFOCFG. FWIDTH), the width of read data follows FWIDTH value, and the read data is aligned to the LSB. Figure 4-2 shows this alignment and a notice of the System Bus address.

Figure 4-2 RX-FIFO Access at Greater Width



When the System Bus access is wider than TX-FIFO width (at DDRHSSPIn_DMFIFOCFG.TXCTRL= 0), it is allowed as well. In this case, the valid byte lane follows the System Bus address, as shown in Figure 4-3.

Figure 4-3 TX-FIFO Access at Greater Width (DDRHSSPIn_DMFIFOCFG.TXCTRL= 0)



When in DDRHSSPIn_DMFIFOCFG.TXCTRL= 1, the setting of DDRHSSPIn_DMFIFOCFG.FWIDTH is ignored at writing into TX-FIFO, and up to 32 bits could be relevant. (For details, please refer to the Table 4-1).

4.1.6. Service Requests

When operating in Direct Mode, Interrupt Service Requests to the MCU are triggered based on the current fill-levels of the TX-FIFO and the RX-FIFO, and their configured threshold values. Alternatively, the external DMA controller can be used for data transfers. When operating in Direct Mode, the DDRHSSPI has an interface with the DMA controller in the MCU for block transfers of data to/from its TX-FIFO and the RX-FIFO.
Interrupt flags are set when the current SPI transfer finishes.

Assertion of Interrupt Service Requests Based on FIFO Levels

The fill levels of both FIFOs are accessible from the system on the DDRHSSPIn_DMFIFOSTATUS. TXFLEVEL and the DDRHSSPIn_DMFIFOSTATUS.RXFLEVEL. The Interrupt Service Requests are generated by the DDRHSSPI based on (1)the FIFO fill levels and (2)their threshold values configured by the MCU.

- The DDRHSSPIn_TXF.TFLETS is set if the DDRHSSPIn_DMFIFOSTATUS.TXFLEVEL is less than or equal to DDRHSSPIn_DMFIFOCFG.TXFTH.

- The DDRHSSPIn_RXF.RFMTS is set if DDRHSSPIn_DMFIFOSTATUS.RXFLEVEL is greater than DDRHSSPIn_DMFIFOCFG.RXFTH.
- The DDRHSSPIn_TXF.TFES is set if the DDRHSSPIn_DMFIFOSTATUS.TXFLEVEL is 0, i.e. TX-FIFO is empty.
- The DDRHSSPIn_RXF.RFFS is set if the DDRHSSPIn_DMFIFOSTATUS.RXFLEVEL is 24, i.e. RX-FIFO is full. If the DDRHSSPI is configured for TX-Only operation, the RX-FIFO is not used.

Assertion of DMA Service Requests Based on FIFO Levels

The DDRHSSPI supports block transfer mechanism in the DMA controller. The DMA Service Requests are generated by the DDRHSSPI based on (1)the FIFO fill levels and (2)their threshold values configured by the MCU. To keep track of the number of successful data transfers to/from the TX-FIFO and/or the RX-FIFO, the DDRHSSPI internally uses two down-counters: the DDRHSSPI RX Block Counter and the DDRHSSPI TX Block Counter. Each of these counters is a 5 bits down counter, which is reloaded with the DMA Block size (for the respective channel) each time the DMA Service Request for that channel is asserted. The counters are decremented with every successful read or write accesses to the RX-FIFO or TX-FIFO. In case of the RX-FIFO accesses, the RX Block Counter is decremented unless the access was from the DAP Controller. The block counters do not underflow (i.e. the counter value remains 0 even if it is decremented while it is already 0).

DMA Write Channel has DDRHSSPIn_FAULTF.DWCBSFS bit, and DMA Read Channel has DDRHSSPIn_FAULTF.DRCBSFS bit. A DMA Block Size Fault is triggered if all of the following conditions are satisfied:

- The DMA Block Counter is decremented (due to a valid System Bus access) while it is already 0, AND
- DDRHSSPIn_DMAEN.RXDMAEN or DDRHSSPIn_DMAEN.TXDMAEN for the corresponding DMA Channel is "1", AND
- DDRHSSPIn_MCTRL.MES= 1, AND
- DDRHSSPIn_MCTRL.CSEN= 0

The DMA Read Channel must be setup to perform a block transfer of "DDRHSSPIn_DMFIFOCFG.RXFTH + 1" transfers. The DMA Write Channel must be setup to perform a block transfer of "24 - DDRHSSPIn_DMFIFOCFG.TXFTH" transfers. These values are reloaded into the DDRHSSPI's internal Block Counters, each time the DMA Service Request is asserted.

The DMA Block Counter is cleared to "0" in either of the following conditions:

- The corresponding DMA channel is disabled (in DDRHSSPIn_DMAEN Register), OR
- DDRHSSPIn_MCTRL.MES= 0, OR
- The mode of operation is switched from Direct Mode to Command Sequencer Mode.

The RX DMA Service Request (for the DMA Read Channel) is asserted if all of the following conditions are satisfied:

- DDRHSSPIn_DMFIFOSTATUS.RXFLEVEL is more than DDRHSSPIn_DMFIFOCFG.RXFTH. This condition is the same as the DDRHSSPIn_RXF.RFMTS= 1, AND
- The DDRHSSPI RX Block Counter value is 0, AND

- The DMA Read Channel acknowledgement signal is de-asserted by the DMA controller, AND
- A previous DMA Read Service Request is not pending, AND
- DDRHSSPIn_DMAEN.RXDMAEN= 1, AND
- DDRHSSPIn_FAULTF.DRCBSFS= 0, AND
- DDRHSSPIn_MCTRL.MES= 1, AND
- DDRHSSPIn_MCTRL.CSEN= 0, AND
- DDRHSSPIn_DMTRP.TRP[3:2]= 0b00

The RX DMA Service Request (for DMA Read Channel) is de-asserted if any of the following conditions is satisfied:

- The DMA Read Channel Service Request has been acknowledged by the DMA controller, OR
- DDRHSSPIn_DMAEN.RXDMAEN= 0, OR
- DDRHSSPIn_MCTRL.MES= 0, OR
- DDRHSSPIn_MCTRL.CSEN="1"

The TX DMA Service Request (for DMA Write Channel) is asserted if all of the following conditions are satisfied:

- DDRHSSPIn_DMFIFOSTATUS.TXFLEVEL is less than or equal to DDRHSSPIn_DMFIFOCFG.TXFTH. This condition is the same as the DDRHSSPIn_TXF.TFLETS= 1, AND
- The DDRHSSPI TX Block Counter value is 0, AND
- The DMA Write Channel acknowledgement signal is de-asserted by the DMA controller, AND
- The Previous DMA Write Service Request is not pending, AND
- DDRHSSPIn_DMAEN.TXDMAEN= 1, AND
- DDRHSSPIn_FAULTF.DWCBSFS= 0, AND
- DDRHSSPIn_MCTRL.MES= 1, AND
- DDRHSSPIn_MCTRL.CSEN= 0, AND
- DDRHSSPIn_DMTRP.TRP[2]= 0

The TX DMA Service Request (for DMA Write Channel) is de-asserted if any of the following conditions is satisfied:

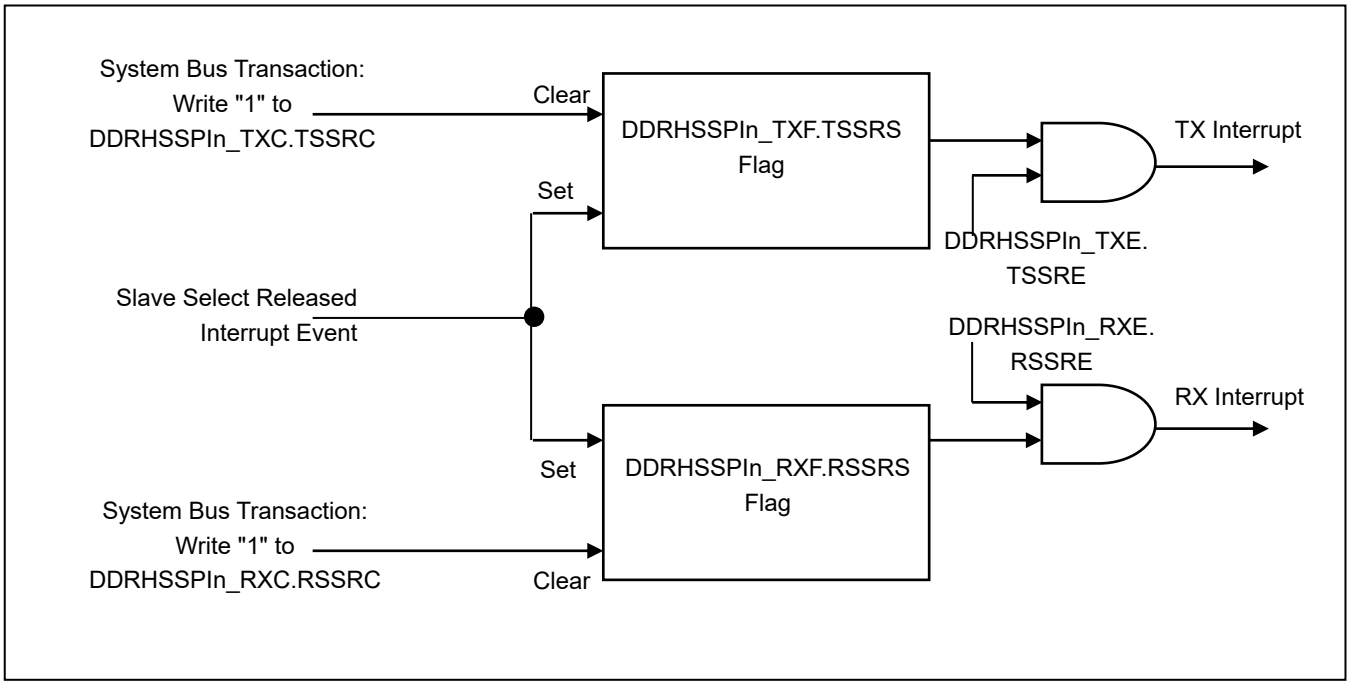
- The DMA Write Channel Service Request has been acknowledged by the DMA controller, OR
- DDRHSSPIn_DMAEN.TXDMAEN= 0, OR
- DDRHSSPIn_MCTRL.MES= 0, OR
- DDRHSSPIn_MCTRL.CSEN="1"

It is highly recommended that either value of DDRHSSPIn_DMFIFOCFG.TXFTH or DDRHSSPIn_DMFIFOCFG.RXFTH should not exceed 23.

Assertion of Service Requests on End of Transfer

While operating in Direct Mode, the DDRHSSPI also triggers interrupts when the Slave Select line is de-asserted. The Slave Select De-assertion event is routed onto two Interrupt Flags: DDRHSSPIn_TXF.TSSRS and DDRHSSPIn_RXF.RSSRS, which have individual Interrupt-Clear and Interrupt-Enable bits. The interrupt flags are routed onto individual Interrupt Signals. This logic is indicated in Figure 4-4.

Figure 4-4 Routing of the "Slave Select Released" Interrupt Event



4.1.7. SPI Transfers

The DDRHSSPI can initiate the transfers onto one of the four Slave Select lines, selected by the DDRHSSPIn_DMPSEL.PSEL. The following are major steps and other relevant information for initiating SPI transfers:

- Communication Attributes of the DDRHSSPI

  Communication over the Serial Interface has several attributes, like: Frequency of the Serial Interface Clock, etc. These communication attributes should be same among Serial Flash Memories. When the DDRHSSPI is working in Direct Mode, it can be interfaced with up to 4 Serial Flash Memories at Legacy/Quad mode (8 at Dual Legacy/Dual Quad mode).
  These device-specific communication attributes can be configured in the DDRHSSPIn_PCC0-3 Registers in the CSR.

- Configuration parameter change

  It is possible to switch between different bit widths and between SDR Mode and DDR Mode on-the-fly, without releasing the Slave Select line. However, it is up to the SW to take care of the timing of the switching from different SPI widths or clock modes so that data in TX-FIFO or RX-FIFO does not get lost or corrupted because of the switch.
  The ability to change clock mode and access width on-the-fly is implemented in such way that the DDRHSSPI can support various Serial Flash Memories access protocols (such as command and address).

- Initiating the Serial Transfers:

  When DDRHSSPIn_MCTRL.MES= 1 and DDRHSSPIn_MCTRL.CSEN= 0, serial transfers are initiated by DDRHSSPI when the DDRHSSPIn_DMSTART.START bit is set to "1".

If the DDRHSSPIn_DMTRP.TRP is programmed such that transmission is enabled, and if the TX-FIFO is empty when the DDRHSSPIn_DMSTART.START bit is set to "1"; then the DDRHSSPI delays the initiation of the serial transfer until the TX-FIFO is written by the software.
Byte Counter Mode is necessary for controlling the transfer length, and the DDRHSSPIn_DMBCS Register will be loaded with the value in DDRHSSPIn_DMBCC Register immediately when the DDRHSSPIn_DMBCC is written.
Once the DDRHSSPIn_DMSTART.START bit is set to "1", it cannot be reset by the software. The DDRHSSPI resets the bit after it has finished the Serial Transfer.
Writing a "1" to the DDRHSSPIn_DMSTART.START bit while SSEL is asserted has no effect.

- Halting a transfer due to lack of TX-DATA or due to lack of RX-FIFO space:

  In TX-Only Mode, an ongoing transfer can be halted by keeping the Slave Select asserted and by halting the Serial Clock. The DDRHSSPI automatically halts the Serial Clock, while it is waiting for the TX-FIFO to be written. The condition to halt a serial transfer in TX-Only Mode is as below.

  - the TX-FIFO and the TX Shift Register are empty

  And, when above condition is not satisfied while in halting, DDRHSSPI gets out of the halting status and resume the serial transfer (with starting the toggling of the Serial Clock).
  In TX-and-RX Mode, halting of serial transfer must be avoided. The condition to halt a serial transfer is as below.

  - the TX-FIFO and the TX Shift Register are empty, OR
  - the RX-FIFO and the RX Shift Register are full

- Controlling the transfer length:

  The transfer length is controlled as follows.
  The MCU is supposed to initialize the DDRHSSPIn_DMBCC.BCC with the number of bytes to be transferred over the Serial Interface, before the Slave Select output is asserted. When the DDRHSSPI transfers are initiated, the DDRHSSPI counts the number of bytes that are transferred and releases the Slave Select line after the number of bytes indicated in DDRHSSPIn_DMBCC.BCC have been transferred.
  When the value of DDRHSSPIn_DMFIFOCFG.FWIDTH is greater than 0, the value of DDRHSSPIn_DMBCC.BCC should be a multiple of "DDRHSSPIn_DMFIFOCFG.FWIDTH + 1".

In TX-Only Mode, and SW sets number of dummy cycles more than 1 FIFO entry, the actual length of dummy cycles on the Serial Interface become same as 1 FIFO entry (the excessive number is ignored).

In TX-and-RX Mode, when transmitted data is for dummy cycles by DDRHSSPIn_DMFIFOCFG.TXCTRL= 1, the received data is input according to the dummy length. Please refer to Table 4-1 for details. The length of dummy cycles shall not be more than 1 FIFO entry.

4.2. Command Sequencer Mode

In Command Sequencer Mode, the DDRHSSPI acts as an SPI master for interfacing with the external Serial Flash Memories. Each of the 4 Slave Select lines can be used for mapping uniform type of Serial Flash Memories such as Cypress FL-S series. Memory accesses initiated by the MCU and the other System Bus masters on the System Bus, are automatically converted to the Serial Flash Memory read commands by the DDRHSSPI.

This section describes the Command Sequencer Mode of the DDRHSSPI.

Notes:

- *To use the Command Sequencer Mode, please make sure that the "Serial Flash" has a "Continuous Read Command Sequence", which skips the instruction bit field and starts with address field from the second SPI transaction. DDRHSSPI needs to work on the first SPI transaction at the Direct Mode (TX-Only Mode) to control the variable bit width on the SPI, and this feature covers the first SPI transaction. Then, the Command Sequencer Mode can be used from the second SPI transaction which continues from the first one. Since the first SPI transaction is controlled under TX-Only Mode, the read data from the Serial Flash are ignored in that transaction. Valid reading of Serial Flash is run at the Command Sequencer Mode.*
- *In memory reading, the bit width on the SPI from the address field should be constantly Quad (or Dual Quad).*

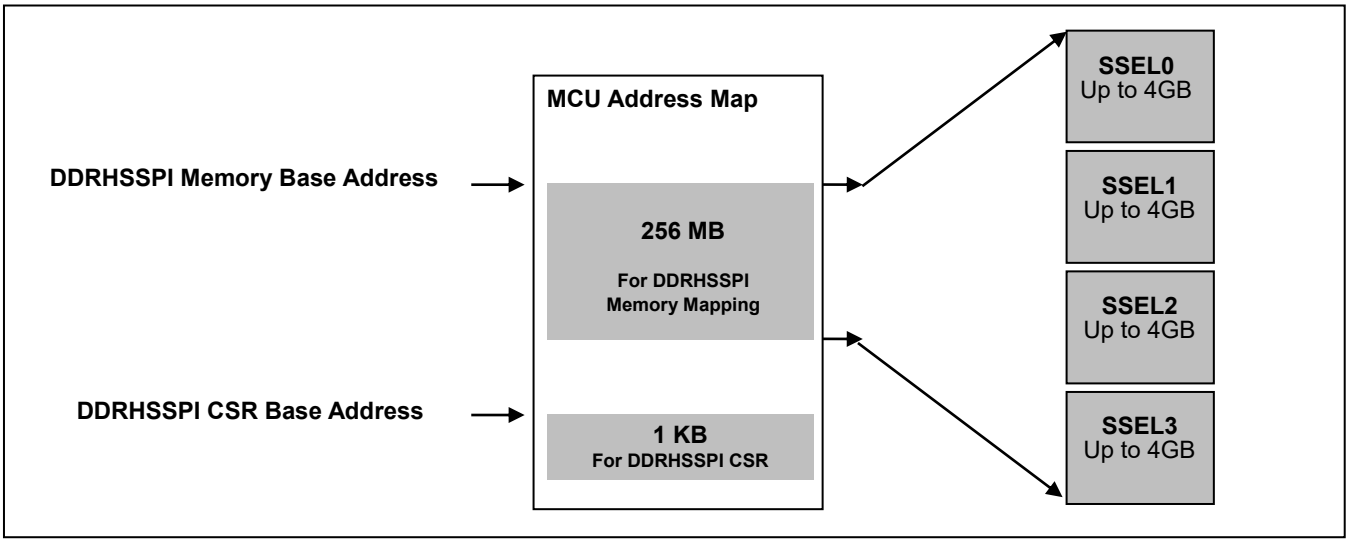
Memory Mapping

The Command Sequencer Mode can be used for memory mapping of up to 4 Slave Selects. All mapped Serial Flash Memories shall be of the same family and with the same internal configuration.

In Command Sequencer Mode, the DDRHSSPI allocates a memory space of 256MB, for mapping up to 4 external Serial Flash Memories. By using the Address Extension mechanism in Command Sequencer, each Slave Select can address a memory of up to 4GB (i.e. 32-bit address bus). The Address Extension mechanism allows concatenation of the most significant bits from a 19-bit Address Extension Register (i.e. the DDRHSSPIn_CSAEXT Register) with partial bits from the System Bus address, to form a 32-bit address to be accessed on each Slave Select. This feature is explained in detail in the subsequent sub-sections of this chapter.

Thus, the 256MB of the MCU address-space can virtually be mapped to the 16GB space of external Serial Flash Memories, as shown in Figure 4-5.

Figure 4-5 Mapping of Serial Flash Memories on the Slave Select Lines



Selection of Slaves

The DDRHSSPIn_CSCFG.MSEL indicates the size of the System Bus address space associated with each Slave Select line.

Based on the value of the DDRHSSPIn_CSCFG.MSEL and the address placed by the MCU (or the other System Bus Master, like the DMA Controller) on the System Bus, the DDRHSSPI Command Sequencer decides which of the 4 Slave Select lines is to be asserted. Please refer to Table 4-5 for details.

As an example, in case DDRHSSPIn_CSCFG.MSEL indicates that the System Bus address space associated with each Slave Select is of 8KB. If the System Bus address is between "the DDRHSSPI Memory Base Address" and "the DDRHSSPI Memory Base Address + 8KB", then the Slave Select 0 is asserted. If the System Bus address is between "the DDRHSSPI Memory Base Address + 8KB" and "the DDRHSSPI Memory Base Address + 16KB", then the Slave Select 1 is asserted, and so on. If the System Bus Address is beyond "the DDRHSSPI Memory Base Address + 32KB", then the address is out of range and the DDRHSSPIn_FAULTF.UMAFS flag is set.

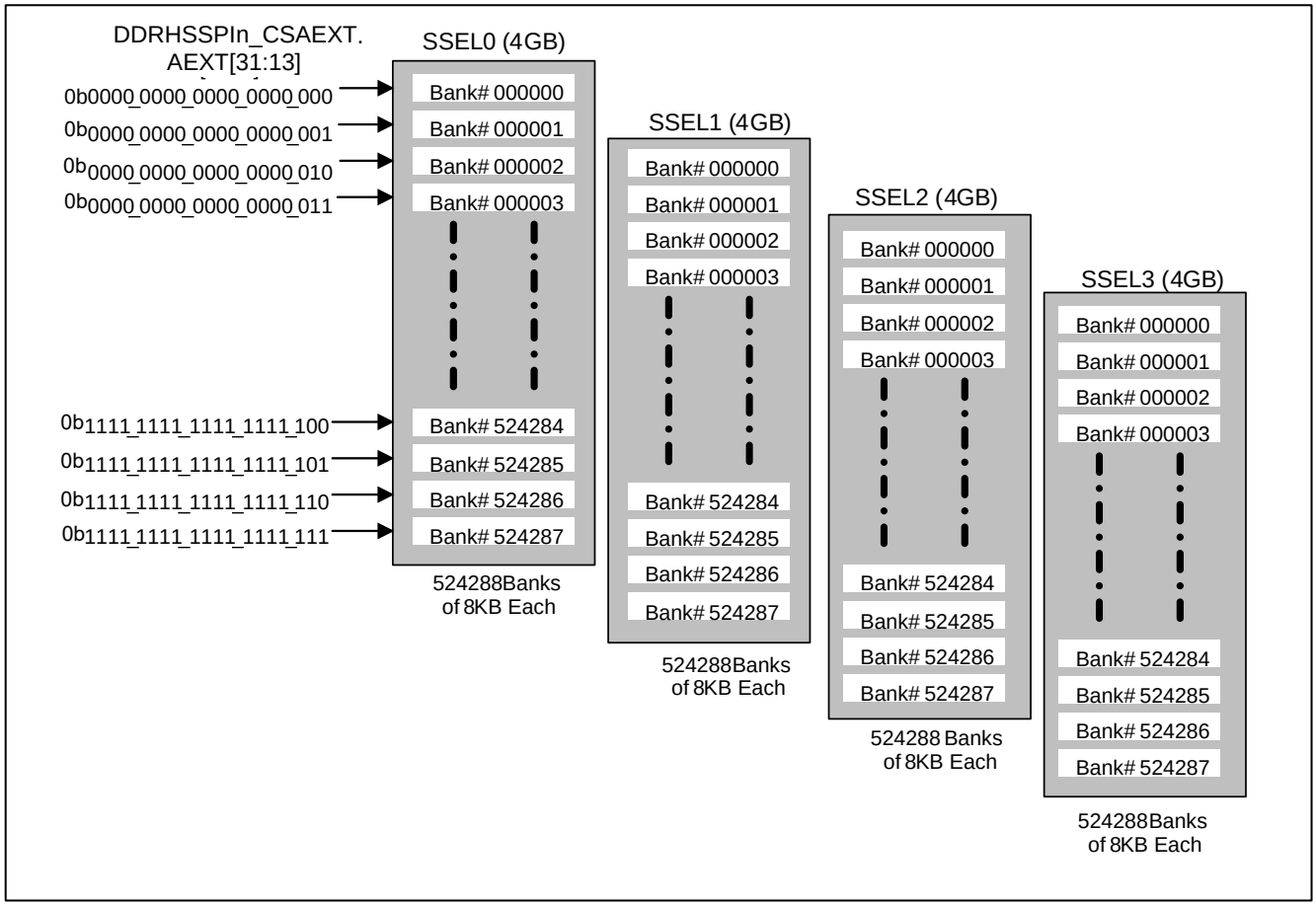
Even if the address of the System Bus access is in the valid range, if the target Slave Select is not set valid, the DDRHSSPIn_FAULTF.UMAFS flag is set as well. For example, if DDRHSSPIn_CSCFG.SSEL0EN is "0", any memory access corresponding to the Slave Select 0 ends up asserting an error of DDRHSSPIn_FAULTF.UMAFS = 1.

Generation of 32-Bit Memory Address

The Address Extension Mechanism allows the MCU to access each Serial Flash Memory of a size of 4GB. Each Serial Flash Memory is accessible by dividing into several memory banks. The size of each bank can be programmed in the DDRHSSPIn_CSCFG.MSEL. Each bank can be selected by changing the value in the DDRHSSPIn_CSAEXT Register. To select a new bank, the software shall program a new value to the DDRHSSPIn_CSAEXT Register. By switching banks, it is possible to address a Serial Flash Memory of 4GB size.

Please refer to Figure 4-6. It shows how each 4GB Serial Flash Memory consists of 524288 banks when the DDRHSSPIn_CSCFG.MSEL is programmed to 0b0000.

Figure 4-6 Addressing 4GB Serial Flash Memories on Each Slave Select, Through Different Banks (DDRHSSPIn_CSCFG.MSEL= 0b0000)



The Least Significant Bits of the System Bus Address received by the DDRHSSPI on the System Bus are used as the offset within the bank selected by the Address Extension bits. The final 32-bit address on the SPI is generated by the following concatenation:

- Upper bits: Register bits DDRHSSPIn_CSAEXT.AEXT[31:m]
- Lower bits: Address on the System Bus[m-1:0]

(m = DDRHSSPIn_CSCFG.MSEL[3:0] + 13.)

For details, please refer to Table 4-5.

Table 4-5 MCU Address Space to Memory Address Mapping

DDRHSSPIn_CSCFG.MSEL	Size of a Memory Bank on Each Slave Select / Size of the System Bus Address Range Associated with Each Slave Select	Number of Slave Select Lines That can be Activated	Number of Bits Used from DDRHSSPIn_CSAEXT Register, for Selection of the Memory Bank on a Slave Select	Number of Bits Used from Address Bus for Addressing the Memory Location within a Bank
0000	8K Bytes	SSEL0, SSEL1, SSEL2 and SSEL3	AEXT[31:13]	HADDR[12:0]
0001	16K Bytes		AEXT[31:14]	HADDR[13:0]
0010	32K Bytes		AEXT[31:15]	HADDR[14:0]
0011	64K Bytes		AEXT[31:16]	HADDR[15:0]
0100	128K Bytes		AEXT[31:17]	HADDR[16:0]
0101	256K Bytes		AEXT[31:18]	HADDR[17:0]
0110	512K Bytes		AEXT[31:19]	HADDR[18:0]
0111	1M Bytes		AEXT[31:20]	HADDR[19:0]
1000	2M Bytes		AEXT[31:21]	HADDR[20:0]
1001	4M Bytes		AEXT[31:22]	HADDR[21:0]
1010	8M Bytes		AEXT[31:23]	HADDR[22:0]
1011	16M Bytes		AEXT[31:24]	HADDR[23:0]
1100	32M Bytes		AEXT[31:25]	HADDR[24:0]
1101	64M Bytes		AEXT[31:26]	HADDR[25:0]
1110	128M Bytes	SSEL0 and SSEL1 Only	AEXT[31:27]	HADDR[26:0]
1111	256M Bytes	SSEL0 Only	AEXT[31:28]	HADDR[27:0]

Last two columns in Table 4-5 indicate which bits from the DDRHSSPIn_CSAEXT.AEXT and the address bus (i.e. HADDR) are concatenated, to get the final 32-bit address of the Serial Flash Memory.

The output memory address on the SPI can be 32 or 24 bits according to the Serial Flash Memory products, and this is controlled by the register settings of DDRHSSPIn_RDCSDC0-11 (for details, refer to Table 4-6).

Internal Prefetch Buffer

In Command Sequencer Mode the DDRHSSPI internally has one Prefetch Buffer for a temporary storage of received data.

This Prefetch Buffer consists of 48 locations, 32-bit wide.

Prefetch Buffer shall support 32 bits, 16 bits and 8 bits reads using the following mechanism:

- When a byte or halfword is read at a random (non-sequential) address on the System Bus, the Command Sequence is generated with the address aligned to the word boundary, which covers the required data.
- In the actual Command Sequence in transmission, the two least significant address bits are fixed to "00" (i.e. these two bits from the System Bus are ignored).

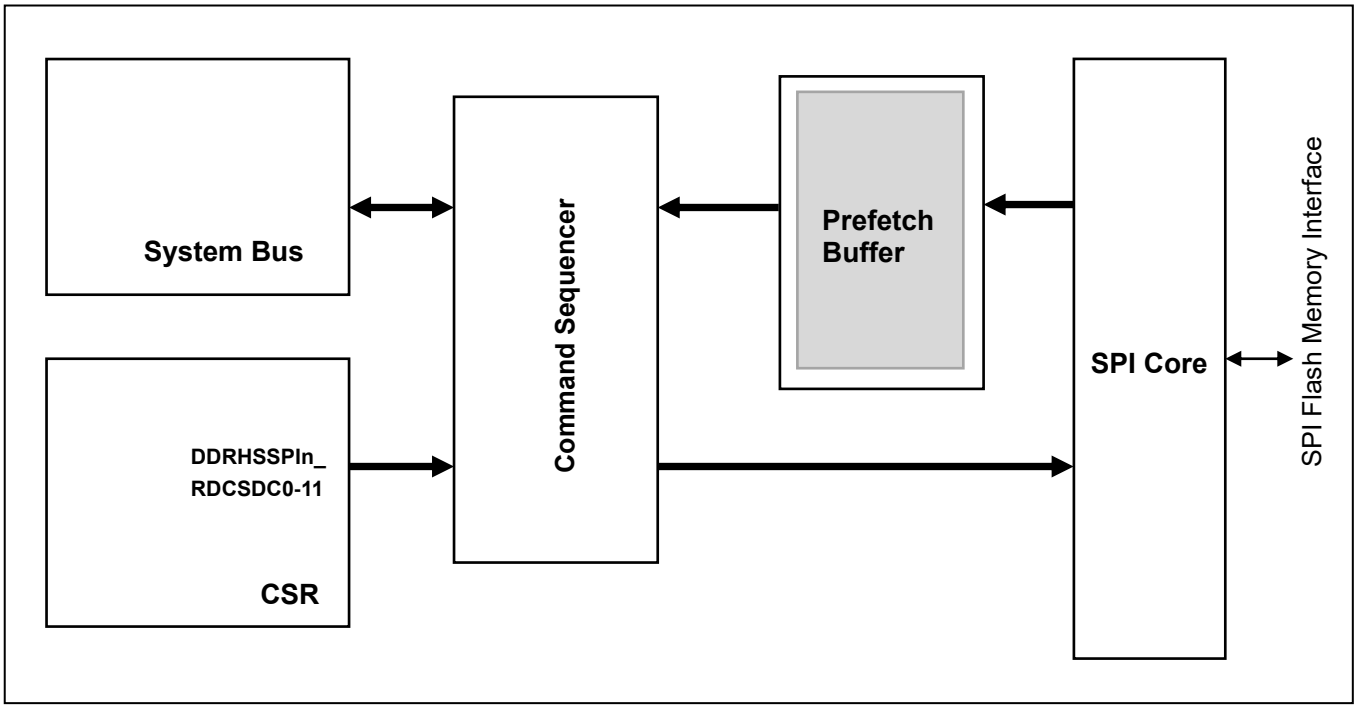
As long as the software reads a series of continuous data from the top most position of the Prefetch Buffer, there will be no additional Command Sequence on the SPI.

When the Prefetch Buffer is read, it internally works as a FIFO, and a word data in the next location pops out only when the current read data includes the last byte in the word line. The typical cases to pop the Prefetch Buffer are as below:

- When the whole aligned word is read, FIFO will be popped
- When the upper aligned halfword (word address + 2) is read, FIFO will be popped
- When the most significant byte (word address + 3) is read, FIFO will be popped
- In all other cases, FIFO will be read and will not be popped

In order to obtain the maximum throughput, word access (i.e. 32-bit access) is highly recommended.

Figure 4-7 DDRHSSPI in Command Sequencer Mode



In order to see if the requested data is already available on the top of the Prefetch Buffer, the SW can read the DDRHSSPIn_CSPREFETCHADDR Register which keeps the first available address at the top of the Prefetch Buffer.

Initiation of Command Sequence

Whenever the Command Sequencer detects a System Bus read access for the memory mapped Serial Flash Memory, it initiates a corresponding Memory Read Command Sequence on the SPI, assembles the received data, and responds to the System Bus with the Memory Read data. The Command Sequencer Mode supports only Memory Read commands of Serial Flash Memory.

While the DDRHSSPI initiates a memory-read command and receives the read-data from the Serial Flash Memory, the DDRHSSPI inserts WAIT states on the System Bus. The DDRHSSPI keeps track of the previous address and the System Bus transfer type issued by the System Bus master. If the new transaction address is not contiguous, then a new serial transaction is issued on the Serial Interface.

Idle Timeout

After a Serial Flash Memory is accessed in the Command Sequencer Mode, the DDRHSSPI keeps asserting the Slave Select line even if the System Bus transaction is over. And DDRHSSPI can deassert the Slave Select automatically after waiting for a certain period of time. To use this feature, it is necessary to set DDRHSSPIn_CSCFG.ITIMEREN bit to "1".

If this feature is disabled, Slave Select remains asserted after any transfer on the Serial Flash Memory.

If this feature is enabled, the following part explains the Idle Timer behavior:

During idle cycles on the System Bus, DDRHSSPI continues reading the Serial Flash Memory until the prefetch Buffer reaches full. After reaching full, the Slave Select shall stay asserted, and the Idle Timer shall start running. If there are no subsequent System Bus accesses to the consecutive memory address during the programmed idle time, the DDRHSSPI deasserts the Slave Select after the Idle Timer expires. The deassertion of Slave Select indicates the termination of the transfer and flushes the Prefetch Buffer. If the consecutive memory read access resumes before the Idle Timer expires, the Idle Timer shall be cleared again. Within the predefined time period, defined by the DDRHSSPIn_CSITIME.ITIME, the DDRHSSPI determines whether to extend the current serial transaction. If the following conditions are satisfied:

- A new System Bus transaction is detected on the System Bus, AND
- The new address is contiguous with the previous transaction

the DDRHSSPI extends the current serial transfer, instead of initiating a new Command Sequence. Thus, it can reduce the access time on the Serial Interface.

If the following conditions are satisfied:

- There is a subsequent memory access during the idle time, AND

- The access is to a non-consecutive memory address, AND

- The requested read data is not ready on the top of the Prefetch Buffer

the DDRHSSPI de-asserts the Slave Select (indicating the termination of the current transfer), even before the Idle Timer expires. Then, the DDRHSSPI flushes the Prefetch Buffer and initiates a new Command Sequence.

Thus, the DDRHSSPIn_CSITIME.ITIME is used to enhance the overall performance of the memory accesses by extending the serial transaction for a programmed period.

The value of the DDRHSSPIn_CSITIME.ITIME is based on the System Bus Clock (iHCLK).

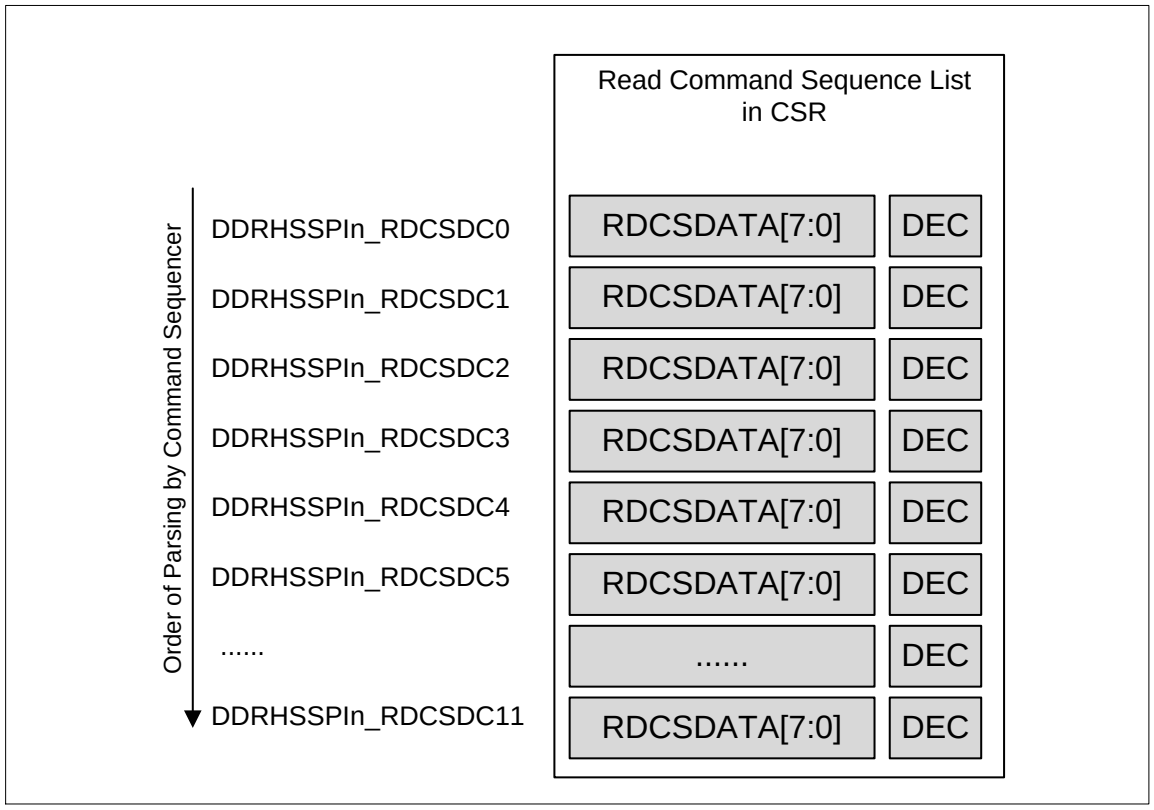
Configuration of Command Sequencer Mode in the CSR

The Command Sequencer supports memory read accesses. The sequence of command phases (i.e. command-phase, address-phase and data-phase) by the combination of Direct Mode and Command Sequencer Mode, is configured by the software (during initialization of the DDRHSSPI) in the CSR.

- Generation of Serial Flash Memory Read Command Sequence

  For memory read transactions, the sequence of command phases (address-phase and data-phase) can be configured in the list of 12 registers: DDRHSSPIn_RDCSDC0-11. Starting from the DDRHSSPIn_RDCSDC0, up to the DDRHSSPIn_RDCSDC11 each of the 12 registers in the list is parsed. Please refer to Figure 4-8.

Figure 4-8 Memory Read Command Sequence List



The DEC bit in each of these registers determines whether to decode RDCSDATA[2:0] (as shown in Table 4-6). If DEC bit is "0", the data byte in RDCSDATA[7:0] is transmitted as it is.

Table 4-6 Decoding of the Read Command Sequence List

DEC	RDCSDATA [7:0]	Description
0	Don't Care	Transmit RDCSDATA[7:0] as it is.
1	0b00000000	Transmit address bits [7:0] of the Serial Flash Memory to be accessed
1	0b00000001	Transmit address bits [15:8] of the Serial Flash Memory to be accessed
1	0b00000010	Transmit address bits [23:16] of the Serial Flash Memory to be accessed
1	0b00000011	Transmit address bits [31:24] of the Serial Flash Memory to be accessed

DEC	RDCSDATA [7:0]	Description
1	0bXXXXX100	Dummy cycle(s). The bit field RDCSDATA [7:3] means the length of dummy cycles: SDATA[7:3]=0b00000 -> Generates 1 dummy cycle of the SCLK SDATA[7:3]=0b00001 -> Generates 2 dummy cycles of the SCLK ..... Bits at "X" are for the value to control the length of dummy cycles.
1	0b00000111	End of List

The Command Sequencer switches to data-read cycles by the following conditions:

- If It gets "End of List", OR
- After the DDRHSSPIn_RDCSDC11 Register

In response to the System Bus Read transaction, during data-read cycles, the serial data on the SDATA lines is sampled, and the assembled data is returned to the System Bus master.

The actual patterns allowed in this Command Sequence List is limited in the following formats. This is based on using a Continuous Read Command Sequence, and the Command Sequence begins with Address field for the Serial FLASH Memory.

(1)24-bit Address

Table 4-7 Command Sequence List for 24-Bit Address

Register	RDCSDATA [7:0]	DEC	Description
DDRHSSPIn_ RDCSDC0	0b00000010	1	address bits [23:16] of the Serial Flash Memory
DDRHSSPIn_ RDCSDC1	0b00000001	1	address bits [15:8] of the Serial Flash Memory
DDRHSSPIn_ RDCSDC2	0b00000000	1	address bits [7:0] of the Serial Flash Memory
DDRHSSPIn_ RDCSDC3	byte data	0	Mode code *1
DDRHSSPIn_ RDCSDC4	0bXXXXX100	1	Dummy cycles Bits at "X" are for the value to control the length of dummy cycles.
DDRHSSPIn_ RDCSDC5	0b00000111	1	End of List

*1: The "Mode" field is a control code following Address field, and the actual code of Mode field depends on the Serial Flash products.

(2)32-bit Address

Table 4-8 Command Sequence List for 32-Bit Address

Register	RDCSDATA [7:0]	DEC	Description
DDRHSSPIn_RDCSDC0	0b00000011	1	address bits [31:24] of the Serial Flash Memory
DDRHSSPIn_RDCSDC1	0b00000010	1	address bits [23:16] of the Serial Flash Memory
DDRHSSPIn_RDCSDC2	0b00000001	1	address bits [15:8] of the Serial Flash Memory
DDRHSSPIn_RDCSDC3	0b00000000	1	address bits [7:0] of the Serial Flash Memory
DDRHSSPIn_RDCSDC4	byte data	0	Mode code *1
DDRHSSPIn_RDCSDC5	0bXXXXX100	1	Dummy cycles Bits at "X" are for the value to control the length of dummy cycles.
DDRHSSPIn_RDCSDC6	0b00000111	1	End of List

*1: The "Mode" field is a control code following Address field, and the actual code of Mode field depends on the Serial Flash products.

The Operation of Memory Read Access

To perform memory read accesses to the Serial Flash Memories, it is necessary to run the following combination of Direct Mode and Command Sequencer Mode, because the Command Sequencer Mode cannot change SPI width on-the-fly.

1. Switch to Direct Mode, if it is in Command Sequencer Mode. Clear DDRHSSPIn_MCTRL.CSEN bit to "0".
2. Write a byte count of the transmission to the DDRHSSPIn_DMBCC.BCC. The byte count shall be a sum of command, address, and mode code. For example, if the address has 4 bytes, the BCC is 6.
3. Set the DDRHSSPIn_DMTRP.TRP to 0b1010 (TX-Only Mode and Quad Mode).
4. When in DDR Mode, set the DDRHSSPIn_DMTRP.DDRM bit to "1". Otherwise, clear the DDRHSSPIn_DMTRP.DDRM bit to "0".
5. Set the DDRHSSPIn_DMFIFOCFG.TXCTRL bit to "1".
6. Write a byte for the command field, into TX-FIFO. TXDATA[12] = 1, TXDATA[9:8] = 0b01 (Legacy Mode on-the-fly). TXDATA[10] = 0 (SDR Mode on-the-fly). TXDATA[7:0] is a byte for the command. Please refer to the data sheet of the Serial Flash Memory about the command code.
7. Write a byte for the address field [31:24], into TX-FIFO. TXDATA[12] = 1, TXDATA[9:8] = 0b10 (Quad Mode on-the-fly). TXDATA[10] = DDRHSSPIn_DMTRP.DDRM bit value. TXDATA[7:0] is a byte for the address[31:24].
8. Write bytes for the address field [23:16], [15:8] and [7:0] into TX-FIFO as well.
9. Write a byte for the mode field following the address field, into TX-FIFO. TXDATA[12] = 1, TXDATA[9:8] = 0b10 (Quad Mode on-the-fly). TXDATA[10] = DDRHSSPIn_DMTRP.DDRM bit value. TXDATA[7:0] is a byte for the mode. Please refer to the data sheet of the Serial Flash Memory about the mode code.
10. Write "1" to the DDRHSSPIn_TXC.TSSRC bit.

11. Set the DDRHSSPIn_TXE.TSSRE bit to "1".
12. Select a Serial Flash Memory by setting the DDRHSSPIn_DMPSEL.PSEL value.
13. Set the DDRHSSPIn_DMSTART.START bit to "1".
14. Wait for the TX Interrupt Request, or run a polling of DDRHSSPIn_TXF Register until the DDRHSSPIn_TXF.TSSRS bit to be "1".
15. Write "1" to the DDRHSSPIn_TXC.TSSRC bit.
16. Clear the DDRHSSPIn_TXE.TSSRE bit to "0".
17. Switch to Command Sequencer Mode. Set DDRHSSPIn_MCTRL.CSEN bit to "1".
18. Set the DDRHSSPIn_CSCFG.MBM same as DDRHSSPIn_DMTRP.TRP[1:0] value. Set DDRHSSPIn_CSCFG.MSEL to appropriate value. Set at least one of DDRHSSPIn_CSCFG.SSEL0EN, DDRHSSPIn_CSCFG.SSEL1EN, DDRHSSPIn_CSCFG.SSEL2EN or DDRHSSPIn_CSCFG.SSEL3EN to "1", in order to contain the Serial Flash Memory selected by DDRHSSPIn_DMPSEL.PSEL. Set DDRHSSPIn_CSCFG.DDRMODE bit same as DDRHSSPIn_DMTRP.DDRM bit.
19. Set the appropriate value to DDRHSSPIn_CSITIME.ITIME.
20. Set the appropriate value to DDRHSSPIn_CSAEXT.AEXT.
21. Write appropriate values to DDRHSSPIn_RDCSDC0-11 Registers according to Table 4-7 or Table 4-8.
22. Perform memory read accesses to the address where Serial Flash Memories are allocated. The memory accesses on the System Bus are translated to serial transactions on the Serial Interface.
23. As long as performing the memory read accesses, there is no need to change the register settings. When in issuing a new set of command, it is necessary to stop current Continuous Read Command Sequence. To stop it, go to the next step.
24. Operate steps from 1 to 16 in Direct Mode again to get out of current Continuous Read Command Sequence. In this operation, please write a different code from the step 9. About the exact code, please refer to the data sheet of the Serial Flash Memory about the mode code.
25. The SW can issue a new set of command.

Prefetch Memory Access in Command Sequencer Mode

After the DDRHSSPI is enabled and before the first access, no prefetch accesses to the Serial Flash Memory shall be performed. Once the first System Bus access is completed and if there are no new System Bus requests, the DDRHSSPI will initiate the continuous prefetch Flash Memory reads, on the Serial Interface until the Prefetch Buffer reaches full.

When there is a memory read access on the System Bus, either of two scenarios in below is expected.

a) Prefetch Buffer miss

In case of a Prefetch Buffer miss, this means that the address requested at the System Bus transaction is not same as the one available on the top of the Prefetch Buffer. This address is also readable at the DDRHSSPIn_CSPREFETCHADDR Register.

In this case, the following sequence occurs:

1. A Prefetch Buffer miss is detected
2. The Prefetch Buffer is flushed
3. A new Command Sequence to the Serial Flash Memory is generated, with a new aligned address

b) Prefetch Buffer hit

In case of a Prefetch Buffer hit, this means that the address requested at the System Bus transaction is same as the one available on the top of the Prefetch Buffer. In this case, the DDRHSSPI can immediately respond with valid data on the System Bus.

In both scenarios, a) and b), when the Prefetch Buffer is full, the SCLK will be halted and the Slave Select will work according to the following conditions.

- DDRHSSPIn_CSCFG.ITIMEREN bit = 1: The Slave Select will be released after the Idle Timer expires.
- DDRHSSPIn_CSCFG.ITIMEREN bit = 0: The Slave Select will remain asserted.

In general, PB keeps its content as long as it is not flushed or popped.

There is difference in terms for PB pop and PB read as below:

- PB pop:
  Access to the PB top word results in reading-and-popping PB (as FIFO).
  The internal read pointer of PB increments.
- PB read:
  Access to the PB top word results in just reading.
  The internal read pointer of PB does not increment.

When in reading PB, if the last byte of a word on the PB top is read, it causes a PB pop, otherwise it causes a PB read. The position of the last byte depends on the access width on the System Bus.

4.3. Address Map of DDRHSSPI

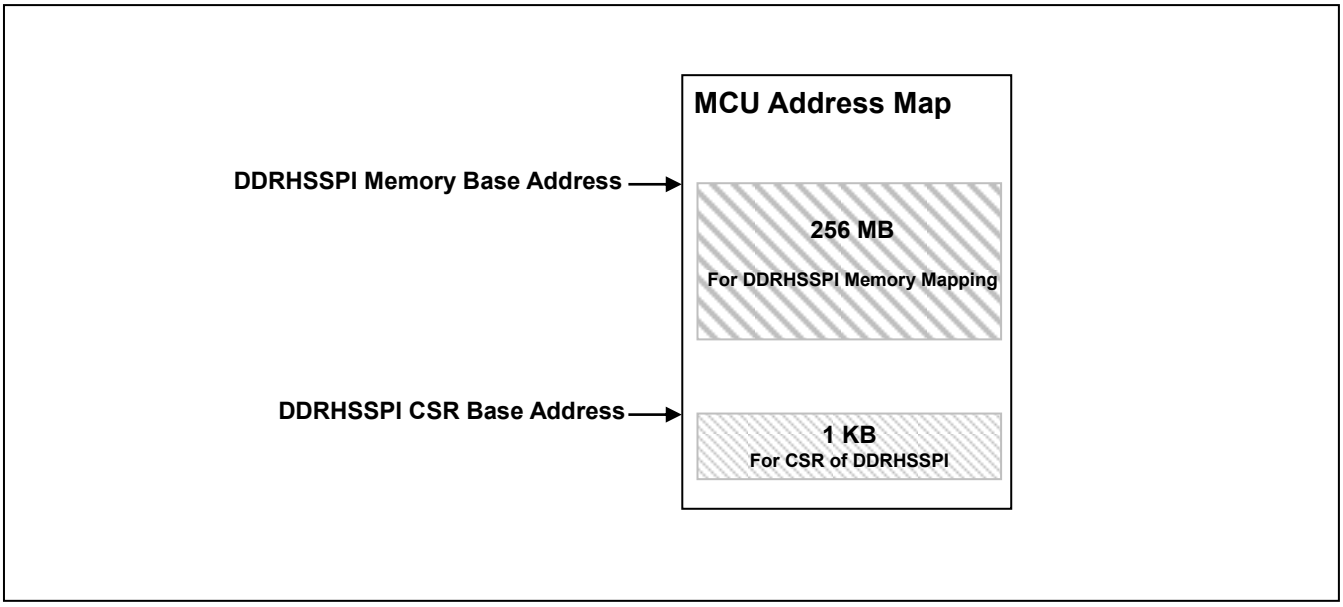
With DDRHSSPI, a memory area of 256MB is allocated in the MCU's address space. This memory area is mapped to the external Serial Flash Memories, using Command Sequencer Mode. An additional 1KB of MCU's address space is mapped to the internal Configuration and Status Registers (i.e. CSRs) of the DDRHSSPI.

The address area allocated to the DDRHSSPI is explained in this section.

Arrangement of DDRHSSPI Address Space in Memory

Figure 4-9 shows the allocation of DDRHSSPI address space in the MCU’s address space.

Figure 4-9 Address Map of DDRHSSPI



4.4. General Use Case Guidelines for DDRHSSPI

This section lists up the guidelines for programming the DDRHSSPI. It is strongly recommended to read these guidelines before programming.

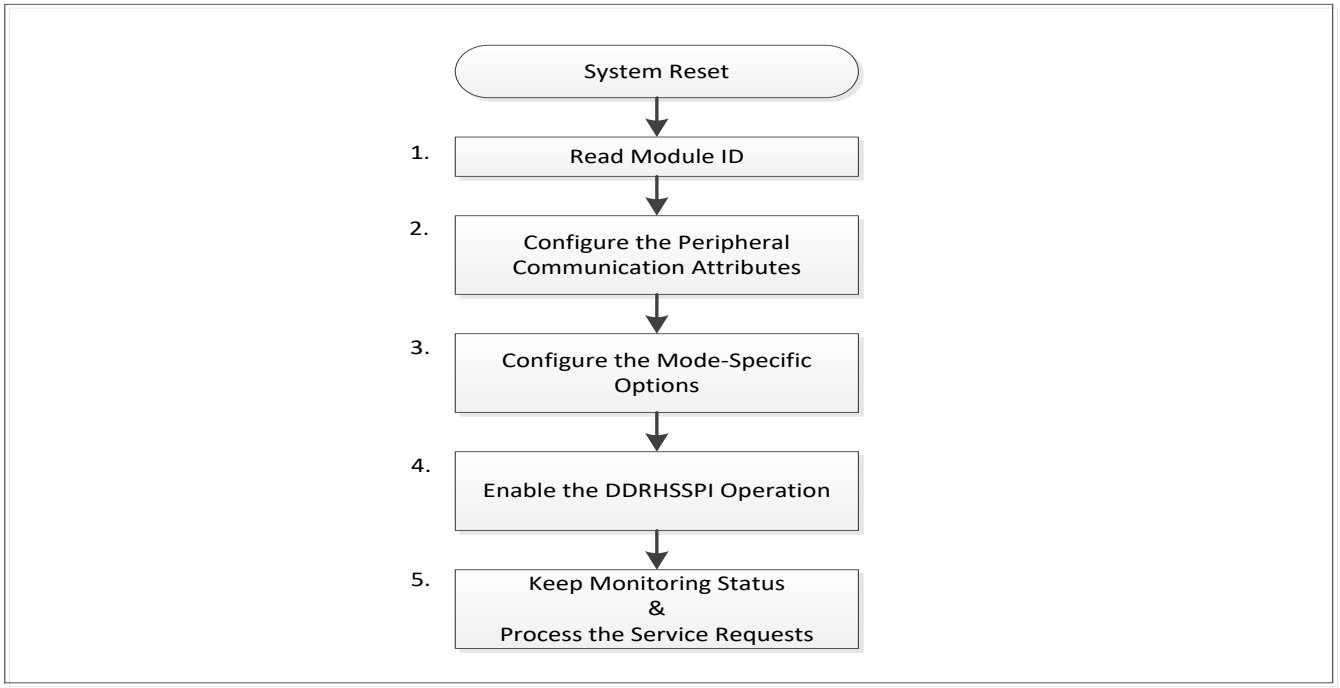
4.4.1. Guidelines on Typical Use Cases of DDRHSSPI

- Any serial-transaction-related parameters and control bits (DDRHSSPIn_DMCFG, DDRHSSPIn_DMBCC, DDRHSSPIn_DMBCS, DDRHSSPIn_DMTRP, DDRHSSPIn_DMPSEL and DDRHSSPIn_DMFIFOCFG) should not be changed while a serial transaction is in progress. Any such changes should be performed only after the current serial transfer has ended (i.e. DDRHSSPIn_TXF.TSSRS= 1 or DDRHSSPIn_RXF.RSSRS= 1). To ensure that the DDRHSSPI has finished all of its transfers, the software can read the DDRHSSPIn_DMFIFOSTATUS.SSACTIVE.
- Direct Mode and Command Sequencer Mode have different configurations of the FIFO, which is the common resource in DDRHSSPI. Hence, when switching from one mode to another, all FIFO data will be lost.
- When the Serial Flash Memory has a command format with changing SPI protocols, DDRHSSPI supports it by Direct Mode. In the Direct Mode, it can transfer the initial set of byte(s) in Legacy Mode using SDR clock, and then transfer the remaining sets of data bytes using the Quad Mode. In such cases, software needs to use DDRHSSPIn_DMFIFOCFG.TXCTRL bit and write TX-FIFO data with additional control such as Data Rate Mode (SDR Mode or DDR Mode) and SPI data width (Legacy Mode, Quad Mode or Octal Mode). Thus, for example, it is possible to transmit the first byte in Legacy Mode with SDR clock, and the following bytes in Quad Mode with DDR clock.
- When Direct Mode is used, the software shall be responsible to take care that the internal FIFOs of DDRHSSPI do not get overrun or underrun. In case the FIFOs get overrun or underrun, the FIFO fill-levels (i.e. DDRHSSPIn_DMFIFOSTATUS.TXFLEVEL and DDRHSSPIn_DMFIFOSTATUS. RXFLEVEL) are no longer pertinent and the software would have to flush the FIFOs.
- The DDRHSSPIn_DMFIFOCFG.RXFLSH and DDRHSSPIn_DMFIFOCFG.TXFLSH bits shall be used by the software to flush the corresponding FIFOs and Shift Registers before using them for any serial transfer. Flushing a FIFO and Shift Registers ensures that they are not pre-loaded with any garbage data (possibly from the previous transfer).
- In Direct Mode with TX-and-RX Mode, whenever the transfer ends, the data from the RX Shift Register is pushed into the RX-FIFO; provided that the RX-FIFO is not full.
    - If the RX-FIFO is already full while a serial transfer is terminating, the serial transfer halts and the RX Shift Register holds the last data as long as DDRHSSPI is halted. As soon as the RX-FIFO is not full anymore, DDRHSSPI comes out of the halt state and the data in the RX Shift Register is pushed into the RX-FIFO before DDRHSSPI releases the Slave Select line. The last data is pushed into the RX-FIFO irrespective of whether the RX Shift Register is filled completely or partially. Thus, in Direct Mode, the received data never remains in the RX Shift Register.
    - In Direct Mode when DDRHSSPI is configure in SDR Mode (when DDRHSSPIn_DMTRP.DDRM bit is set to "0"), it is not allowed to switch to DDR Mode while module is enabled.

4.4.2. Steps in Programming the DDRHSSPI Module

Figure 4-10 gives the general steps a programmer shall follow while using the DDRHSSPI.

Figure 4-10 Programmer's Flowchart: General Steps

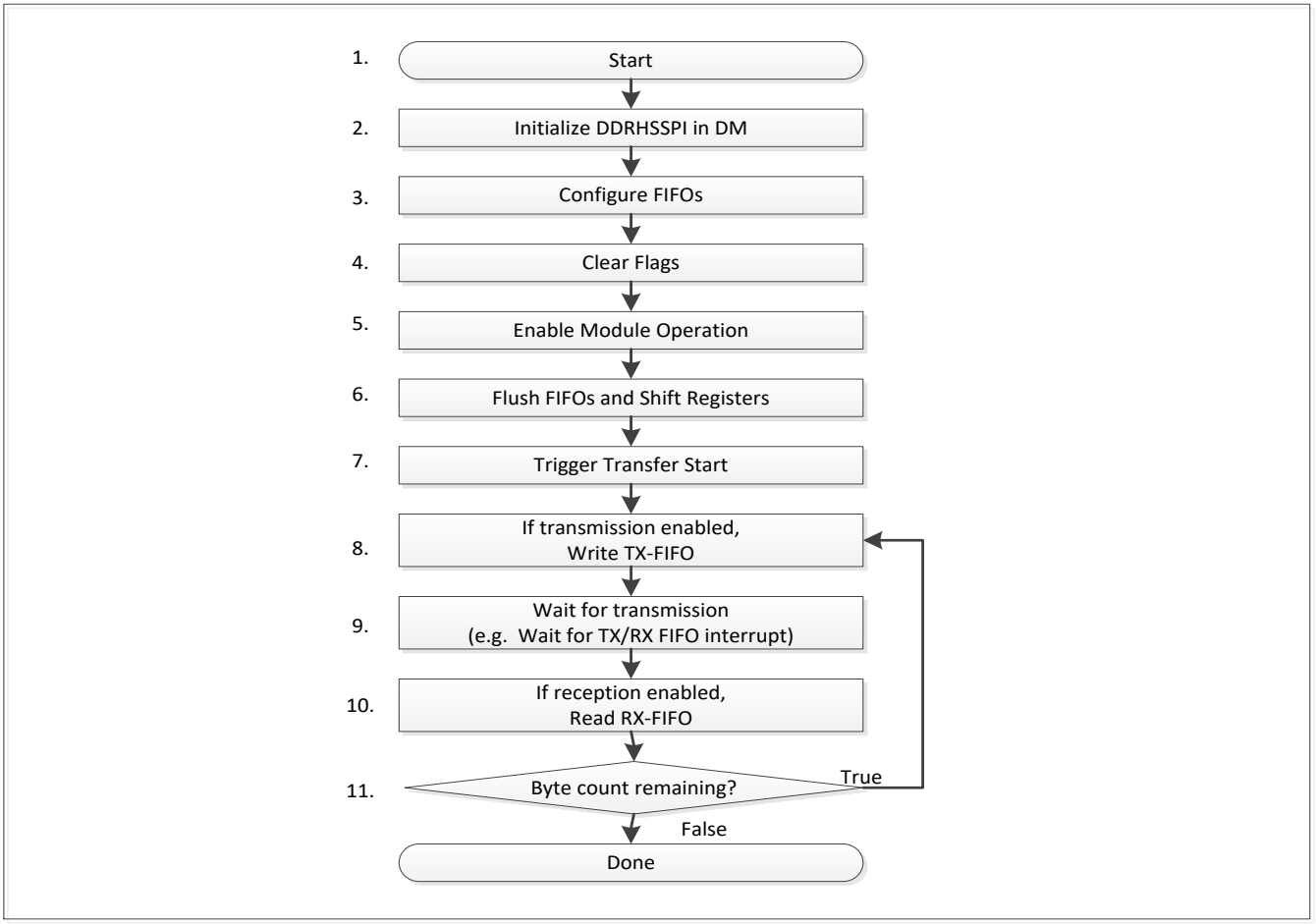


1. After the System Reset, the software shall confirm the Module ID number of DDRHSSPI, by reading the DDRHSSPIn_MID Register. This would help it in identifying the attributes and capabilities supported by the DDRHSSPI.
2. The next step is to configure the Attributes related to the Peripheral Communication with the Serial Flash Memory(Memories) connected with DDRHSSPI. DDRHSSPI can be interfaced with up to 4 Serial Flash Memories at Legacy/Quad mode (8 at Dual Legacy/Dual Quad mode). Serial communication related attributes like Transfer Frequency (i.e. Clock Division Ratio bits), etc. shall be configured in the registers: DDRHSSPIn_PCC0-3. It is very important that these attributes shall be identical among the Serial Flash Memories connected with DDRHSSPI. These configurations shall not be modified while the DDRHSSPI is active. In case the software has to re-program any of these values, the software shall first disable the DDRHSSPI and wait until the current serial transfer is finished.
3. DDRHSSPI can be configured either in Direct Mode or in Command Sequencer Mode, by the DDRHSSPIn_MCTRL.CSEN bit. Depending on which mode is to be used, the software shall configure the mode-specific registers. The registers specific to the Direct Mode are: (DDRHSSPIn_DMCFG, DDRHSSPIn_DMBCC, DDRHSSPIn_DMBCS, DDRHSSPIn_DMTRP, DDRHSSPIn_DMPSEL and DDRHSSPIn_DMFIFOCFG) and the registers specific to the Command Sequencer Mode are: (DDRHSSPIn_RDCSDC0-11, DDRHSSPIn_CSCFG, DDRHSSPIn_CSITIME, DDRHSSPIn_CSAEXT and DDRHSSPIn_CSPBUFFERCFG). When DDRHSSPIn_MID.MID is 0x00000300, there are additional registers (DDRHSSPIn_SDATASAMPLEPT{CNT/LFT/RGH}0-7, DDRHSSPIn_DLP) in Command Sequencer Mode.
4. Only after all module-specific configurations are programmed, the DDRHSSPI can be enabled (by setting the DDRHSSPIn_MCTRL.MEN to "1").
5. Once the DDRHSSPI is enabled, its normal operation begins. The software shall keep monitoring the status of the DDRHSSPI using the various status bits. If the DDRHSSPI is configured for initiating the service requests, it would periodically trigger the service requests (i.e. Interrupts and/or DMA Service Requests). The software would respond to those requests, in order to ensure the normal operation of DDRHSSPI.

4.4.3. Using the DDRHSSPI in Direct Mode of Operation

Figure 4-11 gives the general steps which the SW shall follow while using the DDRHSSPI in Direct Mode.

Figure 4-11 Programmer's Flowchart: DDRHSSPI in Direct Mode of Operation



1. After the System Reset, the software shall initialize the DDRHSSPI by reading the DDRHSSPIn_MID Register and setting the Peripheral Communication related attributes in the DDRHSSPIn_PCC0-3 Registers. Please make sure that the DDRHSSPIn_MCTRL.CSEN bit is cleared to "0".
2. The next step is to configure the transfer protocol (i.e. whether the DDRHSSPI serial transfers use the Legacy or the Quad Protocol and whether the DDRHSSPI would be used only for transmission, or for both transmission and reception) in the DDRHSSPIn_DMTRP.TRP. DDRHSSPI loads the DDRHSSPIn_DMBCC.BCC to know the number of bytes in the serial transfer.
3. Configure the DDRHSSPIn_DMFIFOCFG Register, to set the FIFO threshold levels. By programming these levels, the assertion of the service requests can be controlled. Also configure the DDRHSSPIn_DMFIFOCFG. FWIDTH, to select the width of the FIFOs. Configure the service requests: DDRHSSPI supports both Interrupt Service Request and DMA Service Request, for the normal data read operations from RX-FIFO or write operations to TX-FIFO. For normal operation, either the Interrupt Service Requests or the DMA Service Requests shall be enabled by the software. To enable the Interrupt Service Requests for writing TX-FIFO, please program the bits in the DDRHSSPIn_TXE Register. To enable the Interrupt Service Requests for reading RX-FIFO, please program the bits in the DDRHSSPIn_RXE Register. To enable the DMA Service Request (for writing

and/or reading), please program either/both of the DDRHSSPIn_DMAEN.TXDMAEN and the DDRHSSPIn_DMAEN.RXDMAEN bits. The DMA Read Channel must be setup to perform a block transfer of "DDRHSSPIn_DMFIFOCFG.RXFTH + 1" transfers. The DMA Write Channel must be setup to perform a block transfer of "24 - DDRHSSPIn_DMFIFOCFG.TXFTH" transfers. Select the peripheral (in DDRHSSPIn_DMPSEL.PSEL) on which DDRHSSPI shall initiate the transfer

4. Clear all relevant flags. This finishes the steps in initialization of DDRHSSPI for Direct Mode.
5. Set the DDRHSSPIn_MCTRL.MEN bit, to enable the module.
6. Flush FIFOs to ensure data consistency and avoid any data corruption from previous transfers.
7. When DDRHSSPI is configured, setting the DDRHSSPIn_DMSTART.START bit triggers the start of the serial transaction. Once the serial transaction starts, if the transmit data is available in the TX-FIFO, the DDRHSSPI reads data from TX-FIFO and loads them to the Shift Register. The Shift Register is shifted left and the transmit data is shifted-out onto the Serial Interface. If DDRHSSPI is enabled for Receive operation (in DDRHSSPIn_DMTRP.TRP), the DDRHSSPI receives the serial data with shifting the Shift Register. The received data assembled in the Shift Register is pushed into the RX-FIFO.
8. Write the transmit data to the TX-FIFO via any of DDRHSSPIn_TXFIFO0-23 Registers. Before writing to the DDRHSSPIn_TXFIFO0-23 Register, modify the value of the DDRHSSPIn_DMFIFOCFG. TXCTRL bit appropriately. Generally (i.e. when the write data to the TX-FIFO is to be transmitted as it is), the DDRHSSPIn_DMFIFOCFG.TXCTRL bit shall be "0". Only when in adding some kind of controls such as dummy cycles, the DDRHSSPIn_DMFIFOCFG.TXCTRL bit shall be set to "1". The write access to DDRHSSPIn_TXFIFO0-23 shall be performed after the control of the DDRHSSPIn_DMFIFOCFG.TXCTRL bit.
9. Service Requests are asserted by DDRHSSPI whenever the TX-FIFO level is below the threshold or whenever the DDRHSSPI RX-FIFO level is above the threshold. The software shall write TX-FIFO or read RX-FIFO, to ensure the serial data transfer of DDRHSSPI. After writing or reading the relevant FIFO, the software shall clear the Interrupt Service Requests by writing the DDRHSSPIn_TXC or the DDRHSSPIn_RXC Register. DMA Service Requests are cleared by the handshake responses from the DMA Controller. When the data transfer is not by DMA, the Service Requests can be asserted also by TX-FIFO's empty or RX-FIFO's full status. The software can select the conditions.
10. If reception is enabled in DDRHSSPIn_DMTRP Register (when in TX-and-RX Mode), then the software fetches the received data from the RX-FIFO.
11. Software judges if current serial transfer has finished, by checking (1) the DDRHSSPIn_TXF.TSSRS bit to be "1" or (2)the DDRHSSPIn_DMBCS Register value to be 0x0000. In the normal course of operation, the software usually keeps repeating steps from 8 to 11 until the end of serial transfer.

When the software initiates a new serial transfer again, it starts this flow from step 2.

To switch between the Direct Mode and Command Sequencer Mode, or to re-program any of the parameters that directly affect the serial transfer, the software shall first stop the current transfer and disable the DDRHSSPI (by clearing DDRHSSPIn_MCTRL.MEN bit to "0"). The software can check the status bit DDRHSSPIn_TXF.TSSRS, to see if the current transfer has finished.

4.4.4. Using the DDRHSSPI in Command Sequencer Mode of Operation

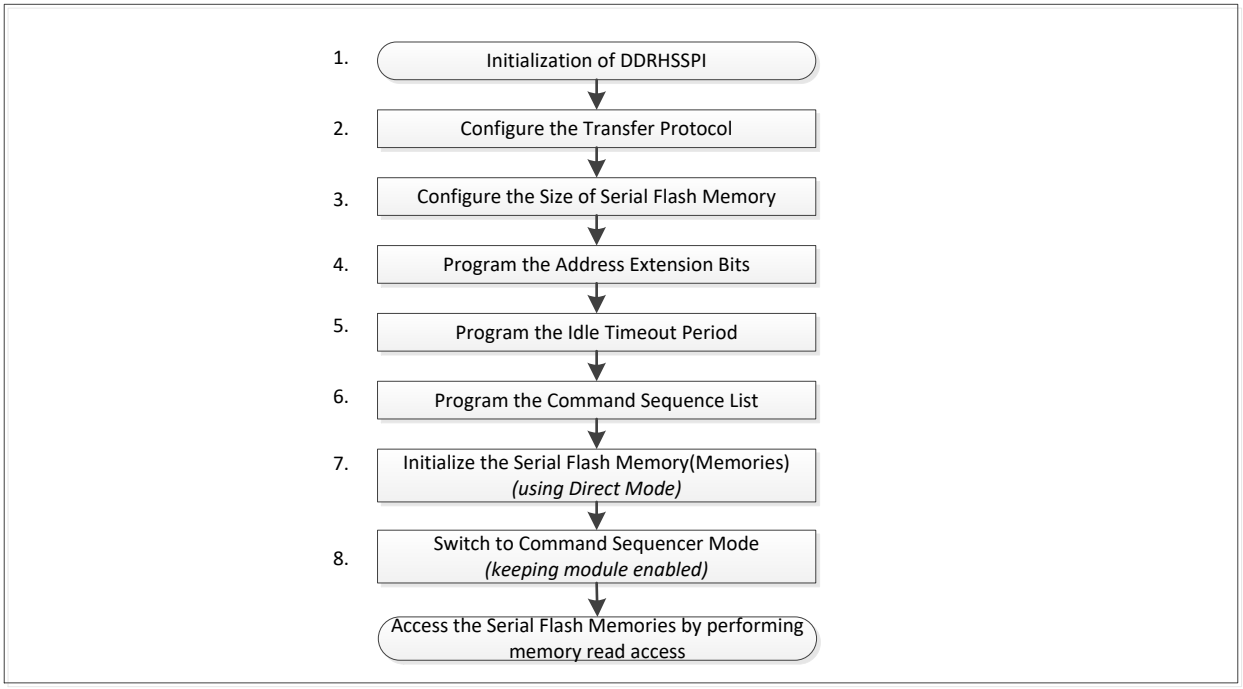
- Using the Memory Mapped Memories

  Following usage rules shall be followed, when interfacing Serial Flash Memories, for memory-mapped accesses in Command Sequencer Mode.

- Usage Rules and Notes
  - In Command Sequencer Mode, all Serial Flash Memories interfaced with DDRHSSPI shall be of same family. Do not mix Serial Flash Memories from different vendor families.
  - If Serial Flash Memories of the same family in different memory sizes are to be interfaced, then while deciding the suitable value of the DDRHSSPIn_CSCFG.MSEL, the Serial Flash Memory with maximum size must be considered. However, it shall be noted here, that the number of bytes from the final memory address that will be transmitted by DDRHSSPI to the Serial Flash Memory is programmed in the Command Sequence lists (in DDRHSSPIn_RDCSDC0-11). Interfacing Serial Flash Memories mixing 32-bit addressing and 24-bit addressing is not allowed in Command Sequencer Mode. This is because the Serial Flash Memory in 24-bit addressing cannot be extended to 32-bit addressing - the address phase in 24-bit addressing is of 3 bytes only.
  - In Command Sequencer Mode, it is impossible to change the SPI width while in the serial transfer. For this reason, for some types of the Serial Flash Memories - the Command Sequencer can be enabled only after the Serial Flash Memory has transited to work in the "Continuous Read Command Sequence". This transition is needed before the first Memory Read access, using the Direct Mode operation of DDRHSSPI.
  - When Command Sequencer Mode is used, internal Prefetch Buffer will be used. Since the Command Sequencer Mode supports only Memory Read accesses, there is only Prefetch Buffer available in Command Sequencer Mode. This buffer is 48 locations deep, 32-bit wide.
  - In Command Sequencer Mode - prefetch of read data is used in order to streamline data reads. This means that during System Bus idle cycles - DDRHSSPI is performing speculative reads from Flash Memory until it gets internal Prefetch Buffer full. There is separate register DDRHSSPIn_CSPREFETCHADDR [31:0] which stores next available prefetch address. In case of address miss-hit on the System Bus, the Prefetch Buffer is flushed automatically.
  - On-the-fly switching of Data Rate Mode (SDR/DDR Mode programmed in DDRHSSPIn_CSCFG.DDRMODE) is not allowed in Command Sequencer Mode while module is enabled.
- Programmer’s Flowchart

  Figure 4-12 shows the general steps how to allocate the Serial Flash Memories onto the address space of MCU.

Figure 4-12 Programmer's Flowchart: Memory Mapping of Serial Flash Memories



1. After the System Reset, the software shall initialize the DDRHSSPI by setting the Peripheral Communication related attributes in the DDRHSSPIn_PCC0-3 Registers. It is very important that these attributes shall be identical among Serial Flash Memories interfaced with the DDRHSSPI. When Serial Flash Memories are to be memory-mapped using Command Sequencer Mode, all Serial Flash Memories shall be of the same family. Therefore, all of DDRHSSPIn_PCC0-3 Registers shall have same configuration values.
2. The next step is to configure the transfer protocol (i.e. whether the DDRHSSPI serial transfers use the Quad or Octal Protocol in the DDRHSSPIn_CSCFG.MBM). The DDRHSSPIn_CSCFG. DDRMODE bit shall be set same as DDRHSSPIn_DMTRP.DDRM bit.
3. Program the DDRHSSPIn_CSCFG.MSEL, according to
   - The address space on the System Bus for mapping the Serial Flash Memory (Memories)
   - The number of Serial Flash Memories interfaced with DDRHSSPI
   Please refer to Section 4.2 for details of the Slave Select.
4. To cover the physical address space in the Serial Flash Memories (up to 16GB), the DDRHSSPIn_CSAEXT Register value gives the upper bits of the address. Please refer to Section 4.2 for details of address generation.
5. The DDRHSSPIn_CSITIME.ITIME helps DDRHSSPI enhance the performance of memory accesses, by continuing previous serial transfer. If DDRHSSPI detects a consecutive memory access during ITIMER period (Slave Select is kept asserted and SCLK is halted), it extends the data transfer without de-asserting current Slave Select. This feature reduces the access time by omitting a new Command Sequence. Program the DDRHSSPIn_CSITIME.ITIME with appropriate idle time-out value.
6. Program the list of Read Command Sequence Registers (i.e. DDRHSSPIn_RDCSDC0-11) with the sequence of the memory read command for the Serial Flash Memory which is interfaced. Please refer to the data sheet of the Serial Flash Memory for details of the Read Command Sequence.

7. The next step is to initialize the Serial Flash Memory that is to be memory mapped. The initialization is specific to the Serial Flash Memory product, including the setting of some control or status bits in its register set. e.g. To use a Serial Flash Memory in a high-performance Quad Mode. Please refer to the data sheet of the Serial Flash Memory to be interfaced. This initialization of the Serial Flash Memory shall be performed using Direct Mode of DDRHSSPI.
8. With this, DDRHSSPI has been configured for accessing the memory-mapped devices (Serial Flash Memories). Switch the DDRHSSPI to Command Sequencer Mode, so that it starts generating the Read Command Sequences on the Serial Interface, by mapping the System Bus accesses to the memory-mapped locations.

Calibration of SDATA Sampling Points

Before running the main process, the software shall adjust the SDATA Sampling Points by setting up registers DDRHSSPIn_SDATASAMPLEPT{LFT/CNT/RGH}0-7. To determine the best sampling points on each bit lane of SDATA[7:0], the software has to run a flow (1)~(3) as below. This flow is an example. The overview of this flow is also illustrated in Figure 4-13.

(1) Calibration sequence

First of all, the software needs to prepare arrays to store the result of Calibration. Each array has a size corresponding to SDATA Sampling Points (=128). It is recommended to prepare 8 arrays to store results for each bit lane of SDATA[7:0].

The software runs the following sequence:

- a) Set DDRHSSPIn_MCTRL.DLPEN bit to "1".
  - The Calibration sequence goes with causing Bus Errors, and these errors have to be ignored by the system. Please make the settings on the system for it.
- b) Set all {R/C/L} of SDATA Sampling Points to 0.
  - The target registers are for current bit lane of SDATA[7:0].
  - E.g. If current bit lane is SDATA[0], write 0x00 to registers DDRHSSPIn_SDATASAMPLEPT{LFT/CNT/RGH}0.
- c) Run a Memory Read access at 32-bits width.
  - From the second round in the loop, the software shall increment the address at least +8.
  - If any of SDATA Sampling Points is not in the valid bit range, the DLP checking results in error and causes a Bus Error.
- d) Check the status of DDRHSSPIn_DLPSAMPLESTATUS register.
  - The software shall check the bit fields for current bit lane of SDATA[7:0], and store the result to the corresponding array.
- e) Clear the DLPFS bit by writing "1" to DDRHSSPIn_FAULTC.DLPFC.
- f) Increment all {R/C/L} of SDATA Sampling Points (+1 on each).
- g) Repeat (c)~(f) until the max value of SDATA Sampling Points (128 rounds).
- h) Repeat (b)~(g) until completing all bit lanes of SDATA[7:0] (8 rounds).
- i) At this moment, DDRHSSPI is expected to be in the status as below:
  - SDATA Sampling Points are the max value.
  - The last Memory Read access has ended in Bus Error.
  - DDRHSSPIn_FAULTF.DLPFS="1"
  - DDRHSSPIn_RXF.DLPERR="1"

- The software holds all arrays of "PASS"/"FAIL" results for each bit lane of SDATA[7:0].

(2) Setting SDATA Sampling Points

Set the registers DDRHSSPIn_SDATASAMPLEPTCNT0-7 to the center positions in respective "PASS" ranges. If there is no "PASS" range in the array, the DDRHSSPI hereafter cannot serve Memory Read access. Else if the "PASS" range is divided by "FAIL(s)", choose the longest "PASS" range and take the center position of it.

DDRHSSPIn_SDATASAMPLEPTLFT0-7 shall be set to less values than DDRHSSPIn_SDATASAMPLEPTCNT0-7 (for each bit lane of SDATA[7:0]).

DDRHSSPIn_SDATASAMPLEPTRGH0-7 shall be set to greater values than DDRHSSPIn_SDATASAMPLEPTCNT0-7 (for each bit lane of SDATA[7:0]).

These register settings for {"LFT", "RGH"} are actually not used in the data reception, and they are just used for detecting the violation of sampling.

The degree of 'less' and 'greater' depends on the window size of "PASS" range, and these differences are used in the re-adjustment of SDATA Sampling Points afterward.

(3) To the main process

The software shall clear DDRHSSPIn_MCTRL.DLPEN bit to "0", if it is not allowed to lose the Memory Read data in the main process by the occurrence of DLP error.

Now the DDRHSSPI is ready for the Memory Read access in the main process. During the main process, the software is required to run a periodical adjustment of SDATA

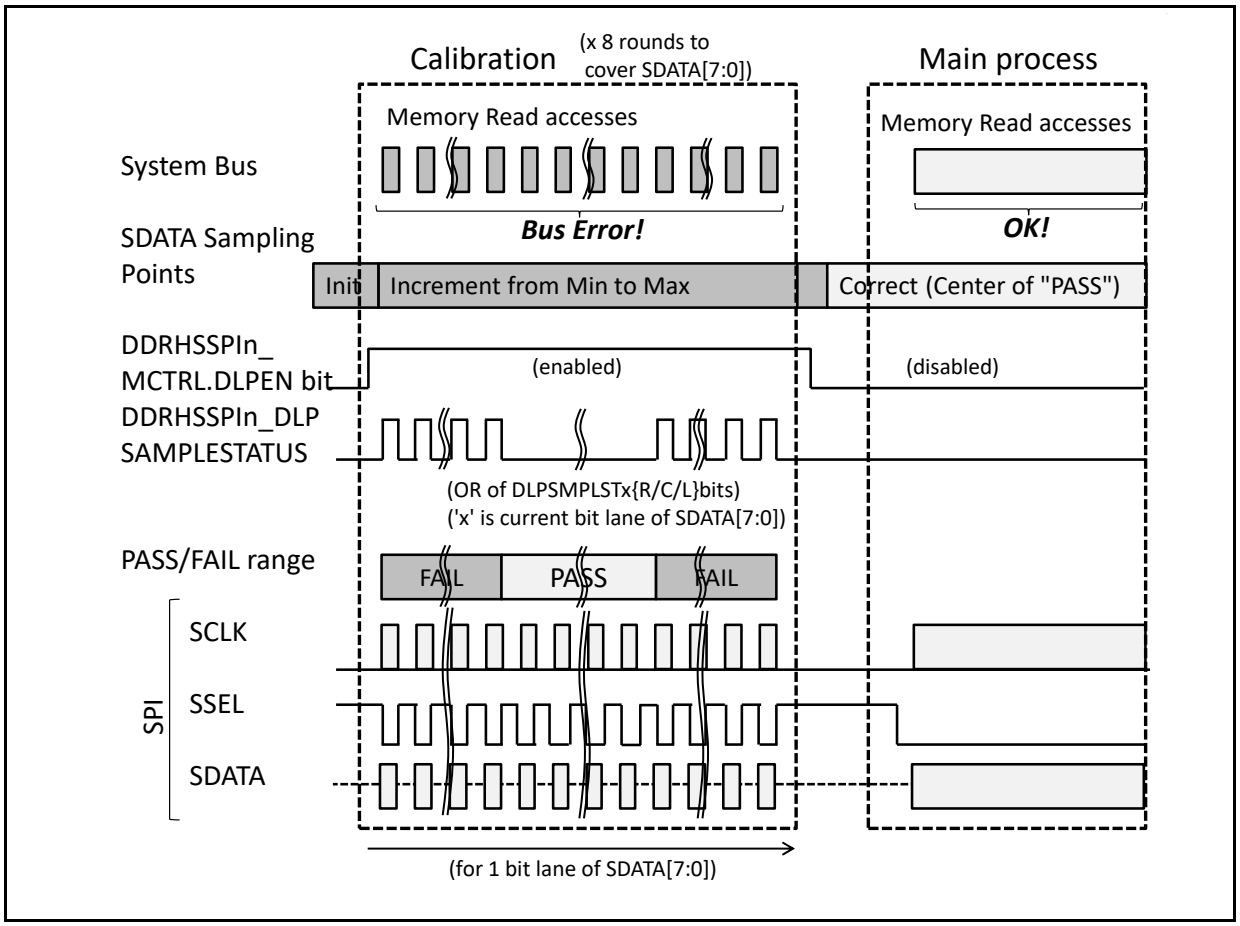
Sampling Points, and this operation is described later in this section.

Notes:

- *If the Serial Flash Memory does not support the DLP function, please change following points.*
  − DDRHSSPIn_MCTRL.DLPEN should be "0".
  − DLP-related status bits are not used.
  − All Serial Flash Memories have to store certain data in certain addresses in advance.These data are used for pattern matching instead of DLP, and shall have bit-toggling pattern on each bit lane of SDATA[7:0] as well as DLP. Please consider the "bit pattern" and "bit alignment" according to the intended SPI protocol (Quad or Dual-Quad).
  − The software has to do pattern-matching checks with the read data, since there is no automatic checking like DLP. Please note that the read data work instead of DLP by the process of software. Therefore, the software shall recognize how many data to read in the DLP-like operation, and gather byte data by re-arranging bits from the read data. [e.g. Memory Read at "DDR & Dual-Quad" protocol] For the pattern matching check, it is assumed to use 8 consecutive bits on each bit lane of SDATA[7:0] as the substitute for DLP. The sum of these pattern-matching bits becomes 64 bits. In this case, the software has to read 2 words (assuming 32-bits access) to check all of these bits.
  − Although the SDATA[7:0] is sampled in parallel, the software has to check the serial bits on each bit lane of SDATA[7:0] individually. The main loop flow is basically same as the case of DLP.
  − The SPI transaction goes on regardless the sampling condition of SDATA[7:0], different from the case using DLP.

The overview of Calibration sequence is shown in the following diagram.

Figure 4-13 Calibration Sequence before the Main Process



Before starting the main process, please read carefully the Notes of 3.4 SPI Data Learning Pattern.

Re-adjustment of SDATA Sampling Points

After setting the best position of SDATA Sampling Points, these points can gradually drift and get close to the edge of SDATA as time passes (e.g. due to the fluctuation of temperature), and the sampling may begin to fail. To avoid the miss-sampling, the software shall periodically adjust the SDATA Sampling Points. The actual interval depends on the system, however, it is expected that the software does not have to run a full Calibration sequence each time.

If the periodical adjustment is frequent enough, each adjustment can change the SDATA Sampling Points +/- 1 [tap] at a time. This operation will not take time compared to the full Calibration, and it is expected that the system can insert this re-adjustment operation in the moment while the main process does not access the Serial Flash Memories. Furthermore, the system can determine the degree of adjustment not only within +/- 1 [tap].

The software runs the following sequence:

a) Set DDRHSSPIn_MCTRL.DLPEN bit to "1".

The handling of Bus Error is same as Calibration. Please make sure that the main process will not access Serial Flash Memories during (a)~(f) in this flow.

b) Run a Memory Read access (at 32-bits width).

The software shall increment the address +8 each time.

c) Check the status of following registers:

DDRHSSPIn_FAULTF.DLPFS bit (optional)
If this bit is "0", go to (f).

DDRHSSPIn_RXF.DLPERR bit (optional)
If this bit is "0", go to (f).

DDRHSSPIn_DLPSAMPLESTATUS register (must)

For each bit lane of SDATA[7:0], the software shall check the status of sampling and update the corresponding SDATA Sampling Point. These judgments are based on the register settings of DDRHSSPIn_SDATASAMPLEPT{LFT/CNT/RGH}0-7 in the Calibration (L < C < R).

- DLPSMPLSTx{R/C/L} = "100": --> Decrease the SDATA Sampling Point.

- DLPSMPLSTx{R/C/L} = "001": --> Increase the SDATA Sampling Point.

- DLPSMPLSTx{R/C/L} = "000": --> No need to change the SDATA Sampling Point.

  If the value of DDRHSSPIn_DLPSAMPLESTATUS register is 0x00000000, go to (f).

- Otherwise: --> Outside the re-adjustable range.

This situation requires a full Calibration and the system shall stop the main process for a moment.

When increasing/decreasing the SDATA Sampling Points, the "CNT" shall be adjusted to the center between "LFT" and "RGH", so as to ensure the PASS range as much as possible. e.g. If DLPSMPLST0{R/C/L} = "100", the software shall decrease only the value of DDRHSSPIn_SDATASAMPLEPTRGH0 registers, then calculate the "CNT" as the average value between "LFT" and "RGH".

d) If DLPFS = "1", write "1" to DDRHSSPIn_FAULTC.DLPFC to clear it.

If DLPERR = "1", write "1" to DDRHSSPIn_RXC.DLPERRC to clear it.

e) Repeat (b)~(d) until DDRHSSPIn_DLPSAMPLESTATUS = 0x00000000.

If the period to run this operation is frequent enough against the fluctuation of SDATA timing, this loop is not always necessary.

f) Clear DDRHSSPIn_MCTRL.DLPEN bit to "0", if it is not allowed to lose the Memory Read data in the main process by the occurrence of DLP error.

g) Main process can access Serial Flash Memories.

Notes:

- *If the Serial Flash Memory does not support the DLP function, the software has to check the result of Memory Read access as well as Calibration sequence (please refer to its Notes).*
- *The system has to ensure the timing for this Re-adjustment operation.*

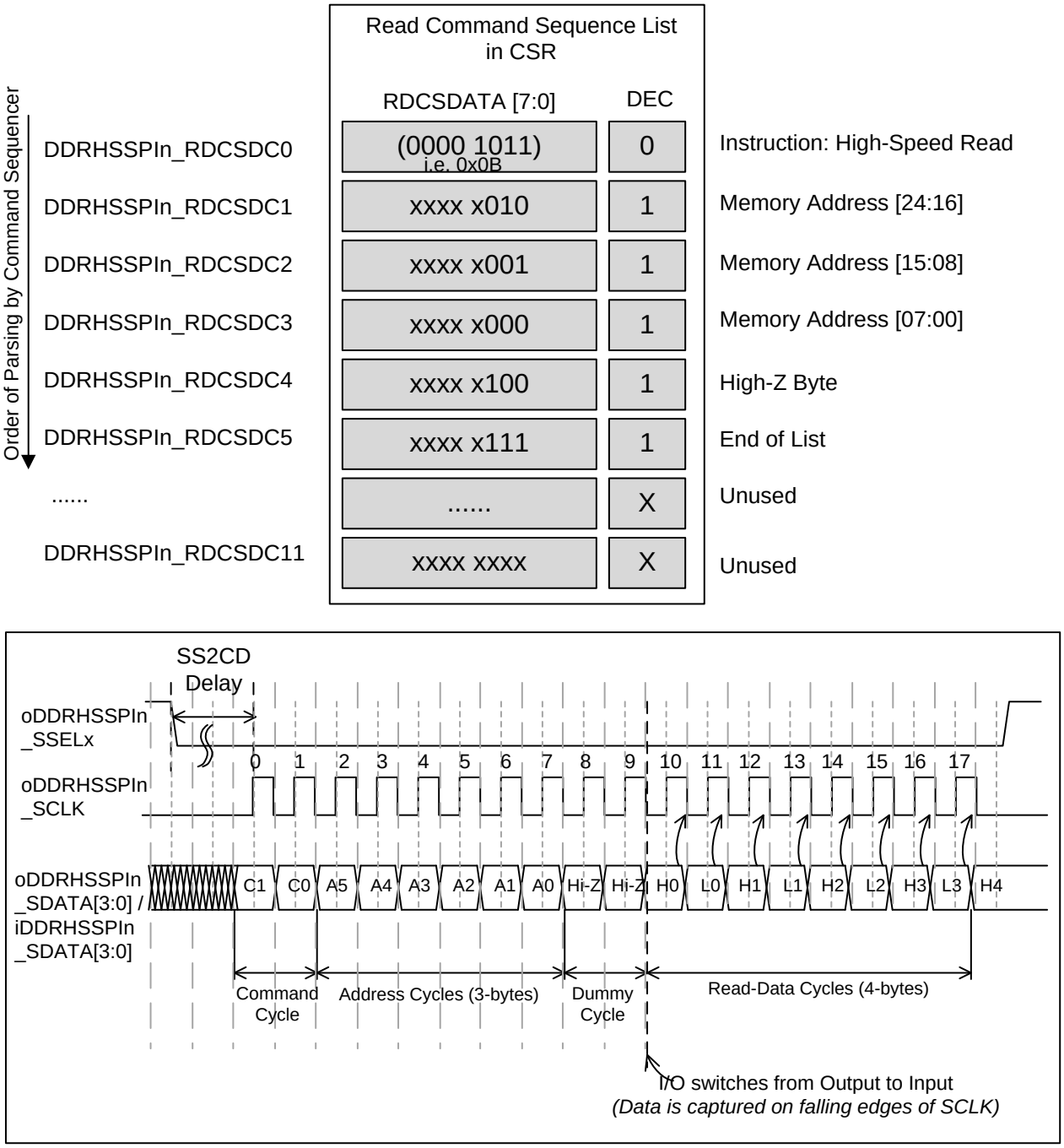
Before starting the main process, please read carefully the Notes of 3.4 SPI Data Learning Pattern.

Timing Diagram for Command Sequencer

Figure 4-14 illustrates with an example, how the Command Sequencer generates the Serial Flash Memory Read Command Sequence. Assuming that the Read Command Sequence list is programmed in DDRHSSPIn_RDCSDC0-11 Registers, as shown in the figure, the Command Sequencer parses the list, starting from DDRHSSPIn_RDCSDC0 Register, and executes the commands as explained in Section 4.2 Command Sequencer Mode.

Figure 4-14 shows also the corresponding timing diagram for a Read Command Sequence, for Mode 4.

Figure 4-14 Read Command Sequence Illustration With Timing Diagram (*Mode4*)



Note:

- *The above example is to illustrate the relation between the Command Sequence List and the serial transmission, and it does not mean the actual example of the SPI transaction for Serial Flash Memory.*

5. Registers

The DDRHSSPI contains various registers to configure its operation, to monitor its status and to read or write the data to be transferred over the Serial Interface.

All registers in DDRHSSPI are explained in this section.

List of DDRHSSPI Registers in Memory

Abbreviation	Register Name	See
DDRHSSPIn_MCTRL	DDRHSSPI Module Control Register	5.1
DDRHSSPIn_PCC0	DDRHSSPI Peripheral Communication Configuration Register 0	5.2
DDRHSSPIn_PCC1	DDRHSSPI Peripheral Communication Configuration Register 1	
DDRHSSPIn_PCC2	DDRHSSPI Peripheral Communication Configuration Register 2	
DDRHSSPIn_PCC3	DDRHSSPI Peripheral Communication Configuration Register 3	
DDRHSSPIn_TXF	DDRHSSPI TX Interrupt Flag Register	5.3
DDRHSSPIn_TXE	DDRHSSPI TX Interrupt Enable Register	5.4
DDRHSSPIn_TXC	DDRHSSPI TX Interrupt Clear Register	5.5
DDRHSSPIn_RXF	DDRHSSPI RX Interrupt Flag Register	5.6
DDRHSSPIn_RXE	DDRHSSPI RX Interrupt Enable Register	5.7
DDRHSSPIn_RXC	DDRHSSPI RX Interrupt Clear Register	5.8
DDRHSSPIn_FAULTF	DDRHSSPI Fault Status Flag Register	5.9
DDRHSSPIn_FAULTC	DDRHSSPI Fault Status Clear Register	5.10
DDRHSSPIn_DMCFG	DDRHSSPI Direct Mode Configuration Register	5.11
DDRHSSPIn_DMAEN	DDRHSSPI DMA Enable Register	5.12
DDRHSSPIn_DMSTART	DDRHSSPI Direct Mode Start Register	5.13
DDRHSSPIn_DMPSEL	DDRHSSPI Direct Mode Peripheral Select Register	5.14
DDRHSSPIn_DMTRP	DDRHSSPI Direct Mode Transfer Protocol Register	5.15
DDRHSSPIn_DMBCC	DDRHSSPI Byte Count Control Register	5.16
DDRHSSPIn_DMBCS	DDRHSSPI Byte Count Status Register	5.17
DDRHSSPIn_DMFIFOSTATUS	DDRHSSPI Direct Mode FIFO Status Register	5.18
DDRHSSPIn_DMFIFOCFG	DDRHSSPI Direct Mode FIFO Configuration Register	5.19

Abbreviation	Register Name	See
DDRHSSPIn_TXFIFO0	DDRHSSPI TX-FIFO Register 0	5.20
DDRHSSPIn_TXFIFO1	DDRHSSPI TX-FIFO Register 1	
DDRHSSPIn_TXFIFO2	DDRHSSPI TX-FIFO Register 2	
DDRHSSPIn_TXFIFO3	DDRHSSPI TX-FIFO Register 3	
DDRHSSPIn_TXFIFO4	DDRHSSPI TX-FIFO Register 4	
DDRHSSPIn_TXFIFO5	DDRHSSPI TX-FIFO Register 5	
DDRHSSPIn_TXFIFO6	DDRHSSPI TX-FIFO Register 6	
DDRHSSPIn_TXFIFO7	DDRHSSPI TX-FIFO Register 7	
DDRHSSPIn_TXFIFO8	DDRHSSPI TX-FIFO Register 8	
DDRHSSPIn_TXFIFO9	DDRHSSPI TX-FIFO Register 9	
DDRHSSPIn_TXFIFO10	DDRHSSPI TX-FIFO Register 10	
DDRHSSPIn_TXFIFO11	DDRHSSPI TX-FIFO Register 11	
DDRHSSPIn_TXFIFO12	DDRHSSPI TX-FIFO Register 12	
DDRHSSPIn_TXFIFO13	DDRHSSPI TX-FIFO Register 13	
DDRHSSPIn_TXFIFO14	DDRHSSPI TX-FIFO Register 14	
DDRHSSPIn_TXFIFO15	DDRHSSPI TX-FIFO Register 15	
DDRHSSPIn_TXFIFO16	DDRHSSPI TX-FIFO Register 16	
DDRHSSPIn_TXFIFO17	DDRHSSPI TX-FIFO Register 17	
DDRHSSPIn_TXFIFO18	DDRHSSPI TX-FIFO Register 18	
DDRHSSPIn_TXFIFO19	DDRHSSPI TX-FIFO Register 19	
DDRHSSPIn_TXFIFO20	DDRHSSPI TX-FIFO Register 20	
DDRHSSPIn_TXFIFO21	DDRHSSPI TX-FIFO Register 21	
DDRHSSPIn_TXFIFO22	DDRHSSPI TX-FIFO Register 22	
DDRHSSPIn_TXFIFO23	DDRHSSPI TX-FIFO Register 23	

Abbreviation	Register Name	See
DDRHSSPIn_RXFIFO0	DDRHSSPI RX-FIFO Register 0	5.21
DDRHSSPIn_RXFIFO1	DDRHSSPI RX-FIFO Register 1	
DDRHSSPIn_RXFIFO2	DDRHSSPI RX-FIFO Register 2	
DDRHSSPIn_RXFIFO3	DDRHSSPI RX-FIFO Register 3	
DDRHSSPIn_RXFIFO4	DDRHSSPI RX-FIFO Register 4	
DDRHSSPIn_RXFIFO5	DDRHSSPI RX-FIFO Register 5	
DDRHSSPIn_RXFIFO6	DDRHSSPI RX-FIFO Register 6	
DDRHSSPIn_RXFIFO7	DDRHSSPI RX-FIFO Register 7	
DDRHSSPIn_RXFIFO8	DDRHSSPI RX-FIFO Register 8	
DDRHSSPIn_RXFIFO9	DDRHSSPI RX-FIFO Register 9	
DDRHSSPIn_RXFIFO10	DDRHSSPI RX-FIFO Register 10	
DDRHSSPIn_RXFIFO11	DDRHSSPI RX-FIFO Register 11	
DDRHSSPIn_RXFIFO12	DDRHSSPI RX-FIFO Register 12	
DDRHSSPIn_RXFIFO13	DDRHSSPI RX-FIFO Register 13	
DDRHSSPIn_RXFIFO14	DDRHSSPI RX-FIFO Register 14	
DDRHSSPIn_RXFIFO15	DDRHSSPI RX-FIFO Register 15	
DDRHSSPIn_RXFIFO16	DDRHSSPI RX-FIFO Register 16	
DDRHSSPIn_RXFIFO17	DDRHSSPI RX-FIFO Register 17	
DDRHSSPIn_RXFIFO18	DDRHSSPI RX-FIFO Register 18	
DDRHSSPIn_RXFIFO19	DDRHSSPI RX-FIFO Register 19	
DDRHSSPIn_RXFIFO20	DDRHSSPI RX-FIFO Register 20	
DDRHSSPIn_RXFIFO21	DDRHSSPI RX-FIFO Register 21	
DDRHSSPIn_RXFIFO22	DDRHSSPI RX-FIFO Register 22	
DDRHSSPIn_RXFIFO23	DDRHSSPI RX-FIFO Register 23	
DDRHSSPIn_RDCSDC0	DDRHSSPI Read Command Sequencer Data/Control Register 0	5.22
DDRHSSPIn_RDCSDC1	DDRHSSPI Read Command Sequencer Data/Control Register 1	
DDRHSSPIn_RDCSDC2	DDRHSSPI Read Command Sequencer Data/Control Register 2	
DDRHSSPIn_RDCSDC3	DDRHSSPI Read Command Sequencer Data/Control Register 3	
DDRHSSPIn_RDCSDC4	DDRHSSPI Read Command Sequencer Data/Control Register 4	
DDRHSSPIn_RDCSDC5	DDRHSSPI Read Command Sequencer Data/Control Register 5	
DDRHSSPIn_RDCSDC6	DDRHSSPI Read Command Sequencer Data/Control Register 6	
DDRHSSPIn_RDCSDC7	DDRHSSPI Read Command Sequencer Data/Control Register 7	
DDRHSSPIn_RDCSDC8	DDRHSSPI Read Command Sequencer Data/Control Register 8	
DDRHSSPIn_RDCSDC9	DDRHSSPI Read Command Sequencer Data/Control Register 9	
DDRHSSPIn_RDCSDC10	DDRHSSPI Read Command Sequencer Data/Control Register 10	
DDRHSSPIn_RDCSDC11	DDRHSSPI Read Command Sequencer Data/Control Register 11	

Abbreviation	Register Name	See
DDRHSSPIn_MID	DDRHSSPI Module ID Register	5.23
DDRHSSPIn_CSPREFETCHADDR	DDRHSSPI Command Sequencer Prefetch Address Register	5.24
DDRHSSPIn_SDATASAMPLEPTCNT0	DDRHSSPI SDATA Center Clock Sample Point Register 0	5.25
DDRHSSPIn_SDATASAMPLEPTCNT1	DDRHSSPI SDATA Center Clock Sample Point Register 1	
DDRHSSPIn_SDATASAMPLEPTCNT2	DDRHSSPI SDATA Center Clock Sample Point Register 2	
DDRHSSPIn_SDATASAMPLEPTCNT3	DDRHSSPI SDATA Center Clock Sample Point Register 3	
DDRHSSPIn_SDATASAMPLEPTCNT4	DDRHSSPI SDATA Center Clock Sample Point Register 4	
DDRHSSPIn_SDATASAMPLEPTCNT5	DDRHSSPI SDATA Center Clock Sample Point Register 5	
DDRHSSPIn_SDATASAMPLEPTCNT6	DDRHSSPI SDATA Center Clock Sample Point Register 6	
DDRHSSPIn_SDATASAMPLEPTCNT7	DDRHSSPI SDATA Center Clock Sample Point Register 7	
DDRHSSPIn_SDATASAMPLEPTLFT0	DDRHSSPI SDATA Left Clock Sample Point Register 0	5.26
DDRHSSPIn_SDATASAMPLEPTLFT1	DDRHSSPI SDATA Left Clock Sample Point Register 1	
DDRHSSPIn_SDATASAMPLEPTLFT2	DDRHSSPI SDATA Left Clock Sample Point Register 2	
DDRHSSPIn_SDATASAMPLEPTLFT3	DDRHSSPI SDATA Left Clock Sample Point Register 3	
DDRHSSPIn_SDATASAMPLEPTLFT4	DDRHSSPI SDATA Left Clock Sample Point Register 4	
DDRHSSPIn_SDATASAMPLEPTLFT5	DDRHSSPI SDATA Left Clock Sample Point Register 5	
DDRHSSPIn_SDATASAMPLEPTLFT6	DDRHSSPI SDATA Left Clock Sample Point Register 6	
DDRHSSPIn_SDATASAMPLEPTLFT7	DDRHSSPI SDATA Left Clock Sample Point Register 7	
DDRHSSPIn_SDATASAMPLEPTRGH0	DDRHSSPI SDATA Right Clock Sample Point Register 0	5.27
DDRHSSPIn_SDATASAMPLEPTRGH1	DDRHSSPI SDATA Right Clock Sample Point Register 1	
DDRHSSPIn_SDATASAMPLEPTRGH2	DDRHSSPI SDATA Right Clock Sample Point Register 2	
DDRHSSPIn_SDATASAMPLEPTRGH3	DDRHSSPI SDATA Right Clock Sample Point Register 3	
DDRHSSPIn_SDATASAMPLEPTRGH4	DDRHSSPI SDATA Right Clock Sample Point Register 4	
DDRHSSPIn_SDATASAMPLEPTRGH5	DDRHSSPI SDATA Right Clock Sample Point Register 5	
DDRHSSPIn_SDATASAMPLEPTRGH6	DDRHSSPI SDATA Right Clock Sample Point Register 6	
DDRHSSPIn_SDATASAMPLEPTRGH7	DDRHSSPI SDATA Right Clock Sample Point Register 7	
DDRHSSPIn_DLP	DDRHSSPI Data Learning Pattern Register	5.28
DDRHSSPIn_DLPSAMPLESTATUS	DDRHSSPI Data Learning Pattern Sample Status Register	5.29
DDRHSSPIn_CSCFG	DDRHSSPI Command Sequencer Configuration Register	5.30
DDRHSSPIn_CSITIME	DDRHSSPI Command Sequencer Idle Time Register	5.31
DDRHSSPIn_CSAEXT	DDRHSSPI Command Sequencer Address Extension Register	5.32
DDRHSSPIn_CSPBUFFERCFG	DDRHSSPI Command Sequencer Prefetch Buffer Configuration Register	5.33
DDRHSSPIn_CSPBUFFERSTATUS	DDRHSSPI Command Sequencer Prefetch Buffer Status Register	5.34

Table 5-1 DDRHSSPI Register Memory Map

Offset	Register Name / Initial Value			
	+3	+2	+1	+0
0x0000_0000	DDRHSSPIn_MCTRL 00000000_00000000_00000000_00000000			
0x0000_0004	DDRHSSPIn_PCC0 0000000_ 00000000_00000000_00000000			

Offset	Register Name / Initial Value			
	+3	+2	+1	+0
0x0000_0008	DDRHSSPIn_PCC1 00000000_00000000_00000000_00000000			
0x0000_000C	DDRHSSPIn_PCC2 00000000_00000000_00000000_00000000			
0x0000_0010	DDRHSSPIn_PCC3 00000000_00000000_00000000_00000000			
0x0000_0014	DDRHSSPIn_TXF 00000000_00000000_00000000_00000000			
0x0000_0018	DDRHSSPIn_TXE 00000000_00000000_00000000_00000000			
0x0000_001C	DDRHSSPIn_TXC 00000000_00000000_00000000_00000000			
0x0000_0020	DDRHSSPIn_RXF 00000000_00000000_00000000_00000000			
0x0000_0024	DDRHSSPIn_RXE 00000000_00000000_00000000_00000000			
0x0000_0028	DDRHSSPIn_RXC 00000000_00000000_00000000_00000000			
0x0000_002C	DDRHSSPIn_FAULTF 00000000_00000000_00000000_00000000			
0x0000_0030	DDRHSSPIn_FAULTC 00000000_00000000_00000000_00000000			
0x0000_0034	-	-	DDRHSSPIn_DMAEN 00000000	DDRHSSPIn_DMCFG 00000000
0x0000_0038	DDRHSSPIn_DMTRP 00000000	DDRHSSPIn_DMPSEL 00000000	-	DDRHSSPIn_DMSTART 00000000
0x0000_003C	DDRHSSPIn_DMBCS 00000000_00000000		DDRHSSPIn_DMBCC 00000000_00000000	
0x0000_0040	DDRHSSPIn_DMFIFOSTATUS 00000000_00000000_00000000_00000000			
0x0000_0044	DDRHSSPIn_DMFIFOCFG 00000000_00000000_00001000_00001111			
0x0000_0048	DDRHSSPIn_TXFIFO0 00000000_00000000_00000000_00000000			
0x0000_004C	DDRHSSPIn_TXFIFO1 00000000_00000000_00000000_00000000			
0x0000_0050	DDRHSSPIn_TXFIFO2 00000000_00000000_00000000_00000000			
0x0000_0054	DDRHSSPIn_TXFIFO3 00000000_00000000_00000000_00000000			
0x0000_0058	DDRHSSPIn_TXFIFO4 00000000_00000000_00000000_00000000			
0x0000_005C	DDRHSSPIn_TXFIFO5 00000000_00000000_00000000_00000000			

Offset	Register Name / Initial Value			
	+3	+2	+1	+0
0x0000_0060	DDRHSSPIn_TXFIFO6 00000000_00000000_00000000_00000000			
0x0000_0064	DDRHSSPIn_TXFIFO7 00000000_00000000_00000000_00000000			
0x0000_0068	DDRHSSPIn_TXFIFO8 00000000_00000000_00000000_00000000			
0x0000_006C	DDRHSSPIn_TXFIFO9 00000000_00000000_00000000_00000000			
0x0000_0070	DDRHSSPIn_TXFIFO10 00000000_00000000_00000000_00000000			
0x0000_0074	DDRHSSPIn_TXFIFO11 00000000_00000000_00000000_00000000			
0x0000_0078	DDRHSSPIn_TXFIFO12 00000000_00000000_00000000_00000000			
0x0000_007C	DDRHSSPIn_TXFIFO13 00000000_00000000_00000000_00000000			
0x0000_0080	DDRHSSPIn_TXFIFO14 00000000_00000000_00000000_00000000			
0x0000_0084	DDRHSSPIn_TXFIFO15 00000000_00000000_00000000_00000000			
0x0000_0088	DDRHSSPIn_TXFIFO16 00000000_00000000_00000000_00000000			
0x0000_008C	DDRHSSPIn_TXFIFO17 00000000_00000000_00000000_00000000			
0x0000_0090	DDRHSSPIn_TXFIFO18 00000000_00000000_00000000_00000000			
0x0000_0094	DDRHSSPIn_TXFIFO19 00000000_00000000_00000000_00000000			
0x0000_0098	DDRHSSPIn_TXFIFO20 00000000_00000000_00000000_00000000			
0x0000_009C	DDRHSSPIn_TXFIFO21 00000000_00000000_00000000_00000000			
0x0000_00A0	DDRHSSPIn_TXFIFO22 00000000_00000000_00000000_00000000			
0x0000_00A4	DDRHSSPIn_TXFIFO23 00000000_00000000_00000000_00000000			
0x0000_00A8	DDRHSSPIn_RXFIFO0 00000000_00000000_00000000_00000000			
0x0000_00AC	DDRHSSPIn_RXFIFO1 00000000_00000000_00000000_00000000			
0x0000_00B0	DDRHSSPIn_RXFIFO2 00000000_00000000_00000000_00000000			
0x0000_00B4	DDRHSSPIn_RXFIFO3 00000000_00000000_00000000_00000000			

Offset	Register Name / Initial Value			
	+3	+2	+1	+0
0x0000_00B8	DDRHSSPIn_RXFIFO4 00000000_00000000_00000000_00000000			
0x0000_00BC	DDRHSSPIn_RXFIFO5 00000000_00000000_00000000_00000000			
0x0000_00C0	DDRHSSPIn_RXFIFO6 00000000_00000000_00000000_00000000			
0x0000_00C4	DDRHSSPIn_RXFIFO7 00000000_00000000_00000000_00000000			
0x0000_00C8	DDRHSSPIn_RXFIFO8 00000000_00000000_00000000_00000000			
0x0000_00CC	DDRHSSPIn_RXFIFO9 00000000_00000000_00000000_00000000			
0x0000_00D0	DDRHSSPIn_RXFIFO10 00000000_00000000_00000000_00000000			
0x0000_00D4	DDRHSSPIn_RXFIFO11 00000000_00000000_00000000_00000000			
0x0000_00D8	DDRHSSPIn_RXFIFO12 00000000_00000000_00000000_00000000			
0x0000_00DC	DDRHSSPIn_RXFIFO13 00000000_00000000_00000000_00000000			
0x0000_00E0	DDRHSSPIn_RXFIFO14 00000000_00000000_00000000_00000000			
0x0000_00E4	DDRHSSPIn_RXFIFO15 00000000_00000000_00000000_00000000			
0x0000_00E8	DDRHSSPIn_RXFIFO16 00000000_00000000_00000000_00000000			
0x0000_00EC	DDRHSSPIn_RXFIFO17 00000000_00000000_00000000_00000000			
0x0000_00F0	DDRHSSPIn_RXFIFO18 00000000_00000000_00000000_00000000			
0x0000_00F4	DDRHSSPIn_RXFIFO19 00000000_00000000_00000000_00000000			
0x0000_00F8	DDRHSSPIn_RXFIFO20 00000000_00000000_00000000_00000000			
0x0000_00FC	DDRHSSPIn_RXFIFO21 00000000_00000000_00000000_00000000			
0x0000_0100	DDRHSSPIn_RXFIFO22 00000000_00000000_00000000_00000000			
0x0000_0104	DDRHSSPIn_RXFIFO23 00000000_00000000_00000000_00000000			
0x0000_0108	DDRHSSPIn_RDCSDC1 00000000_00000000		DDRHSSPIn_RDCSDC0 00000000_00000000	
0x0000_010C	DDRHSSPIn_RDCSDC3 00000000_00000000		DDRHSSPIn_RDCSDC2 00000000_00000000	

Offset	Register Name / Initial Value			
	+3	+2	+1	+0
0x0000_0110	DDRHSSPIn_RDCSDC5 00000000_00000000		DDRHSSPIn_RDCSDC4 00000000_00000000	
0x0000_0114	DDRHSSPIn_RDCSDC7 00000000_00000000		DDRHSSPIn_RDCSDC6 00000000_00000000	
0x0000_0118	DDRHSSPIn_RDCSDC9 00000000_00000000		DDRHSSPIn_RDCSDC8 00000000_00000000	
0x0000_011C	DDRHSSPIn_RDCSDC11 00000000_00000000		DDRHSSPIn_RDCSDC10 00000000_00000000	
0x0000_0120	DDRHSSPIn_MID 00000000_00000000_000000**_0000000*			
0x0000_0124	DDRHSSPIn_CSPREFETCHADDR 00000000_00000000_00000000_00000000			
0x0000_0128	DDRHSSPIn_SDATASA MPLEPTCNT3 00000000	DDRHSSPIn_SDATASA MPLEPTCNT2 00000000	DDRHSSPIn_SDATASA MPLEPTCNT1 00000000	DDRHSSPIn_SDATASA MPLEPTCNT0 00000000
0x0000_012C	DDRHSSPIn_SDATASA MPLEPTCNT7 00000000	DDRHSSPIn_SDATASA MPLEPTCNT6 00000000	DDRHSSPIn_SDATASA MPLEPTCNT5 00000000	DDRHSSPIn_SDATASA MPLEPTCNT4 00000000
0x0000_0130	DDRHSSPIn_SDATASA MPLEPTLFT3 00000000	DDRHSSPIn_SDATASA MPLEPTLFT2 00000000	DDRHSSPIn_SDATASA MPLEPTLFT1 00000000	DDRHSSPIn_SDATASA MPLEPTLFT0 00000000
0x0000_0134	DDRHSSPIn_SDATASA MPLEPTLFT7 00000000	DDRHSSPIn_SDATASA MPLEPTLFT6 00000000	DDRHSSPIn_SDATASA MPLEPTLFT5 00000000	DDRHSSPIn_SDATASA MPLEPTLFT4 00000000
0x0000_0138	DDRHSSPIn_SDATASA MPLEPTRGH3 00000000	DDRHSSPIn_SDATASA MPLEPTRGH2 00000000	DDRHSSPIn_SDATASA MPLEPTRGH1 00000000	DDRHSSPIn_SDATASA MPLEPTRGH0 00000000
0x0000_013C	DDRHSSPIn_SDATASA MPLEPTRGH7 00000000	DDRHSSPIn_SDATASA MPLEPTRGH6 00000000	DDRHSSPIn_SDATASA MPLEPTRGH5 00000000	DDRHSSPIn_SDATASA MPLEPTRGH4 00000000
0x0000_0140	-			
0x0000_0144	DDRHSSPIn_DLP 00000000_00000000_00000000_00000000			
0x0000_0148	DDRHSSPIn_DLPSAMPLESTATUS 00000000_00000000_00000000_00000000			
0x0000_014C	DDRHSSPIn_CSCFG 00000000_00000000_00000000_00000000			
0x0000_0150	DDRHSSPIn_CSITIME 00000000_00000000_11111111_11111111			
0x0000_0154	DDRHSSPIn_CSAEXT 00000000_00000000_00000000_00000000			
0x0000_0158	DDRHSSPIn_CSPBUFFERCFG 00000000_00000000_00000000_00001111			
0x0000_015C	DDRHSSPIn_CSPBUFFERSTATUS 00000000_00000000_00000000_00000000			

Note:

- *The initial register value after reset indicates as follows:*
  - "1": Initial value "1"
  - "0": Initial value "0"
  - “X”: Initial value undefined
  - “-“: Reserved bit/Undefined bit
  - “*”: Initial value "0" or "1" according to the setting

5.1. DDRHSSPI Module Control Register (DDRHSSPIn_MCTRL)

The DDRHSSPI Module Control Register controls the DDRHSSPI. It contains vital bits like the Module Enable bit, the Command Sequencer Mode Enable bit and Data Learning Pattern Enable bit.

The Software can enable/disable the DDRHSSPI operation by using this register.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	Reserved	DLPEN	MES	Reserved	Reserved	CSEN	MEN
ACCESS_TYPE	R0,W0	R0,W0	R/W	R,WX	R0,W0	R/W0	R/W	R/W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:6] Reserved: Reserved

[bit5] DLPEN: Data Learning Pattern Enable

This bit enables when the sampling point of receiving data is checked to use the command with DLP.

This bit shall be fixed to "0" when DDRHSSPIn_MID.MID is 0x00000001.

This bit is valid in DDR Mode of the Command Sequencer Mode.

Bit	Description
0	DLP is disabled.
1	DLP is enabled.

Notes:

- *Under the condition of DLPEN="1", any DLP error in the DDRHSSPIn_DLPSAMPLESTATUS register results in the system bus error, and this is common among {LFT/CNT/RGH}. In that case, the Memory Read data is invalid.*
- *Even if the CNT is sampled correctly, any error on the LFT or RGH leads to the error.*

[bit4] MES: Module Enable Status

This bit shows the status of the DDRHSSPI.

Bit	Description
0	DDRHSSPI is completely disabled and it has entered the power saving mode.
1	DDRHSSPI is enabled.

[bit3:2] Reserved: Reserved

[bit1] CSEN: Command Sequencer Mode Enable

This bit selects the operating mode(Direct Mode or Command Sequencer Mode).

Bit	Description
0	Direct Mode is enabled. Command Sequencer Mode is disabled.
1	Command Sequencer Mode is enabled. Direct Mode is disabled.

[bit0] MEN: Module Enable

This bit enables DDRHSSPI operation.

Bit	Description
0	DDRHSSPI is disabled. DDRHSSPI enters power saving mode.
1	DDRHSSPI is enabled.

After configuring the DDRHSSPI, Software must set this bit to "1", to enable DDRHSSPI in operating mode.

When Software resets this bit:

a) In Direct Mode: DDRHSSPI aborts current SPI transfer and Slave Select is released. After the Slave Select is released, it internally enters a power saving mode.

b) In Command Sequencer Mode: DDRHSSPI generates an unmapped memory access fault interrupt if any further access to memory mapped Serial Flash Memories is received. It aborts ongoing transaction on the Serial Interface and release Slave Select. After the Slave Select has been released, it internally enters a power saving mode.

5.2. DDRHSSPI Peripheral Communication Configuration Registers (DDRHSSPIn_PCC0-3)

The DDRHSSPI Peripheral Communication Configuration Registers 0-3 control the attributes related to the serial communication on Slave Select 0-3. The Software must initialize these registers with the attributes that match the communication attributes of the Serial Flash Memory that is to be interfaced on the corresponding Slave Select line 0-3 of DDRHSSPI. When DDRHSSPI is configured in Direct Mode or in Command Sequencer Mode, each of the 4 registers is used and must be same value.

Only DDRHSSPIn_PCC0 Register is described here. Other registers (i.e. DDRHSSPIn_PCC1-3) have same bit fields.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	Reserved	SSELDEASRT[4]	SSELDEASRT[3]	SSELDEASRT[2]	SSELDEASRT[1]	SSELDEASRT[0]
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	Reserved	Reserved	CDRS[3]	CDRS[2]	CDRS[1]	CDRS[0]	Reserved
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W	R0,W0
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	SS2CD[1]	SS2CD[0]	Reserved	Reserved	ACES	Reserved	Reserved
ACCESS_TYPE	R0/W0	R/W	R/W	R0/W0	R0,W0	R/W	R0/W0	R0/W0
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:21] Reserved: Reserved

[bit20:16] SSELDEASRT[4:0]: Slave Selection De-Assertion Timeout

It defines the time period after SSEL de-assertion when a new SSEL assertion can not occur.

Bits	Description	
	SDR Mode	DDR Mode
00000	Setting this value is prohibited.	
00001	De-assertion time = 4 x SCLK cycle	De-assertion time = 3.5 x SCLK cycle
00010	De-assertion time = 5 x SCLK cycle	De-assertion time = 4.0 x SCLK cycle
...	...	
00111	De-assertion time = 10 x SCLK cycle	De-assertion time = 6.5 x SCLK cycle
...	...	
11111	De-assertion time = 34 x SCLK cycle	De-assertion time = 18.5 x SCLK cycle

[bit15:13] Reserved: Reserved

[bit12:9] CDRS[3:0]: Clock Division Ratio Select for Peripheral

This field decides the Clock Division Ratio for the SCLK.

Bits	Description
0000	Divide by 2
0001	Divide by 4.
0010	Divide by 6.
...	...
0111	Divide by 16.
...	...
1111	Divide by 32.

[bit8:7] Reserved: Reserved

[bit6:5] SS2CD[1:0]: Slave Select to Clock Delay

It defines a setup time for the Serial Flash Memory. By delaying the toggling of SCLK, DDRHSSPI delays the start of data transfer from the Slave Select active edge by a multiple of SCLK cycles.

The delay between the assertion of Slave Select and the first edge on the SCLK is given by:

(SS2CD + 0.50) x SCLK cycle [SDR mode]

(SS2CD + 0.75) x SCLK cycle [DDR mode]

When the Slave Select becomes active, the Serial Flash Memory has to prepare data transfer within the delay time defined by SS2CD bits.

Note:

- *SS2CD[1:0] = "00" must not be used when all the following conditions are met:*
  - DDRHSSPIn_DMTRP.TRP[3:0] = "1011"
  - DDRHSSPIn_DMTRP.DDRM = "1"
  - DDRHSSPIn_DMFIFOCFG.TXCTRL = "1"
  - DDRHSSPIn_TXFIFOx.TXDATA[12][10][9:8] = "1000"

[bit4:3] Reserved: Reserved

[bit2] ACES: Active Clock Edges are Same on Peripheral

This bit decides whether the active edges of the clock used for launching of data and for capturing of data are same or not. This bit takes effect both in Direct Mode and Command Sequencer Mode. This bit is only valid in SDR Mode.

Bit	Description
0	Launching of data and capturing of data is done on alternate (i.e. opposite) edges of clock.
1	Launching of data and capturing of data is done on same edges of clock.

[bit1:0] Reserved: Reserved

5.3. DDRHSSPI TX Interrupt Flag Register (DDRHSSPIn_TXF)

The DDRHSSPI TX Interrupt Flag Register indicates the status of the TX interrupt flags. These interrupt flags except TSSRS flag are set in Direct Mode only. When Command Sequencer Mode is selected, these flags except TSSRS flag are automatically cleared. The interrupt flags except TSSRS flag can be active also while the DDRHSSPI is disabled (DDRHSSPIn_MCTRL.MES = 0).

Software can enable these interrupts and wait for their assertion, or it can also use them in polling mode.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	TSSRS	TFMTS	TFLETS	Reserved	TFOS	TFES	TFFS
ACCESS_TYPE	R0,WX	R,WX	R,WX	R,WX	R0,WX	R,WX	R,WX	R,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:7] Reserved: Reserved

[bit6] TSSRS: Slave Select Released

This interrupt flag indicates that the Slave Select line is released by the DDRHSSPI.

Bit	Description
0	DDRHSSPI does not generate the event which SSEL signal is de-asserted.
1	DDRHSSPI generates the event which SSEL signal is de-asserted.

This interrupt flag triggers the TX Interrupt Service Request if it is enabled in DDRHSSPIn_TXE.TSSRE.

This interrupt flag is cleared when DDRHSSPIn_TXC.TSSRC bit is written "1".

[bit5] TFMTS: TX-FIFO Fill Level is More Than Threshold

This interrupt flag indicates whether the TX-FIFO Fill Level is more than the configured TX-FIFO threshold value or not.

Bit	Description
0	The TX-FIFO fill level is not more than the configured TX-FIFO threshold value.
1	The TX-FIFO fill level is more than the configured TX-FIFO threshold value. This bit is always set while this condition is satisfied.

i.e. DDRHSSPIn_DMFIFOSTATUS.TXFLEVEL is greater than DDRHSSPIn_DMFIFOCFG.TXFTH.

This interrupt flag triggers the TX Interrupt Service Request if it is enabled in DDRHSSPIn_TXE.TFMTE. This interrupt flag shall be used in Direct Mode only. If Command Sequencer Mode is selected, this flag is automatically cleared.

This interrupt flag is cleared when DDRHSSPIn_TXC.TFMTC bit is written "1".

[bit4] TFLETS: TX-FIFO Fill Level is Less Than or Equal to Threshold

This interrupt flag indicates whether the TX-FIFO Fill Level is less than or equal to the configured TX-FIFO threshold value or not.

Bit	Description
0	The TX-FIFO fill level is not less than or equal to the configured TX-FIFO threshold value.
1	The TX-FIFO fill level is less than or equal to the configured TX-FIFO threshold value. This bit is always set while this condition is satisfied.

i.e. DDRHSSPIn_DMFIFOSTATUS.TXFLEVEL is less than or equal to DDRHSSPIn_DMFIFOCFG.TXFTH.

This interrupt flag triggers the TX Interrupt Service Request if it is enabled in DDRHSSPIn_TXE.TFLETE. This interrupt flag shall be used in Direct Mode only. If Command Sequencer Mode is selected, this flag is automatically cleared.

This interrupt flag is cleared when DDRHSSPIn_TXC.TFLETC bit is written "1".

Note:

- *After System Reset, this bit is set to "1".*

[bit3] Reserved: Reserved

[bit2] TFOS: TX-FIFO Overrun

This interrupt flag indicates that the TX-FIFO is overrun.

Bit	Description
0	Any of DDRHSSPIn_TXFIFO0-23 Registers is not written by the Software when the TX-FIFO is full.
1	Any of DDRHSSPIn_TXFIFO0-23 Registers is written by the Software when the TX-FIFO is full.

This interrupt flag triggers the TX Interrupt Service Request if it is enabled in DDRHSSPIn_TXE.TFOE. This interrupt flag shall be used in Direct Mode only. If Command Sequencer Mode is selected, this flag is automatically cleared.

This interrupt flag is cleared when DDRHSSPIn_TXC.TFOC bit is written "1".

[bit1] TFES: TX-FIFO and Shift Register are Empty

This interrupt flag indicates whether the TX-FIFO and TX Shift Register are empty or not.

Bit	Description
0	The TX-FIFO or the TX Shift Register (in SPI Core) are not empty.
1	The TX-FIFO and the TX Shift Register (in SPI Core) are empty. This bit is always set while this condition is satisfied.

This interrupt flag triggers the TX Interrupt Service Request if it is enabled in DDRHSSPIn_TXE.TFEE. This interrupt flag shall be used in Direct Mode only. If Command Sequencer Mode is selected, this flag is automatically cleared.

This interrupt flag is cleared when DDRHSSPIn_TXC.TFEC bit is written "1".

Note:

- *After System Reset, this bit is set to "1".*

[bit0] TFFS: TX-FIFO Full

This interrupt flag indicates whether the TX-FIFO is full or not.

Bit	Description
0	The TX-FIFO is not full.
1	The TX-FIFO is full. This bit is always set while this condition is satisfied.

This interrupt flag triggers the TX Interrupt Service Request, if it is enabled in DDRHSSPIn_TXE.TFFE. This interrupt flag shall be used in Direct Mode only. If Command Sequencer Mode is selected, this flag is automatically cleared.

This interrupt flag is cleared when DDRHSSPIn_TXC.TFFC bit is written "1".

5.4. DDRHSSPI TX Interrupt Enable Register (DDRHSSPIn_TXE)

The DDRHSSPI TX Interrupt Enable Register decides whether the interrupt flags in DDRHSSPIn_TXF Register trigger the TX Interrupt Service Request or not.

The Software must enable these bits if it wants to wait for the assertion of the TX Interrupt Service Request.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	TSSRE	TFMTE	TFLETE	Reserved	TFOE	TFEE	TFFE
ACCESS_TYPE	R0,W0	R/W	R/W	R/W	R0,W0	R/W	R/W	R/W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:7] Reserved: Reserved

[bit6] TSSRE: Slave Select Released Interrupt Enable

This bit decides whether the DDRHSSPIn_TXF.TSSRS interrupt flag is routed on TX Interrupt Service Request or not.

Bit	Description
0	The DDRHSSPIn_TXF.TSSRS interrupt flag does not trigger the TX Interrupt Service Request.
1	The DDRHSSPIn_TXF.TSSRS interrupt flag triggers the TX Interrupt Service Request.

[bit5] TFMTE: TX-FIFO Fill Level is More Than Threshold Interrupt Enable

This bit decides whether the DDRHSSPIn_TXF.TFMTS interrupt flag is routed on TX Interrupt Service Request or not.

Bit	Description
0	The DDRHSSPIn_TXF.TFMTS interrupt flag does not trigger the TX Interrupt Service Request.
1	The DDRHSSPIn_TXF.TFMTS interrupt flag triggers the TX Interrupt Service Request.

[bit4] TFLETE: TX-FIFO Fill Level is Less Than or Equal To Threshold Interrupt Enable

This bit decides whether the DDRHSSPIn_TXF.TFLETS interrupt flag is routed on TX Interrupt Service Request or not.

Bit	Description
0	The DDRHSSPIn_TXF.TFLETS interrupt flag does not trigger the TX Interrupt Service Request.
1	The DDRHSSPIn_TXF.TFLETS interrupt flag triggers the TX Interrupt Service Request.

[bit3] Reserved: Reserved

[bit2] TFOE: TX-FIFO Overrun Interrupt Enable

This bit decides whether the DDRHSSPIn_TXF.TFOS interrupt flag is routed on TX Interrupt Service Request or not.

Bit	Description
0	The DDRHSSPIn_TXF.TFOS interrupt flag does not trigger the TX Interrupt Service Request.
1	The DDRHSSPIn_TXF.TFOS interrupt flag triggers the TX Interrupt Service Request.

[bit1] TFEE: TX-FIFO Empty Interrupt Enable

This bit decides whether the DDRHSSPIn_TXF.TFES interrupt flag is routed on TX Interrupt Service Request or not.

Bit	Description
0	The DDRHSSPIn_TXF.TFES interrupt flag does not trigger the TX Interrupt Service Request.
1	The DDRHSSPIn_TXF.TFES interrupt flag triggers the TX Interrupt Service Request.

[bit0] TFFE: TX-FIFO Full Interrupt Enable

This bit decides whether the DDRHSSPIn_TXF.TFFS interrupt flag is routed on TX Interrupt Service Request or not.

Bit	Description
0	The DDRHSSPIn_TXF.TFFS interrupt flag does not trigger the TX Interrupt Service Request.
1	The DDRHSSPIn_TXF.TFFS interrupt flag triggers the TX Interrupt Service Request.

5.5. DDRHSSPI TX Interrupt Clear Register (DDRHSSPIn_TXC)

The DDRHSSPI TX Interrupt Clear Register is used to clear the interrupt flags in the DDRHSSPIn_TXF Register.

By writing "1" to a bit in this register, the Software can clear the corresponding flag in the DDRHSSPIn_TXF Register.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	TSSRC	TFMTC	TFLETC	Reserved	TFOC	TFEC	TFFC
ACCESS_TYPE	R0,W0	R0,W	R0,W	R0,W	R0,W0	R0,W	R0,W	R0,W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:7] Reserved: Reserved

[bit6] TSSRC: Slave Select Released Interrupt Clear

This bit is used to clear the DDRHSSPIn_TXF.TSSRS interrupt flag.

Bit	Description
0	Writing "0" to this bit has no effect.
1	Writing "1" to this bit clears the DDRHSSPIn_TXF.TSSRS interrupt flag

The read value is "0".

[bit5] TFMTC: TX-FIFO Fill Level More Than Threshold Interrupt Clear

This bit is used to clear the DDRHSSPIn_TXF.TFMTS interrupt flag.

Bit	Description
0	Writing "0" to this bit has no effect.
1	Writing "1" to this bit clears the DDRHSSPIn_TXF.TFMTS interrupt flag.

The read value is "0".

[bit4] TFLETC: TX-FIFO Fill Level Less Than or Equal to Threshold Interrupt Clear

This bit is used to clear the DDRHSSPIn_TXF.TFLETS interrupt flag.

Bit	Description
0	Writing "0" to this bit has no effect.
1	Writing "1" to this bit clears the DDRHSSPIn_TXF.TFLETS interrupt flag.

The read value is "0".

[bit3] Reserved: Reserved

[bit2] TFOC: TX-FIFO Overrun Interrupt Clear

This bit is used to clear the DDRHSSPIn_TXF.TFOS interrupt flag.

Bit	Description
0	Writing "0" to this bit has no effect.
1	Writing "1" to this bit clears the DDRHSSPIn_TXF.TFOS interrupt flag.

The read value is "0".

[bit1] TFEC: TX-FIFO Empty Interrupt Clear

This bit is used to clear the DDRHSSPIn_TXF.TFES interrupt flag.

Bit	Description
0	Writing "0" to this bit has no effect.
1	Writing "1" to this bit clears the DDRHSSPIn_TXF.TFES interrupt flag.

The read value is "0".

[bit0] TFFC: TX-FIFO Full Interrupt Clear

This bit is used to clear the DDRHSSPIn_TXF.TFFS interrupt flag.

Bit	Description
0	Writing "0" to this bit has no effect.
1	Writing "1" to this bit clears the DDRHSSPIn_TXF.TFFS interrupt flag.

The read value is "0".

5.6. DDRHSSPI RX Interrupt Flag Register (DDRHSSPIn_RXF)

The DDRHSSPI RX Interrupt Flag Register indicates the status of the RX interrupt flags. The interrupt flags can be active also while the DDRHSSPI is disabled (DDRHSSPIn_MCTRL.MES = 0).

Software can enable these interrupts and wait for their assertion, or it can also use them in polling mode.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	TEST
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	RX,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	DLPERR	RSSRS	RFMTS	RFLETS	RFUS	Reserved	RFES	RFFS
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R0,WX	R,WX	R,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:9] Reserved: Reserved

[bit8] TEST: Test

This is a test flag. The read value is not defined. Writing to this bit does not influence other functions.

[bit7] DLPERR: Data Learning Pattern Reception Error

This bit is available only when DDRHSSPIn_MID.MID is 0x00000100 or 0x00000300.

This interrupt flag indicates that the one or more SDATA input ports received incorrect DLP from Flash Memory which is caused by wrong sample clock settings in one or more registers among DDRHSSPIn_SDATASAMPLEPTCNT0-7, DDRHSSPIn_SDATASAMPLEPTLFT0-7 and DDRHSSPIn_SDATASAMPLEPTRGH0-7.

Bit	Description
0	DDRHSSPIn_DLP.DLP matches the received DLP data, or DDRHSSPI does not check the DLP data.
1	DDRHSSPIn_DLP.DLP does not match the received DLP data.

This interrupt flag triggers the RX Interrupt Service Request, if it is enabled in DDRHSSPIn_RXE.DLPERRE.

DDRHSSPIn_DLPSAMPLESTATUS Register has information which SDATA input pins detected problems with data sampling. This flag can be used only in DDR Mode of the Command Sequencer Mode.

This interrupt flag is cleared when DDRHSSPIn_RXC.DLPERRC bit is written "1".

[bit6] RSSRS: Slave Select Released

This interrupt flag indicates that the Slave Select is released by the DDRHSSPI.

Bit	Description
0	DDRHSSPI does not generate the event which SSEL signal is de-asserted.
1	DDRHSSPI generates the event which SSEL signal is de-asserted.

This interrupt flag triggers the RX Interrupt Service Request if it is enabled in DDRHSSPIn_RXE.RSSRE.

This interrupt flag is cleared when DDRHSSPIn_RXC.RSSRC bit is written "1".

[bit5] RFMTS: RX-FIFO Fill Level is More Than Threshold

This interrupt flag indicates whether the RX-FIFO Fill Level is more than the configured RX-FIFO threshold value or not.

Bit	Description
0	The RX-FIFO fill level is not more than the configured RX-FIFO threshold value.
1	The RX-FIFO fill level is more than the configured RX-FIFO threshold value. This bit is always set while this condition is satisfied.

i.e. DDRHSSPIn_DMFIFOSTATUS.RXFLEVEL is greater than DDRHSSPIn_DMFIFOCFG.RXFTH.

This interrupt flag triggers the RX Interrupt Service Request, if it is enabled in DDRHSSPIn_RXE.RFMTE.

This interrupt flag shall be used in Direct Mode only. If Command Sequencer Mode is selected, this flag is automatically cleared.

This interrupt flag is cleared when DDRHSSPIn_RXC.RFMTC bit is written "1".

[bit4] RFLETS: RX-FIFO Fill Level is Less Than or Equal to Threshold

This interrupt flag indicates whether the RX-FIFO Fill Level is less than or equal to the configured RX-FIFO threshold value or not.

Bit	Description
0	The RX-FIFO fill level is not less than or equal to the configured RX-FIFO threshold value.

Bit	Description
1	The RX-FIFO fill level is less than or equal to the configured RX-FIFO threshold value. This bit is always set while this condition is satisfied.

i.e. DDRHSSPIn_DMFIFOSTATUS.RXFLEVEL is less than or equal to DDRHSSPIn_DMFIFOCFG.RXFTH.

This interrupt flag triggers the RX Interrupt Service Request, if it is enabled in DDRHSSPIn_RXE.RFLETE.

This interrupt flag shall be used in Direct Mode only. If Command Sequencer Mode is selected, this flag is automatically cleared.

This interrupt flag is cleared when DDRHSSPIn_RXC.RFLETC bit is written "1".

Note:

- *After System Reset, this bit is set to "1".*

[bit3] RFUS: RX-FIFO Under Run

This interrupt flag indicates that the RX-FIFO is under run.

Bit	Description
0	Any of DDRHSSPIn_RXFIFO0-23 Registers is not read by the Software when the RX-FIFO is empty.
1	Any of DDRHSSPIn_RXFIFO0-23 Registers is read by the Software when the RX-FIFO is empty.

This interrupt flag triggers the RX Interrupt Service Request if it is enabled in DDRHSSPIn_RXE.RFUE.

This interrupt flag shall be used in Direct Mode only. If Command Sequencer Mode is selected, this flag is automatically cleared.

This interrupt flag is cleared when DDRHSSPIn_RXC.RFUC bit is written "1".

Note:

- *This flag is not set when the DAP controller reads the DDRHSSPIn_RXFIFO0-23 Registers while the RX-FIFO is empty.*

[bit2] Reserved: Reserved

[bit1] RFES: RX-FIFO Empty

This interrupt flag indicates whether the RX-FIFO is empty or not.

Bit	Description
0	The RX-FIFO is not empty.
1	The RX-FIFO is empty. This bit is always set while this condition is satisfied.

This interrupt flag triggers the RX Interrupt Service Request if it is enabled in DDRHSSPIn_RXE.RFEE.
This interrupt flag shall be used in Direct Mode only. If Command Sequencer Mode is selected, this flag is automatically cleared.

This interrupt flag is cleared when DDRHSSPIn_RXC.RFEC bit is written "1".

Note:

- *After System Reset, this bit is set to "1".*

[bit0] RFFS: RX-FIFO Full

This interrupt flag indicates whether the RX-FIFO is full or not.

Bit	Description
0	The RX-FIFO is not full.
1	The RX-FIFO is full. This bit is always set while this condition is satisfied.

This interrupt flag triggers the RX Interrupt Service Request if it is enabled in DDRHSSPIn_RXE.RFFE. This interrupt flag shall be used in Direct Mode only. If Command Sequencer Mode is selected, this flag is automatically cleared.

This interrupt flag is cleared when DDRHSSPIn_RXC.RFFC bit is written "1".

5.7. DDRHSSPI RX Interrupt Enable Register (DDRHSSPIn_RXE)

The DDRHSSPI RX Interrupt Enable Register decides whether the interrupt flags in DDRHSSPIn_RXF Register trigger the RX Interrupt Service Request or not.

The Software must enable these bits if it wants to wait for the assertion of the RX Interrupt Service Request.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0/W0
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	DLPERRE	RSSRE	RFMTE	RFLETE	RFUE	Reserved	RFEE	RFFE
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R0,W0	R/W	R/W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:8] Reserved: Reserved

[bit7] DLPERRE: Data Learning Pattern Interrupt Enable

This bit decides whether the DDRHSSPIn_RXF.DLPERR interrupt flag is routed on RX Interrupt Service Request or not.

Bit	Description
0	The DDRHSSPIn_RXF.DLPERR interrupt flag does not trigger the RX Interrupt Service Request.
1	The DDRHSSPIn_RXF.DLPERR interrupt flag triggers the RX Interrupt Service Request.

[bit6] RSSRE: Slave Select Released Interrupt Enable

This bit decides whether the DDRHSSPIn_RXF.RSSRS interrupt flag is routed on RX Interrupt Service Request or not.

Bit	Description
0	The DDRHSSPIn_RXF.RSSRS interrupt flag does not trigger the RX Interrupt Service Request.
1	The DDRHSSPIn_RXF.RSSRS interrupt flag triggers the RX Interrupt Service Request.

[bit5] RFMTE: RX-FIFO Fill Level is More Than Threshold Interrupt Enable

This bit decides whether the DDRHSSPIn_RXF.RFMTS interrupt flag is routed on RX Interrupt Service Request or not.

Bit	Description
0	The DDRHSSPIn_RXF.RFMTS interrupt flag does not trigger the RX Interrupt Service Request.
1	The DDRHSSPIn_RXF.RFMTS interrupt flag triggers the RX Interrupt Service Request.

[bit4] RFLETE: RX-FIFO Fill Level is Less Than or Equal To Threshold Interrupt Enable

This bit decides whether the DDRHSSPIn_RXF.RFLETS interrupt flag is routed on RX Interrupt Service Request or not.

Bit	Description
0	The DDRHSSPIn_RXF.RFLETS interrupt flag does not trigger the RX Interrupt Service Request.
1	The DDRHSSPIn_RXF.RFLETS interrupt flag triggers the RX Interrupt Service Request.

[bit3] RFUE: RX-FIFO Under Run Interrupt Enable

This bit decides whether the DDRHSSPIn_RXF.RFUS interrupt flag is routed on RX Interrupt Service Request or not.

Bit	Description
0	The DDRHSSPIn_RXF.RFUS interrupt flag does not trigger the RX Interrupt Service Request.
1	The DDRHSSPIn_RXF.RFUS interrupt flag triggers the RX Interrupt Service Request.

[bit2] Reserved: Reserved

[bit1] RFEE: RX-FIFO Empty Interrupt Enable

This bit decides whether the DDRHSSPIn_RXF.RFES interrupt flag is routed on RX Interrupt Service Request or not.

Bit	Description
0	The DDRHSSPIn_RXF.RFES interrupt flag does not trigger the RX Interrupt Service Request.
1	The DDRHSSPIn_RXF.RFES interrupt flag triggers the RX Interrupt Service Request.

[bit0] RFFE: RX-FIFO Full Interrupt Enable

This bit decides whether the DDRHSSPIn_RXF.RFFS interrupt flag is routed on RX Interrupt Service Request or not.

Bit	Description
0	The DDRHSSPIn_RXF.RFFS interrupt flag does not trigger the RX Interrupt Service Request.
1	The DDRHSSPIn_RXF.RFFS interrupt flag triggers the RX Interrupt Service Request.

5.8. DDRHSSPI RX Interrupt Clear Register (DDRHSSPIn_RXC)

The DDRHSSPI RX Interrupt Clear Register is used to clear the interrupt flags in the DDRHSSPIn_RXF Register.

By writing "1" to a bit in this register, the Software can clear the corresponding flag in the DDRHSSPIn_RXF Register.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	DLPERRC	RSSRC	RFMTC	RFLETC	RFUC	Reserved	RFEC	RFFC
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W0	R0,W	R0,W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:8] Reserved: Reserved

[bit7] DLPERRC: DLP Error Interrupt Clear

This bit is used to clear the DDRHSSPIn_RXF.DLPERR interrupt flag.

Bit	Description
0	Writing "0" to this bit has no effect.
1	Writing "1" to this bit clears the DDRHSSPIn_RXF.DLPERR interrupt flag.

The read value is "0".

[bit6] RSSRC: Slave Select Released Interrupt Clear

This bit is used to clear the DDRHSSPIn_RXF.RSSRS interrupt flag.

Bit	Description
0	Writing "0" to this bit has no effect.
1	Writing "1" to this bit clears the DDRHSSPIn_RXF.RSSRS interrupt flag.

The read value is "0".

[bit5] RFMTC: RX-FIFO Fill Level More Than Threshold Interrupt Clear

This bit is used to clear the DDRHSSPIn_RXF.RFMTS interrupt flag.

Bit	Description
0	Writing "0" to this bit has no effect.
1	Writing "1" to this bit clears the DDRHSSPIn_RXF.RFMTS interrupt flag.

The read value is "0".

[bit4] RFLETC: RX-FIFO Fill Level Less Than or Equal to Threshold Interrupt Clear

This bit is used to clear the DDRHSSPIn_RXF.RFLETS interrupt flag.

Bit	Description
0	Writing "0" to this bit has no effect.
1	Writing "1" to this bit clears the DDRHSSPIn_RXF.RFLETS interrupt flag.

The read value is "0".

[bit3] RFUC: RX-FIFO Under Run Interrupt Clear

This bit is used to clear the DDRHSSPIn_RXF.RFUS interrupt flag.

Bit	Description
0	Writing "0" to this bit has no effect.
1	Writing "1" to this bit clears the DDRHSSPIn_RXF.RFUS interrupt flag.

The read value is "0".

[bit2] Reserved: Reserved

[bit1] RFEC: RX-FIFO Empty Interrupt Clear

This bit is used to clear the DDRHSSPIn_RXF.RFES interrupt flag.

Bit	Description
0	Writing "0" to this bit has no effect.
1	Writing "1" to this bit clears the DDRHSSPIn_RXF.RFES interrupt flag.

The read value is "0".

[bit0] RFFC: RX-FIFO Full Interrupt Clear

This bit is used to clear the DDRHSSPIn_RXF.RFFS interrupt flag.

Bit	Description
0	Writing "0" to this bit has no effect.
1	Writing "1" to this bit clears the DDRHSSPIn_RXF.RFFS interrupt flag.

The read value is "0".

5.9. DDRHSSPI Fault Status Flag Register (DDRHSSPIn_FAULTF)

The DDRHSSPI Fault Status Flag Register indicates the status of the FAULT flag.

Once a fault occurs, the Software needs to take a corrective action. Whenever one of these flags gets set, the corresponding bus access returns with a bus error response.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	DLPFS	Reserved	DRCBSFS	DWCBSFS	PVFS	Reserved	UMAFS
ACCESS_TYPE	R0,WX	R,WX	R0,WX	R,WX	R,WX	R,WX	R0,WX	R,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:7] Reserved: Reserved

[bit6] DLPFS: DLP Error Fault

This status flag indicates that the DLP error fault has occurred on the data received on Serial Interface.

Bit	Description
0	DDRHSSPI does not detect the DLP error.
1	DDRHSSPI detects the DLP error.

The DLP error fault occurs if the DDRHSSPI receives 8 bit DLP value that is not matched DDRHSSPIn_DLP.DLP.

This status flag is cleared when DDRHSSPIn_FAULTC.DLPFC bit is written "1".

[bit5] Reserved: Reserved

[bit4] DRCBSFS: DMA Read Channel Block Size Fault

This status flag indicates that the block size fault has occurred on DMA read channel.

Bit	Description
0	DDRHSSPI does not detect the DMA read channel block size fault.
1	DDRHSSPI detects the DMA read channel block size fault.

The DMA read channel block size fault occurs if the DDRHSSPI RX block counter is 0 and there is a valid read access to the RX-FIFO (except for from the DAP controller).
This status flag is cleared when DDRHSSPIn_FAULTC.DRCBSFC bit is written "1".

[bit3] DWCBSFS: DMA Write Channel Block Size Fault

This status flag indicates that the block size fault has occurred on DMA write channel.

Bit	Description
0	DDRHSSPI does not detect the DMA write channel block size fault.
1	DDRHSSPI detects the DMA write channel block size fault.

The DMA write channel block size fault occurs if the DDRHSSPI TX block counter is 0 and there is a valid write access to the TX-FIFO.

This status flag is cleared when DDRHSSPIn_FAULTC.DWCBSFC bit is written "1".

[bit2] PVFS: Protection Violation Fault

This status flag indicates that a protection violation fault has occurred.

Bit	Description
0	DDRHSSPI does not detect the protection violation fault.
1	DDRHSSPI detects the protection violation fault.

The protection violation fault includes following cases:

- Access to a reserved register except some reserved registers.
- Write access to a read-only register.

This status flag is cleared when DDRHSSPIn_FAULTC.PVFC bit is written "1".

[bit1] Reserved: Reserved

[bit0] UMAFS: Unmapped Memory Access Fault

This status flag indicates that an unmapped memory access fault has occurred.

This bit is set by DDRHSSPI when any of the following event occurs.

- In Direct Mode (i.e. DDRHSSPIn_MCTRL.CSEN = 0), a System Bus access within the 256 MB address range starting from the DDRHSSPI base address is detected.
- In Command Sequencer Mode (i.e. DDRHSSPIn_MCTRL.CSEN = 1), an access to a Serial Flash Memory which is not enabled (in DDRHSSPIn_CSCFG.SSEL0EN-SSEL3EN bits) is detected.
- In Command Sequencer Mode (i.e. DDRHSSPIn_MCTRL.CSEN = 1), a System Bus access to a memory location which is outside the memory range being mapped onto the four Slave Selects (configured through the DDRHSSPIn_CSCFG.MSEL) is detected.
- While the module is disabled (i.e. DDRHSSPIn_MCTRL.MES = 0), an access to a mapped memory is detected.

This status flag is cleared when DDRHSSPIn_FAULTC.UMAFC bit is written "1".

5.10. DDRHSSPI Fault Status Clear Register (DDRHSSPIn_FAULTC)

The DDRHSSPI Fault Status Clear Register is used to clear the status flags in the DDRHSSPIn_FAULTF Register.

By writing "1" to a bit in this register, the Software can clear the corresponding flag in the DDRHSSPIn_FAULTF Register.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	DLPFC	Reserved	DRCBSFC	DWCBSFC	PVFC	Reserved	UMAFC
ACCESS_TYPE	R0,W0	R0,W	R0,W0	R0,W	R0,W	R0,W	R0,W0	R0,W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:7] Reserved: Reserved

[bit6] DLPFC: DLP Error Fault Clear

This bit is used to clear the DDRHSSPIn_FAULTF.DLPFS status flag.

Bit	Description
0	Writing "0" to this bit has no effect.
1	Writing "1" to this bit clears the DDRHSSPIn_FAULTF.DLPFS interrupt flag.

The read value is "0".

[bit5] Reserved: Reserved

[bit4] DRCBSFC: DMA Read Channel Block Size Fault Status Clear

This bit is used to clear the DDRHSSPIn_FAULTF.DRCBSFS status flag.

Bit	Description
0	Writing "0" to this bit has no effect.
1	Writing "1" to this bit clears the DDRHSSPIn_FAULTF.DRCBSFS interrupt flag.

The read value is "0".

[bit3] DWCBSFC: DMA Write Channel Block Size Fault Status Clear

This bit is used to clear the DDRHSSPIn_FAULTF.DWCBSFS status flag.

Bit	Description
0	Writing "0" to this bit has no effect.
1	Writing "1" to this bit clears the DDRHSSPIn_FAULTF.DWCBSFS status flag.

The read value is "0".

[bit2] PVFC: Protection Violation Fault Status Clear

This bit is used to clear the DDRHSSPIn_FAULTF.PVFS status flag.

Bit	Description
0	Writing "0" to this bit has no effect.
1	Writing "1" to this bit clears the DDRHSSPIn_FAULTF.PVFS status flag.

The read value is "0".

[bit1] Reserved: Reserved

[bit0] UMAFC: Unmapped Memory Access Fault Interrupt Clear

This bit is used to clear the DDRHSSPIn_FAULTF.UMAFS status flag.

Bit	Description
0	Writing "0" to this bit has no effect.
1	Writing "1" to this bit clears the DDRHSSPIn_FAULTF.UMAFS status flag.

The read value is "0".

5.11. DDRHSSPI Direct Mode Configuration Register (DDRHSSPIn_DMCFG)

The DDRHSSPI Direct Mode Configuration Register configures the following operational parameter of DDRHSSPI:

- Byte Counter Mode of Slave Select De-assertion

This register is used only when DDRHSSPI is in Direct Mode.

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	SSDC	Reserved
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0/W0	R/W	R0,W0
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit7:2] Reserved: Reserved

[bit1] SSDC: Slave Select De-Assertion Control

The SSDC bit decides how the Slave Select is de-asserted:

Bit	Description
0	This is prohibited when DDRHSSPI is operated in Direct Mode.
1	Byte Counter Mode. DDRHSSPIn_DMBCC.BCC is used to decide when to de-assert the Slave Select.

[bit0] Reserved: Reserved

5.12. DDRHSSPI DMA Enable Register (DDRHSSPIn_DMAEN)

The DDRHSSPI DMA Enable Register can be used by the Software, for enabling/disabling of the DMA Service Requests generated by DDRHSSPI.

This register is used in Direct Mode only.

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	TXDMAEN	RXDMAEN
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0/W0	R/W	R/W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit7:2] Reserved: Reserved

[bit1] TXDMAEN: TX DMA Enable

This bit enable or disable the TX DMA channel(DMA write channel side).

Bit	Description
0	TX DMA channel is disabled.
1	TX DMA channel is enabled.

This bit is valid in Direct Mode.

[bit0] RXDMAEN: RX DMA Enable

This bit enable or disable the RX DMA channel(DMA read channel side).

Bit	Description
0	RX DMA channel is disabled.
1	RX DMA channel is enabled.

This bit is valid in Direct Mode.

5.13. DDRHSSPI Direct Mode Start Register (DDRHSSPIn_DMSTART)

The DDRHSSPI Direct Mode Start Register can be used by the Software, for triggering the start of the serial transfer.

This register is used only in Direct Mode.

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	START
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R,W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit7:1] Reserved: Reserved

[bit0] START: Start Transfer

This bit is a trigger to start a new serial transfer in Direct Mode.

Bit	Description
0	Writing "0" to this bit has no effect.
1	Writing "1" to this bit sets this bit. This bit can be only set by Software and it is cleared by Hardware.

DDRHSSPI resets this bit to "0" in any of following conditions:

- Serial transfer ends.
- Module is disabled (DDRHSSPIn_MCTRL.MES= 0)
- DDRHSSPI is switched from Direct Mode to Command Sequencer Mode.

Writing "1" to this bit when the bit is already set to "1" has no effect on the current serial transfer (in any cases).

5.14. DDRHSSPI Direct Mode Peripheral Select Register (DDRHSSPIn_DMPSEL)

The DDRHSSPI Direct Mode Peripheral Select Register can be used by the Software to select one of 4 Slave Selects for initiating the serial transfer.

This register is used only when DDRHSSPI is in Direct Mode.

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	PSEL[1]	PSEL[0]
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R/W	R/W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit7:2] Reserved: Reserved

[bit1:0] PSEL[1:0]: Peripheral Select

The PSEL bits decide which of the 4 Slave Selects in oDDRHSSPIn_SSEL3-0 is asserted for the current serial transfer.

Bits	Description
00	oDDRHSSPIn_SSEL0 is asserted.
01	oDDRHSSPIn_SSEL1 is asserted.
10	oDDRHSSPIn_SSEL2 is asserted.
11	oDDRHSSPIn_SSEL3 is asserted.

5.15. DDRHSSPI Direct Mode Transfer Protocol Register (DDRHSSPIn_DMTRP)

The DDRHSSPI Direct Mode Transfer Protocol Register configures the transfer protocol of the serial transfer.

This register is used only when DDRHSSPI is in Direct Mode.

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	Reserved	DDRM	Reserved	TRP[3]	TRP[2]	TRP[1]	TRP[0]
ACCESS_TYPE	R0,W0	R0/W0	R/W	R0,W0	R/W	R/W	R/W	R/W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit7:6] Reserved: Reserved

[bit5] DDRM: DDR Mode

This bit indicates whether DDRHSSPI is in DDR Mode or SDR Mode, in Direct Mode.

Bit	Description
0	DDRHSSPI is in Single Data Rate (SDR) Mode.
1	DDRHSSPI is in Dual Data Rate (DDR) Mode.

[bit4] Reserved: Reserved

[bit3:0] TRP[3:0]: Transfer Protocol

Bits TRP[3:2] indicate the Duplex Configuration: TX-and-RX or TX-Only.

Bits TRP[1:0] indicate the using protocol: Legacy, Quad or Octal.

Bits	Description
0000	TX-and-RX in Legacy Mode is configured.
0011	This value is available only when DDRHSSPIn_MID.MID is 0x00000100 or 0x00000300. TX-and-RX in Octal Mode is configured.
1000	TX-Only in Legacy Mode is configured.
1010	TX-Only in Quad Mode is configured.
1011	This value is available only when DDRHSSPIn_MID.MID is 0x00000100 or 0x00000300. TX-Only in Octal Mode is configured.

All other combinations are RESERVED and prohibited.

5.16. DDRHSSPI Byte Count Control Register (DDRHSSPIn_DMBCC)

The DDRHSSPI Byte Count Control Register configures the number of bytes that would be transferred in the serial transfer.

This register is used in Direct Mode only.

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	BCC[15]	BCC[14]	BCC[13]	BCC[12]	BCC[11]	BCC[10]	BCC[9]	BCC[8]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	BCC[7]	BCC[6]	BCC[5]	BCC[4]	BCC[3]	BCC[2]	BCC[1]	BCC[0]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit15:0] BCC[15:0]: Byte Count Control

This field is used by DDRHSSPI in Direct Mode and when DDRHSSPIn_DMCFG.SSDC= 1.

BCC must be programmed by the Software with the number of bytes to be transmitted/received/both.

The value in this field is loaded in a down-counter immediately after it is written in this register, and the counter is decremented when a byte is serially transferred. DDRHSSPI completes the transaction and de-asserts the Slave Select when this down-counter reaches zero.

5.17. DDRHSSPI Byte Count Status Register (DDRHSSPIn_DMBCS)

The DDRHSSPI Byte Count Status Register is a read-only register, which can be used by the Software, to know how many bytes are yet to be transferred, in the current serial transfer.

This register is valid only when DDRHSSPI is configured so that Byte Counter Mode is selected (i.e. if DDRHSSPIn_DMCFG.SSDC= 1 in Direct Mode).

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	BCS[15]	BCS[14]	BCS[13]	BCS[12]	BCS[11]	BCS[10]	BCS[9]	BCS[8]
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	BCS[7]	BCS[6]	BCS[5]	BCS[4]	BCS[3]	BCS[2]	BCS[1]	BCS[0]
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit15:0] BCS[15:0]: Byte Count Status

This read-only field is valid only when DDRHSSPI is in Direct Mode and when DDRHSSPIn_DMCFG.SSDC= 1.

BCS indicates the number of bytes in the current serial transfer that are not yet serially transmitted/received/both. Counter is updated with a new value only when current byte is considered to be transferred i.e. when last bit has been sent to memory or last bit from the memory has been received and put into FIFO.

For each FIFO entry where the number of cycles is less than a number of complete bytes, the BCS will be decremented by upper rounded value. For each FIFO entry where the number of cycles is less than a byte, the BCS will be decremented by one.

For FIFO entries where the number of cycles is more than one byte and less than two bytes, then the BCS will be decremented by two etc.

5.18. DDRHSSPI Direct Mode FIFO Status Register (DDRHSSPIn_DMFIFOSTATUS)

The DDRHSSPI Direct Mode FIFO Status Register contains the status bits like current fill-level of the TX/RX-FIFOs and whether the TX/RX path is active/idle i.e. Slave Select is asserted. This register is used only in Direct Mode.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	SSACTIVE
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	Reserved	Reserved	TXF LEVEL[4]	TXF LEVEL[3]	TXF LEVEL[2]	TXF LEVEL[1]	TXF LEVEL[0]
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	Reserved	Reserved	RXF LEVEL[4]	RXF LEVEL[3]	RXF LEVEL[2]	RXF LEVEL[1]	RXF LEVEL[0]
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:17] Reserved: Reserved

[bit16] SSACTIVE: Slave Select Active

This flag indicates whether a Slave Select is active.

Bit	Description
0	Slave Select is not active.
1	Slave Select is active.

[bit15:13] Reserved: Reserved

[bit12:8] TXFLEVEL[4:0]: Current Fill Level of TX-FIFO
This field indicates the current fill level of the TX-FIFO.

[bit7:5] Reserved: Reserved

[bit4:0] RXFLEVEL[4:0]: Current Fill Level of RX-FIFO
This field indicates the current fill level of the RX-FIFO.

5.19. DDRHSSPI Direct Mode FIFO Configuration Register (DDRHSSPIn_DMFIFOCFG)

The DDRHSSPI FIFO Configuration Register configures the operation of the TX-FIFO and the RX-FIFO. The Software can configure the FIFO threshold levels and the FIFO width.

The Software can also initialize the FIFOs using the TXFLSH and RXFLSH bits in this register.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	Reserved	TXFLSH	RXFLSH	TXCTRL	FWIDTH[1]	FWIDTH[0]
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W	R0,W	R/W	R/W	R/W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	Reserved	Reserved	TXFTH[4]	TXFTH[3]	TXFTH[2]	TXFTH[1]	TXFTH[0]
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	1	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	Reserved	Reserved	RXFTH[4]	RXFTH[3]	RXFTH[2]	RXFTH[1]	RXFTH[0]
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	1	1	1	1

[bit31:21] Reserved: Reserved

[bit20] TXFLSH: TX-FIFO Flush

This bit can be used by the Software to flush the TX-FIFO and TX Shift Register.

Bit	Description
0	Writing "0" has no effect.
1	Writing "1" flushes the TX-FIFO and TX Shift Register. Please do not write "1" to this bit, while DDRHSSPIn_DMFIFOSTATUS.SSACTIVE bit is "1".

The read value is "0".

[bit19] RXFLSH: RX-FIFO Flush

This register can be used by the Software to flush the RX-FIFO and RX Shift Register.

Bit	Description
0	Writing "0" has no effect.
1	Writing "1" flushes the RX-FIFO and RX Shift Register. Please do not write "1" to this bit, while DDRHSSPIn_DMFIFOSTATUS.SSACTIVE bit is "1".

This read value is "0".

[bit18] TXCTRL: TXCTRL Bit to be Written to TX-FIFO

When one of the DDRHSSPIn_TXFIFO0-23 Registers is written, the 33rd bit (TXCTRL bit) in TX-FIFO word takes this value.

Before writing to one of the DDRHSSPIn_TXFIFO0-23 Registers, the Software must update this bit, to control the next entry of TX-FIFO. Please refer to Section 4.1.3 for details of TXCTRL.

[bit17:16] FWIDTH[1:0]: FIFO Width

This field indicates the FIFO Width. Depending on the configured width of the FIFO, the available size of the Shift-Register in the SPI-Core also changes.

Bits	Description
00	TX-FIFO, RX-FIFO and Shift-Register are 8-bit wide.
01	TX-FIFO, RX-FIFO and Shift-Register are 16-bit wide.
11	TX-FIFO, RX-FIFO and Shift-Register are 32-bit wide.

All other combinations are RESERVED and prohibited.

[bit15:13] Reserved: Reserved

[bit12:8] TXFTH[4:0]: TX-FIFO Threshold Level

Software must program this field with the threshold level of the TX-FIFO. The Maximum allowed value is 24 that is equal to TX-FIFO depth.

[bit7:5] Reserved: Reserved

[bit4:0] RXFTH[4:0]: RX-FIFO Threshold Level

Software must program this field with the threshold level of the RX-FIFO. The Maximum allowed value is 24 that is equal to RX-FIFO depth.

5.20. DDRHSSPI TX-FIFO Registers (DDRHSSPIn_TXFIFO0-23)

The DDRHSSPI TX-FIFO Registers are used to push the data into the TX-FIFO. There are 24 registers for the input of the TX-FIFO, and they are all consecutively placed in the register-map. Each of these 24 registers is identical in function and fields. This is because the System Bus protocol does not support burst transfers to the same address. Only DDRHSSPIn_TXFIFO0 Register is described here. Other registers (i.e. DDRHSSPIn_TXFIFO1-23) have same bit fields.

Only 8-bit, 16-bit or 32-bit accesses are allowed to these registers. A write access to these registers by the Software pushes 33-bits into the TX-FIFO. This 33rd bit corresponds to the DDRHSSPIn_DMFIFOCFG. TXCTRL bit. These registers are used only in Direct Mode.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	TXDATA[31]	TXDATA[30]	TXDATA[29]	TXDATA[28]	TXDATA[27]	TXDATA[26]	TXDATA[25]	TXDATA[24]
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	TXDATA[23]	TXDATA[22]	TXDATA[21]	TXDATA[20]	TXDATA[19]	TXDATA[18]	TXDATA[17]	TXDATA[16]
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	TXDATA[15]	TXDATA[14]	TXDATA[13]	TXDATA[12]	TXDATA[11]	TXDATA[10]	TXDATA[9]	TXDATA[8]
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	TXDATA[7]	TXDATA[6]	TXDATA[5]	TXDATA[4]	TXDATA[3]	TXDATA[2]	TXDATA[1]	TXDATA[0]
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:0] TXDATA[31:0]: TX-FIFO Data

Writing to this 32-bit register pushes the data into the next location of TX-FIFO, and increments the write pointer of TX-FIFO. In addition, the DDRHSSPIn_DMFIFOCFG.TXCTRL bit is internally written as the 33rd bit of a TX-FIFO entry.

When the Shift Register width is 32 bits (DDRHSSPIn_DMFIFOCFG.FWIDTH=0b11), only 32-bit access is possible.

When the Shift Register width is 16 bits (DDRHSSPIn_DMFIFOCFG.FWIDTH=0b01), both 16-bit and 32-bit access are possible.

a) 32-bit access shall access the full width of FIFO, but only the lower 16 bits are valid.

b) 16-bit access with any alignment shall access the lower 16 bits of the FIFO.

When the Shift Register width is 8 bits (DDRHSSPIn_DMFIFOCFG.FWIDTH=0b00), then 8-bit, 16-bit and 32-bit access are possible.

a) 32-bit access shall access the full width of FIFO, but only the lower 8 bits are valid.

b) 16-bit access with any alignment shall access the lower 16 bits of the FIFO, but only the lower 8 bits are valid.

c) 8-bit access with any alignment shall access the lower 8 bits of the FIFO.

Other conditions (e.g. 8-bit write access in 32-bit wide setting) will cause a bus error response.

While the TX-FIFO is full, a write access to this register pushes the new data into the TX-FIFO and triggers a TX-FIFO overrun(DDRHSSPIn_TXF.TFOS) event. When the TX-FIFO overrun condition occurs, the integrity of the data transmitted over the Serial Interface is not guaranteed. Before writing to this register, the Software must ensure that the TX-FIFO is not full to avoid an overrun.

5.21. DDRHSSPI RX-FIFO Registers (DDRHSSPIn_RXFIFO0-23)

The DDRHSSPI RX-FIFO Registers are used to pop the data out of the RX-FIFO. There are 24 registers for the output of the RX-FIFO, and they are all consecutively placed in the register-map. Each of these 24 registers is identical in function and fields. This is because the System Bus protocol does not support burst transfers to the same address. Only DDRHSSPIn_RXFIFO0 Register is described here. Other registers (i.e. DDRHSSPIn_RXFIFO1-23) have same bit fields.

8-bit, 16-bit and 32-bit read accesses are allowed to these registers. By reading these registers, the Software can pop the data out of the RX-FIFO. These registers are used in Direct Mode.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	RXDATA[31]	RXDATA[30]	RXDATA[29]	RXDATA[28]	RXDATA[27]	RXDATA[26]	RXDATA[25]	RXDATA[24]
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	RXDATA[23]	RXDATA[22]	RXDATA[21]	RXDATA[20]	RXDATA[19]	RXDATA[18]	RXDATA[17]	RXDATA[16]
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	RXDATA[15]	RXDATA[14]	RXDATA[13]	RXDATA[12]	RXDATA[11]	RXDATA[10]	RXDATA[9]	RXDATA[8]
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	RXDATA[7]	RXDATA[6]	RXDATA[5]	RXDATA[4]	RXDATA[3]	RXDATA[2]	RXDATA[1]	RXDATA[0]
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:0] RXDATA[31:0]: RX-FIFO Data

Reading this register returns a word of data from the RX-FIFO location pointed by the RX-FIFO read pointer. After a read access to this register, the RX-FIFO read pointer is incremented, if the read cycle was initiated by the System Bus master (not the DAP controller). If the DAP controller reads this register, the RX-FIFO read pointer is not incremented.

When the Shift Register width is 32 bits (DDRHSSPIn_DMFIFOCFG.FWIDTH=0b11), only 32-bit access is possible.

When the Shift Register width is 16 bits (DDRHSSPIn_DMFIFOCFG.FWIDTH=0b01), both 16-bit and 32-bit access are possible.

a) 32-bit access shall access the full width of FIFO, but only the lower 16 bits are valid.

b) 16-bit access with any alignment shall access the lower 16 bits of the FIFO.

When the Shift Register width is 8 bits (DDRHSSPIn_DMFIFOCFG.FWIDTH= 0b00), then 8-bit, 16-bit and 32-bit access are possible.

a) 32-bit access shall access the full width of FIFO, but only the lower 8 bits are valid.

b) 16-bit access with any alignment shall access the lower 16 bits of the FIFO, but only the lower 8 bits are valid.

c) 8-bit access with any alignment shall access the lower 8 bits of the FIFO.

Other conditions (e.g. 8-bit write access in 32-bit wide setting) will cause a bus error response.

When the bus width to access the FIFO is greater than the width of FIFO, the upper redundant bits of the read data on the System Bus are filled with "0". For example, if configured FIFO width is 8 bits (DDRHSSPIn_DMFIFOCFG.FWIDTH= 0b00), and the read access of the FIFO is on 32 bits, then

- Only the bits RXDATA[7:0] are valid, AND
- The other bits RXDATA[31:8] are all "0".

The Software must not use any data from the unused most significant bits.

While the RX-FIFO is empty, a read access to this register pops invalid data out of the RX-FIFO. A RX-FIFO under run interrupt (DDRHSSPIn_RXF.RFUS) event is triggered if the read cycle was initiated by the System Bus master other than the DAP controller. If the DAP controller reads this register while the RX-FIFO is empty, the DDRHSSPIn_RXF.RFUS flag is not set.

5.22. DDRHSSPI Read Command Sequencer Data/Control Registers (DDRHSSPIn_RDCSDC0-11)

The DDRHSSPI Read Command Sequencer Data/Control Registers 0-11 are the list of Data/Control registers which configure the phases of the serial transaction generated by the Command Sequencer for Memory Read operations. These registers are used only in Command Sequencer Mode.

Only the DDRHSSPIn_RDCSDC0 Register is explained here. Other registers have identical fields.

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	RDC SDATA[7]	RDC SDATA[6]	RDC SDATA[5]	RDC SDATA[4]	RDC SDATA[3]	RDC SDATA[2]	RDC SDATA[1]	RDC SDATA[0]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	DEC
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R/W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit15:8] RDCSDATA[7:0]: Command Sequencer Data or Control Byte for Memory-Read Transactions

This field contains either a command code or a control code according to the setting of DEC bit.

- When DEC bit is "0":
  The RDCSDATA contains the 8-bit data to be transmitted on the Serial Interface.
- When DEC bit is "1":
  The RDCSDATA[7:0] is decoded as follows.

  RDCSDATA[2:0] = 0b000: Transmit address bits [7:0] of the Serial Flash Memory address.

  RDCSDATA[2:0] = 0b001: Transmit address bits [15:8] of the Serial Flash Memory address.

  RDCSDATA[2:0] = 0b010: Transmit address bits [23:16] of the Serial Flash Memory address.

  RDCSDATA[2:0] = 0b011: Transmit address bits [31:24] of the Serial Flash Memory address.

  RDCSDATA[2:0] = 0b100: Dummy cycle(s) on the SDATA output lines for RDCSDATA [7:3] number of SCLK cycles.

  - RDCSDATA[7:3]= 0b00000 -> Dummy cycle for 1 SCLK cycle.
  - RDCSDATA[7:3]= 0b00001 -> Dummy cycles for 2 SCLK cycles.

  ....

  RDCSDATA[2:0] = **0b**111: End of list.
  All other values of RDCSDATA[2:0] are reserved and must not be used.

[bit7:1] Reserved: Reserved

[bit0] DEC: Decode

This bit controls whether to decode RDCSDATA[2:0].

Bit	Description
0	Transmit the RDCSDATA[7:0] as it is.
1	Decode the RDCSDATA[2:0] to decide the further action.

Notes:

- *The Command Sequencer Mode can use only a Continuous Read Command Sequence of the Serial FLASH Memory.*
- *The transmitted address bits of the Serial Flash Memory address must be registered in the DDRHSSPIn_RDCSDC0 Register.*

5.23. DDRHSSPI Module ID Register (DDRHSSPIn_MID)

This is a read-only register with a unique Module Identification Number which identifies the version of the DDRHSSPI used in the MCU.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	MID[31]	MID[30]	MID[29]	MID[28]	MID[27]	MID[26]	MID[25]	MID[24]
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	MID[23]	MID[22]	MID[21]	MID[20]	MID[19]	MID[18]	MID[17]	MID[16]
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	MID[15]	MID[14]	MID[13]	MID[12]	MID[11]	MID[10]	MID[9]	MID[8]
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX or *1 R1,WX	R0,WX or *1 R1,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0 or *1 1	0 or *1 1

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	MID[7]	MID[6]	MID[5]	MID[4]	MID[3]	MID[2]	MID[1]	MID[0]
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX or *1 R1,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0 or *1 1

*1: For these values, see product specification.

[bit31:0] MID[31:0]: Module ID

This read-only register gives the unique module identification number of DDRHSSPI.

The unique module ID number identifies the version of the DDRHSSPI used in the MCU.

Bits	Description
0x00000001	This version has limitations as below. – Octal protocol (Dual Quad and Dual Legacy) is not supported. ➢ DDRHSSPIn_DMTRP.TRP[1:0] shall not be 0b11. ➢ DDRHSSPIn_CSCFG.MBM[1:0] shall not be 0b11. – Sample Point Control function is not supported. ➢ DDRHSSPIn_SDATASAMPLEPTCNT0-7.SDATASMPTCNT shall be fixed to 0. ➢ DDRHSSPIn_SDATASAMPLEPTLFT0-7.SDATASMPTLFT shall be fixed to 0. ➢ DDRHSSPIn_SDATASAMPLEPTRGH0-7.SDATASMPTRGH shall be fixed to 0. – DLP function is not supported. ➢ DDRHSSPIn_MCTRL.DLPEN shall be fixed to "0".
0x00000100	This version supports the following features. – Octal protocol (Dual Quad and Dual Legacy) – Sample Point Control function – DLP function
0x00000300	The DDRHSSPIn_SDATASAMPLEPT* registers have been extended from six ([5:0]) to seven ([6:0]) bits, All other features are exactly the same as MID = 0x00000100.

5.24. DDRHSSPI Command Sequencer Prefetch Address Register (DDRHSSPIn_CSPREFETCHADDR)

The DDRHSSPI Command Sequencer Prefetch Address Register is used to keep the next available address of memory that is available in Command Sequencer Mode Prefetch Buffer.

This register is used only in Command Sequencer Mode.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	PRF ADDR[31]	PRF ADDR[30]	PRF ADDR[29]	PRF ADDR[28]	PRF ADDR[27]	PRF ADDR[26]	PRF ADDR[25]	PRF ADDR[24]
ACCESS_TYPE	R	R	R	R	R	R	R	R
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	PRF ADDR[23]	PRF ADDR[22]	PRF ADDR[21]	PRF ADDR[20]	PRF ADDR[19]	PRF ADDR[18]	PRF ADDR[17]	PRF ADDR[16]
ACCESS_TYPE	R	R	R	R	R	R	R	R
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	PRF ADDR[15]	PRF ADDR[14]	PRF ADDR[13]	PRF ADDR[12]	PRF ADDR[11]	PRF ADDR[10]	PRF ADDR[9]	PRF ADDR[8]
ACCESS_TYPE	R	R	R	R	R	R	R	R
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	PRF ADDR[7]	PRF ADDR[6]	PRF ADDR[5]	PRF ADDR[4]	PRF ADDR[3]	PRF ADDR[2]	PRF ADDR[1]	PRF ADDR[0]
ACCESS_TYPE	R	R	R	R	R	R	R	R
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:0] PRFADDR[31:0]: Prefetch Buffer Address

This register is used in Command Sequencer Mode.

Reading this register returns address available at the top of the Prefetch Buffer.

Writing to this register is not allowed.

5.25. DDRHSSPI SDATA Center Clock Sample Point Registers (DDRHSSPIn_SDATASAMPLEPTCNT0-7)

This feature is available only when DDRHSSPIn_MID.MID is 0x00000100 or 0x00000300.

The DDRHSSPI SDATA Center Clock Sample Point Registers 0-7 are used for fine tuning and configuring of the clock center sampling point for all incoming SDATA ports. Each SDATA[7:0] input port has one register.

Only the DDRHSSPIn_SDATASAMPLEPTCNT0 Register is explained here. Other registers have identical fields.

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	Reserved	SDATASMPTCNT[5]	SDATASMPTCNT[4]	SDATASMPTCNT[3]	SDATASMPTCNT[2]	SDATASMPTCNT[1]	SDATASMPTCNT[0]
ACCESS_TYPE	R0,W0	R0,W0	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit7:6] Reserved: Reserved

[bit5:0] SDATASMPTCNT[5:0]: SDATA Sample Point Center Control

This field defines, for an incoming SDATA port, sample clock delay.

Clock is delayed through a series of buffers and each buffer output from delay chain, can be selected by putting corresponding value into this register.

Notes:

- *About the actual use cases, refer to the 4.4.4 Using the DDRHSSPI in Command Sequencer Mode of Operation*
  -> "Calibration of SDATA Sampling Points", and "Re-adjustment of SDATA Sampling Points".
- *When in using the DLP check function, please read the Notes of DDRHSSPIn_MCTRL.DLPEN bit carefully.*

5.26. DDRHSSPI SDATA Left Clock Sample Point Registers (DDRHSSPIn_SDATASAMPLEPTLFT0-7)

This feature is available only when DDRHSSPIn_MID.MID is 0x00000100 or 0x00000300.

The DDRHSSPI SDATA Left Clock Sample Point Registers 0-7 are used for fine tuning and configuring of the clock left sampling point for all incoming SDATA ports. Each SDATA[7:0] input port has one register.

Only the DDRHSSPIn_SDATASAMPLEPTLFT0 Register is explained here. Other registers have identical fields.

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	Reserved	SDATASMPTLFT[5]	SDATASMPTLFT[4]	SDATASMPTLFT[3]	SDATASMPTLFT[2]	SDATASMPTLFT[1]	SDATASMPTLFT[0]
ACCESS_TYPE	R0,W0	R0,W0	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit7:6] Reserved: Reserved

[bit5:0] SDATASMPTLFT[5:0]: SDATA Left Clock Sample Point Control

This field defines, for an incoming SDATA port, sample clock delay.

Clock is delayed through a series of buffers and each buffer output from delay chain, can be selected by putting corresponding value into this register.

Notes:

- *About the actual use cases, refer to the 4.4.4 Using the DDRHSSPI in Command Sequencer Mode of Operation*
  -> "Calibration of SDATA Sampling Points", and "Re-adjustment of SDATA Sampling Points".
- *When in using the DLP check function, please read the Notes of DDRHSSPIn_MCTRL.DLPEN bit carefully.*

5.27. DDRHSSPI SDATA Right Clock Sample Point Registers (DDRHSSPIn_SDATASAMPLEPTRGH0-7)

This feature is available only when DDRHSSPIn_MID.MID is 0x00000100 or 0x00000300.

The DDRHSSPI SDATA Right Clock Sample Point Registers 0-7 are used for fine tuning and configuring of the clock right sampling point for all incoming SDATA ports. Each SDATA[7:0] input port has one register.

Only the DDRHSSPIn_SDATASAMPLEPTRGH0 Register is explained here. Other registers have identical fields.

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	Reserved	SDATASMPTRGH[5]	SDATASMPTRGH[4]	SDATASMPTRGH[3]	SDATASMPTRGH[2]	SDATASMPTRGH[1]	SDATASMPTRGH[0]
ACCESS_TYPE	R0,W0	R0,W0	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit7:6] Reserved: Reserved

[bit5:0] SDATASMPTRGH[5:0]: SDATA Right Clock Sample Point Control

This field defines, for an incoming SDATA port, sample clock delay.

Clock is delayed through a series of buffers and each buffer output from delay chain, can be selected by putting corresponding value into this register.

Notes:

- *About the actual use cases, refer to the 4.4.4 Using the DDRHSSPI in Command Sequencer Mode of Operation*
  -> "Calibration of SDATA Sampling Points", and "Re-adjustment of SDATA Sampling Points".
- *When in using the DLP check function, please read the Notes of DDRHSSPIn_MCTRL.DLPEN bit carefully.*

5.28. DDRHSSPI Data Learning Pattern Register (DDRHSSPIn_DLP)

This feature is available only when DDRHSSPIn_MID.MID is 0x00000100 or 0x00000300.

The DDRHSSPI Data Learning Pattern Register is used for programming value of Data Learning Pattern supported by Serial Flash Memories.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	DLP[7]	DLP[6]	DLP[5]	DLP[4]	DLP[3]	DLP[2]	DLP[1]	DLP[0]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:8] Reserved: Reserved

[bit7:0] DLP[7:0]: Data Learning Pattern

This field is used to specify the data pattern of DLP function, and can be used only in Command Sequencer Mode.

Programming this register is possible only if DDRHSSPIn_MCTRL.DLPEN= 1 and DLP is enabled on the Serial Flash Memory side.

It is necessary to write an expected DLP value to this field, before receiving a DLP from the Serial Flash Memory.

Results of comparing DLP data received from the Serial Flash Memory with DLP Register will be stored in DDRHSSPIn_DLPSAMPLESTATUS Register.

Note:

- *When in using the DLP check function, please read the Notes of DDRHSSPIn_MCTRL.DLPEN bit carefully.*

5.29. DDRHSSPI Data Learning Pattern Sample Status Register (DDRHSSPIn_DLPSAMPLESTATUS)

This feature is available only when DDRHSSPIn_MID.MID is 0x00000100 or 0x00000300.

The DDRHSSPI Data Learning Pattern Status Register indicates status of DLP patterns recorded on each SDATA[7:0] input ports.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	DLPSMPLST7R	DLPSMPLST7C	DLPSMPLST7L	Reserved	DLPSMPLST6R	DLPSMPLST6C	DLPSMPLST6L
ACCESS_TYPE	R0,WX	R,WX	R,WX	R,WX	R0,WX	R,WX	R,WX	R,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	DLPSMPLST5R	DLPSMPLST5C	DLPSMPLST5L	Reserved	DLPSMPLST4R	DLPSMPLST4C	DLPSMPLST4L
ACCESS_TYPE	R0,WX	R,WX	R,WX	R,WX	R0,WX	R,WX	R,WX	R,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	DLPSMPLST3R	DLPSMPLST3C	DLPSMPLST3L	Reserved	DLPSMPLST2R	DLPSMPLST2C	DLPSMPLST2L
ACCESS_TYPE	R0,WX	R,WX	R,WX	R,WX	R0,WX	R,WX	R,WX	R,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	DLPSMPLST1R	DLPSMPLST1C	DLPSMPLST1L	Reserved	DLPSMPLST0R	DLPSMPLST0C	DLPSMPLST0L
ACCESS_TYPE	R0,WX	R,WX	R,WX	R,WX	R0,WX	R,WX	R,WX	R,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31] Reserved: Reserved

[bit30] DLPSMPLST7R

Bit	Description
0	Indicates 8-bit data compare match: – Sampled data on the SDATA[7] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTRGH7, WITH – DDRHSSPIn_DLP[7:0] value
1	Indicates 8-bit data compare miss: – Sampled data on the SDATA[7] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTRGH7, WITH – DDRHSSPIn_DLP[7:0] value

[bit29] DLPSMPLST7C

Bit	Description
0	Indicates 8-bit data compare match: – Sampled data on the SDATA[7] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTCNT7, WITH – DDRHSSPIn_DLP[7:0] value
1	Indicates 8-bit data compare miss: – Sampled data on the SDATA[7] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTCNT7, WITH – DDRHSSPIn_DLP[7:0] value

[bit28] DLPSMPLST7L

Bit	Description
0	Indicates 8-bit data compare match: – Sampled data on the SDATA[7] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTLFT7, WITH – DDRHSSPIn_DLP[7:0] value
1	Indicates 8-bit data compare miss: – Sampled data on the SDATA[7] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTLFT7, WITH – DDRHSSPIn_DLP[7:0] value

[bit27] Reserved: Reserved

[bit26] DLPSMPLST6R

Bit	Description
0	Indicates 8-bit data compare match: – Sampled data on the SDATA[6] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTRGH6, WITH – DDRHSSPIn_DLP[7:0] value
1	Indicates 8-bit data compare miss: – Sampled data on the SDATA[6] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTRGH6, WITH – DDRHSSPIn_DLP[7:0] value

[bit25] DLPSMPLST6C

Bit	Description
0	Indicates 8-bit data compare match: – Sampled data on the SDATA[6] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTCNT6, WITH – DDRHSSPIn_DLP[7:0] value
1	Indicates 8-bit data compare miss: – Sampled data on the SDATA[6] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTCNT6, WITH – DDRHSSPIn_DLP[7:0] value

[bit24] DLPSMPLST6L

Bit	Description
0	Indicates 8-bit data compare match: – Sampled data on the SDATA[6] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTLFT6, WITH – DDRHSSPIn_DLP[7:0] value
1	Indicates 8-bit data compare miss: – Sampled data on the SDATA[6] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTLFT6, WITH – DDRHSSPIn_DLP[7:0] value

[bit23] Reserved: Reserved

[bit22] DLPSMPLST5R

Bit	Description
0	Indicates 8-bit data compare match: – Sampled data on the SDATA[5] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTRGH5, WITH – DDRHSSPIn_DLP[7:0] value
1	Indicates 8-bit data compare miss: – Sampled data on the SDATA[5] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTRGH5, WITH – DDRHSSPIn_DLP[7:0] value

[bit21] DLPSMPLST5C

Bit	Description
0	Indicates 8-bit data compare match: – Sampled data on the SDATA[5] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTCNT5, WITH – DDRHSSPIn_DLP[7:0] value
1	Indicates 8-bit data compare miss: – Sampled data on the SDATA[5] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTCNT5, WITH – DDRHSSPIn_DLP[7:0] value

[bit20] DLPSMPLST5L

Bit	Description
0	Indicates 8-bit data compare match: – Sampled data on the SDATA[5] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTLFT5, WITH – DDRHSSPIn_DLP[7:0] value
1	Indicates 8-bit data compare miss: – Sampled data on the SDATA[5] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTLFT5, WITH – DDRHSSPIn_DLP[7:0] value

[bit19] Reserved: Reserved

[bit18] DLPSMPLST4R

Bit	Description
0	Indicates 8-bit data compare match: – Sampled data on the SDATA[4] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTRGH4, WITH – DDRHSSPIn_DLP[7:0] value
1	Indicates 8-bit data compare miss: – Sampled data on the SDATA[4] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTRGH4, WITH – DDRHSSPIn_DLP[7:0] value

[bit17] DLPSMPLST4C

Bit	Description
0	Indicates 8-bit data compare match: – Sampled data on the SDATA[4] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTCNT4, WITH – DDRHSSPIn_DLP[7:0] value
1	Indicates 8-bit data compare miss: – Sampled data on the SDATA[4] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTCNT4, WITH – DDRHSSPIn_DLP[7:0] value

[bit16] DLPSMPLST4L

Bit	Description
0	Indicates 8-bit data compare match: – Sampled data on the SDATA[4] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTLFT4, WITH – DDRHSSPIn_DLP[7:0] value
1	Indicates 8-bit data compare miss: – Sampled data on the SDATA[4] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTLFT4, WITH – DDRHSSPIn_DLP[7:0] value

[bit15] Reserved: Reserved

[bit14] DLPSMPLST3R

Bit	Description
0	Indicates 8-bit data compare match: – Sampled data on the SDATA[3] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTRGH3, WITH – DDRHSSPIn_DLP[7:0] value
1	Indicates 8-bit data compare miss: – Sampled data on the SDATA[3] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTRGH3, WITH – DDRHSSPIn_DLP[7:0] value

[bit13] DLPSMPLST3C

Bit	Description
0	Indicates 8-bit data compare match: – Sampled data on the SDATA[3] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTCNT3, WITH – DDRHSSPIn_DLP[7:0] value
1	Indicates 8-bit data compare miss: – Sampled data on the SDATA[3] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTCNT3, WITH – DDRHSSPIn_DLP[7:0] value

[bit12] DLPSMPLST3L

Bit	Description
0	Indicates 8-bit data compare match: – Sampled data on the SDATA[3] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTLFT3, WITH – DDRHSSPIn_DLP[7:0] value
1	Indicates 8-bit data compare miss: – Sampled data on the SDATA[3] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTLFT3, WITH – DDRHSSPIn_DLP[7:0] value

[bit11] Reserved: Reserved

[bit10] DLPSMPLST2R

Bit	Description
0	Indicates 8-bit data compare match: – Sampled data on the SDATA[2] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTRGH2, WITH – DDRHSSPIn_DLP[7:0] value
1	Indicates 8-bit data compare miss: – Sampled data on the SDATA[2] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTRGH2, WITH – DDRHSSPIn_DLP[7:0] value

[bit9] DLPSMPLST2C

Bit	Description
0	Indicates 8-bit data compare match: – Sampled data on the SDATA[2] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTCNT2, WITH – DDRHSSPIn_DLP[7:0] value
1	Indicates 8-bit data compare miss: – Sampled data on the SDATA[2] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTCNT2, WITH – DDRHSSPIn_DLP[7:0] value

[bit8] DLPSMPLST2L

Bit	Description
0	Indicates 8-bit data compare match: – Sampled data on the SDATA[2] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTLFT2, WITH – DDRHSSPIn_DLP[7:0] value
1	Indicates 8-bit data compare miss: – Sampled data on the SDATA[2] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTLFT2, WITH – DDRHSSPIn_DLP[7:0] value

[bit7] Reserved: Reserved

[bit6] DLPSMPLST1R

Bit	Description
0	Indicates 8-bit data compare match: – Sampled data on the SDATA[1] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTRGH1, WITH – DDRHSSPIn_DLP[7:0] value
1	Indicates 8-bit data compare miss: – Sampled data on the SDATA[1] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTRGH1, WITH – DDRHSSPIn_DLP[7:0] value

[bit5] DLPSMPLST1C

Bit	Description
0	Indicates 8-bit data compare match: – Sampled data on the SDATA[1] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTCNT1, WITH – DDRHSSPIn_DLP[7:0] value
1	Indicates 8-bit data compare miss: – Sampled data on the SDATA[1] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTCNT1, WITH – DDRHSSPIn_DLP[7:0] value

[bit4] DLPSMPLST1L

Bit	Description
0	Indicates 8-bit data compare match: – Sampled data on the SDATA[1] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTLFT1, WITH – DDRHSSPIn_DLP[7:0] value
1	Indicates 8-bit data compare miss: – Sampled data on the SDATA[1] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTLFT1, WITH – DDRHSSPIn_DLP[7:0] value

[bit3] Reserved: Reserved

[bit2] DLPSMPLST0R

Bit	Description
0	Indicates 8-bit data compare match: – Sampled data on the SDATA[0] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTRGH0, WITH – DDRHSSPIn_DLP[7:0] value
1	Indicates 8-bit data compare miss: – Sampled data on the SDATA[0] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTRGH0, WITH – DDRHSSPIn_DLP[7:0] value

[bit1] DLPSMPLST0C

Bit	Description
0	Indicates 8-bit data compare match: – Sampled data on the SDATA[0] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTCNT0, WITH – DDRHSSPIn_DLP[7:0] value
1	Indicates 8-bit data compare miss: – Sampled data on the SDATA[0] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTCNT0, WITH – DDRHSSPIn_DLP[7:0] value

[bit0] DLPSMPLST0L

Bit	Description
0	Indicates 8-bit data compare match: – Sampled data on the SDATA[0] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTLFT0, WITH – DDRHSSPIn_DLP[7:0] value
1	Indicates 8-bit data compare miss: – Sampled data on the SDATA[0] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTLFT0, WITH – DDRHSSPIn_DLP[7:0] value

Notes:

- *When in using the DLP check function, please read the Notes of DDRHSSPIn_MCTRL.DLPEN bit carefully.*
- *These status bits are set along with DDRHSSPIn_FAULTF.DLPFS bit, however, the 'clear' conditions are different as below.*
  - DDRHSSPIn_FAULTF.DLPFS bit:
    - Cleared by writing DDRHSSPIn_FAULTC.DLPFC="1".
  - DDRHSSPIn_DLPSAMPLESTATUS DLPSMPLSTxx bits:
    - Updated at the next Memory Read access (with discontinuous address).
    - 'Set'/'Clear' depends on the latest result of the DLP check in the SPI transaction.

5.30. DDRHSSPI Command Sequencer Configuration Register (DDRHSSPIn_CSCFG)

The DDRHSSPI Command Sequencer Configuration Register configures the Command Sequencer in DDRHSSPI.

This register must be programmed by the Software before enabling the Command Sequencer Mode. Attributes like Transfer Protocol, Slave Select enable/disable and Size of Serial Flash Memory interfaced with DDRHSSPI can be configured in this register.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	ITIMEREN	Reserved	Reserved	Reserved	MSEL[3]	MSEL[2]	MSEL[1]	MSEL[0]
ACCESS_TYPE	R/W	R0,W0	R0,W0	R0/W0	R/W	R/W	R/W	R/W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	Reserved	Reserved	Reserved	SSEL3EN	SSEL2EN	SSEL1EN	SSEL0EN
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	Reserved	Reserved	Reserved	DDRMODE	MBM[1]	MBM[0]	Reserved
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R0,W0
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:24] Reserved: Reserved

[bit23] ITIMEREN: Idle Timer Enable

This bit is set whether Idle Timer is enabled or not.

Bit	Description
0	ITIMER is disabled. SSEL remains asserted after transfers are finished. The programmed value in DDRHSSPIn_CSITIME Register is ignored.
1	ITIMER is enabled. SSEL is de-asserted after the time expiration, that has been programmed in DDRHSSPIn_CSITIME Register.

[bit22:20] Reserved: Reserved

[bit19:16] MSEL[3:0]: Memory Selection Bits

This field indicates the range of the address space associated with each Slave Select. It also indicates the size of each memory banks in the selected Serial Flash Memory.

This field is used by Command Sequencer Mode for two things:

- To assert one of 4 Slave Selects for the memory mapped serial transfer
- To select the size of each memory bank in the selected Serial Flash Memory

Please refer to the Section 4.2 for more details.

[bit15:12] Reserved: Reserved

[bit11] SSEL3EN: Slave Select 3 Enable

This bit enables Slave Select 3 for Command Sequencer Mode.

Bit	Description
0	Access to the Serial Flash Memory mapped on Slave Select 3 is disabled.
1	Access to the Serial Flash Memory mapped on Slave Select 3 is enabled.

[bit10] SSEL2EN: Slave Select 2 Enable

This bit enables Slave Select 2 for Command Sequencer Mode.

Bit	Description
0	Access to the Serial Flash Memory mapped on Slave Select 2 is disabled.
1	Access to the Serial Flash Memory mapped on Slave Select 2 is enabled.

[bit9] SSEL1EN: Slave Select 1 Enable

This bit enables Slave Select 1 for Command Sequencer Mode.

Bit	Description
0	Access to the Serial Flash Memory mapped on Slave Select 1 is disabled.
1	Access to the Serial Flash Memory mapped on Slave Select 1 is enabled.

[bit8] SSEL0EN: Slave Select 0 Enable

This bit enables Slave Select 0 for Command Sequencer Mode.

Bit	Description
0	Access to the Serial Flash Memory mapped on Slave Select 0 is disabled.
1	Access to the Serial Flash Memory mapped on Slave Select 0 is enabled.

[bit7:4] Reserved: Reserved

[bit3] DDRMODE: DDR Mode

This bit indicates whether DDRHSSPI is in DDR Mode or SDR Mode, in Command Sequencer Mode.

Bit	Description
0	DDRHSSPI is in Single Data Rate (SDR) Mode.
1	DDRHSSPI is in Dual Data Rate (DDR) Mode.

[bit2:1] MBM[1:0]: Multi Bit Mode

This field decides the bit width on the Serial Interface in Command Sequencer Mode.

Bits	Description
00	This setting is prohibited.
01	This setting is prohibited.
10	Serial transfer works at Quad Protocol. Read data is sampled on SDATA[3:0]. Memory instruction, address, and other control information are transmitted on SDATA[3:0].
11	This value is available only when DDRHSSPIn_MID.MID is 0x00000100 or 0x00000300. Serial transfer works at Dual Quad Protocol. Read data is sampled on SDATA[7:0]. Memory instruction, address, and other control information are transmitted on SDATA[7:0].

[bit0] Reserved: Reserved

5.31. DDRHSSPI Command Sequencer Idle Time Register (DDRHSSPIn_CSITIME)

The DDRHSSPI Command Sequencer Idle Time Register configures the idle-time-out period of Command Sequencer Mode in the DDRHSSPI.

The Software must program this timeout value before enabling the Command Sequencer Mode. This register is used only in Command Sequencer Mode.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	ITIME[15]	ITIME[14]	ITIME[13]	ITIME[12]	ITIME[11]	ITIME[10]	ITIME[9]	ITIME[8]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	-							
INITIAL_VALUE	1	1	1	1	1	1	1	1

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	ITIME[7]	ITIME[6]	ITIME[5]	ITIME[4]	ITIME[3]	ITIME[2]	ITIME[1]	ITIME[0]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	-							
INITIAL_VALUE	1	1	1	1	1	1	1	1

[bit31:16] Reserved: Reserved

[bit15:0] ITIME[15:0]: Idle Time

This field decides the idle timeout period after a memory access in Command Sequencer Mode.

The idle timeout is available when DDRHSSPIn_CSCFG.ITIMEREN is "1".

Once DDRHSSPI completes the required number of memory read access on the Serial Interface and fills Prefetch Buffer, it keeps the Slave Select asserted. If no more access to the Serial Flash Memory is detected within the idle timeout period, then DDRHSSPI de-asserts the Slave Select. The value of ITIME is based on the system clock (iHCLK).

5.32. DDRHSSPI Command Sequencer Address Extension Register (DDRHSSPIn_CSAEXT)

The DDRHSSPI Command Sequencer Address Extension Register is used to extend the usable size of Serial Flash Memory, which is mapped by the Command Sequencer.

The Software must program this register if the Address Extension feature is required, in order to virtually access a Serial Flash Memory up to 16GB. If Address Extension is not to be used, the Software must reset all bits in this register to "0".

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	AEXT[18]	AEXT[17]	AEXT[16]	AEXT[15]	AEXT[14]	AEXT[13]	AEXT[12]	AEXT[11]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	AEXT[10]	AEXT[9]	AEXT[8]	AEXT[7]	AEXT[6]	AEXT[5]	AEXT[4]	AEXT[3]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	AEXT[2]	AEXT[1]	AEXT[0]	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R/W	R/W	R/W	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:13] AEXT[18:0]: Address Extension Bits

This field is used to extend the address of Serial Flash Memory in Command Sequencer Mode.

The DDRHSSPIn_CSAEXT Register contains the 19 most significant bits [31:13] of the memory address which are generated by the Command Sequencer. The generated memory address on each Slave Select is a concatenation of the appropriate number of bits from the DDRHSSPIn_CSAEXT Register and bits from the address bus. Please refer to the Section 4.2 for more details.

If the address extension is not used, the Software should reset this field to 0x00000.

[bit12:0] Reserved: Reserved

5.33. DDRHSSPI Command Sequencer Prefetch Buffer Configuration Register (DDRHSSPIn_CSPBUFFERCFG)

The DDRHSSPI Prefetch Buffer Configuration Register configures the operation of the Prefetch Buffer. The Software can also flush the Prefetch Buffer using the PBFLSH bit in this register.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	Reserved	Reserved	PBFLSH	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W	R0,W0	R0,W0	R0,W0
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R1,W1	R1,W1	R1,W1	R1,W1
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	1	1	1	1

[bit31:20] Reserved: Reserved

[bit19] PBFLSH: Prefetch Buffer Flush

This bit can be used by the Software to flush the Prefetch Buffer and RX Shift Register.

Bit	Description
0	Writing "0" has no effect.
1	Writing "1" flushes the Prefetch Buffer and RX Shift Register. Please do not write "1" to this bit, while DDRHSSPIn_CSPBUFFERSTATUS.SSACTIVE bit is "1".

The read value is "0".

[bit18:4] Reserved: Reserved

[bit3:0] Reserved: Reserved

5.34. DDRHSSPI Command Sequencer Prefetch Buffer Status Register (DDRHSSPIn_CSPBUFFERSTATUS)

The DDRHSSPI Command Sequencer Prefetch Buffer Status Register contains the status bits like current fill-level of the Prefetch Buffer or whether the Serial Flash Memory access is in progress or not, i.e. Slave Select is still asserted. This register is used only in Command Sequencer Mode.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	SSACTIVE
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	Reserved	PB LEVEL[5]	PB LEVEL[4]	PB LEVEL[3]	PB LEVEL[2]	PB LEVEL[1]	PB LEVEL[0]
ACCESS_TYPE	R0,WX	R0,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:17] Reserved: Reserved

[bit16] SSACTIVE: Slave Select Active

This flag indicates whether a Slave Select is active.

Bit	Description
0	Slave Select is not active.
1	Slave Select is active.

[bit15:6] Reserved: Reserved

[bit5:0] PBLEVEL[5:0]: Current Fill Level of Prefetch Buffer

This field indicates the current fill level of the Prefetch Buffer.

CHAPTER 53: System SRAM Module (SRAM_IF)

This chapter explains the functions and operations of System SRAM module (SRAM_IF).

1. "Outline of SRAM_IF"
2. “Configuration and Block Diagram”
3. "Operation of SRAM_IF"
4. "SRAM_IF Register Set"
5. "How to use SRAM_IF"

SRAMIF-TXXPT03P01R01L06-E1-XX

1. Outline of SRAM_IF

The System SRAM module interacts with System SRAM Macro. It has an ECC logic which can generate interrupts during bit-error detection in SRAM read data. The ECC logic is made testable with the option of injecting bit errors into SRAM read data.

Feature List

- Integrates synchronous static RAM.
- Implements Single-bit error correcting ECC logic
- Generates Interrupt during Single-bit error detection.
- Sends error response on double-bit error detection.
- Error injection by changing both the data bits and ECC bits read from the SRAM to test the ECC functionality.
- Wait state insertion during the SRAM read and write operation to allow adjustment of the memory access time to higher operating frequencies.
- Power management capability to support power gating.
- Parallel read and write either to two different SRAM banks or to a single SRAM bank and register set is supported.
- LRG (Least Recently Granted) priority scheme is used in case of simultaneous read and write requests to the same SRAM bank or register set.

Abbreviations

This section lists the terms and abbreviations used in this chapter.

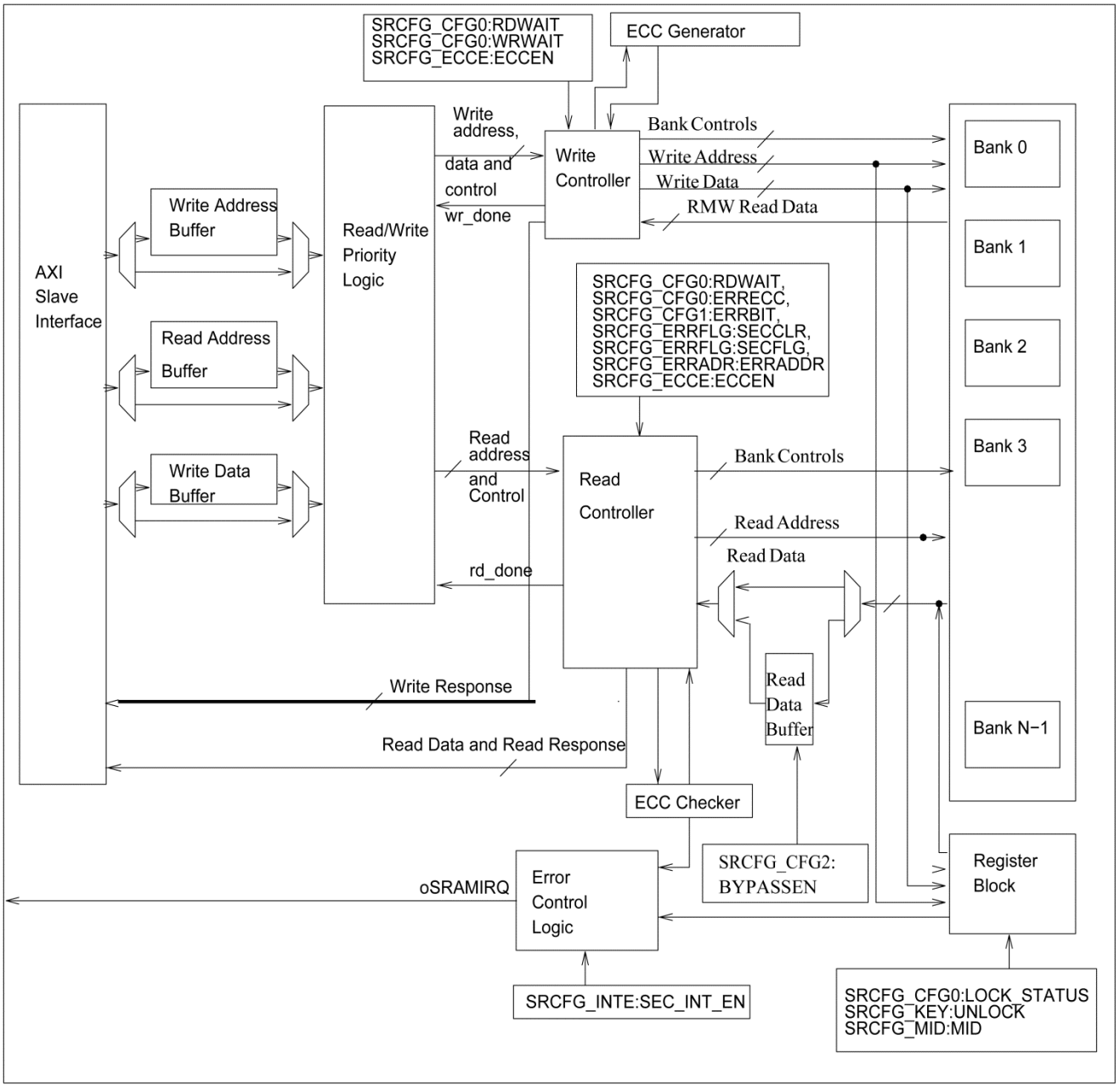
Table 1-1 Terms and Abbreviations

Term	Meaning
SRAM	Static Random Access Memory
ECC	Error Checking and Correcting
AXI	Advanced eXtensible Interface

2. Configuration and Block Diagram

The Figure 2-1 shows the block diagram of SRAM_IF.

Figure 2-1 SRAM_IF Block Diagram



3. Operation of SRAM_IF

This chapter describes the operation of SRAM_IF.

Description of Features Supported

WAIT States

Wait states can be configured for this module by writing to the SRCFG_CFG0 register through AXI Slave interface in privileged mode and only when the SRCFG_CFG0:LOCK_STATUS bit is Zero. An error response will be generated if there is a write to SRCFG_CFG0:RDWAIT or to SRCFG_CFG0:WRWAIT when SRCFG_CFG0:LOCK_STATUS is 1. The range of the wait states that can be configured range from 0 to 3. By default, maximum number of wait states (i.e., 3 wait states) is configured, which means the data will be latched in the fourth clock cycle after the address is given on memory interface for read transaction. Similarly, the write transaction will be on memory interface for configured number of clock cycles in SRCFG_CFG0:WRWAIT.

The read-data from the SRAM is used only after the configured number of wait states is completed. Once the WAIT states are configured, they continue to apply until they are again changed by writing into the configuration registers.

The SRCFG_CFG0:RDWAIT is used by write controller in case of RMW (read-modified writes) and by read controller for read transactions. These blocks will generate the wait states as per configuration and latch the memory data accordingly.

ECC Operation

When ECC checks are enabled, the checks for ECC will be done for every read data from SRAM. In this case, the single-bit errors would be detected and corrected while double-bit errors would be detected and an error response would be given to the master. Similarly, ECC will be calculated for every write transaction if ECC is enabled.

Partial writes are the write transactions that writes only part of the memory address (8-bit or 16-bit) and ECC for these writes will be calculated after merging the new data with old valid data in memory (other than the 8-bit or 16-bit write data). This operation is called read-modified write (RMW). The memory data will be read for the partial write address and will be written after merging the partial write data and recalculating the ECC.

Disabling the ECC will improve the performance for partial writes as it will save the clock cycles which would have been used to read the data from memory and recalculating the ECC. The write will be done in one cycle when ECC is disabled.

ECC generator calculates the ECC for write transactions and ECC checker checks the ECC for read transactions.

ECC Self-Test

This feature is implemented to test the 32-bit ECC logic. This is done by injecting bit-flips either at only the data or at the ECC data or at both that is read from the SRAM.

This is achieved by writing the required data into the SRAM_IF configuration register when the SRCFG_CFG0:LOCK_STATUS bit is Zero. The access to register space when SRCFG_CFG0:LOCK_STATUS=1 will generate an error response. If the errors are to be injected in the data then 32-bit ERRBIT data has to be written to SRCFG_CFG1 and if the errors are to be injecting in the ECC data then 7-bit ERRECC data has to be written to SRCFG_CFG0 register.

Note:

- *The data (32-bit data and 7-bit ECC data) read from RAM locations are always flipped for positions in the configuration registers SRCFG_CFG0:ERRECC and SRCFG_CFG1:ERRBIT having 1. So if this feature is not required, the register values have to be Zero. However, the default value is Zero.*

Error Generation

Error response will be generated when SRAM_IF configuration registers are accessed before unlocking condition or write a value other than the UNLOCK or LOCK value to the SRCFG_KEY register or non-privileged access to the SRAM_IF configuration registers. Also, access to the reserved space of the module generates this error response. Refer: Figure 5-1 for flow chart on usage of SRCFG_KEY:UNLOCK to access the register space.

Interrupt Generation

SRAM_IF generates interrupts in the following conditions:

ECC Checker has detected a Single-bit error in the SRAM Read data (SRCFG_ERRFLG:SECFLG is Set) and SRCFG_INTE:SEC_INT_EN bit is Set.

SRAM_IF also stores the address of the SRAM location where Single-bit error has occurred in SRCFG_ERRADR:ERRADDR register.

Parallel Read/Write Access and Priority Scheme

The AXI interface of SRAM_IF is shared for memory and register accesses. SRAM_IF supports parallel read and write access either to two different SRAM banks or to a single SRAM bank and register set. In case of simultaneous read and write requests to the same SRAM bank or register set, LRG (Least Recently Granted) priority scheme is used to grant one of these transactions. If the last granted request was a read, then write request will be given priority and vice-versa.

4. SRAM_IF Register Set

All registers in SRAM_IF are explained in this section.

SRAM_IF Registers

Table 4-1 Register Map

Offset	Register Name / Initial Value								Block
	+7	+6	+5	+4	+3	+2	+1	+0	
0x00000000	SRCFG_CFG1 00000000_00000000_00000000_00000000				SRCFG_CFG0 00000011_00000011_00000001_00000000				SRAM_IF
0x00000008	SRCFG_KEY 00000000_00000000_0000000_00000000				SRCFG_CFG2 00000000_00000000_00000000_00000000				
0x00000010	SRCFG_INTE 00000000_00000000_00000000_00000000				SRCFG_ERRFLG 00000000_00000000_00000000_00000000				
0x00000018	Reserved				SRCFG_ECCE 00000000_00000000_00000000_00000001				
0x00000020	SRCFG_MID 00000000_00000100_00000011_00000000				SRCFG_ERRADR 00000000_00000000_00000000_00000000				

4.1. SRAM_IF Configuration Register 0 (SRCFG_CFG0)

The SRAM_IF Configuration Register 0 contains 7-bit ERR ECC bit data which is used to inject errors in the ECC data read from SRAM. It also contains 2-bit WAIT state values and a LOCK_STATUS bit.

Bit	31	30	29	28	27	26	25	24
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	RDWAIT[1]	RDWAIT[0]
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W	R/W
Protection Attribute	WPS							
Initial Value	0	0	0	0	0	0	1	1

Bit	23	22	21	20	19	18	17	16
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	WRWAIT[1]	WRWAIT[0]
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W	R/W
Protection Attribute	WPS							
Initial Value	0	0	0	0	0	0	1	1

Bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	LOCK_STATUS
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R,WX
Protection Attribute	WPS							
Initial Value	0	0	0	0	0	0	0	1

Bit	7	6	5	4	3	2	1	0
Field	Reserved	ERRECC[6]	ERRECC[5]	ERRECC[4]	ERRECC[3]	ERRECC[2]	ERRECC[1]	ERRECC[0]
R/W Attribute	R0,WX	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	WPS							
Initial Value	0	0	0	0	0	0	0	0

[bit31:26] Reserved: Reserved Bits

[bit25:24] RDWAIT[1:0]: Read Data Wait State Value

These 2 bits are used to configure the number of data wait states for SRAM read transaction.

Bits	Description
00	0 wait State
01	1 wait State
10	2 wait States
11	3 wait States (default)

[bit23:18] Reserved: Reserved Bits

[bit25:24] WRWAIT[1:0]: Write Data Wait State Value

These 2 bits are used to configure the number of data wait states for SRAM write transaction.

Bits	Description
00	0 wait State
01	1 wait State
10	2 wait States
11	3 wait States (default)

[bit15:9] Reserved: Reserved Bits

[bit8] LOCK_STATUS: SRAM_IF Lock Status

This bit provides locked or unlocked status of SRAM_IF.

Bit	Description
0	SRAM_IF is unlocked and therefore the configuration registers are accessible for write or read
1	SRAM_IF is locked and therefore the configuration registers are not accessible for write (default)

[bit7] Reserved: Reserved Bit

[bit6:0] ERRECC[6:0]: ECC ERRBIT Value

Each bit set in this register will cause an emulated bit-flip in the corresponding bit of read-data ECC bits.

Note:

- *SRCFG_CFG0:ERRECC[6:0] bits have no effect when ECC is disabled (SRCFG_ECCE:ECCEN = 0).*

4.2. SRAM_IF Configuration Register 1 (SRCFG_CFG1)

The SRAM_IF Configuration Register 1 contains the 32-bit ERRBIT data used to corrupt the data read from the SRAM.

Bit	31	30	29	28	27	26	25	24
Field	ERRBIT[31]	ERRBIT[30]	ERRBIT[29]	ERRBIT[28]	ERRBIT[27]	ERRBIT[26]	ERRBIT[25]	ERRBIT[24]
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	WPS							
Initial Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	ERRBIT[23]	ERRBIT[22]	ERRBIT[21]	ERRBIT[20]	ERRBIT[19]	ERRBIT[18]	ERRBIT[17]	ERRBIT[16]
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	WPS							
Initial Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	ERRBIT[15]	ERRBIT[14]	ERRBIT[13]	ERRBIT[12]	ERRBIT[11]	ERRBIT[10]	ERRBIT[9]	ERRBIT[8]
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	WPS							
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	ERRBIT[7]	ERRBIT[6]	ERRBIT[5]	ERRBIT[4]	ERRBIT[3]	ERRBIT[2]	ERRBIT[1]	ERRBIT[0]
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	WPS							
Initial Value	0	0	0	0	0	0	0	0

[bit31:0] ERRBIT[31:0]: ERRBIT Value

Each bit set in this register causes an emulated bit-flip in the corresponding bit of 32-bit read data.

Notes:

- *All subsequent accesses will fail in case ECC injection is configured.*
- *SRCFG_CFG1:ERRBIT[31:0] bits have no effect when ECC is disabled (SRCFG_ECCE:ECCEN = 0).*

4.3. SRAM_IF Configuration Register 2 (SRCFG_CFG2)

The SRAM_IF Configuration Register 2 contains the 1-bit BYPASSEN used to enable the Read Data Buffer (RDB) Bypass path of data read from the SRAM.

Bit	31	30	29	28	27	26	25	24
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
Protection Attribute	WPS							
Initial Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
Protection Attribute	WPS							
Initial Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
Protection Attribute	WPS							
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	BYPASSEN
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W
Protection Attribute	WPS							
Initial Value	0	0	0	0	0	0	0	0

[bit31:1] Reserved: Reserved Bits

[bit0] BYPASSEN: RDB Bypass Disable or Enable

RDB Bypass will be enabled in case this bit is 1.

Bit	Description
0	RDB Bypass path is disabled (default)
1	RDB Bypass path is enabled

Notes:

- *Enabling RDB Bypass path (SRCFG_CFG2:BYPASSEN = 1) will reduce read latency since the read data from SRAM will be directly output to the Bus interface without buffering in RDB (Refer Figure 2-1).*
- *Although the default value of this bit is 0, it is not required to configure this bit to 0.*

4.4. SRAM_IF Unlock/Lock Key Register (SRCFG_KEY)

The SRAM_IF Unlock/Lock Key Register has to be written to Unlock or Lock all accesses to SRAM_IF configuration registers.

Bit	31	30	29	28	27	26	25	24
Field	UNLOCK [31]	UNLOCK [30]	UNLOCK [29]	UNLOCK [28]	UNLOCK [27]	UNLOCK [26]	UNLOCK [25]	UNLOCK [24]
R/W Attribute	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
Protection Attribute	WP							
Initial Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	UNLOCK [23]	UNLOCK [22]	UNLOCK [21]	UNLOCK [20]	UNLOCK [19]	UNLOCK [18]	UNLOCK [17]	UNLOCK [16]
R/W Attribute	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
Protection Attribute	WP							
Initial Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	UNLOCK [15]	UNLOCK [14]	UNLOCK [13]	UNLOCK [12]	UNLOCK [11]	UNLOCK [10]	UNLOCK [9]	UNLOCK [8]
R/W Attribute	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
Protection Attribute	WP							
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	UNLOCK [7]	UNLOCK [6]	UNLOCK [5]	UNLOCK [4]	UNLOCK [3]	UNLOCK [2]	UNLOCK [1]	UNLOCK [0]
R/W Attribute	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
Protection Attribute	WP							
Initial Value	0	0	0	0	0	0	0	0

[bit31:0] UNLOCK[31:0]: SRAM_IF Unlock or Lock Key

If the correct unlock key is written into this register, the SRCFG_CFG0:LOCK_STATUS is set to '0' (unlocked) and all the configuration registers can be written. If the correct lock key is written into this register, the SRCFG_CFG0:LOCK_STATUS is set to '1' (locked) and the configuration registers will not be accessible. Illegal access to static configuration registers or writing value other than lock or unlock value to this register will cause protection error, an error response will be given back.

Locked registers are as follows.

- SRCFG_CFG0
- SRCFG_CFG1
- SRCFG_CFG
- SRCFG_ERRFLG
- SRCFG_INTE

- SRCFG_ECCE
- SRCFG_ERRADR

Note:

– *The "Unlock/Lock" Key values are specified as followings.*

SRAM_IF UNLOCK_VALUE = 0x5ECC551F

SRAM_IF LOCK_VALUE = 0x551FB10C

4.5. SRAM_IF Error Flag Register (SRCFG_ERRFLG)

The SRAM_IF Error Flag Register has the flags to indicate Single errors in SRAM Read data. It also contains the bits to clear these flags.

Bit	31	30	29	28	27	26	25	24
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
Protection Attribute	WPS							
Initial Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
Protection Attribute	WPS							
Initial Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	SECCLR
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,W
Protection Attribute	WPS							
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	SECFLG
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R,WX
Protection Attribute	WPS							
Initial Value	0	0	0	0	0	0	0	0

[bit31:9] Reserved: Reserved Bits

[bit8] SECCLR: Single-Bit Error Flag Clear

If this bit is 1, it clears the flag SRCFG_ERRFLG:SECFLG. This bit always returns 0 on read.

[bit7:1] Reserved: Reserved Bits

[bit0] SECFLG: Single-Bit Error Detection Flag

This bit is set if single-bit error is detected in SRAM read data.

This also causes a maskable interrupt to be sent to CPU. If SRCFG_INTE:SEC_INT_EN bit is 0, no interrupt occurs.

Note:

- *ECC Check is performed in parallel to both the upper and lower 32-bit of SRAM read data. Irrespective of the access size, SRCFG_ERRFLG:SECFLG is set when Single-bit error is detected in either the lower or the upper 32-bit of SRAM read data.*

4.6. SRAM_IF Interrupt Enable Register (SRCFG_INTE)

The SRAM_IF Interrupt Enable Register contains the Interrupt Enable bits.

Bit	31	30	29	28	27	26	25	24
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
Protection Attribute	WPS							
Initial Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
Protection Attribute	WPS							
Initial Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
Protection Attribute	WPS							
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	SEC_INT_EN
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W
Protection Attribute	WPS							
Initial Value	0	0	0	0	0	0	0	0

[bit31:1] Reserved: Reserved Bits

[bit0] SEC_INT_EN: Single-Bit Error Interrupt Enable Bit

Bit	Description
0	Interrupt will not be generated even if SRCFG_ERRFLG:SECFLG bit is set (default)
1	Interrupt will be generated if SRCFG_ERRFLG:SECFLG bit is set (single error detection)

4.7. SRAM_IF ECC Enable Register (SRCFG_ECCE)

The SRAM_IF ECC Enable Register contains the ECC enable bits.

Bit	31	30	29	28	27	26	25	24
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
Protection Attribute	WPS							
Initial Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
Protection Attribute	WPS							
Initial Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
Protection Attribute	WPS							
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	ECCEN
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W
Protection Attribute	WPS							
Initial Value	0	0	0	0	0	0	0	1

[bit31:1] Reserved: Reserved Bits

[bit0] ECCEN: ECCEN Value

This bit enables or disables ECC logic in SRAM_IF. This bit is writable only once by software and all further writes to this byte will be signaled as protection error. An error response will be given back in that scenario.

Bit	Description
0	ECC logic is disabled
1	ECC logic is enabled (default)

4.8. SRAM_IF Error Address Register (SRCFG_ERRADR)

The SRAM_IF Error Address Register contains the absolute SRAM location address value where Single-bit error has occurred. Always ECC check is performed in parallel to both upper and lower 32-bit of SRAM read data and independent of the error location, the 64- bit aligned address is stored in this register.

Bit	31	30	29	28	27	26	25	24
Field	ERRADDR [31]	ERRADDR [30]	ERRADDR [29]	ERRADDR [28]	ERRADDR [27]	ERRADDR [26]	ERRADDR [25]	ERRADDR [24]
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	ERRADDR [23]	ERRADDR [22]	ERRADDR [21]	ERRADDR [20]	ERRADDR [19]	ERRADDR [18]	ERRADDR [17]	ERRADDR [16]
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	ERRADDR [15]	ERRADDR [14]	ERRADDR [13]	ERRADDR [12]	ERRADDR [11]	ERRADDR [10]	ERRADDR [9]	ERRADDR [8]
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	ERRADDR [7]	ERRADDR [6]	ERRADDR [5]	ERRADDR [4]	ERRADDR [3]	ERRADDR [2]	ERRADDR [1]	ERRADDR [0]
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R0,WX	R0,WX	R0,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

[bit31:0] ERRADDR[31:0]: ECC Error Address

This read-only register contains the absolute SRAM location address where single-bit ECC error has occurred. It will be updated with the latest SRAM address location where single-bit ECC error has occurred.

4.9. SRAM_IF Module Identification Register (SRCFG_MID)

The SRAM_IF Module Identification register contains the SRAM_IF module ID.

Bit	31	30	29	28	27	26	25	24
Field	MID[31]	MID[30]	MID[29]	MID[28]	MID[27]	MID[26]	MID[25]	MID[24]
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	MID[23]	MID[22]	MID[21]	MID[20]	MID[19]	MID[18]	MID[17]	MID[16]
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R1,WX	R0,WX	R0,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	1	0	0

Bit	15	14	13	12	11	10	9	8
Field	MID[15]	MID[14]	MID[13]	MID[12]	MID[11]	MID[10]	MID[9]	MID[8]
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R1,WX	R1,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	1	1

Bit	7	6	5	4	3	2	1	0
Field	MID[7]	MID[6]	MID[5]	MID[4]	MID[3]	MID[2]	MID[1]	MID[0]
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

[bit31:0] MID[31:0]: Module ID

This read-only register gives the unique module identification number of SRAM_IF module.

The unique module ID number identifies the version of the SRAM_IF module used in the MCU.

Refer to the specific device datasheet for the module identification number of SRAM_IF.

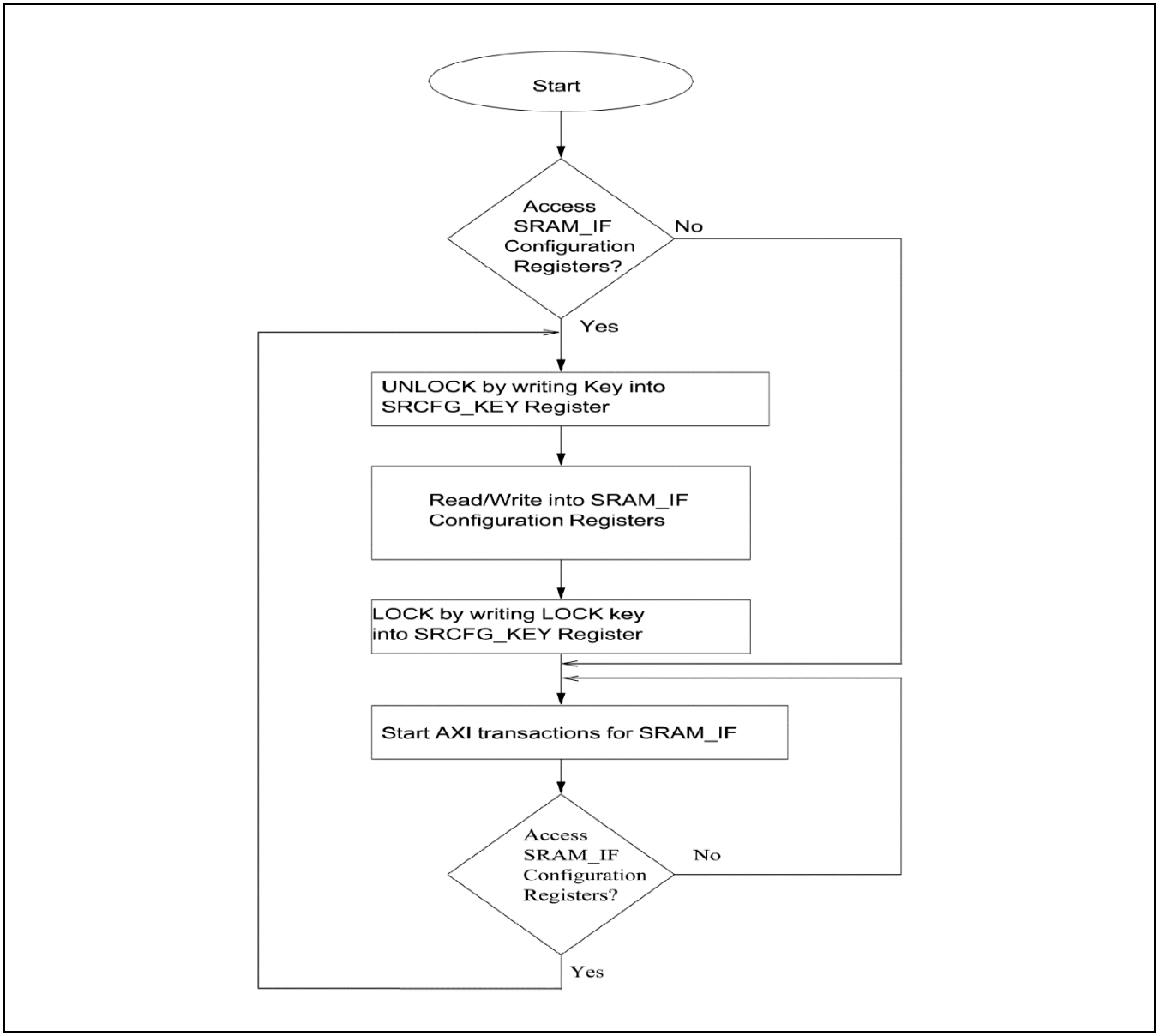
5. How to use SRAM_IF

The following flowchart shows the sequence of steps to be followed to configure the SRAM_IF Registers.

Typical SRAM_IF Configuration Flow

Figure 5-1 shows the steps to write to SRAM_IF Configuration Registers.

Figure 5-1 Configuration Flowchart



CHAPTER 54: Bit-Band Unit

This chapter provides an overview and notes on bit-band unit.

.

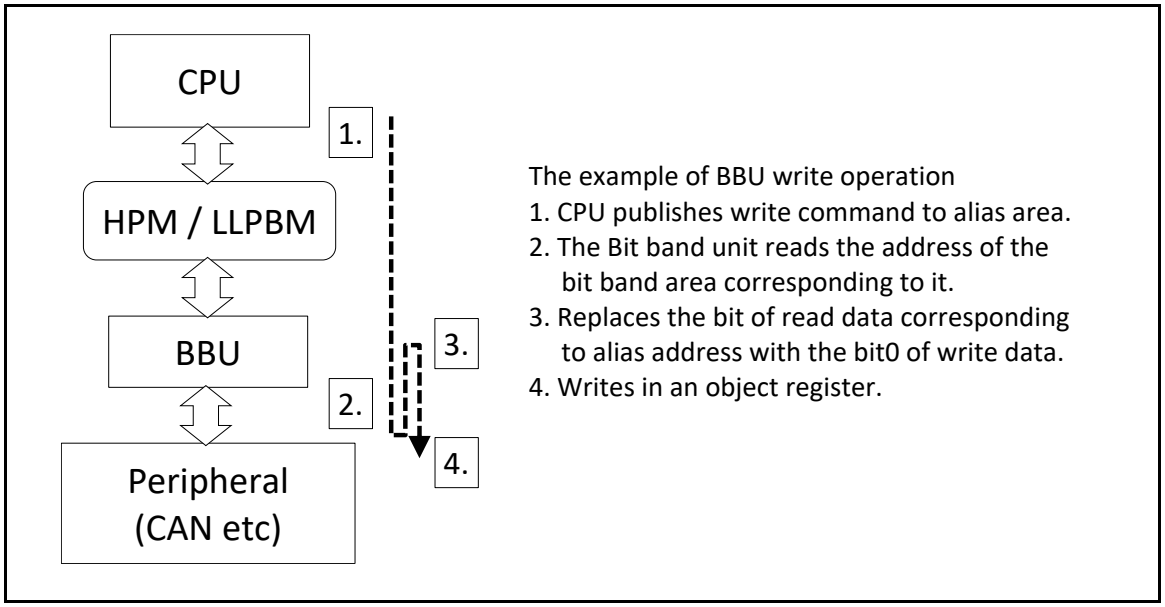
1. **Overview**
2. **Explanation of Operation**

1. Overview

This section shows the features of bit-band unit.

The memory map of this device has an address area called "bit-band alias area" and an address area named "bit-band area". Bit-band unit has a function, by writing 1 byte in bit-band alias area, to set or clear 1 bit corresponding to bit-band area. Also, by reading 1 byte in bit-band alias area, the corresponding 1 bit in bit-band area is read to bit[0] of the read value through bit-band unit.
Bit-band area is mapped over resource area. To set or clear only a specific bit in the register, it is possible to process bit manipulation faster than reading and writing the register.

Figure 1-1 Bit-Band Alias Unit Operations (Write)



2. Explanation of Operation

This section provides an explanation of bit-band unit operation.

Bit-band Area and Bit-band Alias Area

The relationship between bit-band alias area and bit-band area is shown in Figure 2-1. 1 byte of bit-band alias area corresponds to 1 bit of bit-band area. The correspondence between bit-band alias area and bit-band area is shown in Table 2-1.

Figure 2-1 Bit-Band Alias Area and Bit-band Area

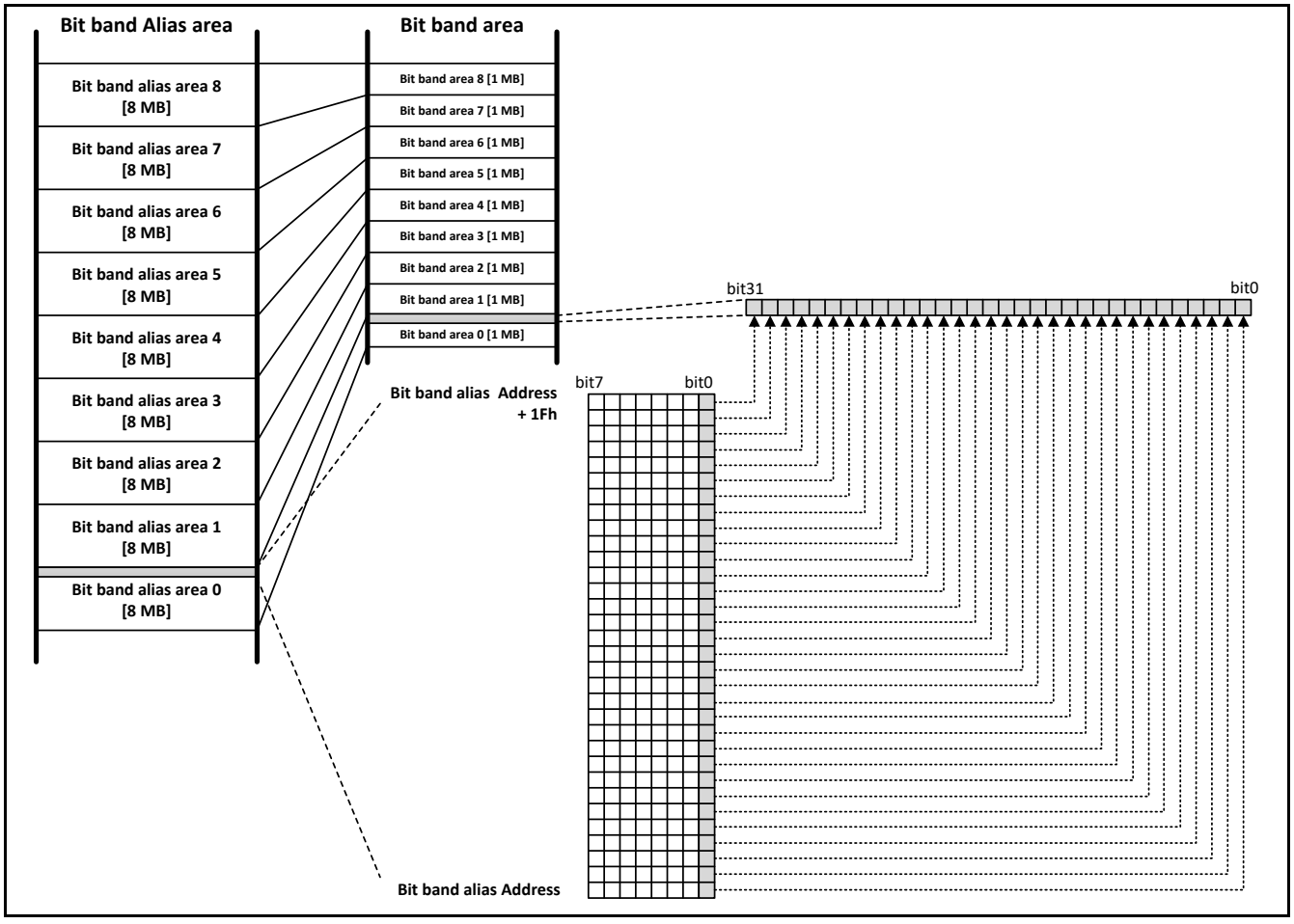


Table 2-1 Correspondence between Bit-band Alias Area and Bit-band Area

Bit-band Area	Group	Bit-band Alias Area
0xB030_0000 to 0xB031_FFFF	SYSC1 Group	0xB100_0000 to 0xB10F_FFFF
0xB040_0000 to 0xB041_FFFF	Memory & Config Group	0xB110_0000 to 0xB11F_FFFF
0xB060_0000 to 0xB06F_FFFF	MCU Config Group	0xB080_0000 to 0xB0FF_FFFF
0xB470_0000 to 0xB47F_FFFF	Common Peripheral Group #2	0xB700_0000 to 0xB77F_FFFF
0xB480_0000 to 0xB487_FFFF	Common Peripheral Group #0	0xB780_0000 to 0xB7BF_FFFF
0xB488_0000 to 0xB48F_FFFF	Common Peripheral Group #1	0xB7C0_0000 to 0xB7FF_FFFF
0xB490_0000 to 0xB497_FFFF	Common Peripheral Group#0 (MCAN)	0xB4C0_0000 to 0xB4FF_FFFF

Bit Set and Bit Clear by Bit-band Unit

If "1" is written in bit[0] in address in bit-band alias area,
the corresponding bit in bit-band area is set to "1". Also, if "0" is written, corresponding bit in bit-band area is cleared to "0". The value of bit[7:1] is not affected.

Notes:

- *In the following case, writing is ignored and a bus error is returned.*
  - When a value other than byte size is written in bit-band alias area
- *Following registers do not support write operations via bit-band unit. If write operations via bit-band unit are performed on them, either a bus error will be returned or an incorrect value will be written.*
  - Registers to which byte accesses are prohibited
  - Registers protected by Keycode

Reading Bit by Bit-band Unit

The corresponding bit of bit-band area is read in bit[0] by reading using the corresponding address of bit-band in bit-band alias domain.

Notes:

- *In the following case, a bus error is returned.*
  - When a value other than byte size is read for bit-band alias area
- *Following registers do not support read operations via bit-band unit. If read operations via bit-band unit are performed on them, either a bus error will be returned or an incorrect value will be read.*
  - Registers to which byte accesses are prohibited

Details of Address Correspondence

The address of bit-band alias single bit manipulation area is calculated using the following formula

(1) + 8 * [(3) - (2)] + (4)

(1) …Base address of bit-band alias single bit manipulation area corresponding to each group
(2) …Base address of bit-band area of each group
(3) …Register address to manipulate in bit-band area for each group
(4) …Bit position to manipulate

Table 2-2 shows an example of SYSC1 Group. Groups other than SYSC1 Group are calculated in the same way.

Table 2-2 Details of Address Correspondence (An Example of SYSC1 Group)

Register Address of Bit-band Area (SYSC1 Group)	Bit Position	Register Address of Bit-band Alias Single Bit Manipulation Area
0xB0300000 (2)	bit[0]	0xB1000000 (1)
	bit[1]	0xB1000001
	bit[2]	0xB1000002
	...	
0xB0300010 (3)	bit[0]	0xB1000080
	bit[1]	0xB1000081
	bit[2]	0xB1000082
	bit[3] (4)	0xB1000083 (=(1)+8*((3)-(2))+(4))
	...	

CHAPTER 55: APPENDIX: Major Changes

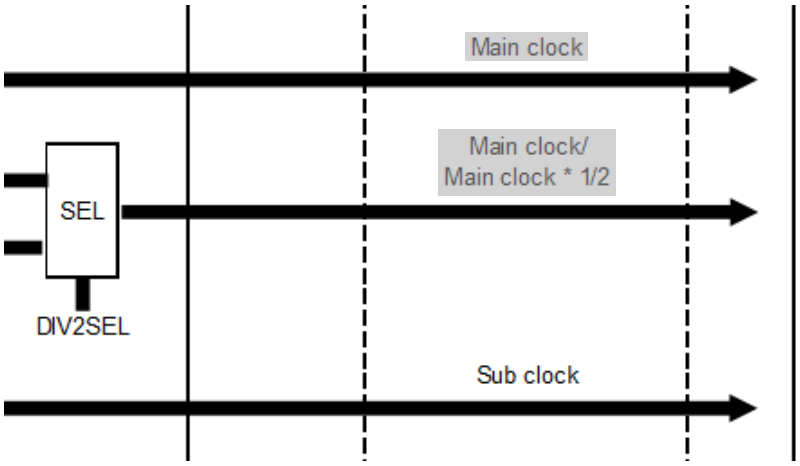
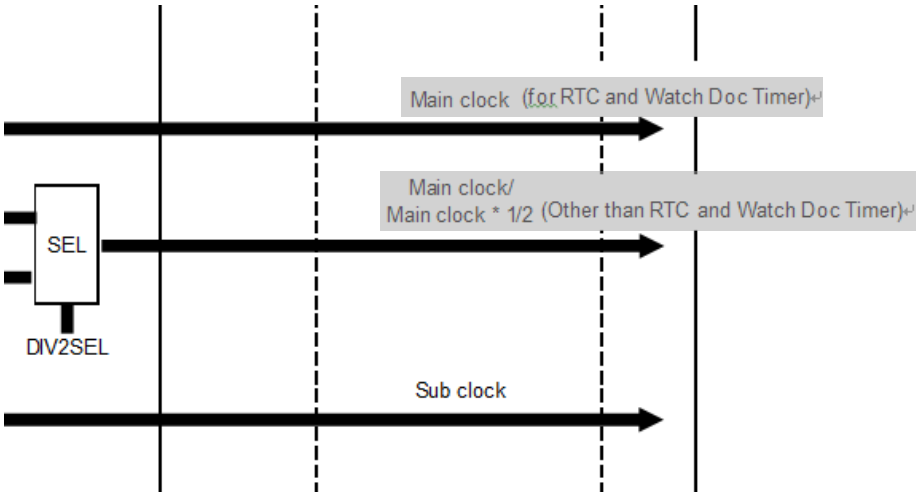
1.1 Major Changes

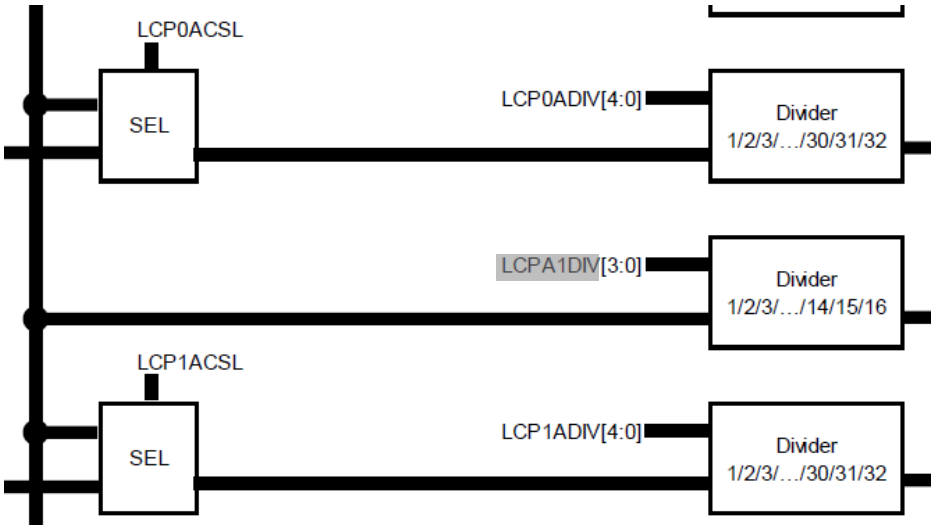
Page	Section	Change Results
Original document code: From MN708-00006-1v0-E to MN708-00006-2v0-E		
43	CHAPTER1: Platform Overview	Error) This chapter explains brief overview of the B/B-PG platform. Correct) This chapter explains brief overview of the B/B platform.

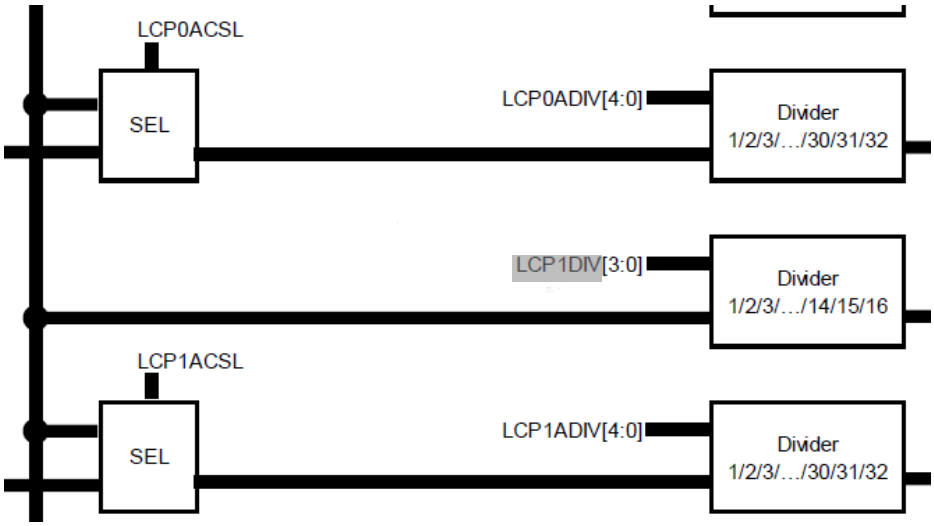
Page	Section	Change Results
44	CHAPTER 1:Platform Overview 1. Overview	Error) This section gives a brief overview of B/B-PG platform. B/B-PG is a latest effort of the MCU platform which targets instrument cluster, graphics controller, gateway and body controller application area. To minimize design efforts of the target MCU, B/B-PG offers rich and competitive feature sets like power saving features and IP portfolio, easy-to-use parameter option to select them, and verification suite which proves its quality. Correct) This section gives a brief overview of B/B platform. B/B is a latest effort of the MCU platform which targets instrument cluster, graphics controller, gateway and body controller application area. To minimize design efforts of the target MCU, B/B offers rich and competitive feature sets like power saving features and IP portfolio, easy-to-use parameter option to select them, and verification suite which proves its quality.

Page	Section	Change Results
45	CHAPTER 1:Platform Overview 1. Overview	Error) ■ Debug assist features Many of peripherals can be forcedly stopped by entering debug mode. Dedicated Reload Times can capture DMA control signal events like DREQ. EICU can capture pin status of peripherals. Correct) ■ Debug assist features Many of peripherals can be forcedly stopped by entering debug mode. Dedicated Reload Times can capture DMA control signal events like DREQ. EICU can capture pin status of peripherals. ・About the resource stop function which is being debugged. When debugging a program, it's possible to suspend the following function separately. (1) DMA (2) RLT (3) RTC (4) HWDT (5) SWDT (6) TPU (7) SCT
46	CHAPTER 1:Platform Overview 1. Overview 1.1. Function Overview	Error) This section gives a brief overview of each function in B/B-PG. Correct) This section gives a brief overview of each function in B/B.
50	CHAPTER 1:Platform Overview 3. Configuration	Error) This section shows the block diagram of B/B-PG platform. Correct) This section shows the block diagram of B/B platform.
54	CHAPTER 1:Platform Overview 5. Memory Map	Error) This section shows memory map of B/B-PG platform. Correct) This section shows memory map of B/B platform.
57	CHAPTER 1:Platform Overview 6. I/O Map	Error) This section shows I/O map of B/B-PG platform. Correct) This section shows I/O map of B/B platform.

Page	Section	Change Results
117	CHAPTER4: Reset 3.Operational Description 3.3.Reset Sequence	Error) Note: −At Power Saving State (It is *1 in figure), the lying reset factor (hard software trigger reset, software reset, and software debugger reset) doesn't generate CPU. Reset Sequence 1.Reset cause occurred Reset cause is captured and will be held until a reset is issued inside the device. 2.Waiting for the clock stop Before a reset, clock source is stopped for the resets which need RAM to be secured. Timeover will occur if the clock does not stop in the specified time. Correct) Note: −At Power Saving State (It is *1 in figure), the lying reset factor (hard software trigger reset, software reset, and software debugger reset) doesn't generate CPU. −RAM inside PF isn't secured by a reset.(TCRAM,BackupRAM,System RAM) Reset Sequence 1.Reset cause occurred Reset cause is captured and will be held until a reset is issued inside the device. 2.Waiting for the clock stop Before a reset, clock source is stopped for the resets which need RAM to be secured. Timeover will occur if the clock does not stop in the specified time.

Page	Section	Change Results
159	CHAPTER 5:Clock System 2. Configuration and Block Diagram 2.2. Clock Generator Configuration and Block Diagram	Error) Figure 2-2 Clock Generator Block Diagram  Correct) Figure 2-2 Clock Generator Block Diagram 

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>163</td><td>CHAPTER 5:Clock System 2. Configuration and Block Diagram</td><td>Error)


Correct)

</td></tr>
<tr><td>236</td><td>CHAPTER 6:Low Power Consumption 3. Explanation of Operation 3.4. Profile</td><td>Error)


Correct)

</td></tr>
</table>

Page	Section	Change Results
465	CHAPTER 6:Low Power Consumption 5. Registers 5.11.7. RUN Clock Divider Register 0 (SYSC1_RUNCKDIVR0)	Error) **[bit20:16] TRCDIV: TRC Clock Divider Setting Bits** These bits set the division ratio of the TRC clock from the source clock (clock domain TRC). bit20:16 / Description 0000 / No division 0001 / Divided by 2 0010 / Divided by 3 0011 / Divided by 4 . . . . / . . . 1101 / Divided by 14 1110 / Divided by 15 1111 / Divided by 16 Correct) **[bit20:16] TRCDIV: TRC clock divider setting bits** These bits set the division ratio of the TRC clock from the source clock (clock domain TRC). Bit20:16 / Description 00000 / No division 00001 / Divided by 2 00010 / Divided by 3 00011 / Divided by 4 ..... / ..... 11101 / Divided by 30 11110 / Divided by 31 11111 / Divided by 32

Page	Section	Change Results
516	CHAPTER 6:Low Power Consumption 5. Registers 5.12.7. PSS Clock Divider Register 0 (SYSC1_PSSCKDIVR0)	Error) **[bit20:16] TRCDIV: TRC Clock Divider Setting Bits** These bits set the division ratio of the TRC clock from the source clock (clock domain TRC). bit20:16 / Description 0000 / No division 0001 / Divided by 2 0010 / Divided by 3 0011 / Divided by 4 . . . . / . . . . 1101 / Divided by 14 1110 / Divided by 15 1111 / Divided by 16 Correct) **[bit20:16] TRCDIV: TRC clock divider setting bits** These bits set the division ratio of the TRC clock from the source clock (clock domain TRC). Bit20:16 / Description 00000 / No division 00001 / Divided by 2 00010 / Divided by 3 00011 / Divided by 4 ..... / ..... 11101 / Divided by 30 11110 / Divided by 31 11111 / Divided by 32

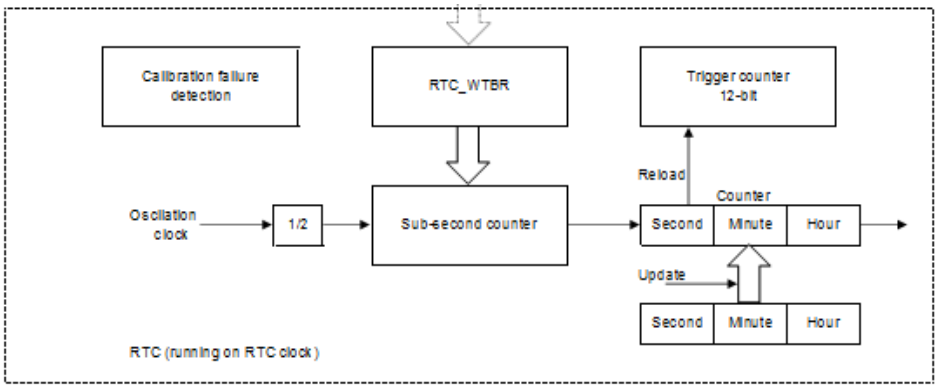
Page	Section	Change Results
567	CHAPTER 6:Low Power Consumption 5. Registers 5.13.7. APP Clock Divider Register 0 (SYSC1_APPCKDIVR0)	Error) **[bit20:16] TRCDIV: TRC Clock Divider Setting Bits** These bits set the division ratio of the TRC clock from the source clock (clock domain TRC). bit20:16 / Description 0000 / No division 0001 / Divided by 2 0010 / Divided by 3 0011 / Divided by 4 . . . / . . . 1101 / Divided by 14 1110 / Divided by 15 1111 / Divided by 16 Correct) **[bit20:16] TRCDIV: TRC clock divider setting bits** These bits set the division ratio of the TRC clock from the source clock (clock domain TRC). Bit20:16 / Description 00000 / No division 00001 / Divided by 2 00010 / Divided by 3 00011 / Divided by 4 ..... / ..... 11101 / Divided by 30 11110 / Divided by 31 11111 / Divided by 32

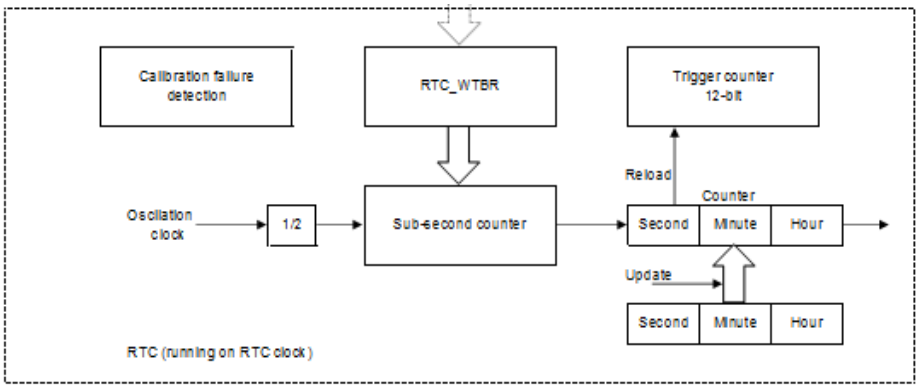
<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>621</td><td>CHAPTER 6:Low Power Consumption
5. Registers
5.14.7. STS Clock Divider Register 0 (SYSC1_STSCKDIVR0)</td><td>Error)

[bit20:16] TRCDIV: TRC Clock Divider Setting Bits
These bits set the division ratio of the TRC clock from the source clock (clock domain TRC).
<table><tr><th>bit20:16</th><th>Description</th></tr><tr><td>0000</td><td>No division</td></tr><tr><td>0001</td><td>Divided by 2</td></tr><tr><td>0010</td><td>Divided by 3</td></tr><tr><td>0011</td><td>Divided by 4</td></tr><tr><td>. . . .</td><td>. . . .</td></tr><tr><td>1101</td><td>Divided by 14</td></tr><tr><td>1110</td><td>Divided by 15</td></tr><tr><td>1111</td><td>Divided by 16</td></tr></table>
Correct)

[bit20:16] TRCDIV: TRC clock divider setting bits
These bits set the division ratio of the TRC clock from the source clock (clock domain TRC).
<table><tr><th>Bit20:16</th><th>Description</th></tr><tr><td>00000</td><td>No division</td></tr><tr><td>00001</td><td>Divided by 2</td></tr><tr><td>00010</td><td>Divided by 3</td></tr><tr><td>00011</td><td>Divided by 4</td></tr><tr><td>.....</td><td>.....</td></tr><tr><td>11101</td><td>Divided by 30</td></tr><tr><td>11110</td><td>Divided by 31</td></tr><tr><td>11111</td><td>Divided by 32</td></tr></table></td></tr>
<tr><td>677</td><td>CHAPTER 8:Clock Supervisor
5. Registers</td><td>Error)
Table 5-2 Clock Supervisor Register Map (B/B-PG)

Correct)
Table 5-2 Clock Supervisor Register Map (B/B)</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>771</td><td>CHAPTER 10:Real Time Clock 2. Configuration and Block Diagram</td><td>Error)


Correct)

</td></tr>
<tr><td>796</td><td>CHAPTER 10:Real Time Clock 4. Registers 4.6. Sub-Second Register (RTC_WTBR)</td><td>Error)
Table 4-3 Example Configuration of RTC_WTBR Register for Different Clock Configurations
<table>
<tr><th></th><th>RTC_WTBR:WTBR[23:0]</th></tr>
<tr><td>Main oscillator, 4MHz</td><td>0x0F423F</td></tr>
<tr><td>RC oscillator, 100kHz</td><td>0x0061A7</td></tr>
<tr><td>Sub oscillator, 32.768kHz</td><td>0x001FFF</td></tr>
</table>
Calculation formula:
RTC_WTBR:WTBR[23:0] = fCLKRTC/2 x 0.5 s – 1

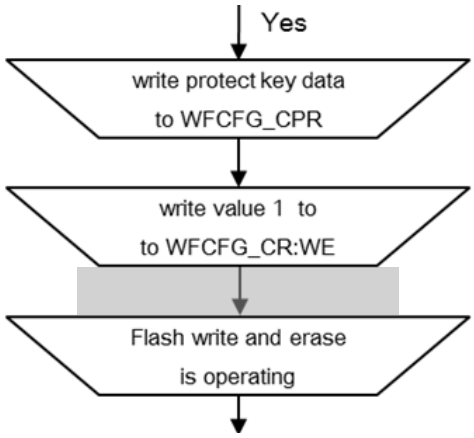
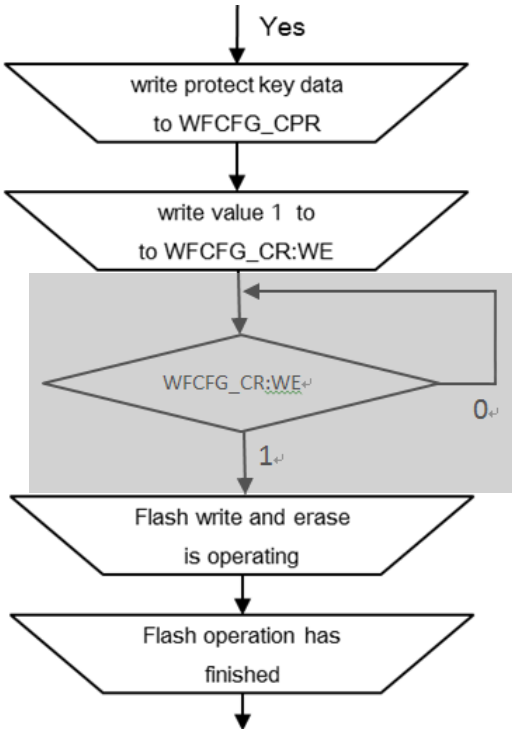
Correct)
Table 4-3 Example Configuration of RTC_WTBR Register for Different Clock Configurations
<table>
<tr><th></th><th>RTC_WTBR:WTBR[23:0]</th></tr>
<tr><td>Main oscillator, 4MHz</td><td>0x0F423F</td></tr>
<tr><td>RC oscillator, 100kHz</td><td>0x0061A7</td></tr>
<tr><td>Sub oscillator, 32.768kHz</td><td>0x001FFF</td></tr>
</table>
Calculation formula:
RTC_WTBR:WTBR[23:0] = fCLKRTC x 0.5 s – 1</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1120</td><td>CHAPTER 18:WorkFLASH
3. Explanation of Operation
3.7. Bus Error Response</td><td>Error)
<table>
<tr><th rowspan="2">Error Factor</th><th colspan="2">WFCFG_BERR</th></tr>
<tr><th>bit</th><th>Field</th></tr>
<tr><td>- Setting the sector number that does not exist, and performing the sector erase.</td><td>11</td><td>ERSIGN</td></tr>
<tr><td>- Writing to a read-only register.</td><td>10</td><td>RORW</td></tr>
<tr><td>- Writing to the mirror area 4
(however, if the reserved area is written, RESA flag is set instead of this bit).</td><td>9</td><td>NWTM</td></tr>
<tr><td>- Command sequencer writes to WorkFLASH or perform a command from WFCFG_SEQCM register (after this action, write operations/commands are ignored) unless the command sequencer is idle.
- Performing the sector erase suspend command or read/reset command while Flash is in the sector erase suspend state.
- Performing write or read operation to the sector to be erased while Flash is in the sector erase suspend state.
- Performing the sector erase suspend command while Flash is in the normal state.</td><td>8</td><td>ACCIGN</td></tr>
</table>
Correct)
<table>
<tr><th rowspan="2">Error Factor</th><th colspan="2">WFCFG_BERR</th></tr>
<tr><th>bit</th><th>Field</th></tr>
<tr><td>- Setting the sector number that does not exist, and performing the sector erase.</td><td>11</td><td>ERSIGN</td></tr>
<tr><td>- Writing to a read-only register.</td><td>10</td><td>RORW</td></tr>
<tr><td>- Writing to the mirror area 4
(however, if the reserved area is written, RESA flag is set instead of this bit).</td><td>9</td><td>NWTM</td></tr>
<tr><td>- Command sequencer writes to WorkFLASH or perform a command from WFCFG_SEQCM register (after this action, write operations/commands are ignored) unless the command sequencer is idle.
- Performing the sector erase suspend command or read/reset command while Flash is in the sector erase suspend state.
- The WorkFLASH configuration protection key register (WFCFG_CPR) is used to protect the following registers from unintended writing:
-WorkFLASH configuration register (WFCFG_CR)
-WorkFLASH ECC control register (WFCFG_ECR)
-WorkFLASH data bit error injection register (WFCFG_DBEIR)
-WorkFLASH ECC bit error injection register (WFCFG_EEIR)
- Performing write or read operation to the sector to be erased while Flash is in the sector erase suspend state.
- Performing the sector erase suspend command while Flash is in the normal state if reading is attempted while the command sequencer is operating, WorkFLASH makes a bus error response.</td><td>8</td><td>ACCIGN</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1120</td><td>CHAPTER 18:WorkFLASH
3. Explanation of Operation
3.7. Bus Error Response</td><td>
Error)
<table>
<tr><td>- Accessing the reserved area of WorkFLASH
(the reserved area in Flash and registers)</td><td>5</td><td>RESA</td></tr>
<tr><td>- Setting the sector number that is protected to ERS bit of WFCFG_SEQCM.

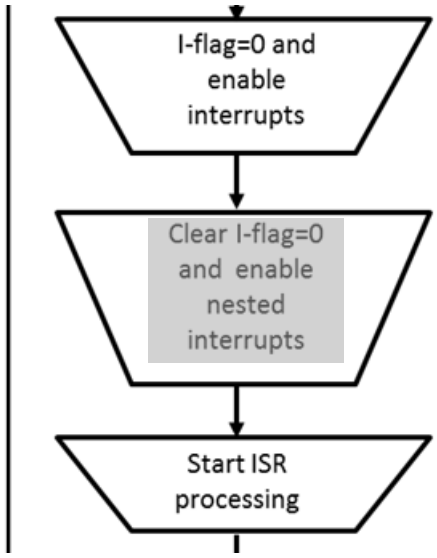
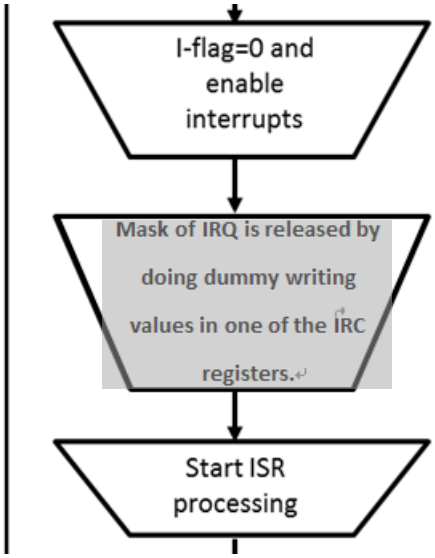
- Setting the number of last two sectors dedicated to SHE to WFCFG_SEQCM.</td><td>4</td><td>RWE</td></tr>
<tr><td>- Accessing Flash or registers with the prohibited data width.</td><td>2</td><td>SIZE</td></tr>
</table>
Correct)
<table>
<tr><td>- Accessing the reserved area of WorkFLASH
(the reserved area in Flash and registers)
<table>
<tr><th>Mirror Area 4 Address</th><th colspan="2"></th><th>RESA</th><th>NWTM</th><th>BUSERR</th></tr>
<tr><td>0E3F_FFFF
0E34_0000</td><td rowspan="4">Reserved</td><td>Reserved-3 (1MB-256KB)</td><td>not set</td><td>not set</td><td>error</td></tr>
<tr><td>0E33_FFFF
0E32_0000</td><td>Reserved-2 (128KB)</td><td>set</td><td>not set</td><td>error</td></tr>
<tr><td>0E31_FFFF
0E31_C000</td><td>Reserved-1 (16KB)</td><td>not set</td><td>set</td><td>error</td></tr>
<tr><td>0E31_BFFF
0E30_0000</td><td>SA0-13</td><td>user area (112KB)</td><td>not set</td><td>set</td><td>error</td></tr>
</table></td><td>5</td><td>RESA</td></tr>
<tr><td>- Setting the sector number that is protected to ERS bit of WFCFG_SEQCM.

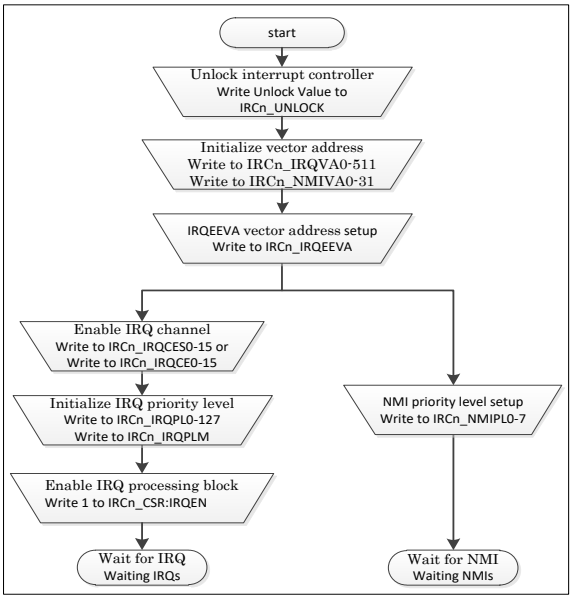
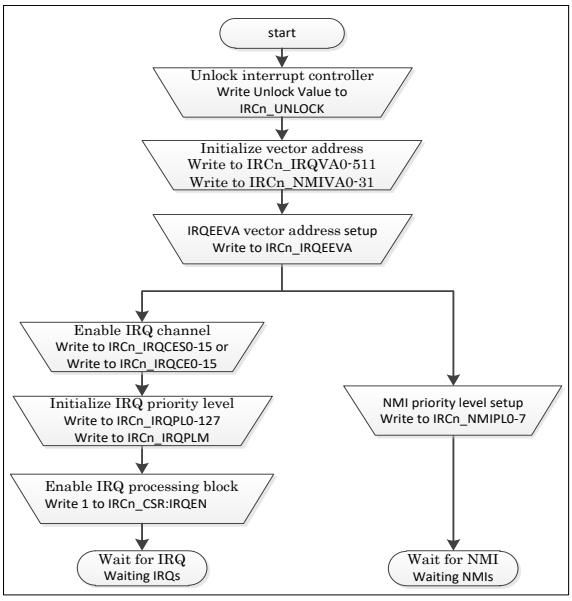
- Setting the number of last two sectors dedicated to SHE to WFCFG_SEQCM.</td><td>4</td><td>RWE</td></tr>
<tr><td>- Accessing Flash or registers with the prohibited data width.</td><td>2</td><td>SIZE</td></tr>
</table>
</td></tr>
</table>

Page	Section	Change Results
1123	CHAPTER 18:WorkFLASH 4. Setting Procedure 4.2. Enabling Writing	Error) Figure 4-1 Procedure for Setting Flash Write Enable  Correct) Figure 4-1 Procedure for Setting Flash Write Enable 

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1152</td><td>CHAPTER 18:WorkFLASH
5. Registers
5.16. WorkFLASH Bus Error Response Status Register: WFCFG_BERR</td><td>Error)
[bit4] RWE: Reserved area access
<table><tr><th>RWE</th><th>Description</th></tr><tr><td>0</td><td>- Access to a protected sector has not been detected.
- No ERS writing violation has been detected.</td></tr><tr><td>1</td><td>- This indicates that a bus error response was made because access to a protected sector was detected.
- A sector assigned to SHE has been written to the ERS[7.0] bits in WFCFG_SEQCM.
- The sector protection function specifies the protected sectors.</td></tr></table>
Correct)
[bit4] RWE: Reserved area access
<table><tr><th>RWE</th><th>Description</th></tr><tr><td>0</td><td>- Access to a protected sector has not been detected.
- No ERS writing violation has been detected.</td></tr><tr><td>1</td><td>- This indicates that a bus error response was made because access to a protected sector was detected.
- A sector assigned to SHE has been written to the ERS[7.0] bits in WFCFG_SEQCM.</td></tr></table></td></tr>
</table>

Page	Section	Change Results
1225	CHAPTER 20: BootROM Software Interface 4.Notes	Error) This section provides notes regarding the BootROM. Interruption setting The BootROM software does not set the I bit and F bit of the core internal register CPSR. (The default value is 1, which means "Disable.") Before using IRQ or FIQ, a user application shall set the I and F bits to 0. Correct) This section provides notes regarding the BootROM. Interruption setting The BootROM software does not set the I bit and F bit of the core internal register CPSR. (The default value is 1, which means "Disable.") Before using IRQ or FIQ, a user application shall set the I and F bits to 0. ARM Core Setting 1. R0-R12 is Clear as 0. 2.bit24(VE) of System Control Register (SCTLR) is set as 1. 3.2'b11 (Privileged and User mode access) is set as bit23:22 of Coprocessor Access Control Register (CPACR) and CP11,CP10 of bit21:20. 4.bit30 of Floating-Point Exception Register (FPEXC) (EN) is set as 1.

Page	Section	Change Results
1257	CHAPTER 22: Interrupt Controller 3.Explanation of Operation	Error) Figure 3-4 Nest Supported IRQ Processing Flow  Correct) Figure 3-4 Nest Supported IRQ Processing Flow 

Page	Section	Change Results
1264	CHAPTER 22:Interrupt Controller 4. Setting Procedure Examples	Error) **Figure 4-1 Interrupt Controller Initial Setting Flow**  Correct) **Figure 4-1 Interrupt Controller Initial Setting Flow**  Transition from RUN to PSS S1. I-FLAG is set to interrupt enable. S2. Read from IRCn_IRQST:nIRQ. →Surely to suppress the nIRQ generation from the IRC. S3. Start the interrupt that is trigger for returning from PSS. S4. Write to IRCn_IRQPLM. →nIRQ function is reawakened. S5. Run the profile that migrate to PSS.. S6. Confirm the interrupt cause and clear. S7. Run WFI.

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1508</td><td>CHAPTER 26:Secure Hardware Extension(SHE)
3. Operation of the Secure Hardware Extension
3.2. Operation of the Data Interface</td><td>Error)
System integration issues

Memory Protection
For safety reasons both masters have a Memory Protection Unit (MPU) in order to protect the memory areas of the MCU. For details on MPU configuration, refer to Chapter X (Memory Protection Unit for the AXI) of the Traveo Cluster Series Hardware Manual.

Correct)
System integration issues

Memory Protection
For safety reasons both masters have a Memory Protection Unit (MPU) in order to protect the memory areas of the MCU. For details on MPU configuration, refer to Chapter 25 (Memory Protection Unit for the AXI of SHE) of the Traveo Cluster Series Hardware Manual.</td></tr>
<tr><td>1885</td><td>CHAPTER 31:CAN FD Controller(MCAN3.2)
5. Registers
5.6. CC Control Register (MCG_CANFDx_CCCR, CPG_CANFDx_CCCR)</td><td>Error)
<table>
<tr><td>bit</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td></tr>
<tr><td>Field</td><td>NISO</td><td>TXP</td><td>EFBI</td><td>PXHD</td><td colspan="2">Reserved</td><td>BRSE</td></tr>
<tr><td>R/W Attribute</td><td>R0,W0</td><td>R/W</td><td>R,WX</td><td>R,WX</td><td colspan="2">R/W</td><td>R/W</td></tr>
<tr><td>Protection Attribute</td><td>-</td><td>-</td><td>-</td><td>-</td><td colspan="2">-</td><td>-</td></tr>
<tr><td>Initial value</td><td>0</td><td>0</td><td>0</td><td>0</td><td colspan="2">00</td><td>0</td></tr>
</table>
Correct)
<table>
<tr><td>bit</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td></tr>
<tr><td>Field</td><td>NISO</td><td>TXP</td><td>EFBI</td><td>PXHD</td><td colspan="2">Reserved</td><td>BRSE</td></tr>
<tr><td>R/W Attribute</td><td>R0,W0</td><td>R/W</td><td>R,WX</td><td>R,WX</td><td colspan="2">R0,W0</td><td>R/W</td></tr>
<tr><td>Protection Attribute</td><td>-</td><td>-</td><td>-</td><td>-</td><td colspan="2">-</td><td>-</td></tr>
<tr><td>Initial value</td><td>0</td><td>0</td><td>0</td><td>0</td><td colspan="2">00</td><td>0</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1889</td><td>CHAPTER 31:CAN FD Controller(MCAN3.2) 5. Registers 5.7. Nominal Bit Timing & Prescaler Register (MCG_CANFDx_NBTP, CPG_CANFDx_NBTP)</td><td>Error)
<table>
<tr><td>Bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Field</td><td>Reserved</td><td colspan="7">NTSEG2[6:0]</td></tr>
<tr><td>R/W Attribute</td><td>R0,WX</td><td colspan="7">R/W</td></tr>
<tr><td>Protection Attribute</td><td>-</td><td colspan="7">-</td></tr>
<tr><td>Initial value</td><td>0</td><td colspan="7">0x03</td></tr>
</table>
Correct)
<table>
<tr><td>Bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Field</td><td>Reserved</td><td colspan="7">NTSEG2[6:0]</td></tr>
<tr><td>R/W Attribute</td><td>R0、W0</td><td colspan="7">R/W</td></tr>
<tr><td>Protection Attribute</td><td>-</td><td colspan="7">-</td></tr>
<tr><td>Initial value</td><td>0</td><td colspan="7">0x03</td></tr>
</table></td></tr>
<tr><td>2048</td><td>CHAPTER 35:Overview of Multi-function Serial Interface 6. Registers of the Multi-Function Serial Interface</td><td>Error)
<table>
<tr><td rowspan="3">SSR /ESCR</td><td>0/1</td><td>REC</td><td>-</td><td>PE</td><td>FRE</td></tr>
<tr><td>2</td><td>REC</td><td>-</td><td>ES</td><td>AWC</td></tr>
<tr><td>3</td><td>REC</td><td>-</td><td>LBD</td><td>FRE</td></tr>
</table>
Correct)
<table>
<tr><td rowspan="3">SSR /ESCR</td><td>0/1</td><td>REC</td><td>-</td><td>PE</td><td>FRE</td></tr>
<tr><td>2</td><td>REC</td><td>-</td><td>Reserved</td><td>AWC</td></tr>
<tr><td>3</td><td>REC</td><td>-</td><td>LBD</td><td>FRE</td></tr>
</table></td></tr>
<tr><td>2048</td><td>CHAPTER 35:Overview of Multi-function Serial Interface 6. Registers of the Multi-Function Serial Interface</td><td>Error)
<table>
<tr><td>SCSCR 1/0</td><td>2</td><td>SST1</td><td>SST0</td><td>SED1</td><td>SED0</td><td>SCD1</td><td>SCD0</td><td>-</td><td>CDIV2</td></tr>
</table>
Correct)
<table>
<tr><td>SCSCR 1/0</td><td>2</td><td>SST1</td><td>SST0</td><td>SED1</td><td>SED0</td><td>SCD1</td><td>SCD0</td><td>SCAM</td><td>CDIV2</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>2403</td><td>CHAPTER 38:I2C Interface (I2C Communication Control Interface) 8. I2C Interface Registers 8.3. I2C Bus Status Register (IBSR)</td><td>Error)
FBT bit clear conditions:

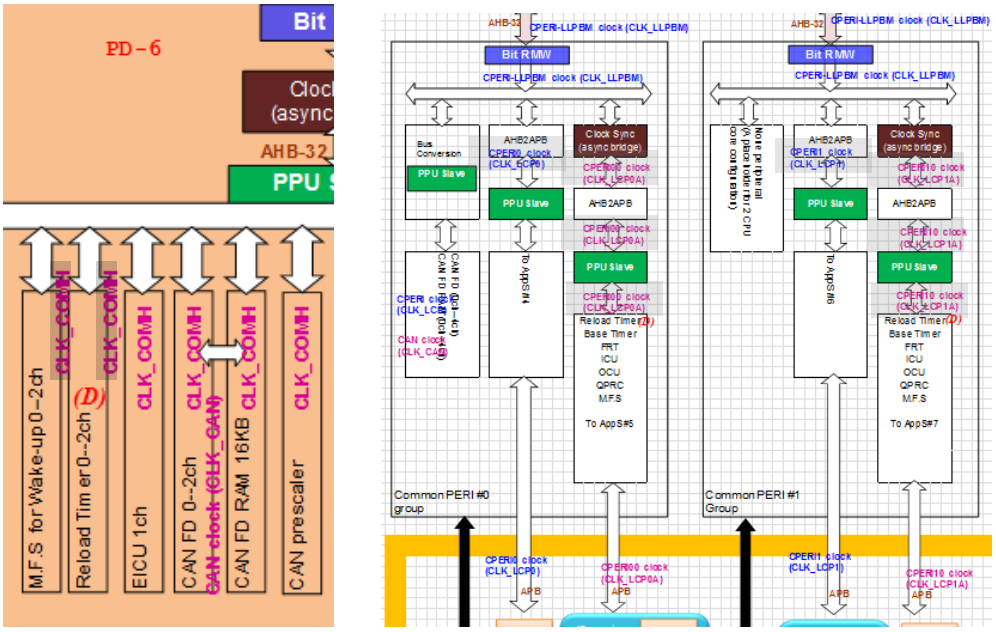
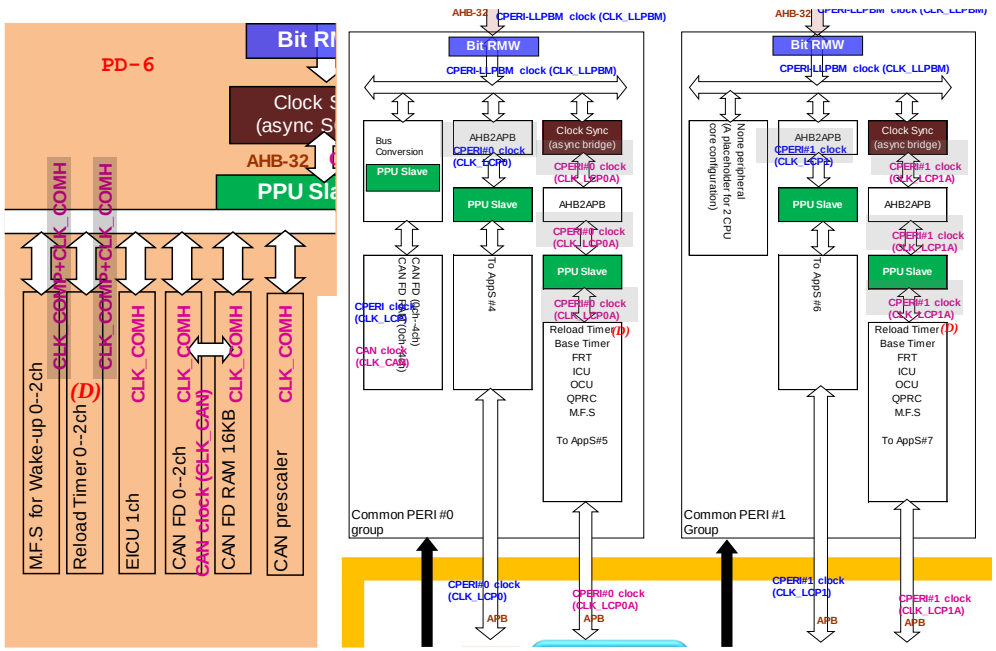
1. The 2nd byte is transmitted and received.

Correct)
FBT bit clear conditions:

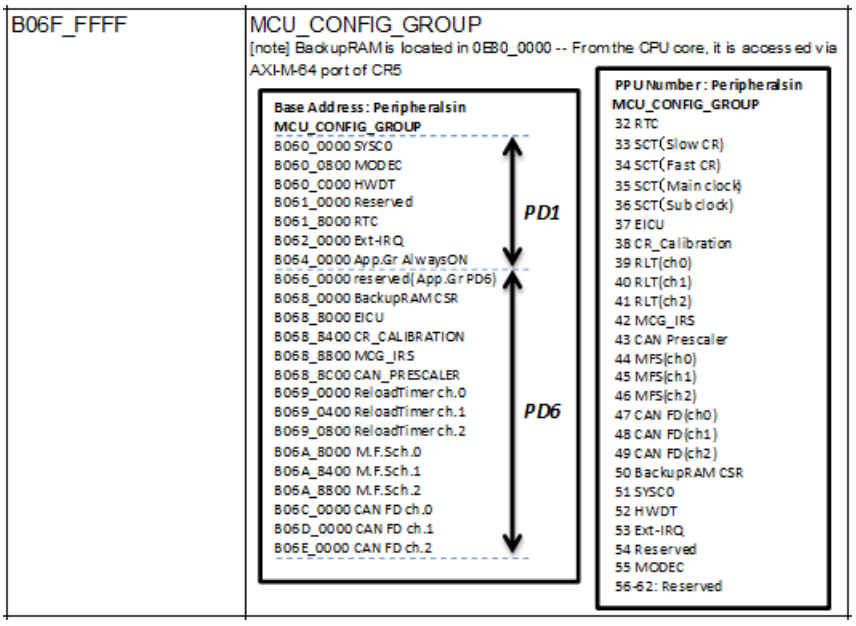
1. The 2nd byte starts to be transmitted and received.</td></tr>
<tr><td>2996</td><td>CHAPTER 50:I/O Port 4. Register List 4.10. Port Setting Register (PPC_PCFGRijj) (i = 0 to (product specification), jj = 00 to 31)</td><td>Error)
[bit14] POD: Port Output Data Bit
This bit indicates the value outputted to a pin.
This bit is enabled only when the port output enable bit (POE) is "1".
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Output "L".</td></tr>
<tr><td>1</td><td>Output "H".</td></tr>
</table>

Correct)
[bit14] POD: Port Output Data Bit
This bit indicates the value outputted to a pin.
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Output "L".</td></tr>
<tr><td>1</td><td>Output "H".</td></tr>
</table>
</td></tr>
</table>

Section	Change Results
002-04854 Rev. *A	
CHAPTER1: Platform Overview 1.OVERVIEW 1.1.Function Overview	Features of "For details" should be corrected as indicated by the shading below (Error) SECDED ECC for TCM ports (both for Flash and for RAM) (Correct) SECDED ECC for TCM ports (for RAM)
CHAPTER1: Platform Overview 1.OVERVIEW 1.1.Function Overview	Features of "For details" should be corrected as indicated by the shading below (Error) SECDED ECC by 8-byte (Correct) SECDED ECC by 8-byte via AXI
CHAPTER1: Platform Overview 2. Product Configuration	Features of "For details" should be corrected as indicated by the shading below (Error) 2MB with SECDED ECC (configurable by option) (Correct) 2MB with SECDED ECC via AXI(configurable by option)

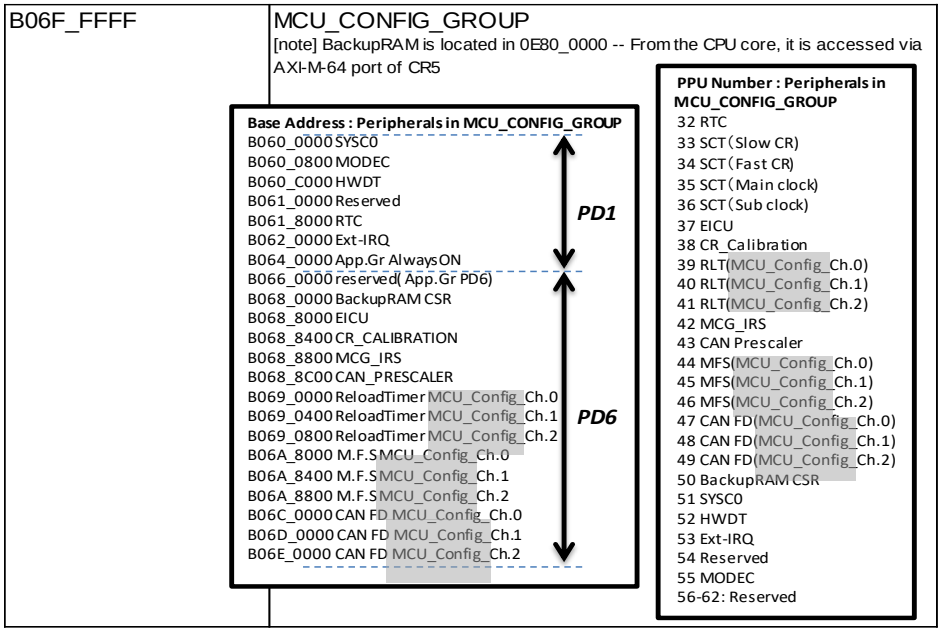
Section	Change Results
CHAPTER1: Platform Overview 3.Configuration	Revised " Figure 3-1 Block Diagram " as follows. The function name should be corrected as indicated by shading below. (Error)  (Correct) 

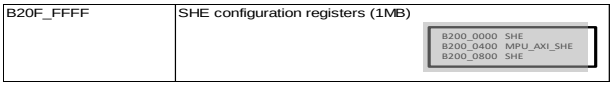
Section	Change Results
CHAPTER1: Platform Overview 6. I/O map	Revised " Figure I/O map (1) " as follows. Features of “For details” should be corrected as indicated by the shading below. (Error)



B20F_FFFF	SHE configuration registers (1MB)

(Correct)





<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER4:
Reset
4.Registers</td><td>Features of "For details" should be corrected as indicated by the shading below

(Error)
<table><tr><td>0x0000_0034</td><td>SYSC_PDRSTATUS
00000000_00000000_00000000_00000000</td></tr></table>
(Correct)
<table><tr><td>0x0000_0034</td><td>SYSC_PDRSTSTATUS
00000000_00000000_00000000_00000000</td></tr></table></td></tr>
<tr><td>CHAPTER4:
Reset
4.Registers
4.1.Reset
Control Register
(SYSC_RSTCN
TR)</td><td>Features of "For details" should be corrected as indicated by the shading below

(Error)
Bit31:24 DBGR: Software debugger reset register bit
<table><tr><th>Bit</th><th>Explanation</th></tr></table>
(Correct)
[bit31:24] DBGR: Software debugger Reset Register Bit
<table><tr><th>DBGR</th><th>Explanation</th></tr></table></td></tr>
<tr><td>CHAPTER4:
Reset
4.Registers
4.1.Reset
Control Register
(SYSC_RSTCN
TR)</td><td>Features of "For details" should be corrected as indicated by the shading below

(Error)
Bit23:16 SWHRST: Software trigger hard reset register bit
<table><tr><th>Bit</th><th>Explanation</th></tr></table>
(Correct)
[bit23:16] SWHRST: Software trigger hard Reset Register Bit
<table><tr><th>SWHRST</th><th>Explanation</th></tr></table></td></tr>
<tr><td>CHAPTER4:
Reset
4.Registers
4.1.Reset
Control Register
(SYSC_RSTCN
TR)</td><td>Features of "For details" should be corrected as indicated by the shading below

(Error)
Bit7:0 SWRST: Software reset register bit
<table><tr><th>Bit</th><th>Explanation</th></tr></table>
(Correct)
[bit7:0] SWRST: Software Reset Register Bit
<table><tr><th>SWRST</th><th>Explanation</th></tr></table></td></tr>
</table>

Section	Change Results
CHAPTER4: Reset 4.Registers 4.2User Reset Factor Register (SYSC_RSTCAUSEUR)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit31 LVDL2R: Extended internal power supply low-voltage detection reset detection bit (Correct) [bit31] LVDL2R: Extended Internal Power Supply Low-Voltage Detection Reset Detection Bit
CHAPTER4: Reset 4.Registers 4.2User Reset Factor Register (SYSC_RSTCAUSEUR)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit30 LVDL1R: Internal power supply low-voltage detection reset detection bit (Correct) [bit30] LVDL1R: Internal Power Supply Low-Voltage Detection Reset Detection Bit
CHAPTER4: Reset 4.Registers 4.2User Reset Factor Register (SYSC_RSTCAUSEUR)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit29 CSVSCRR: Slow CR clock supervisor reset detection bit (Correct) [bit29] CSVSCRR: Slow CR Clock Supervisor Reset Detection Bit
CHAPTER4: Reset 4.Registers 4.2User Reset Factor Register (SYSC_RSTCAUSEUR)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit28 CSVFCRR: Fast CR clock supervisor reset detection bit (Correct) [bit28] CSVFCRR: Fast CR Clock Supervisor Reset Detection Bit
CHAPTER4: Reset 4.Registers 4.2User Reset Factor Register (SYSC_RSTCAUSEUR)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit27 CSVSR0: SSCG0 clock supervisor reset detection bit (Correct) [bit27] CSVSR0: SSCG0 Clock Supervisor Reset Detection Bit
CHAPTER4: Reset 4.Registers 4.2User Reset Factor Register (SYSC_RSTCAUSEUR)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit26 CSVPR0: PLL0 clock supervisor reset detection bit (Correct) [bit26] CSVPR0: PLL0 Clock Supervisor Reset Detection Bit

Section	Change Results
CHAPTER4: Reset 4.Registers 4.2User Reset Factor Register (SYSC_RSTCAUSEUR)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit25 CSVSOR: Sub clock supervisor reset detection bit (Correct) [bit25] CSVSOR: Sub Clock Supervisor Reset Detection Bit
CHAPTER4: Reset 4.Registers 4.2User Reset Factor Register (SYSC_RSTCAUSEUR)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit24 CSVMOR: Main clock supervisor reset detection bit (Correct) [bit24] CSVMOR: Main Clock Supervisor Reset Detection Bit
CHAPTER4: Reset 4.Registers 4.2User Reset Factor Register (SYSC_RSTCAUSEUR)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit23:21 Reserved (Correct) [bit23:21] Reserved
CHAPTER4: Reset 4.Registers 4.2User Reset Factor Register (SYSC_RSTCAUSEUR)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit20 SHRST: Software trigger hard reset detection bit (Correct) [bit20 SHRST: Software Trigger hard Reset Detection Bit
CHAPTER4: Reset 4.Registers 4.2User Reset Factor Register (SYSC_RSTCAUSEUR)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit19:17 Reserved (Correct) [bit19:17] Reserved
CHAPTER4: Reset 4.Registers 4.2User Reset Factor Register (SYSC_RSTCAUSEUR)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit16 SRST: Software reset detection bit t (Correct) [bit16] SRST: Software Reset Detection Bit

Section	Change Results
CHAPTER4: Reset 4.Registers 4.2User Reset Factor Register (SYSC_RSTCAUSEUR)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit15:13 Reserved (Correct) [bit15:13] Reserved
CHAPTER4: Reset 4.Registers 4.2User Reset Factor Register (SYSC_RSTCAUSEUR)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit12 SWDR: Software watchdog reset detection bit (Correct) [bit12] SWDR: Software Watchdog Reset Detection Bit
CHAPTER4: Reset 4.Registers 4.2User Reset Factor Register (SYSC_RSTCAUSEUR)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit11 HWDR: Hardware watchdog reset detection bit (Correct) [bit11] HWDR: Hardware Watchdog Reset Detection Bit
CHAPTER4: Reset 4.Registers 4.2User Reset Factor Register (SYSC_RSTCAUSEUR)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit10 PRFERR: Profile error reset detection bit (Correct) [bit10] PRFERR: Profile Error Reset Detection Bit
CHAPTER4: Reset 4.Registers 4.2User Reset Factor Register (SYSC_RSTCAUSEUR)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit9 SRSTX: nSRST pin input reset detection bit (Correct) [bit9] SRSTX: nSRST Pin Input Reset Detection Bit
CHAPTER4: Reset 4.Registers 4.2User Reset Factor Register (SYSC_RSTCAUSEUR)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit8 IMR: Illegal mode detection reset detection bit (Correct) [bit8] IMR: Illegal Mode Detection Reset Detection Bit

Section	Change Results
CHAPTER4: Reset 4.Registers 4.2User Reset Factor Register (SYSC_RSTCAUSEUR)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit7 LVDH2R: Extended external power supply low-voltage detection reset detection bit (Correct) [bit7] LVDH2R: Extended External Power Supply Low-Voltage Detection Reset Detection Bit
CHAPTER4: Reset 4.Registers 4.2User Reset Factor Register (SYSC_RSTCAUSEUR)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit6 LVDH1R: External power supply low-voltage detection reset detection bit (Correct) [bit6] LVDH1R: External Power Supply Low-Voltage Detection Reset Detection Bit
CHAPTER4: Reset 4.Registers 4.2User Reset Factor Register (SYSC_RSTCAUSEUR)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit5 Reserved (Correct) [bit5] Reserved
CHAPTER4: Reset 4.Registers 4.2User Reset Factor Register (SYSC_RSTCAUSEUR)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit4 RSTX: RSTX pin input reset detection bit (Correct) [bit4] RSTX: RSTX Pin Input Reset Detection Bit
CHAPTER4: Reset 4.Registers 4.2User Reset Factor Register (SYSC_RSTCAUSEUR)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit3 CKTOR: Clock stop wait timeout reset detection bit (Correct) [bit3] CKTOR: Clock Stop Wait Timeout Reset Detection Bit
CHAPTER4: Reset 4.Registers 4.2User Reset Factor Register (SYSC_RSTCAUSEUR)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit2 INITX: INITX detection bit (Correct) [bit2] INITX: INITX Detection Bit

Section	Change Results
CHAPTER4: Reset 4.Registers 4.2User Reset Factor Register (SYSC_RSTCAUSEUR)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit1 RVD: RAM retention low-voltage detection reset detection bit (Correct) [bit1] RVD: RAM Retention Low-Voltage Detection Reset Detection Bit
CHAPTER4: Reset 4.Registers 4.2User Reset Factor Register (SYSC_RSTCAUSEUR)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit0 PONR: Power-on reset detection bit (Correct) [bit0] PONR: Power-on Reset Detection Bit
CHAPTER4: Reset 4.Registers 4.3. User Extended CSV Reset Factor Register (SYSC_EXCSVRSTCAUSEUR)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit31:8 Reserved (Correct) [bit31:8] Reserved
CHAPTER4: Reset 4.Registers 4.3. User Extended CSV Reset Factor Register (SYSC_EXCSVRSTCAUSEUR)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit7 CSVSR 3: SSCG3 clock supervisor reset detection bit (Correct) [bit7] CSVSR 3: SSCG3 Clock Supervisor Reset Detection Bit
CHAPTER4: Reset 4.Registers 4.3. User Extended CSV Reset Factor Register (SYSC_EXCSVRSTCAUSEUR)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit6 CSVSR 2: SSCG2 clock supervisor reset detection bit (Correct) [bit6] CSVSR 2: SSCG2 Clock Supervisor Reset Detection Bit

Section	Change Results
CHAPTER4: Reset 4.Registers 4.3. User Extended CSV Reset Factor Register (SYSC_EXCSV RSTCAUSEUR)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit5 CSVSR 1: SSCG1 clock supervisor reset detection bit (Correct) [bit5] CSVSR 1: SSCG1 Clock Supervisor Reset Detection Bit
CHAPTER4: Reset 4.Registers 4.3. User Extended CSV Reset Factor Register (SYSC_EXCSV RSTCAUSEUR)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit4 Reserved (Correct) [bit4] Reserved
CHAPTER4: Reset 4.Registers 4.3. User Extended CSV Reset Factor Register (SYSC_EXCSV RSTCAUSEUR)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit3 CSVPR3: PLL3 clock supervisor reset detection bit (Correct) [bit3] CSVPR3: PLL3 Clock Supervisor Reset Detection Bit
CHAPTER4: Reset 4.Registers 4.3. User Extended CSV Reset Factor Register (SYSC_EXCSV RSTCAUSEUR)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit2 CSVPR2: PLL2 clock supervisor reset detection bit (Correct) [bit2] CSVPR2: PLL2 Clock Supervisor Reset Detection Bit
CHAPTER4: Reset 4.Registers 4.3. User Extended CSV Reset Factor Register (SYSC_EXCSV RSTCAUSEUR)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit1 CSVPR1: PLL1 clock supervisor reset detection bit (Correct) [bit1] CSVPR1: PLL1 Clock Supervisor Reset Detection Bit

Section	Change Results
CHAPTER4: Reset 4.Registers 4.3. User Extended CSV Reset Factor Register (SYSC_EXCSVRSTCAUSEUR)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit0 Reserved (Correct) [bit0] Reserved
CHAPTER4: Reset 4.Registers 4.4.User PowerDomain Reset Factor Register (SYSC_PDRSTCAUSEUR)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit31:18 Reserved (Correct) [bit31:18] Reserved
CHAPTER4: Reset 4.Registers 4.4.User PowerDomain Reset Factor Register (SYSC_PDRSTCAUSEUR)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit17 PR6R1: PowerDomain6_1 reset detection bit (Correct) [bit17] PR6R1: PowerDomain6_1 Reset Detection Bit
CHAPTER4: Reset 4.Registers 4.4.User PowerDomain Reset Factor Register (SYSC_PDRSTCAUSEUR)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit16 PR6R0: PowerDomain6_0 reset detection bit (Correct) [bit16] PR6R0: PowerDomain6_0 Reset Detection Bit
CHAPTER4: Reset 4.Registers 4.4.User PowerDomain Reset Factor Register (SYSC_PDRSTCAUSEUR)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit15 PR5R3: PowerDomain5_3 reset detection bit (Correct) [bit15] PR5R3: PowerDomain5_3 Reset Detection Bit

Section	Change Results
CHAPTER4: Reset 4.Registers 4.4.User PowerDomain Reset Factor Register (SYSC_PDRSTCAUSEUR)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit14 PR5R2: PowerDomain5_2 reset detection bit (Correct) [bit14] PR5R2: PowerDomain5_2 Reset Detection Bit
CHAPTER4: Reset 4.Registers 4.4.User PowerDomain Reset Factor Register (SYSC_PDRSTCAUSEUR)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit13 PR5R1: PowerDomain5_1 reset detection bit (Correct) [bit13] PR5R1: PowerDomain5_1 Reset Detection Bit
CHAPTER4: Reset 4.Registers 4.4.User PowerDomain Reset Factor Register (SYSC_PDRSTCAUSEUR)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit12 PR5R1: PowerDomain5_0 reset detection bit (Correct) [bit12] PR5R1: PowerDomain5_0 Reset Detection Bit
CHAPTER4: Reset 4.Registers 4.4.User PowerDomain Reset Factor Register (SYSC_PDRSTCAUSEUR)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit11:10 Reserved (Correct) [bit11:10] Reserved
CHAPTER4: Reset 4.Registers 4.4.User PowerDomain Reset Factor Register (SYSC_PDRSTCAUSEUR)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit9 PR4R1: PowerDomain4_1 reset detection bit (Correct) [bit9] PR4R1: PowerDomain4_1 Reset Detection Bit

Section	Change Results
CHAPTER4: Reset 4.Registers 4.4.User PowerDomain Reset Factor Register (SYSC_PDRST CAUSEUR)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit8 PR4R0: PowerDomain4_0 reset detection bit (Correct) [bit8] PR4R0: PowerDomain4_0 Reset Detection Bit
CHAPTER4: Reset 4.Registers 4.4.User PowerDomain Reset Factor Register (SYSC_PDRST CAUSEUR)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit7:5 Reserved (Correct) [bit7:5] Reserved
CHAPTER4: Reset 4.Registers 4.4.User PowerDomain Reset Factor Register (SYSC_PDRST CAUSEUR)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit4 PR3R0: PowerDomain3_0 reset detection bit (Correct) [bit4] PR3R0: PowerDomain3_0 Reset Detection Bit
CHAPTER4: Reset 4.Registers 4.4.User PowerDomain Reset Factor Register (SYSC_PDRST CAUSEUR)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit3:1 Reserved (Correct) [bit3:1] Reserved
CHAPTER4: Reset 4.Registers 4.4.User PowerDomain Reset Factor Register (SYSC_PDRST CAUSEUR)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit0 PD2R0: PowerDomain2 reset detection bit (Correct) [bit0] PD2R0: PowerDomain2 Reset Detection Bit

Section	Change Results
CHAPTER4: Reset 4.Registers 4.5.BootROM Reset Factor Register (SYSC_RSTCAUSEBT)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit31 LVDL2R: Extended internal power supply low-voltage detection reset detection bit (Correct) [bit31] LVDL2R: Extended Internal Power Supply Low-Voltage Detection Reset Detection Bit
CHAPTER4: Reset 4.Registers 4.5.BootROM Reset Factor Register (SYSC_RSTCAUSEBT)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit30 LVDL1R: Internal power supply low-voltage detection reset detection bit (Correct) [bit30] LVDL1R: Internal Power Supply Low-Voltage Detection Reset Detection Bit
CHAPTER4: Reset 4.Registers 4.5.BootROM Reset Factor Register (SYSC_RSTCAUSEBT)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit29 CSVSCRR: Slow CR clock supervisor reset detection bit (Correct) [bit29] CSVSCRR: Slow CR Clock Supervisor Reset Detection Bit
CHAPTER4: Reset 4.Registers 4.5.BootROM Reset Factor Register (SYSC_RSTCAUSEBT)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit28 CSVFCRR: Fast CR clock supervisor reset detection bit (Correct) [bit28] CSVFCRR: Fast CR Clock Supervisor Reset Detection Bit
CHAPTER4: Reset 4.Registers 4.5.BootROM Reset Factor Register (SYSC_RSTCAUSEBT)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit27 CSVSR0: SSCG0 clock supervisor reset detection bit (Correct) [bit27] CSVSR0: SSCG0 Clock Supervisor Reset Detection Bit

Section	Change Results
CHAPTER4: Reset 4.Registers 4.5.BootROM Reset Factor Register (SYSC_RSTCAUSEBT)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit26 CSVPR0: PLL0 clock supervisor reset detection bit (Correct) [bit26] CSVPR0: PLL0 Clock Supervisor Reset Detection Bit
CHAPTER4: Reset 4.Registers 4.5.BootROM Reset Factor Register (SYSC_RSTCAUSEBT)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit25 CSVSOR: Sub clock supervisor reset detection bit (Correct) [bit25] CSVSOR: Sub Clock Supervisor Reset Detection Bit
CHAPTER4: Reset 4.Registers 4.5.BootROM Reset Factor Register (SYSC_RSTCAUSEBT)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit24 CSVMOR: Main clock supervisor reset detection bit (Correct) [bit24] CSVMOR: Main Clock Supervisor Reset Detection Bit
CHAPTER4: Reset 4.Registers 4.5.BootROM Reset Factor Register (SYSC_RSTCAUSEBT)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit23:21 Reserved (Correct) [bit23:21] Reserved
CHAPTER4: Reset 4.Registers 4.5.BootROM Reset Factor Register (SYSC_RSTCAUSEBT)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit20 SHRST: Software trigger hard reset detection bit (Correct) [bit20] SHRST: Software Trigger hard Reset Detection Bit

Section	Change Results
CHAPTER4: Reset 4.Registers 4.5.BootROM Reset Factor Register (SYSC_RSTCAUSEBT)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit19:17 Reserved (Correct) [bit19:17] Reserve
CHAPTER4: Reset 4.Registers 4.5.BootROM Reset Factor Register (SYSC_RSTCAUSEBT)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit16 SRST: Software reset detection bit (Correct) [bit16] SRST: Software Reset Detection Bt
CHAPTER4: Reset 4.Registers 4.5.BootROM Reset Factor Register (SYSC_RSTCAUSEBT)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit15:13 Reserved (Correct) [bit15:13] Reserved
CHAPTER4: Reset 4.Registers 4.5.BootROM Reset Factor Register (SYSC_RSTCAUSEBT)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit12 SWDR: Software watchdog reset detection bit (Correct) [bit12] SWDR: Software Watchdog Reset Detection Bit
CHAPTER4: Reset 4.Registers 4.5.BootROM Reset Factor Register (SYSC_RSTCAUSEBT)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit11 HWDR: Hardware watchdog reset detection bit (Correct) [bit11] HWDR: Hardware Watchdog Reset Detection Bit

Section	Change Results
CHAPTER4: Reset 4.Registers 4.5.BootROM Reset Factor Register (SYSC_RSTCAUSEBT)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit10 PRFERR: Profile error reset detection bit (Correct) [bit10] PRFERR: Profile Error Reset Detection Bit
CHAPTER4: Reset 4.Registers 4.5.BootROM Reset Factor Register (SYSC_RSTCAUSEBT)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit9 SRSTX: nSRST pin input reset detection bit (Correct) [bit9] SRSTX: nSRST Pin Input Reset Detection Bit
CHAPTER4: Reset 4.Registers 4.5.BootROM Reset Factor Register (SYSC_RSTCAUSEBT)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit8 IMR: Illegal mode detection reset detection bit (Correct) [bit8] IMR: Illegal Mode Detection Reset Detection Bit
CHAPTER4: Reset 4.Registers 4.5.BootROM Reset Factor Register (SYSC_RSTCAUSEBT)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit7 LVDH2R: Extended external power supply low-voltage detection reset detection bit (Correct) [bit7] LVDH2R: Extended External Power Supply Low-Voltage Detection Reset Detection Bit
CHAPTER4: Reset 4.Registers 4.5.BootROM Reset Factor Register (SYSC_RSTCAUSEBT)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit6 LVDH1R: External power supply low-voltage detection reset detection bit (Correct) [bit6] LVDH1R: External Power Supply Low-Voltage Detection Reset Detection Bit

Section	Change Results
CHAPTER4: Reset 4.Registers 4.5.BootROM Reset Factor Register (SYSC_RSTCAUSEBT)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit5 Reserved (Correct) [bit5] Reserved
CHAPTER4: Reset 4.Registers 4.5.BootROM Reset Factor Register (SYSC_RSTCAUSEBT)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit4 RSTX: RSTX pin input reset detection bit (Correct) [bit4] RSTX: RSTX Pin Input Reset Detection Bit
CHAPTER4: Reset 4.Registers 4.5.BootROM Reset Factor Register (SYSC_RSTCAUSEBT)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit3 CKTOR: Clock stop wait timeout reset detection bit (Correct) [bit3] CKTOR: Clock Stop Wait Timeout Reset Detection Bit
CHAPTER4: Reset 4.Registers 4.5.BootROM Reset Factor Register (SYSC_RSTCAUSEBT)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit2 INITX: INITX detection bit (Correct) [bit2] INITX: INITX Detection Bit
CHAPTER4: Reset 4.Registers 4.5.BootROM Reset Factor Register (SYSC_RSTCAUSEBT)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit1 RVD: RAM retention low-voltage detection reset detection bit (Correct) [bit1] RVD: RAM Retention Low-Voltage Detection Reset Detection Bit

Section	Change Results
CHAPTER4: Reset 4.Registers 4.5.BootROM Reset Factor Register (SYSC_RSTCAUSEBT)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit0 PONR: Power-on reset detection bit (Correct) [bit0] PONR: Power-on Reset Detection Bit
CHAPTER4: Reset 4.Registers 4.6.BootROM Extended CSV Reset Factor Register (SYSC_EXCSVRSTCAUSEBT)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit31:8 Reserved (Correct) [bit31:8] Reserved
CHAPTER4: Reset 4.Registers 4.6.BootROM Extended CSV Reset Factor Register (SYSC_EXCSVRSTCAUSEBT)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit7 CSVSR3: SSCG3 clock supervisor reset detection bit (Correct) [bit7] CSVSR3: SSCG3 Clock Supervisor Reset Detection Bit
CHAPTER4: Reset 4.Registers 4.6.BootROM Extended CSV Reset Factor Register (SYSC_EXCSVRSTCAUSEBT)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit6 CSVSR2: SSCG2 clock supervisor reset detection bit (Correct) [bit6] CSVSR2: SSCG2 Clock Supervisor Reset Detection Bit
CHAPTER4: Reset 4.Registers 4.6.BootROM Extended CSV Reset Factor Register (SYSC_EXCSVRSTCAUSEBT)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit5 CSVSR1: SSCG1 clock supervisor reset detection bit (Correct) [bit5] CSVSR1: SSCG1 Clock Supervisor Reset Detection Bit

Section	Change Results
CHAPTER4: Reset 4.Registers 4.6.BootROM Extended CSV Reset Factor Register (SYSC_EXCSV RSTCAUSEBT)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit4 Reserved (Correct) [bit4] Reserved
CHAPTER4: Reset 4.Registers 4.6.BootROM Extended CSV Reset Factor Register (SYSC_EXCSV RSTCAUSEBT)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit3 CSVPR3: PLL3 clock supervisor reset detection bit (Correct) [bit3] CSVPR3: PLL3 Clock Supervisor Reset Detection Bit
CHAPTER4: Reset 4.Registers 4.6.BootROM Extended CSV Reset Factor Register (SYSC_EXCSV RSTCAUSEBT)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit2 CSVPR2: PLL2 clock supervisor reset detection bit (Correct) [bit2] CSVPR2: PLL2 Clock Supervisor Reset Detection Bit
CHAPTER4: Reset 4.Registers 4.6.BootROM Extended CSV Reset Factor Register (SYSC_EXCSV RSTCAUSEBT)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit1 CSVPR1: PLL1 clock supervisor reset detection bit (Correct) [bit1] CSVPR1: PLL1 Clock Supervisor Reset Detection Bit
CHAPTER4: Reset 4.Registers 4.6.BootROM Extended CSV Reset Factor Register (SYSC_EXCSV RSTCAUSEBT)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit0 Reserved (Correct) [bit0] Reserved

Section	Change Results
CHAPTER4: Reset 4.Registers 4.7.BootROM PowerDomain Reset Factor Register (SYSC_PDRST CAUSEBT)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit31:18 Reserved (Correct) [bit31:18] Reserved
CHAPTER4: Reset 4.Registers 4.7.BootROM PowerDomain Reset Factor Register (SYSC_PDRST CAUSEBT)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit17 PR6R1: PowerDomain6_1 reset detection bit (Correct) [bit17] PR6R1: PowerDomain6_1 Reset Detection Bit
CHAPTER4: Reset 4.Registers 4.7.BootROM PowerDomain Reset Factor Register (SYSC_PDRST CAUSEBT)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit16 PR6R0: PowerDomain6_0 reset detection bit (Correct) [bit16] PR6R0: PowerDomain6_0 Reset Detection Bit
CHAPTER4: Reset 4.Registers 4.7.BootROM PowerDomain Reset Factor Register (SYSC_PDRST CAUSEBT)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit15 PR5R3: PowerDomain5_3 reset detection bit (Correct) [bit15] PR5R3: PowerDomain5_3 Reset Detection Bit
CHAPTER4: Reset 4.Registers 4.7.BootROM PowerDomain Reset Factor Register (SYSC_PDRST CAUSEBT)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit14 PR5R2: PowerDomain5_2 reset detection bit (Correct) [bit14] PR5R2: PowerDomain5_2 Reset Detection Bit

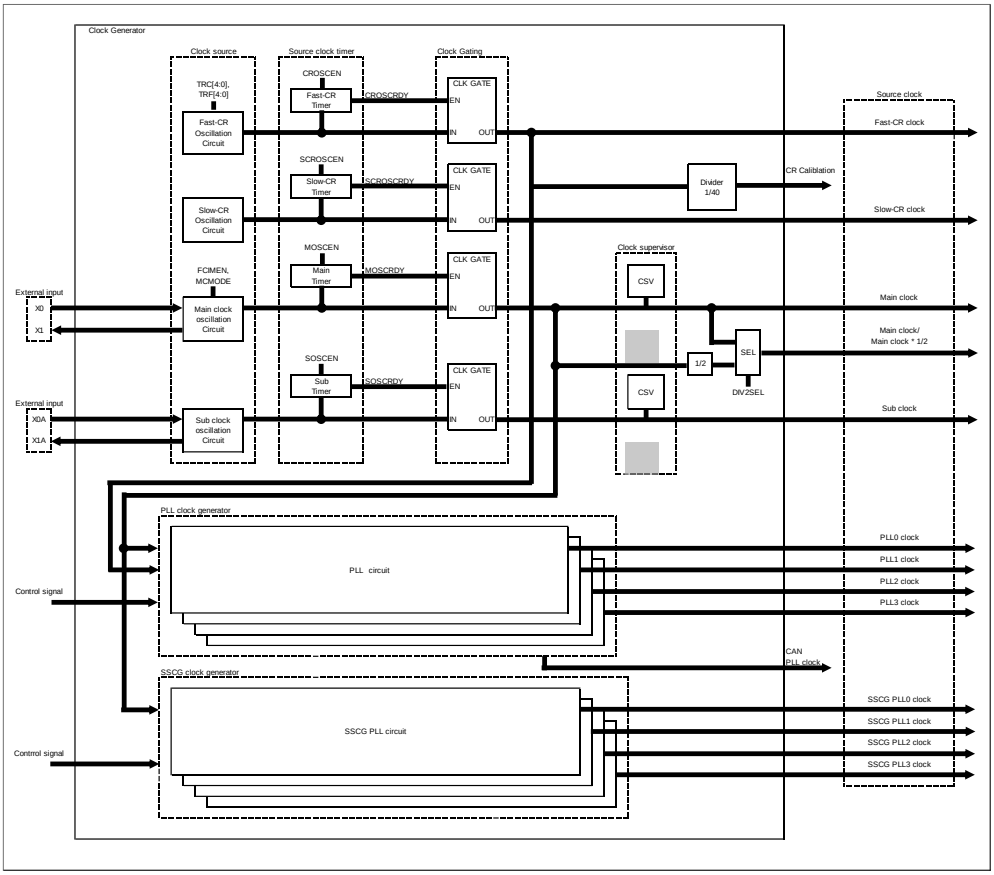
Section	Change Results
CHAPTER4: Reset 4.Registers 4.7.BootROM PowerDomain Reset Factor Register (SYSC_PDRST CAUSEBT)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit13 PR5R1: PowerDomain5_1 reset detection bit (Correct) [bit13] PR5R1: PowerDomain5_1 Reset Detection Bit
CHAPTER4: Reset 4.Registers 4.7.BootROM PowerDomain Reset Factor Register (SYSC_PDRST CAUSEBT)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit12 PR5R1: PowerDomain5_0 reset detection bit (Correct) [bit12] PR5R1: PowerDomain5_0 Reset Detection Bit
CHAPTER4: Reset 4.Registers 4.7.BootROM PowerDomain Reset Factor Register (SYSC_PDRST CAUSEBT)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit11:10 Reserved (Correct) [bit11:10] Reserved
CHAPTER4: Reset 4.Registers 4.7.BootROM PowerDomain Reset Factor Register (SYSC_PDRST CAUSEBT)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit9 PR4R1: PowerDomain4_1 reset detection bit (Correct) [bit9] PR4R1: PowerDomain4_1 Reset Detection Bit
CHAPTER4: Reset 4.Registers 4.7.BootROM PowerDomain Reset Factor Register (SYSC_PDRST CAUSEBT)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit8 PR4R0: PowerDomain4_0 reset detection bit (Correct) [bit8] PR4R0: PowerDomain4_0 Reset Detection Bit

Section	Change Results
CHAPTER4: Reset 4.Registers 4.7.BootROM PowerDomain Reset Factor Register (SYSC_PDRSTCAUSEBT)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit7:5 Reserved (Correct) [bit7:5] Reserved
CHAPTER4: Reset 4.Registers 4.7.BootROM PowerDomain Reset Factor Register (SYSC_PDRSTCAUSEBT)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit4 PR3R0: PowerDomain3_0 reset detection bit (Correct) [bit4] PR3R0: PowerDomain3_0 Reset Detection Bit
CHAPTER4: Reset 4.Registers 4.7.BootROM PowerDomain Reset Factor Register (SYSC_PDRSTCAUSEBT)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit3:1 Reserved (Correct) [bit3:1] Reserved
CHAPTER4: Reset 4.Registers 4.7.BootROM PowerDomain Reset Factor Register (SYSC_PDRSTCAUSEBT)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit0 PD2R0: PowerDomain2 reset detection bit (Correct) [bit0] PD2R0: PowerDomain2 Reset Detection Bit
CHAPTER4: Reset 4.Registers 4.8.PowerDomain Reset Status Register (SYSC_PDRSTSTATUS)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit31:18 Reserved (Correct) [bit31:18] Reserved

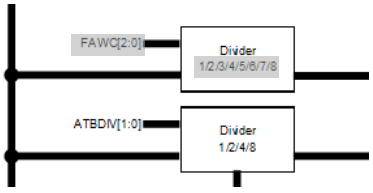
Section	Change Results
CHAPTER4: Reset 4.Registers 4.8.PowerDomain Reset Status Register (SYSC_PDRSTSTATUS)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit17 PD6RS1: PowerDomain6_1 reset status bit (Correct) [bit17] PD6RS1: PowerDomain6_1 Reset Status Bit
CHAPTER4: Reset 4.Registers 4.8.PowerDomain Reset Status Register (SYSC_PDRSTSTATUS)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit16 PD6RS0: PowerDomain6_0 reset detection bit (Correct) [bit16] PD6RS0: PowerDomain6_0 Reset Detection Bit
CHAPTER4: Reset 4.Registers 4.8.PowerDomain Reset Status Register (SYSC_PDRSTSTATUS)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit15 PD5RS3: PowerDomain5_3 reset detection bit (Correct) [bit15] PD5RS3: PowerDomain5_3 Reset Detection Bit
CHAPTER4: Reset 4.Registers 4.8.PowerDomain Reset Status Register (SYSC_PDRSTSTATUS)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit14 PD5RS2: PowerDomain5_2 reset detection bit (Correct) [bit14] PD5RS2: PowerDomain5_2 Reset Detection Bit
CHAPTER4: Reset 4.Registers 4.8.PowerDomain Reset Status Register (SYSC_PDRSTSTATUS)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit13 PD5RS1: PowerDomain5_1 reset detection bit (Correct) [bit13] PD5RS1: PowerDomain5_1 Reset Detection Bit

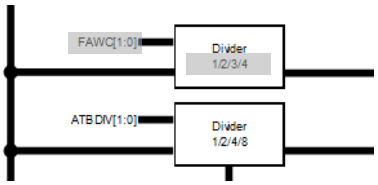
Section	Change Results
CHAPTER4: Reset 4.Registers 4.8.PowerDomain Reset Status Register (SYSC_PDRSTSTATUS)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit12 PD5RS1: PowerDomain5_0 reset detection bit (Correct) [bit12] PD5RS1: PowerDomain5_0 Reset Detection Bit
CHAPTER4: Reset 4.Registers 4.8.PowerDomain Reset Status Register (SYSC_PDRSTSTATUS)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit11:10 Reserved (Correct) [bit11:10] Reserved
CHAPTER4: Reset 4.Registers 4.8.PowerDomain Reset Status Register (SYSC_PDRSTSTATUS)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit9 PD4RS1: PowerDomain4_1 reset detection bit (Correct) [bit9] PD4RS1: PowerDomain4_1 Reset Detection Bit
CHAPTER4: Reset 4.Registers 4.8.PowerDomain Reset Status Register (SYSC_PDRSTSTATUS)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit8 PD4RS0: PowerDomain4_0 reset detection bit (Correct) [bit8] PD4RS0: PowerDomain4_0 Reset Detection Bit
CHAPTER4: Reset 4.Registers 4.8.PowerDomain Reset Status Register (SYSC_PDRSTSTATUS)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit7:5 Reserved (Correct) [bit7:5] Reserved

Section	Change Results
CHAPTER4: Reset 4.Registers 4.8.PowerDomain Reset Status Register (SYSC_PDRSTSTATUS)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit4 PD3RS0: PowerDomain3_0 reset detection bit (Correct) [bit4] PD3RS0: PowerDomain3_0 Reset Detection Bit
CHAPTER4: Reset 4.Registers 4.8.PowerDomain Reset Status Register (SYSC_PDRSTSTATUS)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit3:1 Reserved (Correct) [bit3:1] Reserved
CHAPTER4: Reset 4.Registers 4.8.PowerDomain Reset Status Register (SYSC_PDRSTSTATUS)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit0 PD2RS0: PowerDomain2 reset detection bit (Correct) [bit0] PD2RS0: PowerDomain2 Reset Detection Bit

Section	Change Results
CHAPTER5: Clock System 2.Configuration and Block Diagram 2.2.Clock Generator Configuration and Block Diagram	Revised “Figure 2-2 Clock Generator Block Diagram” as follows. (Correct) 

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER5: Clock System
2.Configuration and Block diagram
2.3.Clock Distributer Configuration and Block Diagram</td><td>Revised “Figure 2-5 Clock domain0 Block Diagram” as follows.
The bit width of FAWC was corrected.
FAWC[2:0] → FAWC[1:0]
Divider 1/2/3/4/5/6/7/8 → 1/2/3/4

(before)


(after)

</td></tr>
<tr><td>CHAPTER5: Clock System
5. Registers
5.1.2. Main Oscillator Control Register(SYSC_MOSCCNTR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
<table>
<tr><td>bit</td><td>bit 7</td><td>bit 6</td><td>bit 5</td><td>bit 4</td><td>bit 3</td><td>bit 2</td><td>bit 1</td><td>bit 0</td></tr>
<tr><td>Field</td><td colspan="7">Reserved</td><td>FCIMEN</td></tr>
<tr><td>Data Attribute</td><td colspan="7">R0,WX</td><td>R/W</td></tr>
<tr><td>Prot_Attr</td><td colspan="8">WPS</td></tr>
<tr><td>Initial Value</td><td colspan="7">0000000</td><td>0</td></tr>
</table>

(Correct)
<table>
<tr><td>bit</td><td>bit 7</td><td>bit 6</td><td>bit 5</td><td>bit 4</td><td>bit 3</td><td>bit 2</td><td>bit 1</td><td>bit 0</td></tr>
<tr><td>Field</td><td colspan="7">Reserved</td><td>FCIMEN</td></tr>
<tr><td>Data Attribute</td><td colspan="7">R0,WX</td><td>R0,W0</td></tr>
<tr><td>Prot_Attr</td><td colspan="8">WPS</td></tr>
<tr><td>Initial Value</td><td colspan="7">0000000</td><td>0</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER5: Clock System
5. Registers
5.1.2. Main Oscillator Control Register(SYSC_MOSCCNTR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit0] FCIMEN : Fast Main Clock Input Enable Control bit
This bit can be set to support higher frequency clock at Main oscillator input pin.
bit Description
0 Fast clock input disable (Main oscillator input)
1 Fast clock input enable

(Correct)
[bit0] FCIMEN : Fast Main Clock Input Enable Control bit
Always set to 0.</td></tr>
<tr><td>CHAPTER6:Low Power Consumption
3.Explanation of Operation
3.5.Interrupts</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
<table><tr><td>When the value written to the RUN update enable register is invalid</td><td>SYSC0_SYSERRIR1: RUNTRGERRIF</td><td>Always enabled because of the NMI level</td><td>Writing "1" to the SYSC0_SYSERRICLR 1:RUNTRGERRICLR bit</td></tr></table>
(Correct)
<table><tr><td>When the value written to the RUN Profile Update Trigger Register is invalid</td><td>SYSC0_SYSERRIR1: RUNTRGERRIF</td><td>Always enabled because of the NMI level</td><td>Writing "1" to the SYSC0_SYSERRICLR 1:RUNTRGERRICLR bit</td></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER6:Low Power Consumption
4. Operation Procedure
4.2.Transition from RUN to PSS</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
The following figure is a flowchart of RUN to PSS transition operations.

(Correct)
■Prepare Wakeup from PSS Mode by Interrupt

There are multiple ways to setup interrupts for wakeup from PSS mode. The major steps that apply in most cases are described in the following list.
Table 4-1 Steps to prepare Wakeup from PSS by Interrupt
<table>
<tr><th>Step</th><th>Purpose</th><th>Action</th></tr>
<tr><td>S1</td><td>Make sure that the CPU accepts interrupts.</td><td>Set the I-flag in PSR of the CPU core to "0". This
enables any interrupt that might occur.</td></tr>
<tr><td>S1</td><td>If it is intended to finish pending interrupts before
PSS mode, wait for the indication from IRCn.</td><td>Poll the IRCn_IRQST:nIRQ flag. While reading
"0", not all ISRs are finished. Once the logic value
"1" is read, the IRC will not accept any
subsequent interrupts. (*1)</td></tr>
<tr><td>S1</td><td>Configure the desired wake-up sources.</td><td>Enable the interrupts that are intended for wake-up, disable others. (*2)</td></tr>
<tr><td>S1</td><td>Remove the lock from IRC.</td><td>Write to any of the registers that are listed in the
description of IRCn_IRQST:nIRQ.
- IRCn_IRQVAr
- IRCn_IRQPL0-127
- IRCn_IRQS0-15
- IRCn_IRQR0-15
- IRCn_IRQCES0-15
- IRCn_IRQCEC0-15
- IRCn_IRQCE0-15
- IRCn_IRQHC
- IRCn_IRQPLM
- IRCn_CSR</td></tr>
<tr><td>S1</td><td>Prepare the transition from RUN to PSS mode</td><td>Follow the steps for transition to PSS mode as
described above.</td></tr>
<tr><td>S1</td><td>Transition to PSS mode</td><td>Execute the WFI instruction.</td></tr>
</table>

Notes:
− (*1): If the read access in step S2 returns a pending interrupt request, the processor will execute the interrupt service routine. After return from interrupt, the IRQ interrupt status bit nIRQ will show "end of return from interrupt instruction".
− (*2): If the interrupt should be used for wakeup without a specific interrupt service routine, this can be achieved by setting the I-Flag to "1" before S3. It must be considered that any write access to one of the registers listed in S4 will remove the lock from the IRC.
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER6:Low Power Consumption 5.Registers</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)<table><tr><td>0x0000_0280</td><td>SYSC0_SYSIDR</td></tr><tr><td>0x0000_0284</td><td>SYSC0_SYSPFIDR</td></tr></table>
(Correct)<table><tr><td>0x0000_0280</td><td>Reserved</td></tr><tr><td>0x0000_0284</td><td>Reserved</td></tr></table></td></tr>
<tr><td>CHAPTER6:Low Power Consumption 5.Registers</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)<table><tr><td>0x0000_0688</td><td>SYSC0_SPESPSWCFGR0</td></tr><tr><td>0x0000_068C</td><td>SYSC0_SPESPSWCFGR1</td></tr><tr><td>0x0000_0690</td><td>SYSC0_SPEWPSWCFGR0</td></tr><tr><td>0x0000_0694</td><td>SYSC0_SPEWPSWCFGR1</td></tr></table>
(Correct)<table><tr><td>0x0000_0688</td><td></td></tr><tr><td>0x0000_068C</td><td>Reserved</td></tr><tr><td>0x0000_0690</td><td>Reserved</td></tr><tr><td>0x0000_0694</td><td>Reserved</td></tr></table></td></tr>
<tr><td>CHAPTER6:Low Power Consumption 5.Registers 5.2.5.RUN Clock Divider Register (SYSC0_RUNCKDIVR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit31:5] Reserved: Reserved bits

(Correct)
[bit31:12] Reserved: Reserved bits</td></tr>
<tr><td>CHAPTER6:Low Power Consumption 5.Registers 5.2.6. RUN PLLx Control Register (SYSC0_RUNPLLxCNTR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit1:0] PLLDIVL: PLLx input clock divider setting bits

(Correct)
[bit1:0] PLLxDIVL: PLLx Input Clock Divider Setting Bits</td></tr>
</table>

Section	Change Results
CHAPTER6:Low Power Consumption 5.Registers 5.3.5. PSS Clock Divider Register (SYSC0_PSSCKDIVR)	Features of "For details" should be corrected as indicated by the shading below. (Error) [bit31:5] Reserved: Reserved bits (Correct) [bit31:12] Reserved: Reserved bits
CHAPTER6:Low Power Consumption 5.Registers 5.3.8. PSS SSCGx Control Register 1 (SYSC0_PSSSSCGxCNTR1)	Features of "For details" should be corrected as indicated by the shading below. (Error) [bit18:17] SSCGFREQ: SSCG PLL clock modulation frequency selection bits (Correct) [bit18:17] SSCGxFREQ: SSCG PLLx Clock modulation Frequency Selection Bits
CHAPTER6:Low Power Consumption 5.Registers 5.3.8. PSS SSCGx Control Register 1 (SYSC0_PSSSSCGxCNTR1)	Features of "For details" should be corrected as indicated by the shading below. (Error) [bit16] SSCGMODE: SSCG PLL modulation mode setting bit (Correct) [bit16] SSCGxMODE: SSCG PLLx modulation Mode setting bit
CHAPTER6:Low Power Consumption 5.Registers 5.3.8. PSS SSCGx Control Register 1 (SYSC0_PSSSSCGxCNTR1)	Features of "For details" should be corrected as indicated by the shading below. (Error) [bit9:0] SSCGRATE: SSCG PLL clock modulation ratio control bits (Correct) [bit9:0] SSCGxRATE: SSCG PLLx Clock modulation Ratio Control Bits

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER6:Low Power Consumption
5.Registers
5.4.3. APP Clock Selection Register (SYSC0_APPCKSELR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit3:0] CDMCUCCSL: Clock domain MCUC clock selection bits.

<table><tr><th>bit3:0</th><th>Description</th></tr></table>
(Correct)
[bit2:0] CDMCUCCSL: Clock Domain MCUC Clock Selection Bits.

<table><tr><th>bit2:0</th><th>Description</th></tr></table></td></tr>
<tr><td>CHAPTER6:Low Power Consumption
5.Registers
5.7.1. System Special Setting Register (SYSC0_SPECFGR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit3:0] EX12VRSTCNT: 1.2V External power supply stabilization time setting bits
These bits set the stabilization time for 1.2V external power supply.
<table><tr><th rowspan="2">bit3:0</th><th colspan="2">High-speed CR clock</th></tr><tr><th>4MHz (ms)</th><th>4MHz (ms)</th></tr></table>
(Correct)
[bit3:0] EX12VRSTCNT: 1.2V External Power Supply Stabilization Time Setting Bits
These bits set the stabilization time for 1.2V external power supply.
<table><tr><th rowspan="2">bit3:0</th><th colspan="2">High-speed CR clock</th></tr><tr><th>4MHz (ms)</th><th>8MHz (ms)</th></tr></table></td></tr>
<tr><td>CHAPTER6:Low Power Consumption
5.Registers
5.11.6. RUN Clock Source Enable Register 2 (SYSC1_RUNCKER2)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit3] ENCLKCD1B0: CD1B0 clock oscillation enable bit
This bit sets the CD1B0 clock.

(Correct)
[bit3] ENCLKCD4B0: CD4B0 Clock Oscillation Enable Bit
This bit sets the CD4B0 clock.</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER6:Low Power Consumption
5.Registers
5.12.6. PSS Clock Source Enable Register 2(SYSC1_PSSCKER2)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit3] ENCLKCD1B0: CD1B0 clock oscillation enable bit
This bit sets the CD1B0 clock.

(Correct)
[bit3] ENCLKCD4B0: CD4B0 Clock Oscillation Enable Bit
This bit sets the CD4B0 clock.</td></tr>
<tr><td>CHAPTER6:Low Power Consumption
5.Registers
5.14.3. STS Clock Selection Register 2 (SYSC1_STSCKSELR2)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
<table><tr><th>bit10:8</th><th>Description</th></tr><tr><td>101</td><td>SSCG0 clock</td></tr><tr><td>110</td><td>Setting prohibited</td></tr><tr><td>111</td><td>Clock fixed at "L"</td></tr></table>
(Correct)
<table><tr><th>bit10:8</th><th>Description</th></tr><tr><td>101</td><td>SSCG0 clock</td></tr><tr><td>110</td><td>PLL1 clock</td></tr><tr><td>111</td><td>Clock fixed at "L"</td></tr></table></td></tr>
<tr><td>CHAPTER7: Low-voltage Detection
1.Overview
1.1.Features</td><td>The function should be deleted as indicated by the shading below.

(Error)
Internal low-voltage detection
This section explains the features of internal low-voltage detection.
■Function: A power-on reset and internal low-voltage detection reset are generated upon the detection of a voltage from the preset detection voltage.
■Preset detection voltage: 0.85 V±0.05 V Release voltage: 0.925 V±0.05 V
■Operation: Always active.
The RAM contents after a power-on reset cannot be guaranteed.

(Correct)
Internal Low-Voltage Detection
This section explains the features of internal low-voltage detection.
■Operation: Always active.
The RAM contents after a power-on reset cannot be guaranteed.</td></tr>
</table>

Section	Change Results
CHAPTER7: Low-voltage Detection 1.Overview 1.1.Features	The function should be deleted as indicated by the shading below. (Error) 1.2 V power supply low-voltage detection The 1.2 V power supply low-voltage detection circuit has two channels. This section explains the features of 1.2 V power supply low-voltage detection. ■Function: A 1.2 V power supply low-voltage detection reset or interrupt is generated upon the detection of a voltage from the preset detection voltage. ■Preset detection voltage: Detection voltage:0.77V±0.05V Release voltage:0.845V±0.05V Detection voltage:0.87V±0.05V Release voltage:0.945V±0.05V Detection voltage:0.97V±0.05V Release voltage:1.045V±0.05V Detection voltage:1.07V±0.05V Release voltage:1.145V±0.05V ■Operation: Always active. The RAM contents after a 1.2 V power supply low-voltage detection reset cannot be guaranteed. (Correct) 1.2 V power supply low-voltage detection The 1.2 V power supply low-voltage detection circuit has two channels. This section explains the features of 1.2 V power supply low-voltage detection. ■Operation: Always active. The RAM contents after a 1.2 V power supply low-voltage detection reset cannot be guaranteed.

Section	Change Results
CHAPTER7: Low-voltage Detection 1.Overview 1.1.Features	The function should be deleted as indicated by the shading below. (Error) 3.3/5.0 V power supply low-voltage detection This section explains the features of 3.3/5.0 V power supply low-voltage detection. The 3.3/5.0 V power supply low-voltage detection circuit has two channels. ■Function: A 3.3/5.0 V power supply low-voltage detection reset or interrupt is generated upon the detection of a voltage within plus or minus the detection error value in volts from the preset detection voltage. ■Preset detection voltage: Detection voltage:2.35V±0.15V Release voltage:2.45V±0.15V Detection voltage:2.75V±0.15V Release voltage:2.85V±0.15V Detection voltage:2.85V±0.15V Release voltage:2.95V±0.15V Detection voltage:3.6V±0.2V Release voltage:3.7V±0.2V Detection voltage:3.8V±0.2V Release voltage:3.9V±0.2V Detection voltage:4.0V±0.2V Release voltage:4.1V±0.2V Detection voltage:4.2V±0.2V Release voltage:4.3V±0.2V Detection voltage:4.4V±0.2V Release voltage:4.5V±0.2V Detection voltage:4.65V±0.25V Release voltage:4.75V±0.25V ■Operation: Operation can be switched between the active and inactive states by user settings. The selection to generate an interrupt or reset is also available. Furthermore, the settings for RUN/PSS can be switched using a profile. In the case where a reset is generated, the reset is issued after all clocks are stopped in order to guarantee the RAM contents after an external low-voltage detection reset. For details, see the chapter "RESET." (Correct) 3.3/5.0 V power supply low-voltage detection This section explains the features of 3.3/5.0 V power supply low-voltage detection. The 3.3/5.0 V power supply low-voltage detection circuit has two channels. ■Function: A 3.3/5.0 V power supply low-voltage detection reset or interrupt is generated upon the detection of a voltage within plus or minus the detection error value in volts from the preset detection voltage. ■Operation: Operation can be switched between the active and inactive states by user settings. The selection to generate an interrupt or reset is also available. Furthermore, the settings for RUN/PSS can be switched using a profile. In the case where a reset is generated, the reset is issued after all clocks are stopped in order to guarantee the RAM contents after an external low-voltage detection reset. For details, see the chapter "RESET."

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER7: Low-voltage Detection
1.Overview
1.1.Features</td><td>The function should be deleted as indicated by the shading below.

(Error)
Table Table7-1Low-voltage detection Specifications Table
<table>
<tr><td>Kind</td><td>Detection Voltage</td><td>Release voltage</td><td>Detection error:</td><td>Operation</td><td>Note</td></tr>
<tr><td>Internal Low-voltage Detection</td><td>0.85V</td><td>0.925 V</td><td>± 0,05V</td><td>Reset is Always active</td><td>Values in RAM can not be guaranteed</td></tr>
<tr><td>1.2V Power Supply Low-voltage Detection</td><td>0.77V
0.87V
0.97V
1.07V</td><td>0.845 V
0.945 V
1.045 V
1.145 V</td><td>±0.05 V</td><td>Interrupt or Reset
Switchable to active/inactive state through user settings</td><td>Values in RAM can not be guaranteed</td></tr>
<tr><td rowspan="3">3.3/5.0V Power Supply Low-voltage Detection</td><td>2.35V
2.75V
2.85V</td><td>2.45V
2.85V
2.95V</td><td>±0.15 V</td><td rowspan="3">Interrupt or Reset
Switchable to active/inactive state through user settings</td><td rowspan="3"></td></tr>
<tr><td>3.6V
3.8V
4.0V
4.2V
4.4V</td><td>3.7V
3.9V
4.1V
4.3V
4.5V</td><td>±0.2 V</td></tr>
<tr><td>4.65V</td><td>4.75V</td><td>± 0.25V</td></tr>
</table>
The following registers are defined below as "LVD registers."

(Correct)
Table Table7-1Low-Voltage Detection Specifications Table

The following registers are defined below as "LVD registers."</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER7:
Low-voltage
Detection
1.Overview
1.1.Features</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
The following bits are defined below as "LVD operation select bits."
Table7-3 LVD operation select bits
<table><tr><td>LVDL0S</td><td>1.2 V power supply low-voltage detection operation select bit ch0</td></tr><tr><td>LVDL1S</td><td>1.2 V power supply low-voltage detection operation select bit ch1</td></tr><tr><td>LVDH0S</td><td>3.3/5.0 V power supply low-voltage detection operation select bit ch0</td></tr><tr><td>LVDH1S</td><td>3.3/5.0 V power supply low-voltage detection operation select bit ch1</td></tr></table>
(Correct)
The following bits are defined below as "LVD operation select bits."
Table7-3 LVD Operation Select Bits
<table><tr><td>LVDL1S</td><td>1.2 V power supply low-voltage detection operation select bit ch1</td></tr><tr><td>LVDL2S</td><td>1.2 V power supply low-voltage detection operation select bit ch2</td></tr><tr><td>LVDH1S</td><td>3.3/5.0 V power supply low-voltage detection operation select bit ch1</td></tr><tr><td>LVDH2S</td><td>3.3/5.0 V power supply low-voltage detection operation select bit ch2</td></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER7:
Low-voltage
Detection
1.Overview
1.1.Features</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
The following bits are defined below as "LVD ready bits."
Table7-4 LVD ready bits
<table><tr><td>LVDL0R DY</td><td>1.2 V power supply low-voltage detection ready bit ch0</td></tr><tr><td>LVDL1R DY</td><td>1.2 V power supply low-voltage detection ready bit ch1</td></tr><tr><td>LVDH0R DY</td><td>3.3/5.0 V power supply low-voltage detection ready bit ch0</td></tr><tr><td>LVDH1R DY</td><td>3.3/5.0 V power supply low-voltage detection ready bit ch1</td></tr></table>
(Correct)
The following bits are defined below as "LVD ready bits."
Table7-4 LVD Ready Bits
<table><tr><td>LVDL1R DY</td><td>1.2 V power supply low-voltage detection ready bit ch1</td></tr><tr><td>LVDL2R DY</td><td>1.2 V power supply low-voltage detection ready bit ch2</td></tr><tr><td>LVDH1R DY</td><td>3.3/5.0 V power supply low-voltage detection ready bit ch1</td></tr><tr><td>LVDH2R DY</td><td>3.3/5.0 V power supply low-voltage detection ready bit ch2</td></tr></table></td></tr>
</table>

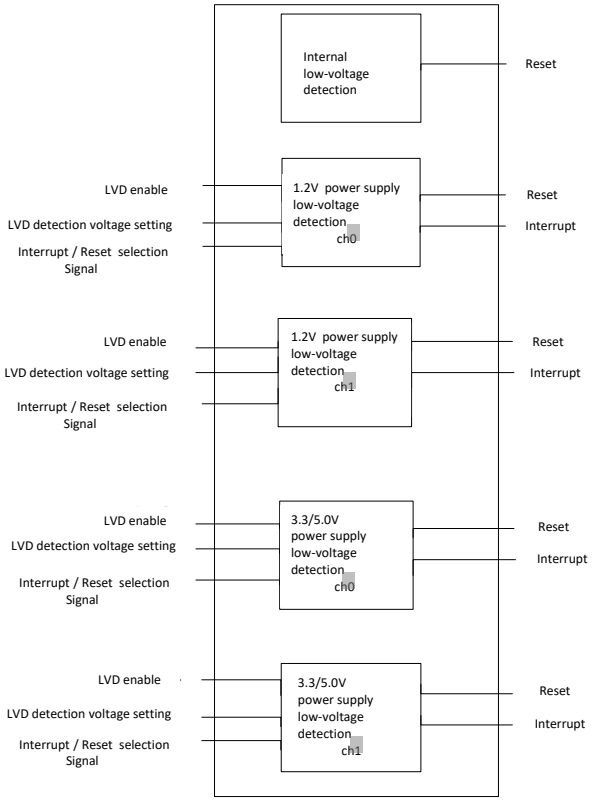
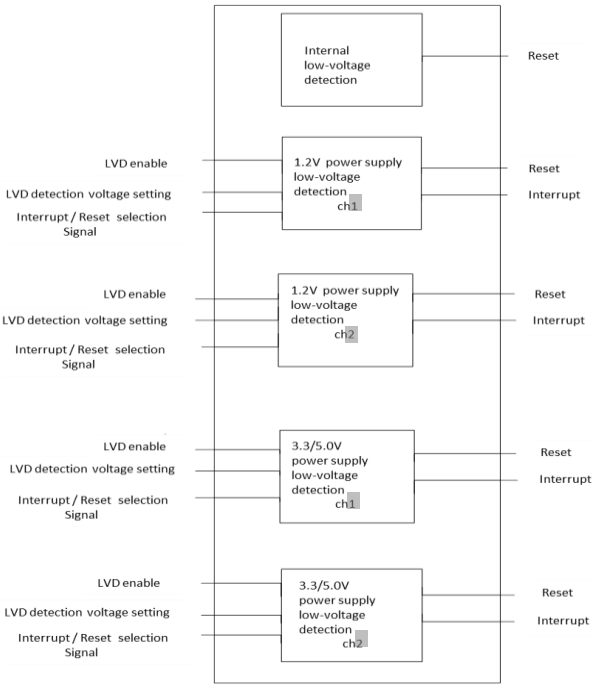
<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER7:
Low-voltage
Detection
1.Overview
1.1.Features</td><td>Features of "For details" should be corrected as indicated by the shading below.

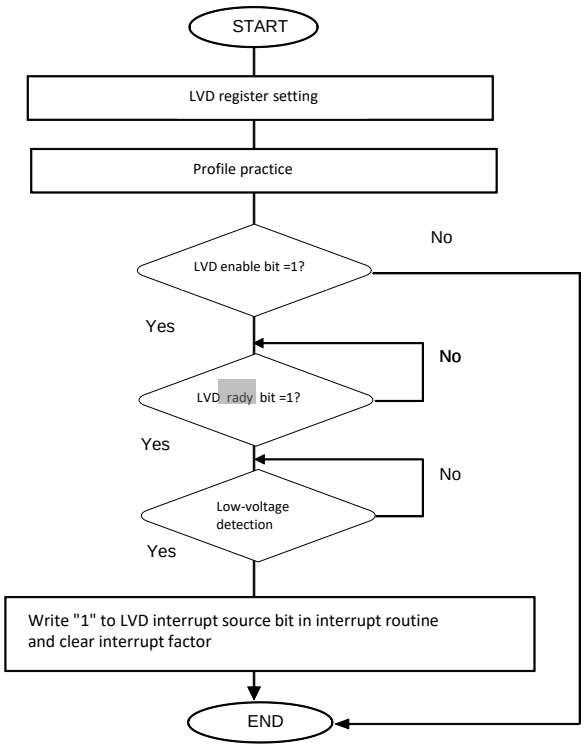
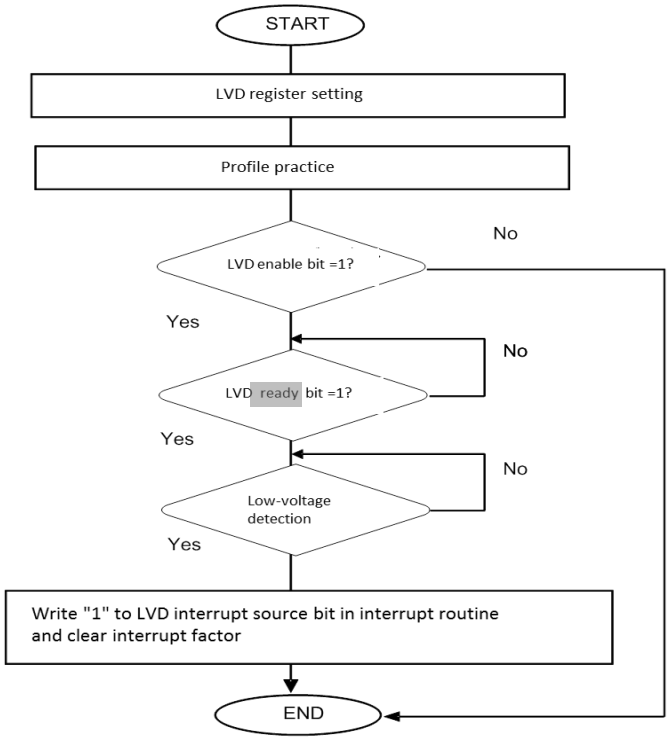
(Error)
The following bits are defined below as "LVD reset source bits."
Table7-5 LVD RZeset source bits
<table><tr><td>LVDL0E</td><td>1.2 V power supply low-voltage detection operation enable bit ch0</td></tr><tr><td>LVDL1E</td><td>1.2 V power supply low-voltage detection operation enable bit ch1</td></tr><tr><td>LVDH0E</td><td>3.3/5.0 V power supply low-voltage detection operation enable bit ch0</td></tr><tr><td>LVDL0E</td><td>1.2 V power supply low-voltage detection operation enable bit ch0</td></tr></table>
(Correct)
The following bits are defined below as "LVD reset source bits."
Table7-5 LVD Reset Source Bits
<table><tr><td>LVDL1E</td><td>1.2 V power supply low-voltage detection operation enable bit ch1</td></tr><tr><td>LVDL2E</td><td>1.2 V power supply low-voltage detection operation enable bit ch2</td></tr><tr><td>LVDH1E</td><td>3.3/5.0 V power supply low-voltage detection operation enable bit ch1</td></tr><tr><td>LVDL2E</td><td>3.3/5.0 V power supply low-voltage detection operation enable bit ch2</td></tr></table></td></tr>
<tr><td>CHAPTER7:
Low-voltage
Detection
1.Overview
1.1.Features</td><td>Features of "For details" should be corrected as indicated by the shading below.

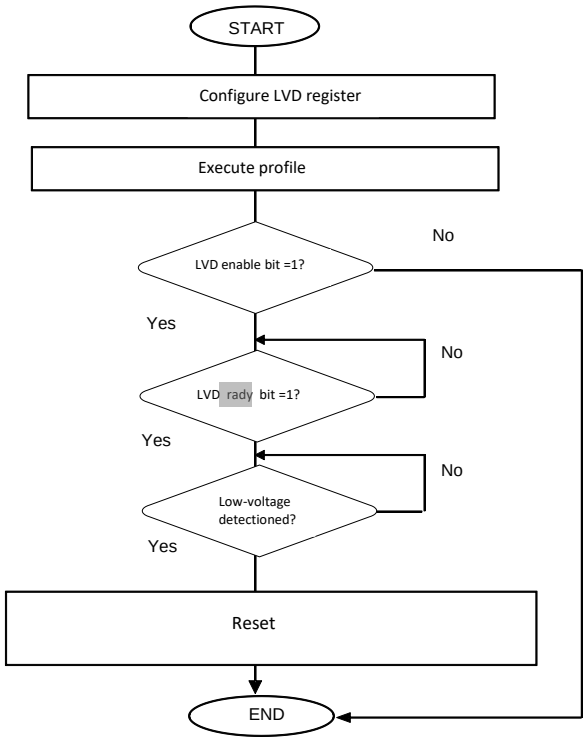
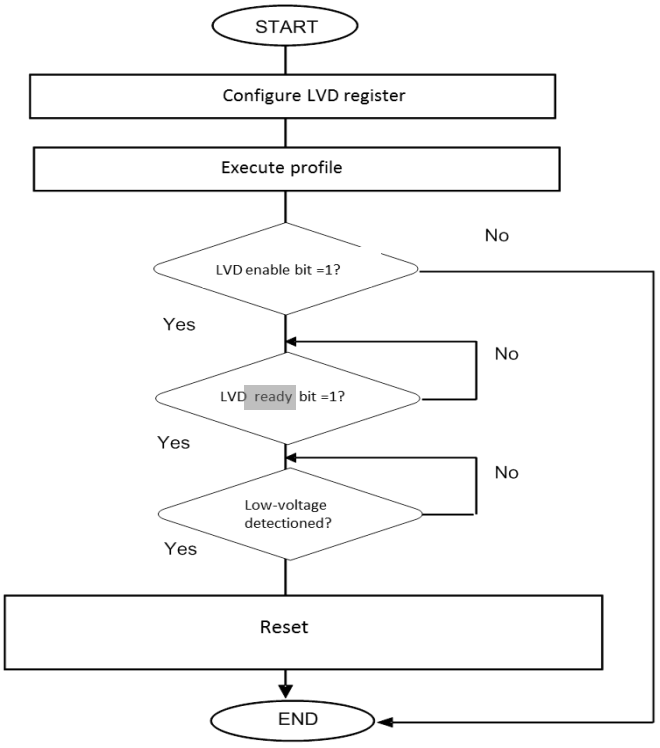
(Error)
Notes:
−In the XX series, functions are mapped as follows:
1.2 V power supply low-voltage detection ch0: 1.2 V internal regulator voltage monitoring
1.2 V power supply low-voltage detection ch1: 1.2 V external power supply monitoring
3.3/5.0 V power supply external low-voltage detection ch0: 5.0 V external power supply monitoring
3.3/5.0 V power supply external low-voltage detection ch1: 3.3 V external power supply monitoring

(Correct)
Notes:
−In the PF-Cluster series, functions are mapped as follows:
1.2 V power supply low-voltage detection ch1: 1.2 V internal regulator voltage monitoring
1.2 V power supply low-voltage detection ch2: 1.2 V external power supply monitoring
3.3/5.0 V power supply external low-voltage detection ch1: 5.0 V external power supply monitoring
3.3/5.0 V power supply external low-voltage detection ch2: 3.3 V external power supply monitoring</td></tr>
</table>

Section	Change Results
CHAPTER7: Low-voltage Detection 1.Overview 1.1.Features	Features of "For details" should be corrected as indicated by the shading below. (Error) −In the XX series, a low-voltage detection interrupt corresponds to an NMI. −The values for the LVD operation select bits vary depending on the chip. −For example, if a register value is fixed, only the functions indicated by the register value can be used. −Also, since operation in the initial state varies depending on the initial value, see explanation of registers. (Correct) −In the PF-Cluster series, a low-voltage detection interrupt corresponds to an NMI. −The values for the LVD operation select bits vary depending on the chip. −For example, if a register value is fixed, only the functions indicated by the register value can be used. −Also, since operation in the initial state varies depending on the initial value, see explanation of registers.

Section	Change Results
CHAPTER7: Low-voltage Detection 2.Configuration and Block Diagrams	Revised " Figure 2-1 Overall Diagram" as follows. (Error)  (Correct) 

Section	Change Results
CHAPTER7: Low-voltage Detection 4.Setting Procedure Examples	Revised "Figure 4-1 Setting the Low-voltage Detection for Interrupts" as follows. (Error)  (Correct) 

Section	Change Results
CHAPTER7: Low-voltage Detection 4.Setting Procedure Examples	Revised "Figure 4-2 Setting the Low-voltage Detection for Resets" as follows. (Error)  (Correct) 

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER7: Low-voltage Detection 5.Operation Examples</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
The change in the register value described here refers to the following bits.
<table>
<tr><td>LVDL0E : low voltage detect enable for 1.2V power supply, ch0</td></tr>
<tr><td>LVDL1E : low voltage detect enable for 1.2V power supply, ch1</td></tr>
<tr><td>LVDH0E : low voltage detect enable for 3.3V or 5.5V power supply, ch0</td></tr>
<tr><td>LVDH1E : low voltage detect enable for 3.3V or 5.5V power supply, ch1</td></tr>
</table>

(Correct)
The change in the register value described here refers to the following bits.
<table>
<tr><td>LVDL1E : low voltage detect enable for 1.2V power supply, ch1</td></tr>
<tr><td>LVDL2E : low voltage detect enable for 1.2V power supply, ch2</td></tr>
<tr><td>LVDH1E : low voltage detect enable for 3.3V or 5.5V power supply, ch1</td></tr>
<tr><td>LVDH2E : low voltage detect enable for 3.3V or 5.5V power supply, ch2</td></tr>
</table>
</td></tr>
<tr><td>CHAPTER8: Clock Supervisor 5.Registers 5.2. Main Clock Supervisor Setting Register 1 (SYSC_CSVMOCFGR1)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit16] JDGSEL: Judgment selection bit
Reset generation condition for when an abnormality is detected can be selected with this bit.
An interrupt can be generated when reset generation condition is not correspondence.
<table>
<tr><th>bit</th><th>Description</th></tr>
<tr><td>0</td><td>Monitoring clock is selected as the clock for clock domain 0/MCUC clock domain /software watchdog timer.</td></tr>
<tr><td>1</td><td>Monitoring clock is selected as the clock for clock domain 0/ MCUC clock domain.</td></tr>
</table>

(Correct)
[bit16] JDGSEL: Judgment Selection Bit
Reset generation condition for when an abnormality is detected can be selected with this bit.
An interrupt can be generated when reset generation condition is not correspondence.
<table>
<tr><th>bit</th><th>Description</th></tr>
<tr><td>0</td><td>Monitoring clock is selected as the clock for clock domain 0 or MCUC clock domain or software watchdog timer.</td></tr>
<tr><td>1</td><td>Monitoring clock is selected as the clock for clock domain 0 or MCUC clock domain.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER8: Clock Supervisor 5.Registers 5.4.Sub Clock Supervisor Setting Register 1 (SYSC_CSVSOCFGR1)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit16] JDGSEL: Judgment selection bit
Reset generation condition for when an abnormality is detected can be selected with this bit.
An interrupt can be generated when reset generation condition is not correspondence.
<table>
<tr><th>bit</th><th>Description</th></tr>
<tr><td>0</td><td>Monitoring clock is selected as the clock for clock domain 0/MCUC clock domain /software watchdog timer.</td></tr>
<tr><td>1</td><td>Monitoring clock is selected as the clock for clock domain 0/ MCUC clock domain.</td></tr>
</table>
(Correct)
[bit16] JDGSEL: Judgment Selection Bit
Reset generation condition for when an abnormality is detected can be selected with this bit.
An interrupt can be generated when reset generation condition is not correspondence.
<table>
<tr><th>bit</th><th>Description</th></tr>
<tr><td>0</td><td>Monitoring clock is selected as the clock for clock domain 0 or MCUC clock domain or software watchdog timer.</td></tr>
<tr><td>1</td><td>Monitoring clock is selected as the clock for clock domain 0 or MCUC clock domain.</td></tr>
</table>
</td></tr>
<tr><td>CHAPTER8: Clock Supervisor 5.Registers 5.6.PLL0 Clock Supervisor Setting Register 1 (SYSC_CSVPLL0CFGR1)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit16] JDGSEL: Judgment selection bit
Reset generation condition for when an abnormality is detected can be selected with this bit.
An interrupt can be generated when reset generation condition is not correspondence.
<table>
<tr><th>bit</th><th>Description</th></tr>
<tr><td>0</td><td>Monitoring clock is selected as the clock for clock domain 0/MCUC clock domain</td></tr>
<tr><td>1</td><td>- (same condition)</td></tr>
</table>
(Correct)
[bit16] JDGSEL: Judgment Selection Bit
Reset generation condition for when an abnormality is detected can be selected with this bit.
An interrupt can be generated when reset generation condition is not correspondence.
<table>
<tr><th>bit</th><th>Description</th></tr>
<tr><td>0</td><td>Monitoring clock is selected as the clock for clock domain 0 or MCUC clock domain</td></tr>
<tr><td>1</td><td>- (same condition)</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER8: Clock Supervisor 5.Registers 5.14.SSCG PLL0 Clock Supervisor Setting Register 1 (SYSC_CSVSP0CFGR1)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit16] JDGSEL: Judgment selection bit
Reset generation condition for when an abnormality is detected can be selected with this bit.
An interrupt can be generated when reset generation condition is not correspondence.
<table>
<tr><th>bit</th><th>Description</th></tr>
<tr><td>0</td><td>Monitoring clock is selected as the clock for clock domain 0/MCUC clock domain</td></tr>
<tr><td>1</td><td>- (same condition)</td></tr>
</table>

(Correct)
[bit16] JDGSEL: Judgment Selection Bit
Reset generation condition for when an abnormality is detected can be selected with this bit.
An interrupt can be generated when reset generation condition is not correspondence.
<table>
<tr><th>bit</th><th>Description</th></tr>
<tr><td>0</td><td>Monitoring clock is selected as the clock for clock domain 0 or MCUC clock domain</td></tr>
<tr><td>1</td><td>- (same condition)</td></tr>
</table>
</td></tr>
<tr><td>CHAPTER9: Source Clock Timer 5.Registers 5.3. High-speed CR Clock Timer Compare Prescaler Register (SYSC_FCRCTCPR)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit19:16] PSCL[3:0]: Prescale bits
These bits select the division ratio of the input clock of the high-speed CR clock timer.
<table>
<tr><th>Bit19:16</th><th>Description</th></tr>
</table>

(Correct)
[bit19:16] PSCL[3:0]: Prescale Bits
These bits select the division ratio of the input clock of the high-speed CR clock timer.
<table>
<tr><th>Bits</th><th>Description</th></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER9: Source Clock Timer 5.Registers 5.9. Low-speed CR Clock Timer Compare Prescaler Register (SYSC_SCRCTCPR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit19:16] PSCL[3:0]: Prescale bits
These bits select the division ratio of the input clock of the low-speed CR clock timer.
<table><tr><th>Bit19:16</th><th>Description</th></tr></table>
(Correct)
[bit19:16] PSCL[3:0]: Prescale Bits
These bits select the division ratio of the input clock of the high-speed CR clock timer.
<table><tr><th>Bits</th><th>Description</th></tr></table></td></tr>
<tr><td>CHAPTER9: Source Clock Timer 5.Registers 5.15. Main Clock Timer Compare Prescaler Register (SYSC_MOCTCPR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit19:16] PSCL[3:0]: Prescale bits
These bits select the division ratio of the input clock of the main clock timer.
<table><tr><th>Bit19:16</th><th>Description</th></tr></table>
(Correct)
bit19:16] PSCL[3:0]: Prescale Bits
These bits select the division ratio of the input clock of the main clock timer.
<table><tr><th>Bits</th><th>Description</th></tr></table></td></tr>
<tr><td>CHAPTER9: Source Clock Timer 5.Registers 5.21.Sub Clock Timer Compare Prescaler Register (SYSC_SOCTCPR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit19:16] PSCL[3:0]: Prescale bits
These bits select the division ratio of the input clock of the sub clock timer.
<table><tr><th>Bit19:16</th><th>Description</th></tr></table>
(Correct)
[bit19:16] PSCL[3:0]: Prescale Bits
These bits select the division ratio of the input clock of the sub clock timer.
<table><tr><th>Bits</th><th>Description</th></tr></table></td></tr>
</table>

Section	Change Results
CHAPTER10: Real Time Clock 2.Configuration and Block Diagram	Features of "For details" should be corrected as indicated by the shading below. (Error) Table 2-1 Block diagram of RTC (Correct) Figure 2-1 Block Diagram of RTC
CHAPTER10: Real Time Clock 2.Configuration and Block Diagram	Features of "For details" should be corrected as indicated by the shading below. (Error) Table 2-2 RTC Timer module diagram (Correct) Figure 2-2 RTC Timer Module Diagram
CHAPTER10: Real Time Clock 2.Configuration and Block Diagram	Features of "For details" should be corrected as indicated by the shading below. (Error) Table 2-3 Calibration module diagram (Correct) Figure 2-3 Calibration Module Diagram
CHAPTER10: Real Time Clock 2.Configuration and Block Diagram	Features of "For details" should be corrected as indicated by the shading below. (Error) Table 2-4 Partial Wake Up Trigger module diagram (Correct) Figure 2-4 Partial Wake Up Trigger Module Diagram
CHAPTER10: Real Time Clock 3. Operation of the Real Time Clock	Features of "For details" should be corrected as indicated by the shading below. (Error) Table 3-1 RTC Timer module operation (Correct) Figure 3-1 RTC Timer Module Operation
CHAPTER10: Real Time Clock 3. Operation of the Real Time Clock	Features of "For details" should be corrected as indicated by the shading below. (Error) The operation of the RTC Timer module, shown in Table 3-1, is described as follows: (Correct) The operation of the RTC Timer module, shown in Figure 3-1, is described as follows:

Section	Change Results
CHAPTER10: Real Time Clock 3. Operation of the Real Time Clock	Features of "For details" should be corrected as indicated by the shading below. (Error) Table 3-2 WOT pin operation (Correct) Figure 3-2 WOT Pin Operation
CHAPTER10: Real Time Clock 3. Operation of the Real Time Clock	Features of "For details" should be corrected as indicated by the shading below. (Error) The following steps describe how to set up the Calibration module. The calibration operation is shown in: Table 3-3 (Correct) The following steps describe how to set up the Calibration module. The calibration operation is shown in Figure 3-3:
CHAPTER10: Real Time Clock 3. Operation of the Real Time Clock	Features of "For details" should be corrected as indicated by the shading below. (Error) Table 3-3 Calibration operation (Correct) Figure 3-3 Calibration Operation
CHAPTER10: Real Time Clock 3. Operation of the Real Time Clock	Features of "For details" should be corrected as indicated by the shading below. (Error) Table 3-4 Cyclic Calibration (Correct) Figure 3-4 Cyclic Calibration
CHAPTER10: Real Time Clock 3. Operation of the Real Time Clock	Features of "For details" should be corrected as indicated by the shading below. (Error) Cyclic Calibration, as shown in Table 3-4, is done in Automatic Calibration mode. (Correct) Cyclic Calibration, as shown in Figure 3-4, is done in Automatic Calibration mode.

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER10: Real Time Clock
4. Registers
4.1. Timer Control Register (RTC_WTCR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
These bits are used to scale the Calibration value in (RTC_CNTCAL), which is determined by the Calibration module, before it is transferred to the Sub-Second Register.
<table><tr><th>Bit[2:0]</th><th>Description</th></tr></table>
(Correct)
These bits are used to scale the Calibration value in (RTC_CNTCAL), which is determined by the Calibration module, before it is transferred to the Sub-Second Register.
<table><tr><th>Bits</th><th>Description</th></tr></table></td></tr>
<tr><td>CHAPTER10: Real Time Clock
4. Registers
4.1. Timer Control Register (RTC_WTCR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit5:4] RCKSEL[1:0] : Clock Select for RTC
These bits are used to select the input clock for the RTC (CLKRTC).
<table><tr><th>Bit[2:0]</th><th>Description</th></tr></table>
(Correct)
[bit5:4] RCKSEL[1:0]: Clock select for RTC
These bits are used to select the input clock for the RTC (CLKRTC).
<table><tr><th>Bits</th><th>Description</th></tr></table></td></tr>
<tr><td>CHAPTER10: Real Time Clock
4. Registers
4.11. Calibration Duration Register (RTC_DURMW)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
Calculation Formula:
RTC_DURMW:DURMW[23:0] = fCLK_MAIN/(2^SCAL) x 0.25 s - 1

(Correct)
Calculation Formula:
RTC_DURMW:DURMW[23:0] = fCLK_MAIN/(2^SCAL) x 0.25 s</td></tr>
<tr><td>CHAPTER10: Real Time Clock
4. Registers
4.14.Partial Wake Up Trigger Control Register (RTC_PWUTRGCR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit31:1] Reserved : Reservation bit

(Correct)
[bit31:26] Reserved: Reservation bit</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER10: Real Time Clock
4. Registers
4.14.Partial Wake Up Trigger Control Register (RTC_PWUTRGCR)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit4] MD : Reload value setting bit to 8ms counter
<table>
<tr><th>MD</th><th>Reload value setting bit to 8ms counter</th></tr>
<tr><td>Writing</td><td>The reload value to 8ms counter is selected.
"0": RTC_PWUTRGCR.C8MRL[17:0] is selected.
The reload setting value is decided in the manual operation.
"1": The value in which the reload value to Sub−Second Counter is divided by 64 is selected.
0.25[s]÷32＝7.8125[ms]≒8[ms]
An appropriate value is set from a set value of Sub-Second register.</td></tr>
<tr><td>Reading</td><td>The writing value can be read.</td></tr>
</table>

(Correct)
[bit4] MD: Reload Value Setting Bit to 8ms Counter
<table>
<tr><th>MD</th><th>Reload value setting bit to 8ms counter</th></tr>
<tr><td>Writing</td><td>The reload value to 8ms counter is selected.
"0": RTC_PWUTRGCR.C8MRL[17:0] is selected.
The reload setting value is decided in the manual operation.
"1": The value in which the reload value to Sub−Second Counter is divided by 64 is selected.
0.25[s]÷32=7.8125[ms]≈8[ms]
An appropriate value is set from a set value of Sub-Second register.</td></tr>
<tr><td>Reading</td><td>The writing value can be read.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER10: Real Time Clock
4. Registers
4.14.Partial Wake Up Trigger Control Register (RTC_PWUTRGCR)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit2:0] SEL : Reload value setting bit to trigger counter
<table>
<tr><th>SEL</th><th>Reload value setting bit to 8ms counter</th></tr>
<tr><td>Writing</td><td>The reload value to the trigger counter that generates the trigger signal of Partial Wake Up is set. The interval time of Partial Wake Up can be set by setting the reload value.
In the trigger counter, 8ms counter is 3bit decreased counter , saying that ALL '0'.
The count value is reload, saying that ALL '0' as for a set value of this bit.
The relation between a set value and the interval time is shown below.
"000" : 8ms
"001" : 16ms
"010" : 24ms
"011" : 32ms
"100" : 40ms
"101" : 48ms
"110" : 56ms
"111" : 64ms</td></tr>
<tr><td>Reading</td><td>The writing value can be read.</td></tr>
</table>

(Correct)
[bit2:0] SEL[2:0]: Reload Value Setting Bit to Trigger Counter
<table>
<tr><th>SEL[2:0]</th><th>Reload value setting bit to 8ms counter</th></tr>
<tr><td>Writing</td><td>The reload value to the trigger counter that generates the trigger signal of Partial Wake Up is set. The interval time of Partial Wake Up can be set by setting the reload value.
In the trigger counter, 8ms counter is 3bit decreased counter , saying that ALL '0'.
The count value is reload, saying that ALL '0' as for a set value of this bit.
The relation between a set value and the interval time is shown below.
"000":8ms
"001":16ms
"010":24ms
"011":32ms
"100":40ms
"101":48ms
"110":56ms
"111":64ms</td></tr>
<tr><td>Reading</td><td>The writing value can be read.</td></tr>
</table>
</td></tr>
</table>

Section	Change Results
CHAPTER10: Real Time Clock 5. Usage Precaution	The function should be deleted as indicated by the shading below. (Error) Cautions ■If '1' is written to the update bit (RTC_WTCR:UPDT) at the same time the update completes, the update bit (RTC_WTCR:UPDT) is set to '0' ■When the Second counter holds a value of 59 even if '1' is written to the update bit (RTC_WTCR:UPDT), the Hour/Minute/Second counters are not updated, and the update bit remains '0'. In order to update the Hour/Minute/Second counters: - Write '0' to the start bit (RTC_WTCR:ST) - Clear the Hour/Minute/Second registers to ‘0’ - Write '1' to the start bit (RTC_WTCR:ST) ■If the peripheral clock is stopped after updating the Hour/Minute/Second counters using the update bit (RTC_WTCR:UPDT), read the Hour/Minute/Second Registers to confirm that they have been updated before stopping the peripheral clock (Correct) Cautions ■If '1' is written to the update bit (RTC_WTCR:UPDT) at the same time the update completes, the update bit (RTC_WTCR:UPDT) is set to '0' ■If the peripheral clock is stopped after updating the Hour/Minute/Second counters using the update bit (RTC_WTCR:UPDT), read the Hour/Minute/Second Registers to confirm that they have been updated before stopping the peripheral clock
CHAPTER10: Real Time Clock 5. Usage Precaution	The function should be deleted as indicated by the shading below. (Error) ■If the Sub-Second Register (RTC_WTBR) is set to '0', the Sub-second counter does not operate, causing the RTC module to also not operate ■If a carry operation occurs (e.g. Second counter overflows and Minute counter increments) while reading the Real Timer Register (RTC_WRT), inconsistent values may be read. To avoid this, the interrupts (RTC_WINS:SUBSEC, RTC_WINS:SEC, RTC_WINS:MIN, RTC_WINS:HOUR, RTC_WINS:DAY) should be used to read the time (HH/MM/SS) (Correct) ■If a carry operation occurs (e.g. Second counter overflows and Minute counter increments) while reading the Real Timer Register (RTC_WRT), inconsistent values may be read. To avoid this, the interrupts (RTC_WINS:SUBSEC, RTC_WINS:SEC, RTC_WINS:MIN, RTC_WINS:HOUR, RTC_WINS:DAY) should be used to read the time (HH/MM/SS)

Section	Change Results
CHAPTER10: Real Time Clock 5. Usage Precaution	The function should be deleted as indicated by the shading below. (Error) ■RTC cannot use the calibration function under following condition: RTC clock*2 < Bus clock ■Register(RTC_WTSR:RUNC)value is changed asynchronously with the bus clock. Therefore register value should read more than once. (Correct) ■Register(RTC_WTSR:RUNC)value is changed asynchronously with the bus clock. Therefore register value should read more than once.
CHAPTER12: Backup RAM Interface 3.Explanation of Operation 3.1. RAMECC Function	Features of "For details" should be corrected as indicated by the shading below. (Error) 2.Set the byte and bit for pseudo-error generation in the ECC pseudo-error generation control register (BURIF_EFEAR). (Correct) 2.Set the byte and bit for pseudo-error generation in the ECC pseudo-error generation control register (BURIF_EFECR).

Section	Change Results
CHAPTER12: Backup RAM Interface 3.Explanation of Operation 3.1. RAMECC Function	Features of "For details" should be corrected as indicated by the shading below. (Error) To enable a test mode, use the ECC data path control register (BURIF_EPDCR) and the ECC test mode key code control register (BURIF_ECCTKCCR). 1.In the ECC test mode key code control register (BURIF_ECCTKCCR), write "1" to a desired test enable bit (TM_ESKPG, TM_ESKPC, or TM_EEACC), and then consecutively write 0x0, 0x1, 0x2, and 0x3 to the KEY bits in that order. 2.Check bits (TM_ESKPG, TM_ESKPC, and TM_EEACC) in the status register (URIF_STATUS) to confirm that the setting of the ECC test mode key code control register (BURIF_ECCTKCCR) is correctly reflected. 3.Write "1" to the SKPG, SKPC, or EACC bit in the ECC data path control register (BURIF_EPDCR) to enable the test. (Correct) To enable a test mode, use the ECC data path control register (BURIF_EDPCR) and the ECC test mode key code control register (BURIF_ECCTKCCR). 1.In the ECC test mode key code control register (BURIF_ECCTKCCR), write "1" to a desired test enable bit (TM_ESKPG, TM_ESKPC, or TM_EEACC), and then consecutively write 0x0, 0x1, 0x2, and 0x3 to the KEY bits in that order. 2.Check bits (TM_ESKPG, TM_ESKPC, and TM_EEACC) in the status register (BURIF_STATUS) to confirm that the setting of the ECC test mode key code control register (BURIF_ECCTKCCR) is correctly reflected. 3.Write "1" to the SKPG, SKPC, or EACC bit in the ECC data path control register (BURIF_EDPCR) to enable the test.

Section	Change Results
CHAPTER12: Backup RAM Interface 3.Explanation of Operation 3.3. RAM Initialization	Features of "For details" should be corrected as indicated by the shading below. (Error) This is specified by the ITYP bit in the TEST initialization function register (BURIF_TICR). The value corresponding to the write value is written in the ECC area. The scope of RAM initialization is specified by the TEST start address register (BURIF_TASAR) and the TEST end address register (BURIF_TAEAR). RAM diagnosis must be performed by applying the following procedure. 1.Before starting diagnosis, read TRUN in the TEST diagnosis function register (BURIF_TTCR) and IRUN in the TEST initialization function register (BURIF_TICR) to confirm that their values are "0". If BURIF_TTCR.TRUN or BURIF_TICR.IRUN is not "0": −Wait until BURIF_TTCR.TRUN = 0, and then clear BURIF_TTCR.TCI. −Wait until BURIF_TICR.IRUN = 0, and then clear BURIF_TICR.ICI. 2.Write to the TEST key code control register (BURIF_TKCCR) 4 times consecutively in the order of "01h", "41h", "81h", and then "C1h" to start the diagnosis. When RAM initialization ends, the IRUN bit in the TEST initialization function register (BURIF_TICR) goes to "0" and RAM initialization ends. RAM initialization is forcibly terminated by writing 4 times consecutively in the order of "00h", "40h", "80h", and then "C0h" in the TEST key code control register (BURIF_TKCCR). The RAM initialization is terminated even if it has not yet been completed. In this case, the initialization result is not guaranteed. (Correct) This is specified by the ITYP bit in the TEST initialization function register (BURIF_TICR). The value corresponding to the write value is written in the ECC area. The scope of RAM initialization is specified by the TEST start address register (BURIF_TASAR) and the TEST end address register (BURIF_TAEAR). RAM initialization must be performed by applying the following procedure. 1.Before starting initialization, read TRUN in the TEST diagnosis function register (BURIF_TTCR) and IRUN in the TEST initialization function register (BURIF_TICR) to confirm that their values are "0". If BURIF_TTCR.TRUN or BURIF_TICR.IRUN is not "0": −Wait until BURIF_TTCR.TRUN = 0, and then clear BURIF_TTCR.TCI. −Wait until BURIF_TICR.IRUN = 0, and then clear BURIF_TICR.ICI. 2.Write to the TEST key code control register (BURIF_TKCCR) 4 times consecutively in the order of "01h", "41h", "81h", and then "C1h" to start the initialization. When RAM initialization ends, the IRUN bit in the TEST initialization function register (BURIF_TICR) goes to "0" and RAM initialization ends. RAM initialization is forcibly terminated by writing 4 times consecutively in the order of "00h", "40h", "80h", and then "C0h" in the TEST key code control register (BURIF_TKCCR). The RAM initialization is terminated even if it has not yet been completed. In this case, the initialization result is not guaranteed.

Section	Change Results
CHAPTER12: Backup RAM Interface 3.Explanation of Operation 3.3. RAM Initialization	Features of "For details" should be corrected as indicated by the shading below. (Error) Interrupt-related registers To generate an interrupt, write "1" to the interrupt generation enable bit (ICI) in the TEST initialization function register (BURIF_TICR) in accordance with the purpose, and then set the interrupt vector. The interrupt request flag is not cleared automatically. For this reason, clear it by software using the event clear register (BURIF_EVENTCLR.ICICLR) before returning from the interrupt processing. (Correct) Interrupt-related Registers To generate an interrupt, write "1" to the interrupt generation enable bit (ICIE) in the TEST initialization function register (BURIF_TICR) in accordance with the purpose, and then set the interrupt vector. The interrupt request flag is not cleared automatically. For this reason, clear it by software using the event clear register (BURIF_EVENTCLR.ICICLR) before returning from the interrupt processing.
CHAPTER12: Backup RAM Interface 3.Explanation of Operation 3.4.Required Number of Cycles	Features of "For details" should be corrected as indicated by the shading below. (Error) The above processing is required for each word address (for 16 K). In addition, 5 writes and 4 reads are required for each word address for the partial write function. (Correct) The above processing is required for each word address (for 16 K). In addition, 5 writes and 4 reads are required for a word address for the partial write function.
CHAPTER12: Backup RAM Interface 3.Explanation of Operation 3.5.Bus Error Response	Features of "For details" should be corrected as indicated by the shading below. (Error) 2.Write access to the Backup RAM interface register in the lock state (BURIF_STATUS. LOCKTATUS=1) (Correct) 2.Write access to the Backup RAM interface register in the lock state (BURIF_STATUS.LOCKSTATUS=1)

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER12: Backup RAM Interface 5.Registers</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
Table 5-1 BACKUP-RAM Interface Registers

<table>
<tr><td>BURIF_EFECR</td><td>BURAMIF ECC pseudo-error generation control register</td><td>5.7</td></tr>
<tr><td>BURIF_EPDCR</td><td>BURAMIF ECC data path control register</td><td>5.8</td></tr>
<tr><td>BURIF_ECCTKCCR</td><td>BURAMIF ECC test mode key code control register</td><td>5.9</td></tr>
</table>

(Correct)
Table 5-1 Backup RAM Interface Registers

<table>
<tr><td>BURIF_EFECR</td><td>BURAMIF ECC pseudo-error generation control register</td><td>5.7</td></tr>
<tr><td>BURIF_EDPCR</td><td>BURAMIF ECC data path control register</td><td>5.8</td></tr>
<tr><td>BURIF_ECCTKCCR</td><td>BURAMIF ECC test mode key code control register</td><td>5.9</td></tr>
</table>
</td></tr>
<tr><td>CHAPTER12: Backup RAM Interface 5.Registers</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
Table 5-2 Register Memory Layout

<table>
<tr><td>0x0000_0010</td><td>BURIF_EFECR
00000000_00000000_00000000</td></tr>
<tr><td>0x0000_0013</td><td>BURIF_EPDCR
00000000</td></tr>
<tr><td>0x0000_0014</td><td>BURIF_ECCTKCCR
00000000</td></tr>
</table>

(Correct)
Table 5-2 Register Memory Layout

<table>
<tr><td>0x0000_0010</td><td>BURIF_EFECR
00000000_00000000_00000000</td></tr>
<tr><td>0x0000_0013</td><td>BURIF_EDPCR
00000000</td></tr>
<tr><td>0x0000_0014</td><td>BURIF_ECCTKCCR
00000000</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER12: Backup RAM Interface
5.Registers</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
<table>
<tr><th>Address</th><th>+3</th><th>+2</th><th>+1</th><th>+0</th></tr>
<tr><td>0x0000_0000</td><td colspan="4">BURIF_UNLOCK</td></tr>
<tr><td>0x0000_0004</td><td colspan="4">BURIF_STATUS</td></tr>
<tr><td>0x0000_0008</td><td colspan="2">BURIF_SEEAR</td><td colspan="2">BURIF_DEEAR</td></tr>
<tr><td>0x0000_000C</td><td>BURIF_EECSR</td><td>Reserved</td><td colspan="2">BURIF_EFEAR</td></tr>
<tr><td>0x0000_0010</td><td>BURIF_EPDCR</td><td colspan="3">BURIF_EFECR</td></tr>
</table>

(Correct)
<table>
<tr><th>Address</th><th>+3</th><th>+2</th><th>+1</th><th>+0</th></tr>
<tr><td>0x0000_0000</td><td colspan="4">BURIF_UNLOCK</td></tr>
<tr><td>0x0000_0004</td><td colspan="4">BURIF_STATUS</td></tr>
<tr><td>0x0000_0008</td><td colspan="2">BURIF_SEEAR</td><td colspan="2">BURIF_DEEAR</td></tr>
<tr><td>0x0000_000C</td><td>BURIF_EECSR</td><td>Reserved</td><td colspan="2">BURIF_EFEAR</td></tr>
<tr><td>0x0000_0010</td><td>BURIF_EDPCR</td><td colspan="3">BURIF_EFECR</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER12: Backup RAM Interface
5.Registers
5.2. BURAMIF Status Register (BURIF_STATUS)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit2] TM_ESKPG: ECC generation function disable status bit
<table>
<tr><th>TM_
ESKPG</th><th>ECC Generation Function Disable Status Bit</th></tr>
<tr><td>During Write Operation</td><td>No effect</td></tr>
<tr><td>During Read Operation</td><td>0: Disabling of the test-mode ECC generation function is not allowed (initial value).
Writing "1" to BURIF_EPDCR.SKPG does not allow disabling of the test-mode ECC generation function.
1: Disabling of the test-mode ECC generation function is allowed.
Writing "1" to BURIF_EPDCR.SKPG allows disabling of the test-mode ECC generation function.</td></tr>
</table>
(Correct)
[bit2] TM_ESKPG: ECC Generation Function Disable Status Bit
<table>
<tr><th>TM_ESKPG</th><th>ECC Generation Function Disable Status Bit</th></tr>
<tr><td>During Write Operation</td><td>No effect</td></tr>
<tr><td>During Read Operation</td><td>0: Disabling of the test-mode ECC generation function is not allowed (initial value).
Writing "1" to BURIF_EDPCR.SKPG does not allow disabling of the test-mode ECC generation function.
1: Disabling of the test-mode ECC generation function is allowed.
Writing "1" to BURIF_EDPCR.SKPG allows disabling of the test-mode ECC generation function.</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER12: Backup RAM Interface
5.Registers
5.2. BURAMIF Status Register (BURIF_STATUS)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit1] TM_ESKPC: ECC test function disable status bit
<table>
<tr><th>TM_
ESKPC</th><th>ECC Test Function Disable Status Bit</th></tr>
<tr><td>During Write Operation</td><td>No effect</td></tr>
<tr><td>During Read Operation</td><td>0: Disabling of the test-mode ECC test function is not allowed (initial value).
Writing "1" to BURIF_EPDCR.SKPC does not allow disabling of the test-mode ECC test function.
1: Disabling of the test-mode ECC test function is allowed.
Writing "1" to BURIF_EPDCR.SKPC allows disabling of the test-mode ECC test function.</td></tr>
</table>

(Correct)
[bit1] TM_ESKPC: ECC Test Function Disable Status Bit
<table>
<tr><th>TM_ESKPC</th><th>ECC Test Function Disable Status Bit</th></tr>
<tr><td>During Write Operation</td><td>No effect</td></tr>
<tr><td>During Read Operation</td><td>0: Disabling of the test-mode ECC test function is not allowed (initial value).
Writing "1" to BURIF_EDPCR.SKPC does not allow disabling of the test-mode ECC test function.
1: Disabling of the test-mode ECC test function is allowed.
Writing "1" to BURIF_EDPCR.SKPC allows disabling of the test-mode ECC test function.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER12: Backup RAM Interface
5.Registers
5.2. BURAMIF Status Register (BURIF_STATUS)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit0] TM_EEACC: ECC area access enable status bit
<table><tr><th>TM_
EEACC</th><th>ECC Area Access Enable Status Bit</th></tr><tr><td>During Write Operation</td><td>No effect</td></tr><tr><td>During Read Operation</td><td>0: Access to the test-mode ECC area is not allowed (initial value).
Writing "1" to BURIF_EPDCR.EACC does not enable access to the test-mode ECC area.
1: Access to test-mode ECC area is allowed.
Writing "1" to BURIF_EPDCR.EACC enables access to the test-mode ECC area.</td></tr></table>
(Correct)
[bit0] TM_EEACC: ECC Area Access Enable Status Bit
<table><tr><th>TM_EEACC</th><th>ECC Area Access Enable Status Bit</th></tr><tr><td>During Write Operation</td><td>No effect</td></tr><tr><td>During Read Operation</td><td>0: Access to the test-mode ECC area is not allowed (initial value).
Writing "1" to BURIF_EDPCR.EACC does not enable access to the test-mode ECC area.
1: Access to test-mode ECC area is allowed.
Writing "1" to BURIF_EDPCR.EACC enables access to the test-mode ECC area.</td></tr></table></td></tr>
<tr><td>CHAPTER12: Backup RAM Interface
5.Registers
5.5.BURAMIF ECC Pseudo-error Generation Address Register (BURIF_EFEAR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
The ECC pseudo-error generation address register specifies the address that generates a pseudo-error for Backup RAM. Writing to this register is possible only when privileged access is available and BURIF_STATUS.LOCKSTATUS is "0".

(Correct)
The ECC pseudo-error generation address register specifies the address that generates a pseudo-error for Backup RAM. Writing to this register is possible only when privileged access is available and BURIF_STATUS.LOCKSTATUS is "0".</td></tr>
</table>

Section	Change Results
CHAPTER12: Backup RAM Interface 5.Registers 5.5.BURAMIF ECC Pseudo-error Generation Address Register (BURIF_EFEAR)	Features of "For details" should be corrected as indicated by the shading below. (Error) [bit14:0] ERR_ADDR: Single-bit error occurrence address bits (Correct) [bit14:0] ERR_ADDR: Pseudo-Single-Bit Error Occurrence Address Setting Bits
CHAPTER12: Backup RAM Interface 5.Registers 5.6.BURAMIF ECC Error Control Register (BURIF_EECSR)	Features of "For details" should be corrected as indicated by the shading below. (Error) This register indicates whether 1-bit error correction or 2-bit error detection has occurred during an ECC test for Backup RAM. This register specifies whether interrupts by 2-bit error detection are enabled. Writing to this register is possible only when privileged access is available and BURIF_STATUS:LOCKSTATUS is "0". (Correct) This register indicates whether 1-bit error correction or 2-bit error detection has occurred during an ECC test for Backup RAM. This register specifies whether interrupts by 2-bit error detection are enabled. Writing to this register is possible only when privileged access is available and BURIF_STATUS.LOCKSTATUS is "0".
CHAPTER12: Backup RAM Interface 5.Registers 5.7.BURAMIF ECC Pseudo-error Generation Control Register (BURIF_EFECR)	Features of "For details" should be corrected as indicated by the shading below. (Error) The ECC pseudo-error generation control register (EFECR) specifies the content of the Backup RAM pseudo-error to be generated in the form of a generation byte and generation bit. Writing to this register is possible only when privileged access is available and BURIF_STATUS:LOCKSTATUS is "0". (Correct) The ECC pseudo-error generation control register (EFECR) specifies the content of the Backup RAM pseudo-error to be generated in the form of a generation byte and generation bit. Writing to this register is possible only when privileged access is available and BURIF_STATUS.LOCKSTATUS is "0".

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER12: Backup RAM Interface
5.Registers
5.7.BURAMIF ECC Pseudo-error Generation Control Register (BURIF_EFECR)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit16] FERR: Pseudo-error generation enable bit
This bit enables the generation of pseudo-ECC-errors
<table>
<tr><th>FERR</th><th>Pseudo-error Generation Enable Bit</th></tr>
<tr><td>During Write Operation</td><td>0: Disable generation of pseudo-ECC-errors.
1: Enable generation of pseudo-ECC-errors.</td></tr>
<tr><td>During Read Operation</td><td>The value set in this bit is read.</td></tr>
</table>

(Correct)
[bit16] FERR: Pseudo-Error Generation Enable Bit
This bit enables the generation of pseudo-ECC-errors.
<table>
<tr><th>FERR</th><th>Pseudo-error Generation Enable Bit</th></tr>
<tr><td>During Write Operation</td><td>0: No effect. When generation of pseudo-ECC-errors is completed, this bit is automatically cleared to "0" by the hardware.
1: Enable generation of pseudo-ECC-errors.</td></tr>
<tr><td>During Read Operation</td><td>0: Generation of pseudo-ECC-errors is the disabled status. (Normal operation)
1: Generation of pseudo-ECC-errors is the enabled status.</td></tr>
</table>
</td></tr>
<tr><td>CHAPTER12: Backup RAM Interface
5.Registers
5.7.BURAMIF ECC Pseudo-error Generation Control Register (BURIF_EFECR)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit7:0] EI7 to EI0: Pseudo-error generation bit setting bits
These bits specify the bit position of the pseudo-ECC-error generation in Backup RAM.
<table>
<tr><th>EI7 to EI0</th><th>Target Bytes in RAM</th></tr>
</table>

(Correct)
[bit7:0] EI7 to EI0: Pseudo-Error Generation Bit Setting Bits
These bits specify the bit position of the pseudo-ECC-error generation in Backup RAM.
<table>
<tr><th>EI7 to EI0</th><th>Target bits in RAM</th></tr>
</table>
</td></tr>
</table>

Section	Change Results
CHAPTER12: Backup RAM Interface 5.Registers 5.8.BURAMIF ECC Data Path Control Register (BURIF_EDPCR)	Features of "For details" should be corrected as indicated by the shading below. (Error) 5.8.BURAMIF ECC Data Path Control Register (BURIF_EPDCR) The ECC data path control register controls disabling of the ECC generation function, disabling of the ECC test function, and enabling of access to the ECC area. Writing to this register is possible only when privileged access is available and BURIF_STATUS:LOCKSTATUS is "0". (Correct) 5.8.BURAMIF ECC Data Path Control Register (BURIF_EDPCR) The ECC data path control register controls disabling of the ECC generation function, disabling of the ECC test function, and enabling of access to the ECC area. Writing to this register is possible only when privileged access is available and BURIF_STATUS.LOCKSTATUS is "0".
CHAPTER12: Backup RAM Interface 5.Registers 5.9. BURAMIF ECC Test Mode Key Code Control Register (BURIF_ECCTKCCR)	Features of "For details" should be corrected as indicated by the shading below. (Error) The ECC test mode key code control register is used to enable ECC pseudo-error generation for Backup RAM, access to the ECC area, and disabling of the ECC generation test function. Writing to this register is possible only when privileged access is available and BURIF_STATUS:LOCKSTATUS is "0". (Correct) The ECC test mode key code control register is used to enable ECC pseudo-error generation for Backup RAM, access to the ECC area, and disabling of the ECC generation test function. Writing to this register is possible only when privileged access is available and BURIF_STATUS.LOCKSTATUS is "0".

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER12: Backup RAM Interface
5.Registers
5.9. BURAMIF ECC Test Mode Key Code Control Register (BURIF_ECCTKCCR)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit2] TM_ESKPG: ECC generation function skip enable bit
<table>
<tr><th>TM_
ESKPG</th><th>ECC Generation Function Skip Enable Bit</th></tr>
<tr><td>During Write Operation</td><td>0: Enable the ECC generation function, regardless of the BURIF_EPDCR.SKPG value. Moreover, writing to BURIF_EPDCR.SKPG by software is disabled.
1: Enable skipping of the ECC generation function. Moreover, BURIF_EDPCR.SKPG="1" enables skipping of the ECC generation function.</td></tr>
<tr><td>During Read Operation</td><td>"0" is always read.</td></tr>
</table>
[bit1] TM_ESKPC: ECC test function skip enable bit
<table>
<tr><th>TM_
ESKPC</th><th>ECC Test Function Skip Enable Bit</th></tr>
<tr><td>During Write Operation</td><td>0: Disable skipping of the ECC test function, regardless of the BURIF_EDPCR.SKPC value. Moreover, writing to BURIF_EDPCR.SKPC by software is disabled.
1: Enable skipping of the ECC test function. Moreover, BURIF_EDPCR.SKPC="1" enables skipping of the ECC test function.</td></tr>
<tr><td>During Read Operation</td><td>"0" is always read.</td></tr>
</table>
(Correct)
[bit2] TM_ESKPG: ECC Generation Function Skip Enable Bit
<table>
<tr><th>TM_ESKPG</th><th>ECC Generation Function Skip Enable Bit</th></tr>
<tr><td>During Write Operation</td><td>0: Enable the ECC generation function, regardless of the BURIF_EDPCR.SKPG value. Moreover, writing to BURIF_EDPCR.SKPG by software is disabled.
1: Enable skipping of the ECC generation function. Moreover, BURIF_EDPCR.SKPG= 1 enables skipping of the ECC generation function.</td></tr>
<tr><td>During Read Operation</td><td>"0" is always read.</td></tr>
</table>
[bit1] TM_ESKPC: ECC Test Function Skip Enable Bit
<table>
<tr><th>TM_ESKPC</th><th>ECC Test Function Skip Enable Bit</th></tr>
<tr><td>During Write Operation</td><td>0: Disable skipping of the ECC test function, regardless of the BURIF_EDPCR.SKPC value. Moreover, writing to BURIF_EDPCR.SKPC by software is disabled.
1: Enable skipping of the ECC test function. Moreover, BURIF_EDPCR.SKPC= 1 enables skipping of the ECC test function.</td></tr>
<tr><td>During Read Operation</td><td>"0" is always read.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER12: Backup RAM Interface
5.Registers
5.9. BURAMIF ECC Test Mode Key Code Control Register (BURIF_ECCTKCCR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
Note:
−Set the same values in the CODE bits while KEY bit operation is being performed. For details on the procedure for setting these bits, see "Figure 4 2 Setting Flow of ECC Test Mode Key Code Control Register"

(Correct)
Note:
−Set the same values in the TM_ESKPG, TM_ESKPGC and TM_EEACC bits while KEY bit operation is being performed. For details on the procedure for setting these bits, see "Figure 4 2 Setting Flow of ECC Test Mode Key Code Control Register"</td></tr>
<tr><td>CHAPTER12: Backup RAM Interface
5.Registers
5.18. BURAMIF TEST Pseudo-error Generation Control Register (BURIF_TFECR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit2:0] ETYP: Pseudo-error generation processing specification
<table><tr><th>ETYP</th><th>Pseudo-error Generation Processing Specification Bit</th></tr><tr><td>During Write Operation</td><td>0bxx1: Pseudo-errors are generated during march diagnosis.
0bx1x: Pseudo-errors are generated during checker diagnosis.
0b1xx: Pseudo-errors are generated during unique diagnosis.
0b000: Generates no pseudo error.</td></tr><tr><td>During Read Operation</td><td>This register can read the setting value for the diagnosis in which the pseudo-error is generated.</td></tr></table>
(Correct)
[bit2:0] ETYP: Pseudo-Error Generation Processing Specification
<table><tr><th>ETYP</th><th>Pseudo-error Generation Processing Specification Bit</th></tr><tr><td>During Write Operation</td><td>0bxx1: Pseudo-errors are generated during march diagnosis.
0bx1x: Pseudo-errors are generated during checker diagnosis.
0b1xx: Pseudo-errors are generated during unique diagnosis.
0b000: Generates no pseudo error.</td></tr><tr><td>During Read Operation</td><td>This bit can read the setting value for the diagnosis in which the pseudo-error is generated.</td></tr></table></td></tr>
<tr><td>CHAPTER12: Backup RAM Interface
5.Registers
5.20. BURAMIF Event Clear Register (BURIF_EVENTCLR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit6] TEICLR: Error occurrence during diagnosis

(Correct)
[bit6] TEICLR: Error Occurrence during Diagnosis Clear Bit</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER12: Backup RAM Interface
5.Registers
5.20. BURAMIF Event Clear Register (BURIF_EVENTCLR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit4] TCICLR: Diagnosis end

(Correct)
[bit4] TCICLR: Diagnosis End Clear Bit</td></tr>
<tr><td>CHAPTER13: External Interrupt
3.Explanation of Operation</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
External interrupt request level
If edge detection is specified as an event input and the noise filter is enabled, a rule applies to the pulse width of an input signal to be recognized as an input edge. For the minimum value of the pulse width, see the XX design sheet.

(Correct)
External Interrupt Request Level
If edge detection is specified as an event input and the noise filter is enabled, a rule applies to the pulse width of an input signal to be recognized as an input edge. For the minimum value of the pulse width, see the PF-Cluster design sheet.</td></tr>
<tr><td>CHAPTER13: External Interrupt
5.Registers
5.2.External Interrupt Enable Set Register (EICxx_ENISR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit31:0] ENS31 to ENS0: External interrupt enable set bits
These bits configure the set control of the EICxx_ENIR register.
<table><tr><th>Bit</th><th>Description</th></tr></table>
(Correct)
[bit31:0] ENS31 to ENS0: External Interrupt Enable Set Bits
These bits configure the set control of the EICxx_ENIR register.
<table><tr><th>ENS31 to ENS0</th><th>Description</th></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER13: External Interrupt 5.Registers 5.3. External Interrupt Enable Clear Register (EICxx_ENICR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit31:0] ENC31 to ENC0: External interrupt enable clear bits
These bits configure the clear control of the EICxx_ENIR register.
<table><tr><th>Bit</th><th>Description</th></tr></table>
(Correct)
[bit31:0] ENC31 to ENC0: External Interrupt Enable Clear Bits
These bits configure the clear control of the EICxx_ENIR register.
<table><tr><th>ENC31 to ENC0</th><th>Description</th></tr></table></td></tr>
<tr><td>CHAPTER13: External Interrupt 5.Registers 5.4. External Interrupt Factor Register (EICxx_EIRR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit31:0] ER31 to ER0: External interrupt factor detection bits
These bits retain the detection of an external interrupt request.
<table><tr><th>Bit</th><th>Description</th></tr></table>
(Correct)
[bit31:0] ER31 to ER0: External Interrupt Factor Detection Bits
These bits retain the detection of an external interrupt request.
<table><tr><th>ER31 to ER0</th><th>Description</th></tr></table></td></tr>
<tr><td>CHAPTER13: External Interrupt 5.Registers 5.5. External Interrupt Factor Clear Register (EICxx_EIRCR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit31:0] ERC31 to ERC0: External interrupt factor clear bits
These bits configure the clear control of the EICxx_EIRR register.
<table><tr><th>Bit</th><th>Description</th></tr></table>
(Correct)
[bit31:0] ERC31 to ERC0: External Interrupt Factor Clear Bits
These bits configure the clear control of the EICxx_EIRR register.
<table><tr><th>ERC31 to ERC0</th><th>Description</th></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER13: External Interrupt 5.Registers 5.6. Noise Filter Enable Register (EICxx_NFER)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit31:0] NFE31 to NFE0: Noise filter enable bits
These bits configure the noise filter control of each external interrupt.
<table><tr><th>Bit</th><th>Description</th></tr></table>
(Correct)
[bit31:0] NFE31 to NFE0: Noise Filter Enable Bits
These bits configure the noise filter control of each external interrupt.
<table><tr><th>NFE31 to NFE0</th><th>Description</th></tr></table></td></tr>
<tr><td>CHAPTER13: External Interrupt 5.Registers 5.7. Noise Filter Enable Set Register (EICxx_NFESR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit31:0] NFES31 to NFES0: Noise filter enable set bits
These bits configure the set control of the EICxx_NFER register.
<table><tr><th>Bit</th><th>Description</th></tr></table>
(Correct)
[bit31:0] NFES31 to NFES0: Noise Filter Enable Set Bits
These bits configure the set control of the EICxx_NFER register.
<table><tr><th>NFES31 to NFES0</th><th>Description</th></tr></table></td></tr>
<tr><td>CHAPTER13: External Interrupt 5.Registers 5.8. Noise Filter Enable Clear Register (EICxx_NFECR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit31:0] NFE31 to NFE0: Noise filter enable bits
These bits configure the noise filter control of each external interrupt.
<table><tr><th>Bit</th><th>Description</th></tr></table>
(Correct)
[bit31:0] NFEC31 to NFEC0: Noise Filter Enable Clear Bits
These bits configure the clear control of the EICxx_NFER register.
<table><tr><th>NFEC31 to NFEC0</th><th>Description</th></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER13: External Interrupt 5.Registers 5.10.Non-maskable Interrupt Register (EICxx_NMIR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit8] NMICLR: Non maskable interrupt clear bit
This bit configures the clear control of non maskable interrupts.
<table><tr><th>Bit</th><th>Description</th></tr></table>
(Correct)
[bit8] NMICLR: Non-Maskable Interrupt Clear Bit
This bit configures the clear control of non-maskable interrupts.
<table><tr><th>NMICLR</th><th>Description</th></tr></table></td></tr>
<tr><td>CHAPTER13: External Interrupt 5.Registers 5.12. DMA Request Enable Set Register (EICxx_DRESR)</td><td>(Error)
[bit31:0] DRES31 to DRES0: DMA request enable set bits
These bits configure the set control of the EICxx_DRER register.
<table><tr><th>Bit</th><th>Description</th></tr></table>
(Correct)
[bit31:0] DRES31 to DRES0: DMA Request Enable Set Bits
These bits configure the set control of the EICxx_DRER register.
<table><tr><th>DRES31 to DRES0</th><th>Description</th></tr></table></td></tr>
<tr><td>CHAPTER13: External Interrupt 5.Registers 5.13. DMA Request Enable Clear Register (EICxx_DRECR)</td><td>(Error)
[bit31:0] DREC31 to DREC0: DMA request enable clear bits
These bits configure the clear control of the EICxx_DRER register.
<table><tr><th>Bit</th><th>Description</th></tr></table>
(Correct)
[bit31:0] DREC31 to DREC0: DMA Request Enable Clear Bits
These bits configure the clear control of the EICxx_DRER register.
<table><tr><th>DREC31 to DREC0</th><th>Description</th></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER13: External Interrupt 5.Registers 5.14. DMA Request Flag Register (EICxx_DRFR)</td><td>(Error)
[bit31:0] DRF31 to DRF0: DMA request detection bits
These bits retain the detection of a DMA request.
<table><tr><th>Bit</th><th>Description</th></tr></table>
(Correct)
[bit31:0] DRF31 to DRF0: DMA Request Detection Bits
These bits retain the detection of a DMA request.
<table><tr><th>DRF31 to DRF0</th><th>Description</th></tr></table></td></tr>
<tr><td>CHAPTER13: External Interrupt 5.Registers 5.14. DMA Request Flag Register (EICxx_DRFR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
Note:

(Correct)
Notes:</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER14: Hardware Watchdog Timer
6. Register List
6.1.Hardware Watchdog Protection Register (HWDG_PROT)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit31:0] KEY[31:0]: Protection bits
<table>
<tr><th>bit31:0</th><th>Description</th></tr>
<tr><td>During write operation</td><td>When "0xEDAC_CE55" is written:
Protection lock for register write is released.
When a value other than "0xEDAC_CE55" is written:
Protection lock for register write is not released.</td></tr>
<tr><td>During read operation</td><td>When "0xFFFF_FFFF" is read:
Protection lock for register write has been released.
When "0x0000_0000" is read:
Protection lock for register write has been enabled.</td></tr>
</table>
(Correct)
[bit31:0] KEY[31:0]: Protection Bits
<table>
<tr><th>KEY[31:0]</th><th>Description</th></tr>
<tr><td>During write operation</td><td>When 0xEDAC_CE55 is written:
Protection lock for register write is released.
When a value other than 0xEDAC_CE55 is written:
Protection lock for register write is not released.</td></tr>
<tr><td>During read operation</td><td>When 0xFFFF_FFFF is read:
Protection lock for register write has been released.
When 0x0000_0000 is read:
Protection lock for register write has been enabled.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER14: Hardware Watchdog Timer
6. Register List
6.2. Hardware Watchdog Counter Register (HWDG_CNT)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit31:0] WDGCNT[31:0]: Watchdog counter bits
<table><tr><th>bit31:0</th><th>Description</th></tr><tr><td>During write operation</td><td>Invalid</td></tr><tr><td>During read operation</td><td>Returns the current watchdog counter value.
The watchdog counter value is sampled by using the bus clock.</td></tr></table>
(Correct)
[bit31:0] WDGCNT[31:0]: Watchdog Counter Bits
<table><tr><th>WDGCNT[31:0]</th><th>Description</th></tr><tr><td>During write operation</td><td>Invalid</td></tr><tr><td>During read operation</td><td>Returns the current watchdog counter value.
The watchdog counter value is sampled by using the bus clock.</td></tr></table></td></tr>
<tr><td>CHAPTER14: Hardware Watchdog Timer
6. Register List
6.3.Hardware Watchdog Reset Factor Register (HWDG_RSTCAUSE)</td><td>The item should be added as indicated by the shading below.

(Error)
This register is not initialized by reset. To check the factor of the watchdog reset request, read this register.

(Correct)
This register is not initialized by reset. After a watch dock reset, the value is maintained. To check the factor of the watchdog reset request, read this register.</td></tr>
<tr><td>CHAPTER14: Hardware Watchdog Timer
6. Register List
6.3.Hardware Watchdog Reset Factor Register (HWDG_RSTCAUSE)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit4] RSTCAUSE4: Reset factor bit 4
If the watchdog counter clear protection trigger sequence is executed when the LOCK bit in the Hardware watchdog configuration register (HWDG_CFG) is "0", this bit is set to "1".
<table><tr><th>bit</th><th>Description</th></tr></table>
(Correct)
[bit4] RSTCAUSE4: Reset Factor Bit 4
If the watchdog counter clear protection trigger sequence is executed when the LOCK bit in the Hardware watchdog configuration register (HWDG_CFG) is "0", this bit is set to "1".
<table><tr><th>RSTCAUSE4</th><th>Description</th></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER14: Hardware Watchdog Timer
6. Register List
6.3.Hardware Watchdog Reset Factor Register (HWDG_RSTCAUSE)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit3] RSTCAUSE3: Reset factor bit 3
If the watchdog counter clear protection trigger sequence is executed before the watchdog counter reaches the window lower limit value, this bit is set to "1".
<table><tr><th>bit</th><th>Description</th></tr></table>
(Correct)
[bit3] RSTCAUSE3: Reset Factor Bit 3
If the watchdog counter clear protection trigger sequence is executed before the watchdog counter reaches the window lower limit value, this bit is set to "1".
<table><tr><th>RSTCAUSE3</th><th>Description</th></tr></table></td></tr>
<tr><td>CHAPTER14: Hardware Watchdog Timer
6. Register List
6.3.Hardware Watchdog Reset Factor Register (HWDG_RSTCAUSE)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit2] RSTCAUSE2: Reset factor bit 2
When the watchdog counter reaches the window upper limit value, this bit is set to "1".
<table><tr><th>bit</th><th>Description</th></tr></table>
(Correct)
[bit2] RSTCAUSE2: Reset Factor Bit 2
When the watchdog counter reaches the window upper limit value, this bit is set to "1".
<table><tr><th>RSTCAUSE2</th><th>Description</th></tr></table></td></tr>
<tr><td>CHAPTER14: Hardware Watchdog Timer
6. Register List
6.3.Hardware Watchdog Reset Factor Register (HWDG_RSTCAUSE)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit1] RSTCAUSE1: Reset factor bit 1
When there is a violation of the watchdog counter clear protection trigger sequence, this bit is set to "1". For details, see Figure 3 2.
<table><tr><th>bit</th><th>Description</th></tr></table>
(Correct)
[bit1] RSTCAUSE1: Reset Factor Bit 1
When there is a violation of the watchdog counter clear protection trigger sequence, this bit is set to "1". For details, see Figure 3 2.
<table><tr><th>RSTCAUSE1</th><th>Description</th></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER14: Hardware Watchdog Timer
6. Register List
6.3.Hardware Watchdog Reset Factor Register (HWDG_RSTCAUSE)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit0] RSTCAUSE0: Reset factor bit 0
If the value written in the Hardware watchdog trigger 0/1 register (HWDG_TRG0 or HWDG_TRG1) does not match a proper value, this bit is set to "1".
<table>
<tr><th>bit</th><th>Description</th></tr>
</table>
(Correct)
[bit0] RSTCAUSE0: Reset Factor Bit 0
If the value written in the Hardware watchdog trigger 0/1 register (HWDG_TRG0 or HWDG_TRG1) does not match a proper value, this bit is set to "1".
<table>
<tr><th>RSTCAUSE0</th><th>Description</th></tr>
</table>
</td></tr>
<tr><td>CHAPTER14: Hardware Watchdog Timer
6. Register List
6.4. Hardware Watchdog Trigger 0 Register (HWDG_TRG0)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit7:0] WDGTRG0[7:0]: Watchdog trigger 0 bits
These bits are used to execute the watchdog counter clear protection trigger sequence to clear the watchdog counter.
<table>
<tr><th>bit7:0</th><th>Description</th></tr>
<tr><td>During write operation</td><td>When the HWDG_TRG0CFG value is written:
1 condition for executing the watchdog counter clear protection trigger sequence is met.
When a value other than the HWDG_TRG0CFG value is written:
A watchdog error is generated.</td></tr>
<tr><td>During read operation</td><td>Reads "0b00000000".</td></tr>
</table>
(Correct)
[bit7:0] WDGTRG0[7:0]: Watchdog Trigger 0 Bits
These bits are used to execute the watchdog counter clear protection trigger sequence to clear the watchdog counter.
<table>
<tr><th>WDGTRG0[7:0]</th><th>Description</th></tr>
<tr><td>During write operation</td><td>When the HWDG_TRG0CFG value is written:
1 condition for executing the watchdog counter clear protection trigger sequence is met.
When a value other than the HWDG_TRG0CFG value is written:
A watchdog error is generated.</td></tr>
<tr><td>During read operation</td><td>Reads 0b00000000.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER14: Hardware Watchdog Timer
6. Register List
6.5. Hardware Watchdog Trigger 1 Register (HWDG_TRG1)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit7:0] WDGTRG1[7:0]: Watchdog trigger 1 bits
These bits are used to execute the watchdog counter clear protection trigger sequence to clear the watchdog counter.
<table>
<tr><th>bit7:0</th><th>Description</th></tr>
<tr><td>During write operation</td><td>When the HWDG_TRG1CFG value is written:
1 condition for executing the watchdog counter clear protection trigger sequence is met.
When a value other than the HWDG_TRG1CFG value is written:
A watchdog error is generated.</td></tr>
<tr><td>During read operation</td><td>Reads "0b00000000".</td></tr>
</table>

(Correct)
[bit7:0] WDGTRG1[7:0]: Watchdog Trigger 1 Bits
These bits are used to execute the watchdog counter clear protection trigger sequence to clear the watchdog counter.
<table>
<tr><th>WDGTRG1[7:0]</th><th>Description</th></tr>
<tr><td>During write operation</td><td>When the HWDG_TRG1CFG value is written:
1 condition for executing the watchdog counter clear protection trigger sequence is met.
When a value other than the HWDG_TRG1CFG value is written:
A watchdog error is generated.</td></tr>
<tr><td>During read operation</td><td>Reads 0b00000000.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER14: Hardware Watchdog Timer
6. Register List
6.6. Hardware Watchdog Interrupt Configuration Register (HWDG_INT)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit17] RSTEN: Reset/NMI enable bit
This bit is used to generate either the watchdog reset request or watchdog interrupt request (NMI) in response to a watchdog error. Prohibiting setting this bit to "0" except for the purpose of using it as a test function. (If this bit is set to "0" during application execution, security is compromised.) This bit is equipped with a majority circuit with 3 FFs as a measure against a bit invert caused by the effects of noise or another factor.
<table>
<tr><th>bit</th><th>Description</th></tr>
<tr><td>During write operation</td><td>When "0" is written: The watchdog interrupt request (NMI) is generated.
When "1" is written: The watchdog interrupt request is generated.</td></tr>
<tr><td>During read operation</td><td>Set value read</td></tr>
</table>
[bit16] IRQEN: Prior warning interrupt enable bit
This bit is used to enable generation of the prior warning interrupt request.
<table>
<tr><th>bit</th><th>Description</th></tr>
<tr><td>During write operation</td><td>When "0" is written: The prior warning interrupt request is not generated.
When "1" is written: The prior warning interrupt request is generated.</td></tr>
<tr><td>During read operation</td><td>Set value read</td></tr>
</table>

(Correct)
[bit17] RSTEN: Reset/NMI Enable Bit
This bit is used to generate either the watchdog reset request or watchdog interrupt request (NMI) in response to a watchdog error. Prohibiting setting this bit to "0" except for the purpose of using it as a test function. (If this bit is set to "0" during application execution, security is compromised.) This bit is equipped with a majority circuit with 3 FFs as a measure against a bit invert caused by the effects of noise or another factor.
<table>
<tr><th>RSTEN</th><th>Description</th></tr>
<tr><td>During write operation</td><td>When "0" is written: The watchdog interrupt request (NMI) is generated.
When "1" is written: The watchdog interrupt request is generated.</td></tr>
<tr><td>During read operation</td><td>Set value read</td></tr>
</table>
[bit16] IRQEN: Prior warning Interrupt Enable Bit
This bit is used to enable generation of the prior warning interrupt request.
<table>
<tr><th>IRQEN</th><th>Description</th></tr>
<tr><td>During write operation</td><td>When "0" is written: The prior warning interrupt request is not generated.
When "1" is written: The prior warning interrupt request is generated.</td></tr>
<tr><td>During read operation</td><td>Set value read</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER14: Hardware Watchdog Timer
6. Register List
6.6. Hardware Watchdog Interrupt Configuration Register (HWDG_INT)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit1] NMIFLAG: NMI flag
This bit is set by a watchdog error when the RSTEN bit in the Hardware watchdog interrupt configuration register (HWDG_INT) is "0". When the RSTEN bit in the Hardware watchdog interrupt clear register (HWDG_INT) is "1", the watchdog reset request is generated. You can clear this bit by writing "1" to the NMICLR bit in the Hardware watchdog interrupt configuration register (HWDG_INTCLR).
<table>
<tr><th>bit</th><th>Description</th></tr>
<tr><td>During write operation</td><td>Invalid</td></tr>
<tr><td>During read operation</td><td>When "0" is read: No detection of a watchdog error (NMI) is indicated.
When "1" is read: Detection of a watchdog error (NMI) is indicated.</td></tr>
</table>
[bit0] IRQFLAG: IRQ flag
This bit is set by a watchdog error. The prior warning interrupt is generated when the IRQEN bit in the Hardware watchdog interrupt clear register (HWDG_INT) is "1". You can clear this bit by writing "1" to the IRQCLR bit in the Hardware watchdog interrupt configuration register (HWDG_INTCLR).
<table>
<tr><th>bit</th><th>Description</th></tr>
<tr><td>During write operation</td><td>Invalid</td></tr>
<tr><td>During read operation</td><td>When "0" is read: No detection of a watchdog error (IRQ) is indicated.
When "1" is read: Detection of a watchdog error (IRQ) is indicated.</td></tr>
</table>

(Correct)
[bit1] NMIFLAG: NMI Flag
This bit is set by a watchdog error when the RSTEN bit in the Hardware watchdog interrupt configuration register (HWDG_INT) is "0". When the RSTEN bit in the Hardware watchdog interrupt clear register (HWDG_INT) is "1", the watchdog reset request is generated. You can clear this bit by writing "1" to the NMICLR bit in the Hardware watchdog interrupt configuration register (HWDG_INTCLR).
<table>
<tr><th>NMIFLAG</th><th>Description</th></tr>
<tr><td>During write operation</td><td>Invalid</td></tr>
<tr><td>During read operation</td><td>When "0" is read: No detection of a watchdog error (NMI) is indicated.
When "1" is read: Detection of a watchdog error (NMI) is indicated.</td></tr>
</table>
[bit0] IRQFLAG: IRQ Flag
This bit is set by a watchdog error. The prior warning interrupt is generated when the IRQEN bit in the Hardware watchdog interrupt clear register (HWDG_INT) is "1". You can clear this bit by writing "1" to the IRQCLR bit in the Hardware watchdog interrupt configuration register (HWDG_INTCLR).
<table>
<tr><th>IRQFLAG</th><th>Description</th></tr>
<tr><td>During write operation</td><td>Invalid</td></tr>
<tr><td>During read operation</td><td>When "0" is read: No detection of a watchdog error (IRQ) is indicated.
When "1" is read: Detection of a watchdog error (IRQ) is indicated.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER14: Hardware Watchdog Timer
6. Register List
6.7. Hardware Watchdog Interrupt Clear Register (HWDG_INTCLR)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit1] NMICLR: NMI clear bit
This bit is used to clear the NMIFLAG bit in the Hardware watchdog interrupt configuration register (HWDG_INT).
<table>
<tr><th>bit</th><th>Description</th></tr>
<tr><td>During write operation</td><td>When "0" is written: Invalid
When "1" is written: The NMI flag is cleared.</td></tr>
<tr><td>During read operation</td><td>"0" is read.</td></tr>
</table>
You must clear the NMI after waiting for watchdog counter clear (0x0000_0000). Check the watchdog counter value with HWDG_CNT.

[bit0] IRQCLR: Prior warning interrupt clear bit
This bit is used to clear the IRQFLAG bit in the Hardware watchdog interrupt configuration register (HWDG_INT).
<table>
<tr><th>bit</th><th>Description</th></tr>
<tr><td>During write operation</td><td>When "0" is written: Invalid
When "1" is written: The IRQ flag is cleared.</td></tr>
<tr><td>During read operation</td><td>"0" is read.</td></tr>
</table>

(Correct)
[bit1] NMICLR: NMI Clear Bit
This bit is used to clear the NMIFLAG bit in the Hardware watchdog interrupt configuration register (HWDG_INT).
<table>
<tr><th>NMICLR</th><th>Description</th></tr>
<tr><td>During write operation</td><td>When "0" is written: Invalid
When "1" is written: The NMI flag is cleared.</td></tr>
<tr><td>During read operation</td><td>"0" is read.</td></tr>
</table>
You must clear the NMI after waiting for watchdog counter clear (0x0000_0000). Check the watchdog counter value with HWDG_CNT.

[bit0] IRQCLR: Prior warning Interrupt Clear Bit
This bit is used to clear the IRQFLAG bit in the Hardware watchdog interrupt configuration register (HWDG_INT).
<table>
<tr><th>IRQCLR</th><th>Description</th></tr>
<tr><td>During write operation</td><td>When "0" is written: Invalid
When "1" is written: The IRQ flag is cleared.</td></tr>
<tr><td>During read operation</td><td>"0" is read.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER14: Hardware Watchdog Timer
6. Register List
6.8. Hardware Watchdog Trigger 0 Configuration Register (HWDG_TRG0CFG)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit7:0] WDGTRG0CFG[7:0]: Watchdog trigger 0 configuration bits
These bits are used to define a value to be written to the Hardware watchdog trigger 0 register (HWDG_TRG0) for execution of the watchdog counter clear protection trigger sequence.
<table>
<tr><th>bit7:0</th><th>Description</th></tr>
<tr><td>During write operation</td><td>Set value written</td></tr>
<tr><td>During read operation</td><td>Set value read</td></tr>
</table>
(Correct)
[bit7:0] WDGTRG0CFG[7:0]: Watchdog Trigger 0 Configuration Bits
These bits are used to define a value to be written to the Hardware watchdog trigger 0 register (HWDG_TRG0) for execution of the watchdog counter clear protection trigger sequence.
<table>
<tr><th>WDGTRG0CFG[7:0]</th><th>Description</th></tr>
<tr><td>During write operation</td><td>Set value written</td></tr>
<tr><td>During read operation</td><td>Set value read</td></tr>
</table></td></tr>
<tr><td>CHAPTER14: Hardware Watchdog Timer
6. Register List
6.9. Hardware Watchdog Trigger 1 Configuration Register (HWDG_TRG1CFG)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit7:0] WDGTRG1CFG[7:0]: Watchdog trigger 1 configuration bits
These bits are used to define a value to be written to the Hardware watchdog trigger 1 register (HWDG_TRG1) for execution of the watchdog counter clear protection trigger sequence.
<table>
<tr><th>bit7:0</th><th>Description</th></tr>
<tr><td>During write operation</td><td>Set value written</td></tr>
<tr><td>During read operation</td><td>Set value read</td></tr>
</table>
(Correct)
[bit7:0] WDGTRG1CFG[7:0]: Watchdog Trigger 1 Configuration Bits
These bits are used to define a value to be written to the Hardware watchdog trigger 1 register (HWDG_TRG1) for execution of the watchdog counter clear protection trigger sequence.
<table>
<tr><th>WDGTRG1CFG[7:0]</th><th>Description</th></tr>
<tr><td>During write operation</td><td>Set value written</td></tr>
<tr><td>During read operation</td><td>Set value read</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER14: Hardware Watchdog Timer
6. Register List
6.10. Hardware Watchdog Lower Limit RUN Setting Register (HWDG_RUNLLS)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit31:0] WDGRUNLLS[31:0]: Window lower limit value for RUN set bits
These bits are used to define the window lower limit value for RUN. The reading of these bits returns the set value regardless of the Hardware watchdog configuration register (HWDG_CFG) LOCK bit value.
<table>
<tr><th>bit31:0</th><th>Description</th></tr>
<tr><td>During write operation</td><td>When "0" is written to all 32 bits: The window function is disabled.
When any value except "0" is written: The window function is enabled.</td></tr>
<tr><td>During read operation</td><td>Set value read</td></tr>
</table>

(Correct)
[bit31:0] WDGRUNLLS[31:0]: Window Lower Limit Value for RUN Set Bits
These bits are used to define the window lower limit value for RUN. The reading of these bits returns the set value regardless of the Hardware watchdog configuration register (HWDG_CFG) LOCK bit value.
<table>
<tr><th>WDGRUNLLS[31:0]</th><th>Description</th></tr>
<tr><td>During write operation</td><td>When "0" is written to all 32 bits: The window function is disabled.
When any value except "0" is written: The window function is enabled.</td></tr>
<tr><td>During read operation</td><td>Set value read</td></tr>
</table>
</td></tr>
<tr><td>CHAPTER14: Hardware Watchdog Timer
6. Register List
6.11. Hardware Watchdog Upper Limit RUN Setting Register (HWDG_RUNULS)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit31:0] WDGRUNULS[31:0]: Window upper limit value for RUN set bits
These bits are used to define the window upper limit value for RUN. The reading of these bits returns the set value regardless of the Hardware watchdog configuration register (HWDG_CFG) LOCK bit value.
<table>
<tr><th>bit31:0</th><th>Description</th></tr>
<tr><td>During write operation</td><td>Set value written</td></tr>
<tr><td>During read operation</td><td>Set value read</td></tr>
</table>

(Correct)
[bit31:0] WDGRUNULS[31:0]: Window Upper Limit Value for RUN Set Bits
These bits are used to define the window upper limit value for RUN. The reading of these bits returns the set value regardless of the Hardware watchdog configuration register (HWDG_CFG) LOCK bit value.
<table>
<tr><th>WDGRUNULS[31:0]</th><th>Description</th></tr>
<tr><td>During write operation</td><td>Set value written</td></tr>
<tr><td>During read operation</td><td>Set value read</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER14: Hardware Watchdog Timer
6. Register List
6.12. Hardware Watchdog Lower Limit PSS Setting Register (HWDG_PSSLLS)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit31:0] WDGPSSLLS[31:0]: Window lower limit value for PSS set bits
These bits are used to define the window lower limit value for PSS. The reading of these bits returns the set value regardless of the Hardware watchdog configuration register (HWDG_CFG) LOCK bit value.
<table>
<tr><th>bit31:0</th><th>Description</th></tr>
<tr><td>During write operation</td><td>When "0" is written to all 32 bits: The window function is disabled.
When any value except "0" is written: The window function is enabled.</td></tr>
<tr><td>During read operation</td><td>Set value read</td></tr>
</table>
(Correct)
[bit31:0] WDGPSSLLS[31:0]: Window Lower Limit Value for PSS Set Bits
These bits are used to define the window lower limit value for PSS. The reading of these bits returns the set value regardless of the Hardware watchdog configuration register (HWDG_CFG) LOCK bit value.
<table>
<tr><th>WDGPSSLLS[31:0]</th><th>Description</th></tr>
<tr><td>During write operation</td><td>When "0" is written to all 32 bits: The window function is disabled.
When any value except "0" is written: The window function is enabled.</td></tr>
<tr><td>During read operation</td><td>Set value read</td></tr>
</table></td></tr>
<tr><td>CHAPTER14: Hardware Watchdog Timer
6. Register List
6.13. Hardware Watchdog Upper Limit PSS Setting Register (HWDG_PSSULS)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit31:0] WDGPSSULS[31:0]: Window upper limit value for PSS set bits
These bits are used to define the window upper limit value for PSS. The reading of these bits returns the set value regardless of the Hardware watchdog configuration register (HWDG_CFG) LOCK bit value.
<table>
<tr><th>bit31:0</th><th>Description</th></tr>
<tr><td>During write operation</td><td>Set value written</td></tr>
<tr><td>During read operation</td><td>Set value read</td></tr>
</table>
(Correct)
[bit31:0] WDGPSSULS[31:0]: Window Upper Limit Value for PSS Set Bits
These bits are used to define the window upper limit value for PSS. The reading of these bits returns the set value regardless of the Hardware watchdog configuration register (HWDG_CFG) LOCK bit value.
<table>
<tr><th>WDGPSSULS[31:0]</th><th>Description</th></tr>
<tr><td>During write operation</td><td>Set value written</td></tr>
<tr><td>During read operation</td><td>Set value read</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER14: Hardware Watchdog Timer
6. Register List
6.14. Hardware Watchdog Reset Delay Counter Register (HWDG_RSTDLY)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit15:0] WDGRSTDLY[15:0]: Reset/NMI delay counter bits
These bits define the number of cycles for the delay time that is to be inserted before generation of the watchdog reset request or watchdog interrupt request (NMI). The reference clock for this delay time is the source clock of the watchdog counter.
<table>
<tr><th>bit15:0</th><th>Description</th></tr>
<tr><td>During write operation</td><td>Set value written</td></tr>
<tr><td>During read operation</td><td>"0x0000" is read.</td></tr>
</table>

(Correct)
[bit15:0] WDGRSTDLY[15:0]: Reset/NMI Delay Counter Bits
These bits define the number of cycles for the delay time that is to be inserted before generation of the watchdog reset request or watchdog interrupt request (NMI). The reference clock for this delay time is the source clock of the watchdog counter.
<table>
<tr><th>WDGRSTDLY[15:0]</th><th>Description</th></tr>
<tr><td>During write operation</td><td>Set value written</td></tr>
<tr><td>During read operation</td><td>0x0000 is read.</td></tr>
</table>
</td></tr>
<tr><td>CHAPTER14: Hardware Watchdog Timer
6. Register List
6.15. Hardware Watchdog Configuration Register (HWDG_CFG)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
This bit is set up by BootROM software.
<table>
<tr><th>bit</th><th>Description</th></tr>
<tr><td>During write operation</td><td>When "0" is written: Invalid
When "1" is written: The set values of the registers are locked.</td></tr>
<tr><td>During read operation</td><td>When "0" is read: The lock for the register set values is disabled.
When "1" is read: The lock for the register set values is enabled.</td></tr>
</table>

(Correct)
This bit is set up by BootROM software.
<table>
<tr><th>LOCK</th><th>Description</th></tr>
<tr><td>During write operation</td><td>When "0" is written: Invalid
When "1" is written: The set values of the registers are locked.</td></tr>
<tr><td>During read operation</td><td>When "0" is read: The lock for the register set values is disabled.
When "1" is read: The lock for the register set values is enabled.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER14: Hardware Watchdog Timer
6. Register List
6.15. Hardware Watchdog Configuration Register (HWDG_CFG)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit20:16] OBSSEL[4:0]: Watchdog counter monitor bit output selection bits
These bits are used for selecting any 1 bit in the watchdog counter (32 bits) as the output of the watchdog counter monitor bit.(Support target of this function is different depends on product specification. For details, see "3 Explanation of Operation For this product, output of the watchdog counter monitor bit of MCU output pin is supported by the Hardware watchdog timer.")
<table>
<tr><th>bit4:0</th><th>Description</th></tr>
<tr><td>During write operation</td><td>When "0b00000" is written: Bit 0 is selected for the monitor output.
When "0b00001" is written: Bit 1 is selected for the monitor output.
When "0b00010" is written: Bit 2 is selected for the monitor output.
・
・
・
When "0b11111" is written: Bit 31 is selected for the monitor output.</td></tr>
<tr><td>During read operation</td><td>Set value read</td></tr>
</table>

(Correct)
[bit20:16] OBSSEL[4:0]: WatchDog Counter Monitor Bit Output Selection Bits
These bits are used for selecting any 1 bit in the watchdog counter (32 bits) as the output of the watchdog counter monitor bit.(Support target of this function is different depends on product specification. For details, see "3 Explanation of Operation For this product, output of the watchdog counter monitor bit of MCU output pin is supported by the Hardware watchdog timer.")
<table>
<tr><th>OBSSEL[4:0]</th><th>Description</th></tr>
<tr><td>During write operation</td><td>When 0b00000 is written: Bit 0 is selected for the monitor output.
When 0b00001 is written: Bit 1 is selected for the monitor output.
When 0b00010 is written: Bit 2 is selected for the monitor output.
・
・
・
When 0b11111 is written: Bit 31 is selected for the monitor output.</td></tr>
<tr><td>During read operation</td><td>Set value read</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER14: Hardware Watchdog Timer
6. Register List
6.15. Hardware Watchdog Configuration Register (HWDG_CFG)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit9:8] CLKSEL[1:0]: Clock selection bits
These bits are used to select the source clock for the watchdog counter. When activated, the watchdog counter starts its operation as the high-speed CR clock. For details on clock switching, see "7 Precautions for Using This Device Switching of watchdog counter source clock.
<table>
<tr><th>bit1:0</th><th>Description</th></tr>
<tr><td>During write operation</td><td>When "0bX0" is written: The high-speed CR clock is selected.
When "0bX1" is written: The low-speed CR clock is selected.
Writing data in CLKSEL[1] bits does not affect operation.</td></tr>
<tr><td>During read operation</td><td>Set value read</td></tr>
</table>

(Correct)
[bit9:8] CLKSEL[1:0]: Clock Selection Bits
These bits are used to select the source clock for the watchdog counter. When activated, the watchdog counter starts its operation as the high-speed CR clock. For details on clock switching, see "7 Precautions for Using This Device Switching of watchdog counter source clock.
<table>
<tr><th>CLKSEL[1:0]</th><th>Description</th></tr>
<tr><td>During write operation</td><td>When 0bX0 is written: The high-speed CR clock is selected.
When 0bX1 is written: The low-speed CR clock is selected.
Writing data in CLKSEL[1] bits does not affect operation.</td></tr>
<tr><td>During read operation</td><td>Set value read</td></tr>
</table>
</td></tr>
<tr><td>CHAPTER14: Hardware Watchdog Timer
6. Register List
6.15. Hardware Watchdog Configuration Register (HWDG_CFG)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
<table>
<tr><th>bit</th><th>Description</th></tr>
<tr><td>During write operation</td><td>Invalid</td></tr>
<tr><td>During read operation</td><td>When "0" is read: Watchdog clock stop in PSS is disabled.
When "1" is read: Watchdog clock stop in PSS is enabled.</td></tr>
</table>

(Correct)
<table>
<tr><th>ALLOWSTOPCLK</th><th>Description</th></tr>
<tr><td>During write operation</td><td>Invalid</td></tr>
<tr><td>During read operation</td><td>When "0" is read: Watchdog clock stop in PSS is disabled.
When "1" is read: Watchdog clock stop in PSS is enabled.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER14: Hardware Watchdog Timer
6. Register List
6.15. Hardware Watchdog Configuration Register (HWDG_CFG)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit1] WDENPSS: Watchdog counter for PSS enable bit
This bit is used to enable the watchdog counter in PSS. This bit is enabled when the LOCK bit in the Hardware watchdog configuration register (HWDG_CFG) is set to "1". (This bit is fixed by hardware, "0" is always read.)
<table>
<tr><th>bit</th><th>Description</th></tr>
<tr><td>During write operation</td><td>Invalid</td></tr>
<tr><td>During read operation</td><td>When "0" is read: The watchdog counter in PSS is disabled.
When "1" is read: The watchdog counter in PSS is enabled.</td></tr>
</table>
[bit0] WDENRUN: Watchdog counter for RUN enable bit
This bit is used to enable the watchdog counter in RUN. This bit is enabled when the LOCK bit in the Hardware watchdog configuration register (HWDG_CFG) is set to "1".
This bit indicates the value of the entire system enable signal.
See This section describes the block diagrams of the Hardware watchdog timer.
<table>
<tr><th>bit</th><th>Description</th></tr>
<tr><td>During write operation</td><td>Invalid</td></tr>
<tr><td>During read operation</td><td>When "0" is read: The watchdog counter in RUN is disabled.
When "1" is read: The watchdog counter in RUN is enabled.</td></tr>
</table>
(Correct)
[bit1] WDENPSS: Watchdog Counter for PSS Enable Bit
This bit is used to enable the watchdog counter in PSS. This bit is enabled when the LOCK bit in the Hardware watchdog configuration register (HWDG_CFG) is set to "1". (This bit is fixed by hardware, "0" is always read.)
<table>
<tr><th>WDENPSS</th><th>Description</th></tr>
<tr><td>During write operation</td><td>Invalid</td></tr>
<tr><td>During read operation</td><td>When "0" is read: The watchdog counter in PSS is disabled.
When "1" is read: The watchdog counter in PSS is enabled.</td></tr>
</table>
[bit0] WDENRUN: Watchdog Counter for RUN Enable Bit
This bit is used to enable the watchdog counter in RUN. This bit is enabled when the LOCK bit in the Hardware watchdog configuration register (HWDG_CFG) is set to "1".
This bit indicates the value of the entire system enable signal.
See This section describes the block diagrams of the Hardware watchdog timer.
<table>
<tr><th>WDENRUN</th><th>Description</th></tr>
<tr><td>During write operation</td><td>Invalid</td></tr>
<tr><td>During read operation</td><td>When "0" is read: The watchdog counter in RUN is disabled.
When "1" is read: The watchdog counter in RUN is enabled.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER14: Hardware Watchdog Timer 6. Register List 6.16. Hardware Watchdog Lower Limit RUN Current Register (HWDG_RUNLLC)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit31:0] WDGRUNLLC[31:0]: Window lower limit for RUN current bits
These bits are used to define the window lower limit value for RUN. The reading of these bits returns the initial value, until the Hardware watchdog configuration register (HWDG_CFG) LOCK bit is set.
<table>
<tr><th>bit31:0</th><th>Description</th></tr>
<tr><td>During write operation</td><td>Invalid</td></tr>
<tr><td>During read operation</td><td>Set value read</td></tr>
</table>
(Correct)
[bit31:0] WDGRUNLLC[31:0]: Window Lower Limit for RUN Current Bits
These bits are used to define the window lower limit value for RUN. The reading of these bits returns the initial value, until the Hardware watchdog configuration register (HWDG_CFG) LOCK bit is set.
<table>
<tr><th>WDGRUNLLC[31:0]</th><th>Description</th></tr>
<tr><td>During write operation</td><td>Invalid</td></tr>
<tr><td>During read operation</td><td>Set value read</td></tr>
</table></td></tr>
<tr><td>CHAPTER14: Hardware Watchdog Timer 6. Register List 6.17. Hardware Watchdog Upper Limit RUN Current Register (HWDG_RUNULC)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit31:0] WDGRUNULC[31:0]: Window upper limit for RUN current bits
These bits are used to define the window upper limit value for RUN. The reading of these bits returns the initial value, until the Hardware watchdog configuration register (HWDG_CFG) LOCK bit is set.
<table>
<tr><th>bit31:0</th><th>Description</th></tr>
<tr><td>During write operation</td><td>Invalid</td></tr>
<tr><td>During read operation</td><td>Set value read</td></tr>
</table>
(Correct)
[bit31:0] WDGRUNULC[31:0]: Window Upper Limit for RUN Current Bits
These bits are used to define the window upper limit value for RUN. The reading of these bits returns the initial value, until the Hardware watchdog configuration register (HWDG_CFG) LOCK bit is set.
<table>
<tr><th>WDGRUNULC[31:0]</th><th>Description</th></tr>
<tr><td>During write operation</td><td>Invalid</td></tr>
<tr><td>During read operation</td><td>Set value read</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER14: Hardware Watchdog Timer
6. Register List
6.18.Hardware Watchdog Lower Limit PSS Current Register (HWDG_PSSLLC)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit31:0] WDGPSSLLC[31:0]: Window lower limit for PSS current bits
These bits are used to define the window lower limit value for PSS. The reading of these bits returns the initial value, until the Hardware watchdog configuration register (HWDG_CFG) LOCK bit is set.
<table>
<tr><th>bit31:0</th><th>Description</th></tr>
<tr><td>During write operation</td><td>Invalid</td></tr>
<tr><td>During read operation</td><td>Set value read</td></tr>
</table>
(Correct)
[bit31:0] WDGPSSLLC[31:0]: Window Lower Limit for PSS Current Bits
These bits are used to define the window lower limit value for PSS. The reading of these bits returns the initial value, until the Hardware watchdog configuration register (HWDG_CFG) LOCK bit is set.
<table>
<tr><th>WDGPSSLLC[31:0]</th><th>Description</th></tr>
<tr><td>During write operation</td><td>Invalid</td></tr>
<tr><td>During read operation</td><td>Set value read</td></tr>
</table></td></tr>
<tr><td>CHAPTER14: Hardware Watchdog Timer
6. Register List
6.19. Hardware Watchdog Upper Limit PSS Current Register (HWDG_PSSULC)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit31:0] WDGPSSULC[31:0]: Window upper limit for PSS current bits
These bits are used to define the window upper limit value for PSS. The reading of these bits returns the initial value, until the Hardware watchdog configuration register (HWDG_CFG) LOCK bit is set.
<table>
<tr><th>bit31:0</th><th>Description</th></tr>
<tr><td>During write operation</td><td>Invalid</td></tr>
<tr><td>During read operation</td><td>Set value read</td></tr>
</table>
(Correct)
[bit31:0] WDGPSSULC[31:0]: Window Upper Limit for PSS Current Bits
These bits are used to define the window upper limit value for PSS. The reading of these bits returns the initial value, until the Hardware watchdog configuration register (HWDG_CFG) LOCK bit is set.
<table>
<tr><th>WDGPSSULC[31:0]</th><th>Description</th></tr>
<tr><td>During write operation</td><td>Invalid</td></tr>
<tr><td>During read operation</td><td>Set value read</td></tr>
</table></td></tr>
</table>

Section	Change Results
CHAPTER14: Hardware Watchdog Timer 7. Precautions for Using This Device	Features of "For details" should be corrected as indicated by the shading below. (Error) Switching of watchdog counter source clock The system controller controls the switching of the watchdog counter source clock since the source clock control of the watchdog counter is a part of the system setting information (profile). To change the watchdog counter source clock, follow the sequence below. 12.Setting the CLKSEL bit in the Hardware watchdog configuration register (HWDG_CFG) sets the new source clock of the watchdog counter. Then, writing data to the following registers finalizes the window setting. 13.Hardware watchdog lower limit RUN setting register (HWDG_RUNLLS) 14.Hardware watchdog upper limit RUN setting register (HWDG_RUNULS) 15.Hardware watchdog lower limit PSS setting register (HWDG_PSSLLS) 16.Hardware watchdog upper limit PSS setting register (HWDG_PSSULS) 17.Setting the LOCK bit in the Hardware watchdog configuration register (HWDG_CFG) to "1" locks rewriting of the register set values. 18.Writing "0xAB" to the RUN profile update trigger register of the system controller (SYSC_TRGRUNCNTR) executes RUN profile update. At this point in time, the source clock of the watchdog counter is switched to the new source clock selected by the CLKSEL bit of the Hardware watchdog configuration register (HWDG_CFG). (Correct) Switching of Watchdog Counter Source Clock The system controller controls the switching of the watchdog counter source clock since the source clock control of the watchdog counter is a part of the system setting information (profile). To change the watchdog counter source clock, follow the sequence below. 1.Setting the CLKSEL bit in the Hardware watchdog configuration register (HWDG_CFG) sets the new source clock of the watchdog counter. Then, writing data to the following registers finalizes the window setting. 2.Hardware watchdog lower limit RUN setting register (HWDG_RUNLLS) 3.Hardware watchdog upper limit RUN setting register (HWDG_RUNULS) 4.Hardware watchdog lower limit PSS setting register (HWDG_PSSLLS) 5.Hardware watchdog upper limit PSS setting register (HWDG_PSSULS) 6.Setting the LOCK bit in the Hardware watchdog configuration register (HWDG_CFG) to "1" locks rewriting of the register set values. 7.Writing 0xAB to the RUN profile update trigger register of the system controller (SYSC_TRGRUNCNTR) executes RUN profile update. At this point in time, the source clock of the watchdog counter is switched to the new source clock selected by the CLKSEL bit of the Hardware watchdog configuration register (HWDG_CFG)

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER15: Software Watchdog Timer
1. Overview</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
This section provides an overview of the Software watchdog timer.
The Software watchdog timer is positioned in the MCU configuration group, used for detecting a runaway state caused by a user program, also the Software watchdog timer monitors user program by the CPU.

(Correct)
This section provides an overview of the Software watchdog timer.
The Software watchdog timer is positioned in the SYS1 group, used for detecting a runaway state caused by a user program, also the Software watchdog timer monitors user program by the CPU.</td></tr>
<tr><td>CHAPTER15: Software Watchdog Timer
6. Register List
6.1.Software Watchdog Protection Register (SWDT_PROT)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit31:0] KEY[31:0]: Protection bits
<table>
<tr><th>bit31:0</th><th>Description</th></tr>
<tr><td>During write operation</td><td>When "0xEDAC_CE55" is written:
Protection lock for register write is released.
When a value other than "0xEDAC_CE55" is written:
Protection lock for register write is not released.</td></tr>
<tr><td>During read operation</td><td>When "0xFFFF_FFFF" is read:
Protection lock for register write has been released.
When "0x0000_0000" is read:
Protection lock for register write has been enabled.</td></tr>
</table>
(Correct)
[bit31:0] KEY[31:0]: Protection Bits
<table>
<tr><th>KEY[31:0]</th><th>Description</th></tr>
<tr><td>During write operation</td><td>When 0xEDAC_CE55 is written:
Protection lock for register write is released.
When a value other than 0xEDAC_CE55 is written:
Protection lock for register write is not released.</td></tr>
<tr><td>During read operation</td><td>When 0xFFFF_FFFF is read:
Protection lock for register write has been released.
When 0x0000_0000 is read:
Protection lock for register write has been enabled.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER15: Software Watchdog Timer
6. Register List
6.1.Software Watchdog Protection Register (SWDT_PROT)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
−A protect key will be locked again by writing to the address where is in the same group area (MCU Config Group), however protect key will not be locked by writing to no protect target register.

(Correct)
−A protect key will be locked again by writing to the address where is in the same group area(SYS1 Group), however protect key will not be locked by writing to no protect target register.</td></tr>
<tr><td>CHAPTER15: Software Watchdog Timer
6. Register List
6.2.Software Watchdog Counter Register (SWDG_CNT)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit31:0] WDGCNT[31:0]: Watchdog counter bits
<table><tr><th>bit31:0</th><th>Description</th></tr><tr><td>During write operation</td><td>Invalid</td></tr><tr><td>During read operation</td><td>Returns the current watchdog counter value.
The watchdog counter value is sampled by using the bus clock.</td></tr></table>
(Correct)
[bit31:0] WDGCNT[31:0]: Watchdog Counter Bits
<table><tr><th>WDGCNT[31:0]</th><th>Description</th></tr><tr><td>During write operation</td><td>Invalid</td></tr><tr><td>During read operation</td><td>Returns the current watchdog counter value.
The watchdog counter value is sampled by using the bus clock.</td></tr></table></td></tr>
<tr><td>CHAPTER15: Software Watchdog Timer
6. Register List
6.3.Software Watchdog Reset Factor Register (SWDG_RSTCAUSE)</td><td>The item should be added as indicated by the shading below.

(Error)
This register is not initialized by reset. To check the factor of the watchdog reset request, read this register.

(Correct)
This register is not initialized by reset. After a watch dock reset, the value is maintained. To check the factor of the watchdog reset request, read this register.</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER15: Software Watchdog Timer
6. Register List
6.3.Software Watchdog Reset Factor Register (SWDG_RSTCAUSE)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit4] RSTCAUSE4: Reset factor bit 4
If the watchdog counter clear protection trigger sequence is executed when the LOCK bit in the Software watchdog configuration register (SWDG_CFG) is "0", this bit is set to "1".
<table>
<tr><th>bit</th><th>Description</th></tr>
<tr><td>During write operation</td><td>When "0" is written: This bit cleared
When "1" is written: Invalid</td></tr>
<tr><td>During read operation</td><td>When "0" is read: Indicating no detection of a reset/NMI factor
When "1" is read: Indicating detection of a reset/NMI factor</td></tr>
</table>
[bit3] RSTCAUSE3: Reset factor bit 3
If the watchdog counter clear protection trigger sequence is executed before the watchdog counter reaches the window lower limit value, this bit is set to "1".
<table>
<tr><th>bit</th><th>Description</th></tr>
<tr><td>During write operation</td><td>When "0" is written: This bit cleared
When "1" is written: Invalid</td></tr>
<tr><td>During read operation</td><td>When "0" is read: Indicating no detection of a reset/NMI factor
When "1" is read: Indicating detection of a reset/NMI factor</td></tr>
</table>

(Correct)
[bit4] RSTCAUSE4: Reset Factor Bit 4
If the watchdog counter clear protection trigger sequence is executed when the LOCK bit in the Software watchdog configuration register (SWDG_CFG) is "0", this bit is set to "1".
<table>
<tr><th>RSTCAUSE4</th><th>Description</th></tr>
<tr><td>During write operation</td><td>When "0" is written: This bit cleared
When "1" is written: Invalid</td></tr>
<tr><td>During read operation</td><td>When "0" is read: Indicating no detection of a reset/NMI factor
When "1" is read: Indicating detection of a reset/NMI factor</td></tr>
</table>
[bit3] RSTCAUSE3: Reset Factor Bit 3
If the watchdog counter clear protection trigger sequence is executed before the watchdog counter reaches the window lower limit value, this bit is set to "1".
<table>
<tr><th>RSTCAUSE3</th><th>Description</th></tr>
<tr><td>During write operation</td><td>When "0" is written: This bit cleared
When "1" is written: Invalid</td></tr>
<tr><td>During read operation</td><td>When "0" is read: Indicating no detection of a reset/NMI factor
When "1" is read: Indicating detection of a reset/NMI factor</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER15: Software Watchdog Timer
6. Register List
6.3.Software Watchdog Reset Factor Register (SWDG_RSTCAUSE)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit2] RSTCAUSE2: Reset factor bit 2
When the watchdog counter reaches the window upper limit value, this bit is set to "1".
<table>
<tr><th>bit</th><th>Description</th></tr>
<tr><td>During write operation</td><td>When "0" is written: This bit cleared
When "1" is written: Invalid</td></tr>
<tr><td>During read operation</td><td>When "0" is read: Indicating no detection of a reset/NMI factor
When "1" is read: Indicating detection of a reset/NMI factor</td></tr>
</table>

(Correct)
[bit2] RSTCAUSE2: Reset Factor Bit 2
When the watchdog counter reaches the window upper limit value, this bit is set to "1".
<table>
<tr><th>RSTCAUSE2</th><th>Description</th></tr>
<tr><td>During write operation</td><td>When "0" is written: This bit cleared
When "1" is written: Invalid</td></tr>
<tr><td>During read operation</td><td>When "0" is read: Indicating no detection of a reset/NMI factor
When "1" is read: Indicating detection of a reset/NMI factor</td></tr>
</table>
</td></tr>
<tr><td>CHAPTER15: Software Watchdog Timer
6. Register List
6.3.Software Watchdog Reset Factor Register (SWDG_RSTCAUSE)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit1] RSTCAUSE1: Reset factor bit 1
When there is a violation of the watchdog counter clear protection trigger sequence, this bit is set to "1". For details, see Figure 3-2.
<table>
<tr><th>bit</th><th>Description</th></tr>
<tr><td>During write operation</td><td>When "0" is written: This bit cleared
When "1" is written: Invalid</td></tr>
<tr><td>During read operation</td><td>When "0" is read: Indicating no detection of a reset/NMI factor
When "1" is read: Indicating detection of a reset/NMI factor</td></tr>
</table>

(Correct)
[bit1] RSTCAUSE1: Reset Factor Bit 1
When there is a violation of the watchdog counter clear protection trigger sequence, this bit is set to "1". For details, see Figure 3-2.
<table>
<tr><th>RSTCAUSE1</th><th>Description</th></tr>
<tr><td>During write operation</td><td>When "0" is written: This bit cleared
When "1" is written: Invalid</td></tr>
<tr><td>During read operation</td><td>When "0" is read: Indicating no detection of a reset/NMI factor
When "1" is read: Indicating detection of a reset/NMI factor</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER15: Software Watchdog Timer
6. Register List
6.3.Software Watchdog Reset Factor Register (SWDG_RSTCAUSE)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit0] RSTCAUSE0: Reset factor bit 0
If the value written in the Software watchdog trigger 0/1 register (SWDG_TRG0 or SWDG_TRG1) does not match a proper value, this bit is set to "1".
<table>
<tr><th>bit</th><th>Description</th></tr>
<tr><td>During write operation</td><td>When "0" is written: This bit cleared
When "1" is written: Invalid</td></tr>
<tr><td>During read operation</td><td>When "0" is read: Indicating no detection of a reset/NMI factor
When "1" is read: Indicating detection of a reset/NMI factor</td></tr>
</table>

(Correct)
[bit0] RSTCAUSE0: Reset Factor Bit 0
If the value written in the Software watchdog trigger 0/1 register (SWDG_TRG0 or SWDG_TRG1) does not match a proper value, this bit is set to "1".
<table>
<tr><th>RSTCAUSE0</th><th>Description</th></tr>
<tr><td>During write operation</td><td>When "0" is written: This bit cleared
When "1" is written: Invalid</td></tr>
<tr><td>During read operation</td><td>When "0" is read: Indicating no detection of a reset/NMI factor
When "1" is read: Indicating detection of a reset/NMI factor</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER15: Software Watchdog Timer
6. Register List
6.4. Software Watchdog Trigger 0 Register (SWDG_TRG0)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit7:0] WDGTRG0[7:0]: Watchdog trigger 0 bits
These bits are used to execute the watchdog counter clear protection trigger sequence to clear the watchdog counter.
<table>
<tr><th>bit7:0</th><th>Description</th></tr>
<tr><td>During write operation</td><td>When the SWDG_TRG0CFG value is written:
1 condition for executing the watchdog counter clear protection trigger sequence is met.
When a value other than the SWDG_TRG0CFG value is written:
A watchdog error is generated.</td></tr>
<tr><td>During read operation</td><td>Reads "0b00000000".</td></tr>
</table>

(Correct)
[bit7:0] WDGTRG0[7:0]: Watchdog Trigger 0 Bits
These bits are used to execute the watchdog counter clear protection trigger sequence to clear the watchdog counter.
<table>
<tr><th>WDGTRG0[7:0]</th><th>Description</th></tr>
<tr><td>During write operation</td><td>When the SWDG_TRG0CFG value is written:
1 condition for executing the watchdog counter clear protection trigger sequence is met.
When a value other than the SWDG_TRG0CFG value is written:
A watchdog error is generated.</td></tr>
<tr><td>During read operation</td><td>Reads 0b00000000.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER15: Software Watchdog Timer
6. Register List
6.5. Software Watchdog Trigger 1 Register (SWDG_TRG1)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit7:0] WDGTRG1[7:0]: Watchdog trigger 1 bits
These bits are used to execute the watchdog counter clear protection trigger sequence to clear the watchdog counter.
<table>
<tr><th>bit7:0</th><th>Description</th></tr>
<tr><td>During write operation</td><td>When the SWDG_TRG1CFG value is written:
1 condition for executing the watchdog counter clear protection trigger sequence is met.
When a value other than the SWDG_TRG1CFG value is written:
A watchdog error is generated.</td></tr>
<tr><td>During read operation</td><td>Reads "0b00000000".</td></tr>
</table>

(Correct)
[bit7:0] WDGTRG1[7:0]: Watchdog Trigger 1 Bits
These bits are used to execute the watchdog counter clear protection trigger sequence to clear the watchdog counter.
<table>
<tr><th>WDGTRG1[7:0]</th><th>Description</th></tr>
<tr><td>During write operation</td><td>When the SWDG_TRG1CFG value is written:
1 condition for executing the watchdog counter clear protection trigger sequence is met.
When a value other than the SWDG_TRG1CFG value is written:
A watchdog error is generated.</td></tr>
<tr><td>During read operation</td><td>Reads 0b00000000.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER15: Software Watchdog Timer
6. Register List
6.6. Software Watchdog Interrupt Configuration Register (SWDG_INT)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit17] RSTEN: Reset/NMI enable bit
This bit is used to generate either the watchdog reset request or watchdog interrupt request (NMI) in response to a watchdog error. Prohibiting setting this bit to "0" except for the purpose of using it as a test function. (If this bit is set to "0" during application execution, security is compromised.) This bit is equipped with a majority circuit with 3 FFs as a measure against a bit invert caused by the effects of noise or another factor.
<table>
<tr><th>bit</th><th>Description</th></tr>
<tr><td>During write operation</td><td>When "0" is written: The watchdog interrupt request (NMI) is generated.
When "1" is written: The watchdog interrupt request is generated.</td></tr>
<tr><td>During read operation</td><td>Set value read</td></tr>
</table>
[bit16] IRQEN: Prior warning interrupt enable bit
This bit is used to enable generation of the prior warning interrupt request.
<table>
<tr><th>bit</th><th>Description</th></tr>
<tr><td>During write operation</td><td>When "0" is written: The prior warning interrupt request is not generated.
When "1" is written: The prior warning interrupt request is generated.</td></tr>
<tr><td>During read operation</td><td>Set value read</td></tr>
</table>
(Correct)
[bit17] RSTEN: Reset/NMI Enable Bit
This bit is used to generate either the watchdog reset request or watchdog interrupt request (NMI) in response to a watchdog error. Prohibiting setting this bit to "0" except for the purpose of using it as a test function. (If this bit is set to "0" during application execution, security is compromised.) This bit is equipped with a majority circuit with 3 FFs as a measure against a bit invert caused by the effects of noise or another factor.
<table>
<tr><th>RSTEN</th><th>Description</th></tr>
<tr><td>During write operation</td><td>When "0" is written: The watchdog interrupt request (NMI) is generated.
When "1" is written: The watchdog interrupt request is generated.</td></tr>
<tr><td>During read operation</td><td>Set value read</td></tr>
</table>
[bit16] IRQEN: Prior warning Interrupt Enable Bit
This bit is used to enable generation of the prior warning interrupt request.
<table>
<tr><th>IRQEN</th><th>Description</th></tr>
<tr><td>During write operation</td><td>When "0" is written: The prior warning interrupt request is not generated.
When "1" is written: The prior warning interrupt request is generated.</td></tr>
<tr><td>During read operation</td><td>Set value read</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER15: Software Watchdog Timer
6. Register List
6.6. Software Watchdog Interrupt Configuration Register (SWDG_INT)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit1] NMIFLAG: NMI flag
This bit is set by a watchdog error when the RSTEN bit in the Software watchdog interrupt configuration register (SWDG_INT) is "0". When the RSTEN bit in the Software watchdog interrupt clear register (SWDG_INT) is "1", the watchdog reset request is generated. You can clear this bit by writing "1" to the NMICLR bit in the Software watchdog interrupt configuration register (SWDG_INTCLR).
<table>
<tr><th>bit</th><th>Description</th></tr>
<tr><td>During write operation</td><td>Invalid</td></tr>
<tr><td>During read operation</td><td>When "0" is read: No detection of a watchdog error (NMI) is indicated.
When "1" is read: Detection of a watchdog error (NMI) is indicated.</td></tr>
</table>
[bit0] IRQFLAG: IRQ flag
This bit is set by a watchdog error. The prior warning interrupt is generated when the IRQEN bit in the Software watchdog interrupt clear register (SWDG_INT) is "1". You can clear this bit by writing "1" to the IRQCLR bit in the Software watchdog interrupt configuration register (SWDG_INTCLR).
<table>
<tr><th>bit</th><th>Description</th></tr>
<tr><td>During write operation</td><td>Invalid</td></tr>
<tr><td>During read operation</td><td>When "0" is read: No detection of a watchdog error (IRQ) is indicated.
When "1" is read: Detection of a watchdog error (IRQ) is indicated.</td></tr>
</table>

(Correct)
[bit1] NMIFLAG: NMI Flag
This bit is set by a watchdog error when the RSTEN bit in the Software watchdog interrupt configuration register (SWDG_INT) is "0". When the RSTEN bit in the Software watchdog interrupt clear register (SWDG_INT) is "1", the watchdog reset request is generated. You can clear this bit by writing "1" to the NMICLR bit in the Software watchdog interrupt configuration register (SWDG_INTCLR).
<table>
<tr><th>NMIFLAG</th><th>Description</th></tr>
<tr><td>During write operation</td><td>Invalid</td></tr>
<tr><td>During read operation</td><td>When "0" is read: No detection of a watchdog error (NMI) is indicated.
When "1" is read: Detection of a watchdog error (NMI) is indicated.</td></tr>
</table>
[bit0] IRQFLAG: IRQ Flag
This bit is set by a watchdog error. The prior warning interrupt is generated when the IRQEN bit in the Software watchdog interrupt clear register (SWDG_INT) is "1". You can clear this bit by writing "1" to the IRQCLR bit in the Software watchdog interrupt configuration register (SWDG_INTCLR).
<table>
<tr><th>IRQFLAG</th><th>Description</th></tr>
<tr><td>During write operation</td><td>Invalid</td></tr>
<tr><td>During read operation</td><td>When "0" is read: No detection of a watchdog error (IRQ) is indicated.
When "1" is read: Detection of a watchdog error (IRQ) is indicated.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER15: Software Watchdog Timer
6. Register List
6.7. Software Watchdog Interrupt Clear Register (SWDG_INTCLR)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit1] NMICLR: NMI clear bit
This bit is used to clear the NMIFLAG bit in the Software watchdog interrupt configuration register (SWDG_INT).
<table>
<tr><th>bit</th><th>Description</th></tr>
<tr><td>During write operation</td><td>When "0" is written: Invalid
When "1" is written: The NMI flag is cleared.</td></tr>
<tr><td>During read operation</td><td>"0" is read.</td></tr>
</table>
You must clear the NMI after waiting for watchdog counter clear (0x0000_0000). Check the watchdog counter value with SWDG_CNT.

[bit0] IRQCLR: Prior warning interrupt clear bit
This bit is used to clear the IRQFLAG bit in the Software watchdog interrupt configuration register (SWDG_INT).
<table>
<tr><th>bit</th><th>Description</th></tr>
<tr><td>During write operation</td><td>When "0" is written: Invalid
When "1" is written: The IRQ flag is cleared.</td></tr>
<tr><td>During read operation</td><td>"0" is read.</td></tr>
</table>

(Correct)
[bit1] NMICLR: NMI Clear Bit
This bit is used to clear the NMIFLAG bit in the Software watchdog interrupt configuration register (SWDG_INT).
<table>
<tr><th>NMICLR</th><th>Description</th></tr>
<tr><td>During write operation</td><td>When "0" is written: Invalid
When "1" is written: The NMI flag is cleared.</td></tr>
<tr><td>During read operation</td><td>"0" is read.</td></tr>
</table>
You must clear the NMI after waiting for watchdog counter clear (0x0000_0000). Check the watchdog counter value with SWDG_CNT.

[bit0] IRQCLR: Prior warning Interrupt Clear Bit
This bit is used to clear the IRQFLAG bit in the Software watchdog interrupt configuration register (SWDG_INT).
<table>
<tr><th>IRQCLR</th><th>Description</th></tr>
<tr><td>During write operation</td><td>When "0" is written: Invalid
When "1" is written: The IRQ flag is cleared.</td></tr>
<tr><td>During read operation</td><td>"0" is read.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER15: Software Watchdog Timer 6. Register List 6.8. Software Watchdog Trigger 0 Configuration Register (SWDG_TRG0CFG)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit7:0] WDGTRG0CFG[7:0]: Watchdog trigger 0 configuration bits
These bits are used to define a value to be written to the Software watchdog trigger 0 register (SWDG_TRG0) for execution of the watchdog counter clear protection trigger sequence.
<table>
<tr><th>bit7:0</th><th>Description</th></tr>
<tr><td>During write operation</td><td>Set value written</td></tr>
<tr><td>During read operation</td><td>Set value read</td></tr>
</table>
(Correct)
[bit7:0] WDGTRG0CFG[7:0]: Watchdog Trigger 0 Configuration Bits
These bits are used to define a value to be written to the Software watchdog trigger 0 register (SWDG_TRG0) for execution of the watchdog counter clear protection trigger sequence.
<table>
<tr><th>WDGTRG0CFG[7:0]</th><th>Description</th></tr>
<tr><td>During write operation</td><td>Set value written</td></tr>
<tr><td>During read operation</td><td>Set value read</td></tr>
</table></td></tr>
<tr><td>CHAPTER15: Software Watchdog Timer 6. Register List 6.9. Software Watchdog Trigger 1 Configuration Register (SWDG_TRG1CFG)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit7:0] WDGTRG1CFG[7:0]: Watchdog trigger 1 configuration bits
These bits are used to define a value to be written to the Software watchdog trigger 1 register (SWDG_TRG1) for execution of the watchdog counter clear protection trigger sequence.
<table>
<tr><th>bit7:0</th><th>Description</th></tr>
<tr><td>During write operation</td><td>Set value written</td></tr>
<tr><td>During read operation</td><td>Set value read</td></tr>
</table>
(Correct)
[bit7:0] WDGTRG1CFG[7:0]: Watchdog Trigger 1 Configuration Bits
These bits are used to define a value to be written to the Software watchdog trigger 1 register (SWDG_TRG1) for execution of the watchdog counter clear protection trigger sequence.
<table>
<tr><th>WDGTRG1CFG[7:0]</th><th>Description</th></tr>
<tr><td>During write operation</td><td>Set value written</td></tr>
<tr><td>During read operation</td><td>Set value read</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER15: Software Watchdog Timer
6. Register List
6.10. Software Watchdog Lower Limit RUN Setting Register (SWDG_RUNLLS)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit31:0] WDGRUNLLS[31:0]: Window lower limit value for RUN set bits
These bits are used to define the window lower limit value for RUN. The reading of these bits returns the set value regardless of the Software watchdog configuration register (SWDG_CFG) LOCK bit value.
<table>
<tr><th>bit31:0</th><th>Description</th></tr>
<tr><td>During write operation</td><td>When "0" is written to all 32 bits: The window function is disabled.
When any value except "0" is written: The window function is enabled.</td></tr>
<tr><td>During read operation</td><td>Set value read</td></tr>
</table>

(Correct)
[bit31:0] WDGRUNLLS[31:0]: Window Lower Limit Value for RUN Set Bits
These bits are used to define the window lower limit value for RUN. The reading of these bits returns the set value regardless of the Software watchdog configuration register (SWDG_CFG) LOCK bit value.
<table>
<tr><th>WDGRUNLLS[31:0]</th><th>Description</th></tr>
<tr><td>During write operation</td><td>When "0" is written to all 32 bits: The window function is disabled.
When any value except "0" is written: The window function is enabled.</td></tr>
<tr><td>During read operation</td><td>Set value read</td></tr>
</table>
</td></tr>
<tr><td>CHAPTER15: Software Watchdog Timer
6. Register List
6.11. Software Watchdog Upper Limit RUN Setting Register (SWDG_RUNULS)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit31:0] WDGRUNULS[31:0]: Window upper limit value for RUN set bits
These bits are used to define the window upper limit value for RUN. The reading of these bits returns the set value regardless of the Software watchdog configuration register (SWDG_CFG) LOCK bit value.
<table>
<tr><th>bit31:0</th><th>Description</th></tr>
<tr><td>During write operation</td><td>Set value written</td></tr>
<tr><td>During read operation</td><td>Set value read</td></tr>
</table>

(Correct)
[bit31:0] WDGRUNULS[31:0]: Window Upper Limit Value for RUN Set Bits
These bits are used to define the window upper limit value for RUN. The reading of these bits returns the set value regardless of the Software watchdog configuration register (SWDG_CFG) LOCK bit value.
<table>
<tr><th>WDGRUNULS[31:0]</th><th>Description</th></tr>
<tr><td>During write operation</td><td>Set value written</td></tr>
<tr><td>During read operation</td><td>Set value read</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER15: Software Watchdog Timer
6. Register List
6.12. Software Watchdog Lower Limit PSS Setting Register (SWDG_PSSLLS)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit31:0] WDGPSSLLS[31:0]: Window lower limit value for PSS set bits
These bits are used to define the window lower limit value for PSS. The reading of these bits returns the set value regardless of the Software watchdog configuration register (SWDG_CFG) LOCK bit value.
<table>
<tr><th>bit31:0</th><th>Description</th></tr>
<tr><td>During write operation</td><td>When "0" is written to all 32 bits: The window function is disabled.
When any value except "0" is written: The window function is enabled.</td></tr>
<tr><td>During read operation</td><td>Set value read</td></tr>
</table>

(Correct)
[bit31:0] WDGPSSLLS[31:0]: Window Lower Limit Value for PSS Set Bits
These bits are used to define the window lower limit value for PSS. The reading of these bits returns the set value regardless of the Software watchdog configuration register (SWDG_CFG) LOCK bit value.
<table>
<tr><th>WDGPSSLLS[31:0]</th><th>Description</th></tr>
<tr><td>During write operation</td><td>When "0" is written to all 32 bits: The window function is disabled.
When any value except "0" is written: The window function is enabled.</td></tr>
<tr><td>During read operation</td><td>Set value read</td></tr>
</table>
</td></tr>
<tr><td>CHAPTER15: Software Watchdog Timer
6. Register List
6.13. Software Watchdog Upper Limit PSS Setting Register (SWDG_PSSULS)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit31:0] WDGPSSULS[31:0]: Window upper limit value for PSS set bits
These bits are used to define the window upper limit value for PSS. The reading of these bits returns the set value regardless of the Software watchdog configuration register (SWDG_CFG) LOCK bit value.
<table>
<tr><th>bit31:0</th><th>Description</th></tr>
<tr><td>During write operation</td><td>Set value written</td></tr>
<tr><td>During read operation</td><td>Set value read</td></tr>
</table>

(Correct)
[bit31:0] WDGPSSULS[31:0]: Window Upper Limit Value for PSS Set Bits
These bits are used to define the window upper limit value for PSS. The reading of these bits returns the set value regardless of the Software watchdog configuration register (SWDG_CFG) LOCK bit value.
<table>
<tr><th>WDGPSSULS[31:0]</th><th>Description</th></tr>
<tr><td>During write operation</td><td>Set value written</td></tr>
<tr><td>During read operation</td><td>Set value read</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER15: Software Watchdog Timer
6. Register List
6.14. Software Watchdog Reset Delay Counter Register (SWDG_RSTDLY)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit15:0] WDGRSTDLY[15:0]: Reset/NMI delay counter bits
These bits define the number of cycles for the delay time that is to be inserted before generation of the watchdog reset request or watchdog interrupt request (NMI). The reference clock for this delay time is the source clock of the watchdog counter.
<table>
<tr><th>bit15:0</th><th>Description</th></tr>
<tr><td>During write operation</td><td>Set value written</td></tr>
<tr><td>During read operation</td><td>"0x0000" is read.</td></tr>
</table>

(Correct)
[bit15:0] WDGRSTDLY[15:0]: Reset/NMI Delay Counter Bits
These bits define the number of cycles for the delay time that is to be inserted before generation of the watchdog reset request or watchdog interrupt request (NMI). The reference clock for this delay time is the source clock of the watchdog counter.
<table>
<tr><th>WDGRSTDLY[15:0]</th><th>Description</th></tr>
<tr><td>During write operation</td><td>Set value written</td></tr>
<tr><td>During read operation</td><td>0x0000 is read.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER15: Software Watchdog Timer
6. Register List
6.15. Software Watchdog Configuration Register (SWDG_CFG)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit24] LOCK: Lock bit
You can write data to this bit only once. This register is used to prevent rewriting of the set values of various registers. When this bit is "0", you can rewrite the setting values of various registers. When this bit is set to "1" the watchdog counter is automatically cleared. This bit is equipped with a majority circuit with 3 FFs as a measure against a bit invert caused by the effects of noise or another factor.
<table>
<tr><th>bit</th><th>Description</th></tr>
<tr><td>During write operation</td><td>When "0" is written: Invalid
When "1" is written: The set values of the registers are locked.</td></tr>
<tr><td>During read operation</td><td>When "0" is read: The lock for the register set values is disabled.
When "1" is read: The lock for the register set values is enabled.</td></tr>
</table>

(Correct)
[bit24] LOCK: Lock Bit
You can write data to this bit only once. This register is used to prevent rewriting of the set values of various registers. When this bit is "0", you can rewrite the setting values of various registers. When this bit is set to "1" the watchdog counter is automatically cleared. This bit is equipped with a majority circuit with 3 FFs as a measure against a bit invert caused by the effects of noise or another factor.
<table>
<tr><th>LOCK</th><th>Description</th></tr>
<tr><td>During write operation</td><td>When "0" is written: Invalid
When "1" is written: The set values of the registers are locked.</td></tr>
<tr><td>During read operation</td><td>When "0" is read: The lock for the register set values is disabled.
When "1" is read: The lock for the register set values is enabled.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER15: Software Watchdog Timer
6. Register List
6.15. Software Watchdog Configuration Register (SWDG_CFG)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit20:16] OBSSEL[4:0]: Watchdog counter monitor bit output selection bits
These bits are used for selecting any 1 bit in the watchdog counter (32 bits) as the output of the watchdog counter monitor bit.(Support target of this function is different depends on product specification. For details, see "3 Explanation of Operation For this product, output of the watchdog counter monitor bit of MCU output pin is supported by the Software watchdog timer.")
<table>
<tr><th>bit4:0</th><th>Description</th></tr>
<tr><td>During write operation</td><td>When "0b00000" is written: Bit 0 is selected for the monitor output.
When "0b00001" is written: Bit 1 is selected for the monitor output.
When "0b00010" is written: Bit 2 is selected for the monitor output.
・
・
・
When "0b11111" is written: Bit 31 is selected for the monitor output.</td></tr>
<tr><td>During read operation</td><td>Set value read</td></tr>
</table>

(Correct)
[bit20:16] OBSSEL[4:0]: Watchdog Counter Monitor Bit Output Selection Bits
These bits are used for selecting any 1 bit in the watchdog counter (32 bits) as the output of the watchdog counter monitor bit.(Support target of this function is different depends on product specification. For details, see "3 Explanation of Operation For this product, output of the watchdog counter monitor bit of MCU output pin is supported by the Software watchdog timer.")
<table>
<tr><th>OBSSEL[4:0]</th><th>Description</th></tr>
<tr><td>During write operation</td><td>When 0b00000 is written: Bit 0 is selected for the monitor output.
When 0b00001 is written: Bit 1 is selected for the monitor output.
When 0b00010 is written: Bit 2 is selected for the monitor output.
・
・
・
When 0b11111 is written: Bit 31 is selected for the monitor output.</td></tr>
<tr><td>During read operation</td><td>Set value read</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER15: Software Watchdog Timer
6. Register List
6.15. Software Watchdog Configuration Register (SWDG_CFG)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit9:8] CLKSEL[1:0]: Clock selection bits
These bits are used to select the source clock for the watchdog counter. When activated, the watchdog counter starts its operation as the high-speed CR clock. For details on clock switching, see "7 Precautions for Using This Device Switching of watchdog counter source clock."
<table>
<tr><th>bit1:0</th><th>Description</th></tr>
<tr><td>During write operation</td><td>When "0b00" is written: The high-speed CR clock is selected.
When "0b01" is written: The low-speed CR clock is selected.
When "0b10" is written: The sub clock is selected.
When "0b11" is written: The main clock is selected.</td></tr>
<tr><td>During read operation</td><td>Set value read</td></tr>
</table>
[bit7:3] Reserved: Reserved bits

[bit2] ALLOWSTOPCLK: Clock stop for PSS enable bit
This bit is used to enable transition to PSS that involves the source clock stop of the watchdog counter. This bit is valid only when the WDENPSS bit in the Software watchdog configuration register (SWDG_CFG) is "1".
<table>
<tr><th>bit</th><th>Description</th></tr>
<tr><td>During write operation</td><td>When "0" is written: Watchdog clock stop in PSS is disabled.
When "1" is written: Watchdog clock stop in PSS is enabled.</td></tr>
<tr><td>During read operation</td><td>Set value read</td></tr>
</table>
(Correct)
[bit9:8] CLKSEL[1:0]: Clock Selection Bits
These bits are used to select the source clock for the watchdog counter. When activated, the watchdog counter starts its operation as the high-speed CR clock. For details on clock switching, see "7 Precautions for Using This Device Switching of watchdog counter source clock."
<table>
<tr><th>CLKSEL[1:0]</th><th>Description</th></tr>
<tr><td>During write operation</td><td>When 0b00 is written: The high-speed CR clock is selected.
When 0b01 is written: The low-speed CR clock is selected.
When 0b10 is written: The sub clock is selected.
When 0b11 is written: The main clock is selected.</td></tr>
<tr><td>During read operation</td><td>Set value read</td></tr>
</table>
[bit7:3] Reserved: Reserved bits

[bit2] ALLOWSTOPCLK: Clock Stop for PSS Enable Bit
This bit is used to enable transition to PSS that involves the source clock stop of the watchdog counter. This bit is valid only when the WDENPSS bit in the Software watchdog configuration register (SWDG_CFG) is "1".
<table>
<tr><th>ALLOWSTOPCLK</th><th>Description</th></tr>
<tr><td>During write operation</td><td>When "0" is written: Watchdog clock stop in PSS is disabled.
When "1" is written: Watchdog clock stop in PSS is enabled.</td></tr>
<tr><td>During read operation</td><td>Set value read</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER15: Software Watchdog Timer
6. Register List
6.15. Software Watchdog Configuration Register (SWDG_CFG)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit1] WDENPSS: Watchdog counter for PSS enable bit
This bit is used to enable the watchdog counter in PSS. This bit is enabled when the LOCK bit in the Software watchdog configuration register (SWDG_CFG) is set to "1".
This bit is equipped with a majority circuit with 3 FFs as a measure against a bit invert caused by the effects of noise or another factor.
<table>
<tr><th>bit</th><th>Description</th></tr>
<tr><td>During write operation</td><td>When "0" is written: The watchdog counter in PSS is disabled.
When "1" is written: The watchdog counter in PSS is enabled.</td></tr>
<tr><td>During read operation</td><td>Set value read</td></tr>
</table>
[bit0] WDENRUN: Watchdog counter for RUN enable bit
This bit is used to enable the watchdog counter in RUN. This bit is enabled when the LOCK bit in the Software watchdog configuration register (SWDG_CFG) is set to "1".
This bit is equipped with a majority circuit with 3 FFs as a measure against a bit invert caused by the effects of noise or another factor.
<table>
<tr><th>bit</th><th>Description</th></tr>
<tr><td>During write operation</td><td>When "0" is written: The watchdog counter in RUN is disabled.
When "1" is written: The watchdog counter in RUN is enabled.</td></tr>
<tr><td>During read operation</td><td>Set value read</td></tr>
</table>

(Correct)
[bit1] WDENPSS: Watchdog Counter for PSS Enable Bit
This bit is used to enable the watchdog counter in PSS. This bit is enabled when the LOCK bit in the Software watchdog configuration register (SWDG_CFG) is set to "1".
This bit is equipped with a majority circuit with 3 FFs as a measure against a bit invert caused by the effects of noise or another factor.
<table>
<tr><th>WDENPSS</th><th>Description</th></tr>
<tr><td>During write operation</td><td>When "0" is written: The watchdog counter in PSS is disabled.
When "1" is written: The watchdog counter in PSS is enabled.</td></tr>
<tr><td>During read operation</td><td>Set value read</td></tr>
</table>
[bit0] WDENRUN: Watchdog Counter for RUN Enable Bit
This bit is used to enable the watchdog counter in RUN. This bit is enabled when the LOCK bit in the Software watchdog configuration register (SWDG_CFG) is set to "1".
This bit is equipped with a majority circuit with 3 FFs as a measure against a bit invert caused by the effects of noise or another factor.
<table>
<tr><th>WDENRUN</th><th>Description</th></tr>
<tr><td>During write operation</td><td>When "0" is written: The watchdog counter in RUN is disabled.
When "1" is written: The watchdog counter in RUN is enabled.</td></tr>
<tr><td>During read operation</td><td>Set value read</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER15: Software Watchdog Timer
6. Register List
6.16. Software Watchdog Lower Limit RUN Current Register (SWDG_RUNLLC)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit31:0] WDGRUNLLC[31:0]: Window lower limit for RUN current bits
These bits are used to define the window lower limit value for RUN. The reading of these bits returns the initial value, until the Software watchdog configuration register (SWDG_CFG) LOCK bit is set.
<table>
<tr><th>bit31:0</th><th>Description</th></tr>
<tr><td>During write operation</td><td>Invalid</td></tr>
<tr><td>During read operation</td><td>Set value read</td></tr>
</table>
(Correct)
[bit31:0] WDGRUNLLC[31:0]: Window Lower Limit for RUN Current Bits
These bits are used to define the window lower limit value for RUN. The reading of these bits returns the initial value, until the Software watchdog configuration register (SWDG_CFG) LOCK bit is set.
<table>
<tr><th>WDGRUNLLC[31:0]</th><th>Description</th></tr>
<tr><td>During write operation</td><td>Invalid</td></tr>
<tr><td>During read operation</td><td>Set value read</td></tr>
</table></td></tr>
<tr><td>CHAPTER15: Software Watchdog Timer
6. Register List
6.17. Software Watchdog Upper Limit RUN Current Register (SWDG_RUNULC)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit31:0] WDGRUNULC[31:0]: Window upper limit for RUN current bits
These bits are used to define the window upper limit value for RUN. The reading of these bits returns the initial value, until the Software watchdog configuration register (SWDG_CFG) LOCK bit is set.
<table>
<tr><th>bit31:0</th><th>Description</th></tr>
<tr><td>During write operation</td><td>Invalid</td></tr>
<tr><td>During read operation</td><td>Set value read</td></tr>
</table>
(Correct)
[bit31:0] WDGRUNULC[31:0]: Window Upper Limit for RUN Current Bits
These bits are used to define the window upper limit value for RUN. The reading of these bits returns the initial value, until the Software watchdog configuration register (SWDG_CFG) LOCK bit is set.
<table>
<tr><th>WDGRUNULC[31:0]</th><th>Description</th></tr>
<tr><td>During write operation</td><td>Invalid</td></tr>
<tr><td>During read operation</td><td>Set value read</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER15: Software Watchdog Timer 6. Register List 6.18. Software Watchdog Lower Limit PSS Current Register (SWDG_PSSLLC)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit31:0] WDGPSSLLC[31:0]: Window lower limit for PSS current bits
These bits are used to define the window lower limit value for PSS. The reading of these bits returns the initial value, until the Software watchdog configuration register (SWDG_CFG) LOCK bit is set.
<table>
<tr><th>bit31:0</th><th>Description</th></tr>
<tr><td>During write operation</td><td>Invalid</td></tr>
<tr><td>During read operation</td><td>Set value read</td></tr>
</table>
(Correct)
[bit31:0] WDGPSSLLC[31:0]: Window Lower Limit for PSS Current Bits
These bits are used to define the window lower limit value for PSS. The reading of these bits returns the initial value, until the Software watchdog configuration register (SWDG_CFG) LOCK bit is set.
<table>
<tr><th>WDGPSSLLC[31:0]</th><th>Description</th></tr>
<tr><td>During write operation</td><td>Invalid</td></tr>
<tr><td>During read operation</td><td>Set value read</td></tr>
</table></td></tr>
<tr><td>CHAPTER15: Software Watchdog Timer 6. Register List 6.19. Software Watchdog Upper Limit PSS Current Register (SWDG_PSSULC)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit31:0] WDGPSSULC[31:0]: Window upper limit for PSS current bits
These bits are used to define the window upper limit value for PSS. The reading of these bits returns the initial value, until the Software watchdog configuration register (SWDG_CFG) LOCK bit is set.
<table>
<tr><th>bit31:0</th><th>Description</th></tr>
<tr><td>During write operation</td><td>Invalid</td></tr>
<tr><td>During read operation</td><td>Set value read</td></tr>
</table>
(Correct)
[bit31:0] WDGPSSULC[31:0]: Window Upper Limit for PSS Current Bits
These bits are used to define the window upper limit value for PSS. The reading of these bits returns the initial value, until the Software watchdog configuration register (SWDG_CFG) LOCK bit is set.
<table>
<tr><th>WDGPSSULC[31:0]</th><th>Description</th></tr>
<tr><td>During write operation</td><td>Invalid</td></tr>
<tr><td>During read operation</td><td>Set value read</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER16: TCRAM Interface</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
CHAPTER16: TCRAM INTERFACE
This chapter provides an overview of the interface between the Cortex(TM)-R5F BTCM port and SRAM, and describes their configuration, operation, and registers.

(Correct)
CHAPTER16: TCRAM Interface
This chapter provides an overview of the interface between the Cortex-R5F BTCM port and SRAM, and describes their configuration, operation, and registers.</td></tr>
<tr><td>CHAPTER16: TCRAM Interface
5. Registers
5.9. TCRAM IF TEST Error Address Register 0 (TRCFGn_TEAR0)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit31:29]TER: Diagnosis error source identification
<table>
<tr><th>TER</th><th>Diagnosis Error Source Identification Bits</th></tr>
<tr><td>During Write Operation</td><td>Writing data to these bits causes a bus error.</td></tr>
<tr><td>During Read Operation</td><td>Holds the diagnosis pattern when an error occurs during RAM diagnosis.
ERR_ADDR[14:0] is effective only when any of these bits is set to "1".
0b001: Error occurred in the march diagnosis
0b010: Error occurred in the checker diagnosis
0b100: Error occurred in the unique diagnosis
0b000: No error occurred
A RAM diagnosis start instruction triggers the hardware to initialize (clear to "000") these bits.</td></tr>
</table>

(Correct)
[bit31:29]TER: Diagnosis Error Source Identification
<table>
<tr><th>TER</th><th>Diagnosis Error Source Identification Bits</th></tr>
<tr><td>During Write Operation</td><td>Writing data to these bits causes a bus error.</td></tr>
<tr><td>During Read Operation</td><td>Holds the diagnosis pattern when an error occurs during RAM diagnosis.
ERR_ADDR[14:0] is effective only when any of these bits is set to "1".
0b001: Error occurred in the march diagnosis
0b010: Error occurred in the checker diagnosis
0b100: Error occurred in the unique diagnosis
0b000: No error occurred
A RAM diagnosis start instruction triggers the hardware to initialize (clear to 0b000) these bits.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER16: TCRAM Interface
5. Registers
5.9. TCRAM IF TEST Error Address Register 0 (TRCFGn_TEAR0)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit14:0] ERR_ADDR: Error occurrence address
<table><tr><th>ERR_ADDR</th><th>Error Occurrence Address Bit</th></tr><tr><td>During Write Operation</td><td>Writing data to these bits causes a bus error.</td></tr><tr><td>During Read Operation</td><td>Holds the address when an error occurs during RAM diagnosis. These bits indicate a valid value only when TER is not "000".
Triggered by the specification of RAM diagnosis start, the hardware initializes these bits (clears them to "0b000000000000000").</td></tr></table>
(Correct)
[bit14:0] ERR_ADDR: Error Occurrence Address
<table><tr><th>ERR_ADDR</th><th>Error Occurrence Address Bit</th></tr><tr><td>During Write Operation</td><td>Writing data to these bits causes a bus error.</td></tr><tr><td>During Read Operation</td><td>Holds the address when an error occurs during RAM diagnosis. These bits indicate a valid value only when TER is not 0b000.
Triggered by the specification of RAM diagnosis start, the hardware initializes these bits
(clears them to 0b0000000_00000000).</td></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER16: TCRAM Interface
5. Registers
5.10. TCRAM IF TEST Error Address Register 0 (TRCFGn_TEAR0)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit31:29]TER: Diagnosis error source identification
<table>
<tr><th>TER</th><th>Diagnosis Error Source Identification Bits</th></tr>
<tr><td>During Write Operation</td><td>Writing data to these bits causes a bus error.</td></tr>
<tr><td>During Read Operation</td><td>Holds the diagnosis pattern when an error occurs during RAM diagnosis.
ERR_ADDR[14:0] is effective only when any of these bits is set to "1".
0b001: Error occurred in the march diagnosis
0b010: Error occurred in the checker diagnosis
0b100: Error occurred in the unique diagnosis
0b000: No error occurred
A RAM diagnosis start instruction triggers the hardware to initialize (clear to "000") these bits.
Note:
When any of these bits becomes "1", even if an error occurs during another diagnosis item, the corresponding error source bit does not become "1".</td></tr>
</table>

(Correct)
[bit31:29]TER: Diagnosis Error Source Identification
<table>
<tr><th>TER</th><th>Diagnosis Error Source Identification Bits</th></tr>
<tr><td>During Write Operation</td><td>Writing data to these bits causes a bus error.</td></tr>
<tr><td>During Read Operation</td><td>Holds the diagnosis pattern when an error occurs during RAM diagnosis.
ERR_ADDR[14:0] is effective only when any of these bits is set to "1".
0b001: Error occurred in the march diagnosis
0b010: Error occurred in the checker diagnosis
0b100: Error occurred in the unique diagnosis
0b000: No error occurred
A RAM diagnosis start instruction triggers the hardware to initialize (clear to 0b000) these bits.
Note:
When any of these bits becomes "1", even if an error occurs during another diagnosis item, the corresponding error source bit does not become "1".</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER16: TCRAM Interface
5. Registers
5.10. TCRAM IF TEST Error Address Register 0 (TRCFGn_TEAR0)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit14:0] ERR_ADDR: Error occurrence address
<table>
<tr><th>ERR_ADDR</th><th>Error Occurrence Address Bit</th></tr>
<tr><td>During Write Operation</td><td>Writing data to these bits causes a bus error.</td></tr>
<tr><td>During Read Operation</td><td>Holds the address when an error occurs during RAM diagnosis. These bits are effective only when TER is not "000".
A RAM diagnosis start instruction triggers the hardware to initialize (clear to "0b0000000000000000") these bits.</td></tr>
</table>

(Correct)
[bit14:0] ERR_ADDR: Error Occurrence Address
<table>
<tr><th>ERR_ADDR</th><th>Error Occurrence Address Bit</th></tr>
<tr><td>During Write Operation</td><td>Writing data to these bits causes a bus error.</td></tr>
<tr><td>During Read Operation</td><td>Holds the address when an error occurs during RAM diagnosis. These bits are effective only when TER is not 0b000.
A RAM diagnosis start instruction triggers the hardware to initialize (clear to 0b0000000_00000000) these bits.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER16: TCRAM Interface
5. Registers
5.11. TCRAM IF TEST Error Address Register 2 (TRCFGn_TEAR2)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit31:29]TER: Diagnosis error source identification
<table>
<tr><th>TER</th><th>Diagnosis Error Source Identification Bits</th></tr>
<tr><td>During Write Operation</td><td>Writing data to these bits causes a bus error.</td></tr>
<tr><td>During Read Operation</td><td>Holds the diagnosis pattern when an error occurs during RAM diagnosis.
ERR_ADDR[14:0] is effective only when any of these bits is set to "1".
0b001: Error occurred in the march diagnosis
0b010: Error occurred in the checker diagnosis
0b100: Error occurred in the unique diagnosis
0b000: No error occurred
A RAM diagnosis start instruction triggers the hardware to initialize (clear to "000") these bits.
Note:
When any of these bits becomes "1", even if an error occurs during another diagnosis item, the corresponding error source bit does not become "1".</td></tr>
</table>

(Correct)
[bit31:29]TER: Diagnosis Error Source Identification
<table>
<tr><th>TER</th><th>Diagnosis Error Source Identification Bits</th></tr>
<tr><td>During Write Operation</td><td>Writing data to these bits causes a bus error.</td></tr>
<tr><td>During Read Operation</td><td>Holds the diagnosis pattern when an error occurs during RAM diagnosis.
ERR_ADDR[14:0] is effective only when any of these bits is set to "1".
0b001: Error occurred in the march diagnosis
0b010: Error occurred in the checker diagnosis
0b100: Error occurred in the unique diagnosis
0b000: No error occurred
A RAM diagnosis start instruction triggers the hardware to initialize (clear to 0b000) these bits.
Note:
When any of these bits becomes "1", even if an error occurs during another diagnosis item, the corresponding error source bit does not become "1".</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER16: TCRAM Interface
5. Registers
5.11. TCRAM IF TEST Error Address Register 2 (TRCFGn_TEAR2)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit14:0] ERR_ADDR: Error occurrence address
<table>
<tr><th>ERR_ADDR</th><th>Error Occurrence Address Bit</th></tr>
<tr><td>During Write Operation</td><td>Writing data to these bits causes a bus error.</td></tr>
<tr><td>During Read Operation</td><td>Holds the address when an error occurs during RAM diagnosis. These bits are effective only when TER is not "000".
A RAM diagnosis start instruction triggers the hardware to initialize (clear to "0b0000000000000000") these bits.</td></tr>
</table>

(Correct)
[bit14:0] ERR_ADDR: Error Occurrence Address
<table>
<tr><th>ERR_ADDR</th><th>Error Occurrence Address Bit</th></tr>
<tr><td>During Write Operation</td><td>Writing data to these bits causes a bus error.</td></tr>
<tr><td>During Read Operation</td><td>Holds the address when an error occurs during RAM diagnosis. These bits are effective only when TER is not 0b000.
A RAM diagnosis start instruction triggers the hardware to initialize (clear to 0b0b0000000_00000000) these bits.</td></tr>
</table>
</td></tr>
<tr><td>CHAPTER17: TCFLASH
2.Configuration
2.2.Address/Sector Map of TCFLASH</td><td>
Revised " Figure 2-4 Address/Sector Map of TCFLASH ” as follows.
Clerical error was corrected.
MB → KB

(Error)
▪ * 1 MB + 64 MB consists of small sector SA0-7 (8 KB x 8 = 64 KB) and large sector SA8-23 (64 KB x 16 = 1024 KB).
▪ The regions other than this are reserved regions.
▪
▪ * 2 MB + 64 MB consists of small sector SA0-7 (8 KB x 8 = 64 KB) and large sector SA8-39 (64 KB x 32 = 2048 KB).
▪ The regions other than this are reserved regions.
▪
▪ * 4 MB + 64 MB consists of small sector SA0-7 (8 KB x 8 = 64 KB) and large sector SA8-71 (64 KB x 64 = 4096 KB).
▪ The regions other than this are reserved regions.

(Correct)
▪ * 1 MB + 64 KB consists of small sector SA0-7 (8 KB x 8 = 64 KB) and large sector SA8-23 (64 KB x 16 = 1024 KB).
▪ The regions other than this are reserved regions.
▪
▪ * 2 MB + 64 KB consists of small sector SA0-7 (8 KB x 8 = 64 KB) and large sector SA8-39 (64 KB x 32 = 2048 KB).
▪ The regions other than this are reserved regions.
▪
▪ * 4 MB + 64 KB consists of small sector SA0-7 (8 KB x 8 = 64 KB) and large sector SA8-71 (64 KB x 64 = 4096 KB).
▪ The regions other than this are reserved regions.
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER17: TCFLASH
3. Explanation of Operation
3.8. ECC Generation and Check</td><td>Added item as follows

(Error)
TCM space is performed in the Cortex-R5F core.
The contents described from 3.8.1 to 3.8.3 are the operations performed in TCFLASH when it is read-accessed via the AXI space.
To test the check function, the ECC logic has the function to insert an error into the data and ECC that are read from flash memory. Error injection is performed during read-access both via the TCM space and via the AXI space.
ECC is enabled/disabled by setting the ECCOFF bit in the TCFCFG_FECCCTRL register.

(Correct)
TCM space is performed in the Cortex-R5F core.
The ECC movement in TCM port is based on ECC setting inside the CPU.
The contents described from 3.8.1 to 3.8.3 are the operations performed in TCFLASH when it is read-accessed via the AXI space.
To test the check function, the ECC logic has the function to insert an error into the data and ECC that are read from flash memory. Error injection is performed during read-access both via the TCM space and via the AXI space.
ECC is enabled/disabled by setting the ECCOFF bit in the TCFCFG_FECCCTRL register.</td></tr>
<tr><td>CHAPTER17: TCFLASH
5. Registers</td><td>Added item as follows

(Error)
<table>
<tr><td>0x090</td><td>TCFCFG_BRCFG
00000000_00010000_00000000_11111111</td><td>TCM buffer region configuration register</td></tr>
<tr><td>0x098</td><td>TCFCFG_BRAT
00000000_00000000_00000000_00000001</td><td>TCM buffer region attribute register</td></tr>
<tr><td>0x09C to 0x0FC</td><td>-</td><td>Reserved area</td></tr>
</table>

(Correct)
<table>
<tr><td>0x090</td><td>TCFCFG_BRCFG
00000000_00010000_00000000_11111111</td><td>TCM buffer region configuration register</td></tr>
<tr><td>0x094</td><td>-</td><td>Reserved area</td></tr>
<tr><td>0x098</td><td>TCFCFG_BRAT
00000000_00000000_00000000_00000001</td><td>TCM buffer region attribute register</td></tr>
<tr><td>0x09C to 0x0FC</td><td>-</td><td>Reserved area</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER19: BootROM Hardware Interface 3.Explanation of Operation</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
ARM® CortexTM-R5 exception vector processing

(Correct)
ARM Cortex-R5 Exception Vector Processing</td></tr>
<tr><td>CHAPTER19: BootROM Hardware Interface 5.Registers</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
ROM is FFFF_0000.

(Correct)
ROM is 0xFFFF_0000.</td></tr>
<tr><td>CHAPTER19: BootROM Hardware Interface 5.Registers 5.1.EXCFG Lock Release Register (EXCFG_UNLOCK)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit31:0] UNLOCK: Lock release bits of BootROM hardware interface
These bits control the write lock of the setting registers of the BootROM hardware interface. All registers except EXCFG_UNLOCK are subject to sequence protection by this register.
<table>
<tr><th>bit31:0</th><th>Description</th></tr>
<tr><td>0xACC5B007</td><td>Unlock value (which enables writing)</td></tr>
<tr><td>0xB007ECF6</td><td>Lock value (which disables writing)</td></tr>
<tr><td>Other than above</td><td>Setting prohibited (bus error returned)</td></tr>
</table>
(Correct)
[bit31:0] UNLOCK: Lock Release Bits of BootROM Hardware Interface
These bits control the write lock of the setting registers of the BootROM hardware interface. All registers except EXCFG_UNLOCK are subject to sequence protection by this register.
<table>
<tr><th>UNLOCK</th><th>Description</th></tr>
<tr><td>0xACC5B007</td><td>Unlock value (which enables writing)</td></tr>
<tr><td>0xB007ECF6</td><td>Lock value (which disables writing)</td></tr>
<tr><td>Other than above</td><td>Setting prohibited (bus error returned)</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER21: Exclusive Access Memory (EAM) 3.Explanation of Operation</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
The letter "x" in LDRx, STRx, LDREXx, STREXx, and so on in the commands described below represents the size of the command: "D" = Doubleword, "" = Word, "H" = Halfword, and "B" = Byte. (There is not suffix of "x" in the Word case.)

(Correct)
The letter "x" in LDRx, STRx, LDREXx, STREXx, and so on in the commands described below represents the size of the command: "D" = Doubleword, "W" = Word, "H" = Halfword, and "B" = Byte. (There is not suffix of "x" in the Word case.)</td></tr>
<tr><td>CHAPTER23: Time Protection 5. Registers 5.1.TPU Lock Release Register (TPU0_UNLOCK)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit31:0] UNLOCK: Lock release bits of the time protection unit
These bits control the write lock of the configuration registers of the time protection unit.
TPU0_CFG and TPU0_TCN10 to TPU0_TCN17 are subject to sequence protection by this register.
<table>
<tr><th>bit31:0</th><th>Description</th></tr>
<tr><td>0xACC5A110</td><td>Unlock value (which enables writing)</td></tr>
<tr><td>0xB10CACC5</td><td>Lock value (which disables writing)</td></tr>
<tr><td>Other than above</td><td>Setting prohibited (bus error returned)</td></tr>
</table>
(Correct)
[bit31:0] UNLOCK: Lock Release Bits of the Time Protection Unit
These bits control the write lock of the configuration registers of the time protection unit.
TPU0_CFG and TPU0_TCN10 to TPU0_TCN17 are subject to sequence protection by this register.
<table>
<tr><th>UNLOCK</th><th>Description</th></tr>
<tr><td>0xACC5A110</td><td>Unlock value (which enables writing)</td></tr>
<tr><td>0xB10CACC5</td><td>Lock value (which disables writing)</td></tr>
<tr><td>Other than above</td><td>Setting prohibited (bus error returned)</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER23: Time Protection
5. Registers
5.3.TPU Configuration Register (TPU0_CFG)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit21:16] GLBPS: Global prescaler division setting bit
This bit sets the division ratio of the global prescaler. The input system clock is divided according to the configured division ratio, generating a clock of a lower frequency. A divided clock is provided to each timer.
<table>
<tr><th colspan="6">bit21:16</th><th>Description</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1/1</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1/2</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1/3</td></tr>
<tr><td colspan="6">-
-
-</td><td>-
-
-</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1/64</td></tr>
</table>

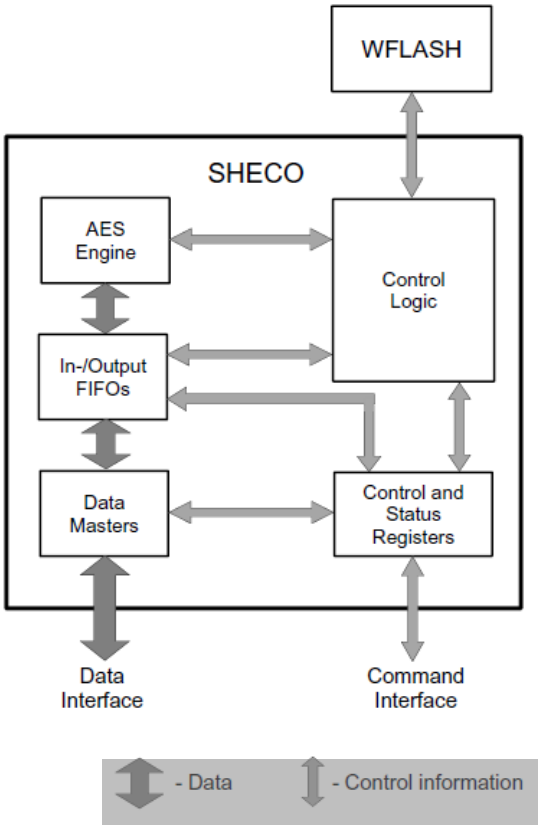
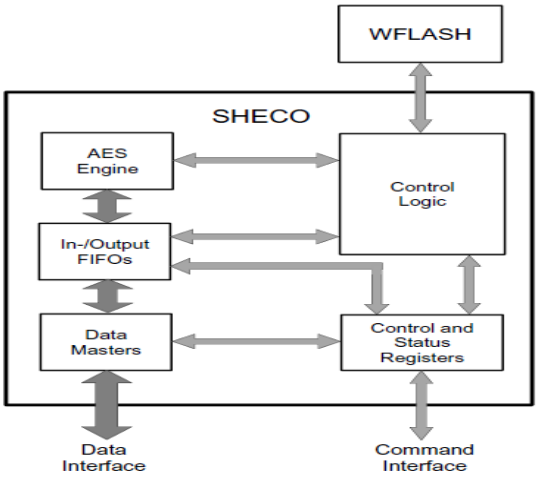
(Correct)
[bit21:16] GLBPS: Global Prescaler Division Setting Bit
This bit sets the division ratio of the global prescaler. The input system clock is divided according to the configured division ratio, generating a clock of a lower frequency. A divided clock is provided to each timer.
<table>
<tr><th colspan="6">Bit</th><th>Description</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1/1</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1/2</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1/3</td></tr>
<tr><td colspan="6">-
-
-</td><td>-
-
-</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1/64</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER23: Time Protection
5. Registers
5.8. TPU Timer m Control Register 1 (TPU0_TCN1m)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit1:0] PS: Individual prescaler division setting bit
This bit sets the division ratio of the individual prescaler. An individual prescaler takes the output clock of the global prescaler, and divides it according to the configured division ratio to generate a clock of a timer-specific frequency.
<table>
<tr><th colspan="2">bit1:0</th><th>Description</th></tr>
<tr><td>0</td><td>0</td><td>Divided by 1</td></tr>
<tr><td>0</td><td>1</td><td>Divided by 2</td></tr>
<tr><td>1</td><td>0</td><td>Divided by 4</td></tr>
<tr><td>1</td><td>1</td><td>Divided by 16</td></tr>
</table>
Notes:
−For details on timer operation modes, the preload function, and the free-run function, see "3Explanation of Operation."
−TPU0_TCN10 to TPU0_TCN17 share the same register bit configuration.
−It is prohibition that carries out timer starting by TPU0_CFG:GLBPS="0b000000" and TPU0_TCN1m:PS="0b00."

(Correct)
[bit1:0] PS: Individual Prescaler Division Setting Bit
This bit sets the division ratio of the individual prescaler. An individual prescaler takes the output clock of the global prescaler, and divides it according to the configured division ratio to generate a clock of a timer-specific frequency.
<table>
<tr><th colspan="2">Bits</th><th>Description</th></tr>
<tr><td>0</td><td>0</td><td>Divided by 1</td></tr>
<tr><td>0</td><td>1</td><td>Divided by 2</td></tr>
<tr><td>1</td><td>0</td><td>Divided by 4</td></tr>
<tr><td>1</td><td>1</td><td>Divided by 16</td></tr>
</table>
Notes:
−For details on timer operation modes, the preload function, and the free-run function, see "3Explanation of Operation."
−TPU0_TCN10 to TPU0_TCN17 share the same register bit configuration.
−It is prohibition that carries out timer starting by TPU0_CFG:GLBPS= 0b000000 and TPU0_TCN1m:PS= 0b00.
</td></tr>
<tr><td>CHAPTER25: Memory Protection Unit for AXI of SHE</td><td>A chapter of "Memory Protection Unit for AXI of SHE" is added newly by the 2nd edition.</td></tr>
</table>

Section	Change Results
CHAPTER26: Secure Hardware Extension (SHE) 1.Outline of the Secure Hardware Extension (SHE) Module	Features of "For details" should be corrected as indicated by the shading below. (Error) Table 1-1 Block diagram of the SHE module (Correct) Figure 1-1 Block Diagram of the SHE Module

Section	Change Results
CHAPTER26: Secure Hardware Extension (SHE) 1.Outline of the Secure Hardware Extension (SHE) Module	Revised " Figure 1-1 Block Diagram of the SHE Module" as follows. (Error)  (Correct) 

Section	Change Results
CHAPTER26: Secure Hardware Extension (SHE) 3.Operation of the Secure Hardware Extension 3.1.Operation of the Command Interface	Features of "For details" should be corrected as indicated by the shading below. (Error) Sequential command execution The processing of a sequential SHE command is presented in Table 3-1 and is described in the following. Table 3-1 Flow chart for the execution of a sequential SHE command (Correct) Sequential command execution The processing of a sequential SHE command is presented in Figure 1-1 and is described in the following. Figure 3-1 Flow Chart for the Execution of a Sequential SHE Command
CHAPTER26: Secure Hardware Extension (SHE) 3.Operation of the Secure Hardware Extension 3.1.Operation of the Command Interface	Features of "For details" should be corrected as indicated by the shading below. (Error) A sequential SHE command can only be started if SHE is in idle mode, i.e. the BUSY bit in the SHE status register (SHE_STATUS) is low. In order to start the execution of the required SHE command, the user has to write the command opcode into the command register (see Section 2.1.1 SHE Command Register (SHE_CMD) description). The commands are defined in the SHE Functional Specification (Chapter 7) and listed in Table 3-2 below. (Correct) A sequential SHE command can only be started if SHE is in idle mode, i.e. the BUSY bit in the SHE status register (SHE_STATUS) is low. In order to start the execution of the required SHE command, the user has to write the command opcode into the command register (see Section 2.1.1 SHE Command Register (SHE_CMD) description). The commands are defined in the SHE Functional Specification (Chapter 7) and listed in Table 3-1 below.
CHAPTER26: Secure Hardware Extension (SHE) 3.Operation of the Secure Hardware Extension 3.1.Operation of the Command Interface	Features of "For details" should be corrected as indicated by the shading below. (Error) Table 3-2 Sequential SHE commands started through the SHE_CMD register (Correct) Table 3-1 Sequential SHE Commands Started Through the SHE_CMD Register

Section	Change Results
CHAPTER26: Secure Hardware Extension (SHE) 3.Operation of the Secure Hardware Extension 3.1.Operation of the Command Interface	Features of "For details" should be corrected as indicated by the shading below. (Error) Table 3-3 Sequential order for parameter and result transfer (Correct) Table 3-2 Sequential Order for Parameter and Result Transfer
CHAPTER26: Secure Hardware Extension (SHE) 3.Operation of the Secure Hardware Extension 3.1.Operation of the Command Interface	Features of "For details" should be corrected as indicated by the shading below. (Error) Table 3-4 Timing diagram for execution of a sequential SHE command (Correct) Figure 3-2 Timing Diagram for Execution of a Sequential SHE Command
CHAPTER26: Secure Hardware Extension (SHE) 3.Operation of the Secure Hardware Extension 3.1.Operation of the Command Interface	Features of "For details" should be corrected as indicated by the shading below. (Error) Table 3-5 illustrates the timing flow for the execution of the CMD_CANCEL command. Table 3-5 Execution of CMD_CANCEL command (Correct) Figure 3-3 illustrates the timing flow for the execution of the CMD_CANCEL command. Figure 3-3 Execution of CMD_CANCEL Command
CHAPTER26: Secure Hardware Extension (SHE) 3.Operation of the Secure Hardware Extension 3.1.Operation of the Command Interface	Features of "For details" should be corrected as indicated by the shading below. (Error) Table 3-6 Execution of CMD_CANCEL command issued after the previous command has completed (Correct) Figure 3-4 Execution of CMD_CANCEL Command Issued after the Previous Command Has Completed

Section	Change Results
CHAPTER26: Secure Hardware Extension (SHE) 3.Operation of the Secure Hardware Extension 3.1.Operation of the Command Interface	Features of "For details" should be corrected as indicated by the shading below. (Error) Table 3-7 Execution of CMD_CANCEL command when no command execution is ongoing. (Correct) Figure 3-5 Execution of CMD_CANCEL Command When No Command Execution Is Ongoing.
CHAPTER26: Secure Hardware Extension (SHE) 3.Operation of the Secure Hardware Extension 3.1.Operation of the Command Interface	Features of "For details" should be corrected as indicated by the shading below. (Error) Table 3-8 Block diagram of SHE interrupt generation circuitry (Correct) Figure 3-6 Block Diagram of SHE Interrupt Generation Circuitry
CHAPTER26: Secure Hardware Extension (SHE) 3.Operation of the Secure Hardware Extension 3.1.Operation of the Command Interface 3.1.1.Power Saving Functionality	Features of "For details" should be corrected as indicated by the shading below. (Error) Table 3-9 SHE clock disabling/enabling (Correct) Figure 3-7 SHE Clock Disabling/Enabling

Section	Change Results
CHAPTER26: Secure Hardware Extension (SHE) 3.Operation of the Secure Hardware Extension 3.2.Operation of the data interface	Features of "For details" should be corrected as indicated by the shading below. (Error) Figure 3-1 SHE data flow diagram (Correct) Figure 3-8 SHE Data Flow Diagram
CHAPTER26: Secure Hardware Extension (SHE) 3.Operation of the Secure Hardware Extension 3.2.Operation of the data interface	Features of "For details" should be corrected as indicated by the shading below. (Error) Table 3-10 Lockable Registers List (Correct) Table 3-3 Lockable Registers List
CHAPTER26: Secure Hardware Extension (SHE) 3.Operation of the Secure Hardware Extension 3.3.Operation of the FIFO Unit	Features of "For details" should be corrected as indicated by the shading below. (Error) Table 3-11 FIFO data ordering (Correct) Figure 3-9 FIFO Data Ordering
CHAPTER26: Secure Hardware Extension (SHE) 3.Operation of the Secure Hardware Extension 3.3.Operation of the FIFO Unit	Features of "For details" should be corrected as indicated by the shading below. (Error) Table 3-12 FIFO data ordering for parameters wider than 64 bits (Correct) Figure 3-10 FIFO Data Ordering for Parameters Wider than 64 Bits

Section	Change Results
CHAPTER26: Secure Hardware Extension (SHE) 5.Enhanced Secure Hardware Extension 5.1.Additional generate purpose keys	Features of "For details" should be corrected as indicated by the shading below. (Error) Table 4-3 KEY_<n> (n = 11..20) addressing (Correct) Table 5-1 KEY_<n> (n = 11..20) Addressing
CHAPTER26: Secure Hardware Extension (SHE) 5.Enhanced Secure Hardware Extension 5.3.Additional User-accessible Functions	Features of "For details" should be corrected as indicated by the shading below. (Error) Table 4-4 ESHE commands (Correct) Table 5-2 ESHE Commands
CHAPTER26: Secure Hardware Extension (SHE) 5.Enhanced Secure Hardware Extension 5.3.Additional User-accessible Functions	Features of "For details" should be corrected as indicated by the shading below. (Error) The order in which parameters and data for processing are expected as well as the order in which results are generated is shown in Table 4-5. Table 4-5 Parameter order for ESHE commands (Correct) The order in which parameters and data for processing are expected as well as the order in which results are generated is shown in Table 5-4. Table 5-3 Parameter Order for ESHE Commands

Section	Change Results
CHAPTER26: Secure Hardware Extension (SHE) 5.Enhanced Secure Hardware Extension 5.3.Additional User-accessible Functions 5.3.1.CMD_LOAD_KEY_ESHE	Features of "For details" should be corrected as indicated by the shading below. (Error) Table 4-6 Parameter order for ESHE commands (Correct) Table 5-4 Parameter Order for ESHE Commands

Section	Change Results
CHAPTER27: DMA Controller 3.Operational Description	Features of "For details" should be corrected as indicated by the shading below. (Error) ■DMAi_R:DBE = '1', DMAi_R:DB[1:0] = '10' (stop on debug event), and a 'debug event' is pending ■DMAi_R:DBE = '1', DMAi_R:DB[1:0] = '01' (halt on debug event), and a 'debug event' is pending If any of the above conditions is true, DMA Stop/Halt Request Flag is '1' indicating that DMA transfers of all channels are requested to halt or stop. The condition that all channels are halted or have come to a stop after a global halt or stop request is indicated by DMA Stop/Halt Status Flag (DMAi_R:DSHS). DMA Stop/Halt Status Flag is '0' if one or more channels are not yet stopped or halted and '1' if all channels are stopped or halted DMAC debug behavior The DMA Controller can be configured to react in a predefined way on a 'debug event' (e.g. debugger break point). The feature to react on a 'debug event' can be enabled with bit Debug Enable (DMAi_R:DBE). If enabled the DMACs behavior depend on the setting of Debug Behavior (DMAi_R:DB[1:0]). This feature is disabled after reset. The behavior can be configured to stop all transfers (DMAi_R:DB[1:0] = '10'), or to halt all transfers (DMAi_R:DB[1:0] = '01'), or just to continue operation independent of debug events (DMAi_R:DB[1:0] = '00'). Initial value is to continue operation independent of debug events. (Correct) ■DMAi_R:DBE = 1, DMAi_R:DB[1:0] = 0b10 (stop on debug event), and a 'debug event' is pending ■DMAi_R:DBE = 1, DMAi_R:DB[1:0] = 0b01 (halt on debug event), and a 'debug event' is pending If any of the above conditions is true, DMA Stop/Halt Request Flag is "1" indicating that DMA transfers of all channels are requested to halt or stop. The condition that all channels are halted or have come to a stop after a global halt or stop request is indicated by DMA Stop/Halt Status Flag (DMAi_R:DSHS). DMA Stop/Halt Status Flag is "0" if one or more channels are not yet stopped or halted and "1" if all channels are stopped or halted DMAC Debug Behavior The DMA Controller can be configured to react in a predefined way on a 'debug event' (e.g. debugger break point). The feature to react on a 'debug event' can be enabled with bit Debug Enable (DMAi_R:DBE). If enabled the DMACs behavior depend on the setting of Debug Behavior (DMAi_R:DB[1:0]). This feature is disabled after reset. The behavior can be configured to stop all transfers (DMAi_R:DB[1:0] = 0b10), or to halt all transfers (DMAi_R:DB[1:0] = 0b01), or just to continue operation independent of debug events (DMAi_R:DB[1:0] = 0b00). Initial value is to continue operation independent of debug events.

Section	Change Results
CHAPTER27: DMA Controller 3.Operational Description 3.1.DMA Channels	Features of "For details" should be corrected as indicated by the shading below. (Error) If a DMA transfer is completed successfully and the completion interrupt raised, Stop Status will show 'normal end' (DMAi_Bn:SS = '101'). If it is ended in error and the error interrupt raised, Stop Status will show one of the following possibilities: ■Stop request (DMAi_Bn:SS = '010') ■Source access error (DMAi_Bn:SS = '011') ■Destination access error (DMAi_Bn:SS = '100') (Correct) If a DMA transfer is completed successfully and the completion interrupt raised, Stop Status will show 'normal end' (DMAi_Bn:SS = 0b101). If it is ended in error and the error interrupt raised, Stop Status will show one of the following possibilities: ■Stop request (DMAi_Bn:SS = 0b010) ■Source access error (DMAi_Bn:SS = 0b011) ■Destination access error (DMAi_Bn:SS = 0b100)
CHAPTER27: DMA Controller 3.Operational Description 3.1.DMA Channels	Features of "For details" should be corrected as indicated by the shading below. (Error) Both interrupts, completion as well as error interrupt can be masked with bits Completion Interrupt (DMAi_Bn:CI) and Error Interrupt (DMAi_Bn:EI) respective. If these bits are set to '1' the interrupts are not masked. All unmasked completion interrupts are ORed and signaled to the Interrupt Controller. The same is done for the error interrupts. All completion Interrupt Flags are in addition to the channel registers, available in two 32-bit registers (DMAi_DIRQ1 and DMAi_DIRQ2) for easier software handling. All Error Interrupts are handled in the same way and are available in register DMAi_EDIRQ1 and DMAi_EDIRQ2. Completion interrupt DMAi_Bn:DQ must be cleared by setting Clear DIRQ (DMAi_Cn:CD). Error interrupt EQ must be cleared by setting Clear EDIRQ (DMAi_Cn:CE). Stop Status will be cleared to 'initial value' (DMAi_Bn:SS = '000') if DMAi_Bn:DQ or DMAi_Bn:EQ is set to '1'. (Correct) Both interrupts, completion as well as error interrupt can be masked with bits Completion Interrupt (DMAi_Bn:CI) and Error Interrupt (DMAi_Bn:EI) respective. If these bits are set to "1" the interrupts are not masked. All unmasked completion interrupts are ORed and signaled to the Interrupt Controller. The same is done for the error interrupts. All completion Interrupt Flags are in addition to the channel registers, available in two 32-bit registers (DMAi_DIRQ1 and DMAi_DIRQ2) for easier software handling. All Error Interrupts are handled in the same way and are available in register DMAi_EDIRQ1 and DMAi_EDIRQ2. Completion interrupt DMAi_Bn:DQ must be cleared by setting Clear DIRQ (DMAi_Cn:CD). Error interrupt EQ must be cleared by setting Clear EDIRQ (DMAi_Cn:CE). Stop Status will be cleared to 'initial value' (DMAi_Bn:SS = 0b000) if DMAi_Bn:DQ or DMAi_Bn:EQ is set to "1".

Section	Change Results
CHAPTER27: DMA Controller 3.Operational Description 3.3.DMA Arbiter	The item should be added as indicated by the shading below. (Error) The arbitration schemes are explained in detail in the following sections. The arbitration scheme can be changed any time, however it becomes effective only after the current running data transfer has been completed at the next transfer gap. (Correct) The arbitration schemes are explained in detail in the following sections. The arbitration scheme can be changed any time, however it becomes effective only after the current running data transfer has been completed at the next transfer gap. About the timing of the priority judgement, it is judged after the end of transferring 1 block, and before the start of transferring the next block.

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER27:
DMA Controller
4. Registers</td><td>Features of "For details" should be corrected as indicated by the shading below.
(Error)
<table>
<tr><td>0x0000_1018(DMAC#0)
0x0000_5018(DMAC#1)
~
0x0000_201F(DMAC#0)
0x0000_601F(DMAC#1)</td><td>Reserved</td></tr>
<tr><td>0x0000_2020 (DMAC#0)
0x0000_6020(DMAC#1)
+ ((m-8)*0x04)</td><td>DMAi_CMICICm
00000000_00000000_00000000_00000000</td></tr>
<tr><td>0x0000_223C(DMAC#0)
0x0000_623C(DMAC#1)
~
0x0000_27FF(DMAC#0)
0x0000_67FF(DMAC#1)</td><td>Reserved</td></tr>
<tr><td>0x0000_2800(DMAC#0)
0x0000_6800(DMAC#1)
+ (n*0x04)</td><td>DMAi_CMCHICn
00000000_00000000_00000000_0000(*1)(*1)(*1)(*1)</td></tr>
<tr><td>0x0000_2840(DMAC#0)
0x0000_6840(DMAC#1)
~</td><td>Reserved</td></tr>
</table>
(Correct)
<table>
<tr><td>0x0000_1018(DMAC#0)
0x0000_5018(DMAC#1)
to
0x0000_201F(DMAC#0)
0x0000_601F(DMAC#1)</td><td>Reserved</td></tr>
<tr><td>0x0000_2020 (DMAC#0)
0x0000_6020(DMAC#1)
The reserve area is changed by the following arithmetic expression.
+ ((m-8)*0x04)</td><td>DMAi_CMICICm
00000000_00000000_00000000_00000000</td></tr>
<tr><td>0x0000_223C(DMAC#0)
0x0000_623C(DMAC#1)
to
0x0000_27FF(DMAC#0)
0x0000_67FF(DMAC#1)</td><td>Reserved</td></tr>
<tr><td>0x0000_2800(DMAC#0)
0x0000_6800(DMAC#1)
The reserve area is changed by the following arithmetic expression.
+ (n*0x04)</td><td>DMAi_CMCHICn
00000000_00000000_00000000_0000*1</td></tr>
<tr><td>0x0000_2840(DMAC#0)
0x0000_6840(DMAC#1)
.....</td><td>Reserved</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER27: DMA Controller
4. Registers
4.1. DMA Controller Global Configuration Register (DMAi_R)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit30] DSHR : DMA Stop/Halt Request Flag
This bit indicates that the DMA transfers of all channels have been requested to halt or disable.
This bit is set to '0' by hardware if none of the conditions below are true.
This bit is set to '1' by hardware if one or more of the conditions below are true.
Conditions for DMA Stop/Halt Request Flag:
■DMAi_R:DE is set to '0' (all channels are disabled)
■DMAi_R:DH is set to '1' (all channels are halted)
■DMAi_R:DBE is set to '1', DMAi_R:DB is set to '10' (stop on debug events), and a debug event is pending
■DMAi_R:DBE is set to '1', DMAi_R:DB is set to '01' (halt on debug events), and a debug event is pending
<table>
<tr><th>DSHR</th><th>Description</th></tr>
<tr><td>0</td><td>Indicates that global halt/disable condition of DMAC is removed</td></tr>
<tr><td>1</td><td>Indicates that DMA transfers of all channels are requested to halt or disable</td></tr>
</table>

(Correct)
[bit30] DSHR : DMA stop/halt request flag
This bit indicates that the DMA transfers of all channels have been requested to halt or disable.
This bit is set to "0" by hardware if none of the conditions below are true.
This bit is set to "1" by hardware if one or more of the conditions below are true.
Conditions for DMA Stop/Halt Request Flag:
■DMAi_R:DE is set to "0" (all channels are disabled)
■DMAi_R:DH is set to "1" (all channels are halted)
■DMAi_R:DBE is set to "1", DMAi_R:DB is set to '10' (stop on debug events), and a debug event is pending
■DMAi_R:DBE is set to "1", DMAi_R:DB is set to '01' (halt on debug events), and a debug event is pending
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Indicates that global halt/disable condition of DMAC is removed</td></tr>
<tr><td>1</td><td>Indicates that DMA transfers of all channels are requested to halt or disable</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER27: DMA Controller
4. Registers
4.1. DMA Controller Global Configuration Register (DMAi_R)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit29] DBE : Debug Enable
This bit determines whether DMAC reacts on a debug event (i.e. debugger break point).
The behavior of the DMAC on the occurrence of a debug event depends on configuration bits, Debug Behavior (DMAi_R:DB).
<table>
<tr><th>DBE</th><th>Description</th></tr>
<tr><td>0</td><td>DMAC does not react on debug events</td></tr>
<tr><td>1</td><td>DMAC reacts on debug events. Reaction depends on setting of Debug Behavior (DMAi_R:DB)</td></tr>
</table>

(Correct)
[bit29] DBE : Debug enable
This bit determines whether DMAC reacts on a debug event (i.e. debugger break point).
The behavior of the DMAC on the occurrence of a debug event depends on configuration bits, Debug Behavior (DMAi_R:DB).
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>DMAC does not react on debug events</td></tr>
<tr><td>1</td><td>DMAC reacts on debug events. Reaction depends on setting of Debug Behavior (DMAi_R:DB)</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER27: DMA Controller
4. Registers
4.1. DMA Controller Global Configuration Register (DMAi_R)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit28:27] PR[1:0] : Priority Type
These bits select the arbitration scheme of the DMAC arbiter. In case of dynamic priority, channel priority is updated at the transfer gap.
<table>
<tr><th>PR[1:0]</th><th>Description</th></tr>
<tr><td>00</td><td>Fixed priority</td></tr>
<tr><td>01</td><td>Dynamic priority</td></tr>
<tr><td>10</td><td>Round robin</td></tr>
<tr><td>11</td><td>Reserved</td></tr>
</table>

(Correct)
[bit28:27] PR[1:0] : Priority type
These bits select the arbitration scheme of the DMAC arbiter. In case of dynamic priority, channel priority is updated at the transfer gap.
<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>00</td><td>Fixed priority</td></tr>
<tr><td>01</td><td>Dynamic priority</td></tr>
<tr><td>10</td><td>Round robin</td></tr>
<tr><td>11</td><td>Reserved</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER27: DMA Controller
4. Registers
4.1. DMA Controller Global Configuration Register (DMAi_R)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit26] DH : DMA Halt
When this bit is set to a '1', all DMA channels are halted and do not perform DMA transfers until this bit is set back to '0'. After it is cleared the halted DMA transfers continue at the point they were halted.
If this bit is set to '1' while a DMA transfer is ongoing, DMAC halts the transfer at the next transfer gap.
About the transfer gap, refer to the description of DMAi_R:DE bit.

[bit25:24] DB[1:0] : Debug Behavior
<table>
<tr><th>DB[1:0]</th><th>Description</th></tr>
<tr><td>00</td><td>DMAC continues on debug event</td></tr>
<tr><td>01</td><td>DMAC halts all transfers on debug event</td></tr>
<tr><td>10</td><td>DMAC stops all transfers on debug event</td></tr>
<tr><td>11</td><td>Reserved</td></tr>
</table>

(Correct)
[bit26] DH : DMA halt
When this bit is set to a "1", all DMA channels are halted and do not perform DMA transfers until this bit is set back to "0". After it is cleared the halted DMA transfers continue at the point they were halted.
If this bit is set to "1" while a DMA transfer is ongoing, DMAC halts the transfer at the next transfer gap.
About the transfer gap, refer to the description of DMAi_R:DE bit.

[bit25:24] DB[1:0] : Debug behavior
<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>00</td><td>DMAC continues on debug event</td></tr>
<tr><td>01</td><td>DMAC halts all transfers on debug event</td></tr>
<tr><td>10</td><td>DMAC stops all transfers on debug event</td></tr>
<tr><td>11</td><td>Reserved</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER27: DMA Controller
4. Registers
4.1. DMA Controller Global Configuration Register (DMAi_R)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit0] DSHS : DMA Stop/Halt Status Flag
<table>
<tr><th>DSHS</th><th>Description</th></tr>
<tr><td>0</td><td>Indicate that DMA transfer of at least one channel is running</td></tr>
<tr><td>1</td><td>Indicate that DMA transfers of all channels are halted or disabled</td></tr>
</table>
(Correct)
[bit0] DSHS : DMA stop/halt status flag
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Indicate that DMA transfer of at least one channel is running</td></tr>
<tr><td>1</td><td>Indicate that DMA transfers of all channels are halted or disabled</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER27: DMA Controller
4. Registers
4.7. DMA Controller Channel Configuration A Register Channel 'n' (DMAi_An)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit31] EB : Enable Bit
This bit is used to enable/disable a DMA channel. If this bit is set to '1', the channel is enabled and waits for a request to start a DMA transfer (Before that, DMAi_R:DE bit needs to be set to '1' already). If this bit is set to '0', the channel is disabled and does not perform a DMA transfer. When this bit is set to '0' during a running DMA transfer which will not complete at the next transfer gap, the DMA transfer is terminated. This is regarded as a forced stop and an error interrupt is generated. When this bit is set to '0' while the last block of a DMA transfer is running, the DMA transfer completes at the next transfer gap and a completion interrupt is generated. About the transfer gap, refer to the description of DMAi_R:DE bit. This bit is useful to re-configure each configuration register of the channel after a DMA transfer.
<table>
<tr><th>EB</th><th>Description</th></tr>
<tr><td>0</td><td>Channel is disabled</td></tr>
<tr><td>1</td><td>Channel is enabled</td></tr>
</table>

(Correct)
[bit31] EB : Enable bit
This bit is used to enable/disable a DMA channel. If this bit is set to "1", the channel is enabled and waits for a request to start a DMA transfer (Before that, DMAi_R:DE bit needs to be set to "1" already). If this bit is set to "0", the channel is disabled and does not perform a DMA transfer. When this bit is set to "0" during a running DMA transfer which will not complete at the next transfer gap, the DMA transfer is terminated. This is regarded as a forced stop and an error interrupt is generated. When this bit is set to "0" while the last block of a DMA transfer is running, the DMA transfer completes at the next transfer gap and a completion interrupt is generated. About the transfer gap, refer to the description of DMAi_R:DE bit. This bit is useful to re-configure each configuration register of the channel after a DMA transfer.
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Channel is disabled</td></tr>
<tr><td>1</td><td>Channel is enabled</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER27: DMA Controller
4. Registers
4.7. DMA Controller Channel Configuration A Register Channel 'n' (DMAi_An)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit30] PB : Pause Bit
This bit is used to halt the transfer of the DMA channel. If this bit is set to '1', this channel halts the transfer and does not perform a DMA transfer until this bit is cleared.
When this bit is set to '1' while no transfer is ongoing DMAC enters the halt state immediately. When it was set to '1' during a running transfer, the halt state is entered at the next transfer gap. If the DMA transfer completes at the next transfer gap a completion interrupt is issued.
When this bit is set to '0' the halt condition is cleared and DMAC waits for the next request to continue the DMA transfer.
This bit is useful to halt a DMA transfer without re-configuration of each configuration register of the channel.
<table>
<tr><th>PB</th><th>Description</th></tr>
<tr><td>0</td><td>Channel is not halted</td></tr>
<tr><td>1</td><td>Channel is halted</td></tr>
</table>

(Correct)
[bit30] PB : Pause bit
This bit is used to halt the transfer of the DMA channel. If this bit is set to "1", this channel halts the transfer and does not perform a DMA transfer until this bit is cleared.
When this bit is set to "1" while no transfer is ongoing DMAC enters the halt state immediately. When it was set to "1" during a running transfer, the halt state is entered at the next transfer gap. If the DMA transfer completes at the next transfer gap a completion interrupt is issued.
When this bit is set to "0" the halt condition is cleared and DMAC waits for the next request to continue the DMA transfer.
This bit is useful to halt a DMA transfer without re-configuration of each configuration register of the channel.
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Channel is not halted</td></tr>
<tr><td>1</td><td>Channel is halted</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER27: DMA Controller
4. Registers
4.7. DMA Controller Channel Configuration A Register Channel 'n' (DMAi_An)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit28:27] IS[1:0] : Input Select
These bits are used to select the trigger source of a DMA transfer request. When the trigger source of a DMA transfer is a software request, IS bits are set to '00'. When the trigger source of a DMA transfer is a hardware request, IS bits are set to '01'.
<table>
<tr><th>IS[1:0]</th><th>Description</th></tr>
<tr><td>00</td><td>Software request</td></tr>
<tr><td>01</td><td>Hardware request</td></tr>
<tr><td>10</td><td>Reserved</td></tr>
<tr><td>11</td><td>Reserved</td></tr>
</table>

(Correct)
[bit28:27] IS[1:0] : Input Select
These bits are used to select the trigger source of a DMA transfer request. When the trigger source of a DMA transfer is a software request, IS bits are set to '00'. When the trigger source of a DMA transfer is a hardware request, IS bits are set to '01'.
<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>00</td><td>Software request</td></tr>
<tr><td>01</td><td>Hardware request</td></tr>
<tr><td>10</td><td>Reserved</td></tr>
<tr><td>11</td><td>Reserved</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER27: DMA Controller
4. Registers
4.7. DMA Controller Channel Configuration A Register Channel 'n' (DMAi_An)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit26] AL : Alternate
This bit decides whether the data transfers should alternate between read and write or should be contiguous reads followed by contiguous writes. The alternation takes place after each incremental read burst and after each single data read.
<table>
<tr><th>AL</th><th>Description</th></tr>
<tr><td>0</td><td>Contiguous</td></tr>
<tr><td>1</td><td>Alternate</td></tr>
</table>

(Correct)
[bit26] AL : Alternate
This bit decides whether the data transfers should alternate between read and write or should be contiguous reads followed by contiguous writes. The alternation takes place after each incremental read burst and after each single data read.
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Contiguous</td></tr>
<tr><td>1</td><td>Alternate</td></tr>
</table>
</td></tr>
<tr><td>CHAPTER27: DMA Controller
4. Registers
4.7. DMA Controller Channel Configuration A Register Channel 'n' (DMAi_An)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit25:24] BL[1:0] : Beat Limit
Beat Limit controls the maximum burst length the AHB master can make on the AHB bus for this DMA channel.
<table>
<tr><th>BL[1:0]</th><th>Description</th></tr>
<tr><td>00</td><td>Single transfer (SINGLE)</td></tr>
<tr><td>01</td><td>4-beat incrementing burst (INCR4)</td></tr>
<tr><td>10</td><td>8-bit incrementing burst (INCR8)</td></tr>
<tr><td>11</td><td>16-bit incrementing burst (INCR16)</td></tr>
</table>

(Correct)
[bit25:24] BL[1:0] : Beat Limit
Beat Limit controls the maximum burst length the AHB master can make on the AHB bus for this DMA channel.
<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>00</td><td>Single transfer (SINGLE)</td></tr>
<tr><td>01</td><td>4-beat incrementing burst (INCR4)</td></tr>
<tr><td>10</td><td>8-bit incrementing burst (INCR8)</td></tr>
<tr><td>11</td><td>16-bit incrementing burst (INCR16)</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER27: DMA Controller
4. Registers
4.8. DMA Controller Channel Configuration B Register Channel 'n' (DMAi_Bn)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit31] DQ : Flag of DIRQ
DQ is set to '1' when a DMA transfer completed successfully. DQ is cleared by hardware if DMAi_Cn:CD (Clear DIRQ) bit is set to '1'. Otherwise the value is retained.

[bit30] EQ : Flag of EDIRQ
EQ is set to '1' when a DMA transfer is finished with an error. EQ is cleared by hardware if DMAi_Cn:CE (Clear EDIRQ) bit is set to '1'. Otherwise the value is retained.

[bit29:28] MS[1:0] : Mode Select
MS sets the transfer mode of the channel.
<table>
<tr><th>MS[1:0]</th><th>Description</th></tr>
<tr><td>00</td><td>Block transfer mode</td></tr>
<tr><td>01</td><td>Burst transfer mode</td></tr>
<tr><td>10</td><td>Reserved</td></tr>
<tr><td>11</td><td>Reserved</td></tr>
</table>

(Correct)
[bit31] DQ : Flag of DIRQ
DQ is set to "1" when a DMA transfer completed successfully. DQ is cleared by hardware if DMAi_Cn:CD (Clear DIRQ) bit is set to "1". Otherwise the value is retained.

[bit30] EQ : Flag of EDIRQ
EQ is set to "1" when a DMA transfer is finished with an error. EQ is cleared by hardware if DMAi_Cn:CE (Clear EDIRQ) bit is set to "1". Otherwise the value is retained.

[bit29:28] MS[1:0] : Mode select
MS sets the transfer mode of the channel.
<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>00</td><td>Block transfer mode</td></tr>
<tr><td>01</td><td>Burst transfer mode</td></tr>
<tr><td>10</td><td>Reserved</td></tr>
<tr><td>11</td><td>Reserved</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER27: DMA Controller
4. Registers
4.8. DMA Controller Channel Configuration B Register Channel 'n' (DMAi_Bn)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit27:26] TW[1:0] : Transfer Width
TW specifies the data width for every data transfer of the DMA transfer.
<table>
<tr><th>TW[1:0]</th><th>Description</th></tr>
<tr><td>00</td><td>Byte</td></tr>
<tr><td>01</td><td>Half word</td></tr>
<tr><td>10</td><td>Word</td></tr>
<tr><td>11</td><td>Double word</td></tr>
</table>
[bit25] SR : Software Trigger Ready
This bit is used to indicate a condition (*) that the DMA channel is ready to receive a software request. The following conditions can cause that the DMA channel is not ready to receive a software request. If one or more of the conditions below are true SR is set to '0' by hardware. If none of the conditions are true SR is set to '1' by hardware.
■DMAi_R:DE == '0'; DMAC is disabled
■DMAi_R:DH == '1'; All DMA channels are halted
■DMAi_An:EB == '0'; DMA channel is disabled
■DMAi_An:PB == '1'; DMA channel is halted
■DMAi_An:IS != '00'; Input select is not set to software
■DEBUG == '1' and DMAi_R:DF == '1' and DMAi_R:DB == '01' or DMAi_R:DB == '10', debug event is pending while debug flag is set and Debug Behavior is set to HALT or STOP
■DMA transfer is active and a software request is pending

(Correct)
[bit27:26] TW[1:0] : Transfer width
TW specifies the data width for every data transfer of the DMA transfer.
<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>00</td><td>Byte</td></tr>
<tr><td>01</td><td>Half word</td></tr>
<tr><td>10</td><td>Word</td></tr>
<tr><td>11</td><td>Double word</td></tr>
</table>
[bit25] SR : Software trigger ready
This bit is used to indicate a condition* that the DMA channel is ready to receive a software request. The following conditions can cause that the DMA channel is not ready to receive a software request. If one or more of the conditions below are true SR is set to "0" by hardware. If none of the conditions are true SR is set to "1" by hardware.
■DMAi_R:DE == 0; DMAC is disabled
■DMAi_R:DH == 1; All DMA channels are halted
■DMAi_An:EB == 0; DMA channel is disabled
■DMAi_An:PB == 1; DMA channel is halted
■DMAi_An:IS != 0b00; Input select is not set to software
■DEBUG == 1 and DMAi_R:DF == 1 and DMAi_R:DB == 0b01 or DMAi_R:DB == 0b10, debug event is pending while debug flag is set and Debug Behavior is set to HALT or STOP
■DMA transfer is active and a software request is pending
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER27: DMA Controller
4. Registers
4.8. DMA Controller Channel Configuration B Register Channel 'n' (DMAi_Bn)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
(*) Refer the explanation of DMAiAn:ST bit in DMA Controller Channel Configuration A Register Channel 'n' (DMAi_An) for the rest of the conditions.

[bit20] EI : Error Interrupt Enable
This bit is used to control the issue of an error interrupt (EDIRQ). If this bit is set to '1', an error interrupt is issued due to any of the following transfer errors:
■Transfer stop request by signal DSTP, or disable the transfer with DMAi_An:EB or DMAi_R:DE, or debug event (if DMAi_R:DBE == '1' and DMAi_R:DB[1:0] == '10')
■Source access error
■Destination access error

<table>
<tr><th>EI</th><th>Description</th></tr>
<tr><td>0</td><td>Error interrupt issuance is disabled</td></tr>
<tr><td>1</td><td>Error interrupt issuance is enabled</td></tr>
</table>

(Correct)
*: Refer the explanation of DMAiAn:ST bit in DMA Controller Channel Configuration A Register Channel 'n' (DMAi_An) for the rest of the conditions.

[bit20] EI : Error interrupt enable
This bit is used to control the issue of an error interrupt (EDIRQ). If this bit is set to "1", an error interrupt is issued due to any of the following transfer errors:
■Transfer stop request by signal DSTP, or disable the transfer with DMAi_An:EB or DMAi_R:DE, or debug event (if DMAi_R:DBE == 1 and DMAi_R:DB[1:0] == 0b10)
■Source access error
■Destination access error

<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Error interrupt issuance is disabled</td></tr>
<tr><td>1</td><td>Error interrupt issuance is enabled</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER27: DMA Controller
4. Registers
4.8. DMA Controller Channel Configuration B Register Channel 'n' (DMAi_Bn)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit19] CI : Completion Interrupt Enable
This bit is used to control the issue of a completion interrupt (DIRQ). If this bit is set to '1', a completion interrupt is issued after the DMA transfer completed normally.
<table>
<tr><th>CI</th><th>Description</th></tr>
<tr><td>0</td><td>Completion interrupt issuance is disabled</td></tr>
<tr><td>1</td><td>Completion interrupt issuance is enabled</td></tr>
</table>

(Correct)
[bit19] CI : Completion interrupt enable
This bit is used to control the issue of a completion interrupt (DIRQ). If this bit is set to "1", a completion interrupt is issued after the DMA transfer completed normally.
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Completion interrupt issuance is disabled</td></tr>
<tr><td>1</td><td>Completion interrupt issuance is enabled</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER27: DMA Controller
4. Registers
4.8. DMA Controller Channel Configuration B Register Channel 'n' (DMAi_Bn)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit18:16] SS[2:0] : Stop Status
These bits are used to show the end code of DMA transfer. SS is set by hardware when an error or completion interrupt is raised and it is cleared by hardware when either DMAi_Cn:CE or DMAi_Cn:CD is set to '1'.
<table>
<tr><th>SS[2:0]</th><th>Description</th></tr>
<tr><td>000</td><td>Initial value.
Status : None</td></tr>
<tr><td>001</td><td>Reserved</td></tr>
<tr><td>010</td><td>Stop request by:
– DSTP
– Channel disable (DMAi_An:EB='0')
– DMA disable (DMAi_R:DE='0')
– Debug event (DMAi_R:DBE='1' and DMAi_R:DB='10')
Status : Stop</td></tr>
</table>

(Correct)
[bit18:16] SS[2:0] : Stop status
These bits are used to show the end code of DMA transfer. SS is set by hardware when an error or completion interrupt is raised and it is cleared by hardware when either DMAi_Cn:CE or DMAi_Cn:CD is set to "1".
<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>000</td><td>Initial value.
Status : None</td></tr>
<tr><td>001</td><td>Reserved</td></tr>
<tr><td>010</td><td>Stop request by:
– DSTP
– Channel disable (DMAi_An:EB= 0)
– DMA disable (DMAi_R:DE= 0)
– Debug event (DMAi_R:DBE= 1 and DMAi_R:DB= 0b10)
Status : Stop</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER27: DMA Controller
4. Registers
4.12. DMA Controller Channel Configuration D Register Channel 'n' (DMAi_Dn)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit31] FS : Fixed Source Address
This bit is used to keep the source address at a fix value.
<table>
<tr><th>FS</th><th>Description</th></tr>
<tr><td>0</td><td>Source address is incremented</td></tr>
<tr><td>1</td><td>Source address is kept fix. AHB master only makes single transfers to access the source</td></tr>
</table>
[bit30] DES : Decrement Source Address
If this bit is set the source address on the AHB interface is decremented on each AHB transfer. In this mode the AHB master makes only single transfers to access the source.
<table>
<tr><th>DES</th><th>Description</th></tr>
<tr><td>0</td><td>Source address is incremented</td></tr>
<tr><td>1</td><td>Source address is decremented</td></tr>
</table>
(Correct)
[bit31] FS : Fixed Source address
This bit is used to keep the source address at a fix value.
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Source address is incremented</td></tr>
<tr><td>1</td><td>Source address is kept fix. AHB master only makes single transfers to access the source</td></tr>
</table>
[bit30] DES : Decrement source address
If this bit is set the source address on the AHB interface is decremented on each AHB transfer. In this mode the AHB master makes only single transfers to access the source.
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Source address is incremented</td></tr>
<tr><td>1</td><td>Source address is decremented</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER27: DMA Controller
4. Registers
4.12. DMA Controller Channel Configuration D Register Channel 'n' (DMAi_Dn)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
Note:
−Fixed Source Address (DMAi_Dn:FS) has higher priority than Decrement Source Address. This bit is only effective if DMAi_Dn:FS = '0' and DMAi_Dn:FBS = '0'.

[bit29] US : Update source address
<table>
<tr><th>US</th><th>Description</th></tr>
<tr><td>0</td><td>DMAi_SAn is not updated after DMA transfer is successfully completed. DMAi_SAn is retained</td></tr>
<tr><td>1</td><td>DMAi_SAn is updated with the next address after the DMA transfer is successfully completed. E.g. last source address is 0x0BF000FC</td></tr>
</table>
Note:
− This bit is only effective if DMAi_Dn:FBS = '0'

[bit28] FBS : Fixed Block Source Address
<table>
<tr><th>FBS</th><th>Description</th></tr>
<tr><td>0</td><td>Start address of first block of DMA transfer is set to the value stored in DMAi_SAn. Start address of consecutive blocks is the address following the previous block last address (according to Transfer Width (DMAi_Bn:TW))</td></tr>
<tr><td>1</td><td>Start address of each block is set to the value stored in DMAi_SAn</td></tr>
</table>

(Correct)
Note:
−Fixed Source Address (DMAi_Dn:FS) has higher priority than Decrement Source Address. This bit is only effective if DMAi_Dn:FS = 0 and DMAi_Dn:FBS = 0.

[bit29] US : Update source address
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>DMAi_SAn is not updated after DMA transfer is successfully completed. DMAi_SAn is retained</td></tr>
<tr><td>1</td><td>DMAi_SAn is updated with the next address after the DMA transfer is successfully completed. E.g. last source address is 0x0BF000FC</td></tr>
</table>
Note:
−This bit is only effective if DMAi_Dn:FBS = 0

[bit28] FBS : Fixed block source address
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Start address of first block of DMA transfer is set to the value stored in DMAi_SAn. Start address of consecutive blocks is the address following the previous block last address (according to Transfer Width (DMAi_Bn:TW))</td></tr>
<tr><td>1</td><td>Start address of each block is set to the value stored in DMAi_SAn</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER27: DMA Controller
4. Registers
4.12. DMA Controller Channel Configuration D Register Channel 'n' (DMAi_Dn)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
Note:
−Setting of FBS is only effective if DMAi_Dn:FS = '0' and (DMAi_Bn:MS = '00' (block transfer mode) or DMAi_Bn:MS = '01' (burst transfer mode)).

[bit15] FD : Fixed Destination Address
This bit is used to keep the destination address at a fix value.
<table>
<tr><th>FD</th><th>Description</th></tr>
<tr><td>0</td><td>Destination address is incremented</td></tr>
<tr><td>1</td><td>Destination address is kept fixed. AHB master only makes single transfers to access the destination</td></tr>
</table>
[bit14] DED : Decrement Destination Address
If this bit is set the destination address on the AHB interface is decremented on each AHB transfer. In this mode the AHB master makes only single transfers to access the destination.
<table>
<tr><th>DED</th><th>Description</th></tr>
<tr><td>0</td><td>Destination address is incremented</td></tr>
<tr><td>1</td><td>Destination address is decremented</td></tr>
</table>
Note:
−Fixed Destination Address (DMAi_Dn:FD) has higher priority than Decrement Destination Address. This bit is only effective if DMAi_Dn:FD = '0' and DMAi_Dn:FBD = '0'.

(Correct)
Note:
−Setting of FBS is only effective if DMAi_Dn:FS = 0 and (DMAi_Bn:MS = 0b00 (block transfer mode) or DMAi_Bn:MS = 0b01 (burst transfer mode)).

[bit15] FD : Fixed destination address
This bit is used to keep the destination address at a fix value.
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Destination address is incremented</td></tr>
<tr><td>1</td><td>Destination address is kept fixed. AHB master only makes single transfers to access the destination</td></tr>
</table>
[bit14] DED : Decrement destination address
If this bit is set the destination address on the AHB interface is decremented on each AHB transfer. In this mode the AHB master makes only single transfers to access the destination.
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Destination address is incremented</td></tr>
<tr><td>1</td><td>Destination address is decremented</td></tr>
</table>
Note:
−Fixed Destination Address (DMAi_Dn:FD) has higher priority than Decrement Destination Address. This bit is only effective if DMAi_Dn:FD = 0 and DMAi_Dn:FBD = 0.
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER27: DMA Controller
4. Registers
4.12. DMA Controller Channel Configuration D Register
Channel 'n' (DMAi_Dn)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit13] UD : Update Destination Address
<table>
<tr><th>UD</th><th>Description</th></tr>
<tr><td>0</td><td>DMAi_DAn is not updated after DMA transfer is successfully completed. DMAi_DAn is retained</td></tr>
<tr><td>1</td><td>DMAi_DAn is updated with the next address after the DMA transfer is successfully completed. E.g. last destination address was 0x0BF40010, DMAi_Bn:TW = 10 (word) then DMAi_DAn is updated with address 0x0BF40014.</td></tr>
</table>
Note:
−This bit is effective only if DMAi_Dn:FBD = '0'

[bit12] FBD : Fixed Block Destination Address
<table>
<tr><th>FBD</th><th>Description</th></tr>
<tr><td>0</td><td>Start address of first block of DMA transfer is set to the value stored in DMAi_DAn. Start address of consecutive blocks is the address following the previous block last address (according to Transfer Width (DMAi_Bn:TW)</td></tr>
<tr><td>1</td><td>Start address of each block is set to the value stored in DMAi_DAn</td></tr>
</table>
Note:
−Setting of FBD is only effective if FD = '0' and (DMAi_Bn:MS = '00' (block transfer mode) or DMAi_Bn:MS = '01' (burst transfer mode)).

(Correct)
[bit13] UD : Update destination address
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>DMAi_DAn is not updated after DMA transfer is successfully completed. DMAi_DAn is retained</td></tr>
<tr><td>1</td><td>DMAi_DAn is updated with the next address after the DMA transfer is successfully completed. E.g. last destination address was 0x0BF40010, DMAi_Bn:TW = 10 (word) then DMAi_DAn is updated with address 0x0BF40014.</td></tr>
</table>
Note:
− This bit is effective only if DMAi_Dn:FBD = 0

[bit12] FBD : Fixed block destination address
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Start address of first block of DMA transfer is set to the value stored in DMAi_DAn. Start address of consecutive blocks is the address following the previous block last address (according to Transfer Width (DMAi_Bn:TW)</td></tr>
<tr><td>1</td><td>Start address of each block is set to the value stored in DMAi_DAn</td></tr>
</table>
Note:
−Setting of FBD is only effective if FD = 0 and (DMAi_Bn:MS = 0b00 (block transfer mode) or DMAi_Bn:MS = 0b01 (burst transfer mode)).
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER27: DMA Controller
4. Registers
4.16. DMA Controller Client Matrix Internal Client Interface Configuration Register 'm' (DMAi_CMICICm)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit28] Reserved
Always write '0' to this bit.

[bit27] BEHSTPACK : Behavior Stop Acknowledge
BEHSTPACK sets the behavior of the Internal DMA Client Interface 'm' output signal DSTP_ACK[m] if the client interface is not selected in any of the Channel Configuration Registers (DMAi_CMCHICn).
Note:
−This configuration bit is read0 in DMAi_CMICICx (x=10, 11, … , (product specification))
<table>
<tr><th>BEHSTPACK</th><th>Description</th></tr>
<tr><td>0</td><td>DSTP_ACK[m] outputs inactive logic level</td></tr>
<tr><td>1</td><td>DSTP[m] connects directly to DSTP_ACK[m]</td></tr>
</table>

(Correct)
[bit28] Reserved
Always write "0" to this bit.

[bit27] BEHSTPACK : Behavior stop acknowledge
BEHSTPACK sets the behavior of the Internal DMA Client Interface 'm' output signal DSTP_ACK[m] if the client interface is not selected in any of the Channel Configuration Registers (DMAi_CMCHICn).
Note:
−This configuration bit is read0 in DMAi_CMICICx (x=10, 11, … , (product specification))
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>DSTP_ACK[m] outputs inactive logic level</td></tr>
<tr><td>1</td><td>DSTP[m] connects directly to DSTP_ACK[m]</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER27: DMA Controller
4. Registers
4.16. DMA Controller Client Matrix Internal Client Interface Configuration Register 'm' (DMAi_CMICICm)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit25] BEHREQACK : Behavior Request Acknowledge
BEHREQACK sets the behavior of the Internal DMA Client Interface 'm' output signal DREQ_ACK[m] if the client interface is not selected in any of the Channel Configuration Registers (DMAi_CMCHICn).
<table>
<tr><th>BEHREQACK</th><th>Description (1)</th></tr>
<tr><td>0</td><td>DREQ_ACK[m] outputs inactive logic level.</td></tr>
<tr><td>1</td><td>DREQ[m] connects directly to DREQ_ACK[m].</td></tr>
</table>
(Correct)
[bit25] BEHREQACK : Behavior request acknowledge
BEHREQACK sets the behavior of the Internal DMA Client Interface 'm' output signal DREQ_ACK[m] if the client interface is not selected in any of the Channel Configuration Registers (DMAi_CMCHICn).
<table>
<tr><th>Bit</th><th>Description (1)</th></tr>
<tr><td>0</td><td>DREQ_ACK[m] outputs inactive logic level.</td></tr>
<tr><td>1</td><td>DREQ[m] connects directly to DREQ_ACK[m].</td></tr>
</table></td></tr>
<tr><td>CHAPTER27: DMA Controller
4. Registers
4.16. DMA Controller Client Matrix Internal Client Interface Configuration Register 'm' (DMAi_CMICICm)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
This bit also works as DMA request enable signal to the corresponding peripheral which send DMA request by IRQ. The bit must be changed while the peripheral function is disabled to guarantee that its IRQ is kept to disabled..
<table>
<tr><th>BEHREQACK</th><th>Description (2)</th></tr>
<tr><td>0</td><td>For the corresponding peripheral whose DMA request is controlled by IRQ:
DMA request by IRQ is disabled.</td></tr>
<tr><td>1</td><td>For the corresponding peripheral whose DMA request is controlled by IRQ:
DMA request by IRQ is enabled.</td></tr>
</table>
(Correct)
This bit also works as DMA request enable signal to the corresponding peripheral which send DMA request by IRQ. The bit must be changed while the peripheral function is disabled to guarantee that its IRQ is kept to disabled..
<table>
<tr><th>Bit</th><th>Description (2)</th></tr>
<tr><td>0</td><td>For the corresponding peripheral whose DMA request is controlled by IRQ:
DMA request by IRQ is disabled.</td></tr>
<tr><td>1</td><td>For the corresponding peripheral whose DMA request is controlled by IRQ:
DMA request by IRQ is enabled.</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER27: DMA Controller
4. Registers
4.17. DMA Controller Client Matrix Channel Interface Configuration Register 'n' (DMAi_CMCHICn)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
<table>
<tr><td>bit</td><td>bit 7</td><td>bit 6</td><td>bit 5</td><td>bit 4</td><td>bit 3</td><td>bit 2</td><td>bit 1</td><td>bit 0</td></tr>
<tr><td>Field</td><td>CI[7]</td><td>CI[6]</td><td>CI[5]</td><td>CI[4]</td><td>CI[3]</td><td>CI[2]</td><td>CI[1]</td><td>CI[0]</td></tr>
<tr><td>Attribute</td><td>R,W</td><td>R,W</td><td>R,W</td><td>R,W</td><td>R,W</td><td>R,W</td><td>R,W</td><td>R,W</td></tr>
<tr><td>Protection</td><td colspan="8">WP</td></tr>
<tr><td>Initial Value</td><td>0</td><td>0</td><td>0</td><td>0</td><td>(*1)</td><td>(*1)</td><td>(*1)</td><td>(*1)</td></tr>
</table>
(*1) Initial value of DMAi_CMCHICn:CI[8:0] is 'n'.
(e.g.) Initial value of DMAi_CMCHIC0:CI[8:0]=000000000
Initial value of DMAi_CMCHIC5:CI[8:0]=000000101

(Correct)
<table>
<tr><td>bit</td><td>bit 7</td><td>bit 6</td><td>bit 5</td><td>bit 4</td><td>bit 3</td><td>bit 2</td><td>bit 1</td><td>bit 0</td></tr>
<tr><td>Field</td><td>CI[7]</td><td>CI[6]</td><td>CI[5]</td><td>CI[4]</td><td>CI[3]</td><td>CI[2]</td><td>CI[1]</td><td>CI[0]</td></tr>
<tr><td>Attribute</td><td>R,W</td><td>R,W</td><td>R,W</td><td>R,W</td><td>R,W</td><td>R,W</td><td>R,W</td><td>R,W</td></tr>
<tr><td>Protection</td><td colspan="8">WP</td></tr>
<tr><td>Initial Value</td><td>0</td><td>0</td><td>0</td><td>0</td><td>*1</td><td>*1</td><td>*1</td><td>*1</td></tr>
</table>
*1: Initial value of DMAi_CMCHICn:CI[8:0] is 'n'.
(e.g.) Initial value of DMAi_CMCHIC0:CI[8:0]= 0b000000000
Initial value of DMAi_CMCHIC5:CI[8:0]= 0b000000101
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER27: DMA Controller
4. Registers
4.17. DMA Controller Client Matrix Channel Interface Configuration Register 'n' (DMAi_CMCHICn)</td><td>Features of "For details" should be corrected as indicated by the shading below.
(Error)
<table>
<tr><th>RLESEL</th><th>Description</th></tr>
<tr><td>000000</td><td>CMCHIC Reload event of DMAi_CMCHICn is disabled.</td></tr>
<tr><td>000100</td><td>Negative edge of TOT of Reload Timer ch.0 in the corresponding DMA Additional Control</td></tr>
<tr><td>000101</td><td>Negative edge of TOT of Reload Timer ch.1 in the corresponding DMA Additional Control</td></tr>
<tr><td>000110</td><td>Negative edge of TOT of Reload Timer ch.2 in the corresponding DMA Additional Control</td></tr>
<tr><td>000111</td><td>Negative edge of TOT of Reload Timer ch.3 in the corresponding DMA Additional Control</td></tr>
<tr><td>001000
~
001111</td><td>CMCHIC Reload event of DMAi_CMCHICn is disabled.</td></tr>
<tr><td>010000</td><td>BSTART from ch.0</td></tr>
<tr><td>010001</td><td>BSTART from ch.1</td></tr>
<tr><td>~</td><td>~</td></tr>
<tr><td>011111</td><td>BSTART from ch.15</td></tr>
<tr><td>100000</td><td>BDONE from ch.0</td></tr>
<tr><td>100001</td><td>BDONE from ch.1</td></tr>
<tr><td>~</td><td>~</td></tr>
<tr><td>101111</td><td>BDONE from ch.15</td></tr>
<tr><td>Else</td><td>No signal is selected</td></tr>
</table>
(Correct)
<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>000000</td><td>CMCHIC Reload event of DMAi_CMCHICn is disabled.</td></tr>
<tr><td>000100</td><td>Negative edge of TOT of Reload Timer ch.0 in the corresponding DMA Additional Control</td></tr>
<tr><td>000101</td><td>Negative edge of TOT of Reload Timer ch.1 in the corresponding DMA Additional Control</td></tr>
<tr><td>000110</td><td>Negative edge of TOT of Reload Timer ch.2 in the corresponding DMA Additional Control</td></tr>
<tr><td>000111</td><td>Negative edge of TOT of Reload Timer ch.3 in the corresponding DMA Additional Control</td></tr>
<tr><td>001000
to
001111</td><td>CMCHIC Reload event of DMAi_CMCHICn is disabled.</td></tr>
<tr><td>010000</td><td>BSTART from ch.0</td></tr>
<tr><td>010001</td><td>BSTART from ch.1</td></tr>
<tr><td>.....</td><td>.....</td></tr>
<tr><td>011111</td><td>BSTART from ch.15</td></tr>
<tr><td>100000</td><td>BDONE from ch.0</td></tr>
<tr><td>100001</td><td>BDONE from ch.1</td></tr>
<tr><td>.....</td><td>.....</td></tr>
<tr><td>101111</td><td>BDONE from ch.15</td></tr>
<tr><td>Else</td><td>No signal is selected</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER27: DMA Controller
4. Registers
4.17. DMA Controller Client Matrix Channel Interface Configuration Register 'n' (DMAi_CMCHICn)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit18:16] RLSLOT[2:0] : Reload Slot Replacement
The bit specifies which bank is copied DMAi_CHCHICn at CMCHIC reload event. Please refer "CHAPTER: DMA COMPLEX SUBSYSTEM" for more details.
<table>
<tr><th>RLSLOT</th><th>Description</th></tr>
<tr><td>000</td><td>Value of DMAAn_CHCHICRDB0 register is copied to DMAi_CMCHICn at CMCHIC reload event.</td></tr>
<tr><td>001</td><td>Value of DMAAn_CHCHICRDB1 register is copied to DMAi_CMCHICn at CMCHIC reload event.</td></tr>
<tr><td>~</td><td>~</td></tr>
<tr><td>111</td><td>Value of DMAAn_CHCHICRDB8 register is copied to DMAi_CMCHICn at CMCHIC reload event.</td></tr>
</table>

(Correct)
[bit18:16] RLSLOT[2:0] : Reload slot replacement
The bit specifies which bank is copied DMAi_CMCHICn at CMCHIC reload event. Please refer "CHAPTER: DMA COMPLEX SUBSYSTEM" for more details.
<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>000</td><td>Value of DMAAn_CMCHICRDB0 register is copied to DMAi_CMCHICn at CMCHIC reload event.</td></tr>
<tr><td>001</td><td>Value of DMAAn_CMCHICRDB1 register is copied to DMAi_CMCHICn at CMCHIC reload event.</td></tr>
<tr><td>.....</td><td>.....</td></tr>
<tr><td>111</td><td>Value of DMAAn_CMCHICRDB8 register is copied to DMAi_CMCHICn at CMCHIC reload event.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER28: DMA COMPLEX SUBSYSTEM
4. Registers
4.4. DMA Additional Control Additional Status Register 3 (DMAAn_ASR3)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit15:0] REQ[15:0]
When the bit is '1', the channel is received a request but it has not executed.
<table>
<tr><th>REQ[x]</th><th>Description</th></tr>
<tr><td>1</td><td>There is a pending request in channel 'x'.</td></tr>
<tr><td>0</td><td>No request</td></tr>
</table>
(Correct)
[bit15:0] REQ[15:0]
When the bit is "1", the channel is received a request but it has not executed.
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>1</td><td>There is a pending request in channel 'x'.</td></tr>
<tr><td>0</td><td>No request</td></tr>
</table></td></tr>
<tr><td>CHAPTER28: DMA COMPLEX SUBSYSTEM
7.Operation of the 32-bit Reload Timer in DMA COMPLEX SUBSYSTEM</td><td>The function should be deleted as indicated by the shading below.

(Error)
7.1Internal clock and external event counter operations of 32-bit Reload Timer
7.2Underflow operation of 32-bit Reload Timer
7.3Output functions of 32-bit Reload Timer
7.4Counter operation state
7.5DMA operation

(Correct)
7.1 Internal Clock and External Event Counter Operations of 32-bit Reload Timer
7.2 Underflow Operation of 32-bit Reload Timer
7.3 Output Functions of 32-bit Reload Timer
7.4 Counter Operation
7.5 DMA</td></tr>
</table>

Section	Change Results
CHAPTER28: DMA COMPLEX SUBSYSTEM 8.Registers in DMA COMPLEX SUBSYSTEM 8.2.Timer Control Status Register (DMAAn_RLTm_TMCSR)	The item should be added as indicated by the shading below. (Error) Note: −When DBGE is set to ’1’ and the processor is in debug state, the timer counter operation is paused, and writing to the DMAAn_RLTm_TMRLR register directly updates the timer counter (DMAAn_RLTm_TMR register). When the processor leaves debug state or DBGE is set to ’0’, the timer counter operation is resumed. (Correct) Note: −When DBGE is set to "1" and the processor is in debug state, the timer counter operation is paused, and writing to the DMAAn_RLTm_TMRLR register directly updates the timer counter (DMAAn_RLTm_TMR register). When the processor leaves debug state or DBGE is set to "0", the timer counter operation is resumed. −This bit is enable at ☆from ch. 0 to ch. 3 of CPER#2 ☆from ch.0 to ch. 1 of MCU_CONFIG_GROUP.

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER28: DMA COMPLEX SUBSYSTEM
8.Registers in DMA COMPLEX SUBSYSTEM
8.2.Timer Control Status Register (DMAAn_RLTm_TMCSR)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
Table 8-2 DMAAn_RLTm_TMCSR:MOD[2:0] bit settings for internal clock mode
(DMAAn_RLTm_TMCSR:CSL1 / DMAAn_RLTm_TMCSR:CSL2 = '00', '01', or '10')
<table>
<tr><th>MOD[2]</th><th>MOD[1]</th><th>MOD[0]</th><th>Input function</th><th>Active edge or level</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>Trigger disabled</td><td>-</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td rowspan="3">Trigger input</td><td>Rising edge</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>Falling edge</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>Both edge</td></tr>
<tr><td>1</td><td>x</td><td>0</td><td rowspan="2">Gate input</td><td>'L' level</td></tr>
<tr><td>1</td><td>x</td><td>1</td><td>'H' level</td></tr>
</table>
Table 8-3 DMAAn_RLTm_TMCSR:MOD[2:0] bit settings for event counter mode
(DMAAn_RLTm_TMCSR:CSL1 / DMAAn_RLTm_TMCSR:CSL2 = '11')
<table>
<tr><th>MOD[2]</th><th>MOD[1]</th><th>MOD[0]</th><th>Input function</th><th>Active edge or level</th></tr>
<tr><td rowspan="4">x</td><td>0</td><td>0</td><td>-</td><td>-</td></tr>
<tr><td>0</td><td>1</td><td rowspan="3">Event input</td><td>Rising edge</td></tr>
<tr><td>1</td><td>0</td><td>Falling edge</td></tr>
<tr><td>1</td><td>1</td><td>Both edge</td></tr>
</table>
(Correct)
Table 8-2 DMAAn_RLTm_TMCSR:MOD[2:0] Bit Settings for Internal Clock Mode
(DMAAn_RLTm_TMCSR:CSL1 / DMAAn_RLTm_TMCSR:CSL2 = 0b00, 0b01, or 0b10)
<table>
<tr><th>MOD[2]</th><th>MOD[1]</th><th>MOD[0]</th><th>Input Function</th><th>Active Edge or Level</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>Trigger disabled</td><td>-</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td rowspan="3">Trigger input</td><td>Rising edge</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>Falling edge</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>Both edge</td></tr>
<tr><td>1</td><td>x</td><td>0</td><td rowspan="2">Gate input</td><td>"L" level</td></tr>
<tr><td>1</td><td>x</td><td>1</td><td>"H" level</td></tr>
</table>
Table 8-3 DMAAn_RLTm_TMCSR:MOD[2:0] Bit Settings for Event Counter Mode
(DMAAn_RLTm_TMCSR:CSL1 / DMAAn_RLTm_TMCSR:CSL2 = 0b11)
<table>
<tr><th>MOD[2]</th><th>MOD[1]</th><th>MOD[0]</th><th>Input Function</th><th>Active Edge or Level</th></tr>
<tr><td rowspan="4">x</td><td>0</td><td>0</td><td>-</td><td>-</td></tr>
<tr><td>0</td><td>1</td><td rowspan="3">Event input</td><td>Rising edge</td></tr>
<tr><td>1</td><td>0</td><td>Falling edge</td></tr>
<tr><td>1</td><td>1</td><td>Both edge</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER32: External Time Stamp Counter For CAN FD 4.Example of the Operation</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
4.4.Example of the Operation

(Correct)
4.Example of the Operation</td></tr>
<tr><td>CHAPTER32: External Time Stamp Counter For CAN FD 5.Register 5.1.Time Stamp Control Register (MCG_CANFDx_TSCNTR, CPG_CANFDx_TSCNTR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
<table>
<tr><td>bit</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td>Field</td><td colspan="8">Reserved</td></tr>
<tr><td>R/W attribute</td><td>R0,W0</td><td>R0,W0</td><td>R0,W0</td><td>R0,W0</td><td>R0,W0</td><td>R0,W0</td><td>R0,W0</td><td>R0,W0</td></tr>
<tr><td>Protection attribute</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>Initial value</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
<table>
<tr><td>Bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Field</td><td colspan="7">Reserved</td><td>CCLR</td></tr>
<tr><td>R/W Attribute</td><td>R0,W0</td><td>R0,W0</td><td>R0,W0</td><td>R0,W0</td><td>R0,W0</td><td>R0,W0</td><td>R0,W0</td><td>R0,W</td></tr>
<tr><td>Protection attribute</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>Initial value</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
(Correct)
<table>
<tr><td>bit</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td>Field</td><td colspan="8">Reserved</td></tr>
<tr><td>R/W Attribute</td><td colspan="8">R0, W0</td></tr>
<tr><td>Protection attribute</td><td colspan="8">-</td></tr>
<tr><td>Initial value</td><td colspan="8">00000000</td></tr>
</table>
<table>
<tr><td>bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Field</td><td colspan="7">Reserved</td><td>CCLR</td></tr>
<tr><td>R/W Attribute</td><td colspan="7">R0, W0</td><td>R0, W</td></tr>
<tr><td>Protection attribute</td><td colspan="7">-</td><td></td></tr>
<tr><td>Initial value</td><td colspan="7">0000000</td><td>0</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER32: External Time Stamp Counter For CAN FD
5.Register
5.2.Time Stamp Mode Register (MCG_CANFDx_TSMDR, MCG_CANFDx_TSMDR)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
<table>
<tr><td>bit</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td>Field</td><td colspan="8">Reserved</td></tr>
<tr><td>R/W attribute</td><td>R0,W0</td><td>R0,W0</td><td>R0,W0</td><td>R0,W0</td><td>R0,W0</td><td>R0,W0</td><td>R0,W0</td><td>R0,W0</td></tr>
<tr><td>Protection attribute</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>Initial value</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
<table>
<tr><td>bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Field</td><td colspan="7">Reserved</td><td>CNTEN</td></tr>
<tr><td>R/W attribute</td><td>R0,W0</td><td>R0,W0</td><td>R0,W0</td><td>R0,W0</td><td>R0,W0</td><td>R0,W0</td><td>R0,W0</td><td>R/W</td></tr>
<tr><td>Protection attribute</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>Initial value</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
(Correct)
<table>
<tr><td>bit</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td>Field</td><td colspan="8">Reserved</td></tr>
<tr><td>R/W Attribute</td><td colspan="8">R0,W0</td></tr>
<tr><td>Protection attribute</td><td colspan="8">-</td></tr>
<tr><td>Initial value</td><td colspan="8">00000000</td></tr>
</table>
<table>
<tr><td>bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Field</td><td colspan="7">Reserved</td><td>CNTEN</td></tr>
<tr><td>R/W Attribute</td><td colspan="7">R0,W0</td><td>R/W</td></tr>
<tr><td>Protection attribute</td><td colspan="7">-</td><td></td></tr>
<tr><td>Initial value</td><td colspan="7">0000000</td><td>0</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER32: External Time Stamp Counter For CAN FD
5.Register
5.3.Time Stamp Divider Register (MCG_CANFDx_TSDIVR, CPG_CANFDx_TSDIVR)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
<table>
<tr><td>bit</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td>Field</td><td colspan="8">Reserved</td></tr>
<tr><td>R/W attribute</td><td>R0,W0</td><td>R0,W0</td><td>R0,W0</td><td>R0,W0</td><td>R0,W0</td><td>R0,W0</td><td>R0,W0</td><td>R0,W0</td></tr>
<tr><td>Protection attribute</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>Initial value</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
<table>
<tr><td>Bit</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td>Field</td><td colspan="8">Reserved</td></tr>
<tr><td>R/W attribute</td><td>R0,W0</td><td>R0,W0</td><td>R0,W0</td><td>R0,W0</td><td>R0,W0</td><td>R0,W0</td><td>R0,W0</td><td>R0,W0</td></tr>
<tr><td>Protection attribute</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>Initial value</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
(Correct)
<table>
<tr><td>bit</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td>Field</td><td colspan="8">Reserved</td></tr>
<tr><td>R/W Attribute</td><td colspan="8">R0,W0</td></tr>
<tr><td>Protection attribute</td><td colspan="8">-</td></tr>
<tr><td>Initial value</td><td colspan="8">00000000</td></tr>
</table>
<table>
<tr><td>bit</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td>Field</td><td colspan="8">Reserved</td></tr>
<tr><td>R/W Attribute</td><td colspan="8">R0,W0</td></tr>
<tr><td>Protection attribute</td><td colspan="8">-</td></tr>
<tr><td>Initial value</td><td colspan="8">00000000</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER32: External Time Stamp Counter For CAN FD
5.Register
5.3.Time Stamp Divider Register (MCG_CANFDx_TSDIVR, CPG_CANFDx_TSDIVR)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
<table>
<tr><td>Bit</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td>Field</td><td>CDIV15</td><td>CDIV14</td><td>CDIV13</td><td>CDIV12</td><td>CDIV11</td><td>CDIV10</td><td>CDIV9</td><td>CDIV8</td></tr>
<tr><td>R/W attribute</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td></tr>
<tr><td>Protection attribute</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>Initial Value</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
<table>
<tr><td>Bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Field</td><td>CDIV7</td><td>CDIV6</td><td>CDIV5</td><td>CDIV4</td><td>CDIV3</td><td>CDIV2</td><td>CDIV1</td><td>CDIV0</td></tr>
<tr><td>R/W attribute</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td></tr>
<tr><td>Protection attribute</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>Initial value</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
(Correct)
<table>
<tr><td>bit</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td>Field</td><td>CDIV15</td><td>CDIV14</td><td>CDIV13</td><td>CDIV12</td><td>CDIV11</td><td>CDIV10</td><td>CDIV9</td><td>CDIV8</td></tr>
<tr><td>R/W Attribute</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td></tr>
<tr><td>Protection attribute</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>Initial value</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
<table>
<tr><td>bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Field</td><td>CDIV7</td><td>CDIV6</td><td>CDIV5</td><td>CDIV4</td><td>CDIV3</td><td>CDIV2</td><td>CDIV1</td><td>CDIV0</td></tr>
<tr><td>R/W Attribute</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td></tr>
<tr><td>Protection attribute</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td></td></tr>
<tr><td>Initial value</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
</td></tr>
</table>

Section	Change Results
CHAPTER33: CAN FD Message RAM ECC Function 1.Overview	Features of "For details" should be corrected as indicated by the shading below. (Error) Note: −ECC error generation function of the message RAM is not supported for Traveo Platform. Therefore CAN FD ECC Error Insertion Control Register (FDFECR) is not writeable. (Correct) Note: −ECC error generation function of the message RAM is not supported for Traveo Platform. Therefore CAN FD ECC False Error Control Register (FDFECR) is not writeable.

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER33: CAN FD Message RAM ECC Function 6.Registers</td><td>The item should be added as indicated by the shading below.

(Error)
This section explains the registers of the CAN FD Message RAM ECC Function.

(Correct)
This section explains the registers of the CAN FD Message RAM ECC Function.

Table 6-1 List of the Registers of the CAN FD Message RAM ECC Function
<table>
<tr><th>Abbreviated Register Name</th><th>Register Name</th><th>Reference</th></tr>
<tr><td>MCG_CANFDx_FDECR</td><td>CAN FD ECC error control register of MCU Config Group (channel 0,1,2)</td><td>6.1</td></tr>
<tr><td>CPG_CANFDx_FDECR</td><td>CAN FD ECC error control register of Common Peripheral #0 Group (channel 0,1,2,3,4)</td><td>6.1</td></tr>
<tr><td>MCG_CANFDx_FDESR</td><td>CAN FD ECC error status register of MCU Config Group (channel 0,1,2)</td><td>6.2</td></tr>
<tr><td>CPG_CANFDx_FDESR</td><td>CAN FD ECC error status register of Common Peripheral #0 Group (channel 0,1,2,3,4)</td><td>6.2</td></tr>
<tr><td>MCG_CANFDx_FDESCR</td><td>CAN FD ECC error status clear register of MCU Config Group (channel 0,1,2)</td><td>6.3</td></tr>
<tr><td>CPG_CANFDx_FDESCR</td><td>CAN FD ECC error status clear register of Common Peripheral #0 Group (channel0,1,2,3,4)</td><td>6.3</td></tr>
<tr><td>MCG_CANFDx_FDDEAR</td><td>CAN FD ECC double-bit error address register of MCU Config Group (channel 0,1,2)</td><td>6.4</td></tr>
<tr><td>CPG_CANFDx_FDDEAR</td><td>CAN FD ECC double-bit error address register of Common Peripheral #0 Group(channel 0,1,2,3,4)</td><td>6.4</td></tr>
<tr><td>MCG_CANFDx_FDSEAR</td><td>CAN FD ECC single-bit error address register of MCU Config Group (channel 0,1,2)</td><td>6.5</td></tr>
<tr><td>CPG_CANFDx_FDSEAR</td><td>CAN FD ECC single-bit error address register of Common Peripheral #0 Group (channel0,1,2,3,4)</td><td>6.5</td></tr>
<tr><td>MCG_CANFDx_FDFECR</td><td>CAN FD ECC false error control register of MCU Config Group (channel 0,1,2)</td><td>6.6</td></tr>
<tr><td>CPG_CANFDx_FDFECR</td><td>CAN FD ECC false error control register of Common Peripheral #0 Group (channel0,1,2,3,4)</td><td>6.6</td></tr>
<tr><td></td><td></td><td></td></tr>
</table>
</td></tr>
</table>

Section	Change Results
CHAPTER33: CAN FD Message RAM ECC Function 6.Registers	Features of "For details" should be corrected as indicated by the shading below. (Error) Register Map Table 6-1 Register Map of CAN FD MESSAGE RAM ECC FUNCTION (Correct) Register Map Table 6-2 Register Map of CAN FD MESSAGE RAM ECC FUNCTION

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER33: CAN FD
Message RAM ECC Function
6.Registers</td><td>The function should be deleted as indicated by the shading below.

(Error)
Please refer to Hardware Manual or Data Sheet for more details.

Table 6-1 List of the Registers of the CAN FD Message RAM ECC Function
<table>
<tr><th>Abbreviated Register Name</th><th>Register Name</th><th>Reference</th></tr>
<tr><td>MCG_CANFDx_FD ECR</td><td>CAN FD ECC error control register of MCU Config Group (channel 0,1,2)</td><td>6.1</td></tr>
<tr><td>CPG_CANFDx_FD ECR</td><td>CAN FD ECC error control register of Common Peripheral #0 Group (channel 0,1,2,3,4)</td><td>6.1</td></tr>
<tr><td>MCG_CANFDx_FD ESR</td><td>CAN FD ECC error status register of MCU Config Group (channel 0,1,2)</td><td>6.2</td></tr>
<tr><td>CPG_CANFDx_FD ESR</td><td>CAN FD ECC error status register of Common Peripheral #0 Group (channel 0,1,2,3,4)</td><td>6.2</td></tr>
<tr><td>MCG_CANFDx_FD ESCR</td><td>CAN FD ECC error status clear register of MCU Config Group (channel 0,1,2)</td><td>6.3</td></tr>
<tr><td>CPG_CANFDx_FD ESCR</td><td>CAN FD ECC error status clear register of Common Peripheral #0 Group (channel 0,1,2,3,4)</td><td>6.3</td></tr>
<tr><td>MCG_CANFDx_FD DEAR</td><td>CAN FD ECC double-bit error address register of MCU Config Group (channel 0,1,2)</td><td>6.4</td></tr>
<tr><td>CPG_CANFDx_FD DEAR</td><td>CAN FD ECC double-bit error address register of Common Peripheral #0 Group (channel 0,1,2,3,4)</td><td>6.4</td></tr>
<tr><td>MCG_CANFDx_FD SEAR</td><td>CAN FD ECC single-bit error address register of MCU Config Group (channel 0,1,2)</td><td>6.5</td></tr>
<tr><td>CPG_CANFDx_FD SEAR</td><td>CAN FD ECC single-bit error address register of Common Peripheral #0 Group (channel 0,1,2,3,4)</td><td>6.5</td></tr>
<tr><td>MCG_CANFDx_FD FECR</td><td>CAN FD ECC false error control register of MCU Config Group (channel 0,1,2)</td><td>6.6</td></tr>
<tr><td>CPG_CANFDx_FD FECR</td><td>CAN FD ECC false error control register of Common Peripheral #0 Group (channel 0,1,2,3,4)</td><td>6.6</td></tr>
</table>

(Correct)
−Please refer to Hardware Manual or Data Sheet for more details.</td></tr>
</table>

Section	Change Results
CHAPTER33: CAN FD Message RAM ECC Function 6.Registers 6.1.CAN FD ECC Error Control Register (FDECR)	The function should be deleted as indicated by the shading below. (Error) The CAN FD ECC error control register (FDECR) is used to set whether to enable the interrupt when single-bit error correction or double-bit error detection occurs during the ECC check. It is also used to set ECC error detection stop and response to the CPU. CAN FD ECC error control register (FDECR) (Correct) The CAN FD ECC error control register (FDECR) is used to set whether to enable the interrupt when single-bit error correction or double-bit error detection occurs during the ECC check. It is also used to set ECC error detection stop and response to the CPU.
CHAPTER33: CAN FD Message RAM ECC Function 6.Registers 6.2.CAN FD ECC Error Status Register (FDESR)	Features of "For details" should be corrected as indicated by the shading below. (Error) The CAN FD ECC error status register (FDESR) displays whether a single-bit error has been corrected in the ECC check and whether a double-bit error has been detected. When any bit in this register becomes "1", it remains "1" unless it is cleared by using CAN FD ECC error status clear register (FDESCR). CAN FD ECC error status register (FDESR) (Correct) The CAN FD ECC error status register (FDESR) displays whether a single-bit error has been corrected in the ECC check and whether a double-bit error has been detected. When any bit in this register becomes "1", it remains "1" unless it is cleared by using CAN FD ECC error status clear register (FDESCR).
CHAPTER33: CAN FD Message RAM ECC Function 6.Registers 6.3.CAN FD ECC Error Status Clear Register (FDESCR)	The function should be deleted as indicated by the shading below. (Error) The CAN FD ECC error status clear register (FDESCR) is used to clear bits in the CAN FD ECC error status register. CAN FD ECC error status clear register (FDESCR) (Correct) The CAN FD ECC error status clear register (FDESCR) is used to clear bits in the CAN FD ECC error status register.

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER33: CAN FD Message RAM ECC Function 6.Registers 6.4.CAN FD ECC Double-bit Error Address Register (FDDEAR)</td><td>The function should be deleted as indicated by the shading below.

(Error)
The CAN FD ECC double-bit error address register (FDDEAR) indicates the address which a double-bit error has occurred at the Message RAM during the ECC check. This register is valid when the DEI bit in the CAN FD ECC error status register (FDESR) is "1". While the DEI bit in the CAN FD ECC error status register (FDESR) is "1", the value of this register is retained.
CAN FD ECC double-bit error address register (FDDEAR)

(Correct)
The CAN FD ECC double-bit error address register (FDDEAR) indicates the address which a double-bit error has occurred at the Message RAM during the ECC check. This register is valid when the DEI bit in the CAN FD ECC error status register (FDESR) is "1". While the DEI bit in the CAN FD ECC error status register (FDESR) is "1", the value of this register is retained.</td></tr>
<tr><td>CHAPTER33: CAN FD Message RAM ECC Function 6.Registers 6.4.CAN FD ECC Double-bit Error Address Register (FDDEAR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit15:0] DRA15 to DRA0: Double-bit error Message RAM address bits
<table>
<tr><th>bit15:0</th><th>Description</th></tr>
<tr><td>0x0000</td><td>Indicates that a double-bit error has occurred at Message RAM address 0.</td></tr>
<tr><td>0x0004</td><td>Indicates that a double-bit error has occurred at Message RAM address 4.</td></tr>
<tr><td>0x0008</td><td>Indicates that a double-bit error has occurred at Message RAM address 8.</td></tr>
<tr><td>...</td><td>.....</td></tr>
</table>
(Correct)
[bit15:0] DRA15 to DRA0: Double-Bit Error Message RAM Address Bits
<table>
<tr><th>DRA15 to DRA0</th><th>Description</th></tr>
<tr><td>0x0000</td><td>Indicates that a double-bit error has occurred at Message RAM address 0.</td></tr>
<tr><td>0x0004</td><td>Indicates that a double-bit error has occurred at Message RAM address 4.</td></tr>
<tr><td>0x0008</td><td>Indicates that a double-bit error has occurred at Message RAM address 8.</td></tr>
<tr><td>...</td><td>.....</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER33: CAN FD Message RAM ECC Function 6.Registers 6.5.CAN FD ECC Single-bit Error Address Register (FDSEAR)</td><td>The function should be deleted as indicated by the shading below.

(Error)
The CAN FD ECC single-bit error address register (FDSEAR) indicates the address which a single-bit error has occurred at the Message RAM during the ECC check. This register is valid when the SEI bit in the CAN FD ECC error status register (FDESR) is "1". While the SEI bit in the CAN FD ECC error status register (FDESR) is "1", the value of this register is retained.
CAN FD ECC single-bit error address register (FDSEAR)

(Correct)
The CAN FD ECC single-bit error address register (FDSEAR) indicates the address which a single-bit error has occurred at the Message RAM during the ECC check. This register is valid when the SEI bit in the CAN FD ECC error status register (FDESR) is "1". While the SEI bit in the CAN FD ECC error status register (FDESR) is "1", the value of this register is retained.</td></tr>
<tr><td>CHAPTER33: CAN FD Message RAM ECC Function 6.Registers 6.5.CAN FD ECC Single-bit Error Address Register (MCG_CANFDx_FDSEAR, CPG_CANFDx_FDSEAR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit15:0] SRA15 to SRA0: Single-bit error Message RAM address bits
<table>
<tr><th>bit15:0</th><th>Description</th></tr>
<tr><td>0x0000</td><td>Indicates that a single-bit error has occurred at Message RAM address 0.</td></tr>
<tr><td>0x0004</td><td>Indicates that a single-bit error has occurred at Message RAM address 4.</td></tr>
<tr><td>0x0008</td><td>Indicates that a single-bit error has occurred at Message RAM address 8.</td></tr>
<tr><td>...</td><td>.....</td></tr>
</table>
(Correct)
[bit15:0] SRA15 to SRA0: Single-Bit Error Message RAM Address Bits
<table>
<tr><th>SRA15 to SRA0</th><th>Description</th></tr>
<tr><td>0x0000</td><td>Indicates that a single-bit error has occurred at Message RAM address 0.</td></tr>
<tr><td>0x0004</td><td>Indicates that a single-bit error has occurred at Message RAM address 4.</td></tr>
<tr><td>0x0008</td><td>Indicates that a single-bit error has occurred at Message RAM address 8.</td></tr>
<tr><td>...</td><td>.....</td></tr>
</table></td></tr>
<tr><td>CHAPTER33: CAN FD Message RAM ECC Function 6.Registers 6.6.CAN FD ECC False Error Control Register (FDFECR)</td><td>The function should be deleted as indicated by the shading below.

(Error)
The CAN FD ECC false error control register (FDFECR) is used to specify the bytes and bits in which an error is to be generated. This register can be set when the TM_CANFERR signal is "H".
CAN FD ECC false error control register (FDFECR)

(Correct)
The CAN FD ECC false error control register (FDFECR) is used to specify the bytes and bits in which an error is to be generated. This register can be set when the TM_CANFERR signal is "H".</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER33: CAN FD
Message RAM
ECC Function
6.Registers
6.6.CAN FD
ECC False Error
Control Register
(FDFECR)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
<table>
<tr><td>bit</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td>Field</td><td>FERR</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>R/W attribute</td><td>R/W</td><td>R0,W0</td><td>R0,W0</td><td>R0,W0</td><td>R0,W0</td><td>R0,W0</td><td>R0,W0</td><td>R0,W0</td></tr>
<tr><td>Protection attribute</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>Initial value</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
<table>
<tr><td>bit</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td>Field</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>EY2</td><td>EY1</td><td>EY0</td></tr>
<tr><td>R/W attribute</td><td>R0,W0</td><td>R0,W0</td><td>R0,W0</td><td>R0,W0</td><td>R0,W0</td><td>R/W</td><td>R/W</td><td>R/W</td></tr>
<tr><td>Protection attribute</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>Initial value</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
(Correct)
<table>
<tr><td>bit</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td>Field</td><td>FERR</td><td colspan="7">Reserved</td></tr>
<tr><td>R/W attribute</td><td>R/W</td><td colspan="7">R0,W0</td></tr>
<tr><td>Protection attribute</td><td>-</td><td colspan="7">-</td></tr>
<tr><td>Initial value</td><td>0</td><td colspan="7">0000000</td></tr>
</table>
<table>
<tr><td>bit</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td>Field</td><td colspan="5">Reserved</td><td>EY2</td><td>EY1</td><td>EY0</td></tr>
<tr><td>R/W attribute</td><td colspan="5">R0,W0</td><td>R/W</td><td>R/W</td><td>R/W</td></tr>
<tr><td>Protection attribute</td><td colspan="5">-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>Initial value</td><td colspan="5">00000</td><td>0</td><td>0</td><td>0</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER33: CAN FD
Message RAM ECC Function
6.Registers
6.6.CAN FD ECC False Error Control Register (FDFECR)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit18:16] EY2 to EY0: Error byte specification bits
<table>
<tr><th>bit18:16</th><th>Description</th></tr>
<tr><td>EY0</td><td>Sets bit15 to 0 in the Message RAM as error targets.</td></tr>
<tr><td>EY1</td><td>Sets bit31 to 16 in the Message RAM as error targets.</td></tr>
<tr><td>EY2</td><td>Sets bit38 to 32 in the Message RAM (bits for ECC) error targets.</td></tr>
</table>

(Correct)
[bit18:16] EY2 to EY0: Error Byte Specification Bits
<table>
<tr><th>bit</th><th>Description</th></tr>
<tr><td>EY0</td><td>Sets bit15 to 0 in the Message RAM as error targets.</td></tr>
<tr><td>EY1</td><td>Sets bit31 to 16 in the Message RAM as error targets.</td></tr>
<tr><td>EY2</td><td>Sets bit38 to 32 in the Message RAM (bits for ECC) error targets.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER33: CAN FD
Message RAM
ECC Function
6.Registers
6.6.CAN FD ECC False Error Control Register (FDFECR)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit15:0] EI15 to EI0: Error bit specification bits
<table>
<tr><th>bit15:0</th><th>Description</th></tr>
<tr><td>EI0</td><td>In the 2 bytes specified by EY2 to EY0, bit0 is an error target.</td></tr>
<tr><td>EI1</td><td>In the 2 bytes specified by EY2 to EY0, bit1 is an error target.</td></tr>
<tr><td>EI2</td><td>In the 2 bytes specified by EY2 to EY0, bit2 is an error target.</td></tr>
<tr><td>EI3</td><td>In the 2 bytes specified by EY2 to EY0, bit3 is an error target.</td></tr>
<tr><td>EI4</td><td>In the 2 bytes specified by EY2 to EY0, bit4 is an error target.</td></tr>
<tr><td>EI5</td><td>In the 2 bytes specified by EY2 to EY0, bit5 is an error target.</td></tr>
<tr><td>RI6</td><td>In the 2 bytes specified by EY2 to EY0, bit6 is an error target.</td></tr>
<tr><td>EI7</td><td>In the 2 bytes specified by EY1 to EY0, bit7 is an error target.</td></tr>
<tr><td>EI8</td><td>In the 2 bytes specified by EY1 to EY0, bit8 is an error target.</td></tr>
<tr><td>EI9</td><td>In the 2 bytes specified by EY1 to EY0, bit9 is an error target.</td></tr>
<tr><td>EI10</td><td>In the 2 bytes specified by EY1 to EY0, bit10 is an error target.</td></tr>
<tr><td>EI11</td><td>In the 2 bytes specified by EY1 to EY0, bit11 is an error target.</td></tr>
<tr><td>EI12</td><td>In the 2 bytes specified by EY1 to EY0, bit12 is an error target.</td></tr>
<tr><td>EI13</td><td>In the 2 bytes specified by EY1 to EY0, bit13 is an error target.</td></tr>
<tr><td>EI14</td><td>In the 2 bytes specified by EY1 to EY0, bit14 is an error target.</td></tr>
<tr><td>EI15</td><td>In the 2 bytes specified by EY1 to EY0, bit15 is an error target.</td></tr>
</table>

(Correct)
[bit15:0] EI15 to EI0: Error Bit Specification Bits
<table>
<tr><th>bit</th><th>Description</th></tr>
<tr><td>EI0</td><td>In the 2 bytes specified by EY2 to EY0, bit0 is an error target.</td></tr>
<tr><td>EI1</td><td>In the 2 bytes specified by EY2 to EY0, bit1 is an error target.</td></tr>
<tr><td>EI2</td><td>In the 2 bytes specified by EY2 to EY0, bit2 is an error target.</td></tr>
<tr><td>EI3</td><td>In the 2 bytes specified by EY2 to EY0, bit3 is an error target.</td></tr>
<tr><td>EI4</td><td>In the 2 bytes specified by EY2 to EY0, bit4 is an error target.</td></tr>
<tr><td>EI5</td><td>In the 2 bytes specified by EY2 to EY0, bit5 is an error target.</td></tr>
<tr><td>EI6</td><td>In the 2 bytes specified by EY2 to EY0, bit6 is an error target.</td></tr>
<tr><td>EI7</td><td>In the 2 bytes specified by EY1 to EY0, bit7 is an error target.</td></tr>
<tr><td>EI8</td><td>In the 2 bytes specified by EY1 to EY0, bit8 is an error target.</td></tr>
<tr><td>EI9</td><td>In the 2 bytes specified by EY1 to EY0, bit9 is an error target.</td></tr>
<tr><td>EI10</td><td>In the 2 bytes specified by EY1 to EY0, bit10 is an error target.</td></tr>
<tr><td>EI11</td><td>In the 2 bytes specified by EY1 to EY0, bit11 is an error target.</td></tr>
<tr><td>EI12</td><td>In the 2 bytes specified by EY1 to EY0, bit12 is an error target.</td></tr>
<tr><td>EI13</td><td>In the 2 bytes specified by EY1 to EY0, bit13 is an error target.</td></tr>
<tr><td>EI14</td><td>In the 2 bytes specified by EY1 to EY0, bit14 is an error target.</td></tr>
<tr><td>EI15</td><td>In the 2 bytes specified by EY1 to EY0, bit15 is an error target.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER35: Overview of Multi-function Serial Interface
1.Overview of the UART (Asynchronous Serial Interface)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
<table><tr><th></th><th></th><th>Function</th></tr><tr><td>1</td><td>Data</td><td>− Full duplex, double buffering (when FIFOs are not used)
− Transmission and reception FIFOs (16 bytes each) (when FIFOs are used)</td></tr></table>
(Correct)
<table><tr><th></th><th>Function</th><th>Description</th></tr><tr><td>1</td><td>Data</td><td>− Full duplex, double buffering (when FIFOs are not used)
− Transmission and reception FIFOs (64 bytes each) (when FIFOs are used)</td></tr></table></td></tr>
<tr><td>CHAPTER35: Overview of Multi-function Serial Interface
1.Overview of the UART (Asynchronous Serial Interface)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
<table><tr><td>14</td><td>FIFO option</td><td>− Transmission and reception FIFOs are provided (16 bytes for the transmission FIFO and 16 bytes for the reception FIFO).
− Transmission FIFO and reception FIFO can be selected.
− Transmission data can be retransmitted.
− The timing of the reception FIFO interrupt can be changed by software.
− Independent FIFO reset is supported.</td></tr></table>
(Correct)
<table><tr><td>14</td><td>FIFO option</td><td>− Transmission and reception FIFOs are provided (64 bytes for the transmission FIFO and 64 bytes for the reception FIFO).
− Transmission FIFO and reception FIFO can be selected.
− Transmission data can be retransmitted.
− The timing of the reception FIFO interrupt can be changed by software.
− Independent FIFO reset is supported.</td></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER35: Overview of Multi-function Serial Interface
2.Overview of the CSIO (Clock Synchronous Serial Interface)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
<table><tr><th></th><th></th><th>Function</th></tr><tr><td>1</td><td>Data buffer</td><td>− Full duplex, double buffering (when FIFOs are not used)
− Transmission and reception FIFOs (16 bytes each) (when FIFOs are used)</td></tr></table>
(Correct)
<table><tr><th></th><th>Function</th><th>Description</th></tr><tr><td>1</td><td>Data buffer</td><td>− Full duplex, double buffering (when FIFOs are not used)
− Transmission and reception FIFOs (64 bytes each) (when FIFOs are used)</td></tr></table></td></tr>
<tr><td>CHAPTER35: Overview of Multi-function Serial Interface
2.Overview of the CSIO (Clock Synchronous Serial Interface)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
<table><tr><td>12</td><td>FIFO option</td><td>− Transmission and reception FIFOs are provided (16 bytes for the transmission FIFO and 16 bytes for the reception FIFO).
− Transmission FIFO and reception FIFO can be selected.
− Transmission data can be retransmitted.
− The timing of the reception FIFO interrupt can be changed by software.
− Independent FIFO reset is supported.</td></tr></table>
(Correct)
<table><tr><td>12</td><td>FIFO option</td><td>− Transmission and reception FIFOs are provided (64 bytes for the transmission FIFO and 64 bytes for the reception FIFO).
− Transmission FIFO and reception FIFO can be selected.
− Transmission data can be retransmitted.
− The timing of the reception FIFO interrupt can be changed by software.
− Independent FIFO reset is supported.</td></tr></table></td></tr>
<tr><td>CHAPTER35: Overview of Multi-function Serial Interface
3.Overview of the LIN Interface (v2.1) (LIN Communication Control Interface (v2.1))
3.1.Manual Mode</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
The LIN interface (v2.1) (LIN communication control interface (v2.1)) provides the functions to support the LIN bus. In addition, this interface has FIFOs (16 bytes each) for transmission and reception.

(Correct)
The LIN interface (v2.1) (LIN communication control interface (v2.1)) provides the functions to support the LIN bus. In addition, this interface has FIFOs (64 bytes each) for transmission and reception.</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER35: Overview of Multi-function Serial Interface
3.Overview of the LIN Interface (v2.1) (LIN Communication Control Interface (v2.1))
3.1.Manual Mode</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
<table><tr><th></th><th></th><th>Function</th></tr><tr><td>1</td><td>Data buffer</td><td>− Full duplex, double buffering (when FIFOs are not used)
− Transmission and reception FIFOs (16 bytes each) (when FIFOs are used)</td></tr></table>
(Correct)
<table><tr><th></th><th>Function</th><th>Description</th></tr><tr><td>1</td><td>Data buffer</td><td>− Full duplex, double buffering (when FIFOs are not used)
− Transmission and reception FIFOs (64 bytes each) (when FIFOs are used)</td></tr></table></td></tr>
<tr><td>CHAPTER35: Overview of Multi-function Serial Interface
3.Overview of the LIN Interface (v2.1) (LIN Communication Control Interface (v2.1))
3.1.Manual Mode</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
<table><tr><td>12</td><td>FIFO option</td><td>− Transmission and reception FIFOs are provided (16 bytes for the transmission FIFO and 16 bytes for the reception FIFO).
− Transmission FIFO and reception FIFO can be selected.
− Transmission data can be retransmitted.
− The timing of the reception FIFO interrupt can be changed by software.
− Independent FIFO reset is supported.</td></tr></table>
(Correct)
<table><tr><td>12</td><td>FIFO option</td><td>− Transmission and reception FIFOs are provided (64 bytes for the transmission FIFO and 64 bytes for the reception FIFO).
− Transmission FIFO and reception FIFO can be selected.
− Transmission data can be retransmitted.
− The timing of the reception FIFO interrupt can be changed by software.
− Independent FIFO reset is supported.</td></tr></table></td></tr>
<tr><td>CHAPTER35: Overview of Multi-function Serial Interface
3.Overview of the LIN Interface (v2.1) (LIN Communication Control Interface (v2.1))
3.2.Assist Mode</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
The LIN interface (v2.1) (LIN communication control interface (v2.1)) provides the functions to support the LIN bus. The header section can be automatically transmitted/detected in LIN communication. In addition, this interface has FIFOs (16 bytes each) for transmission and reception.

(Correct)
The LIN interface (v2.1) (LIN communication control interface (v2.1)) provides the functions to support the LIN bus. The header section can be automatically transmitted/detected in LIN communication. In addition, this interface has FIFOs (64 bytes each) for transmission and reception.</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER35: Overview of Multi-function Serial Interface
3.Overview of the LIN Interface (v2.1) (LIN Communication Control Interface (v2.1))
3.2.Assist Mode</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
<table><tr><th></th><th></th><th>Function</th></tr><tr><td>1</td><td>Data buffer</td><td>− Full duplex, double buffering (when FIFOs are not used)
− Transmission and reception FIFOs (16 bytes each) (when FIFOs are used)</td></tr></table>
(Correct)
<table><tr><th></th><th>Function</th><th>Description</th></tr><tr><td>1</td><td>Data buffer</td><td>− Full duplex, double buffering (when FIFOs are not used)
− Transmission and reception FIFOs (64 bytes each) (when FIFOs are used)</td></tr></table></td></tr>
<tr><td>CHAPTER35: Overview of Multi-function Serial Interface
3.Overview of the LIN Interface (v2.1) (LIN Communication Control Interface (v2.1))
3.2.Assist Mode</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
<table><tr><td>12</td><td>FIFO option</td><td>− Transmission and reception FIFOs are provided (16 bytes for the transmission FIFO and 16 bytes for the reception FIFO).
− Transmission FIFO and reception FIFO can be selected.
− Transmission data can be retransmitted.
− The timing of the transmission/reception FIFO interrupt can be changed by software.
− Independent FIFO reset is supported.</td></tr></table>
(Correct)
<table><tr><td>12</td><td>FIFO option</td><td>− Transmission and reception FIFOs are provided (64 bytes for the transmission FIFO and 64 bytes for the reception FIFO).
− Transmission FIFO and reception FIFO can be selected.
− Transmission data can be retransmitted.
− The timing of the transmission/reception FIFO interrupt can be changed by software.
− Independent FIFO reset is supported.</td></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER35: Overview of Multi-function Serial Interface
4.Overview of the I2C Interface (I2C Communication Control Interface)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
The I²C interface (I²C communication control interface) supports the I²C bus and operates as a master/slave device on the I²C bus. In addition, this interface has FIFOs (16 bytes each) for transmission and reception.

(Correct)
The I²C interface (I²C communication control interface) supports the I²C bus and operates as a master/slave device on the I²C bus. In addition, this interface has FIFOs (64 bytes each) for transmission and reception.</td></tr>
<tr><td>CHAPTER35: Overview of Multi-function Serial Interface
4.Overview of the I2C Interface (I2C Communication Control Interface)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)<table><tr><th></th><th></th><th>Function</th></tr><tr><td>1</td><td>Data buffer</td><td>− Full duplex, double buffering (when FIFOs are not used)
− Transmission and reception FIFOs (16 bytes each) (when FIFOs are used)</td></tr></table>
(Correct)<table><tr><th></th><th>Function</th><th>Description</th></tr><tr><td>1</td><td>Data buffer</td><td>− Full duplex, double buffering (when FIFOs are not used)
− Transmission and reception FIFOs (64 bytes each) (when FIFOs are used)</td></tr></table></td></tr>
<tr><td>CHAPTER35: Overview of Multi-function Serial Interface
4.Overview of the I2C Interface (I2C Communication Control Interface)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)<table><tr><td>10</td><td>FIFO</td><td>− Transmission and reception FIFOs are provided (16 bytes for the transmission FIFO and 16 bytes for the reception FIFO).
− Transmission FIFO and reception FIFO can be selected.
− Transmission data can be retransmitted.
− The timing of the reception FIFO interrupt can be changed by software.
− Independent FIFO reset is supported.</td></tr></table>
(Correct)<table><tr><td>10</td><td>FIFO</td><td>− Transmission and reception FIFOs are provided (64 bytes for the transmission FIFO and 64 bytes for the reception FIFO).
− Transmission FIFO and reception FIFO can be selected.
− Transmission data can be retransmitted.
− The timing of the reception FIFO interrupt can be changed by software.
− Independent FIFO reset is supported.</td></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER35: Overview of Multi-function Serial Interface 6.Registers of the Multi-function Serial Interface</td><td>Revised “This section lists the multi-function serial interface registers” as follows.

(Correct)

<table>
<tr><th>operation mode</th><th>register select signal</th><th>DATAO[31:24]/ DATAI[31:24]</th><th>DATAO[23:16]/ DATAI[23:16]</th><th>DATAO[15:8]/ DATAI[15:8]</th><th>DATAO[7:0]/ DATAI[7:0]</th></tr>
<tr><td>mode0 to 3</td><td rowspan="2">SSRSEL, ESCRSEL SCRSEL, SMRSEL</td><td rowspan="2">SSR[7:0]</td><td>ESCR[7:0]</td><td>SCR[7:0]</td><td rowspan="2">SMR[7:0]</td></tr>
<tr><td>mode4</td><td>IBSR[7:0]</td><td>IBCR[7:0]</td></tr>
<tr><td>mode0, 1</td><td rowspan="3">RDRSEL[1:0]</td><td>-</td><td>-</td><td>RDR[9]</td><td rowspan="3">RDR[7:0]</td></tr>
<tr><td>mode2</td><td>RDR[31:24]</td><td>RDR[23:16]</td><td>RDR[15:8]</td></tr>
<tr><td>mode3, 4</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>mode0 to 4</td><td>STMRSEL[1:0] SACSRSEL[1:0]</td><td>STMR[15:8]</td><td>STMR[7:0]</td><td>SACSR[15:8]</td><td>SACSR[7:0]</td></tr>
<tr><td>mode0, 1</td><td rowspan="3">SCSCRSEL [1:0] STMCRSEL [1:0]</td><td>-</td><td>-</td><td rowspan="3">STMCR[15:8]</td><td rowspan="3">STMCR[7:0]</td></tr>
<tr><td>mode2</td><td>SCSCR[15:8]</td><td>SCSCR[7:0]</td></tr>
<tr><td>mode3</td><td>SFUR[15:8]</td><td>SFUR[7:0]</td></tr>
<tr><td>mode4</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>mode0, 1</td><td rowspan="4">SCSTRSEL [3:0]</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>mode2</td><td>SCSTR[31:24]</td><td>SCSTR[23:16]</td><td>SCSTR[15:8]</td><td>SCSTR[7:0]</td></tr>
<tr><td>mode3</td><td>LAMSR[7:0]</td><td>LAMCR[7:0]</td><td>SFLR[15:8]</td><td>SFLR[7:0]</td></tr>
<tr><td>mode4</td><td>-</td><td>-</td><td>EIBCR[7:0]</td><td>NFCR[7:0]</td></tr>
<tr><td>mode0, 1</td><td rowspan="3">SCSFRSEL [2:0]</td><td rowspan="3">-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>mode2</td><td>SCSFR[23:16]</td><td>SCSFR[15:8]</td><td>SCSFR[7:0]</td></tr>
<tr><td>mode3, 4</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>mode0, 1</td><td rowspan="4">TBYTESEL [3:0]</td><td>-</td><td>-</td><td>-</td><td rowspan="2">TBYTE[7:0]</td></tr>
<tr><td>mode2</td><td>TBYTE[13:24]</td><td>TBYTE[23:16]</td><td>TBYTE[15:8]</td></tr>
<tr><td>mode3</td><td>LAMESR[7:0]</td><td>LAMERT[7:0]</td><td>LAMIER[7:0]</td><td>LAMRID[7:0]/ LAMTID[7:0]</td></tr>
<tr><td>mode4</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>mode0 to 3</td><td rowspan="2">ISMKSEL ISBASEL BGR1SEL BGR0SEL</td><td rowspan="2">-</td><td rowspan="2">-</td><td rowspan="2">BGR1[7:0]</td><td rowspan="2">BGR0[7:0]</td></tr>
<tr><td>mode4</td></tr>
<tr><td>mode0 to 4</td><td>FBYTE2SEL FBYTE1SEL FCR1SEL FCR0SEL</td><td>FBYTE2[7:0]</td><td>FBYTE1[7:0]</td><td>FCR1[7:0]</td><td>FCR0[7:0]</td></tr>
<tr><td>mode0 to 4</td><td>FTICR2SEL FTICR1SEL</td><td>-</td><td>-</td><td>FTICR2[7:0]</td><td>FTICR1[7:0]</td></tr>
</table>
3050462701</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER35: Overview of Multi-function Serial Interface 6.Registers of the Multi-function Serial Interface</td><td>Revised “List of clear registers” as follows.

(Correct)
<table>
<tr><th>operation mode</th><th>register select signal</th><th>DATAO[31:24]/ DATAI[31:24]</th><th>DATAO[23:16]/ DATAI[23:16]</th><th>DATAO[15:8]/ DATAI[15:8]</th><th>DATAO[7:0]/ DATAI[7:0]</th></tr>
<tr><td>mode3</td><td rowspan="2">SSRSEL, ESCRSEL SCRSEL, SMRSEL</td><td>SSRC[7:0]</td><td>ESCRC[7:0]</td><td>SCRC[7:0]</td><td>SMRC[7:0]</td></tr>
<tr><td>mode4</td><td>SSRC[7:0]</td><td>IBSRC[7:0]</td><td>IBCRC[7:0]</td><td>SMRC[7:0]</td></tr>
<tr><td>mode0 to 4</td><td>STMRSEL[1:0] SACSRSEL[1:0]</td><td>-</td><td>-</td><td>SACSRC[15:8]</td><td>SACSRC[7:0]</td></tr>
<tr><td>mode3</td><td>SCSTRSEL[3:0]</td><td>LAMSRC[7:0]</td><td>LAMCRC[7:0]</td><td>-</td><td>-</td></tr>
<tr><td>mode3</td><td>TBYTESEL[3:0]</td><td>LAMESRC[7:0]</td><td>-</td><td>LAMIERC[7:0]</td><td>-</td></tr>
<tr><td>mode0 to 4</td><td>FBYTE2SEL FBYTE1SEL FCR1SEL FCR0SEL</td><td>-</td><td>-</td><td>FCR1C[7:0]</td><td>FCR0C[7:0]</td></tr>
</table>
3050462711</td></tr>
<tr><td>CHAPTER35: Overview of Multi-function Serial Interface 6.Registers of the Multi-function Serial Interface</td><td>Revised “List of set registers” as follows.

(Correct)
<table>
<tr><th>operation mode</th><th>register select signal</th><th>DATAO[31:24]/ DATAI[31:24]</th><th>DATAO[23:16]/ DATAI[23:16]</th><th>DATAO[15:8]/ DATAI[15:8]</th><th>DATAO[7:0]/ DATAI[7:0]</th></tr>
<tr><td>mode3</td><td rowspan="2">SSRSEL, ESCRSEL SCRSEL, SMRSEL</td><td>SSRS[7:0]</td><td>ESCRS[7:0]</td><td>SCRS[7:0]</td><td>SMRS[7:0]</td></tr>
<tr><td>mode4</td><td>SSRS[7:0]</td><td>-</td><td>IBCRS[7:0]</td><td>SMRS[7:0]</td></tr>
<tr><td>mode0 to 4</td><td>STMRSEL[1:0] SACSRSEL[1:0]</td><td>-</td><td>-</td><td>SACSRS[15:8]</td><td>SACSRS[7:0]</td></tr>
<tr><td>mode3</td><td>SCSTRSEL[3:0]</td><td>-</td><td>LAMCRS[7:0]</td><td>-</td><td>-</td></tr>
<tr><td>mode3</td><td>TBYTESEL[3:0]</td><td>-</td><td>-</td><td>LAMIERS[7:0]</td><td>-</td></tr>
<tr><td>mode0 to 4</td><td>FBYTE2SEL FBYTE1SEL FCR1SEL FCR0SEL</td><td>-</td><td>-</td><td>FCR1S[7:0]</td><td>FCR0S[7:0]</td></tr>
</table>
3050462721</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER36: UART (Asynchronous Serial Interface) 1.Overview of the UART (Asynchronous Serial Interface)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
<table>
<tr><th></th><th></th><th>Function</th></tr>
<tr><td>1</td><td>Data</td><td>− Full-duplex double buffer (when FIFO is not used)
− Transmission and reception FIFOs (64 bytes for each) (when FIFO is used)</td></tr>
</table>

(Correct)
<table>
<tr><th></th><th>Function</th><th>Description</th></tr>
<tr><td>1</td><td>Data</td><td>− Full-duplex double buffer (when FIFO is not used)
− Transmission and reception FIFOs (64 bytes for each) (when FIFO is used)</td></tr>
</table>
</td></tr>
<tr><td>CHAPTER36: UART (Asynchronous Serial Interface) 10.Registers of UART (Asynchronous Serial Interface) 10.1.Serial Control Register (SCR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
When it is read, "0" is always read.
<table>
<tr><th rowspan="2">UPCL</th><th colspan="2">Programmable Clear Bit</th></tr>
<tr><th>Write</th><th>Read</th></tr>
</table>

(Correct)
When it is read, "0" is always read.
<table>
<tr><th rowspan="2">Bit</th><th colspan="2">Description</th></tr>
<tr><th>Write</th><th>Read</th></tr>
</table>
</td></tr>
<tr><td>CHAPTER36: UART (Asynchronous Serial Interface) 10.Registers of UART (Asynchronous Serial Interface) 10.1.Serial Control Register (SCR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■When ECR:EISEL = 1, and the RIE bit and the reception data flag bit (SSR:RDRF) are "1", a reception interrupt request is issued.
<table>
<tr><th>RIE</th><th>Reception Interrupt Enable Bit</th></tr>
</table>

(Correct)
■When ECR:EISEL = 1, and the RIE bit and the reception data flag bit (SSR:RDRF) are "1", a reception interrupt request is issued.
<table>
<tr><th>Bit</th><th>Description</th></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER36: UART (Asynchronous Serial Interface) 10.Registers of UART (Asynchronous Serial Interface) 10.1.Serial Control Register (SCR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■When ECR:EISEL = 1, and the TIE bit and the SSR:TDRE bit are "1", a transmission interrupt request is issued.
<table><tr><th>TIE</th><th>Transmission Interrupt Enable Bit</th></tr></table>
(Correct)
■When ECR:EISEL = 1, and the TIE bit and the SSR:TDRE bit are "1", a transmission interrupt request is issued.
<table><tr><th>Bit</th><th>Description</th></tr></table></td></tr>
<tr><td>CHAPTER36: UART (Asynchronous Serial Interface) 10.Registers of UART (Asynchronous Serial Interface) 10.1.Serial Control Register (SCR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■A transmission bus idle interrupt request is issued when the TBIE bit and the TBI bit are "1".
<table><tr><th>TBIE</th><th>Transmission Bus Idle Interrupt Enable Bit</th></tr></table>
(Correct)
■A transmission bus idle interrupt request is issued when the TBIE bit and the TBI bit are "1".
<table><tr><th>Bit</th><th>Description</th></tr></table></td></tr>
<tr><td>CHAPTER36: UART (Asynchronous Serial Interface) 10.Registers of UART (Asynchronous Serial Interface) 10.1.Serial Control Register (SCR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■"1": Enable reception operation.
<table><tr><th>RXE</th><th>Reception Enable Bit</th></tr></table>
(Correct)
■"1": Enable reception operation.
<table><tr><th>Bit</th><th>Description</th></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER36: UART (Asynchronous Serial Interface) 10.Registers of UART (Asynchronous Serial Interface) 10.1.Serial Control Register (SCR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■"1": Enable transmission operation.<table><tr><th>TXE</th><th>Transmission Enable Bit</th></tr></table>
(Correct)
■"1": Enable transmission operation.<table><tr><th>Bit</th><th>Description</th></tr></table></td></tr>
<tr><td>CHAPTER36: UART (Asynchronous Serial Interface) 10.Registers of UART (Asynchronous Serial Interface) 10.2.Serial Mode Register (SMR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit7:5] MD2, MD1, MD0 : Operation mode setting bits
These bits set the operation mode of the Asynchronous Serial Interface.
"0b000": Set operation mode 0 (asynchronous normal mode).
"0b001": Set operation mode 1 (asynchronous multi-processor mode).
"0b010": Set operation mode 2 (clock synchronous mode).
"0b011": Set operation mode 3 (LIN communication mode).
"0b100": Set operation mode 4 (I2C mode).
This section describes the registers and their operations in operation mode 0 (asynchronous normal mode) and operation mode 1 (asynchronous multi-processor mode).<table><tr><th>MD2</th><th>MD1</th><th>MD0</th><th>Operation Mode Setting Bit</th></tr></table>
(Correct)
[bit7:5] MD[2:0]: Operation Mode Setting Bits
These bits set the operation mode of the Asynchronous Serial Interface.
0b000: Set operation mode 0 (asynchronous normal mode).
0b001: Set operation mode 1 (asynchronous multi-processor mode).
0b010: Set operation mode 2 (clock synchronous mode).
0b011: Set operation mode 3 (LIN communication mode).
0b100: Set operation mode 4 (I2C mode).
This section describes the registers and their operations in operation mode 0 (asynchronous normal mode) and operation mode 1 (asynchronous multi-processor mode).<table><tr><th>Bits</th><th>Description</th></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER36: UART (Asynchronous Serial Interface) 10.Registers of UART (Asynchronous Serial Interface) 10.2.Serial Mode Register (SMR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
"1": Set the SIN pin for external interrupts.<table><tr><td>WUCR</td><td>WAKE UP Control Bit</td></tr></table>
(Correct)
"1": Set the SIN pin for external interrupts.<table><tr><td>Bit</td><td>Description</td></tr></table></td></tr>
<tr><td>CHAPTER36: UART (Asynchronous Serial Interface) 10.Registers of UART (Asynchronous Serial Interface) 10.2.Serial Mode Register (SMR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
Settings for which SBL is "1" and ESCR:ESBL is "1": The stop bit length is set to 4 bits.<table><tr><td>SBL</td><td>Stop Bit Length Selection Bit</td></tr></table>
(Correct)
Settings for which SBL is "1" and ESCR:ESBL is "1": The stop bit length is set to 4 bits.<table><tr><td>Bit</td><td>Description</td></tr></table></td></tr>
<tr><td>CHAPTER36: UART (Asynchronous Serial Interface) 10.Registers of UART (Asynchronous Serial Interface) 10.2.Serial Mode Register (SMR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
This bit selects between whether the lowest bit is transferred first (LSB first, BDS = 0) or the highest bit is transferred first (MSB first, BDS = 1), for transfer serial data.<table><tr><td>BDS</td><td>Transfer Direction Selection Bit</td></tr></table>
(Correct)
This bit selects between whether the lowest bit is transferred first (LSB first, BDS = 0) or the highest bit is transferred first (MSB first, BDS = 1), for transfer serial data.<table><tr><td>Bit</td><td>Description</td></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER36: UART (Asynchronous Serial Interface) 10.Registers of UART (Asynchronous Serial Interface) 10.2.Serial Mode Register (SMR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
This bit enables/stops the output of serial data.
<table><tr><th>SOE</th><th>Serial Data Output Enable Bit</th></tr></table>
(Correct)
This bit enables/stops the output of serial data.
<table><tr><th>Bit</th><th>Description</th></tr></table></td></tr>
<tr><td>CHAPTER36: UART (Asynchronous Serial Interface) 10.Registers of UART (Asynchronous Serial Interface) 10.3.Serial Status Register (SSR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
If it is read, "0" is always read.
<table><tr><th rowspan="2">REC</th><th colspan="2">Reception Error Flag Clear Bit</th></tr><tr><th>Write</th><th>Read</th></tr></table>
(Correct)
If it is read, "0" is always read.
<table><tr><th rowspan="2">Bit</th><th colspan="2">Description</th></tr><tr><th>Write</th><th>Read</th></tr></table></td></tr>
<tr><td>CHAPTER36: UART (Asynchronous Serial Interface) 10.Registers of UART (Asynchronous Serial Interface) 10.3.Serial Status Register (SSR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■If this flag is set when the reception FIFO is used, the reception FIFO enable bits will be cleared, and no reception data will be stored in the reception FIFO.
<table><tr><th>PE</th><th>Parity Error Flag Bit</th></tr></table>
(Correct)
■If this flag is set when the reception FIFO is used, the reception FIFO enable bits will be cleared, and no reception data will be stored in the reception FIFO.
<table><tr><th>Bit</th><th>Description</th></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER36: UART (Asynchronous Serial Interface) 10.Registers of UART (Asynchronous Serial Interface) 10.3.Serial Status Register (SSR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■If this flag is set when the reception FIFO is used, the reception FIFO enable bits will be cleared, and no reception data will be stored in the reception FIFO.
<table><tr><td>FRE</td><td>Framing Error Flag Bit</td></tr></table>
(Correct)
■If this flag is set when the reception FIFO is used, the reception FIFO enable bits will be cleared, and no reception data will be stored in the reception FIFO.
<table><tr><td>Bit</td><td>Description</td></tr></table></td></tr>
<tr><td>CHAPTER36: UART (Asynchronous Serial Interface) 10.Registers of UART (Asynchronous Serial Interface) 10.3.Serial Status Register (SSR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■If this flag is set when the reception FIFO is used, the reception FIFO enable bits will be cleared, and no reception data will be stored in the reception FIFO.
<table><tr><td>ORE</td><td>Overrun Error Flag Bit</td></tr></table>
(Correct)
■If this flag is set when the reception FIFO is used, the reception FIFO enable bits will be cleared, and no reception data will be stored in the reception FIFO.
<table><tr><td>Bit</td><td>Description</td></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER36: UART (Asynchronous Serial Interface) 10.Registers of UART (Asynchronous Serial Interface) 10.3.Serial Status Register (SSR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■When ECR:RXBLKEN = 1, and the reception FIFO is used, this bit is cleared to "0" when the data count in the reception FIFO becomes equal to or less than the setting of the FBYTE register.
<table><tr><th>RDRF</th><th>Reception Data Full Flag Bit</th></tr></table>
Note:
−When the reception FIFO is used and RDRF has become "1", resetting the reception FIFO (FCR0:FCL2, FCL1 = "1") does not result in RDRF being set to "0". Therefore, in order to set RDRF to "0" after resetting the reception FIFO, perform a dummy read of the reception data register during the reception disable status (SCR:RXE = "0").

(Correct)
■When ECR:RXBLKEN = 1, and the reception FIFO is used, this bit is cleared to "0" when the data count in the reception FIFO becomes equal to or less than the setting of the FBYTE register.
<table><tr><th>Bit</th><th>Description</th></tr></table>
Note:
−When the reception FIFO is used and RDRF has become "1", resetting the reception FIFO (FCR0:FCL2, FCL1 = 1) does not result in RDRF being set to "0". Therefore, in order to set RDRF to "0" after resetting the reception FIFO, perform a dummy read of the reception data register during the reception disable status (SCR:RXE = 0).</td></tr>
<tr><td>CHAPTER36: UART (Asynchronous Serial Interface) 10.Registers of UART (Asynchronous Serial Interface) 10.3.Serial Status Register (SSR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■For set/reset timing of the TDRE bit when the transmission FIFO is used, see "CHAPTER 1:2.4 Occurrence of Interrupts and Flag Set Timing When the Transmission FIFO Is Used."
<table><tr><th>TDRE</th><th>Transmission Data Empty Flag Bit</th></tr></table>
(Correct)
■For set/reset timing of the TDRE bit when the transmission FIFO is used, see "CHAPTER 1:2.4 Occurrence of Interrupts and Flag Set Timing When the Transmission FIFO Is Used."
<table><tr><th>Bit</th><th>Description</th></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER36: UART (Asynchronous Serial Interface) 10.Registers of UART (Asynchronous Serial Interface) 10.3.Serial Status Register (SSR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■A transmission interrupt request is issued if this bit is set to "1" and a transmission bus idle interrupt is enabled (SCR:TBIE = 1).
<table><tr><td>TBI</td><td>Transmission Bus Idle Flag Bit</td></tr></table>
(Correct)
■A transmission interrupt request is issued if this bit is set to "1" and a transmission bus idle interrupt is enabled (SCR:TBIE = 1).
<table><tr><td>Bit</td><td>Description</td></tr></table></td></tr>
<tr><td>CHAPTER36: UART (Asynchronous Serial Interface) 10.Registers of UART (Asynchronous Serial Interface) 10.4.Extended Communication Control Register (ESCR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■"1": Enable hardware flow control.
<table><tr><td>FLWEN</td><td>Flow Control Enable Bit</td></tr></table>
(Correct)
■"1": Enable hardware flow control.
<table><tr><td>Bit</td><td>Description</td></tr></table></td></tr>
<tr><td>CHAPTER36: UART (Asynchronous Serial Interface) 10.Registers of UART (Asynchronous Serial Interface) 10.4.Extended Communication Control Register (ESCR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
Settings for which SBL is "1" and ESCR:ESBL is "1": The stop bit length is set to 4 bits.
<table><tr><td>ESBL</td><td>Extended Stop Bit Length Selection Bit</td></tr></table>
(Correct)
Settings for which SBL is "1" and ESCR:ESBL is "1": The stop bit length is set to 4 bits.
<table><tr><td>Bit</td><td>Extended Stop Bit Length</td></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER36: UART (Asynchronous Serial Interface) 10.Registers of UART (Asynchronous Serial Interface) 10.4.Extended Communication Control Register (ESCR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
This bit selects the NRZ format or the inverted NRZ format as the serial data format.<table><tr><th>INV</th><th>Inverted Serial Data Format Bit</th></tr></table>
(Correct)
This bit selects the NRZ format or the inverted NRZ format as the serial data format.<table><tr><th>Bit</th><th>Description</th></tr></table></td></tr>
<tr><td>CHAPTER36: UART (Asynchronous Serial Interface) 10.Registers of UART (Asynchronous Serial Interface) 10.4.Extended Communication Control Register (ESCR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■"1": Parity bits are appended.<table><tr><th>PEN</th><th>Parity Enable Bit</th></tr></table>
(Correct)
■"1": Parity bits are appended.<table><tr><th>Bit</th><th>Description</th></tr></table></td></tr>
<tr><td>CHAPTER36: UART (Asynchronous Serial Interface) 10.Registers of UART (Asynchronous Serial Interface) 10.4.Extended Communication Control Register (ESCR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■"1": Set odd parity.<table><tr><th>P</th><th>Parity Selection Bit</th></tr></table>
(Correct)
■"1": Set odd parity.<table><tr><th>Bit</th><th>Description</th></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER36: UART (Asynchronous Serial Interface) 10.Registers of UART (Asynchronous Serial Interface) 10.4.Extended Communication Control Register (ESCR)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit2:0] L2, L1, L0 : Data length selection bits
These bits specify the data length of the transmission/reception data.
■"0b000": The data length is 8 bits.
■"0b001": The data length is 5 bits.
■"0b010": The data length is 6 bits.
■"0b011": The data length is 7 bits.
■"0b100": The data length is 9 bits.
<table>
<tr><th>L2</th><th>L1</th><th>L0</th><th>Data Length Selection Bit</th></tr>
</table>

(Correct)
[bit2:0] L[2:0]: Data Length Selection Bits
These bits specify the data length of the transmission/reception data.
■0b000: The data length is 8 bits.
■0b001: The data length is 5 bits.
■0b010: The data length is 6 bits.
■0b011: The data length is 7 bits.
■0b100: The data length is 9 bits.
<table>
<tr><th>Bits</th><th>Data Length</th></tr>
</table>
</td></tr>
<tr><td>CHAPTER36: UART (Asynchronous Serial Interface) 10.Registers of UART (Asynchronous Serial Interface) 10.6.Serial Auxiliary Control Status Register (SACSR)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
■This bit is set when the SACSRS:STSTS bit in the set register is set to "1".
<table>
<tr><th>STST</th><th>Serial Test Bit</th></tr>
</table>

(Correct)
■This bit is set when the SACSRS:STSTS bit in the set register is set to "1".
<table>
<tr><th>Bit</th><th>Serial Test Bit</th></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER36: UART (Asynchronous Serial Interface) 10.Registers of UART (Asynchronous Serial Interface) 10.6.Serial Auxiliary Control Status Register (SACSR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit10:9] TRG1-0 : Trigger Selection Bits
These bits select the method for detecting external trigger edges for activating the serial timer.
<table><tr><th>TRG1</th><th>TRG0</th><th>Edge Detection Method for External Trigger</th></tr></table>
(Correct)
[bit10:9] TRG[1:0]: Trigger Selection Bits
These bits select the method for detecting external trigger edges for activating the serial timer.
<table><tr><th>Bits</th><th>Edge Detection Method for External Trigger</th></tr></table></td></tr>
<tr><td>CHAPTER36: UART (Asynchronous Serial Interface) 10.Registers of UART (Asynchronous Serial Interface) 10.6.Serial Auxiliary Control Status Register (SACSR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■This bit is reset when the SACSRC:TINTC bit in the clear register is set to "1".
<table><tr><th>TINT</th><th>Description</th></tr></table>
(Correct)
■This bit is reset when the SACSRC:TINTC bit in the clear register is set to "1".
<table><tr><th>Bit</th><th>Description</th></tr></table></td></tr>
<tr><td>CHAPTER36: UART (Asynchronous Serial Interface) 10.Registers of UART (Asynchronous Serial Interface) 10.6.Serial Auxiliary Control Status Register (SACSR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■This bit is set when the SACSRS:TINTES bit in the set register is set to "1".
<table><tr><th>TINTE</th><th>Description</th></tr></table>
(Correct)
■This bit is set when the SACSRS:TINTES bit in the set register is set to "1".
<table><tr><th>Bit</th><th>Description</th></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER36: UART (Asynchronous Serial Interface) 10.Registers of UART (Asynchronous Serial Interface) 10.6.Serial Auxiliary Control Status Register (SACSR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■This bit is set when the SACSRC:TSYNEC bit in the set register is set to "1".
<table><tr><th>TSYNE</th><th>Description</th></tr></table>
(Correct)
■This bit is set when the SACSRC:TSYNEC bit in the set register is set to "1".
<table><tr><th>Bit</th><th>Description</th></tr></table></td></tr>
<tr><td>CHAPTER36: UART (Asynchronous Serial Interface) 10.Registers of UART (Asynchronous Serial Interface) 10.6.Serial Auxiliary Control Status Register (SACSR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■This bit is set when the SACSRS:TRGES bit in the set register is set to "1".
<table><tr><th>TRGE</th><th>Description</th></tr></table>
(Correct)
■This bit is set when the SACSRS:TRGES bit in the set register is set to "1".
<table><tr><th>Bit</th><th>Description</th></tr></table></td></tr>
<tr><td>CHAPTER36: UART (Asynchronous Serial Interface) 10.Registers of UART (Asynchronous Serial Interface) 10.6.Serial Auxiliary Control Status Register (SACSR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit4:1] TDIV3-0 : Timer Operation Clock Division Bits

(Correct)
[bit4:1] TDIV3, TDIV2, TDIV1, TDIV0: Timer Operation Clock Division Bits</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER36: UART (Asynchronous Serial Interface) 10.Registers of UART (Asynchronous Serial Interface) 10.6.Serial Auxiliary Control Status Register (SACSR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■This bit is set when the SACSRC:TSYNEC bit in the set register is set to "1".
<table><tr><th>TMRE</th><th>Serial Timer Enable Bit</th></tr></table>
(Correct)
■This bit is set when the SACSRC:TSYNEC bit in the set register is set to "1".
<table><tr><th>Bit</th><th>Description</th></tr></table></td></tr>
<tr><td>CHAPTER36: UART (Asynchronous Serial Interface) 10.Registers of UART (Asynchronous Serial Interface) 10.9.Transfer Byte Register (TBYTE0)</td><td>The item should be added as indicated by the shading below.

(Error)
The transfer byte register 0 (TBYTE0) is used for synchronous transmission or external trigger transmission.

(Correct)
[bit7:0] TBYTE0: Transfer Byte Resister
The transfer byte register 0 (TBYTE0) is used for synchronous transmission or external trigger transmission.</td></tr>
<tr><td>CHAPTER36: UART (Asynchronous Serial Interface) 10.Registers of UART (Asynchronous Serial Interface) 10.10.Baud Rate Generator Register 1, 0 (BGR1, BGR0)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit15] EXT : External clock selection bit
<table><tr><th>EXT</th><th>External Clock Selection Bit</th></tr></table>
(Correct)
[bit15] EXT : External Clock Selection Bit
<table><tr><th>Bit</th><th>Description</th></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER36: UART (Asynchronous Serial Interface) 10.Registers of UART (Asynchronous Serial Interface) 10.10.Baud Rate Generator Register 1, 0 (BGR1, BGR0)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit14:8] BGR1 : Baud rate generator register 1
<table><tr><th>BGR1</th><th>Baud Rate Generator Register 1</th></tr></table>
(Correct)
[bit14:8] BGR1 : Baud Rate Generator Register 1
<table><tr><th>BGR1</th><th>Description</th></tr></table></td></tr>
<tr><td>CHAPTER36: UART (Asynchronous Serial Interface) 10.Registers of UART (Asynchronous Serial Interface) 10.10.Baud Rate Generator Register 1, 0 (BGR1, BGR0)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit7:0] BGR0 : Baud rate generator register 0
<table><tr><th>BGR0</th><th>Baud Rate Generator Register 0</th></tr></table>
Notes:
−Use 16-bit access for writing a value to the baud rate generator register (BGR1, BGR0).
−If the setting values of the baud rate generator registers (BGR1, BGR0) are changed, the new values are not reloaded until the counter value becomes "00000"h. Therefore, in order to enable the new value immediately, execute programmable clear (UPCL) after changing the values of BGR1/0.

(Correct)
[bit7:0] BGR0 : Baud Rate Generator Register 0
<table><tr><th>BGR0</th><th>Description</th></tr></table>
Notes:
−Use 16-bit access for writing a value to the baud rate generator register (BGR1, BGR0).
−If the setting values of the baud rate generator registers (BGR1, BGR0) are changed, the new values are not reloaded until the counter value becomes 0x00000. Therefore, in order to enable the new value immediately, execute programmable clear (UPCL) after changing the values of BGR1/0.</td></tr>
<tr><td>CHAPTER36: UART (Asynchronous Serial Interface) 10.Registers of UART (Asynchronous Serial Interface) 10.11.FIFO Control Register 1 (FCR1)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
"1": Enable FLST bit detection.
<table><tr><th>FLSTE</th><th>Retransmission Data Lost Detection Enable Bit</th></tr></table>
(Correct)
"1": Enable FLST bit detection.
<table><tr><th>Bit</th><th>Description</th></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER36: UART (Asynchronous Serial Interface) 10.Registers of UART (Asynchronous Serial Interface) 10.11.FIFO Control Register 1 (FCR1)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■When ECR:TXBLKEN = 1
−FTICR setting value ≥ Free space in the transmission FIFO
<table><tr><th>FDRQ</th><th>Transmission FIFO Data Request Bit</th></tr></table>
(Correct)
■When ECR:TXBLKEN = 1
−FTICR setting value ≥ Free space in the transmission FIFO
<table><tr><th>Bit</th><th>Description</th></tr></table></td></tr>
<tr><td>CHAPTER36: UART (Asynchronous Serial Interface) 10.Registers of UART (Asynchronous Serial Interface) 10.11.FIFO Control Register 1 (FCR1)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■This bit is set when the FCR1S:FTIES bit in the set register is set to "1".
<table><tr><th>FTIE</th><th>Transmission FIFO Interrupt Enable Bit</th></tr></table>
(Correct)
■This bit is set when the FCR1S:FTIES bit in the set register is set to "1".
<table><tr><th>Bit</th><th>Description</th></tr></table></td></tr>
<tr><td>CHAPTER36: UART (Asynchronous Serial Interface) 10.Registers of UART (Asynchronous Serial Interface) 10.11.FIFO Control Register 1 (FCR1)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
"1": Allocation is made as a transmission FIFO: FIFO2 and reception FIFO: FIFO1.
<table><tr><th>FSEL</th><th>FIFO Selection Bit</th></tr></table>
(Correct)
"1": Allocation is made as a transmission FIFO: FIFO2 and reception FIFO: FIFO1.
<table><tr><th>Bit</th><th>Description</th></tr></table></td></tr>
<tr><td>CHAPTER36: UART (Asynchronous Serial Interface) 10.Registers of UART (Asynchronous Serial Interface) 10.12.FIFO Control Register 0 (FCR0)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
When performing retransmission with this bit set to "1", resets the FIFO and then writes the data to the FIFO again.
<table><tr><th>FLST</th><th>FIFO Retransmission Data Lost Flag Bit</th></tr></table>
(Correct)
When performing retransmission with this bit set to "1", resets the FIFO and then writes the data to the FIFO again.
<table><tr><th>Bit</th><th>Description</th></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER36: UART (Asynchronous Serial Interface) 10.Registers of UART (Asynchronous Serial Interface) 10.12.FIFO Control Register 0 (FCR0)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■This bit is set when the FCR0S:FLDS bit in the set register is set to "1".
<table><tr><th>FLD</th><th>FIFO Pointer Reload Bit</th></tr></table>
(Correct)
■This bit is set when the FCR0S:FLDS bit in the set register is set to "1".
<table><tr><th>Bit</th><th>Description</th></tr></table></td></tr>
<tr><td>CHAPTER36: UART (Asynchronous Serial Interface) 10.Registers of UART (Asynchronous Serial Interface) 10.12.FIFO Control Register 0 (FCR0)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
"0": No effect.
<table><tr><th rowspan="2">FSET</th><th colspan="2">FIFO Pointer Saving Bit</th></tr><tr><th>Write</th><th>Read</th></tr></table>
(Correct)
"0": No effect
<table><tr><th rowspan="2">Bit</th><th colspan="2">Description</th></tr><tr><th>Write</th><th>Read</th></tr></table></td></tr>
<tr><td>CHAPTER36: UART (Asynchronous Serial Interface) 10.Registers of UART (Asynchronous Serial Interface) 10.12.FIFO Control Register 0 (FCR0)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■This bit is set when the FCR0S:FCL2S bit in the set register is set to "1".
<table><tr><th rowspan="2">FCL2</th><th colspan="2">FIFO2 Reset Bit</th></tr><tr><th>Write</th><th>Read</th></tr></table>
(Correct)
■This bit is set when the FCR0S:FCL2S bit in the set register is set to "1".
<table><tr><th rowspan="2">Bit</th><th colspan="2">Description</th></tr><tr><th>Write</th><th>Read</th></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER36: UART (Asynchronous Serial Interface) 10.Registers of UART (Asynchronous Serial Interface) 10.12.FIFO Control Register 0 (FCR0)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■This bit is set when the FCR0S:FCL1S bit in the set register is set to "1".
<table><tr><th rowspan="2">FCL1</th><th colspan="2">FIFO1 Reset Bit</th></tr><tr><th>Write</th><th>Read</th></tr></table>
(Correct)
■This bit is set when the FCR0S:FCL1S bit in the set register is set to "1".
<table><tr><th rowspan="2">Bit</th><th colspan="2">Description</th></tr><tr><th>Write</th><th>Read</th></tr></table></td></tr>
<tr><td>CHAPTER36: UART (Asynchronous Serial Interface) 10.Registers of UART (Asynchronous Serial Interface) 10.12.FIFO Control Register 0 (FCR0)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■To use FIFO2 as the reception FIFO, first disable reception (SCR:RXE = 0) and then change the setting of this bit when the reception buffer is empty (SSR:RDRF = "0") and the reception FIFO does not contain valid data (FBYTE2 = 0).
■To use FIFO2 as the reception FIFO, first disable reception (SCR:RXE = 0) and then set this bit to "1" when the reception buffer is empty (SSR:RDRF = "0").
■Disabling FIFO2 does not change the state of FIFO2.
■This bit is reset when the FCR0C:FE2C bit in the clear register is set to "1".
■This bit is set when the FCR0S:FE2S bit in the set register is set to "1".

<table><tr><th>FE2</th><th>FIFO2 Operation Enable Bit</th></tr></table>

(Correct)
■To use FIFO2 as the reception FIFO, first disable reception (SCR:RXE = 0) and then change the setting of this bit when the reception buffer is empty (SSR:RDRF = 0) and the reception FIFO does not contain valid data (FBYTE2 = 0).
■To use FIFO2 as the reception FIFO, first disable reception (SCR:RXE = 0) and then set this bit to "1" when the reception buffer is empty (SSR:RDRF = 0).
■Disabling FIFO2 does not change the state of FIFO2.
■This bit is reset when the FCR0C:FE2C bit in the clear register is set to "1".
■This bit is set when the FCR0S:FE2S bit in the set register is set to "1".

<table><tr><th>Bit</th><th>Description</th></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER36: UART (Asynchronous Serial Interface) 10.Registers of UART (Asynchronous Serial Interface) 10.12.FIFO Control Register 0 (FCR0)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■To use FIFO1 as the transmission or reception FIFO, set this bit to "1" or "0" when the transmission or reception buffer is empty ((SSR:TDRE = "1") or (SSR:RDRF = "0")), respectively.
■To use FIFO1 as the reception FIFO, first disable reception (SCR:RXE = 0) and then change the setting of this bit when the reception buffer is empty (SSR:RDRF = "0").
■Disabling FIFO1 does not change the state of FIFO1.
■This bit is reset when the FCR0C:FE1C bit in the clear register is set to "1".
■This bit is set when the FCR0S:FE1S bit in the set register is set to "1".

<table><tr><th>FE1</th><th>FIFO1 Operation Enable Bit</th></tr></table>

(Correct)
■To use FIFO1 as the transmission or reception FIFO, set this bit to "1" or "0" when the transmission or reception buffer is empty ((SSR:TDRE = 1) or (SSR:RDRF = 0)), respectively.
■To use FIFO1 as the reception FIFO, first disable reception (SCR:RXE = 0) and then change the setting of this bit when the reception buffer is empty (SSR:RDRF = 0).
■Disabling FIFO1 does not change the state of FIFO1.
■This bit is reset when the FCR0C:FE1C bit in the clear register is set to "1".
■This bit is set when the FCR0S:FE1S bit in the set register is set to "1".

<table><tr><th>Bit</th><th>Description</th></tr></table></td></tr>
<tr><td>CHAPTER36: UART (Asynchronous Serial Interface) 10.Registers of UART (Asynchronous Serial Interface) 10.13.FIFO Byte Register (FBYTE)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
FBYTE2, FBYTE1: FIFO2 data count indication bit, FIFO1 data count indication bit
<table><tr><th>FE1</th><th>FIFO1 Operation Enable Bit</th></tr></table>

(Correct)
[bit15:8] FBYTE2: FIFO2 Data Count Indication byte
[bit7:0] FBYTE1: FIFO1 Data Count Indication Byte
<table><tr><th>FBYTE2, FBYTE1</th><th>Description</th></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER36: UART (Asynchronous Serial Interface) 10.Registers of UART (Asynchronous Serial Interface) 10.14.Transmission FIFO Interrupt Control Register (FTICR)</td><td>eatures of "For details" should be corrected as indicated by the shading below.

(Error)
FTICR2, FTICR1: FIFO2 data count indication bit, FIFO1 data count indication bit
<table><tr><th>FE1</th><th>FIFO1 Operation Enable Bit</th></tr></table>
(Correct)
[bit15:8] FTICR2: FIFO2 Data Count Indication byte
[bit7:0] FTICR1:FIFO1 Data Count Indication Byte
<table><tr><th>FTICR2, FTICR1</th><th>Description</th></tr></table></td></tr>
<tr><td>CHAPTER36: UART (Asynchronous Serial Interface) 10.Registers of UART (Asynchronous Serial Interface) 10.15.Extended Control Register (ECR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
This bit selects between the output of an error interrupt from an interrupt request pin and output from an error interrupt request pin. For details on interrupt output selection, see Table 2-2 and Table 2-3.
<table><tr><th>EISEL</th><th>Error Interrupt Request Output Selection Bit</th></tr></table>
This bit selects between the output of an error interrupt from an interrupt request pin and output from an error interrupt request pin. For details on interrupt output selection, see Table 2- 2 and Table 2-3.
<table><tr><th>Bit</th><th>Description</th></tr></table></td></tr>
<tr><td>CHAPTER36: UART (Asynchronous Serial Interface) 10.Registers of UART (Asynchronous Serial Interface) 10.15.Extended Control Register (ECR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
This bit enables/disables reception error interrupt request output. For details on the target interrupt factors, see Table 2-3.
<table><tr><th>REIE</th><th>Reception Error Interrupt Enable Bit</th></tr></table>
(Correct)
This bit enables/disables reception error interrupt request output. For details on the target interrupt factors, see Table 2-3.
<table><tr><th>Bit</th><th>Description</th></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER36: UART (Asynchronous Serial Interface) 10.Registers of UART (Asynchronous Serial Interface) 10.15.Extended Control Register (ECR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
This bit enables/disables transmission error interrupt request output. For details on the target interrupt factors, see Table 2-2.
<table><tr><td>TEIE</td><td>Transmission Error Interrupt Enable Bit</td></tr></table>
(Correct)
This bit enables/disables transmission error interrupt request output. For details on the target interrupt factors, see Table 2-2.
<table><tr><td>Bit</td><td>Description</td></tr></table></td></tr>
<tr><td>CHAPTER36: UART (Asynchronous Serial Interface) 10.Registers of UART (Asynchronous Serial Interface) 10.15.Extended Control Register (ECR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■"1": Perform DMA transfer in block transfer mode.
<table><tr><td>RXBLKEN</td><td>Reception Block Transfer Setting Bit</td></tr></table>
(Correct)
■"1": Perform DMA transfer in block transfer mode.
<table><tr><td>Bit</td><td>Description</td></tr></table></td></tr>
<tr><td>CHAPTER36: UART (Asynchronous Serial Interface) 10.Registers of UART (Asynchronous Serial Interface) 10.15.Extended Control Register (ECR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■"1": Perform DMA transfer in block transfer mode..
<table><tr><td>TXBLKEN</td><td>Transmission Block Transfer Setting Bit</td></tr></table>
(Correct)
■"1": Perform DMA transfer in block transfer mode..
<table><tr><td>Bit</td><td>Description</td></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER36: UART (Asynchronous Serial Interface) 10.Registers of UART (Asynchronous Serial Interface) 10.16.Extended Status Register (ESR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■For details on the output of interrupt requests, see Table 2-3.
<table>
<tr><th>RXUDR</th><th>Reception FIFO Under run Flag Bit</th></tr>
<tr><td>0</td><td>No reception FIFO under run has occurred.</td></tr>
<tr><td>1</td><td>A reception FIFO under run has occurred.</td></tr>
</table>
Notes:
−Software reset (SCR:UPCL = "1") resets this bit to "0".

(Correct)
■For details on the output of interrupt requests, see Table 2-3.
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>No reception FIFO under run has occurred.</td></tr>
<tr><td>1</td><td>A reception FIFO under run has occurred.</td></tr>
</table>
Notes:
−Software reset (SCR:UPCL = 1) resets this bit to "0".</td></tr>
<tr><td>CHAPTER36: UART (Asynchronous Serial Interface) 10.Registers of UART (Asynchronous Serial Interface) 10.16.Extended Status Register (ESR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■For details on the output of interrupt requests, see Table 2-2.
<table>
<tr><th>TXOVR</th><th>Transmission FIFO Overrun Flag Bit</th></tr>
<tr><td>0</td><td>No transmission FIFO overrun has occurred.</td></tr>
<tr><td>1</td><td>A transmission FIFO overrun has occurred.</td></tr>
</table>
Notes:
−Software reset (SCR:UPCL = "1") resets this bit to "0".

(Correct)
■For details on the output of interrupt requests, see Table 2-2.
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>No transmission FIFO overrun has occurred.</td></tr>
<tr><td>1</td><td>A transmission FIFO overrun has occurred.</td></tr>
</table>
Notes:
−Software reset (SCR:UPCL = 1) resets this bit to "0".</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER36: UART (Asynchronous Serial Interface) 10.Registers of UART (Asynchronous Serial Interface) 10.16.Extended Status Register (ESR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■For details on the output of interrupt requests, see Table 2-3.
<table>
<tr><th>RBERR</th><th>Reception Block Transfer Error Bit</th></tr>
<tr><td>0</td><td>No reception block transfer error has occurred.</td></tr>
<tr><td>1</td><td>A reception block transfer error has occurred.</td></tr>
</table>
Notes:
−Software reset (SCR:UPCL = "1") resets this bit to "0".

(Correct)
■For details on the output of interrupt requests, see Table 2-3.
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>No reception block transfer error has occurred.</td></tr>
<tr><td>1</td><td>A reception block transfer error has occurred.</td></tr>
</table>
Notes:
−Software reset (SCR:UPCL = 1) resets this bit to "0".</td></tr>
<tr><td>CHAPTER36: UART (Asynchronous Serial Interface) 10.Registers of UART (Asynchronous Serial Interface) 10.16.Extended Status Register (ESR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■For details on the output of interrupt requests, see Table 2-2.
<table>
<tr><th>TBERR</th><th>Transmission Block Transfer Error Bit</th></tr>
<tr><td>0</td><td>No transmission block transfer error has occurred.</td></tr>
<tr><td>1</td><td>A transmission block transfer error has occurred.</td></tr>
</table>
Notes:
−Software reset (SCR:UPCL = "1") resets this bit to "0".

(Correct)
■For details on the output of interrupt requests, see Table 2-2.
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>No transmission block transfer error has occurred.</td></tr>
<tr><td>1</td><td>A transmission block transfer error has occurred.</td></tr>
</table>
Notes:
−Software reset (SCR:UPCL = 1) resets this bit to "0".</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER36: UART (Asynchronous Serial Interface) 10.Registers of UART (Asynchronous Serial Interface) 10.17.Transmission Block Size Register (TBSIZE)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
TBSIZE Bit
<table><tr><td>Write</td><td>Set the number of transmission blocks.</td></tr><tr><td>Read</td><td>Read the setting.</td></tr></table>
(Correct)
[bit7:0] TBSIZE: Transmission Block Size Byte
<table><tr><th>TBSIZE</th><th>Description</th></tr><tr><td>Write</td><td>Set the number of transmission blocks.</td></tr><tr><td>Read</td><td>Read the setting.</td></tr></table></td></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 1.Overview of the CSIO (Clock Synchronous Serial Interface)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
<table><tr><td></td><td></td><th>Function</th></tr><tr><td>1</td><td>Data buffer</td><td>− Full-duplex double buffer (when FIFO is not used)
− Transmission and reception FIFOs (64 bytes each)*1 (when FIFO is used)</td></tr></table>
(Correct)
<table><tr><td></td><th>Function</th><th>Description</th></tr><tr><td>1</td><td>Data buffer</td><td>− Full-duplex double buffer (when FIFO is not used)
− Transmission and reception FIFOs (64 bytes each)*1 (when FIFO is used)</td></tr></table></td></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 3.Operation of the CSIO (Clock Synchronous Serial Interface) 3.3.SPI Transfer (I)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
Master operation (SCR:MS = 0, SMR:SCKE = 1, SCSCR:CSEN3 to CSEN0 = "0000"b)

(Correct)
Master Operation (SCR:MS = 0, SMR:SCKE = 1, SCSCR:CSEN3 to CSEN0 = 0x0000)</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 7.Dedicated Baud Rate Generator 7.2.Setup Procedure and Program Flow for the CSIO (Clock Synchronous Serial Interface)</td><td>The function should be deleted as indicated by the shading below.

(Error)
Flow chart
■FIFO not used

(Correct)
■FIFO not used</td></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.1.Serial Control Register (SCR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
At reading, "0" is always read.
<table><tr><th rowspan="2">UPCL</th><th colspan="2">Programmable Clear Bit</th></tr><tr><th>Write</th><th>Read</th></tr><tr><td>0</td><td>No effect</td><td rowspan="2">"0" is always read.</td></tr><tr><td>1</td><td>Programmable clear</td></tr></table>
(Correct)
At reading, "0" is always read.
<table><tr><th rowspan="2">Bit</th><th colspan="2">Description</th></tr><tr><th>Write</th><th>Read</th></tr><tr><td>0</td><td>No effect</td><td rowspan="2">"0" is always read.</td></tr><tr><td>1</td><td>Programmable clear</td></tr></table></td></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.1.Serial Control Register (SCR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■1: Set slave mode.
<table><tr><th>MS</th><th>Master/Slave Function Select Bit</th></tr></table>
(Correct)
■1: Set slave mode.
<table><tr><th>Bit</th><th>Description</th></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.1.Serial Control Register (SCR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■1: Support SPI.<table><tr><th>SPI</th><th>SPI-supporting Bit</th></tr><tr><td>0</td><td>Normal synchronous transfer</td></tr><tr><td>1</td><td>SPI support</td></tr></table>Notes:
−Set this bit when transmission and reception are disabled (TXE = RXE = 0).
−This bit is used in any of the following cases:
−The chip select pins are disabled (SCSCR:CSEN3 to CSEN0 = "0000"b).

(Correct)
■1: Support SPI.<table><tr><th>Bit</th><th>Description</th></tr><tr><td>0</td><td>Normal synchronous transfer</td></tr><tr><td>1</td><td>SPI support</td></tr></table>Notes:
−Set this bit when transmission and reception are disabled (TXE = RXE = 0).
−This bit is used in any of the following cases:
−The chip select pins are disabled (SCSCR:CSEN3 to CSEN0 = 0b0000).</td></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.1.Serial Control Register (SCR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■For ECR:EISEL = 1, a reception interrupt request is issued when the SMR:RIE bit and the reception data flag bit (SSR:RDRF) are both "1".<table><tr><th>RIE</th><th>Reception Interrupt Enable Bit</th></tr></table>(Correct)
■For ECR:EISEL = 1, a reception interrupt request is issued when the SMR:RIE bit and the reception data flag bit (SSR:RDRF) are both "1".<table><tr><th>Bit</th><th>Description</th></tr></table></td></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.1.Serial Control Register (SCR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■For ECR:EISEL = 1, a reception interrupt request is issued when the TIE bit and the SSR:TDRE bit are both "1" or when one of the error flag bits (ESR:TXOVR, TBERR) is "1".<table><tr><th>TIE</th><th>Transmission Interrupt Enable Bit</th></tr></table>(Correct)
■For ECR:EISEL = 1, a reception interrupt request is issued when the TIE bit and the SSR:TDRE bit are both "1" or when one of the error flag bits (ESR:TXOVR, TBERR) is "1".<table><tr><th>Bit</th><th>Description</th></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.1.Serial Control Register (SCR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■A transmission bus idle interrupt request is output when the TBIE bit and the SSR:TBI bit are both "1".
<table><tr><td>TBIE</td><td>Transmission Bus Idle Interrupt Enable Bit</td></tr></table>
(Correct)
■A transmission bus idle interrupt request is output when the TBIE bit and the SSR:TBI bit are both "1".
<table><tr><td>Bit</td><td>Description</td></tr></table></td></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.1.Serial Control Register (SCR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■1: Enable data frame reception.
<table><tr><td>RXE</td><td>Reception Enable Bit</td></tr></table>
(Correct)
■1: Enable data frame reception.
<table><tr><td>Bit</td><td>Description</td></tr></table></td></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.1.Serial Control Register (SCR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■1: Enable data frame transmission.
<table><tr><td>TXE</td><td>Transmission Enable Bit</td></tr></table>
(Correct)
■1: Enable data frame transmission.
<table><tr><td>Bit</td><td>Description</td></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.2.Serial Mode Register (SMR)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit7:5] MD2, MD1, MD0: Operation mode setting bits
These bits set the operation modes.
"0b000": Set operation mode 0 (asynchronous normal mode).
"0b001": Set operation mode 1 (asynchronous multi-processor mode).
"0b010": Set operation mode 2 (clock synchronous mode).
"0b011": Set operation mode 3 (LIN communication mode).
"0b100": Set operation mode 4 (I2C mode).
This section describes the registers and their operations in operation mode 2 (clock synchronous mode).
<table>
<tr><th>MD2</th><th>MD1</th><th>MD0</th><th>Operation Mode Setting Bit</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>Operation mode 0 (asynchronous normal mode)</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>Operation mode 1 (asynchronous multi-processor mode)</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>Operation mode 2 (clock synchronous mode)</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>Operation mode 3 (LIN communication mode)</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>Operation mode 4 (I2C mode)</td></tr>
</table>

(Correct)
[bit7:5] MD[2:0]: Operation Mode Setting Bits
These bits set the operation modes.
0b000: Set operation mode 0 (asynchronous normal mode).
0b001: Set operation mode 1 (asynchronous multi-processor mode).
0b010: Set operation mode 2 (clock synchronous mode).
0b011: Set operation mode 3 (LIN communication mode).
0b100: Set operation mode 4 (I2C mode).
This section describes the registers and their operations in operation mode 2 (clock synchronous mode).
<table>
<tr><th colspan="3">Bits</th><th>Description</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>Operation mode 0 (asynchronous normal mode)</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>Operation mode 1 (asynchronous multi-processor mode)</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>Operation mode 2 (clock synchronous mode)</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>Operation mode 3 (LIN communication mode)</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>Operation mode 4 (I2C mode)</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.2.Serial Mode Register (SMR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
1: The pin used for external interrupts is the SCK pin.
<table><tr><td>WUCR</td><td>WAKE UP Control Bit</td></tr></table>
(Correct)
1: The pin used for external interrupts is the SCK pin.
<table><tr><td>Bit</td><td>Description</td></tr></table></td></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.2.Serial Mode Register (SMR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■During normal transfer, reception data is sampled at a falling edge of the serial clock. In SPI transfer, it is sampled at a rising edge of the serial clock.
<table><tr><td>SCINV</td><td>Serial Clock Invert Bit</td></tr></table>
(Correct)
■During normal transfer, reception data is sampled at a falling edge of the serial clock. In SPI transfer, it is sampled at a rising edge of the serial clock.
<table><tr><td>Bit</td><td>Description</td></tr></table></td></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.2.Serial Mode Register (SMR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
This bit selects whether the lowest bit is transferred first (LSB first, BDS = 0) or the highest bit is transferred first (MSB first, BDS = 1), when transferring serial data. When chip select is used in master mode (SCR:MS = 0), the bit is used for communication with serial chip select pin 0.
<table><tr><td>BDS</td><td>Transfer Direction Selection Bit</td></tr></table>
(Correct)
This bit selects whether the lowest bit is transferred first (LSB first, BDS = 0) or the highest bit is transferred first (MSB first, BDS = 1), when transferring serial data. When chip select is used in master mode (SCR:MS = 0), the bit is used for communication with serial chip select pin 0.
<table><tr><td>Bit</td><td>Description</td></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.2.Serial Mode Register (SMR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
This bit enables the input or output of the serial clock.
<table><tr><th>SCKE</th><th>Serial Clock Input/output Enable Bit</th></tr></table>
(Correct)
This bit enables the input or output of the serial clock.
<table><tr><th>Bit</th><th>Description</th></tr></table></td></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.2.Serial Mode Register (SMR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
This bit enables/stops the output of serial data.
<table><tr><th>SOE</th><th>Serial Data Output Enable Bit</th></tr></table>
(Correct)
This bit enables/stops the output of serial data.
<table><tr><th>Bit</th><th>Description</th></tr></table></td></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.3.Serial Status Register (SSR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
Upon reading, "0" is always read.
<table><tr><th rowspan="2">REC</th><th colspan="2">Reception Error Flag Clear Bit</th></tr><tr><th>Write</th><th>Read</th></tr><tr><td>0</td><td>No effect</td><td rowspan="2">"0" is always read.</td></tr><tr><td>1</td><td>Clear the reception error flag (ORE).</td></tr></table>
(Correct)
Upon reading, "0" is always read.
<table><tr><th rowspan="2">Bit</th><th colspan="2">Description</th></tr><tr><th>Write</th><th>Read</th></tr><tr><td>0</td><td>No effect</td><td rowspan="2">"0" is always read.</td></tr><tr><td>1</td><td>Clear the reception error flag (ORE).</td></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.3.Serial Status Register (SSR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
This bit selects between 16- and 32-bit access when accessing the transmission data register (TDR) and the reception data register (RDR).
<table><tr><td>AWC</td><td>Access Width Control Bit</td></tr></table>
(Correct)
This bit selects between 16- and 32-bit access when accessing the transmission data register (TDR) and the reception data register (RDR).
<table><tr><td>Bit</td><td>Description</td></tr></table></td></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.3.Serial Status Register (SSR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■If this flag is set when the reception FIFO is used, the reception FIFO enable bits are cleared, and no reception data is stored in the reception FIFO.
<table><tr><td>ORE</td><td>Overrun Error Flag Bit</td></tr></table>
(Correct)
■If this flag is set when the reception FIFO is used, the reception FIFO enable bits are cleared, and no reception data is stored in the reception FIFO.
<table><tr><td>Bit</td><td>Description</td></tr></table></td></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.3.Serial Status Register (SSR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■For ECR:RXBLKEN = 1, when the reception FIFO is used, this bit is cleared to "0" when the number of data items in the reception FIFO becomes equal to or less than the value set for the FBYTE register.
<table><tr><td>RDRF</td><td>Reception Data Full Flag Bit</td></tr></table>
(Correct)
■For ECR:RXBLKEN = 1, when the reception FIFO is used, this bit is cleared to "0" when the number of data items in the reception FIFO becomes equal to or less than the value set for the FBYTE register.
<table><tr><td>Bit</td><td>Description</td></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.3.Serial Status Register (SSR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■For the set/reset timing of the TDRE bit when using the transmission FIFO,
see "2.4 Interrupt Generation and Flag Set Timing When the Transmission FIFO Is Used."
<table><tr><th>TDRE</th><th>Transmission Data Empty Flag Bit</th></tr></table>
(Correct)
■For the set/reset timing of the TDRE bit when using the transmission FIFO,
see "2.4 Interrupt Generation and Flag Set Timing When the Transmission FIFO Is Used."
<table><tr><th>Bit</th><th>Description</th></tr></table></td></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.3.Serial Status Register (SSR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■When this bit is "1", if transmission bus idle interrupts are enabled (SCR:TBIE = 1), a transmission interrupt request is output.
<table><tr><th>TBI</th><th>Transmission Bus Idle Flag Bit</th></tr></table>
(Correct)
■When this bit is "1", if transmission bus idle interrupts are enabled (SCR:TBIE = 1), a transmission interrupt request is output.
<table><tr><th>Bit</th><th>Description</th></tr></table></td></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.4.Extended Communication Control Register (ESCR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
Upon reading, "0" is always read.
<table><tr><th rowspan="2">SOP</th><th>Serial Output Pin Set Bit</th><th></th></tr><tr><th>Write</th><th>Read</th></tr><tr><td>0</td><td>No effect</td><td rowspan="2">"0" is always read.</td></tr><tr><td>1</td><td>Set the SOUT pin to "H".</td></tr></table>
(Correct)
Upon reading, "0" is always read.
<table><tr><th rowspan="2">Bit</th><th colspan="2">Description</th></tr><tr><th>Write</th><th>Read</th></tr><tr><td>0</td><td>No effect</td><td rowspan="2">"0" is always read.</td></tr><tr><td>1</td><td>Set the SOUT pin to "H".</td></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.4.Extended Communication Control Register (ESCR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■Data length of serial data
<table><tr><th>CSFE</th><th>Chip Select Format Enable</th></tr></table>
(Correct)
■Data length of serial data
<table><tr><th>Bit</th><th>Description</th></tr></table></td></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.4.Extended Communication Control Register (ESCR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit4:3] WT1, WT0: Data transmission/reception wait select bits
These bits specify the number of waits for the transmission or reception of continuous data in master mode.
In slave mode, the operation is that for "00".
■"00": Output SCK continuously.
■"01": Output SCK after a 1-bit time wait.
■"10": Output SCK after a 2-bit time wait.
■"11": Output SCK after a 3-bit time wait.
<table><tr><th>WT1</th><th>WT0</th><th>Data Transmission/Reception Wait Select Bit</th></tr></table>
(Correct)
[bit4:3] WT[1:0]: Data Transmission/Reception Wait Select Bits
These bits specify the number of waits for the transmission or reception of continuous data in master mode.
In slave mode, the operation is that for "00".
■"00": Output SCK continuously.
■"01": Output SCK after a 1-bit time wait.
■"10": Output SCK after a 2-bit time wait.
■"11": Output SCK after a 3-bit time wait.
<table><tr><th>Bits</th><th>Description</th></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.4.Extended Communication Control Register (ESCR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
These bits specify the data length of the transmission/reception data. When chip select is used in master mode (SCR:MS = 0), this bit is used for communication with serial chip select pin 0.
<table><tr><th>L3</th><th>L2</th><th>L1</th><th>L0</th><th>Data Length Selection Bit</th></tr></table>
(Correct)
These bits specify the data length of the transmission/reception data. When chip select is used in master mode (SCR:MS = 0), this bit is used for communication with serial chip select pin 0.
<table><tr><th>L3</th><th>L2</th><th>L1</th><th>L0</th><th>Data Length</th></tr></table></td></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.5.Reception Data Register/Transmission Data Register (RDR/TDR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■Serial data signals transmitted to the serial input pin (SIN pin) is converted by the shift register, and the result is stored in the reception data register (RDR).
According to the data length, the reception data is stored, starting with the lower bits, while the other bits are set to "0". Example: D7 to D0 = "45"h, D31 to D8 = 0 if the data length is 8 bits, and "45"h is received.

(Correct)
■Serial data signals transmitted to the serial input pin (SIN pin) is converted by the shift register, and the result is stored in the reception data register (RDR).
According to the data length, the reception data is stored, starting with the lower bits, while the other bits are set to "0". Example: D7 to D0 = 0x45, D31 to D8 = 0x000000 if the data length is 8 bits, and 0x45 is received.</td></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.6.Serial Auxiliary Control Status Register (SACSR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■This bit is set when the SACSRS:STSTS bit in the set register is set to "1".
<table><tr><th>STST</th><th>Serial Test Bit</th></tr></table>
(Correct)
■This bit is set when the SACSRS:STSTS bit in the set register is set to "1".
<table><tr><th>Bit</th><th>Description</th></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.6.Serial Auxiliary Control Status Register (SACSR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■This bit is set when the SACSRS:TBEENS bit in the set register is set to "1".
<table><tr><th>TBEEN</th><th>Transfer Byte Error Enable Bit</th></tr></table>
(Correct)
■This bit is set when the SACSRS:TBEENS bit in the set register is set to "1".
<table><tr><th>Bit</th><th>Description</th></tr></table></td></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.6.Serial Auxiliary Control Status Register (SACSR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■This bit is set when the SACSRS:CSEIES bit in the set register is set to "1".
<table><tr><th>CSEIE</th><th>Chip Select Error Interrupt Enable Bit</th></tr></table>
(Correct)
■This bit is set when the SACSRS:CSEIES bit in the set register is set to "1".
<table><tr><th>Bit</th><th>Description</th></tr></table></td></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.6.Serial Auxiliary Control Status Register (SACSR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■This bit is reset when the SACSRC:CSEC bit in the clear register is set to "1".
<table><tr><th>CSE</th><th>Chip Select Error Flag Bit</th></tr></table>
(Correct)
■This bit is reset when the SACSRC:CSEC bit in the clear register is set to "1".
<table><tr><th>Bit</th><th>Description</th></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.6.Serial Auxiliary Control Status Register (SACSR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit10:9] TRG1, TRG0 : Trigger Selection Bits
These bits select the method for detecting the external trigger edges for activating the serial timer.
<table>
<tr><th>TRG1</th><th>TRG0</th><th>Edge Detection Method for External Trigger</th></tr>
</table>

(Correct)
[bit10:9] TRG[1:0] : Trigger selection bits
These bits select the method for detecting the external trigger edges for activating the serial timer.
<table>
<tr><th>Bit</th><th>Edge Detection Method for External Trigger</th></tr>
</table>
</td></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.6.Serial Auxiliary Control Status Register (SACSR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■This bit is reset when the SACSRC:TINTC bit in the clear register is set to "1".
<table>
<tr><th>TINT</th><th>Description</th></tr>
<tr><td>0</td><td>No timer interrupt request</td></tr>
<tr><td>1</td><td>Timer interrupt request issued</td></tr>
</table>

Notes:
−Software reset (SCR:UPCL = "1") resets this bit to "0".

(Correct)
■This bit is reset when the SACSRC:TINTC bit in the clear register is set to "1".
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>No timer interrupt request</td></tr>
<tr><td>1</td><td>Timer interrupt request issued</td></tr>
</table>

Notes:
−Software reset (SCR:UPCL = 1) resets this bit to "0".</td></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.6.Serial Auxiliary Control Status Register (SACSR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■This bit is set when the SACSRS:TINTES bit in the set register is set to "1".
<table>
<tr><th>TINTE</th><th>Description</th></tr>
</table>

(Correct)
■This bit is set when the SACSRS:TINTES bit in the set register is set to "1".
<table>
<tr><th>Bit</th><th>Description</th></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.6.Serial Auxiliary Control Status Register (SACSR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■This bit is set when the SACSRS:TSYNES bit in the set register is set to "1".
<table><tr><th>TSYNE</th><th>Description</th></tr></table>
(Correct)
■This bit is set when the SACSRS:TSYNES bit in the set register is set to "1".
<table><tr><th>Bit</th><th>Description</th></tr></table></td></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.6.Serial Auxiliary Control Status Register (SACSR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■This bit is set when the SACSRS:TRGES bit in the set register is set to "1".
<table><tr><th>TRGE</th><th>Description</th></tr></table>
(Correct)
■This bit is set when the SACSRS:TRGES bit in the set register is set to "1".
<table><tr><th>Bit</th><th>Description</th></tr></table></td></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.6.Serial Auxiliary Control Status Register (SACSR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit4:1] TDIV3-0 : Timer Operation Clock Division Bits

(Correct)
[bit4:1] TDIV[3:0]: Timer operation clock division bits</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.6.Serial Auxiliary Control Status Register (SACSR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■This bit is set when the SACSRS:TMRES bit in the set register is set to "1".
<table><tr><th>TMRE</th><th>Serial Timer Enable Bit</th></tr></table>
(Correct)
■This bit is set when the SACSRS:TMRES bit in the set register is set to "1".
<table><tr><th>Bit</th><th>Description</th></tr></table></td></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.7.Serial Timer Register (STMR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit15-0]TM15-0 : Timer Data Bits

(Correct)
[bit15-0] TM[15:0]: Timer Data Bits</td></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.8.Serial Timer Comparison Register (STMCR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit15:0] TC15-0 : Compare Bits
These bits set a comparison value for the serial timer.
These bits are compared with the serial timer register (STMR). If the serial timer register (STMR) coincides with these bits when the serial timer register is updated, the serial timer register is set to "0". At this time, if synchronous transmission is disabled (SACSR:TSYNE = "0"), the timer interrupt flag (SACSR:TINT) is set to "1". If synchronous transmission is enabled (SACSR:TSYNE = "1"), the transmission is activated.

(Correct)
[bit15:0] TC[15:0]: Compare bits
These bits set a comparison value for the serial timer.
These bits are compared with the serial timer register (STMR). If the serial timer register (STMR) coincides with these bits when the serial timer register is updated, the serial timer register is set to "0". At this time, if synchronous transmission is disabled (SACSR:TSYNE = 0), the timer interrupt flag (SACSR:TINT) is set to "1". If synchronous transmission is enabled (SACSR:TSYNE = 1), the transmission is activated.</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface)
9.Registers of the CSIO (Clock Synchronous Serial Interface)
9.9.Serial Chip Select Control Status Register (SCSCR)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit15:14] SST1-0 : Serial Chip Select Start Bits
These bits select the pin at which serial chip select is to start.
If transmission is changed from the disabled state (SCR:TXE = "0") to the enabled state (SCR:TXE = "1"), and transmission data is written to TDR, the serial chip select pins become active sequentially, starting with the one set with these bits.
<table>
<tr><th>SST1</th><th>SST0</th><th>Start Pin</th></tr>
<tr><td>0</td><td>0</td><td>SCS0</td></tr>
<tr><td>0</td><td>1</td><td>SCS1</td></tr>
<tr><td>1</td><td>0</td><td>SCS2</td></tr>
<tr><td>1</td><td>1</td><td>SCS3</td></tr>
</table>
Notes:
−These bits can be changed only when transmission and reception are disabled (SCR:TXE = RXE = "0").
−If the serial chip select start bits (SST1, SST0) and the serial chip select end bits (SED1, SED0) are set to the same values, only the set serial chip select pin becomes active.
−In slave mode (SCR:MS = 1), the settings of these bits are invalid.
−Only those serial chip select pins for which serial chip select is enabled (CSEN = 1) become active.
−When using serial chip select in master mode (SCR:MS = "0"), enable serial chip select (CSEN = 1) for the serial chip select pin that is set with these bits.

(Correct)
[bit15:14] SST[1:0]: Serial chip select start bits
These bits select the pin at which serial chip select is to start.
If transmission is changed from the disabled state (SCR:TXE = 0) to the enabled state (SCR:TXE = 1), and transmission data is written to TDR, the serial chip select pins become active sequentially, starting with the one set with these bits.
<table>
<tr><th colspan="2">Bits</th><th>Start Pin</th></tr>
<tr><td>0</td><td>0</td><td>SCS0</td></tr>
<tr><td>0</td><td>1</td><td>SCS1</td></tr>
<tr><td>1</td><td>0</td><td>SCS2</td></tr>
<tr><td>1</td><td>1</td><td>SCS3</td></tr>
</table>
Notes:
−These bits can be changed only when transmission and reception are disabled (SCR:TXE = RXE = 0).
−If the serial chip select start bits (SST1, SST0) and the serial chip select end bits (SED1, SED0) are set to the same values, only the set serial chip select pin becomes active.
−In slave mode (SCR:MS = 1), the settings of these bits are invalid.
−Only those serial chip select pins for which serial chip select is enabled (CSEN = 1) become active.
−When using serial chip select in master mode (SCR:MS = 0), enable serial chip select (CSEN = 1) for the serial chip select pin that is set with these bits.
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface)
9.Registers of the CSIO (Clock Synchronous Serial Interface)
9.9.Serial Chip Select Control Status Register (SCSCR)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit13:12] SED1-0 : Serial Chip Select End Bits
These bits select the pin at which serial chip select is to end.
If the serial chip select pins become active, up to the serial chip select pin that is set with these bits, the next serial chip select pin to become active is that specified with the serial chip select start bits (SST1, SST0).
<table>
<tr><th>SED1</th><th>SED0</th><th>End Pin</th></tr>
<tr><td>0</td><td>0</td><td>SCS0</td></tr>
<tr><td>0</td><td>1</td><td>SCS1</td></tr>
<tr><td>1</td><td>0</td><td>SCS2</td></tr>
<tr><td>1</td><td>1</td><td>SCS3</td></tr>
</table>
Notes:
−These bits can be changed only when transmission and reception are disabled (SCR:TXE = RXE = "0").
−If the serial chip select start bits (SST1, SST0) and the serial chip select end bits (SED1, SED0) are set to the same values, only the set serial chip select pin becomes active.
−Only those serial chip select pins for which serial chip select is enabled (CSEN = 1) become active.
−In slave mode (SCR:MS = 1), the settings of these bits are invalid.
−When serial chip select is used in master mode (SCR:MS = "0"), enable serial chip select (CSEN = 1) for the serial chip select pin that is set with these bits.

(Correct)
[bit13:12] SED[1:0]: Serial chip select end bits
These bits select the pin at which serial chip select is to end.
If the serial chip select pins become active, up to the serial chip select pin that is set with these bits, the next serial chip select pin to become active is that specified with the serial chip select start bits (SST1, SST0).
<table>
<tr><th colspan="2">Bits</th><th>End Pin</th></tr>
<tr><td>0</td><td>0</td><td>SCS0</td></tr>
<tr><td>0</td><td>1</td><td>SCS1</td></tr>
<tr><td>1</td><td>0</td><td>SCS2</td></tr>
<tr><td>1</td><td>1</td><td>SCS3</td></tr>
</table>
Notes:
−These bits can be changed only when transmission and reception are disabled (SCR:TXE = RXE = 0).
−If the serial chip select start bits (SST1, SST0) and the serial chip select end bits (SED1, SED0) are set to the same values, only the set serial chip select pin becomes active.
−Only those serial chip select pins for which serial chip select is enabled (CSEN = 1) become active.
−In slave mode (SCR:MS = 1), the settings of these bits are invalid.
−When serial chip select is used in master mode (SCR:MS = 0), enable serial chip select (CSEN = 1) for the serial chip select pin that is set with these bits.
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.9.Serial Chip Select Control Status Register (SCSCR)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit11:10] SCD1-0 : Serial Chip Select Display Bits
These bits display the active serial chip select pin.
<table>
<tr><th>SCD1</th><th>SCD0</th><th>Display Pin</th></tr>
<tr><td>0</td><td>0</td><td>SCS0</td></tr>
<tr><td>0</td><td>1</td><td>SCS1</td></tr>
<tr><td>1</td><td>0</td><td>SCS2</td></tr>
<tr><td>1</td><td>1</td><td>SCS3</td></tr>
</table>
Notes:
−If the serial chip select pins are inactive, these bits display the next serial chip select pin to become active.
−These bits are set to "00"b in slave mode (SCR:MS = "1"), at a software reset (SCR:UPCL = 1), or when transmission is disabled (SCR:TXE = "0").

(Correct)
[bit11:10] SCD[1:0]: Serial chip select display bits
These bits display the active serial chip select pin..
<table>
<tr><th colspan="2">Bits</th><th>Display Pin</th></tr>
<tr><td>0</td><td>0</td><td>SCS0</td></tr>
<tr><td>0</td><td>1</td><td>SCS1</td></tr>
<tr><td>1</td><td>0</td><td>SCS2</td></tr>
<tr><td>1</td><td>1</td><td>SCS3</td></tr>
</table>
Notes:
−If the serial chip select pins are inactive, these bits display the next serial chip select pin to become active.
−These bits are set to 0b00 in slave mode (SCR:MS = 1), at a software reset (SCR:UPCL = 1), or when transmission is disabled (SCR:TXE = 0).
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface)
9.Registers of the CSIO (Clock Synchronous Serial Interface)
9.9.Serial Chip Select Control Status Register (SCSCR)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit9] SCAM : Serial Chip Select Active Retention Bit
This bit selects whether to retain the active state of the serial chip select pin.
For details, see "5 Operation of Serial Chip Select," "Serial chip select active level retention operation (SCSCR:SCAM = 1) (valid only in master mode (SCR:MS = 0))."
<table>
<tr><th>SCAM</th><th>Serial Chip Select Active Retention Bit</th></tr>
<tr><td>0</td><td>Do not retain the active state of the serial chip select pin.</td></tr>
<tr><td>1</td><td>Retain the active state of the serial chip select pin.</td></tr>
</table>
Notes:
− If transmission is disabled (SCR:TXE = "0") and a software reset is performed (SCR:UPCL = "1"), the serial chip select pin will be inactive regardless of the value of this bit.

(Correct)
[bit9] SCAM: Serial chip select Active retention bit
This bit selects whether to retain the active state of the serial chip select pin.
For details, see "5 Operation of Serial Chip Select," "Serial Chip Select Active Level Retention Operation (SCSCR:SCAM = 1) (Valid Only in Master Mode (SCR:MS = 0))."
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Do not retain the active state of the serial chip select pin.</td></tr>
<tr><td>1</td><td>Retain the active state of the serial chip select pin.</td></tr>
</table>
Notes:
− If transmission is disabled (SCR:TXE = 0) and a software reset is performed (SCR:UPCL = 1), the serial chip select pin will be inactive regardless of the value of this bit.
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.9.Serial Chip Select Control Status Register (SCSCR)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit8:6] CDIV2-0 : Serial Chip Select Timing Operating Clock Division Bits
These bits set the division ratio of the serial chip select timing operating clock.
<table>
<tr><th rowspan="2">CDIV2</th><th rowspan="2">CDIV1</th><th rowspan="2">CDIV0</th><th colspan="7">Serial Chip Select Timing Operating Clock</th></tr>
<tr><th>Division Ratio</th><th>Φ= 8 MHz</th><th>Φ= 10 MHz</th><th>Φ= 16 MHz</th><th>Φ= 20 MHz</th><th>Φ= 24 MHz</th><th>Φ= 32 MHz</th></tr>
</table>

(Correct)
[bit8:6] CDIV[2:0]: Serial chip select timing operating clock division bits
These bits set the division ratio of the serial chip select timing operating clock.
<table>
<tr><th rowspan="2">Bits</th><th colspan="7">Serial Chip Select Timing Operating Clock</th></tr>
<tr><th>Division Ratio</th><th>Φ= 8 MHz</th><th>Φ= 10 MHz</th><th>Φ= 16 MHz</th><th>Φ= 20 MHz</th><th>Φ= 24 MHz</th><th>Φ= 32 MHz</th></tr>
</table>
</td></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.9.Serial Chip Select Control Status Register (SCSCR)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
This bit is used for chip select pin 0.
<table>
<tr><th>CSLVL</th><th>Serial Chip Select Level Set Bit</th></tr>
<tr><td>0</td><td>Set the inactive level to "L".</td></tr>
<tr><td>1</td><td>Set the inactive level to "H".</td></tr>
</table>

Notes:
−This bit can be changed only when transmission and reception are disabled (SCR:TXE = RXE = "0").

(Correct)
This bit is used for chip select pin 0.
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Set the inactive level to "L".</td></tr>
<tr><td>1</td><td>Set the inactive level to "H".</td></tr>
</table>

Notes:
−This bit can be changed only when transmission and reception are disabled (SCR:TXE = RXE = 0).
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface)
9.Registers of the CSIO (Clock Synchronous Serial Interface)
9.9.Serial Chip Select Control Status Register (SCSCR)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit4:1] CSEN3-0 : Serial Chip Select Enable Bits
These bits enable or disable each serial chip select pin.
The CSEN3 bit corresponds to the SCS3 pin, the CSEN2 bit to the SCS2 pin, the CSEN1 bit to the SCS1 pin, and the CSEN0 bit to the SCS0 pin.
In slave mode (SCR:MS = 1), only the CSEN0 bit is used to enable or disable serial chip pins.
<table>
<tr><th>CSEN</th><th>Serial Chip Select Enable Bit</th></tr>
<tr><td>0</td><td>Disable the operation of the serial chip select pin.</td></tr>
<tr><td>1</td><td>Enable the operation of the serial chip select pin.</td></tr>
</table>
Notes:
−These bits can be changed only when transmission and reception are disabled (SCR:TXE = RXE = "0").
−In master mode (SCR:MS = 0), if CSEN3 to CSEN0 are set to "0000"b, transmission/reception operations are performed regardless of the status of the serial chip select pin.

(Correct)
[bit4:1] CSEN3, CSEN2, CSEN1, CSEN0: Serial chip select enable bits
These bits enable or disable each serial chip select pin.
The CSEN3 bit corresponds to the SCS3 pin, the CSEN2 bit to the SCS2 pin, the CSEN1 bit to the SCS1 pin, and the CSEN0 bit to the SCS0 pin.
In slave mode (SCR:MS = 1), only the CSEN0 bit is used to enable or disable serial chip pins.
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Disable the operation of the serial chip select pin.</td></tr>
<tr><td>1</td><td>Enable the operation of the serial chip select pin.</td></tr>
</table>
Notes:
− These bits can be changed only when transmission and reception are disabled (SCR:TXE = RXE = 0).
− In master mode (SCR:MS = 0), if CSEN3 to CSEN0 are set to 0b0000, transmission/reception operations are performed regardless of the status of the serial chip select pin.
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.9.Serial Chip Select Control Status Register (SCSCR)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
This bit enables or disables the output of serial chip select pins.
<table>
<tr><th>CSOE</th><th>Serial Chip Select Output Enable Bit</th></tr>
<tr><td>0</td><td>Disable the output of all serial chip select pins.</td></tr>
<tr><td>1</td><td>Enable the output of all serial chip select pins.</td></tr>
</table>
Notes:
−This bit can be changed only when transmission and reception are disabled (SCR:TXE = RXE = "0").
−In slave mode (SCR:MS = "1"), set this bit to "0".

(Correct)
This bit enables or disables the output of serial chip select pins.
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Disable the output of all serial chip select pins.</td></tr>
<tr><td>1</td><td>Enable the output of all serial chip select pins.</td></tr>
</table>
Notes:
−This bit can be changed only when transmission and reception are disabled (SCR:TXE = RXE = 0).
−In slave mode (SCR:MS = 1), set this bit to "0".
</td></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.10.Serial Chip Select Timing Registers (SCSTR3 to SCSTR0)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit15:8] CSSU7-0 : Serial Chip Select Setup Delay Bits
These bits set the time from the instant that the serial chip select pin becomes active until the serial clock is output. If "00"h is set in these bits, the serial clock is output at the same time as the serial chip select pin becomes active.
<table>
<tr><th>CSSU7</th><th>CSSU6</th><th>CSSU5</th><th>CSSU4</th><th>CSSU3</th><th>CSSU2</th><th>CSSU1</th><th>CSSU0</th><th>Setup Delay Time</th></tr>
</table>
(Correct)
[bit15:8] CSSU[7:0]: Serial chip select setup delay bits
These bits set the time from the instant that the serial chip select pin becomes active until the serial clock is output. If 0x00 is set in these bits, the serial clock is output at the same time as the serial chip select pin becomes active.
<table>
<tr><th>Bits</th><th>Setup Delay Time</th></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.10.Serial Chip Select Timing Registers (SCSTR3 to SCSTR0)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
Notes:
−These bits can be changed only when transmission and reception are disabled (SCR:TXE = RXE = "0").
−In slave mode (SCR:MS = 1), the settings of these bits are invalid.

[bit7:0] CSHD7-0 : Serial Chip Select Hold Delay Bits

(Correct)
Notes:
−These bits can be changed only when transmission and reception are disabled (SCR:TXE = RXE = 0).
−In slave mode (SCR:MS = 1), the settings of these bits are invalid.

[bit7:0] CSHD[7:0]: Serial chip select hold delay bits</td></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.10.Serial Chip Select Timing Registers (SCSTR3 to SCSTR0)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit7:0] CSHD7-0 : Serial Chip Select Hold Delay Bits
These bits set the time from the instant that the output of the serial clock ends until the serial chip select pin becomes inactive.
If these bits are set to "00"h, the output of the serial clock ends at the same time as the serial chip select pin becomes inactive.
<table>
<tr><th>CSHD7</th><th>CSHD6</th><th>CSHD5</th><th>CSHD4</th><th>CSHD3</th><th>CSHD2</th><th>CSHD1</th><th>CSHD0</th><th>Hold Delay Time</th></tr>
</table>

(Correct)
[bit7:0] CSHD[7:0]: Serial chip select hold delay bits
These bits set the time from the instant that the output of the serial clock ends until the serial chip select pin becomes inactive.
If these bits are set to 0x00, the output of the serial clock ends at the same time as the serial chip select pin becomes inactive.
<table>
<tr><th>Bits</th><th>Hold Delay Time</th></tr>
</table>
</td></tr>
</table>

Section	Change Results
CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.10.Serial Chip Select Timing Registers (SCSTR3 to SCSTR0)	Features of "For details" should be corrected as indicated by the shading below. (Error) Notes: −These bits can be changed only when transmission and reception are disabled (SCR:TXE = RXE = "0"). −In slave mode (SCR:MS = 1), the settings of these bits are invalid. Figure 9-12 Bit Configuration of Serial Chip Select Timing Registers (SCSTR3, SCSTR2) (Correct) Notes: −These bits can be changed only when transmission and reception are disabled (SCR:TXE = RXE = 0). −In slave mode (SCR:MS = 1), the settings of these bits are invalid. Figure 9-12 Bit Configuration of Serial Chip Select Timing Registers (SCSTR3, SCSTR2)
CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.10.Serial Chip Select Timing Registers (SCSTR3 to SCSTR0)	Features of "For details" should be corrected as indicated by the shading below. (Error) [bit15:0] CSDS15-0 : Serial Chip Deselect Bits (Correct) [bit15:0] CSDS[15:0]: Serial chip deselect bits

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.11.Serial Chip Select Format Registers (SCSFR2 to SCSFR0)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
When the chip select data format is enabled (ESCR:CSFE = 1), this bit selects the inactive-time level of serial chip select pin 2.
<table>
<tr><th>CS2CSLVL</th><th>Serial Chip Select Pin 2
Serial Chip Select Level Set Bit</th></tr>
<tr><td>0</td><td>Set the inactive level to "L".</td></tr>
<tr><td>1</td><td>Set the inactive level to "H".</td></tr>
</table>
Notes:
−This bit can be changed only when transmission and reception are disabled (SCR:TXE = RXE = "0").

(Correct)
When the chip select data format is enabled (ESCR:CSFE = 1), this bit selects the inactive-time level of serial chip select pin 2.
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Set the inactive level to "L".</td></tr>
<tr><td>1</td><td>Set the inactive level to "H".</td></tr>
</table>
Notes:
−This bit can be changed only when transmission and reception are disabled (SCR:TXE = RXE = 0).</td></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.11.Serial Chip Select Format Registers (SCSFR2 to SCSFR0)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■In normal transfer, reception data is sampled at a falling edge of the serial clock. In SPI transfer, it is sampled at a rising edge of the serial clock.
<table>
<tr><th>CS2SCINV</th><th>Serial Chip Select Pin 2
Serial Clock Invert Bit</th></tr>
<tr><td>0</td><td>Mark level "H" format</td></tr>
<tr><td>1</td><td>Mark level "L" format</td></tr>
</table>
Notes:
−This bit can be changed only when transmission and reception are disabled (SCR:TXE = RXE = "0").

(Correct)
■In normal transfer, reception data is sampled at a falling edge of the serial clock. In SPI transfer, it is sampled at a rising edge of the serial clock.
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Mark level "H" format</td></tr>
<tr><td>1</td><td>Mark level "L" format</td></tr>
</table>
Notes:
−This bit can be changed only when transmission and reception are disabled (SCR:TXE = RXE = 0).</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.11.Serial Chip Select Format Registers (SCSFR2 to SCSFR0)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■1: Support SPI.<table><tr><th>CS2SPI</th><th>Serial Chip Select Pin 2
SPI Support Bit</th></tr><tr><td>0</td><td>Normal synchronous transfer</td></tr><tr><td>1</td><td>SPI support</td></tr></table>Notes:
−This bit can be changed only when transmission and reception are disabled (SCR:TXE = RXE = "0").

(Correct)
■1: Support SPI.<table><tr><th>Bit</th><th>Description</th></tr><tr><td>0</td><td>Normal synchronous transfer</td></tr><tr><td>1</td><td>SPI support</td></tr></table>Notes:
−This bit can be changed only when transmission and reception are disabled (SCR:TXE = RXE = 0).</td></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.11.Serial Chip Select Format Registers (SCSFR2 to SCSFR0)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
When the chip select data format is enabled (ESCR:CSFE = 1), this bit selects the transfer of transfer serial data when serial chip select pin 2 is active, setting either transfer of the lowest bit first (LSB first, BDS = 0) or transfer of the highest bit first (MSB first, BDS = 1).<table><tr><th>CS2BDS</th><th>Serial Chip Select Pin 2
Transfer Direction Select Bit</th></tr><tr><td>0</td><td>LSB first (lowest bit transferred first)</td></tr><tr><td>1</td><td>MSB first (highest bit transferred first)</td></tr></table>Notes:
−This bit can be changed only when transmission and reception are disabled (SCR:TXE = RXE = "0").

(Correct)
When the chip select data format is enabled (ESCR:CSFE = 1), this bit selects the transfer of transfer serial data when serial chip select pin 2 is active, setting either transfer of the lowest bit first (LSB first, BDS = 0) or transfer of the highest bit first (MSB first, BDS = 1).<table><tr><th>Bit</th><th>Description</th></tr><tr><td>0</td><td>LSB first (lowest bit transferred first)</td></tr><tr><td>1</td><td>MSB first (highest bit transferred first)</td></tr></table>Notes:
−This bit can be changed only when transmission and reception are disabled (SCR:TXE = RXE = 0).</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.11.Serial Chip Select Format Registers (SCSFR2 to SCSFR0)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit11:8] CS2L3, L2, L1, L0: Bits for selecting the data length of serial chip select pin 2
When the chip select data format is enabled (ESCR:CSFE = 1), these bits specify the data length of the transmission/reception data to be assumed when serial chip select pin 2 is active.
<table>
<tr><th>CS2
L3</th><th>CS2
L2</th><th>CS2
L1</th><th>CS2
L0</th><th>Serial Chip Select Pin 2
Data Length Selection Bit</th></tr>
</table>

(Correct)
[bit11:8] CS2L[3:0]: Bits for Selecting the Data Length of Serial Chip Select Pin 2
When the chip select data format is enabled (ESCR:CSFE = 1), these bits specify the data length of the transmission/reception data to be assumed when serial chip select pin 2 is active.
<table>
<tr><th>Bits</th><th>Data Length</th></tr>
</table>
</td></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.11.Serial Chip Select Format Registers (SCSFR2 to SCSFR0)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
When the chip select data format is enabled (ESCR:CSFE = 1), this bit selects the inactive-time level of serial chip select pin 1.
<table>
<tr><th>CS1CSLVL</th><th>Serial Chip Select Pin 1
Serial Chip Select Level Set Bit</th></tr>
<tr><td>0</td><td>Set the inactive level to "L".</td></tr>
<tr><td>1</td><td>Set the inactive level to "H".</td></tr>
</table>

Notes:
−This bit can be changed only when transmission and reception are disabled (SCR:TXE = RXE = "0").

(Correct)
When the chip select data format is enabled (ESCR:CSFE = 1), this bit selects the inactive-time level of serial chip select pin 1.
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Set the inactive level to "L".</td></tr>
<tr><td>1</td><td>Set the inactive level to "H".</td></tr>
</table>

Notes:
−This bit can be changed only when transmission and reception are disabled (SCR:TXE = RXE = 0).
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.11.Serial Chip Select Format Registers (SCSFR2 to SCSFR0)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■In normal transfer, reception data is sampled at a falling edge of the serial clock. In SPI transfer, it is sampled at a rising edge of the serial clock.
<table>
<tr><th>CS1SCINV</th><th>Serial Chip Select Pin 1
Serial Clock Invert Bit</th></tr>
<tr><td>0</td><td>Mark level "H" format</td></tr>
<tr><td>1</td><td>Mark level "L" format</td></tr>
</table>
Notes:
−This bit can be changed only when transmission and reception are disabled (SCR:TXE = RXE = "0").

(Correct)
■In normal transfer, reception data is sampled at a falling edge of the serial clock. In SPI transfer, it is sampled at a rising edge of the serial clock.
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Mark level "H" format</td></tr>
<tr><td>1</td><td>Mark level "L" format</td></tr>
</table>
Notes:
−This bit can be changed only when transmission and reception are disabled (SCR:TXE = RXE = 0).</td></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.11.Serial Chip Select Format Registers (SCSFR2 to SCSFR0)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■1: Support SPI.
<table>
<tr><th>CS1SPI</th><th>Serial Chip Select Pin 1
SPI-supporting Bit</th></tr>
<tr><td>0</td><td>Normal synchronous transfer</td></tr>
<tr><td>1</td><td>SPI support</td></tr>
</table>
Notes:
−This bit can be changed only when transmission and reception are disabled (SCR:TXE = RXE = "0").

(Correct)
■1: Support SPI.
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Normal synchronous transfer</td></tr>
<tr><td>1</td><td>SPI support</td></tr>
</table>
Notes:
−This bit can be changed only when transmission and reception are disabled (SCR:TXE = RXE = 0).</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.11.Serial Chip Select Format Registers (SCSFR2 to SCSFR0)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
When the chip select data format is enabled (ESCR:CSFE = 1), this bit selects the transfer of transfer serial data when serial chip select pin 1 is active, setting either transfer of the lowest bit first (LSB first, BDS = 0) or transfer of the highest bit first (MSB first, BDS = 1).
<table>
<tr><th>CS1BDS</th><th>Serial Chip Select Pin 1
Transfer Direction Select Bit</th></tr>
<tr><td>0</td><td>LSB first (lowest bit transferred first)</td></tr>
<tr><td>1</td><td>MSB first (highest bit transferred first)</td></tr>
</table>
Notes:
−This bit can be changed only when transmission and reception are disabled (SCR:TXE = RXE = "0").

(Correct)
When the chip select data format is enabled (ESCR:CSFE = 1), this bit selects the transfer of transfer serial data when serial chip select pin 1 is active, setting either transfer of the lowest bit first (LSB first, BDS = 0) or transfer of the highest bit first (MSB first, BDS = 1).
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>LSB first (lowest bit transferred first)</td></tr>
<tr><td>1</td><td>MSB first (highest bit transferred first)</td></tr>
</table>
Notes:
−This bit can be changed only when transmission and reception are disabled (SCR:TXE = RXE = 0).
</td></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.11.Serial Chip Select Format Registers (SCSFR2 to SCSFR0)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit3:0] CS3L3, L2, L1, L0: Bits for selecting the data length of serial chip select pin 3
When the chip select data format is enabled (ESCR:CSFE = 1), these bits specify the data length of the transmission/reception data to be assumed when serial chip select pin 3 is active.
<table>
<tr><th>CS1
L3</th><th>CS1
L2</th><th>CS1
L1</th><th>CS1
L0</th><th>Serial Chip Select Pin 1
Data Length Selection Bit</th></tr>
</table>
(Correct)
[bit3:0] CS1L[3:0]: Bits for Selecting the Data Length of Serial Chip Select Pin 1
When the chip select data format is enabled (ESCR:CSFE = 1), these bits specify the data length of the transmission/reception data to be assumed when serial chip select pin 1 is active.
<table>
<tr><th>Bits</th><th>Data Lengtht</th></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.11.Serial Chip Select Format Registers (SCSFR2 to SCSFR0)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
When the chip select data format is enabled (ESCR:CSFE = 1), this bit selects the inactive-time level of serial chip select pin 3.
<table>
<tr><th>CS3CSLVL</th><th>Serial Chip Select Pin 3
Serial Chip Select Level Set</th></tr>
<tr><td>0</td><td>Set the inactive level to "L".</td></tr>
<tr><td>1</td><td>Set the inactive level to "H".</td></tr>
</table>
Notes:
−This bit can be changed only when transmission and reception are disabled (SCR:TXE = RXE = "0").

(Correct)
When the chip select data format is enabled (ESCR:CSFE = 1), this bit selects the inactive-time level of serial chip select pin 3.
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Set the inactive level to "L".</td></tr>
<tr><td>1</td><td>Set the inactive level to "H".</td></tr>
</table>
Notes:
−This bit can be changed only when transmission and reception are disabled (SCR:TXE = RXE = 0).</td></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.11.Serial Chip Select Format Registers (SCSFR2 to SCSFR0)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■In normal transfer, reception data is sampled at a falling edge of the serial clock. In SPI transfer, it is sampled at a rising edge of the serial clock.
<table>
<tr><th>CS3SCINV</th><th>Serial Chip Select Pin 3
Clock Invert Bit</th></tr>
<tr><td>0</td><td>Mark level "H" format</td></tr>
<tr><td>1</td><td>Mark level "L" format</td></tr>
</table>
Notes:
−This bit can be changed only when transmission and reception are disabled (SCR:TXE = RXE = "0").

(Correct)
■In normal transfer, reception data is sampled at a falling edge of the serial clock. In SPI transfer, it is sampled at a rising edge of the serial clock.
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Mark level "H" format</td></tr>
<tr><td>1</td><td>Mark level "L" format</td></tr>
</table>
Notes:
−This bit can be changed only when transmission and reception are disabled (SCR:TXE = RXE = 0).</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.11.Serial Chip Select Format Registers (SCSFR2 to SCSFR0)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■1: Support SPI.<table><tr><th>CS3SPI</th><th>Serial Chip Select Pin 3
SPI-supporting Bit</th></tr><tr><td>0</td><td>Normal synchronous transfer</td></tr><tr><td>1</td><td>SPI support</td></tr></table>Notes:
−This bit can be changed only when transmission and reception are disabled (SCR:TXE = RXE = "0").

(Correct)
■1: Support SPI.<table><tr><th>Bit</th><th>Description</th></tr><tr><td>0</td><td>Normal synchronous transfer</td></tr><tr><td>1</td><td>SPI support</td></tr></table>Notes:
−This bit can be changed only when transmission and reception are disabled (SCR:TXE = RXE = 0).</td></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.11.Serial Chip Select Format Registers (SCSFR2 to SCSFR0)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
When the chip select data format is enabled (ESCR:CSFE = 1), this bit selects the transfer of transfer serial data when serial chip select pin 3 is active, setting either transfer of the lowest bit first (LSB first, BDS = 0) or transfer of the highest bit first (MSB first, BDS = 1).<table><tr><th>CS3BDS</th><th>Serial Chip Select Pin 3
Transfer Direction Select Bit</th></tr><tr><td>0</td><td>LSB first (lowest bit transferred first)</td></tr><tr><td>1</td><td>MSB first (highest bit transferred first)</td></tr></table>Notes:
−This bit can be changed only when transmission and reception are disabled (SCR:TXE = RXE = "0").

(Correct)
When the chip select data format is enabled (ESCR:CSFE = 1), this bit selects the transfer of transfer serial data when serial chip select pin 3 is active, setting either transfer of the lowest bit first (LSB first, BDS = 0) or transfer of the highest bit first (MSB first, BDS = 1).<table><tr><th>Bit</th><th>Description</th></tr><tr><td>0</td><td>LSB first (lowest bit transferred first)</td></tr><tr><td>1</td><td>MSB first (highest bit transferred first)</td></tr></table>Notes:
−This bit can be changed only when transmission and reception are disabled (SCR:TXE = RXE = 0).</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.11.Serial Chip Select Format Registers (SCSFR2 to SCSFR0)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit3:0] CS3L3, L2, L1, L0: Bits for selecting the data length of serial chip select pin 3
When the chip select data format is enabled (ESCR:CSFE = 1), these bits specify the data length of the transmission/reception data to be assumed when serial chip select pin 3 is active.
<table>
<tr><th>CS3 L3</th><th>CS3 L2</th><th>CS3 L1</th><th>CS3 L0</th><th>Serial Chip Select Pin 3 Data Length Selection Bit</th></tr>
</table>
(Correct)
[bit3:0] CS3L[3:0]: Bits for Selecting the Data Length of Serial Chip Select Pin 3
When the chip select data format is enabled (ESCR:CSFE = 1), these bits specify the data length of the transmission/reception data to be assumed when serial chip select pin 3 is active.
<table>
<tr><th>Bits</th><th>Data Length</th></tr>
</table></td></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.12.Transfer Byte Registers (TBYTE3 to TBYTE0)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
If the value of one of these bits is changed during transmission (SSR:TBI = 0), the new number of transfer data items becomes valid after the end of transmission with the old number of transfer data items.
<table>
<tr><th>TBYTE</th><th>Transfer Byte Register</th></tr>
<tr><td>Write</td><td>Writing to TBYTE</td></tr>
<tr><td>Read</td><td>Setting of TBYTE</td></tr>
</table>
Notes:
− If one of these bits is set to (00)h, the transfer count will be 8.

(Correct)
If the value of one of these bits is changed during transmission (SSR:TBI = 0), the new number of transfer data items becomes valid after the end of transmission with the old number of transfer data items.
[bit15] TBYTE: Transfer byte
<table>
<tr><th>TBYTE</th><th>Description</th></tr>
<tr><td>Write</td><td>Writing to TBYTE</td></tr>
<tr><td>Read</td><td>Setting of TBYTE</td></tr>
</table>
Notes:
− If one of these bits is set to 0x00, the transfer count will be 8.</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.13.Baud Rate Generator Registers 1, 0 (BGR1, BGR0)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit14:8] BGR1 : Baud rate generator register 1
<table>
<tr><th>BGR1</th><th>Baud Rate Generator Register 1</th></tr>
<tr><td>Write</td><td>Write to reload counter bits 8 to 14.</td></tr>
<tr><td>Read</td><td>Read the setting value of BGR1.</td></tr>
</table>

[bit7:0] BGR0: Baud rate generator register 0
<table>
<tr><th>BGR0</th><th>Baud Rate Generator Register 0</th></tr>
<tr><td>Write</td><td>Write to reload counter bits 0 to 7.</td></tr>
<tr><td>Read</td><td>Read the setting value of BGR0.</td></tr>
</table>

Notes:
−Use 16-bit access for writing a value to baud rate generator register (BGR1, BGR0).
−If the reload value is even-numbered, the "H" width and the "L" width of the serial clock will be as described below, depending on the setting of the SCINV bit. If it is odd-numbered, the "H" width and the "L" width of the serial clock will be the same.
When SMR:SCINV = "0", the "H" width of the serial clock will be longer by 1 cycle of the bus clock.
When SMR:SCINV = "1", the "L" width of the serial clock will be longer by 1 cycle of the bus clock.
−Set the reload value to 2 or more in master mode and to 3 or more in slave mode.
−If the set values of the baud rate generator registers (BGR1, BGR0) are changed, the new values are reloaded when the counter value becomes "00000"h. Thus, to have the new settings take effect immediately, execute a CSIO reset (SCR:UPCL) after changing the settings of BGR1/0.

(Correct)
[bit14:8] BGR1 : Baud Rate Generator Register 1
<table>
<tr><th>BGR1</th><th>Description</th></tr>
<tr><td>Write</td><td>Write to reload counter bits 8 to 14.</td></tr>
<tr><td>Read</td><td>Read the setting value of BGR1.</td></tr>
</table>

[bit7:0] BGR0: Baud Rate Generator Register 0
<table>
<tr><th>BGR0</th><th>Description</th></tr>
<tr><td>Write</td><td>Write to reload counter bits 0 to 7.</td></tr>
<tr><td>Read</td><td>Read the setting value of BGR0.</td></tr>
</table>

Notes:
−Use 16-bit access for writing a value to baud rate generator register (BGR1, BGR0).
−If the reload value is even-numbered, the "H" width and the "L" width of the serial clock will be as described below, depending on the setting of the SCINV bit. If it is odd-numbered, the "H" width and the "L" width of the serial clock will be the same.
When SMR:SCINV = 0, the "H" width of the serial clock will be longer by 1 cycle of the bus clock.
When SMR:SCINV = 1, the "L" width of the serial clock will be longer by 1 cycle of the bus clock.
−Set the reload value to 2 or more in master mode and to 3 or more in slave mode.
−If the set values of the baud rate generator registers (BGR1, BGR0) are changed, the new values are reloaded when the counter value becomes 0x00000. Thus, to have the new settings take effect immediately, execute a CSIO reset (SCR:UPCL) after changing the settings of BGR1/0.
−When the reception FIFO is used, set the reception FIFO idle detection enable bit (FCR1:FRIIE) to "1", and if performing operation in slave mode, set baud rates in BGR1/0.
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.14.FIFO Control Register 1 (FCR1)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
"1": Enable FLST bit detection.
<table><tr><th>FLSTE</th><th>Retransmission Data Lost Detection Enable Bit</th></tr></table>
(Correct)
"1": Enable FLST bit detection.
<table><tr><th>Bit</th><th>Description</th></tr></table></td></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.14.FIFO Control Register 1 (FCR1)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
"1": Enable the detection of reception idle status.
<table><tr><th>FRIIE</th><th>Reception FIFO Idle Detection Enable Bit</th></tr></table>
(Correct)
"1": Enable the detection of reception idle status.
<table><tr><th>Bit</th><th>Description</th></tr></table></td></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.14.FIFO Control Register 1 (FCR1)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■For ECR:TXBLKEN = 1
−FTICR setting value ≧ number of vacant data items in the transmission FIFO
<table><tr><th>FDRQ</th><th>Transmission FIFO Data Request Bit</th></tr></table>
(Correct)
■For ECR:TXBLKEN = 1
−FTICR setting value ≥ number of vacant data items in the transmission FIFO
<table><tr><th>Bit</th><th>Description</th></tr></table></td></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.14.FIFO Control Register 1 (FCR1)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■This bit is set when the FCR1S:FTIES bit in the set register is set to "1".
<table><tr><th>FTIE</th><th>Transmission FIFO Interrupt Enable Bit</th></tr></table>
(Correct)
■This bit is set when the FCR1S:FTIES bit in the set register is set to "1".
<table><tr><th>Bit</th><th>Description</th></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.14.FIFO Control Register 1 (FCR1)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
"1": Allocate as the transmission FIFO: FIFO2, and as the reception FIFO: FIFO1.
<table><tr><th>FSEL</th><th>FIFO Selection Bit</th></tr></table>
(Correct)
"1": Allocate as the transmission FIFO: FIFO2, and as the reception FIFO: FIFO1.
<table><tr><th>Bit</th><th>Description</th></tr></table></td></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.15.FIFO Control Register 0 (FCR0)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
If "1" is set in this bit, the data pointed to by the read pointer saved with the FSET bit is overwritten. For this reason, re-transmission with the FLD bit cannot be set even if an error occurs. When performing retransmission with this bit set to "1", reset the FIFO and then write the data to the FIFO again.
<table><tr><th>FLST</th><th>FIFO Retransmission Data Lost Flag Bit</th></tr></table>
(Correct)
If "1" is set in this bit, the data pointed to by the read pointer saved with the FSET bit is overwritten. For this reason, re-transmission with the FLD bit cannot be set even if an error occurs. When performing retransmission with this bit set to "1", reset the FIFO and then write the data to the FIFO again.
<table><tr><th>Bit</th><th>Description</th></tr></table></td></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.15.FIFO Control Register 0 (FCR0)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■This bit is set when the FCR0S:FLDS bit in the set register is set to "1".
<table><tr><th>FLD</th><th>FIFO Pointer Reload Bit</th></tr></table>
(Correct)
■This bit is set when the FCR0S:FLDS bit in the set register is set to "1".
<table><tr><th>Bit</th><th>Description</th></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.15.FIFO Control Register 0 (FCR0)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
"0": No effect.<table><tr><th rowspan="2">FSET</th><th colspan="2">FIFO Retransmission Data Lost Flag Bit</th></tr><tr><th>Write</th><th>Read</th></tr></table>
(Correct)
"0": No effect.<table><tr><th rowspan="2">Bit</th><th colspan="2">Description</th></tr><tr><th>Write</th><th>Read</th></tr></table></td></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.15.FIFO Control Register 0 (FCR0)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■This bit is set when the FCR0S:FCL2S bit in the set register is set to "1".<table><tr><th rowspan="2">FCL2</th><th colspan="2">FIFO2 Reset Bit</th></tr><tr><th>Write</th><th>Read</th></tr></table>
(Correct)
■This bit is set when the FCR0S:FCL2S bit in the set register is set to "1".<table><tr><th rowspan="2">Bit</th><th colspan="2">Description</th></tr><tr><th>Write</th><th>Read</th></tr></table></td></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.15.FIFO Control Register 0 (FCR0)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■This bit is set when the FCR0S:FCL1S bit in the set register is set to "1".<table><tr><th rowspan="2">FCL1</th><th colspan="2">FIFO1 Reset Bit</th></tr><tr><th>Write</th><th>Read</th></tr></table>
(Correct)
■This bit is set when the FCR0S:FCL1S bit in the set register is set to "1".<table><tr><th rowspan="2">Bit</th><th colspan="2">Description</th></tr><tr><th>Write</th><th>Read</th></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface)
9.Registers of the CSIO (Clock Synchronous Serial Interface)
9.15.FIFO Control Register 0 (FCR0)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■To use FIFO1 as the transmission or reception FIFO, set this bit to "1" or "0" when the transmission or reception buffer is empty ((SSR:TDRE = "1") or (SSR:RDRF = "0")), respectively.
■To use FIFO1 as a reception FIFO, first disable reception (SCR:RXE = 0) and then change the setting of this bit when the reception buffer is empty (SSR:RDRF = "0").
■Disabling FIFO2 does not change the state of FIFO2.
■This bit is reset when the FCR0C:FE2C bit in the clear register is set to "1".
■This bit is set when the FCR0S:FE2S bit in the set register is set to "1".
<table><tr><th>FE2</th><th>FIFO2 Operation Enable Bit</th></tr></table>
(Correct)
■To use FIFO1 as the transmission or reception FIFO, set this bit to "1" or "0" when the transmission or reception buffer is empty ((SSR:TDRE = 1) or (SSR:RDRF = 0)), respectively.
■To use FIFO1 as a reception FIFO, first disable reception (SCR:RXE = 0) and then change the setting of this bit when the reception buffer is empty (SSR:RDRF = 0).
■Disabling FIFO2 does not change the state of FIFO2.
■This bit is reset when the FCR0C:FE2C bit in the clear register is set to "1".
■This bit is set when the FCR0S:FE2S bit in the set register is set to "1".
<table><tr><th>Bit</th><th>Description</th></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.15.FIFO Control Register 0 (FCR0)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■To use FIFO1 as the transmission or reception FIFO, set this bit to "1" or "0" when the transmission or reception buffer is empty ((SSR:TDRE = "1") or (SSR:RDRF = "0")), respectively.
■To use FIFO1 as a reception FIFO, first disable reception (SCR:RXE = 0) and then change the setting of this bit when the reception buffer is empty (SSR:RDRF = "0").
■Disabling FIFO1 does not change the state of FIFO1.
■This bit is reset when the FCR0C:FE1C bit in the clear register is set to "1".
■This bit is set when bit FCR0S:FE1S in the set register is set to "1".
<table><tr><th>FE1</th><th>FIFO1 Operation Enable Bit</th></tr></table>
(Correct)
■To use FIFO1 as the transmission or reception FIFO, set this bit to "1" or "0" when the transmission or reception buffer is empty ((SSR:TDRE = 1) or (SSR:RDRF = 0)), respectively.
■To use FIFO1 as a reception FIFO, first disable reception (SCR:RXE = 0) and then change the setting of this bit when the reception buffer is empty (SSR:RDRF = 0).
■Disabling FIFO1 does not change the state of FIFO1.
■This bit is reset when the FCR0C:FE1C bit in the clear register is set to "1".
■This bit is set when bit FCR0S:FE1S in the set register is set to "1".
<table><tr><th>Bit</th><th>Description</th></tr></table></td></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.16.FIFO Byte Register (FBYTE)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
FBYTE2, FBYTE1: FIFO2 data count indication bit, FIFO1 data count indication bit
<table><tr><td>Write</td><td>Set the transfer count.</td></tr><tr><td>Read</td><td>Read valid data count.</td></tr></table>
(Correct)
[bit15:8] FBYTE2: FIFO2 Data Count Indication Bit
[bit7:0] FBYTE1: FIFO1 Data Count Indication Bit
<table><tr><th>FBYTE2, FBYTE1</th><th>Description</th></tr><tr><td>Write</td><td>Set the transfer count.</td></tr><tr><td>Read</td><td>Read valid data count.</td></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.17.Transmission FIFO Interrupt Control Register (FTICR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
FTICR2, FTICR1: FIFO2 data count indication bit, FIFO1 data count indication bit
<table><tr><td>Write</td><td>Set the number of valid data items that generate an interrupt.</td></tr><tr><td>Read</td><td>Read valid data count.</td></tr></table>
(Correct)
FTICR2: FIFO2 Data Count Indication Bit
FTICR1: FIFO1 Data Count Indication Bit
<table><tr><th>FTICR2, FTICR1</th><th>Description</th></tr><tr><td>Write</td><td>Set the number of valid data items that generate an interrupt.</td></tr><tr><td>Read</td><td>Read valid data count.</td></tr></table></td></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.18.Extended Control Register (ECR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
This bit switches the output of the error interrupts from the interrupt request pin and instead outputs them through the error interrupt request pin. For details on interrupt output selection, see Table 2-2 and Table 2-3.
<table><tr><th>EISEL</th><th>Error Interrupt Request Output Selection Bit</th></tr></table>
(Correct)
This bit switches the output of the error interrupts from the interrupt request pin and instead outputs them through the error interrupt request pin. For details on interrupt output selection, see Table 2-2 and Table 2-3.
<table><tr><th>Bit</th><th>Description</th></tr></table></td></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.18.Extended Control Register (ECR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
For details on the target interrupt factors, see Table 2-3.
<table><tr><th>REIE</th><th>Reception Error Interrupt Enable Bit</th></tr></table>
(Correct)
For details on the target interrupt factors, see Table 2-3.
<table><tr><th>Bit</th><th>Description</th></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.18.Extended Control Register (ECR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
For details on the target interrupt factors, see Table 2-2.
<table><tr><td>TEIE</td><td>Transmission Error Interrupt Enable Bit</td></tr></table>
(Correct)
For details on the target interrupt factors, see Table 2-2.
<table><tr><td>Bit</td><td>Description</td></tr></table></td></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.18.Extended Control Register (ECR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■"1": Perform DMA transfer in block transfer mode.
<table><tr><td>RXBLKEN</td><td>Reception Block Transfer Set Bit</td></tr></table>
(Correct)
■"1": Perform DMA transfer in block transfer mode.
<table><tr><td>Bit</td><td>Description</td></tr></table></td></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.18.Extended Control Register (ECR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■"1": Perform DMA transfer in block transfer mode.
<table><tr><td>TXBLKEN</td><td>Transmission Block Transfer Set Bit</td></tr></table>
(Correct)
■"1": Perform DMA transfer in block transfer mode.
<table><tr><td>Bit</td><td>Description</td></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.19.Extended Status Register (ESR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■Please refer to Table 2-3 for output of interrupt request.
<table>
<tr><th>RXUDR</th><th>Reception FIFO Under run Flag Bit</th></tr>
<tr><td>0</td><td>Indicate that reception FIFO under run is not occurring.</td></tr>
<tr><td>1</td><td>Indicate that reception FIFO under run is occurring.</td></tr>
</table>
Notes:
−Software reset (SCR:UPCL = "1") resets this bit to "0".

(Correct)
■Please refer to Table 2-3 for output of interrupt request.
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Indicate that reception FIFO under run is not occurring.</td></tr>
<tr><td>1</td><td>Indicate that reception FIFO under run is occurring.</td></tr>
</table>
Notes:
−Software reset (SCR:UPCL = 1) resets this bit to "0"</td></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.19.Extended Status Register (ESR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■Please refer to Table 2-2 for output of interrupt request.
<table>
<tr><th>TXOVR</th><th>Transmission FIFO Overrun Flag Bit</th></tr>
<tr><td>0</td><td>Indicate that transmission FIFO overrun is not occurring.</td></tr>
<tr><td>1</td><td>Indicate that transmission FIFO overrun is occurring.</td></tr>
</table>
Notes:
−Software reset (SCR:UPCL = "1") resets this bit to "0".

(Correct)
■Please refer to Table 2-2 for output of interrupt request.
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Indicate that transmission FIFO overrun is not occurring.</td></tr>
<tr><td>1</td><td>Indicate that transmission FIFO overrun is occurring.</td></tr>
</table>
Notes:
−Software reset (SCR:UPCL = 1) resets this bit to "0"</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.19.Extended Status Register (ESR)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
■Please refer to Table 2-3 for output of interrupt request.
<table>
<tr><th>RBERR</th><th>Reception Block Transfer Error Bit</th></tr>
<tr><td>0</td><td>Indicate that a reception block transfer error is not occurring.</td></tr>
<tr><td>1</td><td>Indicate that a reception block transfer error is occurring.</td></tr>
</table>
Notes:
−Software reset (SCR:UPCL = "1") resets this bit to "0".
−If reception block transfer enable (ECR:RXBLKEN) is "0", this bit is set to "0".
−If reception FIFO enable is "0"(FE2 = "0" for FSEL = "0", FE1 = "0" for FSEL = "1"), this bit is set to "0".
−If the reception FIFO is reset (FCL2 = "1" FSEL = "0", FCL1 = "1" for FSEL = "1"), this bit is set to "0".

(Correct)
■Please refer to Table 2-3 for output of interrupt request.
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Indicate that a reception block transfer error is not occurring.</td></tr>
<tr><td>1</td><td>Indicate that a reception block transfer error is occurring.</td></tr>
</table>
Notes:
−Software reset (SCR:UPCL = 1) resets this bit to "0".
−If reception block transfer enable (ECR:RXBLKEN) is "0", this bit is set to "0".
−If reception FIFO enable is "0"(FE2 = 0 for FSEL = 0, FE1 = 0 for FSEL = 1), this bit is set to "0".
−If the reception FIFO is reset (FCL2 = 1 FSEL = 0, FCL1 = 1 for FSEL = 1), this bit is set to "0".
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.19.Extended Status Register (ESR)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
■Please refer to Table 2-2 for output of interrupt request.
<table>
<tr><th>TBERR</th><th>Transmission Block Transfer Error Bit</th></tr>
<tr><td>0</td><td>Indicate that a transmission block transfer error is not occurring.</td></tr>
<tr><td>1</td><td>Indicate that a transmission block transfer error is occurring.</td></tr>
</table>
Notes:
−Software reset (SCR:UPCL = "1") resets this bit to "0".
−If reception block transfer enable (ECR:RXBLKEN) is "0", this bit is set to "0".
−If reception FIFO enable is "0"(FE2 = "0" for FSEL = "0", FE1 = "0" for FSEL = "1"), this bit is set to "0".
−If the reception FIFO is reset (FCL2 = "1" FSEL = "0", FCL1 = "1" for FSEL = "1"), this bit is set to "0".

(Correct)
■Please refer to Table 2-2 for output of interrupt request.
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Indicate that a transmission block transfer error is not occurring.</td></tr>
<tr><td>1</td><td>Indicate that a transmission block transfer error is occurring.</td></tr>
</table>
Notes:
−Software reset (SCR:UPCL = 1) resets this bit to "0".
−If reception block transfer enable (ECR:RXBLKEN) is "0", this bit is set to "0".
−If reception FIFO enable is "0"(FE2 = 0 for FSEL = 0, FE1 = 0 for FSEL = 1), this bit is set to "0".
−If the reception FIFO is reset (FCL2 = 1 FSEL = 0, FCL1 = 1 for FSEL = 1), this bit is set to "0".
</td></tr>
<tr><td>CHAPTER37: CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface) 9.20.Transmission Block Size Register (TBSIZE)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
TBSIZE bit
<table>
<tr><td>Write</td><td>Sets the number of transmission blocks.</td></tr>
<tr><td>Read</td><td>Read the setting value.</td></tr>
</table>
(Correct)
:[bit7:0] TBSIZE: Transmission Block Size byte
<table>
<tr><th>TBSIZE</th><th>Description</th></tr>
<tr><td>Write</td><td>Sets the number of transmission blocks.</td></tr>
<tr><td>Read</td><td>Read the setting value.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface) 1.Overview of the I2C Interface (I2C Communication Control Interface)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
<table><tr><th></th><th></th><th>Function</th></tr><tr><td>1</td><td>Data buffer</td><td>− Full duplex double buffering (when no FIFO is used)
− Transmission and reception FIFOs (64 bytes each) (when the FIFOs are used)</td></tr></table>
(Correct)
<table><tr><th></th><th>Function</th><th>Description</th></tr><tr><td>1</td><td>Data buffer</td><td>− Full duplex double buffering (when no FIFO is used)
− Transmission and reception FIFOs (64 bytes each) (when the FIFOs are used)</td></tr></table></td></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface) 3.I2C Interface Operation 3.2.Slave Mode</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■Reserved address detection
If the 1st byte matches the reserved address ("0000xxxx" or "1111xxxx"), the IBCR:INT bit is set to "1" after the 8th bit of data is received, regardless of whether the transmission and reception FIFOs are enabled, to wait for the I2C bus. After reading the reception data at this time, set the following.

(Correct)
■Reserved address detection
If the 1st byte matches the reserved address (0b0000xxxx or 0b1111xxxx), the IBCR:INT bit is set to "1" after the 8th bit of data is received, regardless of whether the transmission and reception FIFOs are enabled, to wait for the I2C bus. After reading the reception data at this time, set the following.</td></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface) 8.I2C Interface Registers 8.1.I2C Bus Control Register (IBCR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
*1 ACK response: Indicates that the SDA of the I2C bus is "L" during the acknowledgment interval.
<table><tr><th>MSS</th><th>Master/Slave Selection Bit</th></tr><tr><td>0</td><td>Select slave mode.</td></tr><tr><td>1</td><td>Select master mode.</td></tr></table>
(Correct)
*1 ACK response: Indicates that the SDA of the I2C bus is "L" during the acknowledgment interval.
<table><tr><th>MSS Bit</th><th>Description</th></tr><tr><td>0</td><td>Select slave mode.</td></tr><tr><td>1</td><td>Select master mode.</td></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface) 8.I2C Interface Registers 8.1.I2C Bus Control Register (IBCR)</td><td>
The item should be added as indicated by the shading below.

(Error)
[bit14] ACT/SCC: Operation flag/Repeated start condition generation bit
The meaning of this bit varies depending on whether the operation is read or write.
<table>
<tr><td>Write</td><td>SCC bit</td></tr>
<tr><td>Read</td><td>ACT bit</td></tr>
</table>

(Correct)
[bit14] ACT/SCC: Operation flag/Repeated Start Condition Generation Bit
The meaning of this bit varies depending on whether the operation is read or write.
<table>
<tr><th>ACT/SCC</th><th>Description</th></tr>
<tr><td>Write</td><td>SCC bit</td></tr>
<tr><td>Read</td><td>ACT bit</td></tr>
</table>
</td></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface) 8.I2C Interface Registers 8.1.I2C Bus Control Register (IBCR)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
Writing "1" to this bit in master mode executes a repeated start. Writing "0" is invalid.
<table>
<tr><th rowspan="2">ACT/SCC</th><th colspan="2">Operation Flag/Repeated Start Condition Generation Bit</th></tr>
<tr><th>Write</th><th>Read</th></tr>
<tr><td>0</td><td>No effect</td><td>No operation</td></tr>
<tr><td>1</td><td>Generate the repeated start condition.</td><td>The I²C is operating.</td></tr>
</table>

(Correct)
Writing "1" to this bit in master mode executes a repeated start. Writing "0" is invalid.
<table>
<tr><th rowspan="2">Bit</th><th colspan="2">Description</th></tr>
<tr><th>Write</th><th>Read</th></tr>
<tr><td>0</td><td>No effect</td><td>No operation</td></tr>
<tr><td>1</td><td>Generate the repeated start condition.</td><td>The I²C is operating.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface) 8.I2C Interface Registers 8.1.I2C Bus Control Register (IBCR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)<table><tr><th>ACKE</th><th>Acknowledgment Enable Bit</th></tr><tr><td>0</td><td>Disable acknowledgment.</td></tr><tr><td>1</td><td>Enable acknowledgment.</td></tr></table>
(Correct)<table><tr><th>Bit</th><th>Description</th></tr><tr><td>0</td><td>Disable acknowledgment.</td></tr><tr><td>1</td><td>Enable acknowledgment.</td></tr></table></td></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface) 8.I2C Interface Registers 8.1.I2C Bus Control Register (IBCR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)<table><tr><th>WSEL</th><th>Wait Selection Bit</th></tr><tr><td>0</td><td>Wait after acknowledgment (9th bit).</td></tr><tr><td>1</td><td>Wait after data transmission is completed (8th bit).</td></tr></table>
(Correct)<table><tr><th>Bit</th><th>Description</th></tr><tr><td>0</td><td>Wait after acknowledgment (9th bit).</td></tr><tr><td>1</td><td>Wait after data transmission is completed (8th bit).</td></tr></table></td></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface) 8.I2C Interface Registers 8.1.I2C Bus Control Register (IBCR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)<table><tr><th>CNDE</th><th>Condition Detection Interrupt Enable Bit</th></tr><tr><td>0</td><td>Disable repeated start or stop condition interrupts.</td></tr><tr><td>1</td><td>Enable repeated start or stop condition interrupts.</td></tr></table>
(Correct)<table><tr><th>Bit</th><th>Description</th></tr><tr><td>0</td><td>Disable repeated start or stop condition interrupts.</td></tr><tr><td>1</td><td>Enable repeated start or stop condition interrupts.</td></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface)
8.I2C Interface Registers
8.1.I2C Bus Control Register (IBCR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
<table><tr><th>INTE</th><th>Interrupt Enable Bit</th></tr><tr><td>0</td><td>Disable interrupts.</td></tr><tr><td>1</td><td>Enable interrupts.</td></tr></table>
(Correct)
<table><tr><th>Bit</th><th>Description</th></tr><tr><td>0</td><td>Disable interrupts.</td></tr><tr><td>1</td><td>Enable interrupts.</td></tr></table></td></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface)
8.I2C Interface Registers
8.1.I2C Bus Control Register (IBCR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
<table><tr><th>BER</th><th>Bus Error Detection Bit</th></tr><tr><td>0</td><td>There is no error.</td></tr><tr><td>1</td><td>An error is detected.</td></tr></table>
(Correct)
<table><tr><th>Bit</th><th>Description</th></tr><tr><td>0</td><td>There is no error.</td></tr><tr><td>1</td><td>An error is detected.</td></tr></table></td></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface)
8.I2C Interface Registers
8.1.I2C Bus Control Register (IBCR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
Writing "1" to the INT bit is invalid when DMA mode is disabled (SSR:DMA="0").
<table><tr><th rowspan="2">INT</th><th colspan="2">Interrupt Flag Bit</th></tr><tr><th>Write</th><th>Read</th></tr><tr><td>0</td><td>No effect</td><td>No interrupt request</td></tr><tr><td>1</td><td>No effect</td><td>Interrupt request</td></tr></table>
(Correct)
Writing "1" to the INT bit is invalid when DMA mode is disabled (SSR:DMA= 0).
<table><tr><th rowspan="2">Bit</th><th colspan="2">Description</th></tr><tr><th>Write</th><th>Read</th></tr><tr><td>0</td><td>No effect</td><td>No interrupt request</td></tr><tr><td>1</td><td>No effect</td><td>Interrupt request</td></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface) 8.I2C Interface Registers 8.2.Serial Mode Register (SMR)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit7:5] MD2, MD1, MD0 : Operation mode setting bits
These bits set the operation mode.
"0b000": Set operation mode 0 (asynchronous normal mode).
"0b001": Set operation mode 1 (asynchronous multi-processor mode).
"0b010": Set operation mode 2 (clock synchronous mode).
"0b011": Set operation mode 3 (LIN communication mode).
"0b100": Set operation mode 4 (I2C mode).
This section explains the register and its operations in operation mode 4 (I2C mode).
<table>
<tr><th>MD2</th><th>MD1</th><th>MD0</th><th>Operation Mode Setting Bit</th></tr>
</table>

(Correct)
[bit7:5] MD[2:0] : Operation Mode Setting Bits
These bits set the operation mode.
"0b000": Set operation mode 0 (asynchronous normal mode).
"0b001": Set operation mode 1 (asynchronous multi-processor mode).
"0b010": Set operation mode 2 (clock synchronous mode).
"0b011": Set operation mode 3 (LIN communication mode).
"0b100": Set operation mode 4 (I2C mode).
This section explains the register and its operations in operation mode 4 (I2C mode).
<table>
<tr><th>Bits</th><th>Description</th></tr>
</table>
</td></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface) 8.I2C Interface Registers 8.2.Serial Mode Register (SMR)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
"1": Set the SDA or SCL pin for external interrupts.
<table>
<tr><th>WUCR</th><th>WAKE UP Control Bit</th></tr>
</table>

(Correct)
"1": Set the SDA or SCL pin for external interrupts.
<table>
<tr><th>Bit</th><th>Description</th></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface) 8.I2C Interface Registers 8.2.Serial Mode Register (SMR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■This bit is set when the SMRS:RIES bit in the set register is set to "1".
<table>
<tr><th>RIE</th><th>Reception Interrupt Enable Bit</th></tr>
<tr><td>0</td><td>Disable reception interrupts.</td></tr>
<tr><td>1</td><td>Enable reception interrupts.</td></tr>
</table>
(Correct)
■This bit is set when the SMRS:RIES bit in the set register is set to "1".
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Disable reception interrupts.</td></tr>
<tr><td>1</td><td>Enable reception interrupts.</td></tr>
</table></td></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface) 8.I2C Interface Registers 8.2.Serial Mode Register (SMR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■This bit is set when the SMRS:RIES bit in the set register is set to "1".
<table>
<tr><th>TIE</th><th>Transmission Interrupt Enable Bit</th></tr>
<tr><td>0</td><td>Disable transmission interrupts.</td></tr>
<tr><td>1</td><td>Enable transmission interrupts.</td></tr>
</table>
(Correct)
■This bit is set when the SMRS:RIES bit in the set register is set to "1".
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Disable transmission interrupts.</td></tr>
<tr><td>1</td><td>Enable transmission interrupts.</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface)
8.I2C Interface Registers
8.3.I2C Bus Status Register (IBSR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
<table><tr><th>FBT</th><th>First Byte Bit</th></tr><tr><td>0</td><td>Not the 1st byte</td></tr><tr><td>1</td><td>The 1st byte is being transmitted and received.</td></tr></table>
(Correct)
<table><tr><th>Bit</th><th>Description</th></tr><tr><td>0</td><td>Not the 1st byte</td></tr><tr><td>1</td><td>The 1st byte is being transmitted and received.</td></tr></table></td></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface)
8.I2C Interface Registers
8.3.I2C Bus Status Register (IBSR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
<table><tr><th>RACK</th><th>Acknowledgment Flag Bit</th></tr><tr><td>0</td><td>"L" reception</td></tr><tr><td>1</td><td>"H" reception</td></tr></table>
(Correct)
<table><tr><th>Bit</th><th>Description</th></tr><tr><td>0</td><td>"L" reception</td></tr><tr><td>1</td><td>"H" reception</td></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface)
8.I2C Interface Registers
8.3.I2C Bus Status Register (IBSR)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
If the RSA bit is set to "1" in the 1st byte, the interrupt flag (IBCR:INT) is set to "1" and the SCL to "L" at the SCL falling of the 8th bit in the 1st byte. This occurs regardless of whether FIFO is enabled or disabled. To read reception data and have the device operate as a slave at this time, set IBCR:ACKE to "1", and clear the interrupt flag (IBCR:INT) to "0". After that, if the TRX bit is "0", the device receives data as a slave. To prevent the device from receiving data in midstream, set the IBCR:ACKE bit to "0". Then, the device does not receive subsequent data.
<table>
<tr><th>RSA</th><th>Reserved Address Detection Bit</th></tr>
<tr><td>0</td><td>Reserved address not detected</td></tr>
<tr><td>1</td><td>Reserved address detected</td></tr>
</table>

(Correct)
If the RSA bit is set to "1" in the 1st byte, the interrupt flag (IBCR:INT) is set to "1" and the SCL to "L" at the SCL falling of the 8th bit in the 1st byte. This occurs regardless of whether FIFO is enabled or disabled. To read reception data and have the device operate as a slave at this time, set IBCR:ACKE to "1", and clear the interrupt flag (IBCR:INT) to "0". After that, if the TRX bit is "0", the device receives data as a slave. To prevent the device from receiving data in midstream, set the IBCR:ACKE bit to "0". Then, the device does not receive subsequent data.
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Reserved address not detected</td></tr>
<tr><td>1</td><td>Reserved address detected</td></tr>
</table>
</td></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface)
8.I2C Interface Registers
8.3.I2C Bus Status Register (IBSR)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
<table>
<tr><th>TRX</th><th>Data Direction Bit</th></tr>
<tr><td>0</td><td>Reception direction</td></tr>
<tr><td>1</td><td>Transmission direction</td></tr>
</table>

(Correct)
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Reception direction</td></tr>
<tr><td>1</td><td>Transmission direction</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface) 8.I2C Interface Registers 8.3.I2C Bus Status Register (IBSR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)<table><tr><th>AL</th><th>Arbitration Lost Bit</th></tr><tr><td>0</td><td>No arbitration lost occurred.</td></tr><tr><td>1</td><td>Arbitration lost occurred.</td></tr></table>
(Correct)<table><tr><th>Bit</th><th>Description</th></tr><tr><td>0</td><td>No arbitration lost occurred.</td></tr><tr><td>1</td><td>Arbitration lost occurred.</td></tr></table></td></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface) 8.I2C Interface Registers 8.3.I2C Bus Status Register (IBSR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)<table><tr><th>RSC</th><th>Repeated Start Condition Check Bit</th></tr><tr><td>0</td><td>The repeated start condition was not detected.</td></tr><tr><td>1</td><td>The repeated start condition was detected.</td></tr></table>
(Correct)<table><tr><th>Bit</th><th>Description</th></tr><tr><td>0</td><td>The repeated start condition was not detected.</td></tr><tr><td>1</td><td>The repeated start condition was detected.</td></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface) 8.I2C Interface Registers 8.3.I2C Bus Status Register (IBSR)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
Writing to this bit is invalid.
<table>
<tr><th>RSC</th><th colspan="2">Repeated Start Condition Check Bit</th></tr>
<tr><td>0</td><td colspan="2">The repeated start condition was not detected.</td></tr>
<tr><td rowspan="2">1</td><td>Master</td><td>The stop condition was detected, or arbitration lost occurred when the stop condition was output.</td></tr>
<tr><td>Slave</td><td>The stop condition was detected.</td></tr>
</table>
(Correct)
Writing to this bit is invalid.
<table>
<tr><th>Bit</th><th colspan="2">Description</th></tr>
<tr><td>0</td><td colspan="2">The repeated start condition was not detected.</td></tr>
<tr><td rowspan="2">1</td><td>Master</td><td>The stop condition was detected, or arbitration lost occurred when the stop condition was output.</td></tr>
<tr><td>Slave</td><td>The stop condition was detected.</td></tr>
</table>
</td></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface) 8.I2C Interface Registers 8.3.I2C Bus Status Register (IBSR)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
<table>
<tr><th>BB</th><th>Bus Status Bit</th></tr>
<tr><td>0</td><td>Bus idle state</td></tr>
<tr><td>1</td><td>Bus transmission and reception state</td></tr>
</table>
(Correct)
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Bus idle state</td></tr>
<tr><td>1</td><td>Bus transmission and reception state</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface) 8.I2C Interface Registers 8.4.Serial Status Register (SSR)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
If it is read, "0" is always read.
<table>
<tr><th rowspan="2">REC</th><th colspan="2">Reception Error Flag Clear Bit</th></tr>
<tr><th>Write</th><th>Read</th></tr>
<tr><td>0</td><td>No effect</td><td rowspan="2">"0" is always read.</td></tr>
<tr><td>1</td><td>Clear the reception error flag (ORE).</td></tr>
</table>
(Correct)
If it is read, "0" is always read.
<table>
<tr><th rowspan="2">Bit</th><th colspan="2">Description</th></tr>
<tr><th>Write</th><th>Read</th></tr>
<tr><td>0</td><td>No effect</td><td rowspan="2">"0" is always read.</td></tr>
<tr><td>1</td><td>Clear the reception error flag (ORE).</td></tr>
</table>
</td></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface) 8.I2C Interface Registers 8.4.Serial Status Register (SSR)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
If it is read, "0" is always read.
<table>
<tr><th rowspan="2">TSET</th><th colspan="2">Transmission Data Empty Flag Set Bit</th></tr>
<tr><th>Write</th><th>Read</th></tr>
<tr><td>0</td><td>No effect</td><td rowspan="2">"0" is always read.</td></tr>
<tr><td>1</td><td>Set the TDRE bit.</td></tr>
</table>
(Correct)
If it is read, "0" is always read.
<table>
<tr><th rowspan="2">Bit</th><th colspan="2">Description</th></tr>
<tr><th>Write</th><th>Read</th></tr>
<tr><td>0</td><td>No effect</td><td rowspan="2">"0" is always read.</td></tr>
<tr><td>1</td><td>Set the TDRE bit.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface) 8.I2C Interface Registers 8.4.Serial Status Register (SSR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
For details, see Table 2-1.
<table><tr><th>DMA</th><th>DMA Mode Enable Bit</th></tr><tr><td>0</td><td>Disable DMA mode.</td></tr><tr><td>1</td><td>Enable DMA mode.</td></tr></table>
(Correct)
For details, see Table 2-1.
<table><tr><th>Bit</th><th>Description</th></tr><tr><td>0</td><td>Disable DMA mode.</td></tr><tr><td>1</td><td>Enable DMA mode.</td></tr></table></td></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface) 8.I2C Interface Registers 8.4.Serial Status Register (SSR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■This bit is set when the SSRS:TBIES bit in the set register is set to "1".
<table><tr><th>TBIE</th><th>Transmission Bus Idle Interrupt Enable Bit</th></tr></table>
(Correct)
■This bit is set when the SSRS:TBIES bit in the set register is set to "1".
<table><tr><th>Bit</th><th>Description</th></tr></table></td></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface) 8.I2C Interface Registers 8.4.Serial Status Register (SSR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■If this flag is set, no reception data is stored in the reception FIFO when used.
<table><tr><th>ORE</th><th>Overrun Error Flag Bit</th></tr></table>
(Correct)
■If this flag is set, no reception data is stored in the reception FIFO when used.
<table><tr><th>Bit</th><th>Description</th></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface) 8.I2C Interface Registers 8.4.Serial Status Register (SSR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
*1 NACK response: Indicates that the SDA of the I²C bus is "H" during an acknowledgment period.
<table>
<tr><th>RDRF</th><th>Reception Data Full Flag Bit</th></tr>
<tr><td>0</td><td>The reception data register (RDR) is empty.</td></tr>
<tr><td>1</td><td>The reception data register RDR contains data.</td></tr>
</table>
(Correct)
*1 NACK response: Indicates that the SDA of the I²C bus is "H" during an acknowledgment period.
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>The reception data register (RDR) is empty.</td></tr>
<tr><td>1</td><td>The reception data register RDR contains data.</td></tr>
</table></td></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface) 8.I2C Interface Registers 8.4.Serial Status Register (SSR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■This bit is set when “1” is written to the TSET bit in the serial status register (SSR). Use it when the desired setting for the TDRE bit is “1” at the detection time of arbitration lost, a bus error, etc.
<table>
<tr><th>TDRE</th><th>Transmission Data Empty Flag Bit</th></tr>
</table>
(Correct)
■This bit is set when "1" is written to the TSET bit in the serial status register (SSR). Use it when the desired setting for the TDRE bit is "1" at the detection time of arbitration lost, a bus error, etc.
<table>
<tr><th>Bit</th><th>Description</th></tr>
</table></td></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface) 8.I2C Interface Registers 8.4.Serial Status Register (SSR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■This bit is “1” when DMA mode is disabled (DMA=”0”).
<table>
<tr><th>TBI</th><th>Transmission Bus Idle Flag Bit</th></tr>
</table>
(Correct)
■This bit is "1" when DMA mode is disabled (DMA= 0).
<table>
<tr><th>Bit</th><th>Description</th></tr>
</table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface) 8.I2C Interface Registers 8.6.Serial Auxiliary Control Status Register (SACSR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit10:9] TRG1-0 : Trigger selection bits
These bits select the method of detecting the edge for an external trigger of serial timer activation.
<table><tr><th>TRG1</th><th>TRG0</th><th>External Trigger Edge Detection Method</th></tr></table>
(Correct)
[bit10:9] TRG[1:0] : Trigger Selection Bits
These bits select the method of detecting the edge for an external trigger of serial timer activation.
<table><tr><th>Bits</th><th>Description</th></tr></table></td></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface) 8.I2C Interface Registers 8.6.Serial Auxiliary Control Status Register (SACSR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■This bit is reset when the SACSRC:TINTC bit in the clear register is set to "1".
<table><tr><th>TINT</th><th>Description</th></tr></table>
(Correct)
■This bit is reset when the SACSRC:TINTC bit in the clear register is set to "1".
<table><tr><th>Bit</th><th>Description</th></tr></table></td></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface) 8.I2C Interface Registers 8.6.Serial Auxiliary Control Status Register (SACSR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■This bit is set when the SACSRS:TINTES bit in the set register is set to "1".
<table><tr><th>TINTE</th><th>Description</th></tr></table>
(Correct)
■This bit is set when the SACSRS:TINTES bit in the set register is set to "1".
<table><tr><th>Bit</th><th>Description</th></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface) 8.I2C Interface Registers 8.6.Serial Auxiliary Control Status Register (SACSR)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
■This bit is set when the SACSRS:TRGES bit in the set register is set to "1".
<table>
<tr><th>TRGE</th><th>Description</th></tr>
</table>
(Correct)
■This bit is set when the SACSRS:TRGES bit in the set register is set to "1".
<table>
<tr><th>Bit</th><th>Description</th></tr>
</table>
</td></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface) 8.I2C Interface Registers 8.6.Serial Auxiliary Control Status Register (SACSR)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit4-1] TDIV3-0 : Timer operation clock division bits
These bits set the division ratio of the serial timer.
<table>
<tr><th rowspan="2">TDIV3</th><th rowspan="2">TDIV2</th><th rowspan="2">TDIV1</th><th rowspan="2">TDIV0</th><th colspan="7">Timer Operation Clock</th></tr>
<tr><th>Division Ratio</th><th>Φ= 8 MHz</th><th>Φ= 10 MHz</th><th>Φ= 16 MHz</th><th>Φ= 20 MHz</th><th>Φ= 24 MHz</th><th>Φ= 32 MHz</th></tr>
</table>
(Correct)
[bit4-1] TDIV[3:0] : Timer Operation Clock Division Bits
These bits set the division ratio of the serial timer.
<table>
<tr><th rowspan="2">Bits</th><th colspan="7">Timer Operation Clock</th></tr>
<tr><th>Division Ratio</th><th>Φ= 8 MHz</th><th>Φ= 10 MHz</th><th>Φ= 16 MHz</th><th>Φ= 20 MHz</th><th>Φ= 24 MHz</th><th>Φ= 32 MHz</th></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface) 8.I2C Interface Registers 8.6.Serial Auxiliary Control Status Register (SACSR)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
■This bit is set when the SACSRS:TMRES bit in the set register is set to "1".
<table>
<tr><th>TMRE</th><th>Serial Timer Enable Bit</th></tr>
<tr><td>0</td><td>Stop the operation of the serial timer.
The value of the serial timer register (STMR), when stopped, is retained.</td></tr>
<tr><td>1</td><td>Initialize the serial timer register (STMR) to "0" and start the operation of the serial timer when this bit changes from "0" to "1".</td></tr>
</table>
(Correct)
■This bit is set when the SACSRS:TMRES bit in the set register is set to "1".
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Stop the operation of the serial timer.
The value of the serial timer register (STMR), when stopped, is retained.</td></tr>
<tr><td>1</td><td>Initialize the serial timer register (STMR) to "0" and start the operation of the serial timer when this bit changes from "0" to "1".</td></tr>
</table>
</td></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface) 8.I2C Interface Registers 8.9.Noise Filter Control Register (NFCR)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
Table 8-2 Relationship between the Noise Filter Time Selection Bits and Bus Clock Frequency
<table>
<tr><th>NFT4</th><th>NFT3</th><th>NFT2</th><th>NFT1</th><th>NFT0</th><th>Bus Clock Frequency [MHz]</th></tr>
</table>
(Correct)
Table 8-2 Relationship between the Noise Filter Time Selection Bits and Bus Clock Frequency
<table>
<tr><th>Bits</th><th>Bus Clock Frequency [MHz]</th></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface) 8.I2C Interface Registers 8.10.Extended I2C Bus Control Register (EIBCR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
This bit indicates the signal level of the SDA line after a signal passes through the noise filter.
<table>
<tr><th>SDAS</th><th>SDA Status Bit</th></tr>
<tr><td>0</td><td>The SDA line is "L".</td></tr>
<tr><td>1</td><td>The SDA line is "H".</td></tr>
</table>
(Correct)
This bit indicates the signal level of the SDA line after a signal passes through the noise filter.
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>The SDA line is "L".</td></tr>
<tr><td>1</td><td>The SDA line is "H".</td></tr>
</table></td></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface) 8.I2C Interface Registers 8.10.Extended I2C Bus Control Register (EIBCR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
This bit indicates the signal level of the SCL line after a signal passes through the noise filter.
<table>
<tr><th>SCLS</th><th>SCL Status Bit</th></tr>
<tr><td>0</td><td>The SCL line is "L".</td></tr>
<tr><td>1</td><td>The SCL line is "H".</td></tr>
</table>
(Correct)
This bit indicates the signal level of the SCL line after a signal passes through the noise filter.
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>The SCL line is "L".</td></tr>
<tr><td>1</td><td>The SCL line is "H".</td></tr>
</table></td></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface) 8.I2C Interface Registers 8.10.Extended I2C Bus Control Register (EIBCR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
This bit controls SDA output when serial output control is enabled (SOCE="1").
<table>
<tr><th>SDAC</th><th>SDA Output Control Bit</th></tr>
</table>
(Correct)
This bit controls SDA output when serial output control is enabled (SOCE= 1).
<table>
<tr><th>Bit</th><th>Description</th></tr>
</table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface) 8.I2C Interface Registers 8.10.Extended I2C Bus Control Register (EIBCR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
This bit controls SCL output when serial output control is enabled (SOCE="1").
<table><tr><th>SCLC</th><th>SCL Output Control Bit</th></tr></table>
(Correct)
This bit controls SCL output when serial output control is enabled (SOCE= 1).
<table><tr><th>Bit</th><th>Description</th></tr></table></td></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface) 8.I2C Interface Registers 8.10.Extended I2C Bus Control Register (EIBCR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■The SCL output control bit (SCLC) controls SCL output.
<table><tr><th>SOCE</th><th>Serial Output Control Enable Bit</th></tr><tr><td>0</td><td>Disable serial output control.</td></tr><tr><td>1</td><td>Enable serial output control.</td></tr></table>
(Correct)
■The SCL output control bit (SCLC) controls SCL output.
<table><tr><th>Bit</th><th>Description</th></tr><tr><td>0</td><td>Disable serial output control.</td></tr><tr><td>1</td><td>Enable serial output control.</td></tr></table></td></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface) 8.I2C Interface Registers 8.10.Extended I2C Bus Control Register (EIBCR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
This bit selects whether to continue or stop I^{2}C operation after a bus error occurs (IBSR:BER="1").
<table><tr><th>BEC</th><th>Bus Error Control Bit</th></tr><tr><td>0</td><td>Stop I^{2}C operation.</td></tr><tr><td>1</td><td>Continue I^{2}C operation.</td></tr></table>
(Correct)
This bit selects whether to continue or stop I^{2}C operation after a bus error occurs (IBSR:BER= 1).
<table><tr><th>Bit</th><th>Description</th></tr><tr><td>0</td><td>Stop I^{2}C operation.</td></tr><tr><td>1</td><td>Continue I^{2}C operation.</td></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface) 8.I2C Interface Registers 8.11.7-bit Slave Address Mask Register (ISMK)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
"1": Enable I²C operation.
<table>
<tr><th>EN</th><th>I²C-UART Operation Enable Bit</th></tr>
<tr><td>0</td><td>Disable</td></tr>
<tr><td>1</td><td>Enable</td></tr>
</table>
(Correct)
"1": Enable I²C operation.
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Disable</td></tr>
<tr><td>1</td><td>Enable</td></tr>
</table>
</td></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface) 8.I2C Interface Registers 8.11.7-bit Slave Address Mask Register (ISMK)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit14:8] SM6-0 : Slave address mask bits
These bits specify whether a 7-bit slave address and the received address are excluded from comparison.
Bits set to "1": Compare them.
Bits set to "0": Handle them as a match.
<table>
<tr><th>SM6-0</th><th>7-bit Slave Address Mask Bit</th></tr>
</table>
(Correct)
[bit14:8] SM[n] n=6 to 0 : Slave Address Mask Bits
These bits specify whether a 7-bit slave address and the received address are excluded from comparison.
Bits set to "1": Compare them.
Bits set to "0": Handle them as a match.
<table>
<tr><th>SM[n] n=6 to 0</th><th>Description</th></tr>
</table>
</td></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface) 8.I2C Interface Registers 8.12.7-bit Slave Address Register (ISBA)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
"1": Compare the ISBA and ISMK settings with the 1st byte received.
<table>
<tr><th>SAEN</th><th>Slave Address Enable Bit</th></tr>
</table>
(Correct)
"1": Compare the ISBA and ISMK settings with the 1st byte received.
<table>
<tr><th>Bit</th><th>Description</th></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface) 8.I2C Interface Registers 8.12.7-bit Slave Address Register (ISBA)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit6:0] SA6-0 : 7-bit slave address
■If slave address detection is enabled (SAEN="1"), the 7-bit slave address register (ISBA) compares the 7-bit data received after the detection of the (repeated) start condition with this register. If all bits match, the device operates in slave mode, and ACK is output. Then, the received slave address is set in this register. (If SAEN="0", ACK is not output.)
■The address bit for which "0" is set in ISMK is excluded from the comparison.
<table>
<tr><th>SA</th><th>Slave Address Setting Bit</th></tr>
<tr><td>6 to 0</td><td>7-bit slave address</td></tr>
</table>
(Correct)
[bit6:0] SA[6:0] : 7-Bit Slave Address
■If slave address detection is enabled (SAEN= 1), the 7-bit slave address register (ISBA) compares the 7-bit data received after the detection of the (repeated) start condition with this register. If all bits match, the device operates in slave mode, and ACK is output. Then, the received slave address is set in this register. (If SAEN= 0, ACK is not output.)
■The address bit for which "0" is set in ISMK is excluded from the comparison.
<table>
<tr><th>SA[6:0]</th><th>Description</th></tr>
<tr><td>Write / Read</td><td>7-bit slave address</td></tr>
</table>
</td></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface) 8.I2C Interface Registers 8.13.Baud Rate Generator Registers 1, 0 (BGR1, BGR0)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit14:8] BGR1 : Baud rate generator register 1
<table>
<tr><th>BGR1</th><th>Baud Rate Generator Register 1</th></tr>
</table>
(Correct)
[bit14:8] BGR1 : Baud Rate Generator Register 1
<table>
<tr><th>BGR1</th><th>Description</th></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface) 8.I2C Interface Registers 8.13.Baud Rate Generator Registers 1, 0 (BGR1, BGR0)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit7:0] BGR0 : Baud rate generator register 0
<table><tr><th>BGR0</th><th>Baud Rate Generator Register 0</th></tr></table>
(Correct)
[bit7:0] BGR0 : Baud Rate Generator Register 0
<table><tr><th>BGR0</th><th>Description</th></tr></table></td></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface) 8.I2C Interface Registers 8.14.FIFO Control Register 1 (FCR1)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
"1": Enable FCR0:FLST bit detection.
<table><tr><th>FLSTE</th><th>Retransmission Data Lost Detection Enable Bit</th></tr></table>
(Correct)
"1": Enable FCR0:FLST bit detection.
<table><tr><th>Bit</th><th>Description</th></tr></table></td></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface) 8.I2C Interface Registers 8.14.FIFO Control Register 1 (FCR1)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
"0": Disable detection of the reception idle state.
"1": Enable detection of the reception idle state.
<table><tr><th>FRIIE</th><th>Reception FIFO Idle Detection Enable Bit</th></tr></table>
(Correct)
"0": Disable detection of the reception idle state.
"1": Enable detection of the reception idle state.
<table><tr><th>Bit</th><th>Description</th></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface) 8.I2C Interface Registers 8.14.FIFO Control Register 1 (FCR1)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■This bit is set when the FCR1S:FTIES bit in the set register is set to "1".
<table><tr><th>FTIE</th><th>Transmission FIFO Interrupt Enable Bit</th></tr></table>
(Correct)
■This bit is set when the FCR1S:FTIES bit in the set register is set to "1".
<table><tr><th>Bit</th><th>Description</th></tr></table></td></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface) 8.I2C Interface Registers 8.14.FIFO Control Register 1 (FCR1)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
"1": Allocate FIFO2 as the transmission FIFO and FIFO1 as the reception FIFO.
<table><tr><th>FSEL</th><th>FIFO Selection Bit</th></tr><tr><td>0</td><td>Transmission FIFO: FIFO1; reception FIFO: FIFO2</td></tr><tr><td>1</td><td>Transmission FIFO: FIFO2; reception FIFO: FIFO1</td></tr></table>
(Correct)
"1": Allocate FIFO2 as the transmission FIFO and FIFO1 as the reception FIFO.
<table><tr><th>Bit</th><th>Description</th></tr><tr><td>0</td><td>Transmission FIFO: FIFO1; reception FIFO: FIFO2</td></tr><tr><td>1</td><td>Transmission FIFO: FIFO2; reception FIFO: FIFO1</td></tr></table></td></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface) 8.I2C Interface Registers 8.15.FIFO Control Register 0 (FCR0)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
"1", perform a FIFO reset, and write data to the FIFO again.
<table><tr><th>FLST</th><th>FIFO Retransmission Data Lost Flag Bit</th></tr></table>
(Correct)
"1", perform a FIFO reset, and write data to the FIFO again.
<table><tr><th>Bit</th><th>Description</th></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface) 8.I2C Interface Registers 8.15.FIFO Control Register 0 (FCR0)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■This bit is set when the FCR0S:FLDS bit in the set register is set to "1".
<table><tr><th>FLD</th><th>FIFO Pointer Reload Bit</th></tr></table>
(Correct)
■This bit is set when the FCR0S:FLDS bit in the set register is set to "1".
<table><tr><th>Bit</th><th>Description</th></tr></table></td></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface) 8.I2C Interface Registers 8.15.FIFO Control Register 0 (FCR0)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
"0": No effect
<table><tr><th>FSET</th><th>FIFO Pointer Saving Bit</th></tr></table>
(Correct)
"0": No effect
<table><tr><th>Bit</th><th>Description</th></tr></table></td></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface) 8.I2C Interface Registers 8.15.FIFO Control Register 0 (FCR0)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■This bit is set when the FCR0S:FCL2S bit in the set register is set to "1".
<table><tr><th rowspan="2">FCL2</th><th colspan="2">FIFO2 Reset Bit</th></tr><tr><th>Write</th><th>Read</th></tr></table>
(Correct)
■This bit is set when the FCR0S:FCL2S bit in the set register is set to "1".
<table><tr><th rowspan="2">Bit</th><th colspan="2">Description</th></tr><tr><th>Write</th><th>Read</th></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface) 8.I2C Interface Registers 8.15.FIFO Control Register 0 (FCR0)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■This bit is set when the FCR0S:FCL2S bit in the set register is set to "1".
<table>
<tr><th rowspan="2">FCL1</th><th colspan="2">FIFO1 Reset Bit</th></tr>
<tr><th>Write</th><th>Read</th></tr>
</table>
(Correct)
■This bit is set when the FCR0S:FCL2S bit in the set register is set to "1".
<table>
<tr><th rowspan="2">Bit</th><th colspan="2">Description</th></tr>
<tr><th>Write</th><th>Read</th></tr>
</table></td></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface) 8.I2C Interface Registers 8.15.FIFO Control Register 0 (FCR0)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■This bit is set when the FCR0S:FE2S bit in the set register is set to "1".
<table>
<tr><th>FE2</th><th>FIFO2 Operation Enable Bit</th></tr>
<tr><td>0</td><td>Disable FIFO2 operation.</td></tr>
<tr><td>1</td><td>Enable FIFO2 operation.</td></tr>
</table>
(Correct)
■This bit is set when the FCR0S:FE2S bit in the set register is set to "1".
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Disable FIFO2 operation.</td></tr>
<tr><td>1</td><td>Enable FIFO2 operation.</td></tr>
</table></td></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface) 8.I2C Interface Registers 8.15.FIFO Control Register 0 (FCR0)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■This bit is set when the FCR0S:FE1S bit in the set register is set to "1".
<table>
<tr><th>FE1</th><th>FIFO1 Operation Enable Bit</th></tr>
<tr><td>0</td><td>Disable FIFO2 operation.</td></tr>
<tr><td>1</td><td>Enable FIFO2 operation.</td></tr>
</table>
(Correct)
■This bit is set when the FCR0S:FE1S bit in the set register is set to "1".
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Disable FIFO1 operation.</td></tr>
<tr><td>1</td><td>Enable FIFO1 operation.</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface) 8.I2C Interface Registers 8.16.FIFO Byte Register (FBYTE)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
FBYTE2, FBYTE1: FIFO2 data count indication bit, FIFO1 data count indication bit

(Correct)
[bit15:8] FBYTE2: FIFO2 Data Count Indication byte
[bit7:0] FBYTE1: FIFO1 Data Count Indication Byte</td></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface) 8.I2C Interface Registers 8.16.FIFO Byte Register (FBYTE)</td><td>The item should be added as indicated by the shading below.

(Error)<table><tr><th>Write</th><th>Set the transfer count.</th></tr><tr><td>Read</td><td>Read the valid data count.</td></tr></table>
(Correct)<table><tr><th>FBYTE2, FBYTE1</th><th>Description</th></tr><tr><td>Write</td><td>Set the transfer count.</td></tr><tr><td>Read</td><td>Read the valid data count.</td></tr></table></td></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface) 8.I2C Interface Registers 8.17.Transmission FIFO Interrupt Control Register (FTICR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
FTICR2, FTICR1: FIFO2 data count indication bit, FIFO1 data count indication bit

(Correct)
[bit15:8] FTICR2: FIFO2 Data Count Indication byte
[bit7:0] FTICR1: FIFO1 Data Count Indication Byte</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface) 8.I2C Interface Registers 8.17.Transmission FIFO Interrupt Control Register (FTICR)</td><td>The item should be added as indicated by the shading below.

(Error)<table><tr><td>Write</td><td>Set the valid data count that generates an interrupt.</td></tr><tr><td>Read</td><td>Read the valid data count.</td></tr></table>
(Correct)<table><tr><th>FTICR2, FTICR1</th><th>Description</th></tr><tr><td>Write</td><td>Set the valid data count that generates an interrupt.</td></tr><tr><td>Read</td><td>Read the valid data count.</td></tr></table></td></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface) 8.I2C Interface Registers 8.18.Extended Control Register (ECR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
This error interrupt output bit switches between output from an interrupt request pin and output from an error interrupt request pin. For details on the interrupt output selection, see Table 2-2 and Table 2-3.<table><tr><th>EISEL</th><th>Error Interrupt Request Output Selection Bit</th></tr></table>
(Correct)
This error interrupt output bit switches between output from an interrupt request pin and output from an error interrupt request pin. For details on the interrupt output selection, see Table 2-2 and Table 2-3.<table><tr><th>Bit</th><th>Description</th></tr></table></td></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface) 8.I2C Interface Registers 8.18.Extended Control Register (ECR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
This bit enables/disables reception error interrupt request output. For details on the target interrupt factors, see Table 2-3.<table><tr><th>REIE</th><th>Reception Error Interrupt Enable Bit</th></tr></table>
(Correct)
This bit enables/disables reception error interrupt request output. For details on the target interrupt factors, see Table 2-3.<table><tr><th>Bit</th><th>Description</th></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface) 8.I2C Interface Registers 8.18.Extended Control Register (ECR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
This bit enables/disables transmission error interrupt request output. For details on the target interrupt factors, see Table 2-2.
<table><tr><td>TEIE</td><td>Transmission Error Interrupt Enable Bit</td></tr></table>
(Correct)
This bit enables/disables transmission error interrupt request output. For details on the target interrupt factors, see Table 2-2.
<table><tr><td>Bit</td><td>Description</td></tr></table></td></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface) 8.I2C Interface Registers 8.18.Extended Control Register (ECR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■"1": Perform DMA transfer in block transfer mode.
<table><tr><td>RXBLKEN</td><td>Reception Block Transfer Setting Bit</td></tr></table>
(Correct)
■"1": Perform DMA transfer in block transfer mode.
<table><tr><td>Bit</td><td>Description</td></tr></table></td></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface) 8.I2C Interface Registers 8.18.Extended Control Register (ECR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■"1": Perform DMA transfer in block transfer mode.
<table><tr><td>TXBLKEN</td><td>Transmission Block Transfer Setting Bit</td></tr></table>
(Correct)
■"1": Perform DMA transfer in block transfer mode.
<table><tr><td>Bit</td><td>Description</td></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface) 8.I2C Interface Registers 8.19.Extended Status Register (ESR).Extended Control Register (ECR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■For details on interrupt request output, see Table 2-3.
<table>
<tr><th>RXUDR</th><th>Reception FIFO Under run Flag Bit</th></tr>
<tr><td>0</td><td>No reception FIFO under run has occurred.</td></tr>
<tr><td>1</td><td>A reception FIFO under run has occurred.</td></tr>
</table>
(Correct)
■For details on interrupt request output, see Table 2-3.
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>No reception FIFO under run has occurred.</td></tr>
<tr><td>1</td><td>A reception FIFO under run has occurred.</td></tr>
</table></td></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface) 8.I2C Interface Registers 8.19.Extended Status Register (ESR).Extended Control Register (ECR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■ For details on interrupt request output, see Table 2-2.
<table>
<tr><th>TXOVR</th><th>Transmission FIFO Overrun Flag Bit</th></tr>
<tr><td>0</td><td>No transmission FIFO overrun has occurred.</td></tr>
<tr><td>1</td><td>A transmission FIFO overrun has occurred.</td></tr>
</table>
(Correct)
■ For details on interrupt request output, see Table 2-2.
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>No transmission FIFO overrun has occurred.</td></tr>
<tr><td>1</td><td>A transmission FIFO overrun has occurred.</td></tr>
</table></td></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface) 8.I2C Interface Registers 8.19.Extended Status Register (ESR).Extended Control Register (ECR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■For details on interrupt request output, see Table 2-3.
<table>
<tr><th>RBERR</th><th>Reception Block Transfer Error Bit</th></tr>
<tr><td>0</td><td>No reception block transfer error has occurred.</td></tr>
<tr><td>1</td><td>Reception block transfer error has occurred.</td></tr>
</table>
(Correct)
■For details on interrupt request output, see Table 2-3.
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>No reception block transfer error has occurred.</td></tr>
<tr><td>1</td><td>Reception block transfer error has occurred.</td></tr>
</table></td></tr>
</table>

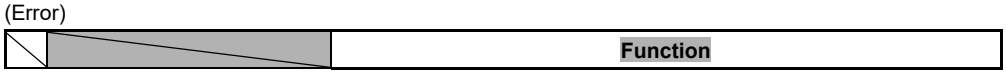
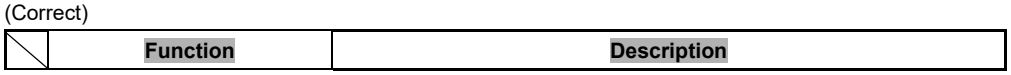
<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface) 8.I2C Interface Registers 8.19.Extended Status Register (ESR).Extended Control Register (ECR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

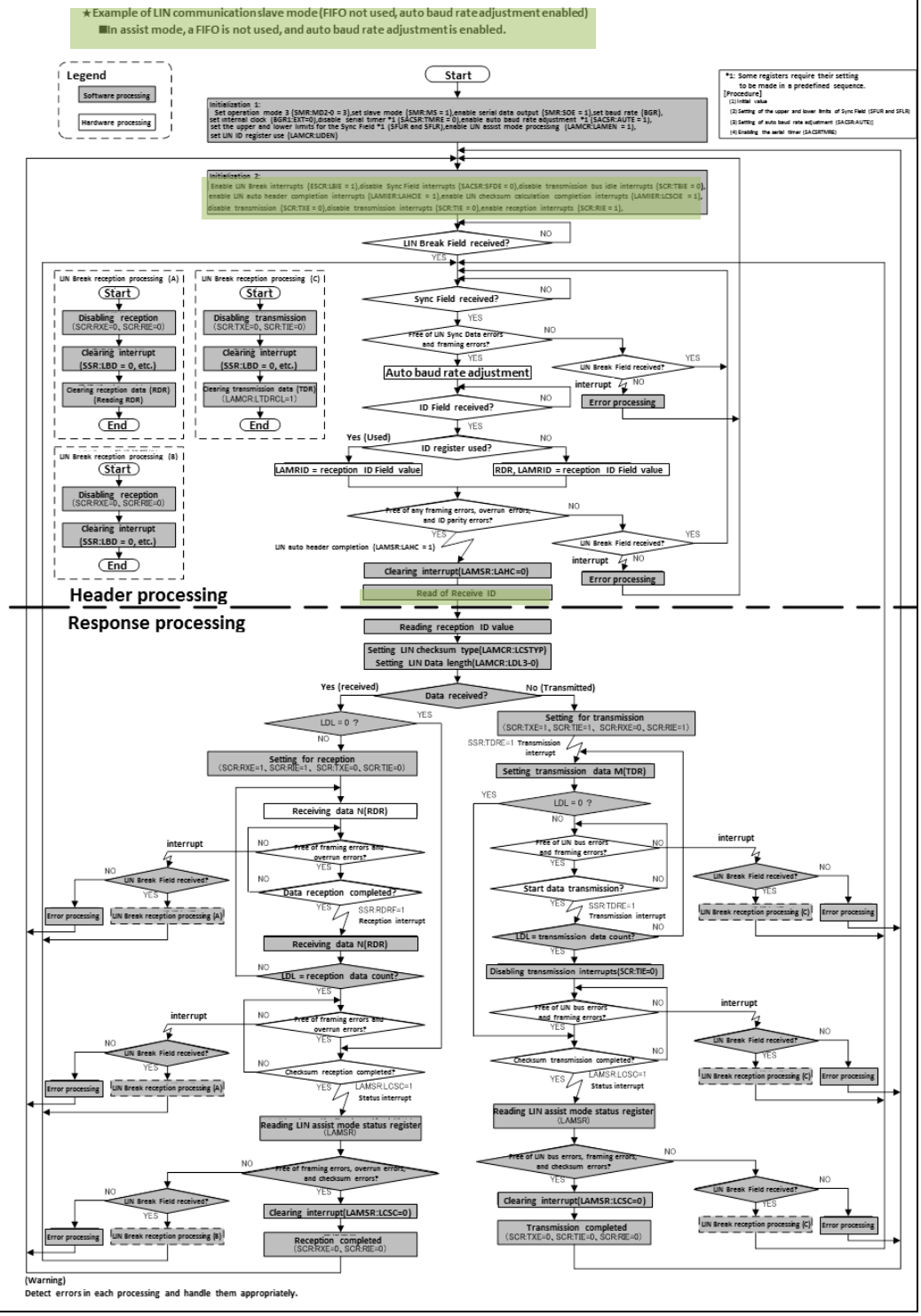
(Error)
■For details on interrupt request output, see Table 2-2.
<table>
<tr><th>TBERR</th><th>Transmission Block Transfer Error Bit</th></tr>
<tr><td>0</td><td>No transmission block transfer error has occurred.</td></tr>
<tr><td>1</td><td>A transmission block transfer error has occurred.</td></tr>
</table>
(Correct)
■For details on interrupt request output, see Table 2-2.
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>No transmission block transfer error has occurred.</td></tr>
<tr><td>1</td><td>A transmission block transfer error has occurred.</td></tr>
</table></td></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface) 8.I2C Interface Registers 8.20.Transmission Block Size Register (TBSIZE)</td><td>Features of "For details" should be corrected as indicated by the shading below.

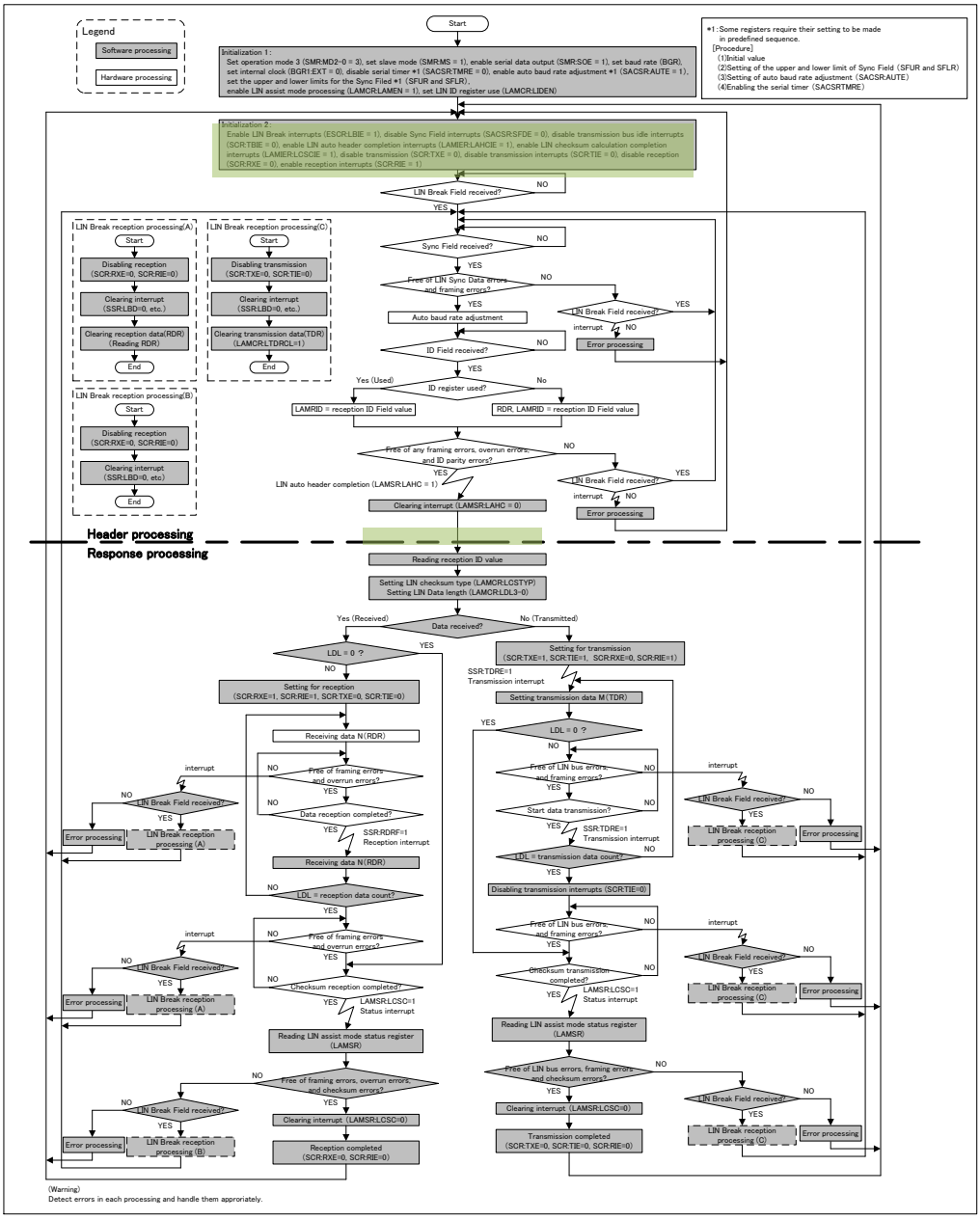
(Error)
TBSIZE bit

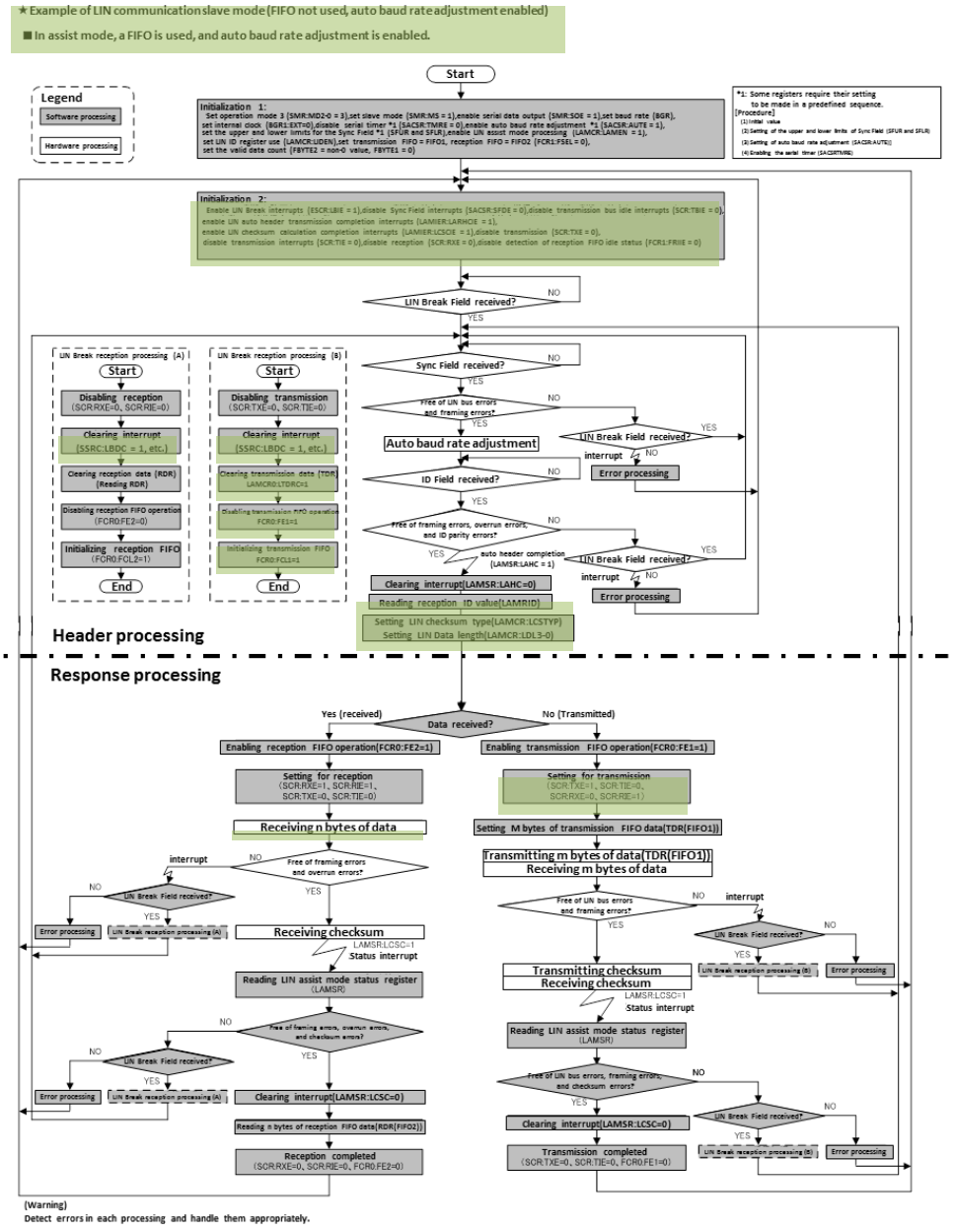
(Correct)
[bit7:0] TBSIZE : Transmission Block size byte</td></tr>
<tr><td>CHAPTER38: I2C Interface (I2C Communication Control Interface) 8.I2C Interface Registers 8.20.Transmission Block Size Register (TBSIZE)</td><td>The item should be added as indicated by the shading below.

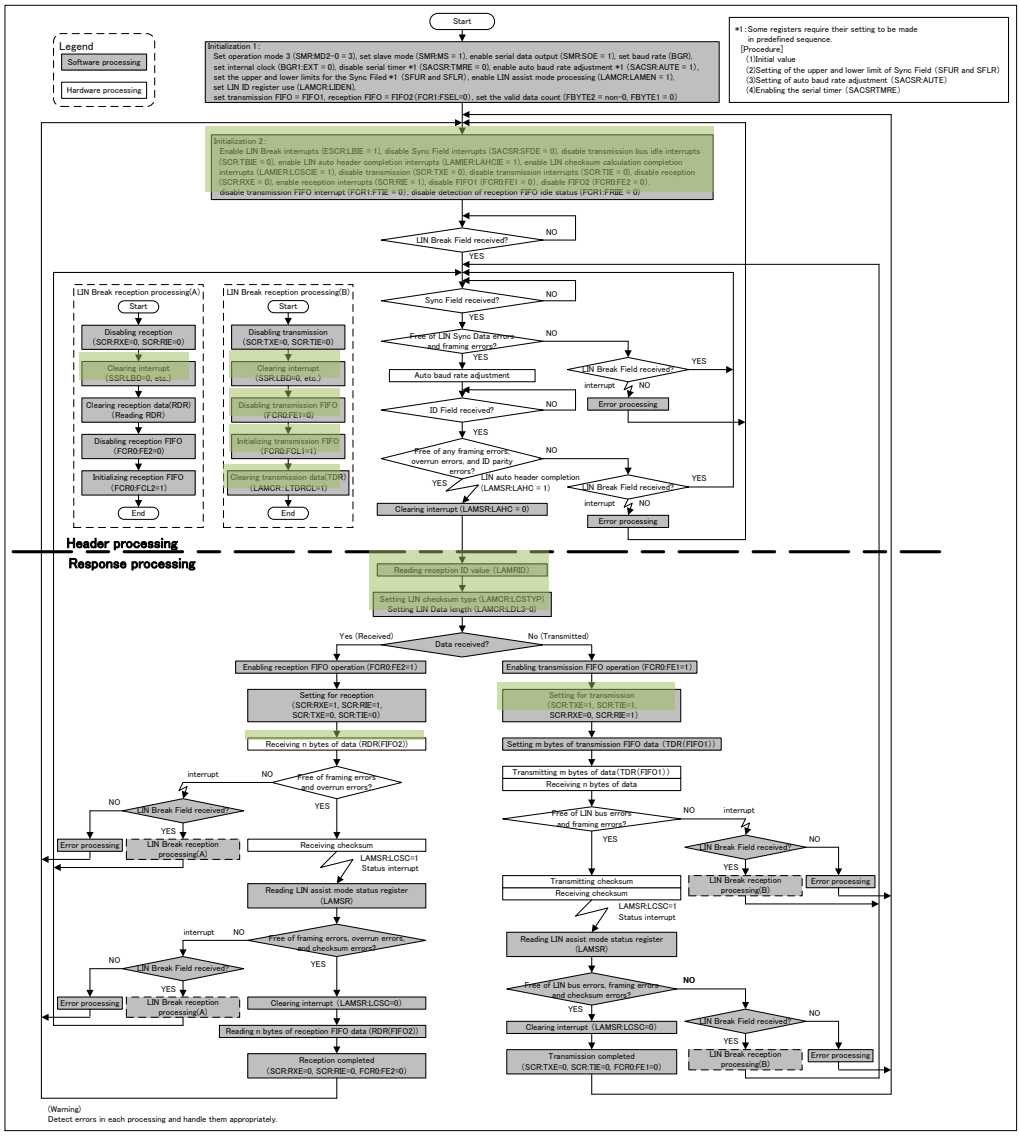
(Error)
<table>
<tr><td>Write</td><td>Set the number of transmission blocks.</td></tr>
<tr><td>Read</td><td>Read the setting value.</td></tr>
</table>
(Correct)
<table>
<tr><th>TBSIZE</th><th>Description</th></tr>
<tr><td>Write</td><td>Set the number of transmission blocks.</td></tr>
<tr><td>Read</td><td>Read the setting value.</td></tr>
</table></td></tr>
</table>

Section	Change Results
CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 1.LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 1.1.Manual Mode	Features of "For details" should be corrected as indicated by the shading below. (Error)  (Correct) 
CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 1.LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 1.2.Assist Mode	Features of "For details" should be corrected as indicated by the shading below. (Error)  (Correct) 
CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 4.Test Mode 4.2.Assist Mode	Features of "For details" should be corrected as indicated by the shading below. (Error) ■KEY1, KEY0 = "00" + write a pseudo error setting value ■KEY1, KEY0 = "01" + write the pseudo error setting value (same as the previous value) ■KEY1, KEY0 = "10" + write the pseudo error setting value (same as the previous value) ■KEY1, KEY0 = "11" + write the pseudo error setting value (same as the previous value) (Correct) ■KEY1, KEY0 = 0b00 + write a pseudo error setting value ■KEY1, KEY0 = 0b01 + write the pseudo error setting value (same as the previous value) ■KEY1, KEY0 = 0b10 + write the pseudo error setting value (same as the previous value) ■KEY1, KEY0 = 0b11 + write the pseudo error setting value (same as the previous value)

Section	Change Results
CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 8.Setup Procedure and Program Flow for Operation Mode 3 (LIN Communication Mode) 8.2.Assist Mode	Revised “Figure 8-8 Example of LIN Communication Slave Mode (Assist Mode/a FIFO Is Not Used)” as follows. (Error) 

Section	Change Results
	(Correct) 

Section	Change Results
CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 8.Setup Procedure and Program Flow for Operation Mode 3 (LIN Communication Mode) 8.2.Assist Mode	Revised “Figure 8-9 Example of LIN Communication Slave Mode (Assist Mode/a FIFO Is Used)” as follows. (Error) 

Section	Change Results
	(Correct) 

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.1.Serial Control Register (SCR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
When it is read, "0" is always read.<table><tr><th rowspan="2">UPCL</th><th colspan="2">Programmable Clear Bit</th></tr><tr><th>Write</th><th>Read</th></tr><tr><td>0</td><td>No effect</td><td rowspan="2">"0" is always read.</td></tr><tr><td>1</td><td>Programmable clear</td></tr></table>
(Correct)
When it is read, "0" is always read.<table><tr><th rowspan="2">Bit</th><th colspan="2">Programmable Clear Bit</th></tr><tr><th>Write</th><th>Read</th></tr><tr><td>0</td><td>No effect</td><td rowspan="2">"0" is always read.</td></tr><tr><td>1</td><td>Programmable clear</td></tr></table></td></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.1.Serial Control Register (SCR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
When set to "1": The system is placed in slave mode.<table><tr><th>MS</th><th>Master/Slave Function Select Bit</th></tr></table>
(Correct)
When set to "1": The system is placed in slave mode.<table><tr><th>Bit</th><th>Master/Slave Function Select Bit</th></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.1.Serial Control Register (SCR)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
"0" is always read.
<table>
<tr><th rowspan="2">LBR</th><th colspan="2">LIN Break Field Setting Bit</th></tr>
<tr><th>Write</th><th>Read</th></tr>
<tr><td>0</td><td>No effect</td><td rowspan="2">"0" always read</td></tr>
<tr><td>1</td><td>In LIN manual mode, a LIN Break Field is generated.
In LIN assist mode, the LIN Break Field to ID Field are transmitted.</td></tr>
</table>

(Correct)
"0" is always read.
<table>
<tr><th rowspan="2">Bit</th><th colspan="2">LIN Break Field Setting Bit</th></tr>
<tr><th>Write</th><th>Read</th></tr>
<tr><td>0</td><td>No effect</td><td rowspan="2">"0" always read</td></tr>
<tr><td>1</td><td>In LIN manual mode, a LIN Break Field is generated.
In LIN assist mode, the LIN Break Field to ID Field are transmitted.</td></tr>
</table>
</td></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.1.Serial Control Register (SCR)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
■This bit is set when the SCRS:RIES bit in the set register is set to "1".
<table>
<tr><th>RIE</th><th>Reception Interrupt Enable Bit</th></tr>
</table>

(Correct)
■This bit is set when the SCRS:RIES bit in the set register is set to "1".
<table>
<tr><th>Bit</th><th>Reception Interrupt Enable Bit</th></tr>
</table>
</td></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.1.Serial Control Register (SCR)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
■This bit is set when the SCRS:TIES bit in the set register is set to "1".
<table>
<tr><th>TIE</th><th>Transmission Interrupt Enable Bit</th></tr>
</table>

(Correct)
■This bit is set when the SCRS:TIES bit in the set register is set to "1".
<table>
<tr><th>Bit</th><th>Transmission Interrupt Enable Bit</th></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.1.Serial Control Register (SCR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■This bit is set when the SCRS:TBIES bit in the set register is set to "1".
<table><tr><td>TBIE</td><td>Transmission Bus Idle Interrupt Enable Bit</td></tr></table>
(Correct)
■This bit is set when the SCRS:TBIES bit in the set register is set to "1".
<table><tr><td>Bit</td><td>Transmission Bus Idle Interrupt Enable Bit</td></tr></table></td></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.1.Serial Control Register (SCR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■When set to "1": Data frame reception operation is enabled.
<table><tr><td>RXE</td><td>Reception Operation Enable Bit</td></tr></table>
(Correct)
■When set to "1": Data frame reception operation is enabled.
<table><tr><td>Bit</td><td>Reception Operation Enable Bit</td></tr></table></td></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.1.Serial Control Register (SCR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■When set to "1": Data frame transmission operation is enabled.
<table><tr><td>TXE</td><td>Transmission Operation Enable Bit</td></tr></table>
(Correct)
■When set to "1": Data frame reception operation is enabled.
<table><tr><td>Bit</td><td>Transmission Operation Enable Bit</td></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.2.Serial Mode Register (SMR)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit7:5] MD2, MD1, MD0: Operation mode setting bits
These bits set the operation mode.
"0b000": Set operation mode 0 (asynchronous normal mode).
"0b001": Set operation mode 1 (asynchronous multi-processor mode).
"0b010": Set operation mode 2 (clock synchronous mode).
"0b011": Set operation mode 3 (LIN communication mode).
"0b100": Set operation mode 4 (I2C mode).
This chapter describes the register and its operations in operation mode 3 (LIN communication mode).
<table>
<tr><th>MD2</th><th>MD1</th><th>MD0</th><th>Operation Mode Setting Bit</th></tr>
</table>

(Correct)
[bit7:5] MD[2:0]: Operation Mode Setting Bits
These bits set the operation mode.
0b000: Set operation mode 0 (asynchronous normal mode).
0b001: Set operation mode 1 (asynchronous multi-processor mode).
0b010: Set operation mode 2 (clock synchronous mode).
0b011: Set operation mode 3 (LIN communication mode).
0b100: Set operation mode 4 (I2C mode).
This chapter describes the register and its operations in operation mode 3 (LIN communication mode).
<table>
<tr><th>Bits</th><th>Operation Mode Setting Bit</th></tr>
</table>
</td></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.2.Serial Mode Register (SMR)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
"1": Set the SIN pin for external interrupts.
<table>
<tr><th>WUCR</th><th>WAKE UP Control Bit</th></tr>
</table>

(Correct)
"1": Set the SIN pin for external interrupts.
<table>
<tr><th>Bit</th><th>WAKE UP Control Bit</th></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.2.Serial Mode Register (SMR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
Settings where SBL is "1" and ESCR:ESBL is "1": The stop bit is set to 4 bits.
<table><tr><td>SBL</td><td>Stop Bit Length Selection Bit</td></tr></table>
(Correct)
Settings where SBL is "1" and ESCR:ESBL is "1": The stop bit is set to 4 bits.
<table><tr><td>Bit</td><td>Stop Bit Length Selection Bit</td></tr></table></td></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.2.Serial Mode Register (SMR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■This bit is set when the SMRS:SOES bit in the set register is set to "1".
<table><tr><td>SOE</td><td>Serial Data Output Enable Bit</td></tr></table>
(Correct)
■This bit is set when the SMRS:SOES bit in the set register is set to "1".
<table><tr><td>Bit</td><td>Serial Data Output Enable Bit</td></tr></table></td></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.3.Serial Status Register (SSR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
If it is read, "0" is always read.
<table><tr><td rowspan="2">REC</td><td colspan="2">Reception Error Flag Clear Bit</td></tr><tr><td>Write</td><td>Read</td></tr></table>
(Correct)
If it is read, "0" is always read.
<table><tr><td rowspan="2">Bit</td><td colspan="2">Reception Error Flag Clear Bit</td></tr><tr><td>Write</td><td>Read</td></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.3.Serial Status Register (SSR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
Writing to this bit has no effect.
<table><tr><td rowspan="2">LBD</td><td colspan="2">LIN Break Field Flag Bit</td></tr><tr><td>Write</td><td>Read</td></tr></table>
(Correct)
Writing to this bit has no effect.
<table><tr><td rowspan="2">Bit</td><td colspan="2">LIN Break Field Flag Bit</td></tr><tr><td>Write</td><td>Read</td></tr></table></td></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.3.Serial Status Register (SSR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■If this flag is set when a reception FIFO is used, the reception FIFO enable bits are cleared, and no reception data is stored to the reception FIFO.
<table><tr><td>FRE</td><td>Framing Error Flag Bit</td></tr></table>
(Correct)
■If this flag is set when a reception FIFO is used, the reception FIFO enable bits are cleared, and no reception data is stored to the reception FIFO.
<table><tr><td>Bit</td><td>Framing Error Flag Bit</td></tr></table></td></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.3.Serial Status Register (SSR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■If this flag is set when a reception FIFO is used, the reception FIFO enable bits are cleared, and no reception data is stored to the reception FIFO.
<table><tr><td>ORE</td><td>Overrun Error Flag Bit</td></tr></table>
(Correct)
■If this flag is set when a reception FIFO is used, the reception FIFO enable bits are cleared, and no reception data is stored to the reception FIFO.
<table><tr><td>Bit</td><td>Overrun Error Flag Bit</td></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.3.Serial Status Register (SSR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■For ECR:RXBLKEN = 1, when the reception FIFO is used, this bit is cleared to "0" when the number of data items in the reception FIFO is equal to or less than the value set in the FBYTE register.
<table><tr><td>RDRF</td><td>Reception Data Full Flag Bit</td></tr></table>
(Correct)
■For ECR:RXBLKEN = 1, when the reception FIFO is used, this bit is cleared to "0" when the number of data items in the reception FIFO is equal to or less than the value set in the FBYTE register.
<table><tr><td>Bit</td><td>Reception Data Full Flag Bit</td></tr></table></td></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.3.Serial Status Register (SSR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■For details on the setting and clearing timing of the TDRE bit when the transmission FIFO is used, see "Occurrence of Interrupts and Flag Set Timing When Using Transmission FIFO.”
<table><tr><td>TDRE</td><td>Transmission Data Empty Flag Bit</td></tr></table>
(Correct)
■For details on the setting and clearing timing of the TDRE bit when the transmission FIFO is used, see "Occurrence of Interrupts and Flag Set Timing When Using Transmission FIFO.”
<table><tr><td>Bit</td><td>Transmission Data Empty Flag Bit</td></tr></table></td></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.3.Serial Status Register (SSR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■A transmission interrupt request is issued if this bit is set to "1" and transmission bus idle interrupt is enabled (SCR:TBIE = 1).
<table><tr><td>TBI</td><td>Transmission Bus Idle Flag Bit</td></tr></table>
(Correct)
■A transmission interrupt request is issued if this bit is set to "1" and transmission bus idle interrupt is enabled (SCR:TBIE = 1).
<table><tr><td>Bit</td><td>Transmission Bus Idle Flag Bit</td></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.4.Extended Communication Control Register (ESCR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
Settings where SBL is "1" and ESCR:ESBL is "1": The stop bit is set to 4 bits.
<table><tr><td>ESBL</td><td>Extended Stop Bit Length Selection Bit</td></tr></table>
(Correct)
Settings where SBL is "1" and ESCR:ESBL is "1": The stop bit is set to 4 bits.
<table><tr><td>Bit</td><td>Extended Stop Bit Length Selection Bit</td></tr></table></td></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.4.Extended Communication Control Register (ESCR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
This bit is set when the ESCRS:LBIES bit in the set register is set to "1".
<table><tr><td>LBIE</td><td>LIN Break Field Detection Interrupt Enable Bit</td></tr></table>
(Correct)
This bit is set when the ESCRS:LBIES bit in the set register is set to "1".
<table><tr><td>Bit</td><td>LIN Break Field Detection Interrupt Enable Bit</td></tr></table></td></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.4.Extended Communication Control Register (ESCR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit5, bit3:2] LBL 2/1/0: LIN Break Field length selection bits (effective only for master operation)
■These bits set the LIN Break Field duration as a number of bits.
■Set these bits before setting the LBR bit in the serial control register (SCR) to "1" (LIN Break Field transmission).
■During slave operation, a LIN Break Field is detected at the 11th bit regardless of the setting of these bits.
<table><tr><td>LBL2</td><td>LBL1</td><td>LBL0</td><td>LIN Break Field Length Selection Bit</td></tr></table>
(Correct)
[bit5, bit3:2] LBL[2:0]: LIN Break Field Length Selection Bits (Effective Only for Master Operation)
■These bits set the LIN Break Field duration as a number of bits.
■Set these bits before setting the LBR bit in the serial control register (SCR) to "1" (LIN Break Field transmission).
■During slave operation, a LIN Break Field is detected at the 11th bit regardless of the setting of these bits.
<table><tr><td>Bits</td><td>LIN Break Field Length Selection Bit</td></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.4.Extended Communication Control Register (ESCR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit1:0] DEL1/0: LIN Break delimiter length selection bits (effective only for master operation)
■These bits set the length of the LIN Break delimiter as a number of bits.
■Set these bits before setting the LBR bit in the serial control register (SCR) to "1" (LIN Break Field transmission).
<table><tr><th>DEL1</th><th>DEL0</th><th>LIN Break Delimiter Length Selection Bit</th></tr></table>
(Correct)
[bit1:0] DEL[1:0]: LIN Break Delimiter Length Selection Bits (Effective Only for Master Operation)
■These bits set the length of the LIN Break delimiter as a number of bits.
■Set these bits before setting the LBR bit in the serial control register (SCR) to "1" (LIN Break Field transmission).
<table><tr><th>Bits</th><th>LIN Break Delimiter Length Selection Bit</th></tr></table></td></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.6.Serial Auxiliary Control Status Register (SACSR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
This bit is set when the SACSRS:STSTS bit in the set register is set to "1".
<table><tr><th>STST</th><th>Serial Test Bit</th></tr></table>
(Correct)
This bit is set when the SACSRS:STSTS bit in the set register is set to "1".
<table><tr><th>Bit</th><th>Serial Test Bit</th></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.6.Serial Auxiliary Control Status Register (SACSR)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
This bit is updated upon the detection of a 5th falling edge of the LIN bus in the Sync Field.
<table>
<tr><th rowspan="2">BST</th><th colspan="2">Baud Rate Setting Flag</th></tr>
<tr><th>Write</th><th>Read</th></tr>
<tr><td>0</td><td rowspan="2">No effect</td><td>Without auto baud rate adjustment</td></tr>
<tr><td>1</td><td>With auto baud rate adjustment</td></tr>
</table>
(Correct)
This bit is updated upon the detection of a 5th falling edge of the LIN bus in the Sync Field.
<table>
<tr><th rowspan="2">Bit</th><th colspan="2">Baud Rate Setting Flag</th></tr>
<tr><th>Write</th><th>Read</th></tr>
<tr><td>0</td><td rowspan="2">No effect</td><td>Without auto baud rate adjustment</td></tr>
<tr><td>1</td><td>With auto baud rate adjustment</td></tr>
</table>
</td></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.6.Serial Auxiliary Control Status Register (SACSR)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
This bit is reset when the SACSRC:SFDC bit in the clear register is set to "1".
<table>
<tr><th rowspan="2">SFD</th><th colspan="2">Sync Field Detection Flag</th></tr>
<tr><th>Write</th><th>Read</th></tr>
<tr><td>0</td><td>No effect</td><td>Sync Field not detected</td></tr>
<tr><td>1</td><td>No effect</td><td>Sync Field detected</td></tr>
</table>
(Correct)
This bit is reset when the SACSRC:SFDC bit in the clear register is set to "1".
<table>
<tr><th rowspan="2">Bit</th><th colspan="2">Sync Field Detection Flag</th></tr>
<tr><th>Write</th><th>Read</th></tr>
<tr><td>0</td><td>No effect</td><td>Sync Field not detected</td></tr>
<tr><td>1</td><td>No effect</td><td>Sync Field detected</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.6.Serial Auxiliary Control Status Register (SACSR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
This bit is set when the SACSRS:SFDES bit in the set register is set to "1".
<table><tr><th>SFDE</th><th>Sync Field Detection Interrupt Enable Bit</th></tr></table>
(Correct)
This bit is set when the SACSRS:SFDES bit in the set register is set to "1".
<table><tr><th>Bit</th><th>Sync Field Detection Interrupt Enable Bit</th></tr></table></td></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.6.Serial Auxiliary Control Status Register (SACSR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
This bit is set when the SACSRS:AUTES bit in the set register is set to "1".
<table><tr><th>AUTE</th><th>Auto Baud Rate Adjustment Bit</th></tr><tr><td>0</td><td>Disable auto baud rate adjustment.</td></tr><tr><td>1</td><td>Enable auto baud rate adjustment.</td></tr></table>
Note:
−This bit is internally fixed to "0" in master mode (SCR:MS = 0).
−Setting this bit to "1" sets the timer operation clock division bits (TDIV3 to TDIV0) to "3h" (division by 8).

(Correct)
This bit is set when the SACSRS:AUTES bit in the set register is set to "1".
<table><tr><th>Bit</th><th>Auto Baud Rate Adjustment Bit</th></tr><tr><td>0</td><td>Disable auto baud rate adjustment.</td></tr><tr><td>1</td><td>Enable auto baud rate adjustment.</td></tr></table>
Note:
−This bit is internally fixed to "0" in master mode (SCR:MS = 0).
−Setting this bit to "1" sets the timer operation clock division bits (TDIV[3:0]) to 0b0011 (division by 8).</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.6.Serial Auxiliary Control Status Register (SACSR)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit10:9] TRG1-0: trigger selection bits
These bits select the detection method for the external trigger edges that are used to activate the serial timer.
<table>
<tr><th>TRG1</th><th>TRG0</th><th>Edge Detection Method for External Trigger</th></tr>
</table>

(Correct)
[bit10:9] TRG[1:0]: Trigger Selection Bits
These bits select the detection method for the external trigger edges that are used to activate the serial timer.
<table>
<tr><th>Bits</th><th>Edge Detection Method for External Trigger</th></tr>
</table>
</td></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.6.Serial Auxiliary Control Status Register (SACSR)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
■This bit is reset when the SACSRC:TINTC bit in the clear register is set to "1.
<table>
<tr><th rowspan="2">TINT</th><th colspan="2">Description</th></tr>
<tr><th>Write</th><th>Read</th></tr>
<tr><td>0</td><td>No effect</td><td>No timer interrupt request</td></tr>
<tr><td>1</td><td>No effect</td><td>Timer interrupt request issued</td></tr>
</table>

(Correct)
■This bit is reset when the SACSRC:TINTC bit in the clear register is set to "1.
<table>
<tr><th rowspan="2">Bit</th><th colspan="2">Description</th></tr>
<tr><th>Write</th><th>Read</th></tr>
<tr><td>0</td><td>No effect</td><td>No timer interrupt request</td></tr>
<tr><td>1</td><td>No effect</td><td>Timer interrupt request issued</td></tr>
</table>
</td></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.6.Serial Auxiliary Control Status Register (SACSR)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
■This bit is set when the bit SACSRS:TINTES in the set register is set to "1".
<table>
<tr><th>TINTE</th><th>Description</th></tr>
</table>

(Correct)
■This bit is set when the bit SACSRS:TINTES in the set register is set to "1".
<table>
<tr><th>Bit</th><th>Description</th></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.6.Serial Auxiliary Control Status Register (SACSR)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
This bit is set when the SACSRS:TRGES bit in the set register is set to "1".
<table>
<tr><th>TRGE</th><th>Description</th></tr>
</table>
(Correct)
This bit is set when the SACSRS:TRGES bit in the set register is set to "1".
<table>
<tr><th>Bit</th><th>Description</th></tr>
</table>
</td></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.6.Serial Auxiliary Control Status Register (SACSR)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit4:1] TDIV3-0: timer operation clock division bits
These bits set the division ratio of the serial timer.
<table>
<tr><th rowspan="2">TDIV3</th><th rowspan="2">TDIV2</th><th rowspan="2">TDIV1</th><th rowspan="2">TDIV0</th><th colspan="7">Timer Operation Clock</th></tr>
<tr><th>Division Ratio</th><th>Φ = 8 MHz</th><th>Φ = 10 MHz</th><th>Φ = 16 MHz</th><th>Φ = 20 MHz</th><th>Φ = 24 MHz</th><th>Φ = 32 MHz</th></tr>
</table>
(Correct)
[bit4:1] TDIV[3:0]: Timer Operation Clock Division Bits
These bits set the division ratio of the serial timer.
<table>
<tr><th rowspan="2">Bits</th><th colspan="7">Timer Operation Clock</th></tr>
<tr><th>Division Ratio</th><th>Φ = 8 MHz</th><th>Φ = 10 MHz</th><th>Φ = 16 MHz</th><th>Φ = 20 MHz</th><th>Φ = 24 MHz</th><th>Φ = 32 MHz</th></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.6.Serial Auxiliary Control Status Register (SACSR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
This bit is set when the SACSRS:TMRES bit in the set register is set to "1".
<table>
<tr><th>TMRE</th><th>Serial Timer Enable Bit</th></tr>
<tr><td>0</td><td>Stop serial timer operation.
When stopped, the value of the serial timer register (STMR) is retained.</td></tr>
<tr><td>1</td><td>Changing this bit from "0" to "1" initializes the serial timer register (STMR) to "0" and starts serial timer operation.</td></tr>
</table>
(Correct)
This bit is set when the SACSRS:TMRES bit in the set register is set to "1".
<table>
<tr><th>Bit</th><th>Serial Timer Enable Bit</th></tr>
<tr><td>0</td><td>Stop serial timer operation.
When stopped, the value of the serial timer register (STMR) is retained.</td></tr>
<tr><td>1</td><td>Changing this bit from "0" to "1" initializes the serial timer register (STMR) to "0" and starts serial timer operation.</td></tr>
</table></td></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.11.Baud Rate Generator Register 1, 0 (BGR1, BGR0)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit15] EXT: External clock selection bit
<table>
<tr><th>EXT</th><th>External Clock Selection Bit</th></tr>
</table>
(Correct)
[bit15] EXT: External Clock Selection Bit
<table>
<tr><th>Bit</th><th>External Clock Selection Bit</th></tr>
</table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.12.LIN Assist Mode Status Register (LAMSR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
−LIN checksum error flag bit (LCSER)
<table><tr><td>LER</td><td>LIN Representative Error Flag Bit</td></tr></table>
(Correct)
−LIN checksum error flag bit (LCSER)
<table><tr><td>Bit</td><td>LIN Representative Error Flag Bit</td></tr></table></td></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.12.LIN Assist Mode Status Register (LAMSR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
−Overrun error flag bit (ORE)
<table><tr><td>SER</td><td>Serial Interface Representative Error Flag Bit</td></tr></table>
(Correct)
−Overrun error flag bit (ORE)
<table><tr><td>Bit</td><td>Serial Interface Representative Error Flag Bit</td></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.12.LIN Assist Mode Status Register (LAMSR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
No effect.
<table>
<tr><th rowspan="2">LCSC</th><th colspan="2">Checksum Calculation Completion Flag Bit</th></tr>
<tr><th>Write</th><th>Read</th></tr>
<tr><td>0</td><td>No effect</td><td>Checksum calculation is in progress.
Or
Checksum calculation start is being awaited.</td></tr>
<tr><td>1</td><td>No effect</td><td>Checksum calculation has been completed.</td></tr>
</table>

(Correct)
No effect.
<table>
<tr><th rowspan="2">Bit</th><th colspan="2">Checksum Calculation Completion Flag Bit</th></tr>
<tr><th>Write</th><th>Read</th></tr>
<tr><td>0</td><td>No effect</td><td>Checksum calculation is in progress.
Or
Checksum calculation start is being awaited.</td></tr>
<tr><td>1</td><td>No effect</td><td>Checksum calculation has been completed.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.12.LIN Assist Mode Status Register (LAMSR)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
No effect.
<table>
<tr><th rowspan="2">LAHC</th><th colspan="2">LIN Auto Header Completion Flag Bit</th></tr>
<tr><th>Write</th><th>Read</th></tr>
<tr><td>0</td><td>No effect</td><td>LIN auto header is being received.
Or, its reception is being awaited.</td></tr>
<tr><td>1</td><td>No effect</td><td>LIN auto header has been received.</td></tr>
</table>

(Correct)
No effect.
<table>
<tr><th rowspan="2">Bit</th><th colspan="2">LIN Auto Header Completion Flag Bit</th></tr>
<tr><th>Write</th><th>Read</th></tr>
<tr><td>0</td><td>No effect</td><td>LIN auto header is being received.
Or, its reception is being awaited.</td></tr>
<tr><td>1</td><td>No effect</td><td>LIN auto header has been received.</td></tr>
</table>
</td></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.13.LIN Assist Mode Control Register (LAMCR)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit7:4] LDL3, LDL2, LDL1, LDL0: LIN data length setting bit
■These bits set the length of the LIN response data within a range of 0 to 8 bytes.
■The setting value should represent the data length.
■Data of the length specified by these bits is sent in the transmission operation.
■Data of the length specified by these bits is received in the reception operation.
<table>
<tr><th>LDL3</th><th>LDL2</th><th>LDL1</th><th>LDL0</th><th>LIN Data Length Setting Bits</th></tr>
</table>

(Correct)
[bit7:4] LDL[3:0]: LIN Data Length Setting Bit
■These bits set the length of the LIN response data within a range of 0 to 8 bytes.
■The setting value should represent the data length.
■Data of the length specified by these bits is sent in the transmission operation.
■Data of the length specified by these bits is received in the reception operation.
<table>
<tr><th>Bits</th><th>LIN Data Length Setting Bits</th></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.13.LIN Assist Mode Control Register (LAMCR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■This bit is set when the LAMCRS:LTDRCLS bit in the set register is set to "1".
<table><tr><th rowspan="2">LTDRCL</th><th colspan="2">LIN Transmission Data Register Clear Bit</th></tr><tr><th>Write</th><th>Read</th></tr></table>
(Correct)
■This bit is set when the LAMCRS:LTDRCLS bit in the set register is set to "1".
<table><tr><th rowspan="2">Bit</th><th colspan="2">LIN Transmission Data Register Clear Bit</th></tr><tr><th>Write</th><th>Read</th></tr></table></td></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.13.LIN Assist Mode Control Register (LAMCR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
"1": The extended checksum is selected.
<table><tr><th>LCSTYP</th><th>LIN Checksum Type Selection Bit</th></tr></table>
(Correct)
"1": The extended checksum is selected.
<table><tr><th>Bit</th><th>LIN Checksum Type Selection Bit</th></tr></table></td></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.13.LIN Assist Mode Control Register (LAMCR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
When set to "1": The received data of LIN ID Field is stored into the LIN assist mode reception ID register (LAMRID).
<table><tr><th rowspan="2">LIDEN</th><th colspan="2">LIN ID Register Use Enable Bit</th></tr><tr><th>Master</th><th>Slave</th></tr></table>
(Correct)
When set to "1": The received data of LIN ID Field is stored into the LIN assist mode reception ID register (LAMRID).
<table><tr><th rowspan="2">Bit</th><th colspan="2">LIN ID Register Use Enable Bit</th></tr><tr><th>Master</th><th>Slave</th></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.13.LIN Assist Mode Control Register (LAMCR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
When set to "1": LIN assist mode is selected.
<table><tr><td>LAMEN</td><td>LIN Assist Mode Enable Bit</td></tr></table>
(Correct)
When set to "1": LIN assist mode is selected.
<table><tr><td>Bit</td><td>LIN Assist Mode Enable Bit</td></tr></table></td></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.14.LIN Assist Mode Interrupt Enable Register (LAMIER)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■This bit is set when the LAMIERS:LCERIES bit in the set register is set to "1".
<table><tr><td>LCSERIE</td><td>LIN Checksum Error Interrupt Enable Bit</td></tr></table>
(Correct)
■This bit is set when the LAMIERS:LCERIES bit in the set register is set to "1".
<table><tr><td>Bit</td><td>LIN Checksum Error Interrupt Enable Bit</td></tr></table></td></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.14.LIN Assist Mode Interrupt Enable Register (LAMIER)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■This bit is set when the LAMIERS:LPTERIES bit in the set register is set to "1".
<table><tr><td>LPTERIE</td><td>LIN ID Parity Error Interrupt Enable Bit</td></tr></table>
(Correct)
■This bit is set when the LAMIERS:LPTERIES bit in the set register is set to "1".
<table><tr><td>Bit</td><td>LIN ID Parity Error Interrupt Enable Bit</td></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.14.LIN Assist Mode Interrupt Enable Register (LAMIER)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■This bit is set when the LAMIERS:LSFERIES bit in the set register is set to "1".
<table><tr><td>LSFERIE</td><td>LIN Sync Data Error Interrupt Enable Bit</td></tr></table>
(Correct)
■This bit is set when the LAMIERS:LSFERIES bit in the set register is set to "1".
<table><tr><td>Bit</td><td>LIN Sync Data Error Interrupt Enable Bit</td></tr></table></td></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.14.LIN Assist Mode Interrupt Enable Register (LAMIER)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■This bit is set when the LAMIERS:LBSERIES bit in the set register is set to "1".
<table><tr><td>LBSERIE</td><td>LIN Bus Error Interrupt Enable Bit</td></tr></table>
(Correct)
■This bit is set when the LAMIERS:LBSERIES bit in the set register is set to "1".
<table><tr><td>Bit</td><td>LIN Bus Error Interrupt Enable Bit</td></tr></table></td></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.14.LIN Assist Mode Interrupt Enable Register (LAMIER)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■This bit is set when the LAMIERS:LCSCIES bit in the set register is set to "1".
<table><tr><td>LCSCIE</td><td>LIN Checksum Calculation Completion Interrupt Enable Bit</td></tr></table>
(Correct)
■This bit is set when the LAMIERS:LCSCIES bit in the set register is set to "1".
<table><tr><td>Bit</td><td>LIN Checksum Calculation Completion Interrupt Enable Bit</td></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.14.LIN Assist Mode Interrupt Enable Register (LAMIER)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■This bit is set when the LAMIERS:LAHCIES bit in the set register is set to "1".
<table><tr><td>LAHCIE</td><td>LIN Auto Header Completion Interrupt Enable Bit</td></tr></table>
(Correct)
■This bit is set when the LAMIERS:LAHCIES bit in the set register is set to "1".
<table><tr><td>Bit</td><td>LIN Auto Header Completion Interrupt Enable Bit</td></tr></table></td></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.16.LIN Assist Mode Error Status Register (LAMESR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■This bit is reset when the LAMESRC:LCSERC bit in the clear register is set to "1".
<table><tr><td rowspan="2">LCSER</td><td colspan="2">LIN Checksum Error Flag Bit</td></tr><tr><td>Write</td><td>Read</td></tr></table>
(Correct)
■This bit is reset when the LAMESRC:LCSERC bit in the clear register is set to "1".
<table><tr><td rowspan="2">Bit</td><td colspan="2">LIN Checksum Error Flag Bit</td></tr><tr><td>Write</td><td>Read</td></tr></table></td></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.16.LIN Assist Mode Error Status Register (LAMESR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■This bit is reset when the LAMESRC:LPTERC bit in the clear register is set to "1".
<table><tr><td rowspan="2">LPTER</td><td colspan="2">LIN ID Parity Error Flag Bit</td></tr><tr><td>Write</td><td>Read</td></tr></table>
(Correct)
■This bit is reset when the LAMESRC:LPTERC bit in the clear register is set to "1".
<table><tr><td rowspan="2">Bit</td><td colspan="2">LIN ID Parity Error Flag Bit</td></tr><tr><td>Write</td><td>Read</td></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.16.LIN Assist Mode Error Status Register (LAMESR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■This bit is reset when the LAMESRC:LSFERC bit in the clear register is set to "1".
<table><tr><th rowspan="2">LSFER</th><th colspan="2">LIN Sync Data Error Flag Bit</th></tr><tr><th>Write</th><th>Read</th></tr></table>
(Correct)
■This bit is reset when the LAMESRC:LSFERC bit in the clear register is set to "1".
<table><tr><th rowspan="2">Bit</th><th colspan="2">LIN Sync Data Error Flag Bit</th></tr><tr><th>Write</th><th>Read</th></tr></table></td></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.16.LIN Assist Mode Error Status Register (LAMESR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■This bit is reset when the LAMESRC:LBSERC bit in the clear register is set to "1".
<table><tr><th rowspan="2">LBSER</th><th colspan="2">LIN Bus Error Flag Bit</th></tr><tr><th>Write</th><th>Read</th></tr></table>
(Correct)
■This bit is reset when the LAMESRC:LBSERC bit in the clear register is set to "1".
<table><tr><th rowspan="2">Bit</th><th colspan="2">LIN Bus Error Flag Bit</th></tr><tr><th>Write</th><th>Read</th></tr></table></td></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.17.LIN Assist Mode Error Test Register (LAMERT)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■To set the pseudo error setting, write a value according to the following procedure.
−KEY1, KEY0 = "00" + write a pseudo error setting value
−KEY1, KEY0 = "01" + write the pseudo error setting value (same as the previous value)
−KEY1, KEY0 = "10" + write the pseudo error setting value (same as the previous value)
−KEY1, KEY0 = "11" + write the pseudo error setting value (same as the previous value)

(Correct)
■To set the pseudo error setting, write a value according to the following procedure.
−KEY1, KEY0 = 0b00 + write a pseudo error setting value
−KEY1, KEY0 = 0b01 + write the pseudo error setting value (same as the previous value)
−KEY1, KEY0 = 0b10 + write the pseudo error setting value (same as the previous value)
−KEY1, KEY0 = 0b11 + write the pseudo error setting value (same as the previous value)</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.17.LIN Assist Mode Error Test Register (LAMERT)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■The pseudo error function is enabled and the error continues to occur until the setting of this bit is disabled (= "0").
<table><tr><td>LCSERT</td><td>LIN Checksum Error Pseudo Error Setting Bit</td></tr></table>
(Correct)
■The pseudo error function is enabled and the error continues to occur until the setting of this bit is disabled (=0).
<table><tr><td>Bit</td><td>LIN Checksum Error Pseudo Error Setting Bit</td></tr></table></td></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.17.LIN Assist Mode Error Test Register (LAMERT)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■The pseudo error function is enabled and the error continues to occur until the setting of this bit is disabled (= "0").
<table><tr><td>LPTERT</td><td>LIN ID Parity Error Pseudo Error Setting Bit</td></tr></table>
(Correct)
■The pseudo error function is enabled and the error continues to occur until the setting of this bit is disabled (=0).
<table><tr><td>Bit</td><td>LIN ID Parity Error Pseudo Error Setting Bit</td></tr></table></td></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.17.LIN Assist Mode Error Test Register (LAMERT)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■The pseudo error function is enabled and this bit continues to output the inverted Sync Field data until this bit set is disabled (= "0").
<table><tr><td>LSFERT</td><td>LIN Sync Data Error Pseudo Error Setting Bit</td></tr></table>
(Correct)
■The pseudo error function is enabled and this bit continues to output the inverted Sync Field data until this bit set is disabled (=0).
<table><tr><td>Bit</td><td>LIN Sync Data Error Pseudo Error Setting Bit</td></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.17.LIN Assist Mode Error Test Register (LAMERT)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■The pseudo error function is enabled and the error continues to occur until the setting of this bit is disabled (= "0").
<table><tr><td>LBSERT</td><td>LIN Bus Error Pseudo Error Setting Bit</td></tr></table>
(Correct)
■The pseudo error function is enabled and the error continues to occur until the setting of this bit is disabled (=0).
<table><tr><td>Bit</td><td>LIN Bus Error Pseudo Error Setting Bit</td></tr></table></td></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.17.LIN Assist Mode Error Test Register (LAMERT)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■The pseudo error function is enabled and the error continues to occur until the setting of this bit is disabled (= "0").
<table><tr><td>FRET</td><td>Framing Error Pseudo Error Setting Bit</td></tr></table>
(Correct)
■The pseudo error function is enabled and the error continues to occur until the setting of this bit is disabled (= 0).
<table><tr><td>Bit</td><td>LIN Auto Header Completion Interrupt Enable Bit</td></tr></table></td></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.18.FIFO Control Register 1 (FCR1)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
"1": Enable FLST bit detection.
<table><tr><td>FLSTE</td><td>Retransmission Data Lost Detection Enable Bit</td></tr></table>
(Correct)
"1": Enable FLST bit detection.
<table><tr><td>Bit</td><td>Retransmission Data Lost Detection Enable Bit</td></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.18.FIFO Control Register 1 (FCR1)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
"1": Enable detection of reception idle status.
<table><tr><td>FRIIE</td><td>Reception FIFO Idle Detection Enable Bit</td></tr></table>
(Correct)
"1": Enable detection of reception idle status.
<table><tr><td>Bit</td><td>Reception FIFO Idle Detection Enable Bit</td></tr></table></td></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.18.FIFO Control Register 1 (FCR1)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■FTICR setting value ≥ number of vacant data items in the transmission FIFO
<table><tr><td>FDRQ</td><td>Transmission FIFO Data Request Bit</td></tr></table>
(Correct)
■FTICR setting value ≥ number of vacant data items in the transmission FIFO
<table><tr><td>Bit</td><td>Transmission FIFO Data Request Bit</td></tr></table></td></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.18.FIFO Control Register 1 (FCR1)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■This bit is set when the FCR1S:FTIES bit in the set register is set to "1".
<table><tr><td>FTIE</td><td>Transmission FIFO Interrupt Enable Bit</td></tr></table>
(Correct)
■This bit is set when the FCR1S:FTIES bit in the set register is set to "1".
<table><tr><td>Bit</td><td>Transmission FIFO Interrupt Enable Bit</td></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.18.FIFO Control Register 1 (FCR1)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
"1": Allocate as transmission FIFO: FIFO2 and as reception FIFO: FIFO1..
<table><tr><td>FSEL</td><td>FIFO Selection Bit</td></tr></table>
(Correct)
"1": Allocate as transmission FIFO: FIFO2 and as reception FIFO: FIFO1.
<table><tr><td>Bit</td><td>FIFO Selection Bit</td></tr></table></td></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.19.FIFO Control Register 0 (FCR0)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
If "1" is set in this bit, the data pointed to by the read pointer saved with the FSET bit is overwritten. For this reason, re-transmission with the FLD bit cannot be set even if an error occurs. When performing retransmission with this bit set to "1", reset the FIFO and then write the data to the FIFO again.
<table><tr><td>FLST</td><td>FIFO Retransmission Data Lost Flag Bit</td></tr></table>
(Correct)
If "1" is set in this bit, the data pointed to by the read pointer saved with the FSET bit is overwritten. For this reason, re-transmission with the FLD bit cannot be set even if an error occurs. When performing retransmission with this bit set to "1", reset the FIFO and then write the data to the FIFO again.
<table><tr><td>Bit</td><td>FIFO Retransmission Data Lost Flag Bit</td></tr></table></td></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.19.FIFO Control Register 0 (FCR0)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■This bit is set when the FCR0S:FLDS bit in the set register is set to "1".
<table><tr><td>FLD</td><td>FIFO Pointer Reload Bit</td></tr></table>
(Correct)
■This bit is set when the FCR0S:FLDS bit in the set register is set to "1".
<table><tr><td>Bit</td><td>FIFO Pointer Reload Bit</td></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.19.FIFO Control Register 0 (FCR0)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
"0": No effect.
<table><tr><td>FSET</td><td>FIFO Pointer Saving Bit</td></tr></table>
(Correct)
"0": No effect.
<table><tr><td>Bit</td><td>FIFO Pointer Saving Bit</td></tr></table></td></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.19.FIFO Control Register 0 (FCR0)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■This bit is set when the FCR0S:FCL2S bit in the set register is set to "1".
<table><tr><td rowspan="2">FCL2</td><td colspan="2">FIFO2 Reset Bit</td></tr><tr><td>Write</td><td>Read</td></tr></table>
(Correct)
■This bit is set when the FCR0S:FCL2S bit in the set register is set to "1".
<table><tr><td rowspan="2">Bit</td><td colspan="2">FIFO2 Reset Bit</td></tr><tr><td>Write</td><td>Read</td></tr></table></td></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.19.FIFO Control Register 0 (FCR0)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■This bit is set when the FCR0S:FCL1S bit in the set register is set to "1".
<table><tr><td rowspan="2">FCL1</td><td colspan="2">FIFO1 Reset Bit</td></tr><tr><td>Write</td><td>Read</td></tr></table>
(Correct)
■This bit is set when the FCR0S:FCL1S bit in the set register is set to "1".
<table><tr><td rowspan="2">Bit</td><td colspan="2">FIFO1 Reset Bit</td></tr><tr><td>Write</td><td>Read</td></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.19.FIFO Control Register 0 (FCR0)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■This bit is set when the FCR0S:FE2S bit in the set register is set to "1".
<table><tr><td>FE2</td><td>FIFO2 Operation Enable Bit</td></tr></table>
(Correct)
■This bit is set when the FCR0S:FE2S bit in the set register is set to "1".
<table><tr><td>Bit</td><td>FIFO2 Operation Enable Bit</td></tr></table></td></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.19.FIFO Control Register 0 (FCR0)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■This bit is set when the FCR0S:FE1S bit in the set register is set to "1".
<table><tr><td>FE1</td><td>FIFO1 Operation Enable Bit</td></tr></table>
(Correct)
■This bit is set when the FCR0S:FE1S bit in the set register is set to "1".
<table><tr><td>Bit</td><td>FIFO1 Operation Enable Bit</td></tr></table></td></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.20.FIFO Byte Register (FBYTE)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
FBYTE2, FBYTE1: FIFO2 data count indication bit, FIFO1 data count indication bit

(Correct)
[bit15:8] FBYTE2: FIFO2 Data Count inDication Bit
[bit7:0] FBYTE1: FIFO1 Data Count inDication Bit</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.20.FIFO Byte Register (FBYTE)</td><td>The item should be added as indicated by the shading below.

(Error)<table><tr><td>Write</td><td>Set the transfer count.</td></tr><tr><td>Read</td><td>Read valid data count.</td></tr></table>
(Correct)<table><tr><th>FBYTE2, FBYTE1</th><th>Description</th></tr><tr><td>Write</td><td>Set the transfer count.</td></tr><tr><td>Read</td><td>Read valid data count.</td></tr></table></td></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.21.Transmission FIFO Interrupt Control Register (FTICR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
FTICR2, FTICR1: FIFO2 data count indication bit, FIFO1 data count indication bit

(Correct)
[bit15:8]FTICR2: FIFO2 Data Count Indication Bit
[bit7:0]FTICR1: FIFO1 Data Count Indication Bit</td></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.21.Transmission FIFO Interrupt Control Register (FTICR)</td><td>The item should be added as indicated by the shading below.

(Error)<table><tr><td>Write</td><td>Set the number of valid data items that generates an interrupt.</td></tr><tr><td>Read</td><td>Read valid data count.</td></tr></table>
(Correct)<table><tr><th>FTICR2, FTICR1</th><th>Description</th></tr><tr><td>Write</td><td>Set the number of valid data items that generates an interrupt.</td></tr><tr><td>Read</td><td>Read valid data count.</td></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.22.Extended Control Register (ECR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
This bit selects whether an error interrupt is output from the interrupt request pin or from the error interrupt request pin. For details on interrupt output selection, see Table 2-2, Table 2-3, Table 2-5, and Table 2-6.
<table><tr><td>EISEL</td><td>Error Interrupt Request Output Selection Bit</td></tr></table>
(Correct)
This bit selects whether an error interrupt is output from the interrupt request pin or from the error interrupt request pin. For details on interrupt output selection, see Table 2-2, Table 2-3, Table 2-5, and Table 2-6.
<table><tr><td>Bit</td><td>Error Interrupt Request Output Selection Bit</td></tr></table></td></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.22.Extended Control Register (ECR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
This bit enables/disables reception error interrupt request output. For details on the target interrupt factors, see Table 2-3 and Table 2-6.
<table><tr><td>REIE</td><td>Reception Error Interrupt Enable Bit</td></tr></table>
(Correct)
This bit enables/disables reception error interrupt request output. For details on the target interrupt factors, see Table 2-3 and Table 2-6.
<table><tr><td>Bit</td><td>Reception Error Interrupt Enable Bit</td></tr></table></td></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.22.Extended Control Register (ECR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
This bit enables/disables transmission error interrupt request output. For details on the target interrupt factors, see Table 2-2 and Table 2-5.
<table><tr><td>TEIE</td><td>Transmission Error Interrupt Enable Bit</td></tr></table>
(Correct)
This bit enables/disables transmission error interrupt request output. For details on the target interrupt factors, see Table 2-2 and Table 2-5.
<table><tr><td>Bit</td><td>Transmission Error Interrupt Enable Bit</td></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.22.Extended Control Register (ECR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■ "1": Perform DMA transfer in block transfer mode.<table><tr><th>RXBLKEN</th><th>Reception Block Transfer Setting Bit</th></tr></table>
(Correct)
■ "1": Perform DMA transfer in block transfer mode.<table><tr><th>Bit</th><th>Reception Block Transfer Setting Bit</th></tr></table></td></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.22.Extended Control Register (ECR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■ "1": Perform DMA transfer in block transfer mode.<table><tr><th>TXBLKEN</th><th>Transmission Block Transfer Setting Bit</th></tr></table>
(Correct)
■ "1": Perform DMA transfer in block transfer mode.<table><tr><th>Bit</th><th>Transmission Block Transfer Setting Bit</th></tr></table></td></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.23.Extended Status Register (ESR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■For details on interrupt request output, see Table 2-3 for manual mode, Table 2-6 for assist mode.<table><tr><th>RXUDR</th><th>Reception FIFO Under run Flag Bit</th></tr><tr><td>0</td><td>Indicate that no reception FIFO under run has occurred.</td></tr><tr><td>1</td><td>Indicate that reception FIFO under run has occurred.</td></tr></table>
(Correct)
■For details on interrupt request output, see Table 2-3 for manual mode, Table 2-6 for assist mode.<table><tr><th>Bit</th><th>Reception FIFO Underrun Flag Bit</th></tr><tr><td>0</td><td>Indicate that no reception FIFO underrun has occurred.</td></tr><tr><td>1</td><td>Indicate that reception FIFO underrun has occurred.</td></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.23.Extended Status Register (ESR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■For details on interrupt request output, see Table 2-2 for manual mode, Table 2-5 for assist mode.
<table>
<tr><th>TXOVR</th><th>Transmission FIFO Overrun Flag Bit</th></tr>
<tr><td>0</td><td>Indicate that no transmission FIFO overrun has occurred.</td></tr>
<tr><td>1</td><td>Indicate that transmission FIFO overrun has occurred.</td></tr>
</table>

(Correct)
■For details on interrupt request output, see Table 2-2 for manual mode, Table 2-5 for assist mode.
<table>
<tr><th>Bit</th><th>Transmission FIFO Overrun Flag Bit</th></tr>
<tr><td>0</td><td>Indicate that no transmission FIFO overrun has occurred.</td></tr>
<tr><td>1</td><td>Indicate that transmission FIFO overrun has occurred.</td></tr>
</table></td></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.23.Extended Status Register (ESR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■For details on interrupt request output, see Table 2-3 for manual mode, Table 2-6 for assist mode.
<table>
<tr><th>RBERR</th><th>Reception Block Transfer Error Bit</th></tr>
<tr><td>0</td><td>Indicate that no reception block transfer error has occurred.</td></tr>
<tr><td>1</td><td>Indicate that a reception block transfer error has occurred.</td></tr>
</table>

(Correct)
■For details on interrupt request output, see Table 2-3 for manual mode, Table 2-6 for assist mode.
<table>
<tr><th>Bit</th><th>Reception Block Transfer Error Bit</th></tr>
<tr><td>0</td><td>Indicate that no reception block transfer error has occurred.</td></tr>
<tr><td>1</td><td>Indicate that a reception block transfer error has occurred.</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.23.Extended Status Register (ESR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■For details on interrupt request output, see Table 2-2 for manual mode, Table 2-5 for assist mode.
<table>
<tr><th>TBERR</th><th>Transmission Block Transfer Error Bit</th></tr>
<tr><td>0</td><td>Indicate that no transmission block transfer error has occurred.</td></tr>
<tr><td>1</td><td>Indicate that a transmission block transfer error has occurred.</td></tr>
</table>
(Correct)
■For details on interrupt request output, see Table 2-2 for manual mode, Table 2-5 for assist mode.
<table>
<tr><th>Bit</th><th>Transmission Block Transfer Error Bit</th></tr>
<tr><td>0</td><td>Indicate that no transmission block transfer error has occurred.</td></tr>
<tr><td>1</td><td>Indicate that a transmission block transfer error has occurred.</td></tr>
</table></td></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.24.Transmission Block Size Register (TBSIZE)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
TBSIZE bit

(Correct)
[bit7:0]TBSIZE</td></tr>
<tr><td>CHAPTER39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.Interface (V2.1) Registers 9.24.Transmission Block Size Register (TBSIZE)</td><td>The item should be added as indicated by the shading below.

(Error)
<table>
<tr><td>Write</td><td>Set the transmission block count.</td></tr>
<tr><td>Read</td><td>Read a setting value.</td></tr>
</table>
(Correct)
<table>
<tr><th>TBSIZE</th><th>Description</th></tr>
<tr><td>Write</td><td>Set the transmission block count.</td></tr>
<tr><td>Read</td><td>Read a setting value.</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER40:
Base Timer
4.32-bit Mode
Operation</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
32-bit mode settings
First, reset the state by setting the FMD2 to FMD0 bits in the TMCR register of the even-numbered channel to "000" for reset mode. Then, in the same way as in 16-bit mode, make reload timer or PWC timer selection and operation settings. To set 32-bit operation mode at this time, write "1" to the T32 bit in the TMCR register. Keep the T32 bit of the odd-numbered channel at "0". There is no need to set reset mode.
For the reload timer, set the upper 16-bit reload value of the 32 bits in the cycle setting register of the odd-numbered channel. Then, set the lower 16-bit reload value in the cycle setting register of the even-numbered channel.
After the writing of the T32 bit, the change to 32-bit operation mode is reflected immediately, so change the settings for both channels in the count stopped state.
To change from 32-bit mode to 16-bit mode, set the FMD2 to FMD0 bits in the TMCR register of the even-numbered channel to "000" for reset mode. This resets the states of both the even-numbered and odd-numbered channels. As a result, the settings for 16-bit mode can be made for each of the channels.

(Correct)
32-bit Mode Settings
First, reset the state by setting the FMD2 to FMD0 bits in the TMCR register of the even-numbered channel to 0b000 for reset mode. Then, in the same way as in 16-bit mode, make reload timer or PWC timer selection and operation settings. To set 32-bit operation mode at this time, write "1" to the T32 bit in the TMCR register. Keep the T32 bit of the odd-numbered channel at "0". There is no need to set reset mode.
For the reload timer, set the upper 16-bit reload value of the 32 bits in the cycle setting register of the odd-numbered channel. Then, set the lower 16-bit reload value in the cycle setting register of the even-numbered channel.
After the writing of the T32 bit, the change to 32-bit operation mode is reflected immediately, so change the settings for both channels in the count stopped state.
To change from 32-bit mode to 16-bit mode, set the FMD2 to FMD0 bits in the TMCR register of the even-numbered channel to 0b000 for reset mode. This resets the states of both the even-numbered and odd-numbered channels. As a result, the settings for 16-bit mode can be made for each of the channels.</td></tr>
<tr><td>CHAPTER40:
Base Timer
7.Registers of
the Base Timer</td><td>The item should be added as indicated by the shading below.

(Error)
<table><tr><td>35</td><td>0x0078_2C00</td><td>23</td><td>0x0088_AC00</td><td>11</td><td>0x0080_AC00</td></tr></table>
Table 7-6 Register Map (Channel No.: 0) (12) (24)

(Correct)
<table><tr><td>35</td><td>0x0078_2C00</td><td>23</td><td>0x0088_AC00</td><td>11</td><td>0x0080_AC00</td></tr></table>
Tables 7-18, 7-19, 7-20 show the register map of each mode.
The "***/***/***/***" expression in each table corresponds to the reload/PMW/PPG/PWC timer, respectively.

Table 7-6 Register Map (Channel No.: 0) (12) (24)</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER40:
Base Timer
7.Registers of
the Base Timer</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
<table><tr><td>0x0000_0034</td><td>Reserved
11111111_11111111</td><td>BT_BTSSSR
11111111_11111111</td></tr><tr><td>0x0000_0038</td><td>Reserved
00000000_00000000</td><td>BT_BTTRR
00000000_00000000</td></tr></table>* The initial value is XXXXXXXX_XXXXXXXX only during reload timer operation.

(Correct)
<table><tr><td>0x0000_0034</td><td>Reserved
11111111_11111111</td><td>BT_BTSSSR0/12/24
11111111_11111111</td></tr><tr><td>0x0000_0038</td><td>Reserved
11111111_11111111</td><td>BT_BTTRR0/12/24
11110000_00000000</td></tr></table>* The initial value is XXXXXXXX_XXXXXXXX only during reload timer operation.</td></tr>
<tr><td>CHAPTER40:
Base Timer
7.Registers of
the Base Timer</td><td>The item should be added as indicated by the shading below.

(Error)
<table><tr><td>0x0000_0014</td><td>Reserved
00000000_00000000</td><td>Reserved
00000000</td><td>BTxx_STCC
00000000</td></tr><tr><td>0x0000_0018</td><td>Reserved
00000000_00000000</td><td>Reserved
00000000</td><td>BTxx_STCS
00000000</td></tr></table>* The initial value is XXXXXXXX_XXXXXXXX only during reload timer operation.

(Correct)
<table><tr><td>0x0000_0014</td><td>Reserved
00000000_00000000</td><td>Reserved
00000000</td><td>BTxx_STCC
00000000</td></tr><tr><td>0x0000_0018</td><td>Reserved
00000000_00000000</td><td>Reserved
00000000</td><td>BTxx_STCS
00000000</td></tr><tr><td>0x0000_001C</td><td>Reserved
00000000_00000000</td><td colspan="2">Reserved/BTxx_PSDR/Reserved/Reserved
00000000_00000000</td></tr><tr><td>0x0000_0020</td><td>Reserved
00000000_00000000</td><td colspan="2">Reserved/BTxx_ADTR/Reserved/Reserved
00000000_00000000</td></tr></table>* The initial value is XXXXXXXX_XXXXXXXX only during reload timer operation.</td></tr>
</table>

Section	Change Results
CHAPTER40: Base Timer 8.Notes on Using the Base Timer	Features of "For details" should be corrected as indicated by the shading below. (Error) ■All the registers of the base timer are initialized when the FMD2 to FMD0 bits in the TMCR register are set to "000" for reset mode. For this reason, all the registers must be reconfigured. ■The settings for bits other than the FMD2 to FMD0 bits in the TMCR register are ignored and initialized when the FMD2 to FMD0 bits in the TMCR register are set to "000" for reset mode. (Correct) ■All the registers of the base timer are initialized when the FMD2 to FMD0 bits in the TMCR register are set to 0b000 for reset mode. For this reason, all the registers must be reconfigured. ■The settings for bits other than the FMD2 to FMD0 bits in the TMCR register are ignored and initialized when the FMD2 to FMD0 bits in the TMCR register are set to 0b000 for reset mode.
CHAPTER40: Base Timer 8.Notes on Using the Base Timer	Features of "For details" should be corrected as indicated by the shading below. (Error) ■If the PWC mode setting (FMD2 to FMD0 = "100") and the measurement start setting (CTEN="1") are made from a system reset and reset mode at the same time, the resulting operation may depend on the state of the immediately preceding measurement signal. ■If a measurement start edge is detected at the same time that a restart is set in continuous measurement mode, the timer starts counting immediately from "0x0001". (Correct) ■If the PWC mode setting (FMD2 to FMD0 = 0b100) and the measurement start setting (CTEN= 1) are made from a system reset and reset mode at the same time, the resulting operation may depend on the state of the immediately preceding measurement signal. ■If a measurement start edge is detected at the same time that a restart is set in continuous measurement mode, the timer starts counting immediately from 0x0001.

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER40: Base Timer
9.Base Timer Description by Function Mode
9.1.PWM Timer Function
9.1.8.Timer Control Registers (TMCR, TMCR2), Status Control Register (STC), Status Control Clear Register (STCC), and Status Control Set Register (STCS)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[TMCR2:bit8,bit14:12] CKS3 to CKS0: Count clock selection bits
■These bits select a count clock for the 17-bit down counter.
■Any modification to the count clock is reflected immediately after the setting is changed. Therefore, modify CKS3 to CKS0 while counting is stopped (CTEN="0"). However, note that the bits can be modified at the same time as "1" is written to the count operation enable bit (CTEN).
<table>
<tr><th>CKS3</th><th>CKS2</th><th>CKS1</th><th>CKS0</th><th>Description</th></tr>
</table>

(Correct)
[TMCR2:bit8,bit14:12] CKS[3:0]: Count Clock Selection Bits
■These bits select a count clock for the 17-bit down counter.
■Any modification to the count clock is reflected immediately after the setting is changed. Therefore, modify CKS3 to CKS0 while counting is stopped (CTEN= 0). However, note that the bits can be modified at the same time as "1" is written to the count operation enable bit (CTEN).
<table>
<tr><th>Bits</th><th>Description</th></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER40: Base Timer
9.Base Timer
Description by Function Mode
9.1.PWM Timer Function
9.1.8.Timer Control Registers (TMCR, TMCR2), Status Control Register (STC), Status Control Clear Register (STCC), and Status Control Set Register (STCS)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit9:8] EGS1 to EGS0: Trigger input edge selection bits
■These bits select the valid edge for an input waveform as an external start factor, and they set trigger conditions.
■For the initial value or the "00" setting, no valid edge is selected for an input waveform, so no external waveform causes a start.
Note:
−If "1" is written to the STRG bit, the software trigger becomes valid regardless of the EGS1 and EGS0 settings.

■Modify EGS1 and EGS0 while counting is stopped (CTEN="0"). However, note that the bits can be modified at the same time as "1" is written to the CTEN bit.
<table>
<tr><th>EGS1</th><th>EGS0</th><th>Description</th></tr>
</table>

(Correct)
[bit9:8] EGS[1:0]: Trigger Input Edge Selection Bits
■These bits select the valid edge for an input waveform as an external start factor, and they set trigger conditions.
■For the initial value or the 0b00 setting, no valid edge is selected for an input waveform, so no external waveform causes a start.
Note:
−If "1" is written to the STRG bit, the software trigger becomes valid regardless of the EGS1 and EGS0 settings.

■Modify EGS1 and EGS0 while counting is stopped (CTEN= 0). However, note that the bits can be modified at the same time as "1" is written to the CTEN bit.
<table>
<tr><th>Bits</th><th>Description</th></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER40: Base Timer 9.Base Timer Description by Function Mode 9.1.PWM Timer Function 9.1.8.Timer Control Registers (TMCR, TMCR2), Status Control Register (STC), Status Control Clear Register (STCC), and Status Control Set Register (STCS)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit6:4] FMD2 to FMD0: Timer function selection bits
■These bits select the timer function.
■If "001" is set in the FMD2 to FMD0 bits, the PWM function is selected.
■Modify these bits while the timer is stopped (CTEN="0"). However, note that the bits can be modified at the same time as "1" is written to the CTEN bit.
<table><tr><th>FMD2</th><th>FMD1</th><th>FMD0</th><th>Description</th></tr></table>
(Correct)
[bit6:4] FMD[2:0]: Timer Function Selection Bits
■These bits select the timer function.
■If 0b001 is set in the FMD2 to FMD0 bits, the PWM function is selected.
■Modify these bits while the timer is stopped (CTEN= 0). However, note that the bits can be modified at the same time as "1" is written to the CTEN bit.
<table><tr><th>Bits</th><th>Description</th></tr></table></td></tr>
<tr><td>CHAPTER40: Base Timer 9.Base Timer Description by Function Mode 9.1.PWM Timer Function 9.1.8.Timer Control Registers (TMCR, TMCR2), Status Control Register (STC), Status Control Clear Register (STCC), and Status Control Set Register (STCS)</td><td>The function should be deleted as indicated by the shading below.

(Error)
Note:
−During timer operation of I/O mode 6 or I/O mode 4, if the falling edge is output from the even-numbered channel, this bit of the odd-numbered channel is cleared to "0".

[bit0] STRG: Software trigger bit

(Correct)
[bit0] STRG: Software Trigger Bit</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER40: Base Timer
9.Base Timer Description by Function Mode
9.2.PPG Timer Function
9.2.6.Timer Control Registers (TMCR, TMCR2), Status Control Register (STC), Status Control Clear Register (STCC), and Status Control Set Register (STCS) When the PPG Timer Is Selected</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[TMCR2:bit8,bit14:12] CKS3 to CKS0: Count clock selection bits
■These bits select a count clock for the 16-bit down counter.
■Any modification to the count clock is reflected immediately after the setting is changed. Therefore, modify CKS3 to CKS0 while counting is stopped (CTEN="0"). However, note that the bits can be modified at the same time as "1" is written to the CTEN bit.

Use of PWM start delay mode and ADC trigger mode
<table><tr><th>CKS3</th><th>CKS2</th><th>CKS1</th><th>CKS0</th><th>Description</th></tr></table>

(Correct)
[TMCR2:bit8,bit14:12] CKS[3:0]: Count Clock Selection Bits
■These bits select a count clock for the 16-bit down counter.
■Any modification to the count clock is reflected immediately after the setting is changed. Therefore, modify CKS3 to CKS0 while counting is stopped (CTEN= 0). However, note that the bits can be modified at the same time as "1" is written to the CTEN bit.

Use of PWM start delay mode and ADC trigger mode
<table><tr><th>Bits</th><th>Description</th></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER40: Base Timer
9.Base Timer Description by Function Mode
9.2.PPG Timer Function
9.2.6.Timer Control Registers (TMCR, TMCR2), Status Control Register (STC), Status Control Clear Register (STCC), and Status Control Set Register (STCS) When the PPG Timer Is Selected</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit9:8] EGS1 to EGS0: Trigger input edge selection bits
■These bits select the valid edge for an input waveform as an external start factor, and they set trigger conditions.
■For the initial value or the "00" setting, no valid edge is selected for an input waveform, so no external waveform causes a start.
Note:
−If "1" is written to the STRG bit, the software trigger becomes valid regardless of the EGS1 and EGS0 settings.

■Modify EGS1 and EGS0 while counting is stopped (CTEN="0"). However, note that the bits can be modified at the same time as "1" is written to the CTEN bit.
<table>
<tr><th>EGS1</th><th>EGS0</th><th>Description</th></tr>
</table>

(Correct)
[bit9:8] EGS[1:0]: Trigger Input Edge Selection Bits
■These bits select the valid edge for an input waveform as an external start factor, and they set trigger conditions.
■For the initial value or the 0b00 setting, no valid edge is selected for an input waveform, so no external waveform causes a start.
Note:
−If "1" is written to the STRG bit, the software trigger becomes valid regardless of the EGS1 and EGS0 settings.

■Modify EGS1 and EGS0 while counting is stopped (CTEN= 0). However, note that the bits can be modified at the same time as "1" is written to the CTEN bit.
<table>
<tr><th>Bits</th><th>Description</th></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER40: Base Timer 9.Base Timer Description by Function Mode 9.2.PPG Timer Function 9.2.6.Timer Control Registers (TMCR, TMCR2), Status Control Register (STC), Status Control Clear Register (STCC), and Status Control Set Register (STCS) When the PPG Timer Is Selected</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit6:4] FMD2 to FMD0: Timer function selection bits
■These bits select the timer function.
■If "010" is set in the FMD2 to FMD0 bits, the PPG function is selected.
■Modify these bits while the timer is stopped (CTEN="0"). However, note that the bits can be modified at the same time as "1" is written to the CTEN bit.
<table><tr><th>FMD2</th><th>FMD1</th><th>FMD0</th><th>Description</th></tr></table>
(Correct)
[bit6:4] FMD[2:0]: Timer Function Selection Bits
■These bits select the timer function.
■If 0b010 is set in the FMD2 to FMD0 bits, the PPG function is selected.
■Modify these bits while the timer is stopped (CTEN= 0). However, note that the bits can be modified at the same time as "1" is written to the CTEN bit.
<table><tr><th>Bits</th><th>Description</th></tr></table></td></tr>
<tr><td>CHAPTER40: Base Timer 9.Base Timer Description by Function Mode 9.2.PPG Timer Function 9.2.6.Timer Control Registers (TMCR, TMCR2), Status Control Register (STC), Status Control Clear Register (STCC), and Status Control Set Register (STCS) When the PPG Timer Is Selected</td><td>The function should be deleted as indicated by the shading below.

(Error)
Note:
−During timer operation of I/O mode 6 or I/O mode 4, if the falling edge is output from the even-numbered channel, this bit of the odd-numbered channel is cleared to "0".

[bit0] STRG: Software trigger bit

(Correct)
[bit0] STRG: Software Trigger Bit</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER40: Base Timer
9.Base Timer Description by Function Mode
9.3.Reload Timer Function
9.3.3.Timer Control Registers (TMCR, TMCR2), Status Control Register (STC), Status Control Clear Register (STCC), and Status Control Set Register (STCS) When the Reload Timer Is Selected</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[TMCR2:bit8,bit14:12] CKS3 to CKS0: Count clock selection bits
■These bits select a count clock for the 16-bit down counter.
■Any modification to the count clock is reflected immediately after the setting is changed. Therefore, modify CKS3 to CKS0 while counting is stopped (CTEN="0"). However, note that the bits can be modified at the same time as "1" is written to the CTEN bit.
<table>
<tr><th>CKS3</th><th>CKS2</th><th>CKS1</th><th>CKS0</th><th>Description</th></tr>
</table>
(Correct)
[TMCR2:bit8,bit14:12] CKS[3:0]: Count Clock Selection Bits
■These bits select a count clock for the 16-bit down counter.
■Any modification to the count clock is reflected immediately after the setting is changed. Therefore, modify CKS3 to CKS0 while counting is stopped (CTEN= 0). However, note that the bits can be modified at the same time as "1" is written to the CTEN bit.
<table>
<tr><th>Bits</th><th>Description</th></tr>
</table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER40: Base Timer 9.Base Timer Description by Function Mode 9.3.Reload Timer Function 9.3.3.Timer Control Registers (TMCR, TMCR2), Status Control Register (STC), Status Control Clear Register (STCC), and Status Control Set Register (STCS) When the Reload Timer Is Selected</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit9:8] EGS1, EGS0: Trigger input edge and gate function level selection bits
■Trigger input selected (TMCR2:GATE="0")
−These bits select the valid edge for an input waveform as an external start factor, and they set trigger conditions.
−For the initial value or the "00" setting, no valid edge is selected for an input waveform, so no external waveform causes a start.
■Gate function selected (TMCR2:GATE="1")
−These bits select a valid level corresponding to an input waveform as an external count factor, and the down counter counts down only while the selected level is valid.
Note:
−If "1" is written to the STRG bit, the software trigger becomes valid regardless of the EGS1 and EGS0 settings.

■Modify EGS1 and EGS0 while counting is stopped (CTEN="0"). However, note that the bits can be modified at the same time as "1" is written to the CTEN bit.
<table>
<tr><th rowspan="2">EGS1</th><th rowspan="2">EGS0</th><th colspan="2">Description</th></tr>
<tr><th>Trigger Input Selected (TMCR2:GATE="0")</th><th>Gate Function Selected (TMCR2:GATE="1")</th></tr>
</table>

(Correct)
[bit9:8] EGS[1:0]: Trigger Input Edge and Gate Function Level Selection Bits
■Trigger input selected (TMCR2:GATE= 0)
−These bits select the valid edge for an input waveform as an external start factor, and they set trigger conditions.
−For the initial value or the 0b00 setting, no valid edge is selected for an input waveform, so no external waveform causes a start.
■Gate function selected (TMCR2:GATE= 1)
−These bits select a valid level corresponding to an input waveform as an external count factor, and the down counter counts down only while the selected level is valid.
Note:
−If "1" is written to the STRG bit, the software trigger becomes valid regardless of the EGS1 and EGS0 settings.

■Modify EGS1 and EGS0 while counting is stopped (CTEN= 0). However, note that the bits can be modified at the same time as "1" is written to the CTEN bit.
<table>
<tr><th rowspan="2">Bits</th><th colspan="2">Description</th></tr>
<tr><th>Trigger Input Selected (TMCR2:GATE= 0)</th><th>Gate Function Selected (TMCR2:GATE= 1)</th></tr>
</table>
</td></tr>
</table>

Section	Change Results
CHAPTER40: Base Timer 9.Base Timer Description by Function Mode 9.3.Reload Timer Function 9.3.3.Timer Control Registers (TMCR, TMCR2), Status Control Register (STC), Status Control Clear Register (STCC), and Status Control Set Register (STCS) When the Reload Timer Is Selected	Features of "For details" should be corrected as indicated by the shading below. (Error) ■If the reload timer function is selected with "011" set in the FMD2 to FMD0 bits, setting the T32 bit to "1" results in operation in 32-bit timer mode. (Correct) ■If the reload timer function is selected with 0b011 set in the FMD2 to FMD0 bits, setting the T32 bit to "1" results in operation in 32-bit timer mode.

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER40: Base Timer
9.Base Timer Description by Function Mode
9.3.Reload Timer Function
9.3.3.Timer Control Registers (TMCR, TMCR2), Status Control Register (STC), Status Control Clear Register (STCC), and Status Control Set Register (STCS) When the Reload Timer Is Selected</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit6:4] FMD2 to FMD0: Timer function selection bits
■These bits select the timer function.
■If "011" is set in the FMD2 to FMD0 bits, the reload timer function is selected.
■Modify these bits while the timer is stopped (CTEN="0"). However, note that the bits can be modified at the same time as "1" is written to the CTEN bit.
<table><tr><th>FMD2</th><th>FMD1</th><th>FMD0</th><th>Description</th></tr></table>
(Correct)
[bit6:4] FMD[2:0]: Timer Function Selection Bits
■These bits select the timer function.
■If 0b011 is set in the FMD2 to FMD0 bits, the reload timer function is selected.
■Modify these bits while the timer is stopped (CTEN= 0). However, note that the bits can be modified at the same time as "1" is written to the CTEN bit.
<table><tr><th>Bits</th><th>Description</th></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER40: Base Timer 9.Base Timer Description by Function Mode 9.3.Reload Timer Function 9.3.3.Timer Control Registers (TMCR, TMCR2), Status Control Register (STC), Status Control Clear Register (STCC), and Status Control Set Register (STCS) When the Reload Timer Is Selected</td><td>The function should be deleted as indicated by the shading below.

(Error)
■If "0" is written to this bit when counter operation is enabled (CTEN bit is "1"), the counter stops.
<table>
<tr><th rowspan="2">bit</th><th colspan="2">Description</th></tr>
<tr><th>Read</th><th>Write</th></tr>
<tr><td>0</td><td>Stop operation.</td><td>Set this bit to “0”.</td></tr>
<tr><td>1</td><td>Enable operation.</td><td>Set this bit to “1”.</td></tr>
</table>
Note:
−During timer operation of I/O mode 6 or I/O mode 4, if the falling edge is output from the even-numbered channel, this bit of the odd-numbered channel is cleared to "0".

(Correct)
■If "0" is written to this bit when counter operation is enabled (CTEN bit is "1"), the counter stops.
<table>
<tr><th rowspan="2">Bit</th><th colspan="2">Description</th></tr>
<tr><th>Read</th><th>Write</th></tr>
<tr><td>0</td><td>Stop operation.</td><td>Set this bit to "0".</td></tr>
<tr><td>1</td><td>Enable operation.</td><td>Set this bit to "1".</td></tr>
</table></td></tr>
<tr><td>CHAPTER40: Base Timer 9.Base Timer Description by Function Mode 9.4.PWC Timer Function 9.4.1.Operations of PWC Timer</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
Single measurement and continuous measurement
The modes for pulse width measurement are a mode where measurement is 1-time only and a mode where measurement is continuous. Each mode is selected with the MDSE bit in TMCR. (See "Selection of operation mode.") The modes differ as described below.

(Correct)
Single Measurement and Continuous Measurement
The modes for pulse width measurement are a mode where measurement is 1-time only and a mode where measurement is continuous. Each mode is selected with the MDSE bit in TMCR. (See "Selection of Operation .") The modes differ as described below.</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER40: Base Timer 9.Base Timer Description by Function Mode 9.4.PWC Timer Function 9.4.2.Timer Control Registers (TMCR, TMCR2), Status Control Register (STC), Status Control Clear Register (STCC), and Status Control Set Register (STCS)When the PWC Timer Is Selected</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[TMCR2:bit8,bit14:12] CKS3 to CKS0: Count clock selection bits
■These bits select a count clock for the 16-bit down counter.
■Any modification to the count clock is reflected immediately after the setting is changed. Therefore, modify CKS3 to CKS0 while counting is stopped (CTEN="0"). However, note that the bits can be modified at the same time as "1" is written to the CTEN bit.
<table>
<tr><th>CKS3</th><th>CKS2</th><th>CKS1</th><th>CKS0</th><th>Description</th></tr>
</table>

(Correct)
[TMCR2:bit8,bit14:12] CKS[3:0]: Count Clock Selection Bits
■These bits select a count clock for the 16-bit down counter.
■Any modification to the count clock is reflected immediately after the setting is changed. Therefore, modify CKS3 to CKS0 while counting is stopped (CTEN= 0). However, note that the bits can be modified at the same time as "1" is written to the CTEN bit.
<table>
<tr><th>Bits</th><th>Description</th></tr>
</table>
</td></tr>
<tr><td>CHAPTER40: Base Timer 9.Base Timer Description by Function Mode 9.4.PWC Timer Function 9.4.2.Timer Control Registers (TMCR, TMCR2), Status Control Register (STC), Status Control Clear Register (STCC), and Status Control Set Register (STCS)When the PWC Timer Is Selected</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit10:8] EGS2 to EGS0: Measurement edge selection bits
■These bits set a measurement edge condition.
■Modify EGS2 to EGS0 while counting is stopped (CTEN="0"). However, note that the bits can be modified at the same time as "1" is written to the CTEN bit.
<table>
<tr><th>EGS2</th><th>EGS1</th><th>EGS0</th><th>Description</th></tr>
</table>

(Correct)
[bit10:8] EGS[2:0]: Measurement Edge Selection Bits
■These bits set a measurement edge condition.
■Modify EGS2 to EGS0 while counting is stopped (CTEN= 0). However, note that the bits can be modified at the same time as "1" is written to the CTEN bit.
<table>
<tr><th>Bits</th><th>Description</th></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER40: Base Timer
9.Base Timer Description by Function Mode
9.4.PWC Timer Function
9.4.2.Timer Control Registers (TMCR, TMCR2), Status Control Register (STC), Status Control Clear Register (STCC), and Status Control Set Register (STCS)When the PWC Timer Is Selected</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
■If the PWC function is selected with "100" set in the FMD2 to FMD0 bits, setting the T32 bit to "1" results in operation in 32-bit PWC mode.

(Correct)
■If the PWC function is selected with 0b100 set in the FMD2 to FMD0 bits, setting the T32 bit to "1" results in operation in 32-bit PWC mode.</td></tr>
<tr><td>CHAPTER40: Base Timer
9.Base Timer Description by Function Mode
9.4.PWC Timer Function
9.4.2.Timer Control Registers (TMCR, TMCR2), Status Control Register (STC), Status Control Clear Register (STCC), and Status Control Set Register (STCS)When the PWC Timer Is Selected</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit6:4] FMD2 to FMD0: Timer function selection bits
■These bits select the timer function.
■If "100" is set in the FMD2 to FMD0 bits, the PWC timer function is selected.
■Modify these bits while the timer is stopped (CTEN="0"). However, note that the bits can be modified at the same time as "1" is written to the CTEN bit.
<table><tr><th>FMD2</th><th>FMD1</th><th>FMD0</th><th>Description</th></tr></table>
(Correct)
[bit6:4] FMD[2:0]: Timer Function Selection Bits
■These bits select the timer function.
■If 0b100 is set in the FMD2 to FMD0 bits, the PWC timer function is selected.
■Modify these bits while the timer is stopped (CTEN= 0). However, note that the bits can be modified at the same time as "1" is written to the CTEN bit.
<table><tr><th>Bits</th><th>Description</th></tr></table></td></tr>
</table>

Section	Change Results
CHAPTER40: Base Timer 9.Base Timer Description by Function Mode 9.4.PWC Timer Function 9.4.2.Timer Control Registers (TMCR, TMCR2), Status Control Register (STC), Status Control Clear Register (STCC), and Status Control Set Register (STCS)When the PWC Timer Is Selected	The function should be deleted as indicated by the shading below. (Error) Note: −During timer operation of I/O mode 6 or I/O mode 4, if the falling edge is output from the even-numbered channel, this bit of the odd-numbered channel is cleared to "0". [bit0] Reserved: Reserved bit The read value is "0". If writing to this bit, write "0". (Correct) [bit0] Reserved: Reserved Bit The read value is "0". If writing to this bit, write "0".
CHAPTER 41: Base Timer I/O Selection Function 4.Registers	The item should be added as indicated by the shading below. (Error) Table 4-2 Register Map (Channel No.: 0) (12) (24) (Correct) Tables 4.7, 4.8, and 4.9 show the register map of each mode. The "***/***/***/***" expression in each table corresponds to the reload/PMW/PPG/PWC timer, respectively. Table 4-2 Register Map (Channel No.: 0) (12) (24)

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER 41: Base Timer I/O Selection Function 4.Registers</td><td>The item should be added as indicated by the shading below.

(Error)
<table><tr><td>0x0000_0018</td><td>Reserved
00000000_00000000</td><td>Reserved
00000000</td><td>BTxx_STCS
00000000</td></tr></table>* The initial value is XXXXXXXX_XXXXXXXX only during reload timer operation.

Table 4-4 Register Map (Channel No.: 2, 4, 6, 8, and 10) (14, 16, 18, 20, and 22) (26, 28, 30, 32, and 34)

(Correct)
<table><tr><td>0x0000_0018</td><td>Reserved
00000000_00000000</td><td>Reserved
00000000</td><td>BTxx_STCS
00000000</td></tr><tr><td>0x0000_001C</td><td>Reserved
00000000_00000000</td><td colspan="2">Reserved/BTxx_PSDR/Reserved/Reserved
00000000_00000000</td></tr><tr><td>0x0000_0020</td><td>Reserved
00000000_00000000</td><td colspan="2">Reserved/BTxx_ADTR/Reserved/Reserved
00000000_00000000</td></tr></table>The initial value is XXXXXXXX_XXXXXXXX only during reload timer operation.

Table 4-4 Register Map (Channel No.: 2, 4, 6, 8, and 10) (14, 16, 18, 20, and 22) (26, 28, 30, 32, and 34)</td></tr>
<tr><td>CHAPTER 41: Base Timer I/O Selection Function 4.Registers 4.1.I/O Selection Registers (BT_BTSEL01, BT_BTSEL23, BT_BTSEL45, BT_BTSEL67, BT_BTSEL_89, BT_BTSEL1011)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit3:0] BTSEL01[3:0]: I/O mode selection bits
These bits set the I/O modes of the 2 channels of base timer channel m and channel n for the following connection.
<table><tr><th>bit3:0</th><th>Description</th></tr></table>
(Correct)
[bit3:0] BTSEL01: I/O Mode Selection Bits
These bits set the I/O modes of the 2 channels of base timer channel m and channel n for the following connection.
<table><tr><th>Bits</th><th>Description</th></tr></table></td></tr>
</table>

Section	Change Results
CHAPTER 41: Base Timer I/O Selection Function 4.Registers 4.2.Simultaneous Soft Start Register (BT_BTSSSR0, BT_BTSSSR12, BT_SSSR24)	Features of "For details" should be corrected as indicated by the shading below. (Error) 4.2.Simultaneous Soft Start Register (BT_BTSSSR0, BT_BTSSSR12, BT_SSSR24) (Correct) 4.2.Simultaneous Soft Start Register (BT_BTSSSR0, BT_BTSSSR12, BT_BTSSSR24)
CHAPTER 41: Base Timer I/O Selection Function 4.Registers 4.2.Simultaneous Soft Start Register (BT_BTSSSR0, BT_BTSSSR12, BT_SSSR24)	Features of "For details" should be corrected as indicated by the shading below. (Error) [bit11:0] SSSR11 to SSSR0: Simultaneous soft start bits These bits represent input signals in I/O modes 5 and 6. For details on connections, see the block diagram of each I/O mode in "3. Explanation of Operation." (Correct) [bit11:0] SSSR[11:0]: Simultaneous Soft Start Bits These bits represent input signals in I/O modes 5 and 6. For details on connections, see the block diagram of each I/O mode in "3. Explanation of Operation."
CHAPTER 41: Base Timer I/O Selection Function 4.Registers 4.2.Simultaneous Soft Start Register (BT_BTSSSR0, BT_BTSSSR12, BT_SSSR24)	The item should be added as indicated by the shading below. (Error) Writing "1" starts the corresponding channel, and writing "0" has no effect. Up to 12 channels with channel numbers 0 to 11 can be started simultaneously. (Correct) Writing "1" starts the corresponding channel, and writing "0" has no effect. Up to 12 channels with channel numbers 0 to 11 can be started simultaneously. Correspondence of the channel number is below. BT_BTSSSR0 : SSSR0-11 = ch0-11 (cperi0) BT_BTSSSR12 : SSSR0-11 = ch12-23 (cperi1) BT_BTSSSR24 : SSSR0-11 = ch24-35 (cperi2)

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER 41: Base Timer I/O Selection Function 4.Registers 4.2.Simultaneous Soft Start Register (BT_BTSSSR0, BT_BTSSSR12, BT_SSSR24)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
<table><tr><th>bit [X]</th><th>Description</th></tr></table>
(Correct)
<table><tr><th>SSSR [X]</th><th>Description</th></tr></table></td></tr>
<tr><td>CHAPTER 41: Base Timer I/O Selection Function 4.Registers 4.3.TOUT Read Register (BT_BTTRR0, BT_BTTRR12, BT_BTTRR24)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit11:0] TRR11 to TRR0: Simultaneous TOUT read bits
These bits represent TOUT status of the Base Timer channels.. For details on connections, see the block diagram of each I/O mode in 3. Explanation of Operation.
<table><tr><th>bit [X]</th><th>Description</th></tr></table>
(Correct)
[bit11:0] TRR[11:0]: Simultaneous TOUT Read Bits
These bits represent TOUT status of the Base Timer channels.. For details on connections, see the block diagram of each I/O mode in 3. Explanation of Operation.
<table><tr><th>TRR[X]</th><th>Description</th></tr></table></td></tr>
</table>

Section	Change Results
CHAPTER42: 32-bit Free-run Timer 2.Explanation of the 32-bit Free-run Timer Operation	Features of "For details" should be corrected as indicated by the shading below. (Error) ■An interrupt request flag can be masked by setting the interrupt mask selection bits (MSI2 to MSI0: bit12 to bit10) of the timer state control register (TCCS). The interrupt mask selection bits (MSI2 to MSI0: bit12 to bit10) are the 3-bit reload down register that reloads the value when the mask count value reaches "000B". The mask count value can also be loaded by directly writing data to the interrupt mask selection bits (MSI2 to MSI0: bit12 to bit10). The number of mask counts is the value set in the interrupt mask selection bits (MSI2 to MSI0: bit12 to bit10). An interrupt request flag is not masked when the mask count value (MSI2 to MSI0: bit12 to bit10) becomes "000B". (Correct) ■An interrupt request flag can be masked by setting the interrupt mask selection bits (MSI2 to MSI0: bit12 to bit10) of the timer state control register (TCCS). The interrupt mask selection bits (MSI2 to MSI0: bit12 to bit10) are the 3-bit reload down register that reloads the value when the mask count value reaches 0x000. The mask count value can also be loaded by directly writing data to the interrupt mask selection bits (MSI2 to MSI0: bit12 to bit10). The number of mask counts is the value set in the interrupt mask selection bits (MSI2 to MSI0: bit12 to bit10). An interrupt request flag is not masked when the mask count value (MSI2 to MSI0: bit12 to bit10) becomes 0x000.

Section	Change Results
CHAPTER42: 32-bit Free-run Timer 3.Registers of the 32-bit Free-run Timer 3.2.Timer Data Register (TCDT)	Features of "For details" should be corrected as indicated by the shading below. (Error) The counter is cleared to the count value "00000000" as soon as any of the following factors occurs. ■Hardware reset ■The timer clear bit (SCLR: bit4) of the timer state control register (TCCS) is "1" while the free-run timer is operating (STOP: bit6=0 of the timer state control register (TCCS)). ■The value of the compare clear register (CPCLR) matches the timer count value in the up count mode (MODE: bit5=0 of the timer state control register (TCCS)). Notes: −The free-run timer is not cleared to "00000000H" even if the timer clear bit (SCLR: bit4) of the timer state control register (TCCS) is set to "1" while the free-run timer is stopped (STOP: bit6=1 of the timer state control register (TCCS)). (Correct) The counter is cleared to the count value 0x00000000 as soon as any of the following factors occurs. ■Hardware reset ■The timer clear bit (SCLR: bit4) of the timer state control register (TCCS) is "1" while the free-run timer is operating (STOP: bit6=0 of the timer state control register (TCCS)). ■The value of the compare clear register (CPCLR) matches the timer count value in the up count mode (MODE: bit5=0 of the timer state control register (TCCS)). Notes: −The free-run timer is not cleared to 0x00000000 even if the timer clear bit (SCLR: bit4) of the timer state control register (TCCS) is set to "1" while the free-run timer is stopped (STOP: bit6=1 of the timer state control register (TCCS)).

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER42: 32-bit Free-run Timer
3.Registers of the 32-bit Free-run Timer
3.3.Timer State Control Register (TCCS)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit12:10] MSI2 to MSI0: Interrupt mask selection bits
For MODE2: bit11=0 of the timer extended state control register (TECCS)
■In the up count mode (MODE: bit5=0), these bits are used to set the number of times to mask the compare clear interrupt flag. In the up/down count mode bit (MODE: bit5=1), these bits are used to set the number of times to mask the 0 detection interrupt flag.
■When these bits are set to "000B", the interrupt flags are not masked.
For MODE2: bit11=1 of the timer extended state control register (TECCS)
■In the up/down count mode (MODE: bit5=1), these bits are used to set the number of times to mask the 0 detection interrupt flag.
■In the up count mode (MODE:bit5=0), it is prohibited to use these bits to make this setting.
<table>
<tr><th colspan="3">Bit</th><th rowspan="2">Description</th></tr>
<tr><th>MSI2</th><th>MSI1</th><th>MSI0</th></tr>
</table>

(Correct)
(Error)
[bit12:10] MSI[2:0]: Interrupt Mask Selection Bits
For MODE2: bit11=0 of the timer extended state control register (TECCS)
■In the up count mode (MODE: bit5=0), these bits are used to set the number of times to mask the compare clear interrupt flag. In the up/down count mode bit (MODE: bit5=1), these bits are used to set the number of times to mask the 0 detection interrupt flag.
■When these bits are set to 0b000, the interrupt flags are not masked.
For MODE2: bit11=1 of the timer extended state control register (TECCS)
■In the up/down count mode (MODE: bit5=1), these bits are used to set the number of times to mask the 0 detection interrupt flag.
■In the up count mode (MODE:bit5=0), it is prohibited to use these bits to make this setting.
<table>
<tr><th>Bits</th><th>Description</th></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER42: 32-bit Free-run Timer
3.Registers of the 32-bit Free-run Timer
3.3.Timer State Control Register (TCCS)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit3:0] CLK3 to CLK0: Clock frequency selection bits
These bits are used to select the count clock frequency of the free-run timer.
The clock frequency is changed immediately upon setting these bits.
<table>
<tr><th colspan="4">bit</th><th colspan="6">Description</th></tr>
<tr><th>CLK3</th><th>CLK2</th><th>CLK1</th><th>CLK0</th><th>Count Clock</th><th>ϕ = 40 MHz</th><th>ϕ = 20 MHz</th><th>ϕ = 10 MHz</th><th>ϕ = 5 MHz</th><th>ϕ = 2.5 MHz</th></tr>
</table>

(Correct)
[bit3:0] CLK[3:0]: Clock Frequency Selection Bits
These bits are used to select the count clock frequency of the free-run timer.
The clock frequency is changed immediately upon setting these bits.
<table>
<tr><th rowspan="2">Bits</th><th colspan="6">Description</th></tr>
<tr><th>Count Clock</th><th>ϕ = 40 MHz</th><th>ϕ = 20 MHz</th><th>ϕ = 10 MHz</th><th>ϕ = 5 MHz</th><th>ϕ = 2.5 MHz</th></tr>
</table>
</td></tr>
<tr><td>CHAPTER42: 32-bit Free-run Timer
3.Registers of the 32-bit Free-run Timer
3.4.Timer Extended State Control Register (TECCS)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit10:8] MSI5 to MSI3: Compare clear interrupt mask selection bits
■These bits are valid only when the interrupt mask mode bit 2 is "1" (MODE2: bit11=1) and the free-run timer is in the up/down count mode (MODE: bit5=1 of the timer state control register (TCCS)). These bits are used to set the number of times to mask the compare clear interrupt flag.
The number of times to mask the 0 detection interrupt flag is set by MSI2 to MSI0: bit12 to bit10 of the timer state control register (TCCS).
■The compare clear interrupt flag is not masked when these bits are set to "000B".
<table>
<tr><th colspan="3">bit</th><th rowspan="2">Description</th></tr>
<tr><th>MSI5</th><th>MSI4</th><th>MSI3</th></tr>
</table>

(Correct)
[bit10:8] MSI[5:3]: Compare Clear Interrupt Mask Selection Bits
■These bits are valid only when the interrupt mask mode bit 2 is "1" (MODE2: bit11=1) and the free-run timer is in the up/down count mode (MODE: bit5=1 of the timer state control register (TCCS)). These bits are used to set the number of times to mask the compare clear interrupt flag.
The number of times to mask the 0 detection interrupt flag is set by MSI2 to MSI0: bit12 to bit10 of the timer state control register (TCCS).
■The compare clear interrupt flag is not masked when these bits are set to 0b000.
<table>
<tr><th>Bits</th><th>Description</th></tr>
</table>
</td></tr>
</table>

Section	Change Results
CHAPTER42: 32-bit Free-run Timer 4.Precautions for Using This Device	Features of "For details" should be corrected as indicated by the shading below. (Error) −When the timer count mode bit (MODE: bit5 of the timer state control register (TCCS)) is in the up/down count mode (MODE: bit5=1), the count value is counted up from "00000000H" to "FFFFFFFFH", and this up count operation is repeated. The 0 detection interrupt flag and the compare clear flag are set to "1" when the count value becomes 0. (Correct) −When the timer count mode bit (MODE: bit5 of the timer state control register (TCCS)) is in the up/down count mode (MODE: bit5=1), the count value is counted up from 0x00000000 to 0xFFFFFFFF, and this up count operation is repeated. The 0 detection interrupt flag and the compare clear flag are set to "1" when the count value becomes 0.
CHAPTER43: 32-bit Input Capture 2.Explanation of 32-bit Input Capture Operation 2.1.Interrupt of the 32-bit Input Capture	Features of "For details" should be corrected as indicated by the shading below. (Error) When an interrupt request is set to enabled (ICE1, ICE0: bit5, bit4 = "11B" in the input capture state control register (ICS)) in this state, then the interrupt request is output to the interrupt controller. (Correct) When an interrupt request is set to enabled (ICE1, ICE0: bit5, bit4 = 0b11 in the input capture state control register (ICS)) in this state, then the interrupt request is output to the interrupt controller.
CHAPTER43: 32-bit Input Capture 3.Registers of the 32-bit Input Capture 3.1.Input Capture Data Registers 0, 1 (IPCP0, IPCP1)	Features of "For details" should be corrected as indicated by the shading below. (Error) [bit31:0] CP31 to CP00: Input capture data value bit The input capture data register 0 (IPCP0) is used to store the value of the free-run timer when a valid edge of the external input pin IN0 signal is detected. (Correct) [bit31:0] CP[31:00]: Input Capture Data Value Bit The input capture data register 0 (IPCP0) is used to store the value of the free-run timer when a valid edge of the external input pin IN0 signal is detected.

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER43: 32-bit Input Capture 3.Registers of the 32-bit Input Capture 3.1.Input Capture Data Registers 0, 1 (IPCP0, IPCP1)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit31:0] CP31 to CP00: Input capture data value bit
Input capture data register 1 (IPCP1) is used to store the value of the free-run timer when a valid edge of the external input pin IN1 signal is detected.
The free-run timer that is mentioned here indicates the operating state of the free-run timer that is connected to the input capture.

(Correct)
[bit31:0] CP[31:00]: Input Capture Data Value Bit
Input capture data register 1 (IPCP1) is used to store the value of the free-run timer when a valid edge of the external input pin IN1 signal is detected.
The free-run timer that is mentioned here indicates the operating state of the free-run timer that is connected to the input capture.</td></tr>
<tr><td>CHAPTER43: 32-bit Input Capture 3.Registers of the 32-bit Input Capture 3.2.Input Capture State Control Register (ICS)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)<table><tr><th colspan="2">bit</th><th rowspan="2">Description</th></tr><tr><th>EG11</th><th>EG10</th></tr></table>
(Correct)<table><tr><th colspan="2">Bits</th><th rowspan="2">Description</th></tr><tr><th>EG11</th><th>EG11</th></tr></table></td></tr>
<tr><td>CHAPTER43: 32-bit Input Capture 3.Registers of the 32-bit Input Capture 3.2.Input Capture State Control Register (ICS)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)<table><tr><th colspan="2">bit</th><th rowspan="2">Description</th></tr><tr><th>EG01</th><th>EG01</th></tr></table>
(Correct)<table><tr><th colspan="2">Bits</th><th rowspan="2">Description</th></tr><tr><th>EG01</th><th>EG00</th></tr></table></td></tr>
</table>

Section	Change Results
CHAPTER45: 32-bit Reload Timer 3.Operation of the 32-bit Reload Timer	Features of "For details" should be corrected as indicated by the shading below. (Error) This section describes the operation of the 32-bit Reload Timer. 3.1Internal clock and external event counter operations of 32-bit Reload Timer 3.2Underflow operation of 32-bit Reload Timer 3.3Output functions of 32-bit Reload Timer 3.4Counter operation state 3.5DMA operation (Correct) This section describes the operation of the 32-bit Reload Timer. CHAPTER 28:7.1 Internal Clock and External Event Counter Operations of 32-bit Reload Timer 7.2 Underflow Operation of 32-bit Reload Timer 7.3 Output Functions of 32-bit Reload Timer 7.4 Counter Operation 7.5 DMA
CHAPTER45: 32-bit Reload Timer 4.Registers 4.2.Timer Control Status Register （RLTn_TMCSR）	The item should be added as indicated by the shading below. (Error) When the processor leaves debug state or DBGE is set to ’0’, the timer counter operation is resumed. (Correct) When the processor leaves debug state or DBGE is set to "0", the timer counter operation is resumed. −This bit is enable at ☆from ch. 0 to ch. 3 of CPER#2 ☆from ch.0 to ch. 1 of MCU_CONFIG_GROUP.

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER45: 32-bit Reload Timer 4.Registers 4.2.Timer Control Status Register （RLTn_TMCSR）</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
Table 4-2 RLTn_TMCSR:MOD[2:0] bit settings for internal clock mode (RLTn_TMCSR:CSL1 / RLTn_TMCSR:CSL2 = '00', '01', or '10')
<table>
<tr><th>MOD[2]</th><th>MOD[1]</th><th>MOD[0]</th><th>Input function</th><th>Active edge or level</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>Trigger disabled</td><td>-</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td rowspan="3">Trigger input</td><td>Rising edge</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>Falling edge</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>Both edge</td></tr>
<tr><td>1</td><td>x</td><td>0</td><td rowspan="2">Gate input</td><td>'L' level</td></tr>
<tr><td>1</td><td>x</td><td>1</td><td>'H' level</td></tr>
</table>

(Correct)
Table 4-2 RLTn_TMCSR:MOD[2:0] Bit Settings for Internal Clock Mode (RLTn_TMCSR:CSL1 / RLTn_TMCSR:CSL2 = 0b00, 0b01, or 0b10)
<table>
<tr><th>MOD[2]</th><th>MOD[1]</th><th>MOD[0]</th><th>Input Function</th><th>Active Edge or Level</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>Trigger disabled</td><td>-</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td rowspan="3">Trigger input</td><td>Rising edge</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>Falling edge</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>Both edge</td></tr>
<tr><td>1</td><td>x</td><td>0</td><td rowspan="2">Gate input</td><td>"L" level</td></tr>
<tr><td>1</td><td>x</td><td>1</td><td>"H" level</td></tr>
</table>
</td></tr>
<tr><td>CHAPTER45: 32-bit Reload Timer 4.Registers 4.2.Timer Control Status Register （RLTn_TMCSR）</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
Table 4-3 RLTn_TMCSR:MOD[2:0] bit settings for event counter mode (RLTn_TMCSR:CSL1 / RLTn_TMCSR:CSL2 = '11')

(Correct)
Table 4-3 RLTn_TMCSR:MOD[2:0] Bit Settings for Event Counter Mode (RLTn_TMCSR:CSL1 / RLTn_TMCSR:CSL2 = 0b11)
</td></tr>
<tr><td>CHAPTER47: QPRC(Quadrature Position/Revolution Counter)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
CHAPTER 1:QUAD POSITION & REVOLUTION COUNTER
This chapter explains the functions and operations of the Quad Position & Revolution Counter (QPRC).

(Correct)
CHAPTER47: QPRC(Quadrature Position/Revolution Counter)
This chapter explains the functions and operations of the QPRC(Quadrature Position/Revolution Counter).
</td></tr>
</table>

Section	Change Results
CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 1.Overview	Features of "For details" should be corrected as indicated by the shading below. (Error) The Quad Position & Revolution Counter is used to measure the position of Position Encoder. Also, it can be used as an up/down counter by its setting. The Quad Position & Revolution Counter contains a 16-bit position counter, a 16-bit revolution counter, two 16-bit compare registers, a control register, and its control circuit. Features of Quad Position & Revolution Counter (Correct) The QPRC(Quadrature Position/Revolution Counter) is used to measure the position of Position Encoder. Also, it can be used as an up/down counter by its setting. The QPRC(Quadrature Position/Revolution Counter) contains a 16-bit position counter, a 16-bit revolution counter, two 16-bit compare registers, a control register, and its control circuit. Features of QPRC(Quadrature Position/Revolution Counter)
CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 2.Configuration	Features of "For details" should be corrected as indicated by the shading below. (Error) The following shows the configuration of Quad Position & Revolution Counter. Figure 2-1 Block diagram of Quad Position & Revolution Counter (Correct) The following shows the configuration of QPRC(Quadrature Position/Revolution Counter). Figure 2-1 Block Diagram of QPRC(Quadrature Position/Revolution Counter)
CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 3.Operations	Features of "For details" should be corrected as indicated by the shading below. (Error) This section explains the operation of Quad Position & Revolution Counter. (Correct) This section explains the operation of QPRC(Quadrature Position/Revolution Counter).

Section	Change Results
CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 3.Operations 3.1.Operation of Position Counter	Features of "For details" should be corrected as indicated by the shading below. (Error) Operation of position counter The position counter receives an input signal from AIN or BIN external pin as an event of count clock, and increments or decrements the counter. As listed in Table 3-1, the position counter can select a counting mode by setting of the position counter mode bits (QCR:PCM[1:0]) of a control register. The counting conditions depend on the selected count mode. (Correct) 3.1.Operation of Position Counter The position counter receives an input signal from AIN or BIN external pin as an event of count clock, and increments or decrements the counter. As listed in Table 3-1, the position counter can select a counting mode by setting of the position counter mode bits(QCR.PCM[1:0]) of a control register. The counting conditions depend on the selected count mode.
CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 3.Operations 3.1.Operation of Position Counter	Features of "For details" should be corrected as indicated by the shading below. (Error) ■If the ZIN function is set to the count clear function (QCR:CGSC="0") ■If the ZIN function is set to the Gate function (QCR:CGSC="1"), the ZIN low-level detection (QCR:CGE[1:0]="01") is set, and the ZIN is logical low ■If the ZIN function is set to the Gate function (QCR:CGSC="1"), the ZIN high-level detection (QCR:CGE[1:0]="10") is set, and the ZIN is logical high (Correct) −If the ZIN function is set to the count clear function(QCR.CGSC= 0) −If the ZIN function is set to the Gate function(QCR.CGSC= 1), the ZIN "L" level detection(QCR.CGE[1:0]= 0b01) is set, and the ZIN is "L" −If the ZIN function is set to the Gate function(QCR.CGSC= 1), the ZIN "H" level detection(QCR.CGE[1:0]= 0b10) is set, and the ZIN is "H"

Section	Change Results
CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 3.Operations 3.1.Operation of Position Counter	Features of "For details" should be corrected as indicated by the shading below. (Error) If the ZIN function is set to the Gate function (QCR:CGSC="1") and if a level other than ZIN high- or low-level detection (QCR:CGE[1:0]="00" or "11") is set, the position counter is not counted up or down. Also, if the AIN and BIN configuration is swapped by SWAP bits of a control register, the AIN and BIN pins are swapped and the position counter is counted up or down. For example, if PC_Mode1 (QCR:PCM[1:0]="01") and AES[1:0]="10" (rising edge) and BES[1:0]="01" (falling edge) are set, the following occurs. ■If QCR:SWAP="0" and when a rising edge of AIN signal is detected, the position counter is counted up. When a falling edge of BIN signal is detected, the position counter is counted down. ■If QCR:SWAP="1", the position counter is counted down at a falling edge of AIN signal but it is counted up at a rising edge of BIN signal. (Correct) If the ZIN function is set to the Gate function(QCR.CGSC= 1) and if a level other than ZIN "H" or "L" level detection(QCR.CGE[1:0]= 0b00 or 0b11) is set, the position counter is not counted up or down. Also, if the AIN and BIN configuration is swapped by SWAP bits of a control register, the AIN and BIN pins are swapped and the position counter is counted up or down. For example, if PC_Mode1(QCR.PCM[1:0]= 0b01) and QCR.AES[1:0]= 0b10(rising edge) and QCR.BES[1:0]= 0b01(falling edge) are set, the following occurs. −If QCR.SWAP= 0 and when a rising edge of AIN signal is detected, the position counter is counted up. When a falling edge of BIN signal is detected, the position counter is counted down. −If QCR.SWAP= 1, the position counter is counted down at a falling edge of AIN signal but it is counted up at a rising edge of BIN signal.

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 3.Operations 3.1.Operation of Position Counter</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
<table>
<tr><th>Position count mode (PC_MODE)</th><th>AIN counting conditions</th><th>BIN counting conditions</th></tr>
<tr><td>Count disable
PC_Mode0:QCR:PCM[1:0]="00"</td><td>Position counter disable</td><td>Position counter disable</td></tr>
<tr><td>Up/down counting
PC_Mode1: QCR:PCM[1:0]="01"</td><td>AIN Active edge</td><td>BIN Active edge</td></tr>
<tr><td>Phase difference count
PC_Mode2:QCR:PCM[1:0]="10"</td><td>AIN Active edge or high/low level</td><td>High/low level or BIN active edge</td></tr>
<tr><td>Counting with direction
PC_Mode3:QCR:PCM[1:0]="11"</td><td>High/low level</td><td>BIN Active edge</td></tr>
</table>

(Correct)
<table>
<tr><th>Position Count Mode(PC_MODE)</th><th>AIN Counting Conditions</th><th>BIN Counting Conditions</th></tr>
<tr><td>Count disable
PC_Mode0:QCR.PCM[1:0]= 0b00</td><td>Position counter disable</td><td>Position counter disable</td></tr>
<tr><td>Up/down counting
PC_Mode1:QCR.PCM[1:0]= 0b01</td><td>AIN Active edge</td><td>BIN Active edge</td></tr>
<tr><td>Phase difference count
PC_Mode2:QCR.PCM[1:0]= 0b10</td><td>AIN Active edge or "H"/"L" level</td><td>"H"/"L" level or BIN active edge</td></tr>
<tr><td>Counting with direction
PC_Mode3:QCR.PCM[1:0]= 0b11</td><td>"H"/"L" level</td><td>BIN Active edge</td></tr>
</table>
</td></tr>
<tr><td>CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 3.Operations 3.1.Operation of Position Counter</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
Note:
−The active edge of AIN signal and the active edge of BIN signal mean a rising edge, a falling edge, or both of the respective signal if they are set by the AIN Detection Edge Select bits (QCR:AES[1:0]="01" or "10" or "11") or by the BIN Detection Edge Select bits (QCR:BES[1:0]="01" or "10" or "11").

(Correct)
Note:
−The active edge of AIN signal and the active edge of BIN signal mean a rising edge, a falling edge, or both of the respective signal if they are set by the AIN Detection Edge Select bits(QCR.AES[1:0]= 0b01 or 0b10 or 0b11) or by the BIN Detection Edge Select bits(QCR.BES[1:0]= 0b01 or 0b10 or 0b11).
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 3.Operations 3.1.Operation of Position Counter</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
Figure 3-1 Operations in up/down count mode
(QCR:AES[1:0]="10", QCR:BES[1:0]="10", QCR:SWAP="0")

(Correct)
Figure 3-1 Operations in Up/Down Count Mode
(QCR.AES[1:0]=0b10, QCR.BES[1:0]=0b10, QCR.SWAP= 0)</td></tr>
<tr><td>CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 3.Operations 3.1.Operation of Position Counter</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
<table><tr><th>Frequency multiplication mode</th><th>AES[1:0] setting</th><th>BES[1:0] setting</th></tr></table>
(Correct)
<table><tr><th>Frequency Multiplication Mode</th><th>QCR.AES[1:0]</th><th>QCR.BES[1:0]</th></tr></table></td></tr>
<tr><td>CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 3.Operations 3.1.Operation of Position Counter</td><td>The function should be deleted as indicated by the shading below.

(Error)
<table><tr><td rowspan="4">1-time frequency multiplication mode</td><td>01</td><td>00</td></tr><tr><td>10</td><td>00</td></tr><tr><td>00</td><td>01</td></tr><tr><td>00</td><td>10</td></tr><tr><td rowspan="2">2-time frequency multiplication mode</td><td>11</td><td>00</td></tr><tr><td>00</td><td>11</td></tr><tr><td>4-time frequency multiplication mode</td><td>11</td><td>11</td></tr></table>
(Correct)
<table><tr><td rowspan="2">2-time frequency multiplication mode</td><td>11</td><td>00</td></tr><tr><td>00</td><td>11</td></tr><tr><td>4-time frequency multiplication mode</td><td>11</td><td>11</td></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 3.Operations 3.1.Operation of Position Counter</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
Table 3-3 Counting in 2-time frequency multiplication mode
(QCR:AES[1:0]="00", QCR:BES[1:0]="11", QCR:SWAP="0")
<table>
<tr><th>Edge detection pin</th><th>Detection edge</th><th>Level Check pin</th><th>Input level</th><th>Counting direction</th><th>Figure 3-2 Timing</th></tr>
<tr><td rowspan="4">BIN</td><td>Rising edge</td><td rowspan="4">AIN</td><td>High</td><td>Up</td><td>(1)</td></tr>
<tr><td>Rising edge</td><td>Low</td><td>Down</td><td>(2)</td></tr>
<tr><td>Falling edge</td><td>High</td><td>Down</td><td>(3)</td></tr>
<tr><td>Falling edge</td><td>Low</td><td>Up</td><td>(4)</td></tr>
</table>

(Correct)
Table 3-3 Counting in 2-Time Frequency Multiplication Mode
(QCR.AES[1:0]=0b00, QCR.BES[1:0]=0b11, QCR.SWAP= 0)
<table>
<tr><th>Edge Detection Pin</th><th>Detection Edge</th><th>Level Check Pin</th><th>Input Level</th><th>Counting Direction</th><th>Figure 3-2 Timing</th></tr>
<tr><td rowspan="4">BIN</td><td>Rising edge</td><td rowspan="4">AIN</td><td>"H"</td><td>Up</td><td>(1)</td></tr>
<tr><td>Rising edge</td><td>"L"</td><td>Down</td><td>(2)</td></tr>
<tr><td>Falling edge</td><td>"H"</td><td>Down</td><td>(3)</td></tr>
<tr><td>Falling edge</td><td>"L"</td><td>Up</td><td>(4)</td></tr>
</table>
</td></tr>
<tr><td>CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 3.Operations 3.1.Operation of Position Counter</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
Figure 3-2 Operation in 2-time frequency multiplication mode
(QCR:AES[1:0]="00", QCR:BES[1:0]="11", QCR:SWAP="0")

(Correct)
Figure 3-2 Operation in 2-Time Frequency Multiplication Mode
(QCR.AES[1:0]=0b00, QCR.BES[1:0]=0b11, QCR.SWAP= 0)
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 3.Operations 3.1.Operation of Position Counter</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
Table 3-4 Counting in 4-time frequency multiplication mode
(QCR:AES[1:0]="11", QCR:BES[1:0]="11")
<table>
<tr><th>Edge detection pin</th><th>Detection edge</th><th>Level Check pin</th><th>Input level</th><th>Counting direction</th><th>Figure 3-3 Timing</th></tr>
<tr><td rowspan="4">BIN</td><td>Rising edge</td><td rowspan="4">AIN</td><td>High</td><td>Up</td><td>(1)</td></tr>
<tr><td>Rising edge</td><td>Low</td><td>Down</td><td>(2)</td></tr>
<tr><td>Falling edge</td><td>High</td><td>Down</td><td>(3)</td></tr>
<tr><td>Falling edge</td><td>Low</td><td>Up</td><td>(4)</td></tr>
<tr><td rowspan="4">AIN</td><td>Rising edge</td><td rowspan="4">BIN</td><td>High</td><td>Down</td><td>(5)</td></tr>
<tr><td>Rising edge</td><td>Low</td><td>Up</td><td>(6)</td></tr>
<tr><td>Falling edge</td><td>High</td><td>Up</td><td>(7)</td></tr>
<tr><td>Falling edge</td><td>Low</td><td>Down</td><td>(8)</td></tr>
</table>

(Correct)
Table 3-4 Counting in 4-Time Frequency Multiplication Mode
(QCR.AES[1:0]=0b11, QCR.BES[1:0]=0b11)
<table>
<tr><th>Edge Detection Pin</th><th>Detection Edge</th><th>Level Check Pin</th><th>Input Level</th><th>Counting Direction</th><th>Figure 3-3 Timing</th></tr>
<tr><td rowspan="4">BIN</td><td>Rising edge</td><td rowspan="4">AIN</td><td>"H"</td><td>Up</td><td>(1)</td></tr>
<tr><td>Rising edge</td><td>"L"</td><td>Down</td><td>(2)</td></tr>
<tr><td>Falling edge</td><td>"H"</td><td>Down</td><td>(3)</td></tr>
<tr><td>Falling edge</td><td>"L"</td><td>Up</td><td>(4)</td></tr>
<tr><td rowspan="4">AIN</td><td>Rising edge</td><td rowspan="4">BIN</td><td>"H"</td><td>Down</td><td>(5)</td></tr>
<tr><td>Rising edge</td><td>"L"</td><td>Up</td><td>(6)</td></tr>
<tr><td>Falling edge</td><td>"H"</td><td>Up</td><td>(7)</td></tr>
<tr><td>Falling edge</td><td>"L"</td><td>Down</td><td>(8)</td></tr>
</table>
</td></tr>
<tr><td>CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 3.Operations 3.1.Operation of Position Counter</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
Figure 3-3 Operation in 4-time frequency multiplication mode
(QCR:AES[1:0]="11", QCR:BES[1:0]="11", QCR:SWAP="0")

(Correct)
Figure 3-3 Operation in 4-Time Frequency Multiplication Mode
(QCR.AES[1:0]=0b11, QCR.BES[1:0]=0b11, QCR.SWAP= 0)
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 3.Operations 3.1.Operation of Position Counter</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
<table>
<tr><th>Edge detection pin</th><th>Detection edge</th><th>Level Check pin</th><th>Input level</th><th>Counting direction</th><th>Figure 3-4 Timing</th></tr>
<tr><td rowspan="2">BIN</td><td>Active edge</td><td rowspan="2">AIN</td><td>High</td><td>Up</td><td>(1)</td></tr>
<tr><td>Active edge</td><td>Low</td><td>Down</td><td>(2)</td></tr>
</table>
(Correct)
<table>
<tr><th>Edge Detection Pin</th><th>Detection Edge</th><th>Level Check Pin</th><th>Input Level</th><th>Counting Direction</th><th>Figure 3-4 Timing</th></tr>
<tr><td rowspan="2">BIN</td><td>Active edge</td><td rowspan="2">AIN</td><td>"H"</td><td>Up</td><td>(1)</td></tr>
<tr><td>Active edge</td><td>"L"</td><td>Down</td><td>(2)</td></tr>
</table>
</td></tr>
<tr><td>CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 3.Operations 3.1.Operation of Position Counter</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
Figure 3-4 Operation in the direction control counting mode
(QCR:AES[1:0]="00", QCR:BES[1:0]="10", QCR:SWAP="0")

(Correct)
Figure 3-4 Operation in the Direction Control Counting Mode
(QCR.AES[1:0]=0b00, QCR.BES[1:0]=0b10, QCR.SWAP= 0)
</td></tr>
<tr><td>CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 3.Operations 3.2.Operation of Revolution Counter</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
Operation of revolution counter

(Correct)
3.2.Operation of Revolution Counter
</td></tr>
</table>

Section	Change Results
CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 3.Operations 3.2.Operation of Revolution Counter	Features of "For details" should be corrected as indicated by the shading below. (Error) RC_Mode0 (QCR:RCM[1:0]="00"): ■The revolution counter is disabled. ■When the ZIN signal is used for counter clear function (QCR:CGSC="0"), the active edge of ZIN signal is reset. Also, the position counter is reset when this counter overflows. (Correct) RC_Mode0(QCR.RCM[1:0]= 0b00) −The revolution counter is disabled. −If the ZIN signal is used for counter clear function(QCR.CGSC= 0), the position counter is reset at an active edge of the ZIN signal. Also, the position counter is reset when this counter overflows.
CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 3.Operations 3.2.Operation of Revolution Counter	Features of "For details" should be corrected as indicated by the shading below. (Error) RC_Mode1 (QCR:RCM[1:0]="01"): ■When ZIN signal is used for the counter clear function (QCR:CGSC="0"), the revolution counter is operated only an active edge of ZIN signal (but an input from the position counter is ignored). (Correct) RC_Mode1(QCR.RCM[1:0]= 0b01) −When ZIN signal is used for the counter clear function(QCR.CGSC= 0), the revolution counter is operated only an active edge of ZIN signal(but an input from the position counter is ignored).
CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 3.Operations 3.2.Operation of Revolution Counter	Features of "For details" should be corrected as indicated by the shading below. (Error) RC_Mode2 (QCR:RCM[1:0]="10"): ■The revolution counter is counted up or down only by the output value of position counter. (Correct) RC_Mode2(QCR.RCM[1:0]= 0b10) −The revolution counter is counted up or down only by the output value of position counter.

Section	Change Results
CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 3.Operations 3.2.Operation of Revolution Counter	Features of "For details" should be corrected as indicated by the shading below. (Error) RC_Mode3 (QCR:RCM[1:0]="11"): ■In this mode, the revolution counter is operated with an output of position counter and when the ZIN signal is used for the counter clear function (QCR:CGSC="0"), the revolution counter is also counted up or down at an active edge of ZIN signal. (Correct) RC_Mode3(QCR.RCM[1:0]= 0b11) −In this mode, the revolution counter is operated with an output of position counter and when the ZIN signal is used for the counter clear function(QCR.CGSC= 0), the revolution counter is also counted up or down at an active edge of ZIN signal.
CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 3.Operations 3.3. Absolute Value of Positions	Features of "For details" should be corrected as indicated by the shading below. (Error) Absolute value of positions (Correct) 3.3.Absolute Value of Positions
CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 3.Operations 3.4.QPRC(Quadrature Position/Revolution Counter) Interrupts	Features of "For details" should be corrected as indicated by the shading below. (Error) Quad Position & Revolution Counter interrupts The following table defines the conditions where an interrupt request of Quad Position & Revolution Counter can generate. (Correct) 3.4.QPRC(Quadrature Position/Revolution Counter) Interrupts The following table defines the conditions where an interrupt request of QPRC(Quadrature Position/Revolution Counter) can generate.

Section	Change Results
CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 3.Operations 3.4.QPRC(Quadrature Position/Revolution Counter) Interrupts	Features of "For details" should be corrected as indicated by the shading below. (Error) Table 3-6 Generation conditions of Quad Position & Revolution Counter interrupt requests (Correct) Table 3-6 Generation Conditions of QPRC(Quadrature Position/Revolution Counter) Interrupt Requests

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 3.Operations 3.4.QPRC(Quadrature Position/Revolution Counter) Interrupts</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
<table>
<tr><th>Interrupt request</th><th>Interrupt request flag</th><th>Interrupt request is enabled if</th><th>Interrupt request is cleared if</th></tr>
<tr><td>Count inversion interrupt request</td><td>QICR: CDCF="1"</td><td>QICR: CDCIE="1"</td><td>QICR:CDCF is set to "0".</td></tr>
<tr><td>Zero index interrupt request</td><td>QICR: ZIIF="1"</td><td rowspan="3">QICR: OUZIE="1"</td><td>QICR:ZIIF is set to "0".</td></tr>
<tr><td>Overflow interrupt request</td><td>QICR: OFDF="1"</td><td>QICR:OFDF is set to "0".</td></tr>
<tr><td>Underflow interrupt request</td><td>QICR: UFDF="1"</td><td>QICR:UFDF is set to "0".</td></tr>
<tr><td>PC and RC match interrupt request</td><td>QICR: QPRCMF="1"</td><td>QICR: QPRCMIE="1"</td><td>QICR.QPRCMF is set to "0".</td></tr>
<tr><td>PC match interrupt request</td><td>QICR: QPCMF="1"</td><td>QICR: QPCMIE="1"</td><td>QICR:QPCMF is set to "0".</td></tr>
<tr><td>PC match and RC match interrupt request</td><td>QICR: QPCNRCMF="1"</td><td>QICR: QPCNRCMIE="1"</td><td>QICR:QPCNRCMF is set to "0".</td></tr>
<tr><td>Outrange interrupt request</td><td>QECR: ORNGF="1"</td><td>QICR: ORNGIE="1"</td><td>QECR:QRNGF is set to "0".</td></tr>
</table>

(Correct)
<table>
<tr><th>Interrupt Request</th><th>Interrupt Request Flag</th><th>Interrupt Request is Enabled If</th><th>Interrupt Request is Cleared If</th></tr>
<tr><td>Count inversion interrupt request</td><td>QICR.CDCF= 1</td><td>QICR.CDCIE= 1</td><td>"0" is written to QICR.CDCF</td></tr>
<tr><td>Zero index interrupt request</td><td>QICR.ZIIF= 1</td><td rowspan="3">QICR.OUZIE= 1</td><td>"0" is written to QICR.ZIIF</td></tr>
<tr><td>Overflow interrupt request</td><td>QICR.OFDF= 1</td><td>"0" is written to QICR.OFDF</td></tr>
<tr><td>Underflow interrupt request</td><td>QICR.UFDF= 1</td><td>"0" is written to QICR.UFDF</td></tr>
<tr><td>PC and RC match interrupt request</td><td>QICR.QPRCMF= 1</td><td>QICR.QPRCMIE= 1</td><td>"0" is written to QICR.QPRCMF</td></tr>
<tr><td>PC match interrupt request</td><td>QICR.QPCMF= 1</td><td>QICR.QPCMIE= 1</td><td>"0" is written to QICR.QPCMF</td></tr>
<tr><td>PC match and RC match interrupt request</td><td>QICR.QPCNRCMF= 1</td><td>QICR.QPCNRCMIE= 1</td><td>"0" is written to QICR.QPCNRCMF</td></tr>
<tr><td>Outrange interrupt request</td><td>QECR.ORNGF= 1</td><td>QICR.ORNGIE= 1</td><td>"0" is written to QECR.QRNGF</td></tr>
</table>
</td></tr>
</table>

Section	Change Results
CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 3.Operations 3.5.Operation Example of QPRC Maximum Position Register(QMPR) Interrupt	Features of "For details" should be corrected as indicated by the shading below. (Error) Operation example of QPRC Maximum Position Register (QMPR) interrupt (Correct) 3.5.Operation Example of QPRC Maximum Position Register(QMPR) Interrupt
CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 3.Operations 3.5.Operation Example of QPRC Maximum Position Register(QMPR) Interrupt	Features of "For details" should be corrected as indicated by the shading below. (Error) ■When the position counter is counted up in RC_Mode0 (QCR:RCM[1:0]="00"), RC_Mode2 (QCR:RCM[1:0]="10") or RC_Mode3 (QCR:RCM[1:0]="11"), the overflow flag (QICR:OFDF) is set to "1" and the position counter is reset. ■When the position counter is counted up in RC_Mode1 (QCR:RCM[1:0]="01"), the overflow flag (QICR:OFDF) is set to "1". During this time, the position counter is not reset but is counted up. The following gives an operation example where the QPRC Maximum Position Register (QMPR) is used in RC_Mode2 (QCR:RCM[1:0]="10"). (Correct) −When the position counter is counted up in RC_Mode0(QCR.RCM[1:0]= 0b00), RC_Mode2(QCR.RCM[1:0]= 0b10) or RC_Mode3(QCR.RCM[1:0]= 0b11), the overflow flag(QICR.OFDF) is set to "1" and the position counter is reset. −When the position counter is counted up in RC_Mode1(QCR.RCM[1:0]= 0b01), the overflow flag(QICR.OFDF) is set to "1". During this time, the position counter is not reset but is counted up. The following gives an operation example where the QPRC Maximum Position Register(QMPR) is used in RC_Mode2(QCR.RCM[1:0]= 0b10).

Section	Change Results
CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 3.Operations 3.5.Operation Example of QPRC Maximum Position Register(QMPR) Interrupt	Features of "For details" should be corrected as indicated by the shading below. (Error) During counting up When the position counter maximum value overflows to "0x0000", the revolution counter is counted up. During this time, the overflow flag (QICRL:OFDF) is set to logical 1. (Correct) During counting up: When the position counter maximum value overflows to 0x0000, the revolution counter is counted up. During this time, the overflow flag(QICR.OFDF) is set to "1".
CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 3.Operations 3.6.Position Counter Reset Mask Function	Features of "For details" should be corrected as indicated by the shading below. (Error) Position counter reset mask function The position counter reset mask function can be used only when RC_Mode0 (QCR:RCM[1:0]="00") or RC_Mode3 (QCR:RCM[1:0]="11") is selected. This function operates regardless of setting of the position counter mode (PC_Mode1, PC_Mode2 or PC_Mode3). (Correct) 3.6.Position Counter Reset Mask Function The position counter reset mask function can be used only when RC_Mode0(QCR.RCM[1:0]= -b00) or RC_Mode3(QCR.RCM[1:0]= 0b11) is selected. This function operates regardless of setting of the position counter mode(PC_Mode1, PC_Mode2 or PC_Mode3).
CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 3.Operations 3.6.Position Counter Reset Mask Function	Features of "For details" should be corrected as indicated by the shading below. (Error) The following gives an operation example where the position counter reset mask function is used in RC_Mode3 (QCR:RCM[1:0]="11"). Example 1: (Correct) The following gives an operation example where the position counter reset mask function is used in RC_Mode3(QCR.RCM[1:0]=0b11). Example 1:

Section	Change Results
CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 3.Operations 3.6.Position Counter Reset Mask Function	Features of "For details" should be corrected as indicated by the shading below. (Error) Figure 3-9 Position counter reset mask operation example 1 (QMPR=99, QCR:PCRM[1:0]="10", QCR:CGSC="0") (Correct) Figure 3-9 Position Counter Reset Mask Operation Example 1 (QMPR=99, QCR.PCRM[1:0]= 0b10, QCR.CGSC= 0)
CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 3.Operations 3.6.Position Counter Reset Mask Function	Features of "For details" should be corrected as indicated by the shading below. (Error) Example 2: An active edge of ZIN signal is ignored for four (4) counts(QPRC=99-96) of position counter after count inversion of position counter. Figure 3-10 Position counter reset mask operation example 2 (QMPR=99, QCR:PCRM[1:0]="10", QCR:CGSC="0") (Correct) Example 2: An active edge of ZIN signal is ignored for four(4) counts(QPRC=99 to 96) of position counter after occurrence of position counter underflow following count inversion of position counter. Figure 3-10 Position Counter Reset Mask Operation Example 2 (QMPR=99, QCR.PCRM[1:0]=0b10, QCR.CGSC= 0)

Section	Change Results
CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 3.Operations 3.6.Position Counter Reset Mask Function	Features of "For details" should be corrected as indicated by the shading below. (Error) The following gives an operation example where the position counter reset mask function is used in RC_Mode0 (QCR:RCM[1:0]="00"). Example 3: An active edge of ZIN signal is ignored for four (4) counts(QPRC=0-3) of position counter after occurrence of position counter overflow if the revolution counter is disabled. Figure 3-11 Position counter reset mask operation example 3 (QMPR=99, QCR:PCRM[1:0]="10", QCR:CGSC="0") (Correct) The following gives an operation example where the position counter reset mask function is used in RC_Mode0(QCR.RCM[1:0]= 0b00). Example 3: An active edge of ZIN signal is ignored for four(4) counts(QPRC=0 to 3) of position counter after occurrence of position counter overflow if the revolution counter is disabled. Figure 3-11 Position Counter Reset Mask Operation Example 3 (QMPR=99, QCR.PCRM[1:0]= 0b10, QCR.CGSC= 0)

Section	Change Results
CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 3.Operations 3.6.Position Counter Reset Mask Function	Features of "For details" should be corrected as indicated by the shading below. (Error) Notes: −While the position counter reset mask function is operating, the mask function is released and the position counter can be reset in the following conditions. −When the position counter mode bit (QCR:PCM[1:0]) is changed −When the revolution counter mode bit (QCR:RCM[1:0]) is changed −When the direction of the position counter is changed −Even if an overflow or underflow of the position counter occurs without inversion of the position counter while the position counter reset mask function is operating in RC_Mode0 (QCR:RCM[1:0]="00") or RC_Mode3 (QCR:RCM[1:0]="11"), the revolution counter is not counted up or down. However, if an overflow occurs, the position counter becomes "0". If an underflow occurs, the QMPR is reloaded to the position counter. The overflow interrupt request flag bit (QICR:OFDF) or the underflow interrupt request flag bit (QICR:UFDF) is set to "1". (Correct) Notes: −While the position counter reset mask function is operating, the mask function is released and the position counter can be reset in the following conditions. −The position counter mode bit(QCR.PCM[1:0]) is changed, or −The revolution counter mode bit(QCR.RCM[1:0]) is changed, or −The direction of the position counter is changed. −Even if an overflow or underflow of the position counter occurs without inversion of the position counter while the position counter reset mask function is operating in RC_Mode0(QCR.RCM[1:0]= 0b00) or RC_Mode3(QCR.RCM[1:0]= 0b11), the revolution counter is not counted up or down. However, if an overflow occurs, the position counter becomes 0x0000. If an underflow occurs, the QMPR is reloaded to the position counter. The overflow interrupt request flag bit(QICR.OFDF) or the underflow interrupt request flag bit(QICR.UFDF) is set to "1".
CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 4.Registers	Features of "For details" should be corrected as indicated by the shading below. (Error) This section explains the configuration and functions of the registers used for the Quad Position & Revolution Counter (QPRC). (Correct) This section explains the configuration and functions of the registers used for the QPRC(Quadrature Position/Revolution Counter).

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 4.Registers</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
List of Quad Position & Revolution Counter registers
<table>
<tr><th>Abbreviation</th><th>Register name</th><th>See</th></tr>
<tr><td>QPCR</td><td>Quad Position & Revolution Counter Position Count Register</td><td>4.1</td></tr>
<tr><td>QRCR</td><td>QPRC Revolution Count Register</td><td>4.2</td></tr>
</table>

(Correct)
Table 4-1 List of QPRC(Quadrature Position/Revolution Counter) Register
<table>
<tr><th>Abbreviation</th><th>Register name</th><th>See</th></tr>
<tr><td>QPCR</td><td>QPRC Position Count Register</td><td>4.1</td></tr>
<tr><td>QRCR</td><td>QPRC Revolution Count Register</td><td>4.2</td></tr>
</table>
</td></tr>
<tr><td>CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 4.Registers 4.1.Quad Position & Revolution Counter Position Count Register (QPCR)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
4.1.Quad Position & Revolution Counter Position Count Register (QPCR)

(Correct)
4.1.Position Count Register(QPCR)
</td></tr>
</table>

Section	Change Results
CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 4.Registers 4.1.Quad Position & Revolution Counter Position Count Register (QPCR)	Features of "For details" should be corrected as indicated by the shading below. (Error) [bit 15:0] QPCR: Reading this register reads out the current value of the position counter. While the position counter stops counting (QCR:PSTP="1"), the count value can be written to this register. This register is set to 0x0000 in the following one of conditions. ■Reset ■A ZIN active edge is detected in the following conditions. −The ZIN function is set to the counter clear function (QCR:CGSC="0") in RC_Mode1 (QCR:RCM[1:0]="01"). −After the position counter has been incremented or decremented by the mask set value when the count inversion of the position counter is not detected where the ZIN function is set to the counter clear function (QCR:CGSC="0") and the reset mask function of the position counter is valid (QCR:PCRM[1:0]="01" or "10" or "11") in RC_Mode0 (QCR:RCM[1:0]="00") or RC_Mode3 (QCR:RCM[1:0]="11") −The ZIN function is set to the counter clear function (QCR:CGSC="0") and the reset mask function of the position counter is invalid (QCR:PCRM[1:0]="00") in RC_Mode0(QCR:RCM[1:0]="00") or RC_Mode3(QCR:RCM[1:0]="11"). (Correct) [bit15:0] QPCR[15:0]: Reading this register reads out the current value of the position counter. While the position counter stops counting(QCR.PSTP= 1), the count value can be written to this register. This register is set to 0x0000 in the following one of conditions. −Reset −A ZIN active edge is detected in the following conditions. −The ZIN function is set to the counter clear function(QCR.CGSC= 0) in RC_Mode1(QCR.RCM[1:0]= 0b01). −After the position counter has been incremented or decremented by the mask set value when the count inversion of the position counter is not detected where the ZIN function is set to the counter clear function(QCR.CGSC= 0) and the reset mask function of the position counter is valid(QCR.PCRM[1:0]= 0b01 or 0b10 or 0b11) in RC_Mode0(QCR.RCM[1:0]= 0b00) or RC_Mode3(QCR.RCM[1:0]= 0b11) −The ZIN function is set to the counter clear function(QCR.CGSC= 0) and the reset mask function of the position counter is invalid(QCR.PCRM[1:0]= 0b00) in RC_Mode0(QCR.RCM[1:0]= 0b00) or RC_Mode3(QCR.RCM[1:0]= 0b11).

Section	Change Results
CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 4.Registers 4.1.Quad Position & Revolution Counter Position Count Register (QPCR)	Features of "For details" should be corrected as indicated by the shading below. (Error) ■A position counter overflow is detected in the following conditions. −RC_Mode2(QCR:RCM[1:0]="10") −After the position counter has been incremented or decremented by the mask set value when the count inversion of the position counter is not detected where the ZIN function is set to the counter clear function (QCR:CGSC="0") and the reset mask function of the position counter is valid (QCR:PCRM[1:0]="01" or "10" or "11") in RC_Mode0 (QCR:RCM[1:0]="00") or RC_Mode3 (QCR:RCM[1:0]="11") −The ZIN function is set to the counter clear function (QCR:CGSC="0") and the reset mask function of the position counter is invalid (QCR:PCRM[1:0]="00") in RC_Mode0(QCR:RCM[1:0]="00") or RC_Mode3(QCR:RCM[1:0]="11"). ■0x0000 is written to this QPCR while the position counter is under suspension (QCR:PSTP="1"). The value of the QPRC Maximum Position Register (QMPR) is set to this register in the following condition. ■A position counter underflow is detected. (Correct) −A position counter overflow is detected in the following conditions. −RC_Mode2(QCR.RCM[1:0]= 0b10) −After the position counter has been incremented or decremented by the mask set value when the count inversion of the position counter is not detected where the ZIN function is set to the counter clear function(QCR.CGSC= 0) and the reset mask function of the position counter is valid(QCR.PCRM[1:0]= 0b01 or 0b10 or 0b11) in RC_Mode0(QCR.RCM[1:0]= 0b00) or RC_Mode3(QCR.RCM[1:0]= 0b11) −The ZIN function is set to the counter clear function(QCR.CGSC= 0) and the reset mask function of the position counter is invalid(QCR.PCRM[1:0]= 0b00) in RC_Mode0(QCR.RCM[1:0]= 0b00) or RC_Mode3(QCR.RCM[1:0]= 0b11). −0x0000 is written to this QPCR while the position counter is under suspension(QCR.PSTP= 1). The value of the QPRC Maximum Position Register(QMPR) is set to this register in the following condition. −A position counter underflow is detected.

Section	Change Results
CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 4.Registers 4.1.Quad Position & Revolution Counter Position Count Register (QPCR)	Features of "For details" should be corrected as indicated by the shading below. (Error) Notes: −Do not access the Quad Position & Revolution Counter Position Count Register (QPCR) with a byte access instruction. −After the count value was written to the Quad Position & Revolution Counter Position Count Register (QPCR) while the position counter was under suspension (QCR:PSTP="1") in RC_Mode0 (QCR:RCM[1:0]="00"), RC_Mode1 (QCR:RCM[1:0]="01"), or RC_Mode3(QCR:RCM[1:0]="11"), if a ZIN active edge is detected with the count function (QCR:CGSC="0"), the Quad Position & Revolution Counter Position Count Register (QPCR) will be set to 0x0000. To write the count value to the Quad Position & Revolution Counter Position Count Register (QPCR), make the ZIN detection edge invalid (QCR:CGE[1:0]="00") before writing it to the QPCR. (Correct) Notes: −Do not access the QPRC Position Count Register(QPCR) with a byte access instruction. −After the count value was written to the QPRC Position Count Register(QPCR) while the position counter was under suspension(QCR.PSTP= 1) in RC_Mode0(QCR.RCM[1:0]= 0b00), RC_Mode1(QCR.RCM[1:0]= 0b01), or RC_Mode3(QCR.RCM[1:0]= 0b11), if a ZIN active edge is detected with the count function(QCR.CGSC= 0), the QPRC Position Count Register(QPCR) will be set to 0x0000. To write the count value to the QPRC Position Count Register(QPCR), make the ZIN detection edge invalid(QCR.CGE[1:0]= 0b00) before writing it to the QPCR.
CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 4.Registers 4.2.QPRC Revolution Count Register (QRCR)	Features of "For details" should be corrected as indicated by the shading below. (Error) [bit 15:0] QRCR: Reading this register reads out the current value of the revolution counter. While the revolution counter stops counting (QCR:RCM[1:0]="00"), the count value can be written to this register. This register is set to 0x0000 in the following one of conditions. ■Reset ■0x0000 is written to this register while the revolution counter is under suspension (QCR.RCM[1:0]="00"). (Correct) [bit15:0] QRCR[15:0] : Reading this register reads out the current value of the revolution counter. While the revolution counter stops counting(QCR.RCM[1:0]= 0b00), the count value can be written to this register. This register is set to 0x0000 in the following one of conditions. −Reset −0x0000 is written to this register while the revolution counter is under suspension(QCR.RCM[1:0]= 0b00).

Section	Change Results
CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 4.Registers 4.2.QPRC Revolution Count Register (QRCR)	Features of "For details" should be corrected as indicated by the shading below. (Error) Notes: −Do not access the QPRC Revolution Count Register (QRCR) with a byte access instruction. −As the direction of the position counter is not detected in PC_Mode0 (QCR:PCM[1:0]="00"), the last position counter direction bit (QICR:DIRPC) becomes indefinite. Therefore, if the mode is changed from PC_Mode0 (QCR:PCM[1:0]="00") to another mode, when a ZIN active edge is detected before an AIN/BIN active edge is detected, the following operations apply. −The position counter is reset if the mode is RC_Mode0 (QCR:RCM[1:0]="00"), RC_Mode1 (QCR:RCM[1:0]="01"), or RC_Mode3 (QCR:RCM[1:0]="11") −The revolution counter is not counted up or down (Correct) Notes: −Do not access the QPRC Revolution Count Register(QRCR) with a byte access instruction. −As the direction of the position counter is not detected in PC_Mode0(QCR.PCM[1:0]= 0b00), the last position counter direction bit(QICR.DIRPC) becomes indefinite. Therefore, if the mode is changed from PC_Mode0(QCR.PCM[1:0]= 0b00) to another mode, when a ZIN active edge is detected before an AIN/BIN active edge is detected, the following operations apply. −The position counter is reset if the mode is RC_Mode0(QCR.RCM[1:0]= 0b00),RC_Mode1(QCR.RCM[1:0]= 0b01), or RC_Mode3(QCR.RCM[1:0]= 0b11) −The revolution counter is not counted up or down.
CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 4.Registers 4.3.QPRC Position Counter Compare Register (QPCCR)	Features of "For details" should be corrected as indicated by the shading below. (Error) [bit 15:0] QPCCR: If the value of this register matches that of the position counter, the QPRC position counter comparison match flag (QICR:QPCMF) is set to "1". This Compare Register can be used only to compare with the count value of the position counter. (Correct) [bit15:0] QPCCR[15:0] : QPRC Position Counter Compare If the value of this register matches that of the position counter, the QPRC position counter comparison match flag(QICR.QPCMF) is set to "1". This Compare Register can be used only to compare with the count value of the position counter.

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 4.Registers 4.4.QPRC Position and Revolution Counter Compare Register (QPRCR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit 15:0] QPRCR:
Select whether to compare with the count value of the position or revolution counter using the RSEL bit of the QPRC Control Register (QCR). If the value of this register matches that of the position or revolution counter, the QPRC position and revolution counter comparison match flag (QICR:QPRCMF) is set to "1".

(Correct)
[bit15:0] QPRCR[15:0] : QPRC Position and Revolution Counter Compare
Select whether to compare with the count value of the position or revolution counter using the RSEL bit of the QPRC Control Register(QCR). If the value of this register matches that of the position or revolution counter, the QPRC position and revolution counter comparison match flag(QICR.QPRCMF) is set to "1".</td></tr>
<tr><td>CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 4.Registers 4.5.QPRC Control Register (QCR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
<table>
<tr><td>bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Field</td><td>SWAP</td><td>RSEL</td><td>CGSC</td><td>PSTP</td><td>RCM1</td><td>RCM0</td><td>PCM1</td><td>PCM0</td></tr>
<tr><td>R/W Attribute</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td></tr>
<tr><td>Protection Attribute</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>Initial value</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>

(Correct)
<table>
<tr><td>bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Field</td><td>SWAP</td><td>RSEL</td><td>CGSC</td><td>PSTP</td><td colspan="2">RCM[1:0]</td><td colspan="2">PCM[1:0]</td></tr>
<tr><td>R/W Attribute</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td colspan="2">R/W</td><td colspan="2">R/W</td></tr>
<tr><td>Protection Attribute</td><td>-</td><td>-</td><td>-</td><td>-</td><td colspan="2">-</td><td colspan="2">-</td></tr>
<tr><td>Initial value</td><td>0</td><td>0</td><td>0</td><td>0</td><td colspan="2">00</td><td colspan="2">00</td></tr>
</table>
</td></tr>
</table>

Section	Change Results
CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 4.Registers 4.5.QPRC Control Register (QCR)	Features of "For details" should be corrected as indicated by the shading below. (Error) Note: −Change the swap bit (SWAP) when the position counter is disabled (PCM[1:0]="00"). (Correct) Note: −Change the swap bit(SWAP) when the position counter is disabled(PCM[1:0]= 0b00).
CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 4.Registers 4.5.QPRC Control Register (QCR)	Features of "For details" should be corrected as indicated by the shading below. (Error) When the counter clear function is enabled (QGSC="0"), the ZIN pin clears the position counter if the revolution count mode is set to RC_Mode0 (RCM[1:0]="00"), RC_Mode1 (RCM[1:0]="01"), or RC_Mode3 (RCM[1:0]="11"). The CGE1 and CGE0 bits of the QCR register clear the position counter by selecting a valid edge of the ZIN pin and detecting the selected edge. When the gate function is enabled (QGSC="1"), the ZIN pin controls the count operation of the position counter. The CGE1 and CGE0 bits of the QCR register count the position counter at the valid level of the ZIN pin. (Correct) When the counter clear function is enabled(QGSC= 0), the ZIN pin clears the position counter if the revolution count mode is set to RC_Mode0(RCM[1:0]= 0b00), RC_Mode1(RCM[1:0]= 0b01), or RC_Mode3(RCM[1:0]= 0b11). The CGE1 and CGE0 bits of the QCR register clear the position counter by selecting a valid edge of the ZIN pin and detecting the selected edge. When the gate function is enabled(QGSC= 1), the ZIN pin controls the count operation of the position counter. The CGE1 and CGE0 bits of the QCR register count the position counter at the valid level of the ZIN pin.

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 4.Registers 4.5.QPRC Control Register (QCR)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit 3:2] RCM1, RCM0: Revolution counter mode bits
These bits are used to select the count mode of the revolution counter and the reset mode of the position counter. For the effect on the position counter, see "Operation of revolution counter".
<table>
<tr><th>bit3</th><th>bit2</th><th>Description</th></tr>
<tr><td>0</td><td>0</td><td>Disables the revolution counter (RC_Mode0).</td></tr>
<tr><td>0</td><td>1</td><td>The revolution counter is counted up or down only with a ZIN active edge (RC_Mode1).</td></tr>
<tr><td>1</td><td>0</td><td>The revolution counter is counted up or down only when overflow or underflow is detected in the position counter that matches QMPR (RC_Mode2).</td></tr>
<tr><td>1</td><td>1</td><td>The revolution counter is counted up or down in two cases: a position counter overflow or underflow is detected and a ZIN active edge is detected (RC_Mode3).</td></tr>
</table>

(Correct)
[bit3:2] RCM[1:0] : Revolution Counter Mode Bits
These bits are used to select the count mode of the revolution counter and the reset mode of the position counter. For the effect on the position counter, see "Operation of Revolution 3.2 Operation of Revolution "
<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>00</td><td>Disables the revolution counter(RC_Mode0).</td></tr>
<tr><td>01</td><td>The revolution counter is counted up or down only with a ZIN active edge(RC_Mode1).</td></tr>
<tr><td>10</td><td>The revolution counter is counted up or down only when overflow or underflow is detected in the position counter that matches QMPR(RC_Mode2).</td></tr>
<tr><td>11</td><td>The revolution counter is counted up or down in two cases: a position counter overflow or underflow is detected and a ZIN active edge is detected(RC_Mode3).</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 4.Registers 4.5.QPRC Control Register (QCR)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit 1:0] PCM1, PCM0: Position counter mode bits
These bits are used to select the count mode of the position counter.
<table>
<tr><th>bit1</th><th>bit0</th><th>Description</th></tr>
<tr><td>0</td><td>0</td><td>Disables the position counter (PC_Mode0) to stop it.</td></tr>
<tr><td>0</td><td>1</td><td>Up-down count mode (PC_Mode1)
Increments the value with an AIN active edge and decrements it with a BIN active edge.</td></tr>
<tr><td>1</td><td>0</td><td>Phase difference count mode (PC_Mode2)
Counts up if AIN is leading BIN and down if BIN is leading AIN.</td></tr>
<tr><td>1</td><td>1</td><td>Directional count mode (PC_Mode3)
Counts up or down with the BIN active edge and AIN level.</td></tr>
</table>

(Correct)
[bit1:0] PCM[1:0] : Position Counter Mode Bits
These bits are used to select the count mode of the position counter.
<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>00</td><td>Disables the position counter(PC_Mode0) to stop it.</td></tr>
<tr><td>01</td><td>Up/down count mode(PC_Mode1)
Increments the value with an AIN active edge and decrements it with a BIN active edge.</td></tr>
<tr><td>10</td><td>Phase difference count mode(PC_Mode2)
Counts up if AIN is leading BIN and down if BIN is leading AIN.</td></tr>
<tr><td>11</td><td>Directional count mode(PC_Mode3)
Counts up or down with the BIN active edge and AIN level.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 4.Registers 4.5.QPRC Control Register (QCR)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
Note:
−As the direction of the position counter is not detected in PC_Mode0 (PCM[1:0]="00"), the last position counter direction bit (QICR:DIRPC) becomes indefinite. Therefore, if the mode is changed from PC_Mode0 (PCM[1:0]="00") to another mode, when a ZIN active edge is detected before an AIN/BIN active edge is detected, the following operations apply.
−The position counter is reset if the mode is RC_Mode0 (RCM[1:0]="00"), RC_Mode1 (RCM[1:0]="01"), or RC_Mode3 (RCM[1:0]="11")
−The revolution counter is not counted up or down

(Correct)
Note:
−As the direction of the position counter is not detected in PC_Mode0(PCM[1:0]= 0b00), the last position counter direction bit(QICR.DIRPC) becomes indefinite. Therefore, if the mode is changed from PC_Mode0(PCM[1:0]= 0b00) to another mode, when a ZIN active edge is detected before an AIN/BIN active edge is detected, the following operations apply.
−The position counter is reset if the mode is RC_Mode0(RCM[1:0]= 0b00), RC_Mode1(RCM[1:0]= 0b01), or RC_Mode3(RCM[1:0]= 0b11)
−The revolution counter is not counted up or down
</td></tr>
<tr><td>CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 4.Registers 4.5.QPRC Control Register (QCR)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
<table>
<tr><td>bit</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td>Field</td><td>CGE1</td><td>CGE0</td><td>BES1</td><td>BES0</td><td>AES1</td><td>AES0</td><td>PCRM1</td><td>PCRM0</td></tr>
<tr><td>R/W Attribute</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td></tr>
<tr><td>Protection Attribute</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>Initial value</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>

(Correct)
<table>
<tr><td>bit</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td>Field</td><td colspan="2">CGE[1:0]</td><td colspan="2">BES[1:0]</td><td colspan="2">AES[1:0]</td><td colspan="2">PCRM[1:0]</td></tr>
<tr><td>R/W Attribute</td><td colspan="2">R/W</td><td colspan="2">R/W</td><td colspan="2">R/W</td><td colspan="2">R/W</td></tr>
<tr><td>Protection Attribute</td><td colspan="2">-</td><td colspan="2">-</td><td colspan="2">-</td><td colspan="2">-</td></tr>
<tr><td>Initial Value</td><td colspan="2">00</td><td colspan="2">00</td><td colspan="2">00</td><td colspan="2">00</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 4.Registers 4.5.QPRC Control Register (QCR)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit 15:14] CGE1, CGE0: Detection edge selection bits
These bits are used to select the detection edge when the ZIN external pin is used for the counter clear function (CGSC="0"). They are also used to select the detection level when the ZIN external pin is used for the gate function (CGSC="1")..
<table>
<tr><th>bit15</th><th>bit14</th><th>ZIN used for counter clear function (CGSC="0")</th><th>ZIN used for gate function (CGSC="1")</th></tr>
<tr><td>0</td><td>0</td><td>Disables edge detection.</td><td>Disables level detection.</td></tr>
<tr><td>0</td><td>1</td><td>Detects a falling edge.</td><td>Detects level "L".</td></tr>
<tr><td>1</td><td>0</td><td>Detects a rising edge.</td><td>Detects level "H".</td></tr>
<tr><td>1</td><td>1</td><td>Detects a rising or falling edge.</td><td>Disables level detection.</td></tr>
</table>

(Correct)
[bit15:14] CGE[1:0] : Detection Edge Selection Bits
These bits are used to select the detection edge when the ZIN external pin is used for the counter clear function(CGSC= 0). They are also used to select the detection level when the ZIN external pin is used for the gate function(CGSC= 1).
<table>
<tr><th>Bits</th><th>ZIN Used for Counter Clear Function (CGSC= 0)</th><th>ZIN Used for Gate Function (CGSC="1")</th></tr>
<tr><td>00</td><td>Disables edge detection.</td><td>Disables level detection.</td></tr>
<tr><td>01</td><td>Detects a falling edge.</td><td>Detects level "L".</td></tr>
<tr><td>10</td><td>Detects a rising edge.</td><td>Detects level "H".</td></tr>
<tr><td>11</td><td>Detects both rising and falling edges.</td><td>Disables level detection.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 4.Registers 4.5.QPRC Control Register (QCR)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit 13:12] BES1, BES0: BIN detection edge selection bits
These bits are used to select the detection edge of the BIN external pin.
<table>
<tr><th>bit13</th><th>bit12</th><th>Description</th></tr>
<tr><td>0</td><td>0</td><td>Disables edge detection.</td></tr>
<tr><td>0</td><td>1</td><td>Detects a falling edge.</td></tr>
<tr><td>1</td><td>0</td><td>Detects a rising edge.</td></tr>
<tr><td>1</td><td>1</td><td>Detects rising and falling edges.</td></tr>
</table>

(Correct)
[bit13:12] BES[1:0] : BIN Detection Edge Selection Bits
These bits are used to select the detection edge of the BIN external pin.
<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>00</td><td>Disables edge detection.</td></tr>
<tr><td>01</td><td>Detects a falling edge.</td></tr>
<tr><td>10</td><td>Detects a rising edge.</td></tr>
<tr><td>11</td><td>Detects both rising and falling edges.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 4.Registers 4.5.QPRC Control Register (QCR)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit 11:10] AES1, AES0: AIN detection edge selection bits
These bits are used to select the detection edge of the AIN external pin.
<table>
<tr><th>bit11</th><th>bit10</th><th>Description</th></tr>
<tr><td>0</td><td>0</td><td>Disables edge detection.</td></tr>
<tr><td>0</td><td>1</td><td>Detects a falling edge.</td></tr>
<tr><td>1</td><td>0</td><td>Detects a rising edge.</td></tr>
<tr><td>1</td><td>1</td><td>Detects rising and falling edges.</td></tr>
</table>

(Correct)
[bit11:10] AES[1:0] : AIN Detection Edge Selection Bits
These bits are used to select the detection edge of the AIN external pin.
<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>00</td><td>Disables edge detection.</td></tr>
<tr><td>01</td><td>Detects a falling edge.</td></tr>
<tr><td>10</td><td>Detects a rising edge.</td></tr>
<tr><td>11</td><td>Detects both rising and falling edges.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER47: QPRC(Quadrature Position/Revolution Counter)
4.Registers
4.5.QPRC Control Register (QCR)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit 9:8] PCRM1, PCRM0: Position counter reset mask bits
These bits are used to specify the period (mask time) to ignore the events shown below after detecting a position counter overflow or underflow or detecting a ZIN active edge.
■Position counter resetting
■Revolution counter increment or decrement

This mask function is released when the count direction of the position counter is changed, and restarts when a position counter overflow or underflow is detected or a ZIN active edge is detected.
<table>
<tr><th>bit9</th><th>bit8</th><th>Description</th></tr>
<tr><td>0</td><td>0</td><td>No reset mask</td></tr>
<tr><td>0</td><td>1</td><td>The position counter reset or a revolution counter count-up or -down events are ignored until the position counter changes twice.</td></tr>
<tr><td>1</td><td>0</td><td>The position counter reset or a revolution counter count-up or -down events are ignored until the position counter changes four times.</td></tr>
<tr><td>1</td><td>1</td><td>The position counter reset or a revolution counter count-up or -down events are ignored until the position counter changes eight times.</td></tr>
</table>

(Correct)
[bit9:8] PCRM[1:0] : Position Counter Reset Mask Bits
These bits are used to specify the period(mask time) to ignore the events shown below after detecting a position counter overflow or underflow or detecting a ZIN active edge.
−Position counter resetting
−Revolution counter increment or decrement

This mask function is released when the count direction of the position counter is changed, and restarts when a position counter overflow or underflow is detected or a ZIN active edge is detected.
<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>00</td><td>No reset mask</td></tr>
<tr><td>01</td><td>The position counter reset or a revolution counter count-up or -down events are ignored until the position counter changes twice.</td></tr>
<tr><td>10</td><td>The position counter reset or a revolution counter count-up or -down events are ignored until the position counter changes four times.</td></tr>
<tr><td>11</td><td>The position counter reset or a revolution counter count-up or -down events are ignored until the position counter changes eight times.</td></tr>
</table>
</td></tr>
</table>

Section	Change Results
CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 4.Registers 4.5.QPRC Control Register (QCR)	Features of "For details" should be corrected as indicated by the shading below. (Error) Notes: −The position counter reset mask function is available only in RC_Mode0 (RCM[1:0]="00") and RC_Mode3 (RCM[1:0]="11"). This function operates regardless of the setting of the position counter mode (PC_Mode1, PC_Mode2, or PC_Mode3). (Correct) Notes: −The position counter reset mask function is available only in RC_Mode0(RCM[1:0]= 0b00) and RC_Mode3(RCM[1:0]= 0b11). This function operates regardless of the setting of the position counter mode(PC_Mode1, PC_Mode2, or PC_Mode3).
CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 4.Registers 4.6.QPRC Extension Control Register (QECR)	Features of "For details" should be corrected as indicated by the shading below. (Error) [bit 15:3] Reserved bit (Correct) [bit15:3] Reserved : Reserved Bits These bits must always be written to "0". The read value is "0".
CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 4.Registers 4.6.QPRC Extension Control Register (QECR)	Features of "For details" should be corrected as indicated by the shading below. (Error) [bit 2] ORNGIE: Outrange interrupt enable bit This bit is used to control whether or not to issue an interrupt notification to the CPU when the outrange interrupt request flag (ORNGF) is set to "1". When this bit is set to "1", an interrupt is generated if the value of the revolution counter gets out of the range (ORNGF="1"). (Correct) [bit2] ORNGIE : Outrange Interrupt Enable Bit This bit is used to control whether or not to issue an interrupt notification to the MCU when the outrange interrupt request flag(ORNGF) is set to "1". When this bit is set to "1", an interrupt is generated if the value of the revolution counter gets out of the range(ORNGF= 1).

Section	Change Results
CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 4.Registers 4.6.QPRC Extension Control Register (QECR)	Features of "For details" should be corrected as indicated by the shading below. (Error) If a positive number is selected as the outrange mode of the revolution counter (ORNGMD="0"), this flag is set to "1" when the revolution counter changes from 0x0001 to 0x0000 after counting down or when it changes from 0xFFFE to 0xFFFF after counting up. If the 8K value is selected as the outrange mode of the revolution counter (ORNGMD="1"), this flag is set to "1" when the revolution counter changes from 0x8001 to 0x8000 after counting down or when it changes from 0x7FFE to 0x7FFF after counting up. This flag can only clear to "0" in write mode. Setting "1" has no effect. "1" is read by the read-modify-write access operation. (Correct) If a positive number is selected as the outrange mode of the revolution counter(ORNGMD= 0), this flag is set to "1" when the revolution counter changes from 0x0001 to 0x0000 after counting down or when it changes from 0xFFFE to 0xFFFF after counting up. If the 8K value is selected as the outrange mode of the revolution counter(ORNGMD= 1), this flag is set to "1" when the revolution counter changes from 0x8001 to 0x8000 after counting down or when it changes from 0x7FFE to 0x7FFF after counting up. This flag can be only cleared to "0" in write mode. Setting "1" has no effect. "1" is read by the read-modify-write access operation.
CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 4.Registers 4.7.Low-Order Bytes of QPRC Interrupt Control Register (QICRL)	Features of "For details" should be corrected as indicated by the shading below. (Error) [bit 7] ZIIF: Zero index interrupt request flag bit This flag is set to "1" when the position counter is reset by the ZIN input. This flag can only clear to "0" in write mode. Setting "1" has no effect. (Correct) [bit7] ZIIF : Zero Index Interrupt Request Flag Bit This flag is set to "1" when the position counter is reset by the ZIN input. This flag can be only cleared to "0" in write mode. Setting "1" has no effect.
CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 4.Registers 4.7.Low-Order Bytes of QPRC Interrupt Control Register (QICRL)	Features of "For details" should be corrected as indicated by the shading below. (Error) Note: −The zero index interrupt request flag bit (ZIIF) is not set to "1" even if ZIN is used as the gate function (QCR:CGSC="1") or the position counter is reset in RC_Mode2 (QCR:RCM[1:0]="10"). (Correct) Note: −The zero index interrupt request flag bit(ZIIF) is not set to "1" even if ZIN is used as the gate function(QCR.CGSC= 1) or the position counter is reset in RC_Mode2(QCR.RCM[1:0]= 0b10).

Section	Change Results
CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 4.Registers 4.7.Low-Order Bytes of QPRC Interrupt Control Register (QICRL)	Features of "For details" should be corrected as indicated by the shading below. (Error) This flag indicates that a position counter overflow occurs. When the position counter is counted up, this bit is set to "1" if the value of the position counter matches that of the QPRC Maximum Position Register (QMPR). This flag can only clear to "0" in write mode. Setting "1" has no effect. (Correct) This flag indicates that a position counter overflow occurs. When the set value of the QPRC Maximum Position Register(QMPR) matches the value of the position counter and the position counter is counted up, this bit is set to "1". This flag can be only cleared to "0" in write mode. Setting "1" has no effect.
CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 4.Registers 4.7.Low-Order Bytes of QPRC Interrupt Control Register (QICRL)	Features of "For details" should be corrected as indicated by the shading below. (Error) [bit 5] UFDF: Underflow interrupt request flag bit This flag indicates that a position counter underflow occurs. When the position counter is counted down, this bit is set to "1" if the position counter is 0x0000. This flag can only clear to "0" in write mode. Setting "1" has no effect. (Correct) [bit5] UFDF : Underflow Interrupt Request Flag Bit This flag indicates that a position counter underflow occurs. When the position counter is 0x0000 and the position counter is counted down, this bit is set to "1". This flag can be only cleared to "0" in write mode. Setting "1" has no effect.
CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 4.Registers 4.7.Low-Order Bytes of QPRC Interrupt Control Register (QICRL)	Features of "For details" should be corrected as indicated by the shading below. (Error) [bit 4] OUZIE: Overflow, underflow, or zero index interrupt enable bit This bit is used to control whether or not to issue an interrupt notification to the CPU when the overflow interrupt request flag bit (OFDF), underflow interrupt request flag bit (UFDF), or zero index interrupt request flag bit (ZIIF) is set to "1". When this bit is set to "1", an interrupt is generated if overflow is detected (OFDF="1"), underflow is detected (UFDF="1"), or zero index is detected (ZIIF="1"). (Correct) [bit4] OUZIE : Overflow, Underflow, or Zero Index Interrupt Enable Bit This bit is used to control whether or not to issue an interrupt notification to the MCU when the overflow interrupt request flag bit(OFDF), underflow interrupt request flag bit(UFDF), or zero index interrupt request flag bit(ZIIF) is set to "1". When this bit is set to "1", an interrupt is generated if overflow is detected(OFDF= 1), underflow is detected(UFDF= 1), or zero index is detected(ZIIF= 1).

Section	Change Results
CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 4.Registers 4.7.Low-Order Bytes of QPRC Interrupt Control Register (QICRL)	Features of "For details" should be corrected as indicated by the shading below. (Error) When the comparison between the position counter and QPRC Position and Revolution Counter Compare Register (QPRCR) is selected (QCR:RSEL="0"), this flag is set to "1" if the value of the position counter matches that of the QPRC Position and Revolution Counter Compare Register (QPRCR). When the comparison between the revolution counter and QPRC Position and Revolution Counter Compare Register (QPRCR) is selected (QCR:RSEL="1"), this flag is set to "1" if the value of the revolution counter matches that of the QPRC Position and Revolution Counter Compare Register (QPRCR). This flag can only clear to "0" in write mode. Setting "1" has no effect. (Correct) When the comparison between the position counter and QPRC Position and Revolution Counter Compare Register(QPRCR) is selected(QCR.RSEL= 0), this flag is set to "1" if the value of the position counter matches that of the QPRC Position and Revolution Counter Compare Register(QPRCR). When the comparison between the revolution counter and QPRC Position and Revolution Counter Compare Register(QPRCR) is selected(QCR.RSEL= 1), this flag is set to "1" if the value of the revolution counter matches that of the QPRC Position and Revolution Counter Compare Register(QPRCR). This flag can be only cleared to "0" in write mode. Setting "1" has no effect.
CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 4.Registers 4.7.Low-Order Bytes of QPRC Interrupt Control Register (QICRL)	Features of "For details" should be corrected as indicated by the shading below. (Error) Notes: −If the register function selection bit (QCR:RSEL) is set to "0", the PC and RC match interrupt request flag bit (QPRCMF) is set to "1" immediately when the following one of conditions is satisfied. −The mode is changed to PC_Mode1 (QCR:PCM[1:0]="01"), PC_Mode2 (QCR:PCM[1:0]="10"), or PC_Mode3 (QCR:PCM[1:0]="11") when the position counter is disabled (QCR:PCM[1:0]="00") and the value of the position counter matches that of the QPRC Position and Revolution Counter Compare Register (QPRCR). −The value of the position counter matches that of the QPRC Position and Revolution Counter Compare Register (QPRCR) when data is written to the Quad Position & Revolution Counter Position Count Register (QPCR) or QPRC Position and Revolution Counter Compare Register (QPRCR) in PC_Mode1 (QCR:PCM[1:0]="01"), PC_Mode2 (QCR:PCM[1:0]="10"), or PC_Mode3 (QCR:PCM[1:0]="11"). (Correct) Notes: −If the register function selection bit(QCR.RSEL) is set to "0", the PC and RC match interrupt request flag bit(QPRCMF) is set to "1" immediately when the following one of conditions is satisfied. −The mode is changed to PC_Mode1(QCR.PCM[1:0]= 0b01), PC_Mode2(QCR.PCM[1:0]= 0b10), or PC_Mode3(QCR.PCM[1:0]= 0b11) when the position counter is disabled(QCR.PCM[1:0]= 0b00) and the value of the position counter matches that of the QPRC Position and Revolution Counter Compare Register(QPRCR). −The value of the position counter matches that of the QPRC Position and Revolution Counter Compare Register(QPRCR) when data is written to the QPRC Position Count Register(QPCR) or QPRC Position and Revolution Counter Compare Register(QPRCR) in PC_Mode1(QCR.PCM[1:0]= 0b01), PC_Mode2(QCR.PCM[1:0]= 0b10), or PC_Mode3(QCR.PCM[1:0]= 0b11)

Section	Change Results
CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 4.Registers 4.7.Low-Order Bytes of QPRC Interrupt Control Register (QICRL)	Features of "For details" should be corrected as indicated by the shading below. (Error) −If the register function selection bit (QCR:RSEL) is set to "1", the PC and RC match interrupt request flag bit (QPRCMF) is set to "1" immediately when the following condition is satisfied. −The value of the revolution counter matches that of the QPRC Position and Revolution Counter Compare Register (QPRCR) by writing data to the QPRC Position and Revolution Counter Compare Register (QPRCR) when the mode is RC_Mode1 (QCR:RCM[1:0]="01"), RC_Mode2 (QCR:RCM[1:0]="10"), or RC_Mode3 (QCR:RCM[1:0]="11"). −The value of the revolution counter matches that of the QPRC Position and Revolution Counter Compare Register (QPRCR) by changing the mode from RC_Mode0 (QCR:RCM[1:0]="00") to another mode. −When the register function selection bit (QCR:RSEL) is changed, the PC and RC match interrupt request flag bit (QPRCMF) is set to "1" immediately if the following one of conditions is satisfied. −The value of the revolution counter matches that of the QPRC Position and Revolution Counter Compare Register (QPRCR) when the register function selection bit (QCR:RSEL) is changed from "0" to "1" in the mode other than RC_Mode0 (QCR.RCM[1:0]="00"). −The value of the position counter matches that of the QPRC Position and Revolution Counter Compare Register (QPRCR) when the register function selection bit (QCR:RSEL) is changed from "1" to "0" in the mode other than RC_Mode0 (QCR.RCM[1:0]="00"). (Correct) −If the register function selection bit(QCR.RSEL) is set to "1", the PC and RC match interrupt request flag bit(QPRCMF) is set to "1" immediately when the following condition is satisfied. −The value of the revolution counter matches that of the QPRC Position and Revolution Counter Compare Register(QPRCR) by writing data to the QPRC Position and Revolution Counter Compare Register(QPRCR) when the mode is RC_Mode1(QCR.RCM[1:0]= 0b01), RC_Mode2(QCR.RCM[1:0]= 0b10), or RC_Mode3(QCR.RCM[1:0]= 0b11). −The value of the revolution counter matches that of the QPRC Position and Revolution Counter Compare Register(QPRCR) by changing the mode from RC_Mode0(QCR.RCM[1:0]= 0b00) to another mode. −When the register function selection bit(QCR.RSEL) is changed, the PC and RC match interrupt request flag bit(QPRCMF) is set to "1" immediately if the following one of conditions is satisfied. −The value of the revolution counter matches that of the QPRC Position and Revolution Counter Compare Register(QPRCR) when the register function selection bit(QCR.RSEL) is changed from "0" to "1" in the mode other than RC_Mode0(QCR.RCM[1:0]= 0b00). −The value of the position counter matches that of the QPRC Position and Revolution Counter Compare Register(QPRCR) when the register function selection bit(QCR.RSEL) is changed from "1" to "0" in the mode other than RC_Mode0(QCR.RCM[1:0]="00")

Section	Change Results
CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 4.Registers 4.7.Low-Order Bytes of QPRC Interrupt Control Register (QICRL)	Features of "For details" should be corrected as indicated by the shading below. (Error) [bit 2] QPRCMIE: PC and RC match interrupt enable bit This bit is used to control whether or not to issue an interrupt notification to the CPU when the PC and RC match interrupt request flag (QPRCMF) is set to "1". When this bit is set to "1", an interrupt is generated if the value of the position or revolution counter matches that of the QPRC Position and Revolution Counter Compare Register (QPRCR) (QPRCMF="1"). (Correct) [bit2] QPRCMIE : PC and RC Match Interrupt Enable Bit This bit is used to control whether or not to issue an interrupt notification to the MCU when the PC and RC match interrupt request flag(QPRCMF) is set to "1". When this bit is set to "1", an interrupt is generated if the value of the position or revolution counter matches that of the QPRC Position and Revolution Counter Compare Register(QPRCR)(QPRCMF= 1).
CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 4.Registers 4.7.Low-Order Bytes of QPRC Interrupt Control Register (QICRL)	Features of "For details" should be corrected as indicated by the shading below. (Error) [bit 1] QPCMF: PC match interrupt request flag bit This flag indicates whether or not the value of the position counter matches that of the QPRC Position Counter Compare Register (QPCCR). This flag is set to "1" if the value of the position counter matches that of the QPRC Position Counter Compare Register (QPCCR). This flag can only clear to "0" in write mode. Setting "1" has no effect. (Correct) [bit1] QPCMF : PC Match Interrupt Request Flag Bit This flag indicates whether or not the value of the position counter matches that of the QPRC Position Counter Compare Register(QPCCR). This flag is set to "1" if the value of the position counter matches that of the QPRC Position Counter Compare Register(QPCCR). This flag can be only cleared to "0" in write mode. Setting "1" has no effect.

Section	Change Results
CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 4.Registers 4.7.Low-Order Bytes of QPRC Interrupt Control Register (QICRL)	Features of "For details" should be corrected as indicated by the shading below. (Error) Note: −The PC match interrupt request flag bit (QPCMF) is set to "1" immediately when the following one of conditions is satisfied. −The mode is changed to PC_Mode1 (QCR:PCM[1:0]="01"), PC_Mode2 (QCR:PCM[1:0]="10"), or PC_Mode3 (QCR:PCM[1:0]="11") when the position counter is disabled (QCR:PCM[1:0]="00") and the value of the position counter matches that of the QPRC Position Counter Compare Register (QPCCR). −The value of the position counter matches that of the QPRC Position Counter Compare Register (QPCCR) by writing to the Quad Position & Revolution Counter Position Count Register (QPCR) when the position counter stop bit (QCR:PSTP) is "1" and when the mode is PC_Mode1 (QCR:PCM[1:0]="01"), PC_Mode2 (QCR:PCM[1:0]="10"), or PC_Mode3 (QCR:PCM[1:0]="11"). −The value of the position counter matches that of the QPRC Position Counter Compare Register (QPCCR) by writing to the QPRC Position Counter Compare Register (QPCCR) when the mode is PC_Mode1 (QCR:PCM[1:0]="01"), PC_Mode2 (QCR:PCM[1:0]="10"), or PC_Mode3 (QCR:PCM[1:0]="11"). (Correct) Note: −The PC match interrupt request flag bit(QPCMF) is set to "1" immediately when the following one of conditions is satisfied. −The mode is changed to PC_Mode1(QCR.PCM[1:0]= 0b01), PC_Mode2(QCR.PCM[1:0]= 0b10), or PC_Mode3(QCR.PCM[1:0]= 0b11) when the position counter is disabled(QCR.PCM[1:0]= 0b00) and the value of the position counter matches that of the QPRC Position Counter Compare Register(QPCCR). −The value of the position counter matches that of the QPRC Position Counter Compare Register(QPCCR) by writing to the QPRC Position Count Register(QPCR) when the position counter stop bit(QCR.PSTP) is "1" and when the mode is PC_Mode1(QCR.PCM[1:0]= 0b01), PC_Mode2(QCR.PCM[1:0]= 0b10), or PC_Mode3(QCR.PCM[1:0]= 0b11). −The value of the position counter matches that of the QPRC Position Counter Compare Register(QPCCR) by writing to the QPRC Position Counter Compare Register(QPCCR) when the mode is PC_Mode1(QCR.PCM[1:0]= 0b01), PC_Mode2(QCR.PCM[1:0]= 0b10), or PC_Mode3(QCR.PCM[1:0]= 0b11).

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 4.Registers 4.7.Low-Order Bytes of QPRC Interrupt Control Register (QICRL)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit 0] QPCMIE: PC match interrupt enable bit
This bit is used to control whether or not to issue an interrupt notification to the CPU when the PC match interrupt request flag (QPCMF) is set to "1".
When this bit is set to "1", an interrupt is generated if the value of the position counter matches that of the QPRC Position Counter Compare Register (QPCCR) (QPCMF="1").

(Correct)
[bit0] QPCMIE : PC Match Interrupt Enable Bit
This bit is used to control whether or not to issue an interrupt notification to the MCU when the PC match interrupt request flag(QPCMF) is set to "1".
When this bit is set to "1", an interrupt is generated if the value of the position counter matches that of the QPRC Position Counter Compare Register(QPCCR)(QPCMF= 1).
</td></tr>
<tr><td>CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 4.Registers 4.8.High-Order Bytes of QPRC Interrupt Control Register (QICRH)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
<table>
<tr><td>bit</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td>Field</td><td colspan="2">Reserved</td><td>QPCNRCMF</td><td>QPCNRCMIE</td><td>DIROU</td><td>DIRPC</td><td>CDCF</td><td>CDCIE</td></tr>
<tr><td>R/W Attribute</td><td>R0,W0</td><td>R0,W0</td><td>R,W</td><td>R/W</td><td>R,WX</td><td>R,WX</td><td>R,W</td><td>R/W</td></tr>
<tr><td>Protection Attribute</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>Initial value</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
(Correct)
<table>
<tr><td>bit</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td>Field</td><td colspan="2">Reserved</td><td>QPCNRCMF</td><td>QPCNRCMIE</td><td>DIROU</td><td>DIRPC</td><td>CDCF</td><td>CDCIE</td></tr>
<tr><td>R/W Attribute</td><td colspan="2">R0,W0</td><td>R,W</td><td>R/W</td><td>R,WX</td><td>R,WX</td><td>R,W</td><td>R/W</td></tr>
<tr><td>Protection Attribute</td><td colspan="2">-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>Initial Value</td><td colspan="2">00</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
</td></tr>
</table>

Section	Change Results
CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 4.Registers 4.8.High-Order Bytes of QPRC Interrupt Control Register (QICRH)	Features of "For details" should be corrected as indicated by the shading below. (Error) [bit 15:14] Reserved (Correct) [bit15:14] Reserved : Reserved Bits These bits must always be written to "0". The read value is "0".
CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 4.Registers 4.8.High-Order Bytes of QPRC Interrupt Control Register (QICRH)	Features of "For details" should be corrected as indicated by the shading below. (Error) This flag is set to "1" when the value of the position counter matches that of the QPRC Position Counter Compare Register (QPCCR) (QPCMF="1") and the value of the revolution counter matches that of the QPRC Position and Revolution Counter Compare Register (QPRCR). This flag can only clear to "0" in write mode. Setting "1" has no effect. "1" is read by the read-modify-write access operation. (Correct) This flag is set to "1" when the value of the position counter matches that of the QPRC Position Counter Compare Register(QPCCR)(QPCMF= 1) and the value of the revolution counter matches that of the QPRC Position and Revolution Counter Compare Register(QPRCR). This flag can be only cleared to "0" in write mode. Setting "1" has no effect. "1" is read by the read-modify-write access operation.

Section	Change Results
CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 4.Registers 4.8.High-Order Bytes of QPRC Interrupt Control Register (QICRH)	Features of "For details" should be corrected as indicated by the shading below. (Error) −The mode is changed to PC_Mode1 (QCR:PCM[1:0]="01"), PC_Mode2 (QCR:PCM[1:0]="10"), or PC_Mode3 (QCR:PCM[1:0]="11") when the position counter is disabled (QCR:PCM[1:0]="00") and the revolution counter is in the mode other than RC_Mode0(QCR:RCM[1:0]="00") while the value of the position counter matches that of the QPRC Position Counter Compare Register (QPCCR) and the value of the revolution counter matches that of the QPRC Position and Revolution Counter Compare Register (QPRCR). −The value of the position counter matches that of the QPRC Position Counter Compare Register (QPCCR) when data is written to the Quad Position & Revolution Counter Position Count Register (QPCR) or QPRC Position Counter Compare Register (QPCCR) where the value of the revolution counter matches that of the QPRC Position & Revolution Counter Compare Register (QPRCR) when the mode is PC_Mode1 (QCR:PCM[1:0]="01"), PC_Mode2 (QCR:PCM[1:0]="10"), or PC_Mode3 (QCR:PCM[1:0]="11") and the revolution counter is in the mode other than RC_Mode0 (QCR:RCM[1:0]="00"). −The value of the revolution counter matches that of the QPRC Position and Revolution Counter Compare Register (QPRCR) when the data is written to the QPRC Position and Revolution Counter Compare Register (QPRCR) in the mode other than RC_Mode0 (QCR:RCM[1:0]="00") where the specified value matches that of the QPRC Position Count Register (QPCR) or QPRC Position Counter Compare Register (QPCCR) in PC_Mode1 (QCR:PCM[1:0]="01"), PC_Mode2 (QCR:PCM[1:0]="10"), or PC_Mode3 (QCR:PCM[1:0]="11"). (Correct) −The mode is changed to PC_Mode1(QCR.PCM[1:0]= 0b01), PC_Mode2(QCR.PCM[1:0]= 0b10), or PC_Mode3(QCR.PCM[1:0]= 0b11) when the position counter is disabled(QCR.PCM[1:0]= 0b00) and the revolution counter is in the mode other than RC_Mode0(QCR.RCM[1:0]= 0b00) while the value of the position counter matches that of the QPRC Position Counter Compare Register(QPCCR) and the value of the revolution counter matches that of the QPRC Position and Revolution Counter Compare Register(QPRCR). −The value of the position counter matches that of the QPRC Position Counter Compare Register(QPCCR) when data is written to the QPRC Position Count Register(QPCR) or QPRC Position Counter Compare Register(QPCCR) where the value of the revolution counter matches that of the QPRC Position & Revolution Counter Compare Register(QPRCR) when the mode is PC_Mode1(QCR.PCM[1:0]= 0b01), PC_Mode2(QCR.PCM[1:0]= 0b10), or PC_Mode3(QCR.PCM[1:0]= 0b11) and the revolution counter is in the mode other than RC_Mode0(QCR.RCM[1:0]= 0b00). −The value of the revolution counter matches that of the QPRC Position and Revolution Counter Compare Register(QPRCR) when the data is written to the QPRC Position and Revolution Counter Compare Register(QPRCR) in the mode other than RC_Mode0(QCR.RCM[1:0]= 0b00) where the specified value matches that of the QPRC Position Count Register(QPCR) or QPRC Position Counter Compare Register(QPCCR) in PC_Mode1(QCR.PCM[1:0]= 0b01), PC_Mode2(QCR.PCM[1:0]= 0b10), or PC_Mode3(QCR.PCM[1:0]= 0b11).

Section	Change Results
CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 4.Registers 4.8.High-Order Bytes of QPRC Interrupt Control Register (QICRH)	Features of "For details" should be corrected as indicated by the shading below. (Error) −The value of the revolution counter matches that of the QPRC Position and Revolution Counter Compare Register (QPRCR) when the mode is changed from RC_Mode0 (QCR:RCM[1:0]="00") to another mode where the specified value matches that of the QPRC Position Count Register (QPCR) or QPRC Position Counter Compare Register (QPCCR) in PC_Mode1 (QCR:PCM[1:0]="01"), PC_Mode2 (QCR:PCM[1:0]="10"), or PC_Mode3 (QCR:PCM[1:0]="11"). (Correct) −The value of the revolution counter matches that of the QPRC Position and Revolution Counter Compare Register(QPRCR) when the mode is changed from RC_Mode0(QCR.RCM[1:0]= 0b00) to another mode where the specified value matches that of the QPRC Position Count Register(QPCR) or QPRC Position Counter Compare Register(QPCCR) in PC_Mode1(QCR.PCM[1:0]= 0b01), PC_Mode2(QCR.PCM[1:0]= 0b10), or PC_Mode3(QCR.PCM[1:0]= 0b11).
CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 4.Registers 4.8.High-Order Bytes of QPRC Interrupt Control Register (QICRH)	Features of "For details" should be corrected as indicated by the shading below. (Error) [bit 12] QPCNRCMIE: PC match and RC match interrupt enable bit This bit is used to control whether or not to issue an interrupt notification to the CPU when the PC match and RC match interrupt request flag (QPCNRCMF) is set to "1". When this bit is set to "1", an interrupt is generated if the value of the position counter matches that of the QPRC Position Counter Compare Register (QPCCR) and the value of the revolution counter matches that of the QPRC Position and Revolution Counter Compare Register (QPRCR) (QPCNRCMF="1"). (Correct) [bit12] QPCNRCMIE : PC Match and RC Match Interrupt Enable Bit This bit is used to control whether or not to issue an interrupt notification to the MCU when the PC match and RC match interrupt request flag(QPCNRCMF) is set to "1". When this bit is set to "1", an interrupt is generated if the value of the position counter matches that of the QPRC Position Counter Compare Register(QPCCR) and the value of the revolution counter matches that of the QPRC Position and Revolution Counter Compare Register(QPRCR)(QPCNRCMF= 1).

Section	Change Results
CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 4.Registers 4.8.High-Order Bytes of QPRC Interrupt Control Register (QICRH)	Features of "For details" should be corrected as indicated by the shading below. (Error) Note: −As the direction of the position counter is not detected in PC_Mode0 (QCR:PCM[1:0]="00"), the last position counter direction bit (QICR:DIRPC) becomes indefinite. Therefore, if the mode is changed from PC_Mode0 (QCR:PCM[1:0]="00") to another mode, when a ZIN active edge is detected before an AIN/BIN active edge is detected, the following operations apply. −The position counter is reset if the mode is RC_Mode0 (QCR:RCM[1:0]="00"), RC_Mode1 (QCR:RCM[1:0]="01"), or RC_Mode3 (QCR:RCM[1:0]="11") −The revolution counter is not counted up or down (Correct) Note: −As the direction of the position counter is not detected in PC_Mode0(QCR.PCM[1:0]= 0b00), the last position counter direction bit(QICR.DIRPC) becomes indefinite. Therefore, if the mode is changed from PC_Mode0(QCR.PCM[1:0]= 0b00) to another mode, when a ZIN active edge is detected before an AIN/BIN active edge is detected, the following operations apply. −The position counter is reset if the mode is RC_Mode0(QCR.RCM[1:0]= 0b00), RC_Mode1(QCR.RCM[1:0]= 0b01), or RC_Mode3(QCR.RCM[1:0]= 0b11) −The revolution counter is not counted up or down
CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 4.Registers 4.8.High-Order Bytes of QPRC Interrupt Control Register (QICRH)	Features of "For details" should be corrected as indicated by the shading below. (Error) [bit 9] CDCF: Count inversion interrupt request flag bit This bit indicates whether or not the position counter inverted the count direction. This bit is set to "1" when the position counter inverts the count direction. Inverting the count direction means that the counter counts down at the next counting after counting up, or the counter counts up at the next counting after counting down. This flag can only clear to "0" in write mode. Setting "1" has no effect. (Correct) [bit9] CDCF : Count Inversion Interrupt Request Flag Bit This bit indicates whether or not the position counter inverted the count direction. This bit is set to "1" when the position counter inverts the count direction. Inverting the count direction means that the counter counts down at the next counting after counting up, or the counter counts up at the next counting after counting down. This flag can be only cleared to "0" in write mode. Setting "1" has no effect.

Section	Change Results
CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 4.Registers 4.8.High-Order Bytes of QPRC Interrupt Control Register (QICRH)	Features of "For details" should be corrected as indicated by the shading below. (Error) Note: −As the direction of the position counter is not detected in PC_Mode0 (QCR:PCM[1:0]="00"), the last position counter direction bit (QICR:DIRPC) becomes indefinite. Therefore, after the mode is changed from PC_Mode0 (QCR:PCM[1:0]="00") to another mode, even if an AIN/BIN active edge is detected and the direction of the position counter is inverted, the count inversion interrupt request flag bit (QICR:CDCF) is not set to "1". (Correct) Note: −As the direction of the position counter is not detected in PC_Mode0(QCR.PCM[1:0]= 0b00), the last position counter direction bit(QICR.DIRPC) becomes indefinite. Therefore, after the mode is changed from PC_Mode0(QCR.PCM[1:0]= 0b00) to another mode, even if an AIN/BIN active edge is detected and the direction of the position counter is inverted, the count inversion interrupt request flag bit(QICR.CDCF) is not set to "1".
CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 4.Registers 4.8.High-Order Bytes of QPRC Interrupt Control Register (QICRH)	Features of "For details" should be corrected as indicated by the shading below. (Error) [bit 8] CDCIE: Count inversion interrupt enable bit This bit is used to control whether or not to issue an interrupt notification to the CPU when the count inversion interrupt request flag (CDCF) is set to "1". When this bit is set to "1", an interrupt is generated if the count direction of the position counter is inverted (CDCF="1"). (Correct) [bit8] CDCIE : Count Inversion Interrupt Enable Bit This bit is used to control whether or not to issue an interrupt notification to the MCU when the count inversion interrupt request flag(CDCF) is set to "1". When this bit is set to "1", an interrupt is generated if the count direction of the position counter is inverted(CDCF= 1).

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER47: QPRC(Quadrature Position/Revolution Counter) 4.Registers 4.9.QPRC Maximum Position Register(QMPR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit 15:0] QMPR:
When the position counter is counted up, a position counter overflow is detected (QICR:OFDF="1") if the set value of the QPRC Maximum Position Register (QMPR) matches the value of the position counter.
When the position counter is counted down, the set value of the QPRC Maximum Position Register (QMPR) is reloaded to the position counter if a position counter underflow is detected (QICR:UFDF="1").

(Correct)
[bit15:0] QMPR[15:0] :
When the set value of the QPRC Maximum Position Register(QMPR) matches the value of the position counter and the position counter is counted up, a position counter overflow is detected(QICR.OFDF= 1).
The set value of the QPRC Maximum Position Register(QMPR) is reloaded to the position counter if a position counter underflow is detected(QICR.UFDF= 1).</td></tr>
<tr><td>CHAPTER50: I/O Port 4. Register List</td><td>The function should be deleted as indicated by the shading below.

(Error)
<table><tr><td>GPIO_PIDRi</td><td>Port input data register</td><td>4.8</td></tr><tr><td>GPIO_PPERi</td><td>Port enable register</td><td>4.9</td></tr><tr><td>GPIO_KEYCDR</td><td>GPIO key code register</td><td>4.10</td></tr></table>
(Correct)
<table><tr><td>GPIO_PIDRi</td><td>Port input data register</td><td>4.8</td></tr><tr><td>GPIO_KEYCDR</td><td>GPIO key code register</td><td>4.9</td></tr><tr><td>PPC_PCFGRijj</td><td>Port setting register</td><td>4.10</td></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER50:
I/O Port
4. Register List</td><td>The function should be deleted as indicated by the shading below.

(Error)
<table>
<tr><td>GPIO_KEYCDR</td><td>GPIO key code register</td><td>4.10</td></tr>
<tr><td>PPC_PCFGRijj</td><td>Port setting register</td><td>4.11</td></tr>
<tr><td>PPC_KEYCDR</td><td>PPC key code register</td><td>4.12</td></tr>
<tr><td>RIC_RESINn</td><td>Resource input setting register</td><td>4.13</td></tr>
<tr><td>RIC_KEYCDR</td><td>RIC key code register</td><td>4.14</td></tr>
</table>

(Correct)
<table>
<tr><td>GPIO_KEYCDR</td><td>GPIO key code register</td><td>4.9</td></tr>
<tr><td>PPC_PCFGRijj</td><td>Port setting register</td><td>4.10</td></tr>
<tr><td>PPC_KEYCDR</td><td>PPC key code register</td><td>4.11</td></tr>
<tr><td>RIC_RESINn</td><td>Resource input setting register</td><td>4.12</td></tr>
<tr><td>RIC_KEYCDR</td><td>RIC key code register</td><td>4.13</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER50:
I/O Port
4. Register List</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
<table>
<tr><td>0x00000340 to 37C</td><td>Reserved</td></tr>
<tr><td>0x00000380</td><td>GPIO_PPER0
00000000_00000000_00000000_00000000</td></tr>
<tr><td>0x00000384</td><td>GPIO_PPER1
00000000_00000000_00000000_00000000</td></tr>
<tr><td>0x00000388</td><td>GPIO_PPER2
00000000_00000000_00000000_00000000</td></tr>
<tr><td>0x0000038C</td><td>GPIO_PPER3
00000000_00000000_00000000_00000000</td></tr>
<tr><td>0x00000390</td><td>GPIO_PPER4
00000000_00000000_00000000_00000000</td></tr>
<tr><td>0x00000394</td><td>GPIO_PPER5
00000000_00000000_00000000_00000000</td></tr>
<tr><td>0x00000398</td><td>GPIO_PPER6
00000000_00000000_00000000_00000000</td></tr>
<tr><td>0x0000039C</td><td>GPIO_PPER7
00000000_00000000_00000000_00000000</td></tr>
</table>
(Correct)
<table>
<tr><td>0x00000340 to 37C</td><td>Reserved</td></tr>
<tr><td>0x00000380</td><td>Reserved</td></tr>
<tr><td>0x00000384</td><td>Reserved</td></tr>
<tr><td>0x00000388</td><td>Reserved</td></tr>
<tr><td>0x0000038C</td><td>Reserved</td></tr>
<tr><td>0x00000390</td><td>Reserved</td></tr>
<tr><td>0x00000394</td><td>Reserved</td></tr>
<tr><td>0x00000398</td><td>Reserved</td></tr>
<tr><td>0x0000039C</td><td>Reserved</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER50:
I/O Port
4. Register List</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
<table>
<tr><td>0x000003A0</td><td>GPIO_PPER8
00000000_00000000_00000000_00000000</td></tr>
<tr><td>0x000003A4</td><td>GPIO_PPER9
00000000_00000000_00000000_00000000</td></tr>
<tr><td>0x000003A8</td><td>GPIO_PPER10
00000000_00000000_00000000_00000000</td></tr>
<tr><td>0x000003AC</td><td>GPIO_PPER11
00000000_00000000_00000000_00000000</td></tr>
<tr><td>0x000003B0</td><td>GPIO_PPER12
00000000_00000000_00000000_00000000</td></tr>
<tr><td>0x000003B4</td><td>GPIO_PPER13
00000000_00000000_00000000_00000000</td></tr>
<tr><td>0x000003B8</td><td>GPIO_PPER14
00000000_00000000_00000000_00000000</td></tr>
<tr><td>0x000003BC</td><td>GPIO_PPER15
00000000_00000000_00000000_00000000</td></tr>
</table>

(Correct)
<table>
<tr><td>0x000003A0</td><td>Reserved</td></tr>
<tr><td>0x000003A4</td><td>Reserved</td></tr>
<tr><td>0x000003A8</td><td>Reserved</td></tr>
<tr><td>0x000003AC</td><td>Reserved</td></tr>
<tr><td>0x000003B0</td><td>Reserved</td></tr>
<tr><td>0x000003B4</td><td>Reserved</td></tr>
<tr><td>0x000003B8</td><td>Reserved</td></tr>
<tr><td>0x000003BC</td><td>Reserved</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER50: I/O Port
4. Register List
4.1.Data Direction Register (GPIO_DDRi) (i = 0 to (product specification))</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit n] DD[n]: Data direction selection bit (n = 0 to 31)
<table><tr><th>bit</th><th>Description</th></tr><tr><td>0</td><td>Set data direction to input.</td></tr><tr><td>1</td><td>Set data direction to output.</td></tr></table>
(Correct)
[bit31:0] DD[n]: Data Direction Selection Bit (n = 0 to 31)
<table><tr><th>DD[n] n=0 to 31t</th><th>Description</th></tr><tr><td>0</td><td>Set data direction to input.</td></tr><tr><td>1</td><td>Set data direction to output.</td></tr></table></td></tr>
<tr><td>CHAPTER50: I/O Port
4. Register List
4.1.Data Direction Register (GPIO_DDRi) (i = 0 to (product specification))</td><td>The item should be added as indicated by the shading below.

(Error)
Notes:
−This is the applicable key code register. To write to this register, the GPIO key code register (GPIO_KEYCDR) must be set.
−If this register is written before the key code is released, a bus error response is returned.
−After key code release, writing to a different address or using a different access size than specified in the key code sequence will not generate an error response.
−Please be sure to read and check a preset value after writing to a GPIO register.

(Correct)
Notes:
−This is the applicable key code register. To write to this register, the GPIO key code register (GPIO_KEYCDR) must be set.
−If this register is written before the key code is released, a bus error response is returned.
−After key code release, writing to a different address or using a different access size than specified in the key code sequence will not generate an error response, and it can't write the data correctly.
−Please be sure to read and check a preset value after writing to a GPIO register.</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER50:
I/O Port
4. Register List
4.2.Data Direction Set Register (GPIO_DDSRi)
(i = 0 to (product specification))</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit n] DDS[n]: Data direction set bit (n = 0 to 31)
<table><tr><th>bit</th><th>Description</th></tr><tr><td>0</td><td>No effect</td></tr><tr><td>1</td><td>Set the data direction selection bit (GPIO_DDRi: DD31 to DD0) to "1".</td></tr></table>
(Correct)
[bit31:0] DDS[n]: Data Direction Set Bit (n = 0 to 31)
<table><tr><th>DDS[n] n=0 to 31</th><th>Description</th></tr><tr><td>0</td><td>No effect</td></tr><tr><td>1</td><td>Set the data direction selection bit (GPIO_DDRi: DD31 to DD0) to "1".</td></tr></table></td></tr>
<tr><td>CHAPTER50:
I/O Port
4. Register List
4.3.Data Direction Clear Register (GPIO_DDCRi)
(i = 0 to (product specification))</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit n] DDC[n]: Data direction clear register (DDC[n]) (n=0 to 31)
<table><tr><th>bit</th><th>Description</th></tr><tr><td>0</td><td>No effect</td></tr><tr><td>1</td><td>Clear the data direction selection bit (GPIO_DDRi: DD31 to DD0) to "0".</td></tr></table>
(Correct)
[bit31:0] DDC[n]: Data Direction Clear Register (DDC[n]) (n=0 to 31)
<table><tr><th>DDC[n] n=0 to31</th><th>Description</th></tr><tr><td>0</td><td>No effect</td></tr><tr><td>1</td><td>Clear the data direction selection bit (GPIO_DDRi: DD31 to DD0) to "0".</td></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER50: I/O Port
4. Register List
4.4.Port Output Data Register (GPIO_PODRi) (i = 0 to (product specification))</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit n] POD[n]: Port output data bit (n = 0 to 31)
<table><tr><th>bit</th><th>Description</th></tr><tr><td>0</td><td>Output "L".</td></tr><tr><td>1</td><td>Output "H".</td></tr></table>
(Correct)
[bit31:0] POD[n]: Port Output Data Bit (n = 0 to 31)
<table><tr><th>POD[n] n=0 to 31</th><th>Description</th></tr><tr><td>0</td><td>Output "L".</td></tr><tr><td>1</td><td>Output "H".</td></tr></table></td></tr>
<tr><td>CHAPTER50: I/O Port
4. Register List
4.5.Port Output Set Register (GPIO_POSRi) (i = 0 to (product specification))</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit n] POS[n]: Port output set bit (n = 0 to 31)
<table><tr><th>bit</th><th>Description</th></tr><tr><td>0</td><td>No effect</td></tr><tr><td>1</td><td>Set the port output data bit (GPIO_PODRi: POD31 to POD0) to "1".</td></tr></table>
(Correct)
[bit31:0] POS[n]: Port Output Set Bit (n = 0 to 31)
<table><tr><th>POS[n] n=0 to 31</th><th>Description</th></tr><tr><td>0</td><td>No effect</td></tr><tr><td>1</td><td>Set the port output data bit (GPIO_PODRi: POD31 to POD0) to "1".</td></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER50:
I/O Port
4. Register List
4.6.Port Output Clear Register (GPIO_POCRi) (i = 0 to (product specification))</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit n] POC[n]: Port output clear bit (n = 0 to 31)
<table><tr><th>bit</th><th>Description</th></tr><tr><td>0</td><td>No effect</td></tr><tr><td>1</td><td>Clear the port output data bit (GPIO_PODRi: POD31 to POD0) to "0".</td></tr></table>
(Correct)
[bit31:0] POC[n]: Port Output Clear Bit (n = 0 to 31)
<table><tr><th>POC[n] n=0 to 31t</th><th>Description</th></tr><tr><td>0</td><td>No effect</td></tr><tr><td>1</td><td>Clear the port output data bit (GPIO_PODRi: POD31 to POD0) to "0".</td></tr></table></td></tr>
<tr><td>CHAPTER50:
I/O Port
4. Register List
4.8.Port Input Data Register (GPIO_PIDRi) (i = 0 to (product specification))</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit n] PID[n]: Port input data register (n = 0 to 31)
<table><tr><th>bit</th><th>Description</th></tr><tr><td>0</td><td>"L" is input.</td></tr><tr><td>1</td><td>"H" is input.</td></tr></table>
(Correct)
[bit31:0] PID[n]: Port Input Data Register (n = 0 to 31)
<table><tr><th>PID[n] n=0 to31</th><th>Description</th></tr><tr><td>0</td><td>"L" is input.</td></tr><tr><td>1</td><td>"H" is input.</td></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER50:
I/O Port
4. Register List
4.9.Port Enable Register (GPIO_PPERi) (i = 0 to (product specification))</td><td>
The function should be deleted as indicated by the shading below.

(Error)
4.9.Port Enable Register (GPIO_PPERi) (i = 0 to (product specification))
This register controls writing to the applicable GPIO function and reading/writing of the Pin Configuration Registers (PPC_PCFGRijj).
After a hardware reset, this register can be written only once in the privileged mode.
A bus error response is returned if this register is written in the user mode.

<table>
<tr><td>Bit</td><td>31</td><td>0</td></tr>
<tr><td>Field</td><td colspan="2">PPE</td></tr>
<tr><td>R/W Attribute</td><td colspan="2">R/W</td></tr>
<tr><td>Protection Attribute</td><td colspan="2">RP/WP</td></tr>
<tr><td>Initial Value</td><td colspan="2">00000000_00000000_00000000_00000000</td></tr>
</table>

[bit n] PPE: Port enable bit (n = 0 to 31)
<table>
<tr><th>bit</th><th>Description</th></tr>
<tr><td>0</td><td>Cannot access a corresponding pin or write to a GPIO channel.</td></tr>
<tr><td>1</td><td>Can access a corresponding pin and write to a GPIO channel.</td></tr>
</table>

Notes:
−The following are the conditions when a bus error occurs.
1.The second and subsequent write accesses in the privileged mode
2.Write access in modes other than the privileged mode (user read is also regarded as a bus error)
−Enable/disable of this register function can be switched by an option setting.

(Correct)
</td></tr>
<tr><td>CHAPTER50:
I/O Port
4. Register List
4.9.GPIO Key Code Register (GPIO_KEYCDR)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
4.10.GPIO Key Code Register (GPIO_KEYCDR)

(Correct)
4.9.GPIO Key Code Register (GPIO_KEYCDR)
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER50:
I/O Port
4. Register List
4.9.GPIO Key Code Register (GPIO_KEYCDR)</td><td>The item should be added as indicated by the shading below.

(Error)
This register sets register writing with a function for protection against erroneous writing. If writing to this register is not done using the prescribed method, writing to the relevant register is invalid.

(Correct)
This register sets register writing with a function for protection against erroneous writing. If writing to this register is not done using the prescribed method, writing to the relevant register is invalid.A lock target register can write only once in after unlocking.When writing continuously, please unlock once again.</td></tr>
<tr><td>CHAPTER50:
I/O Port
4. Register List
4.9.GPIO Key Code Register (GPIO_KEYCDR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit31:30] KEY: Key code bits
These are key code setting bits. Write "00", "01", "10", and "11" continuously to these bits in this order. The key code setting becomes invalid immediately upon any writing to these bits in a different order. In such cases, set the key code again from the beginning.
<table>
<tr><th colspan="2">bit31:30</th><th>Description</th></tr>
<tr><td>0</td><td>0</td><td>1st key code</td></tr>
<tr><td>0</td><td>1</td><td>2nd key code</td></tr>
<tr><td>1</td><td>0</td><td>3rd key code</td></tr>
<tr><td>1</td><td>1</td><td>4th key code</td></tr>
</table>

(Correct)
[bit31:30] KEY: Key Code Bits
These are key code setting bits. Write 0b00, 0b01, 0b10, and 0b11 continuously to these bits in this order. The key code setting becomes invalid immediately upon any writing to these bits in a different order. In such cases, set the key code again from the beginning.
<table>
<tr><th colspan="2">Bits</th><th>Description</th></tr>
<tr><td>0</td><td>0</td><td>1st key code</td></tr>
<tr><td>0</td><td>1</td><td>2nd key code</td></tr>
<tr><td>1</td><td>0</td><td>3rd key code</td></tr>
<tr><td>1</td><td>1</td><td>4th key code</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER50:
I/O Port
4. Register List
4.9.GPIO Key Code Register (GPIO_KEYCDR)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit29:28] SIZE: Access size bits
These bits set the access size for writing to the applicable key code register. Write the same data to these bits when writing key codes "00", "01", "10", and "11" in this order.
<table>
<tr><th colspan="2">bit29:28</th><th>Description</th></tr>
<tr><td>0</td><td>0</td><td>Set byte access.</td></tr>
<tr><td>0</td><td>1</td><td>Set half-word access.</td></tr>
<tr><td>1</td><td>0</td><td>Set word access.</td></tr>
<tr><td>1</td><td>1</td><td>Reserved</td></tr>
</table>

(Correct)
[bit29:28] SIZE: Access Size Bits
These bits set the access size for writing to the applicable key code register. Write the same data to these bits when writing key codes 0b00, 0b01, 0b10, and 0b11 in this order.
<table>
<tr><th colspan="2">Bits</th><th>Description</th></tr>
<tr><td>0</td><td>0</td><td>Set byte access.</td></tr>
<tr><td>0</td><td>1</td><td>Set half-word access.</td></tr>
<tr><td>1</td><td>0</td><td>Set word access.</td></tr>
<tr><td>1</td><td>1</td><td>Reserved</td></tr>
</table>
</td></tr>
<tr><td>CHAPTER50:
I/O Port
4. Register List
4.9.GPIO Key Code Register (GPIO_KEYCDR)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit14:0] RADR: Port address bits
These bits set the lower 15 bits of the address of the applicable key code register. Write the same data to these bits when writing key codes "00", "01", "10", and "11" in this order.
<table>
<tr><th>bit14:0</th><th>Description</th></tr>
<tr><td>-</td><td>Set the lower 15 bits of the address of the applicable key code register.</td></tr>
</table>

(Correct)
[bit14:0] RADR: Port Address Bits
These bits set the lower 15 bits of the address of the applicable key code register. Write the same data to these bits when writing key codes 0b00, 0b01, 0b10, and 0b11 in this order.
<table>
<tr><th>RADR</th><th>Description</th></tr>
<tr><td>-</td><td>Set the lower 15 bits of the address of the applicable key code register.</td></tr>
</table>
</td></tr>
</table>

Section	Change Results
CHAPTER50: I/O Port 4. Register List 4.9.GPIO Key Code Register (GPIO_KEYCDR)	Features of "For details" should be corrected as indicated by the shading below. (Error) Notes: −The key code setting becomes invalid upon any writing to the KEY[1:0] in any order than "00", "01", "10", and "11". In such case, set the key code again from the beginning. −When different data is written to the SIZE[1:0] or the RADR[14:0] while writing the key code "00", "01", "10", and "11", the key code setting will become invalid. In such case, set it again from the beginning. (Correct) Notes: −The key code setting becomes invalid upon any writing to the KEY[1:0] in any order than 0b00, 0b01, 0b10, and 0b11. In such case, set the key code again from the beginning. −When different data is written to the SIZE[1:0] or the RADR[14:0] while writing the key code 0b00, 0b01, 0b10, and 0b11, the key code setting will become invalid. In such case, set it again from the beginning.
CHAPTER50: I/O Port 4. Register List 4.9.GPIO Key Code Register (GPIO_KEYCDR)	The item should be added as indicated by the shading below. (Error) −The applicable key code registers are the following registers. −Data direction register (GPIO_DDRi) −Data direction set register (GPIO_DDSRi) −Data direction clear register (GPIO_DDCRi) −Port input enable register (GPIO_PORTEN) −This register is valid only for word access. (Correct) −The applicable key code registers are the following registers. −Data direction register (GPIO_DDRi) −Data direction set register (GPIO_DDSRi) −Data direction clear register (GPIO_DDCRi) −Port Output Data Register (GPIO_PODRi) −Port Output Set Register (GPIO_POSRi) −Port Output Clear Register (GPIO_POCRi) −Port input enable register (GPIO_PORTEN) −This register is valid only for word access.
CHAPTER50: I/O Port 4. Register List 4.9.GPIO Key Code Register (GPIO_KEYCDR)	Features of "For details" should be corrected as indicated by the shading below. (Error) −There is no effect on the key code protection release process even if registers other than this register and the applicable key code register that is in the release processing (the register set by the RADA) are accessed while writing the key code "00", "01", "10", and "11". (Correct) −There is no effect on the key code protection release process even if registers other than this register and the applicable key code register that is in the release processing (the register set by the RADA) are accessed while writing the key code 0b00, 0b01, 0b10, and 0b11.

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER50: I/O Port
4. Register List
4.10.Port Setting Register (PPC_PCFGRijj) (i = 0 to (product specification), jj = 00 to 31)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
4.11.Port Setting Register (PPC_PCFGRijj) (i = 0 to (product specification), jj = 00 to 31)

(Correct)
4.10.Port Setting Register (PPC_PCFGRijj) (i = 0 to (product specification), jj = 00 to 31)</td></tr>
<tr><td>CHAPTER50: I/O Port
4. Register List
4.10.Port Setting Register (PPC_PCFGRijj) (i = 0 to (product specification), jj = 00 to 31)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit11:10] PIL: Input level bit
This bit selects a pin input level.
<table><tr><th>bit11:10</th><th>Description</th></tr></table>
(Correct)
[bit11:10] PIL: Input Level Bit
This bit selects a pin input level.
<table><tr><th>Bits</th><th>Description</th></tr></table></td></tr>
<tr><td>CHAPTER50: I/O Port
4. Register List
4.10.Port Setting Register (PPC_PCFGRijj) (i = 0 to (product specification), jj = 00 to 31)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit7:6] ODR: Port output drive selection bit
This bit selects an output drive capacity of a port.
<table><tr><th>bit7:6</th><th>Description</th></tr></table>
(Correct)
[bit7:6] ODR: Port Output Drive Selection Bit
This bit selects an output drive capacity of a port.
<table><tr><th>Bits</th><th>Description</th></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER50:
I/O Port
4. Register List
4.10.Port Setting Register (PPC_PCFGRijj) (i = 0 to (product specification), jj = 00 to 31)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit2:0] POF: Port output function selection bit
This bit selects a function to output to a port.
<table><tr><th>bit2:0</th><th>Description</th></tr></table>
(Correct)
[bit2:0] POF: Port Output Function Selection Bit
This bit selects a function to output to a port.
<table><tr><th>Bits</th><th>Description</th></tr></table></td></tr>
<tr><td>CHAPTER50:
I/O Port
4. Register List
4.11.PPC Key Code Register (PPC_KEYCDR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
4.12.PPC Key Code Register (PPC_KEYCDR)

(Correct)
4.11.PPC Key Code Register (PPC_KEYCDR)</td></tr>
<tr><td>CHAPTER50:
I/O Port
4. Register List
4.11.PPC Key Code Register (PPC_KEYCDR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

[bit31:30] KEY: Key code bits
These are key code setting bits. Write "00", "01", "10", and "11" continuously to these bits in this order. The key code setting becomes invalid immediately upon any writing to these bits in a different order. In such cases, set the key code again from the beginning.
<table><tr><th>bit31:30</th><th>Description</th></tr></table>
(Correct)
[bit31:30] KEY: Key Code Bits
These are key code setting bits. Write 0b00, 0b01, 0b10, and 0b11 continuously to these bits in this order. The key code setting becomes invalid immediately upon any writing to these bits in a different order. In such cases, set the key code again from the beginning.
<table><tr><th>Bits</th><th>Description</th></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER50:
I/O Port
4. Register List
4.11.PPC Key Code Register (PPC_KEYCDR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit29:28] SIZE: Access size bits
These bits set the access size for writing to the applicable key code register. Write the same data to these bits when writing key codes "00", "01", "10", and "11" in this order.
<table><tr><th>bit29:28</th><th>Description</th></tr></table>
(Correct)
[bit29:28] SIZE: Access Size Bits
These bits set the access size for writing to the applicable key code register. Write the same data to these bits when writing key codes 0b00, 0b01, 0b10, and 0b11 in this order.
<table><tr><th>Bits</th><th>Description</th></tr></table></td></tr>
<tr><td>CHAPTER50:
I/O Port
4. Register List
4.11.PPC Key Code Register (PPC_KEYCDR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit14:0] RADR: Port address bits
These bits set the lower 15 bits of the address of the applicable key code register. Write the same data to these bits when writing key codes "00", "01", "10", and "11" in this order.
<table><tr><th>bit14:0</th><th>Description</th></tr></table>
(Correct)
[bit14:0] RADR: Port Address Bits
These bits set the lower 15 bits of the address of the applicable key code register. Write the same data to these bits when writing key codes 0b00, 0b01, 0b10, and 0b11 in this order.
<table><tr><th>RADR</th><th>Description</th></tr></table></td></tr>
</table>

Section	Change Results
CHAPTER50: I/O Port 4. Register List 4.11.PPC Key Code Register (PPC_KEYCDR)	Features of "For details" should be corrected as indicated by the shading below. (Error) Notes: −The key code setting becomes invalid upon any writing to the KEY[1:0] in any order than "00", "01", "10", and "11". In such case, set the key code again from the beginning. −When different data is written to the SIZE[1:0] or the RADR[14:0] while writing the key code "00", "01", "10", and "11", the key code setting will become invalid. In such case, set it again from the beginning. −The applicable key code register is the port setting register (PPC_PCFGRijj). −This register is valid only for word access. −There is no effect on the key code protection release process even if registers other than this register and the applicable key code register that is in the release processing (the register set by the RADA) are accessed while writing the key code "00", "01", "10", and "11". (Correct) Notes: −The key code setting becomes invalid upon any writing to the KEY[1:0] in any order than 0b00, 0b01, 0b10, and 0b11. In such case, set the key code again from the beginning. −When different data is written to the SIZE[1:0] or the RADR[14:0] while writing the key code 0b00, 0b01, 0b10, and 0b11, the key code setting will become invalid. In such case, set it again from the beginning. −The applicable key code register is the port setting register (PPC_PCFGRijj). −This register is valid only for word access. −There is no effect on the key code protection release process even if registers other than this register and the applicable key code register that is in the release processing (the register set by the RADA) are accessed while writing the key code 0b00, 0b01, 0b10, and 0b11.
CHAPTER50: I/O Port 4. Register List 4.12.Resource Input Setting Register (RIC_RESINn) (n = 0 to (product specification))	Features of "For details" should be corrected as indicated by the shading below. (Error) 4.13.Resource Input Setting Register (RIC_RESINn) (n = 0 to (product specification)) (Correct) 4.12.Resource Input Setting Register (RIC_RESINn) (n = 0 to (product specification))

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER50: I/O Port 4. Register List 4.12.Resource Input Setting Register (RIC_RESINn) (n = 0 to (product specification))</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit11:8] PORTSEL: Resource selection bit
This bit selects an input to a corresponding resource.
<table><tr><th>bit11:8</th><th>Description</th></tr></table>
(Correct)
[bit11:8] PORTSEL: Resource Selection Bit
This bit selects an input to a corresponding resource.
<table><tr><th>Bits</th><th>Description</th></tr></table></td></tr>
<tr><td>CHAPTER50: I/O Port 4. Register List 4.12.Resource Input Setting Register (RIC_RESINn) (n = 0 to (product specification))</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit3:0] RESSEL: Resource selection bit
This bit selects an input to a resource.
<table><tr><th>bit3:0</th><th>Description</th></tr></table>
(Correct)
[bit3:0] RESSEL: Resource Selection Bit
This bit selects an input to a resource..
<table><tr><th>Bits</th><th>Description</th></tr></table></td></tr>
<tr><td>CHAPTER50: I/O Port 4. Register List 4.13.RIC Key Code Register (RIC_KEYCDR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
4.14.RIC Key Code Register (RIC_KEYCDR)

(Correct)
4.13.RIC Key Code Register (RIC_KEYCDR)</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER50:
I/O Port
4. Register List
4.13.RIC Key Code Register (RIC_KEYCDR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit31:30] KEY: Key code bits
These are key code setting bits. Write "00", "01", "10", and "11" continuously to these bits in this order. The key code setting becomes invalid immediately upon any writing to these bits in a different order. In such cases, set the key code again from the beginning.
<table><tr><th>bit31:30</th><th>Description</th></tr></table>
(Correct)
[bit31:30] KEY: Key Code Bits
These are key code setting bits. Write 0b00, 0b01, 0b10, and 0b11 continuously to these bits in this order. The key code setting becomes invalid immediately upon any writing to these bits in a different order. In such cases, set the key code again from the beginning.
<table><tr><th>Bits</th><th>Description</th></tr></table></td></tr>
<tr><td>CHAPTER50:
I/O Port
4. Register List
4.13.RIC Key Code Register (RIC_KEYCDR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit29:28] SIZE: Access size bits
These bits set the access size for writing to the applicable key code register. Write the same data to these bits when writing key codes "00", "01", "10", and "11" in this order.
<table><tr><th>bit29:28</th><th>Description</th></tr></table>
(Correct)
[bit29:28] SIZE: Access Size Bits
These bits set the access size for writing to the applicable key code register. Write the same data to these bits when writing key codes 0b00, 0b01, 0b10, and 0b11 in this order.
<table><tr><th>Bits</th><th>Description</th></tr></table></td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER50:
I/O Port
4. Register List
4.13.RIC Key Code Register (RIC_KEYCDR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit14:0] RADR: Port address bits
These bits set the lower 15 bits of the address of the applicable key code register. Write the same data to these bits when writing key codes "00", "01", "10", and "11" in this order.
<table><tr><th>bit14:0</th><th>Description</th></tr></table>
(Correct)
[bit14:0] RADR: Port Address Bits
These bits set the lower 15 bits of the address of the applicable key code register. Write the same data to these bits when writing key codes 0b00, 0b01, 0b10, and 0b11 in this order.
<table><tr><th>RADR</th><th>Description</th></tr></table></td></tr>
<tr><td>CHAPTER50:
I/O Port
4. Register List
4.13.RIC Key Code Register (RIC_KEYCDR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
Notes:
−The key code setting becomes invalid upon any writing to the KEY[1:0] in any order than "00", "01", "10", and "11". In such case, set the key code again from the beginning.
−When different data is written to the SIZE[1:0] or the RADR[14:0] while writing the key code "00", "01", "10", and "11", the key code setting will become invalid. In such case, set it again from the beginning.
−The applicable key code register is the resource input setting register (RIC_RESINn).
−This register is valid only for word access.
−There is no effect on the key code protection release process even if registers other than this register and the applicable key code register that is in the release processing (the register set by the RADA) are accessed while writing the key code "00", "01", "10", and "11".

(Correct)
Notes:
−The key code setting becomes invalid upon any writing to the KEY[1:0] in any order than 0b00, 0b01, 0b10, and 0b11. In such case, set the key code again from the beginning.
−When different data is written to the SIZE[1:0] or the RADR[14:0] while writing the key code 0b00, 0b01, 0b10, and 0b11, the key code setting will become invalid. In such case, set it again from the beginning.
−The applicable key code register is the resource input setting register (RIC_RESINn).
−This register is valid only for word access.
−There is no effect on the key code protection release process even if registers other than this register and the applicable key code register that is in the release processing (the register set by the RADA) are accessed while writing the key code 0b00, 0b01, 0b10, and 0b11.</td></tr>
</table>

Section	Change Results
CHAPTER51: Peripheral Protection Unit 3.Operation of the PPU 3.1. Access Type and PPU Operation	Features of "For details" should be corrected as indicated by the shading below. (Error) Table 3-1 Access permission at Read/Write mode in privilege mode access (Correct) Table CHAPTER 10:3-1 Access Permission at Read/Write Mode in Privilege Mode Access
CHAPTER51: Peripheral Protection Unit 3.Operation of the PPU 3.1. Access Type and PPU Operation	Features of "For details" should be corrected as indicated by the shading below. (Error) Table 3-2 Access permission at Read/Access mode in privilege mode access (Correct) Table CHAPTER 10:3-2 Access Permission at Read/Access Mode in Privilege Mode Access
CHAPTER51: Peripheral Protection Unit 3.Operation of the PPU 3.1. Access Type and PPU Operation	Features of "For details" should be corrected as indicated by the shading below. (Error) Table 3-3 Access permission at Read/Write mode in user mode access (Correct) Table CHAPTER 10:3-3 Access Permission at Read/Write Mode in User Mode Access
CHAPTER51: Peripheral Protection Unit 3.Operation of the PPU 3.1. Access Type and PPU Operation	Features of "For details" should be corrected as indicated by the shading below. (Error) Table 3-4 Access permission at Read/Access mode in user mode access (Correct) Table CHAPTER 10:3-4 Access Permission at Read/Access Mode in User Mode Access
CHAPTER51: Peripheral Protection Unit 3.Operation of the PPU 3.2.Access protection to PPU register	Features of "For details" should be corrected as indicated by the shading below. (Error) Table 3-8 Write to PPU0_PPRi, PPU0_PURi, PPU0_PPWAi, PPU0_PUWAi by DMA transfer (Correct) Table CHAPTER 10:3-8 Write to PPU0_PPRi, PPU0_PURi, PPU0_PPWAi, PPU0_PUWAi by DMA transfer

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER51: Peripheral Protection Unit
3.Operation of the PPU
3.4.Privilege mode forced change function</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
<table>
<tr><td>Access mode from the bus master</td><td>PPU0_PFENi:PFEN[x]</td><td>Access mode to the peripheral</td><td>Access mode to be used for PPU access control</td></tr>
<tr><td>User mode</td><td>0</td><td>User mode</td><td>User mode</td></tr>
<tr><td>Privilege mode</td><td>Don’t' care</td><td>Privilege mode</td><td>Privilege mode</td></tr>
<tr><td>User mode</td><td>1</td><td>Privilege mode</td><td>User mode</td></tr>
</table>
(Correct)
<table>
<tr><td>Access mode from the bus master</td><td>PPU0_PFENi:PFEN[x]</td><td>Access mode to the peripheral</td><td>Access mode to be used for PPU access control</td></tr>
<tr><td>User mode</td><td>0</td><td>User mode</td><td>User mode</td></tr>
<tr><td>Privilege mode</td><td>Don’t care</td><td>Privilege mode</td><td>Privilege mode</td></tr>
<tr><td>User mode</td><td>1</td><td>Privilege mode</td><td>User mode</td></tr>
</table>
</td></tr>
<tr><td>CHAPTER51: Peripheral Protection Unit
4.Registers
4.2.PPU Status Register (PPU0_SR)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit 20:16] VL[4:0] : Violation location
This register contains information for access destination of the latest violated access. As application specific PPU protection area, areas of up to seven are available.
<table>
<tr><th>bit</th><th>Description</th></tr>
</table>
(Correct)
[bit 20:16] VL[4:0] : Violation Location
This register contains information for access destination of the latest violated access. As application specific PPU protection area, areas of up to seven are available.
<table>
<tr><th>Bits</th><th>Description</th></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER51: Peripheral Protection Unit 4.Registers 4.8.PPU Privileged Read Attribute Register (PPU0_PPRi)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit x] PPR[x] : PPU Privileged mode Read Attribute Bit (x=31~0)
Read attribute for Privileged mode access of the peripheral #(i*32+x).
<table>
<tr><th>bit</th><th>Description</th></tr>
<tr><td>0</td><td>Read attribute in privileged mode is disabled.</td></tr>
<tr><td>1</td><td>Read attribute in privilege mode is enabled.</td></tr>
</table>

(Correct)
[bit31:0] PPR[x] : PPU Privileged Mode Read Attribute Bit (x=31 to 0)
Read attribute for Privileged mode access of the peripheral #(i*32+x).
<table>
<tr><th>PPR [x] x=31 to 0</th><th>Description</th></tr>
<tr><td>0</td><td>Read attribute in privileged mode is disabled.</td></tr>
<tr><td>1</td><td>Read attribute in privilege mode is enabled.</td></tr>
</table>
</td></tr>
<tr><td>CHAPTER51: Peripheral Protection Unit 4.Registers 4.9.PPU User Read Attribute Register (PPU0_PURi)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit x] PUR[x] : PPU User mode Read Attribute Bit (x=31~0)
Read attribute for user mode access of the peripheral #(i*32+x).
<table>
<tr><th>bit</th><th>Description</th></tr>
<tr><td>0</td><td>Read attribute in user mode is disabled.</td></tr>
<tr><td>1</td><td>Read attribute in user mode is enabled.</td></tr>
</table>

(Correct)
[bit31:0] PUR[x] : PPU User Mode Read Attribute Bit (x=31 to 0)
Read attribute for user mode access of the peripheral #(i*32+x).
<table>
<tr><th>PUR [x] x=31 to 0</th><th>Description</th></tr>
<tr><td>0</td><td>Read attribute in user mode is disabled.</td></tr>
<tr><td>1</td><td>Read attribute in user mode is enabled.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER51: Peripheral Protection Unit
4.Registers
4.10.PPU Privileged Write or Access Attribute Register (PPU0_PPWAi)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit x] PPWA[x] : PPU Privileged mode Write or Access Attribute Bit (x=31~0)
Write or access attribute for Privileged mode access of the peripheral #(i*32+x).
<table>
<tr><th>bit</th><th>Description</th></tr>
<tr><td>0</td><td>[Read/Write mode]
Write attribute in privileged mode is disabled.
[Read/Access mode]
Access attribute in privileged mode is disabled.</td></tr>
<tr><td>1</td><td>[Read/Write mode]
Write attribute in privileged mode is enabled.
[Read/Access mode]
Access attribute in privileged mode is enabled.</td></tr>
</table>

(Correct)
[bit31:0] PPWA[x] : PPU Privileged Mode Write or Access Attribute Bit (x=31 to 0)
Write or access attribute for Privileged mode access of the peripheral #(i*32+x).
<table>
<tr><th>PPWA [x] x=31 to 0</th><th>Description</th></tr>
<tr><td>0</td><td>[Read/Write mode]
Write attribute in privileged mode is disabled.
[Read/Access mode]
Access attribute in privileged mode is disabled.</td></tr>
<tr><td>1</td><td>[Read/Write mode]
Write attribute in privileged mode is enabled.
[Read/Access mode]
Access attribute in privileged mode is enabled.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER51: Peripheral Protection Unit 4.Registers 4.11.PPU User Write or Access Attribute Register (PPU0_PUWAi)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit x] PUWA[x] : PPU User mode Write or Access Attribute Bit (x=31~0)
Write or access attribute for User mode access of the peripheral #(i*32+x).
<table>
<tr><th>bit</th><th>Description</th></tr>
<tr><td>0</td><td>[Read/Write mode]
Write attribute in user mode is disabled.
[Read/Access mode]
Access attribute in user mode is disabled.</td></tr>
<tr><td>1</td><td>[Read/Write mode]
Write attribute in user mode is enabled.
[Read/Access mode]
Access attribute in user mode is enabled.</td></tr>
</table>

(Correct)
[bit31:0] PUWA[x] : PPU User Mode Write or Access Attribute Bit (x=31 to 0)
Write or access attribute for User mode access of the peripheral #(i*32+x).
<table>
<tr><th>PUWA[x] x=31 to 0</th><th>Description</th></tr>
<tr><td>0</td><td>[Read/Write mode]
Write attribute in user mode is disabled.
[Read/Access mode]
Access attribute in user mode is disabled.</td></tr>
<tr><td>1</td><td>[Read/Write mode]
Write attribute in user mode is enabled.
[Read/Access mode]
Access attribute in user mode is enabled.</td></tr>
</table>
</td></tr>
<tr><td>CHAPTER51: Peripheral Protection Unit 4.Registers 4.13.PPU Privilege Mode Forced change function Enable Register (PPU0_PFENi)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit x] PFEN[x] : PPU Privilege mode Forced change function enable Bit (x=31~0)

(Correct)
[bit31:0] PFEN[x] : PPU Privilege Mode Forced Change Function Enable Bit (x=31 to 0)
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER52:
DDR High
Speed SPI
Controller
1.Overview</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
Features of the DDR High Speed SPI Controller
−In case of DDRHSSPIn_MID.MID=0x00000001
−Supports Legacy Mode and Quad Mode
−Allows the use of up to 4 Serial Flash Memories
−In case of DDRHSSPIn_MID.MID=0x00000100

(Correct)
Features of the DDR High Speed SPI Controller
−In case of DDRHSSPIn_MID.MID=0x00000001
−Supports Legacy Mode and Quad Mode
−Allows the use of up to 4 Serial Flash Memories
−In case of DDRHSSPIn_MID.MID = 0x00000100 or 0x00000300</td></tr>
<tr><td>CHAPTER52:
DDR High
Speed SPI
Controller
1.Overview</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)<table><tr><td>SDR</td><td>Standard Data Rate</td></tr></table>
(Correct)<table><tr><td>SDR</td><td>Single Data Rate</td></tr></table></td></tr>
<tr><td>CHAPTER52:
DDR High
Speed SPI
Controller
3.Serial
Interface
3.3.SPI Input
Data Sampling
Point</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
This feature is available only when DDRHSSPIn_MID.MID is 0x00000100.

(Correct)
This feature is available only when DDRHSSPIn_MID.MID is 0x00000100 or 0x00000300.</td></tr>
<tr><td>CHAPTER52:
DDR High
Speed SPI
Controller
3.Serial
Interface
3.3.SPI Input
Data Sampling
Point</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
It is recommended to set values of corresponding registers such that SDATASMPTLFT <= SDATASMPTCNT<= SDATASMPTRGH.

(Correct)
It is recommended to set values of corresponding registers such that SDATASMPTLFT ≤ SDATASMPTCNT≤ SDATASMPTRGH.</td></tr>
</table>

Section	Change Results
CHAPTER52: DDR High Speed SPI Controller 3.Serial Interface 3.4.SPI Data Learning Pattern	Features of "For details" should be corrected as indicated by the shading below. (Error) This feature is available only when DDRHSSPIn_MID.MID is 0x00000100. (Correct) This feature is available only when DDRHSSPIn_MID.MID is 0x00000100 or 0x00000300.
CHAPTER52: DDR High Speed SPI Controller 3.Serial Interface 3.5.SPI Data Protocol	Features of "For details" should be corrected as indicated by the shading below. (Error) This feature is available only when DDRHSSPIn_MID.MID is 0x00000100. (Correct) This feature is available only when DDRHSSPIn_MID.MID is 0x00000100 or 0x00000300.
CHAPTER52: DDR High Speed SPI Controller 3.Serial Interface 3.6.Shift Direction	Features of "For details" should be corrected as indicated by the shading below. (Error) Figure 4-7 depicts the direction in which the data in the Shift Register is shifted to/from the serial data lines, under Quad Mode. The waveforms assume DDRHSSPIn_DMFIFOCFG.FWIDTH = "00". The Figures depict that the transmit data is loaded into the Shift Register from the TX-FIFO. However, the source of transmit data can be other registers, such as DDRHSSPIn_RDCSDC0-11.RDCSDATA. When DDRHSSPIn_MID.MID is 0x00000100, DDRHSSPI is available for Dual Legacy Mode and Dual Quad Mode. Figure 3-6 and Figure 3-7 depict the direction in which the data in the Shift Register is shifted to/from the serial data lines, under Dual Legacy Mode and Dual Quad Mode. (Correct) Figure 4-7 depicts the direction in which the data in the Shift Register is shifted to/from the serial data lines, under Quad Mode. The waveforms assume DDRHSSPIn_DMFIFOCFG.FWIDTH = 0b00. The Figures depict that the transmit data is loaded into the Shift Register from the TX-FIFO. However, the source of transmit data can be other registers, such as DDRHSSPIn_RDCSDC0-11.RDCSDATA. When DDRHSSPIn_MID.MID is 0x00000100 or 0x00000300, DDRHSSPI is available for Dual Legacy Mode and Dual Quad Mode. Figure 52.3-6 and Figure 52.3-7 depict the direction in which the data in the Shift Register is shifted to/from the serial data lines, under Dual Legacy Mode and Dual Quad Mode
CHAPTER52: DDR High Speed SPI Controller 3.Serial Interface 3.6.Shift Direction	Features of "For details" should be corrected as indicated by the shading below. (Error) Figure 3-5 Shift Direction (Legacy Mode and Quad Mode) (Assumptions: DDRHSSPIn_DMFIFOCFG.FWIDTH="00") (Correct) Figure 3-5 Shift Direction (Legacy Mode and Quad Mode) (Assumptions: DDRHSSPIn_DMFIFOCFG.FWIDTH= 0b00)

Section	Change Results
CHAPTER52: DDR High Speed SPI Controller 3.Serial Interface 3.6.Shift Direction	Features of "For details" should be corrected as indicated by the shading below. (Error) Figure 3-6 Shift Direction (Dual Legacy Mode and Dual Quad Mode in Direct Mode) (Assumptions: DDRHSSPIn_DMFIFOCFG.FWIDTH="00") (Correct) Figure 3-6 Shift Direction (Dual Legacy Mode and Dual Quad Mode in Direct Mode) (Assumptions: DDRHSSPIn_DMFIFOCFG.FWIDTH= 0b00)
CHAPTER52: DDR High Speed SPI Controller 3.Serial Interface 3.6.Shift Direction	Features of "For details" should be corrected as indicated by the shading below. (Error) Figure 3-7 Shift Direction (Dual Quad Mode in Command Sequencer Mode) (Assumptions: DDRHSSPIn_DMFIFOCFG.FWIDTH="00") (Correct) Figure 3-7 Shift Direction (Dual Quad Mode in Command Sequencer Mode) (Assumptions: DDRHSSPIn_DMFIFOCFG.FWIDTH= 0b00)
CHAPTER52: DDR High Speed SPI Controller 4.Operations of the DDRHSSPI	Features of "For details" should be corrected as indicated by the shading below. (Error) If DDRHSSPIn_MID.MID is 0x00000100, DDRHSSPI can be connected with up to 8 Serial Flash Memories. At the connection with 8 Serial Flash Memories, each of the 4 Slave Select outputs is connected with 2 Serial Flash Memories. (Correct) If DDRHSSPIn_MID.MID is 0x00000100 or 0x00000300, DDRHSSPI can be connected with up to 8 Serial Flash Memories. At the connection with 8 Serial Flash Memories, each of the 4 Slave Select outputs is connected with 2 Serial Flash Memories.
CHAPTER52: DDR High Speed SPI Controller 4.Operations of the DDRHSSPI 4.1.Direct Mode	Features of "For details" should be corrected as indicated by the shading below. (Error) −TX-Only Mode This is only to transmit data to serial interface, and supports Legacy Mode and Quad Mode. If DDRHSSPIn_MID.MID is 0x00000100, Dual Legacy Mode and Dual Quad Mode are added. This mode supports both SDR and DDR, and is used in the following cases. (Correct) −TX-Only Mode This is only to transmit data to serial interface, and supports Legacy Mode and Quad Mode. If DDRHSSPIn_MID.MID is 0x00000100 or 0x00000300, Dual Legacy Mode and Dual Quad Mode are added. This mode supports both SDR and DDR, and is used in the following cases.

Section	Change Results
CHAPTER52: DDR High Speed SPI Controller 4.Operations of the DDRHSSPI 4.1.Direct Mode 4.1.2.FIFO Size	Features of "For details" should be corrected as indicated by the shading below. (Error) Note: −When the value of DDRHSSPIn_DMFIFOCFG.FWIDTH is greater than 0, the value of DDRHSSPIn_DMBCC.BCC should be programmed along the boundary of bytes according to DDRHSSPIn_DMFIFOCFG.FWIDTH, e.g. when FWIDTH is "11", the BCC should be a multiple of 4. (Correct) Note: −When the value of DDRHSSPIn_DMFIFOCFG.FWIDTH is greater than 0, the value of DDRHSSPIn_DMBCC.BCC should be programmed along the boundary of bytes according to DDRHSSPIn_DMFIFOCFG.FWIDTH, e.g. when FWIDTH is 0b11, the BCC should be a multiple of 4.
CHAPTER52: DDR High Speed SPI Controller 4.Operations of the DDRHSSPI 4.1.Direct Mode 4.1.3.TX-FIFO Control	Features of "For details" should be corrected as indicated by the shading below. (Error) (1) DDRHSSPIn_DMTRP.TRP[1:0]≠"11" FWIDTH TRP[1:0] SDATA "00" "00" TXDATA[7:0] will be sent on SDATA[0]. "10" TXDATA[7:0] will be sent on SDATA[3:0]. "01" "00" TXDATA[15:8] and TXDATA[7:0] will be sequentially sent on SDATA[0]. "10" TXDATA[15:8] and TXDATA[7:0] will be sequentially sent on SDATA[3:0]. "11" "00" TXDATA[31:24], TXDATA[23:16], TXDATA[15:8] and TXDATA[7:0] will be sequentially sent on SDATA[0]. "10" TXDATA[31:24], TXDATA[23:16], TXDATA[15:8] and TXDATA[7:0] will be sequentially sent on SDATA[3:0]. In TX-and-RX Mode, it supports only Legacy Mode. So, RX-FIFO receives data from SDATA[1]. (Correct) (1) DDRHSSPIn_DMTRP.TRP[1:0] ≠ 0b11 FWIDTH TRP[1:0] SDATA 0b00 0b00 TXDATA[7:0] will be sent on SDATA[0]. 0b10 TXDATA[7:0] will be sent on SDATA[3:0]. 0b01 0b00 TXDATA[15:8] and TXDATA[7:0] will be sequentially sent on SDATA[0]. 0b10 TXDATA[15:8] and TXDATA[7:0] will be sequentially sent on SDATA[3:0]. 0b11 0b00 TXDATA[31:24], TXDATA[23:16], TXDATA[15:8] and TXDATA[7:0] will be sequentially sent on SDATA[0]. 0b10 TXDATA[31:24], TXDATA[23:16], TXDATA[15:8] and TXDATA[7:0] will be sequentially sent on SDATA[3:0]. In TX-and-RX Mode, it supports only Legacy Mode. So, RX-FIFO receives data from SDATA[1].

Section	Change Results
CHAPTER52: DDR High Speed SPI Controller 4.Operations of the DDRHSSPI 4.1.Direct Mode 4.1.3.TX-FIFO Control	Features of "For details" should be corrected as indicated by the shading below. (Error) (2) DDRHSSPIn_DMTRP.TRP[1:0]="11" (Only when DDRHSSPIn_MID.MID is 0x00000100) FWIDTH TRP[1:0] SDATA "00" "11" {TXDATA[7:4], TXDATA[3:0]} will be sent on {SDATA[7:4], SDATA[3:0]}. When in DDR Mode (DDRHSSPIn_DMFIFOCFG.DDRM="1"), TRP="11" is not allowed. "01" "11" {TXDATA[15:12], TXDATA[11:8]} and {TXDATA[7:4], TXDATA[3:0]} will be sequentially sent on {SDATA[7:4], SDATA[3:0]}. "11" "11" {TXDATA[31:28], TXDATA[27:24]}, {TXDATA[23:20], TXDATA[19:16]}, {TXDATA[15:12], TXDATA[11:8]} and {TXDATA[7:4], TXDATA[3:0]} will be sequentially sent on {SDATA[7:4], SDATA[3:0]}. When in TXCTRL="0", DDRHSSPI cannot transfer data at Dual Legacy Mode, since the SPI protocol is controlled just by DDRHSSPIn_DMTRP.TRP. Therefore, TX-and-RX Mode is not available for Dual Legacy Mode under this condition. (Correct) (2) DDRHSSPIn_DMTRP.TRP[1:0]= 0b11 (Only when DDRHSSPIn_MID.MID is 0x00000100 or 0x00000300) FWIDTH TRP[1:0] SDATA 0b00 0b11 {TXDATA[7:4], TXDATA[3:0]} will be sent on {SDATA[7:4], SDATA[3:0]}. When in DDR Mode (DDRHSSPIn_DMFIFOCFG.DDRM= 1), TRP= 0b11 is not allowed. 0b01 0b11 {TXDATA[15:12], TXDATA[11:8]} and {TXDATA[7:4], TXDATA[3:0]} will be sequentially sent on {SDATA[7:4], SDATA[3:0]}. 0b11 0b11 {TXDATA[31:28], TXDATA[27:24]}, {TXDATA[23:20], TXDATA[19:16]}, {TXDATA[15:12], TXDATA[11:8]} and {TXDATA[7:4], TXDATA[3:0]} will be sequentially sent on {SDATA[7:4], SDATA[3:0]}. When in TXCTRL= 0, DDRHSSPI cannot transfer data at Dual Legacy Mode, since the SPI protocol is controlled just by DDRHSSPIn_DMTRP.TRP. Therefore, TX-and-RX Mode is not available for Dual Legacy Mode under this condition.

Section	Change Results
CHAPTER52: DDR High Speed SPI Controller 4.Operations of the DDRHSSPI 4.1.Direct Mode 4.1.3.TX-FIFO Control	Features of "For details" should be corrected as indicated by the shading below. (Error) Common in (1) and (2): The actual valid byte lane of TXDATA varies according to the lower address bits for TX-FIFO, e.g. if FWIDTH="00" AND HSIZE=0x0 AND HADDR[1:0]="01", TXDATA[15:8] is valid instead of TXDATA[7:0]. And reading RX-FIFO is as well. (Correct) Common in (1) and (2): The actual valid byte lane of TXDATA varies according to the lower address bits for TX-FIFO, e.g. if FWIDTH= 0b00 AND HSIZE=0x0 AND HADDR[1:0]= 0b01, TXDATA[15:8] is valid instead of TXDATA[7:0]. And reading RX-FIFO is as well.

Section	Change Results
CHAPTER52: DDR High Speed SPI Controller 4.Operations of the DDRHSSPI 4.1.Direct Mode 4.1.3.TX-FIFO Control	Features of "For details" should be corrected as indicated by the shading below. (Error) Serial data output lines are tri-stated for defined number of SCLK cycles . The number of dummy cycles is defined in TXDATA[7:4] bits in following way: "0000" : Serial data output lines are tri-stated for 1 SCLK cycle "0001" : Serial data output lines are tri-stated for 2 SCLK cycles .... "1111" : Serial data output lines are tri-stated for 16 SCLK cycles Please note that maximum number of dummy cycles programmed per one TX-FIFO entry, is limited to be equivalent to sending one TX-FIFO level (with defined DDRHSSPIn_DMFIFOCFG.FWIDTH). The length of dummy cycles is limited as below. [Max Dummy Cycles] = [Bit number of 1 FIFO entry] / [1 Byte Time (cycle number)] For example, under the following conditions: −Quad Mode (by DDRHSSPIn_DMTRP.TRP[1:0]) −SDR data rate −number of dummy cycles is 4 −DDRHSSPIn_DMFIFOCFG.FWIDTH is "01" (16 bit-wide). (Correct) Serial data output lines are tri-stated for defined number of SCLK cycles . The number of dummy cycles is defined in TXDATA[7:4] bits in following way: 0b0000: Serial data output lines are tri-stated for 1 SCLK cycle 0b0001: Serial data output lines are tri-stated for 2 SCLK cycles .... 0b1111: Serial data output lines are tri-stated for 16 SCLK cycles Please note that maximum number of dummy cycles programmed per one TX-FIFO entry, is limited to be equivalent to sending one TX-FIFO level (with defined DDRHSSPIn_DMFIFOCFG.FWIDTH). The length of dummy cycles is limited as below. [Max Dummy Cycles] = [Bit number of 1 FIFO entry] / [1 Byte Time (cycle number)] For example, under the following conditions: −Quad Mode (by DDRHSSPIn_DMTRP.TRP[1:0]) −SDR data rate −number of dummy cycles is 4 −DDRHSSPIn_DMFIFOCFG.FWIDTH is 0b01 (16 bit-wide).

Section	Change Results
CHAPTER52: DDR High Speed SPI Controller 4.Operations of the DDRHSSPI 4.1.Direct Mode 4.1.3.TX-FIFO Control	Features of "For details" should be corrected as indicated by the shading below. (Error) SDR Mode In SDR Mode (DDRHSSPIn_DMTRP.DDRM = "0"), TXDATA[10] is ignored, and TXDATA[7:0] will be sent in SDR Mode, depending on TXDATA[9:8]: "00": Legacy Mode −If DDRHSSPIn_MID.MID is 0x00000100 and DDRHSSPIn_DMTRP. TRP is configured to Octal Mode (DDRHSSPIn_DMTRP.TRP[1:0]= "11"), this works as Dual Legacy Mode, and a byte data will be sent as below (4 SCLK cycles): (Correct) SDR Mode In SDR Mode (DDRHSSPIn_DMTRP.DDRM = 0), TXDATA[10] is ignored, and TXDATA[7:0] will be sent in SDR Mode, depending on TXDATA[9:8]: 0b00: Legacy Mode −If DDRHSSPIn_MID.MID is 0x00000100 or 0x00000300 and DDRHSSPIn_DMTRP. TRP is configured to Octal Mode (DDRHSSPIn_DMTRP.TRP[1:0]= 0b11), this works as Dual Legacy Mode, and a byte data will be sent as below (4 SCLK cycles):

Section	Change Results
CHAPTER52: DDR High Speed SPI Controller 4.Operations of the DDRHSSPI 4.1.Direct Mode 4.1.3.TX-FIFO Control	Features of "For details" should be corrected as indicated by the shading below. (Error) The actual valid byte lane of RXDATA varies according to the lower address bits for RX-FIFO, e.g. if FWIDTH="00" AND HSIZE=0x0 AND HADDR[1:0]="01", RXDATA[15:8] is valid instead of RXDATA[7:0]. −If DDRHSSPIn_DMTRP.TRP is not configured to Octal Mode (DDRHSSPIn_DMTRP.TRP[1:0]≠"11"), then TXDATA[7:0] will be sent in Legacy Mode (8 SCLK cycles) on SDATA[0]. On the RX side of TX-and-RX Mode, a byte data will be received from SDATA[1] (8 SCLK cycles). "01": Not allowed. "10": Quad Mode −If DDRHSSPIn_DMTRP.TRP is not configured to Octal Mode (DDRHSSPIn_DMTRP.TRP[1:0]≠"11"), then TXDATA[7:0] will be sent in Quad Mode (2 SCLK cycles) on SDATA[3:0]. −If DDRHSSPIn_DMTRP.TRP is configured to Octal Mode (DDRHSSPIn_DMTRP.TRP[1:0]="11"), then it is not allowed to set TXDATA[9:8] to "10". "11": Dual Quad Mode (Only when DDRHSSPIn_MID.MID is 0x00000100) −If DDRHSSPIn_DMTRP.TRP is configured to Octal Mode (DDRHSSPIn_DMTRP.TRP[1:0]="11"), this works as Dual Quad Mode, and a byte data will be sent as below (1 SCLK cycles): −TXDATA[7:4] will be sent on SDATA[7:4]. −TXDATA[3:0] will be sent on SDATA[3:0]. −If DDRHSSPIn_DMTRP.TRP is not configured to Octal Mode (DDRHSSPIn_DMTRP.TRP[1:0]≠"11"), then it is not allowed to set TXDATA[9:8] to "11". (Correct) The actual valid byte lane of RXDATA varies according to the lower address bits for RX-FIFO, e.g. if FWIDTH= 0b00 AND HSIZE=0x0 AND HADDR[1:0]= 0b01, RXDATA[15:8] is valid instead of RXDATA[7:0]. −If DDRHSSPIn_DMTRP.TRP is not configured to Octal Mode (DDRHSSPIn_DMTRP.TRP[1:0] ≠ 0b11), then TXDATA[7:0] will be sent in Legacy Mode (8 SCLK cycles) on SDATA[0]. On the RX side of TX-and-RX Mode, a byte data will be received from SDATA[1] (8 SCLK cycles). 0b01: Not allowed. 0b10: Quad Mode −If DDRHSSPIn_DMTRP.TRP is not configured to Octal Mode (DDRHSSPIn_DMTRP.TRP[1:0] ≠ 0b11), then TXDATA[7:0] will be sent in Quad Mode (2 SCLK cycles) on SDATA[3:0]. −If DDRHSSPIn_DMTRP.TRP is configured to Octal Mode (DDRHSSPIn_DMTRP.TRP[1:0]= 0b11), then it is not allowed to set TXDATA[9:8] to 0b10. 0b11: Dual Quad Mode (Only when DDRHSSPIn_MID.MID is 0x00000100 or 0x00000300) −If DDRHSSPIn_DMTRP.TRP is configured to Octal Mode (DDRHSSPIn_DMTRP.TRP[1:0]= 0b11), this works as Dual Quad Mode, and a byte data will be sent as below (1 SCLK cycles): −TXDATA[7:4] will be sent on SDATA[7:4]. −TXDATA[3:0] will be sent on SDATA[3:0]. −If DDRHSSPIn_DMTRP.TRP is not configured to Octal Mode (DDRHSSPIn_DMTRP.TRP[1:0] ≠ 0b11), then it is not allowed to set TXDATA[9:8] to 0b11.

Section	Change Results
CHAPTER52: DDR High Speed SPI Controller 4.Operations of the DDRHSSPI 4.1.Direct Mode 4.1.3.TX-FIFO Control	Features of "For details" should be corrected as indicated by the shading below. (Error) DDR Mode In DDR Mode (DDRHSSPIn_DMTRP.DDRM = "1"), then the behavior depends on TXDATA[10]: (1)SDR on-the-fly If TXDATA[10] = "0", then TXDATA[7:0] will be sent same as SDR Mode above. (2)DDR on-the-fly If TXDATA[10] = "1", then the behavior depends on bits TXDATA[9:8]: "00": Legacy Mode −If DDRHSSPIn_DMTRP.TRP is configured to Octal Mode (DDRHSSPIn_DMTRP.TRP[1:0]="11"), then it is not allowed to set TXDATA[9:8] to "00". −If DDRHSSPIn_DMTRP.TRP is not configured to Octal Mode (DDRHSSPIn_DMTRP.TRP[1:0]≠"11"), then TXDATA[7:0] will be sent in Legacy Mode (4 SCLK cycles) on SDATA[0]. "01": Not allowed. "10": Quad Mode (Correct) DDR Mode In DDR Mode (DDRHSSPIn_DMTRP.DDRM = 1), then the behavior depends on TXDATA[10]: (1)SDR on-the-fly If TXDATA[10] = 0, then TXDATA[7:0] will be sent same as SDR Mode above. (2)DDR on-the-fly If TXDATA[10] = 1, then the behavior depends on bits TXDATA[9:8]: 0b00: Legacy Mode −If DDRHSSPIn_DMTRP.TRP is configured to Octal Mode (DDRHSSPIn_DMTRP.TRP[1:0]= 0b11), then it is not allowed to set TXDATA[9:8] to 0b00. −If DDRHSSPIn_DMTRP.TRP is not configured to Octal Mode (DDRHSSPIn_DMTRP.TRP[1:0] ≠ 0b11), then TXDATA[7:0] will be sent in Legacy Mode (4 SCLK cycles) on SDATA[0]. 0b01: Not allowed. 0b10: Quad Mode

Section	Change Results
CHAPTER52: DDR High Speed SPI Controller 4.Operations of the DDRHSSPI 4.1.Direct Mode 4.1.3.TX-FIFO Control	Features of "For details" should be corrected as indicated by the shading below. (Error) −If DDRHSSPIn_DMTRP.TRP is not configured to Octal Mode (DDRHSSPIn_DMTRP.TRP[1:0]≠"11"), then TXDATA[7:0] will be sent in Quad Mode (1 SCLK cycles) on SDATA[3:0]. −If DDRHSSPIn_DMTRP.TRP is configured to Octal Mode (DDRHSSPIn_DMTRP.TRP[1:0]="11"), then it is not allowed to set TXDATA[9:8] to "10". "11": Dual Quad Mode (Only when DDRHSSPIn_MID.MID is 0x00000100) −If DDRHSSPIn_DMTRP.TRP is configured to Octal Mode (DDRHSSPIn_DMTRP.TRP[1:0]="11"), this works as Dual Quad Mode, and two bytes of data will be sent as below (1 SCLK cycles): −TXDATA[31:28][23:20] will be sent on SDATA[7:4]. −TXDATA[27:24][19:16] will be sent on SDATA[3:0]. −If DDRHSSPIn_DMTRP.TRP is not configured to Octal Mode (DDRHSSPIn_DMTRP.TRP[1:0]≠"11"), then it is not allowed to set TXDATA[9:8] to "11". (Correct) −If DDRHSSPIn_DMTRP.TRP is not configured to Octal Mode (DDRHSSPIn_DMTRP.TRP[1:0] ≠ 0b11), then TXDATA[7:0] will be sent in Quad Mode (1 SCLK cycles) on SDATA[3:0]. −If DDRHSSPIn_DMTRP.TRP is configured to Octal Mode (DDRHSSPIn_DMTRP.TRP[1:0]= 0b11), then it is not allowed to set TXDATA[9:8] to 0b10. 0b11: Dual Quad Mode (Only when DDRHSSPIn_MID.MID is 0x00000100 or 0x00000300) −If DDRHSSPIn_DMTRP.TRP is configured to Octal Mode (DDRHSSPIn_DMTRP.TRP[1:0]= 0b11), this works as Dual Quad Mode, and two bytes of data will be sent as below (1 SCLK cycles): −TXDATA[31:28][23:20] will be sent on SDATA[7:4]. −TXDATA[27:24][19:16] will be sent on SDATA[3:0]. −If DDRHSSPIn_DMTRP.TRP is not configured to Octal Mode (DDRHSSPIn_DMTRP.TRP[1:0] ≠ 0b11), then it is not allowed to set TXDATA[9:8] to 0b11.

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER52: DDR High Speed SPI Controller
4.Operations of the DDRHSSPI
4.1.Direct Mode
4.1.4.Available Command Formats of Serial Flash Memory</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
(1)SDR (DDRHSSPIn_DMTRP.DDRM="0")
Table 4-2 The available Serial Flash command format (SDR)
<table>
<tr><th colspan="2">DDRHSSPIn_MID</th><th rowspan="2">Command</th><th rowspan="2">Address</th><th rowspan="2">Mode (*1)</th><th rowspan="2">Dummy</th></tr>
<tr><th>0x0000 0001</th><th>0x0000 0100</th></tr>
</table>

(Correct)
(1)SDR (DDRHSSPIn_DMTRP.DDRM= 0)
Table 4-2 The Available Serial Flash Command Format (SDR)
<table>
<tr><th colspan="2">DDRHSSPIn_MID</th><th rowspan="2">Command</th><th rowspan="2">Address</th><th rowspan="2">Mode (*1)</th><th rowspan="2">Dummy</th></tr>
<tr><th>0x0000 0001</th><th>0x0000 0100 or 0x0000 0300</th></tr>
</table>
</td></tr>
<tr><td>CHAPTER52: DDR High Speed SPI Controller
4.Operations of the DDRHSSPI
4.1.Direct Mode
4.1.4.Available Command Formats of Serial Flash Memory</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
(2)DDR (DDRHSSPIn_DMTRP.DDRM="1")
Table 4-3 The available Serial Flash command format (DDR)
<table>
<tr><th colspan="2">DDRHSSPIn_MID</th><th rowspan="2">Command</th><th rowspan="2">Address</th><th rowspan="2">Mode (*1)</th><th rowspan="2">Dummy</th></tr>
<tr><th>0x0000 0001</th><th>0x0000 0100</th></tr>
</table>

(Correct)
(2)DDR (DDRHSSPIn_DMTRP.DDRM= 1)
Table 4-3 The Available Serial Flash Command Format (DDR)
<table>
<tr><th colspan="2">DDRHSSPIn_MID</th><th rowspan="2">Command</th><th rowspan="2">Address</th><th rowspan="2">Mode (*1)</th><th rowspan="2">Dummy</th></tr>
<tr><th>0x0000 0001</th><th>0x0000 0100 or 0x0000 0300</th></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER52: DDR High Speed SPI Controller
4.Operations of the DDRHSSPI
4.1.Direct Mode
4.1.4.Available Command Formats of Serial Flash Memory</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
<table>
<tr><th colspan="2">DDRHSSPIn_MID</th><th rowspan="2">DDRHSSPIn_DMTRP.TRP[1:0]</th><th rowspan="2">TXDATA[9:8] *1</th><th rowspan="2">Rx</th><th rowspan="2">Tx</th></tr>
<tr><th>0x00000001</th><th>0x00000100</th></tr>
<tr><td>Available</td><td>Available</td><td>"00" (Legacy)</td><td>"00" (Legacy)</td><td>SDATA[1]</td><td>SDATA[0]</td></tr>
<tr><td>N/A</td><td>N/A</td><td>"00" (Legacy)</td><td>"10" (Quad)</td><td>Not allowed</td><td>Not allowed</td></tr>
<tr><td>Available</td><td>Available</td><td>"10" (Quad)</td><td>"00" (Legacy)</td><td>SDATA[1]</td><td>SDATA[0]</td></tr>
<tr><td>Available</td><td>Available</td><td>"10" (Quad)</td><td>"10" (Quad)</td><td>Not allowed</td><td>SDATA[3:0]</td></tr>
<tr><td>N/A</td><td>Available</td><td>"11" (Dual Quad)</td><td>"00" (Dual Legacy)</td><td>SDATA[1], SDATA[5]</td><td>SDATA[0], SDATA[4]</td></tr>
<tr><td>N/A</td><td>Available</td><td>"11" (Dual Quad)</td><td>"10" (Quad)</td><td>Not allowed</td><td>Not allowed</td></tr>
<tr><td>N/A</td><td>Available</td><td>"11" (Dual Quad)</td><td>"11" (Dual Quad)</td><td>Not allowed</td><td>SDATA[7:0]</td></tr>
</table>
*1 With conditions of DDRHSSPIn_DMFIFOCFG.TXCTRL="1" and TXDATA[12]="1"

(Correct)
<table>
<tr><th colspan="2">DDRHSSPIn_MID</th><th rowspan="2">DDRHSSPIn_DMTRP.TRP[1:0]</th><th rowspan="2">TXDATA[9:8] *1</th><th rowspan="2">Rx</th><th rowspan="2">Tx</th></tr>
<tr><th>0x00000001</th><th>0x00000100 or 0x00000300</th></tr>
<tr><td>Available</td><td>Available</td><td>0b00 (Legacy)</td><td>0b00 (Legacy)</td><td>SDATA[1]</td><td>SDATA[0]</td></tr>
<tr><td>N/A</td><td>N/A</td><td>0b00 (Legacy)</td><td>0b10 (Quad)</td><td>Not allowed</td><td>Not allowed</td></tr>
<tr><td>Available</td><td>Available</td><td>0b10 (Quad)</td><td>0b00 (Legacy)</td><td>SDATA[1]</td><td>SDATA[0]</td></tr>
<tr><td>Available</td><td>Available</td><td>0b10 (Quad)</td><td>0b10 (Quad)</td><td>Not allowed</td><td>SDATA[3:0]</td></tr>
<tr><td>N/A</td><td>Available</td><td>0b11 (Dual Quad)</td><td>0b00 (Dual Legacy)</td><td>SDATA[1], SDATA[5]</td><td>SDATA[0], SDATA[4]</td></tr>
<tr><td>N/A</td><td>Available</td><td>0b11 (Dual Quad)</td><td>0b10 (Quad)</td><td>Not allowed</td><td>Not allowed</td></tr>
<tr><td>N/A</td><td>Available</td><td>0b11 (Dual Quad)</td><td>0b11 (Dual Quad)</td><td>Not allowed</td><td>SDATA[7:0]</td></tr>
</table>
*1 With conditions of DDRHSSPIn_DMFIFOCFG.TXCTRL= 1 and TXDATA[12]= 1
</td></tr>
</table>

Section	Change Results
CHAPTER52: DDR High Speed SPI Controller 4.Operations of the DDRHSSPI 4.1.Direct Mode 4.1.5.FIFO Accesses in Direct Mode	Features of "For details" should be corrected as indicated by the shading below. (Error) If DDRHSSPIn_DMFIFOCFG.FWIDTH is set to "11" (32-bit wide mode), then only 32-bit access to the FIFO is allowed. If DDRHSSPIn_DMFIFOCFG.FWIDTH is set to "01" (16-bit wide mode), then 16-bit and 32-bit accesses to the FIFO are allowed. A 32-bit access covers a whole word line for one FIFO entry. A 16-bit access with any alignment covers lower 16 bits of one FIFO entry. If DDRHSSPIn_DMFIFOCFG.FWIDTH is set to "00" (8-bit wide mode), then 8-bit, 16-bit and 32-bit accesses to the FIFO are allowed. A 32-bit access covers a whole word line for one FIFO entry. A 16-bit access with any alignment covers lower 16 bits of one FIFO entry. An 8-bit access with any alignment covers the lower 8 bits of one FIFO entry. (Correct) If DDRHSSPIn_DMFIFOCFG.FWIDTH is set to 0b11 (32-bit wide mode), then only 32-bit access to the FIFO is allowed. If DDRHSSPIn_DMFIFOCFG.FWIDTH is set to 0b01 (16-bit wide mode), then 16-bit and 32-bit accesses to the FIFO are allowed. A 32-bit access covers a whole word line for one FIFO entry. A 16-bit access with any alignment covers lower 16 bits of one FIFO entry. If DDRHSSPIn_DMFIFOCFG.FWIDTH is set to 0b00 (8-bit wide mode), then 8-bit, 16-bit and 32-bit accesses to the FIFO are allowed. A 32-bit access covers a whole word line for one FIFO entry. A 16-bit access with any alignment covers lower 16 bits of one FIFO entry. An 8-bit access with any alignment covers the lower 8 bits of one FIFO entry.
CHAPTER52: DDR High Speed SPI Controller 4.Operations of the DDRHSSPI 4.1.Direct Mode 4.1.5.FIFO Accesses in Direct Mode	Features of "For details" should be corrected as indicated by the shading below. (Error) When in DDRHSSPIn_DMFIFOCFG.TXCTRL="1", the setting of DDRHSSPIn_DMFIFOCFG.FWIDTH is ignored at writing into TX-FIFO, and up to 32 bits could be relevant. (please refer to the Table 52.4-1). Note: −In case that DDRHSSPI is configured in: −DDR Mode, AND −Dual Quad Mode setting DDRHSSPIn_DMFIFOCFG.FWIDTH to "00" (8-bit wide mode) is not allowed. (Correct) When in DDRHSSPIn_DMFIFOCFG.TXCTRL= 1, the setting of DDRHSSPIn_DMFIFOCFG.FWIDTH is ignored at writing into TX-FIFO, and up to 32 bits could be relevant. (please refer to the Table 52.4-1). Note: −In case that DDRHSSPI is configured in: −DDR Mode, AND −Dual Quad Mode setting DDRHSSPIn_DMFIFOCFG.FWIDTH to 0b00 (8-bit wide mode) is not allowed.

Section	Change Results
CHAPTER52: DDR High Speed SPI Controller 4.Operations of the DDRHSSPI 4.1.Direct Mode 4.1.6.Service Requests	Features of "For details" should be corrected as indicated by the shading below. (Error) The RX DMA Service Request (for the DMA Read Channel) is asserted if all of the following conditions are satisfied: −DDRHSSPIn_DMFIFOSTATUS.RXFLEVEL is more than DDRHSSPIn_DMFIFOCFG.RXFTH. This condition is the same as the DDRHSSPIn_RXF.RFMTS="1", AND −The DDRHSSPI RX Block Counter value is 0, AND −The DMA Read Channel acknowledgement signal is de-asserted by the DMA controller, AND −A previous DMA Read Service Request is not pending, AND −DDRHSSPIn_DMAEN.RXDMAEN="1", AND −DDRHSSPIn_FAULTF.DRCBSFS="0", AND −DDRHSSPIn_MCTRL.MES="1", AND −DDRHSSPIn_MCTRL.CSEN="0", AND −DDRHSSPIn_DMTRP.TRP[3:2]="00" (Correct) The RX DMA Service Request (for the DMA Read Channel) is asserted if all of the following conditions are satisfied: −DDRHSSPIn_DMFIFOSTATUS.RXFLEVEL is more than DDRHSSPIn_DMFIFOCFG.RXFTH. This condition is the same as the DDRHSSPIn_RXF.RFMTS= 1, AND −The DDRHSSPI RX Block Counter value is 0, AND −The DMA Read Channel acknowledgement signal is de-asserted by the DMA controller, AND −A previous DMA Read Service Request is not pending, AND −DDRHSSPIn_DMAEN.RXDMAEN= 1, AND −DDRHSSPIn_FAULTF.DRCBSFS= 0, AND −DDRHSSPIn_MCTRL.MES= 1, AND −DDRHSSPIn_MCTRL.CSEN= 0, AND −DDRHSSPIn_DMTRP.TRP[3:2]= 0b00
CHAPTER52: DDR High Speed SPI Controller 4.Operations of the DDRHSSPI 4.2.Command Sequencer Mode	Features of "For details" should be corrected as indicated by the shading below. (Error) Please refer to Figure 4-6. It shows how each 4GB Serial Flash Memory consists of 524288 banks when the DDRHSSPIn_CSCFG.MSEL is programmed to "0000". (Correct) Please refer to Figure 4-6. It shows how each 4GB Serial Flash Memory consists of 524288 banks when the DDRHSSPIn_CSCFG.MSEL is programmed to 0b0000.

Section	Change Results
CHAPTER52: DDR High Speed SPI Controller 4.Operations of the DDRHSSPI 4.2.Command Sequencer Mode	Features of "For details" should be corrected as indicated by the shading below. (Error) Figure 4-6 Addressing 4GB Serial Flash Memories on each Slave Select, Through Different Banks (DDRHSSPIn_CSCFG.MSEL="0000") (Correct) Figure 4-6 Addressing 4GB Serial Flash Memories on Each Slave Select, Through Different Banks (DDRHSSPIn_CSCFG.MSEL= 0b0000)

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER52: DDR High Speed SPI Controller
4.Operations of the DDRHSSPI
4.2.Command Sequencer Mode</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
Table 4-6 Decoding of the Read Command Sequence List
<table>
<tr><th>DEC</th><th>RDCSDATA [7:0]</th><th>Description</th></tr>
<tr><td>0</td><td>Don’t Care</td><td>Transmit RDCSDATA[7:0] as it is.</td></tr>
<tr><td>1</td><td>0000_0000</td><td>Transmit address bits [7:0] of the Serial Flash Memory to be accessed</td></tr>
<tr><td>1</td><td>0000_0001</td><td>Transmit address bits [15:8] of the Serial Flash Memory to be accessed</td></tr>
<tr><td>1</td><td>0000_0010</td><td>Transmit address bits [23:16] of the Serial Flash Memory to be accessed</td></tr>
<tr><td>1</td><td>0000_0011</td><td>Transmit address bits [31:24] of the Serial Flash Memory to be accessed</td></tr>
<tr><td>1</td><td>XXXX_X100</td><td>Dummy cycle(s). The bit field RDCSDATA [7:3] means the length of dummy cycles:
SDATA[7:3]="00000" -> Generates 1 dummy cycle of the SCLK
SDATA[7:3]="00001" -> Generates 2 dummy cycles of the SCLK
.....
Bits at "X" are for the value to control the length of dummy cycles.</td></tr>
<tr><td>1</td><td>0000_0111</td><td>End of List</td></tr>
</table>
(Correct)
Table 4-6 Decoding of the Read Command Sequence List
<table>
<tr><th>DEC</th><th>RDCSDATA [7:0]</th><th>Description</th></tr>
<tr><td>0</td><td>Don’t Care</td><td>Transmit RDCSDATA[7:0] as it is.</td></tr>
<tr><td>1</td><td>0b00000000</td><td>Transmit address bits [7:0] of the Serial Flash Memory to be accessed</td></tr>
<tr><td>1</td><td>0b00000001</td><td>Transmit address bits [15:8] of the Serial Flash Memory to be accessed</td></tr>
<tr><td>1</td><td>0b00000010</td><td>Transmit address bits [23:16] of the Serial Flash Memory to be accessed</td></tr>
<tr><td>1</td><td>0b00000011</td><td>Transmit address bits [31:24] of the Serial Flash Memory to be accessed</td></tr>
<tr><td>1</td><td>0bXXXXX100</td><td>Dummy cycle(s). The bit field RDCSDATA [7:3] means the length of dummy cycles:
SDATA[7:3]=0b00000 -> Generates 1 dummy cycle of the SCLK
SDATA[7:3]=0b00001 -> Generates 2 dummy cycles of the SCLK
.....
Bits at "X" are for the value to control the length of dummy cycles.</td></tr>
<tr><td>1</td><td>0b00000111</td><td>End of List</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER52: DDR High Speed SPI Controller
4.Operations of the DDRHSSPI
4.2.Command Sequencer Mode</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
Table 4-7 Command Sequence List for 24-bit address
<table>
<tr><th>Register</th><th>RDCSDATA [7:0]</th><th>DEC</th><th>Description</th></tr>
<tr><td>DDRHSSPIn_ RDCSDC0</td><td>0000_0002</td><td>1</td><td>address bits [23:16] of the Serial Flash Memory</td></tr>
<tr><td>DDRHSSPIn_ RDCSDC1</td><td>0000_0001</td><td>1</td><td>address bits [15:8] of the Serial Flash Memory</td></tr>
<tr><td>DDRHSSPIn_ RDCSDC2</td><td>0000_0000</td><td>1</td><td>address bits [7:0] of the Serial Flash Memory</td></tr>
<tr><td>DDRHSSPIn_ RDCSDC3</td><td>byte data</td><td>0</td><td>Mode code (*1)</td></tr>
<tr><td>DDRHSSPIn_ RDCSDC4</td><td>XXXX_X100</td><td>1</td><td>Dummy cycles
Bits at "X" are for the value to control the length of dummy cycles.</td></tr>
<tr><td>DDRHSSPIn_ RDCSDC5</td><td>0000_0111</td><td>1</td><td>End of List</td></tr>
</table>
(*1)The "Mode" field is a control code following Address field, and the actual code of Mode field depends on the Serial Flash products.

(Correct)
Table 4-7 Command Sequence List for 24-bit Address
<table>
<tr><th>Register</th><th>RDCSDATA [7:0]</th><th>DEC</th><th>Description</th></tr>
<tr><td>DDRHSSPIn_ RDCSDC0</td><td>0b00000010</td><td>1</td><td>address bits [23:16] of the Serial Flash Memory</td></tr>
<tr><td>DDRHSSPIn_ RDCSDC1</td><td>0b00000001</td><td>1</td><td>address bits [15:8] of the Serial Flash Memory</td></tr>
<tr><td>DDRHSSPIn_ RDCSDC2</td><td>0b00000000</td><td>1</td><td>address bits [7:0] of the Serial Flash Memory</td></tr>
<tr><td>DDRHSSPIn_ RDCSDC3</td><td>byte data</td><td>0</td><td>Mode code *1</td></tr>
<tr><td>DDRHSSPIn_ RDCSDC4</td><td>0bXXXXX100</td><td>1</td><td>Dummy cycles
Bits at "X" are for the value to control the length of dummy cycles.</td></tr>
<tr><td>DDRHSSPIn_ RDCSDC5</td><td>0b00000111</td><td>1</td><td>End of List</td></tr>
</table>
*1: The "Mode" field is a control code following Address field, and the actual code of Mode field depends on the Serial Flash products.
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER52: DDR High Speed SPI Controller
4.Operations of the DDRHSSPI
4.2.Command Sequencer Mode</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
Table 4-8 Command Sequence List for 32-bit address
<table>
<tr><th>Register</th><th>RDCSDATA [7:0]</th><th>DEC</th><th>Description</th></tr>
<tr><td>DDRHSSPIn_ RDCSDC0</td><td>0000_0003</td><td>1</td><td>address bits [31:24] of the Serial Flash Memory</td></tr>
<tr><td>DDRHSSPIn_ RDCSDC1</td><td>0000_0002</td><td>1</td><td>address bits [23:16] of the Serial Flash Memory</td></tr>
<tr><td>DDRHSSPIn_ RDCSDC2</td><td>0000_0001</td><td>1</td><td>address bits [15:8] of the Serial Flash Memory</td></tr>
<tr><td>DDRHSSPIn_ RDCSDC3</td><td>0000_0000</td><td>1</td><td>address bits [7:0] of the Serial Flash Memory</td></tr>
<tr><td>DDRHSSPIn_ RDCSDC4</td><td>byte data</td><td>0</td><td>Mode code (*1)</td></tr>
<tr><td>DDRHSSPIn_ RDCSDC5</td><td>XXXX_X100</td><td>1</td><td>Dummy cycles
Bits at "X" are for the value to control the length of dummy cycles.</td></tr>
<tr><td>DDRHSSPIn_ RDCSDC6</td><td>0000_0111</td><td>1</td><td>End of List</td></tr>
</table>
(*1)The "Mode" field is a control code following Address field, and the actual code of Mode field depends on the Serial Flash products.

(Correct)
Table 4-8 Command Sequence List for 32-bit Address
<table>
<tr><th>Register</th><th>RDCSDATA [7:0]</th><th>DEC</th><th>Description</th></tr>
<tr><td>DDRHSSPIn_ RDCSDC0</td><td>0b00000011</td><td>1</td><td>address bits [31:24] of the Serial Flash Memory</td></tr>
<tr><td>DDRHSSPIn_ RDCSDC1</td><td>0b00000010</td><td>1</td><td>address bits [23:16] of the Serial Flash Memory</td></tr>
<tr><td>DDRHSSPIn_ RDCSDC2</td><td>0b00000001</td><td>1</td><td>address bits [15:8] of the Serial Flash Memory</td></tr>
<tr><td>DDRHSSPIn_ RDCSDC3</td><td>0b00000000</td><td>1</td><td>address bits [7:0] of the Serial Flash Memory</td></tr>
<tr><td>DDRHSSPIn_ RDCSDC4</td><td>byte data</td><td>0</td><td>Mode code *1</td></tr>
<tr><td>DDRHSSPIn_ RDCSDC5</td><td>0bXXXXX100</td><td>1</td><td>Dummy cycles
Bits at "X" are for the value to control the length of dummy cycles.</td></tr>
<tr><td>DDRHSSPIn_ RDCSDC6</td><td>0b00000111</td><td>1</td><td>End of List</td></tr>
</table>
*1: The "Mode" field is a control code following Address field, and the actual code of Mode field depends on the Serial Flash products.
</td></tr>
</table>

Section	Change Results
CHAPTER52: DDR High Speed SPI Controller 4.Operations of the DDRHSSPI 4.2.Command Sequencer Mode	Features of "For details" should be corrected as indicated by the shading below. (Error) 3.Set the DDRHSSPIn_DMTRP.TRP to "1010" (TX-Only Mode and Quad Mode). 4.When in DDR Mode, set the DDRHSSPIn_DMTRP.DDRM bit to "1". Otherwise, clear the DDRHSSPIn_DMTRP.DDRM bit to "0". 5.Set the DDRHSSPIn_DMFIFOCFG.FWIDTH to "00". 6.Set the DDRHSSPIn_DMFIFOCFG.TXCTRL bit to "1". 7.Write a byte for the command field, into TX-FIFO. TXDATA[12] = "1", TXDATA[9:8] = "01" (Legacy Mode on-the-fly). TXDATA[10] = "0" (SDR Mode on-the-fly). TXDATA[7:0] is a byte for the command. Please refer to the data sheet of the Serial Flash Memory about the command code. 8.Write a byte for the address field [31:24], into TX-FIFO. TXDATA[12] = "1", TXDATA[9:8] = "10" (Quad Mode on-the-fly). TXDATA[10] = DDRHSSPIn_DMTRP.DDRM bit value. TXDATA[7:0] is a byte for the address[31:24]. 9.Write bytes for the address field [23:16], [15:8] and [7:0] into TX-FIFO as well. 10.Write a byte for the mode field following the address field, into TX-FIFO. TXDATA[12] = "1", TXDATA[9:8] = "10" (Quad Mode on-the-fly). TXDATA[10] = DDRHSSPIn_DMTRP.DDRM bit value. TXDATA[7:0] is a byte for the mode. Please refer to the data sheet of the Serial Flash Memory about the mode code. (Correct) 3.Set the DDRHSSPIn_DMTRP.TRP to 0b1010 (TX-Only Mode and Quad Mode). 4.When in DDR Mode, set the DDRHSSPIn_DMTRP.DDRM bit to "1". Otherwise, clear the DDRHSSPIn_DMTRP.DDRM bit to "0". 5.Set the DDRHSSPIn_DMFIFOCFG.FWIDTH to 0b00. 6.Set the DDRHSSPIn_DMFIFOCFG.TXCTRL bit to "1". 7.Write a byte for the command field, into TX-FIFO. TXDATA[12] = 1, TXDATA[9:8] = 0b01 (Legacy Mode on-the-fly). TXDATA[10] = 0 (SDR Mode on-the-fly). TXDATA[7:0] is a byte for the command. Please refer to the data sheet of the Serial Flash Memory about the command code. 8.Write a byte for the address field [31:24], into TX-FIFO. TXDATA[12] = 1, TXDATA[9:8] = 0b10 (Quad Mode on-the-fly). TXDATA[10] = DDRHSSPIn_DMTRP.DDRM bit value. TXDATA[7:0] is a byte for the address[31:24]. 9.Write bytes for the address field [23:16], [15:8] and [7:0] into TX-FIFO as well. 10.Write a byte for the mode field following the address field, into TX-FIFO. TXDATA[12] = 1, TXDATA[9:8] = 0b10 (Quad Mode on-the-fly). TXDATA[10] = DDRHSSPIn_DMTRP.DDRM bit value. TXDATA[7:0] is a byte for the mode. Please refer to the data sheet of the Serial Flash Memory about the mode code.

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER52: DDR High Speed SPI Controller 5.Registers 5.2.DDRHSSPI Peripheral Communication Configuration Registers (DDRHSSPIn_PCC0-3)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
It defines the time period after SSEL de-assertion when a new SSEL assertion can not occur.
<table><tr><th rowspan="2">SSELDEASRT[4:0]</th><th colspan="2">Description</th></tr><tr><th>SDR Mode</th><th>DDR Mode</th></tr></table>
(Correct)
It defines the time period after SSEL de-assertion when a new SSEL assertion can not occur.
<table><tr><th rowspan="2">Bits</th><th colspan="2">Description</th></tr><tr><th>SDR Mode</th><th>DDR Mode</th></tr></table></td></tr>
<tr><td>CHAPTER52: DDR High Speed SPI Controller 5.Registers 5.2.DDRHSSPI Peripheral Communication Configuration Registers (DDRHSSPIn_PCC0-3)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
This field decides the Clock Division Ratio for the SCLK.
<table><tr><th>CDRS[3:0]</th><th>Description</th></tr></table>
(Correct)
This field decides the Clock Division Ratio for the SCLK.
<table><tr><th>Bits</th><th>Description</th></tr></table></td></tr>
<tr><td>CHAPTER52: DDR High Speed SPI Controller 5.Registers 5.6.DDRHSSPI RX Interrupt Flag Register (DDRHSSPIn_RXF)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
This bit is available only when DDRHSSPIn_MID.MID is 0x00000100.

(Correct)
This bit is available only when DDRHSSPIn_MID.MID is 0x00000100 or 0x00000300.</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER52: DDR High Speed SPI Controller 5.Registers 5.14.DDRHSSPI Direct Mode Peripheral Select Register (DDRHSSPIn_DMPSEL)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
The PSEL bits decide which of the 4 Slave Selects in oDDRHSSPIn_SSEL3-0 is asserted for the current serial transfer.
<table>
<tr><th>PSEL[1:0]</th><th>Description</th></tr>
</table>

(Correct)
The PSEL bits decide which of the 4 Slave Selects in oDDRHSSPIn_SSEL3-0 is asserted for the current serial transfer.
<table>
<tr><th>Bits</th><th>Description</th></tr>
</table>
</td></tr>
<tr><td>CHAPTER52: DDR High Speed SPI Controller 5.Registers 5.15.DDRHSSPI Direct Mode Transfer Protocol Register (DDRHSSPIn_DMTRP)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
Bits TRP[1:0] indicate the using protocol : Legacy, Quad or Octal.
<table>
<tr><th>TRP[3:0]</th><th>Description</th></tr>
<tr><td>0000</td><td>TX-and-RX in Legacy Mode is configured.</td></tr>
<tr><td>0011</td><td>This value is available only when DDRHSSPIn_MID.MID is 0x00000100.
TX-and-RX in Octal Mode is configured.</td></tr>
<tr><td>1000</td><td>TX-Only in Legacy Mode is configured.</td></tr>
<tr><td>1010</td><td>TX-Only in Quad Mode is configured.</td></tr>
<tr><td>1011</td><td>This value is available only when DDRHSSPIn_MID.MID is 0x00000100.
TX-Only in Octal Mode is configured.</td></tr>
</table>

(Correct)
Bits TRP[1:0] indicate the using protocol : Legacy, Quad or Octal.
<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>0000</td><td>TX-and-RX in Legacy Mode is configured.</td></tr>
<tr><td>0011</td><td>This value is available only when DDRHSSPIn_MID.MID is 0x00000100 or 0x00000300.
TX-and-RX in Octal Mode is configured.</td></tr>
<tr><td>1000</td><td>TX-Only in Legacy Mode is configured.</td></tr>
<tr><td>1010</td><td>TX-Only in Quad Mode is configured.</td></tr>
<tr><td>1011</td><td>This value is available only when DDRHSSPIn_MID.MID is 0x00000100 or 0x00000300.
TX-Only in Octal Mode is configured.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER52: DDR High Speed SPI Controller
5.Registers
5.19.DDRHSSPI Direct Mode FIFO Configuration Register (DDRHSSPIn_DMFIFOCFG)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
This field indicates the FIFO Width. Depending on the configured width of the FIFO, the available size of the Shift-Register in the SPI-Core also changes.
<table><tr><th>FWIDTH[1:0]</th><th>Description</th></tr></table>
(Correct)
This field indicates the FIFO Width. Depending on the configured width of the FIFO, the available size of the Shift-Register in the SPI-Core also changes.
<table><tr><th>Bits</th><th>Description</th></tr></table></td></tr>
<tr><td>CHAPTER52: DDR High Speed SPI Controller
5.Registers
5.20.DDRHSSPI TX-FIFO Registers (DDRHSSPIn_TXFIFO0-23)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
When the Shift Register width is 32 bits (DDRHSSPIn_DMFIFOCFG.FWIDTH="11"), only 32-bit access is possible.
When the Shift Register width is 16 bits (DDRHSSPIn_DMFIFOCFG.FWIDTH="01"), both 16-bit and 32-bit access are possible.
a) 32-bit access shall access the full width of FIFO, but only the lower 16 bits are valid.
b) 16-bit access with any alignment shall access the lower 16 bits of the FIFO.
When the Shift Register width is 8 bits (DDRHSSPIn_DMFIFOCFG.FWIDTH="00"), then 8-bit, 16-bit and 32-bit access are possible.

(Correct)
When the Shift Register width is 32 bits (DDRHSSPIn_DMFIFOCFG.FWIDTH=0b11), only 32-bit access is possible.
When the Shift Register width is 16 bits (DDRHSSPIn_DMFIFOCFG.FWIDTH=0b01), both 16-bit and 32-bit access are possible.
a) 32-bit access shall access the full width of FIFO, but only the lower 16 bits are valid.
b) 16-bit access with any alignment shall access the lower 16 bits of the FIFO.
When the Shift Register width is 8 bits (DDRHSSPIn_DMFIFOCFG.FWIDTH=0b00), then 8-bit, 16-bit and 32-bit access are possible.</td></tr>
</table>

Section	Change Results
CHAPTER52: DDR High Speed SPI Controller 5.Registers 5.21.DDRHSSPI RX-FIFO Registers (DDRHSSPIn_RXFIFO0-23)	Features of "For details" should be corrected as indicated by the shading below. (Error) When the Shift Register width is 32 bits (DDRHSSPIn_DMFIFOCFG.FWIDTH="11"), only 32-bit access is possible. When the Shift Register width is 16 bits (DDRHSSPIn_DMFIFOCFG.FWIDTH="01"), both 16-bit and 32-bit access are possible. a) 32-bit access shall access the full width of FIFO, but only the lower 16 bits are valid. b) 16-bit access with any alignment shall access the lower 16 bits of the FIFO. When the Shift Register width is 8 bits (DDRHSSPIn_DMFIFOCFG.FWIDTH="00"), then 8-bit, 16-bit and 32-bit access are possible. (Correct) When the Shift Register width is 32 bits (DDRHSSPIn_DMFIFOCFG.FWIDTH=0b11), only 32-bit access is possible. When the Shift Register width is 16 bits (DDRHSSPIn_DMFIFOCFG.FWIDTH=0b01), both 16-bit and 32-bit access are possible. a) 32-bit access shall access the full width of FIFO, but only the lower 16 bits are valid. b) 16-bit access with any alignment shall access the lower 16 bits of the FIFO. When the Shift Register width is 8 bits (DDRHSSPIn_DMFIFOCFG.FWIDTH= 0b00), then 8-bit, 16-bit and 32-bit access are possible.
CHAPTER52: DDR High Speed SPI Controller 5.Registers 5.21.DDRHSSPI RX-FIFO Registers (DDRHSSPIn_RXFIFO0-23)	Features of "For details" should be corrected as indicated by the shading below. (Error) When the bus width to access the FIFO is greater than the width of FIFO, the upper redundant bits of the read data on the System Bus are filled with "0". For example, if configured FIFO width is 8 bits (DDRHSSPIn_DMFIFOCFG.FWIDTH="00"), and the read access of the FIFO is on 32 bits, then (Correct) When the bus width to access the FIFO is greater than the width of FIFO, the upper redundant bits of the read data on the System Bus are filled with "0". For example, if configured FIFO width is 8 bits (DDRHSSPIn_DMFIFOCFG.FWIDTH= 0b00), and the read access of the FIFO is on 32 bits, then

Section	Change Results
CHAPTER52: DDR High Speed SPI Controller 5.Registers 5.22.DDRHSSPI Read Command Sequencer Data/Control Registers (DDRHSSPIn_RDCSDC0-11)	Features of "For details" should be corrected as indicated by the shading below. (Error) −When DEC bit is "1": The RDCSDATA[7:0] is decoded as follows. RDCSDATA[2:0] = "000": Transmit address bits [7:0] of the Serial Flash Memory address. RDCSDATA[2:0] = "001": Transmit address bits [15:8] of the Serial Flash Memory address. RDCSDATA[2:0] = "010": Transmit address bits [23:16] of the Serial Flash Memory address. RDCSDATA[2:0] = "011": Transmit address bits [31:24] of the Serial Flash Memory address. RDCSDATA[2:0] = "100": Dummy cycle(s) on the SDATA output lines for RDCSDATA [7:3] number of SCLK cycles. - RDCSDATA[7:3]="00000" -> Dummy cycle for 1 SCLK cycle. - RDCSDATA[7:3]="00001" -> Dummy cycles for 2 SCLK cycles. .... RDCSDATA[2:0] = "111": End of list. (Correct) −When DEC bit is "1": The RDCSDATA[7:0] is decoded as follows. RDCSDATA[2:0] = 0b000: Transmit address bits [7:0] of the Serial Flash Memory address. RDCSDATA[2:0] = 0b001: Transmit address bits [15:8] of the Serial Flash Memory address. RDCSDATA[2:0] = 0b010: Transmit address bits [23:16] of the Serial Flash Memory address. RDCSDATA[2:0] = 0b011: Transmit address bits [31:24] of the Serial Flash Memory address. RDCSDATA[2:0] = 0b100: Dummy cycle(s) on the SDATA output lines for RDCSDATA [7:3] number of SCLK cycles. - RDCSDATA[7:3]= 0b00000 -> Dummy cycle for 1 SCLK cycle. - RDCSDATA[7:3]= 0b00001 -> Dummy cycles for 2 SCLK cycles. .... RDCSDATA[2:0] = 0b111: End of list.

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER52: DDR High Speed SPI Controller
5.Registers
5.23.DDRHSSPI Module ID Register (DDRHSSPIn_MID)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
<table>
<tr><td>BIT_OFFSET</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td>BIT_NAME</td><td>MID[15]</td><td>MID[14]</td><td>MID[13]</td><td>MID[12]</td><td>MID[11]</td><td>MID[10]</td><td>MID[9]</td><td>MID[8]</td></tr>
<tr><td>ACCESS_TYPE</td><td>R0,WX</td><td>R0,WX</td><td>R0,WX</td><td>R0,WX</td><td>R0,WX</td><td>R0,WX</td><td>R0,WX</td><td>R0,WX</td></tr>
<tr><td>PROT_TYPE</td><td colspan="8">-</td></tr>
<tr><td>INITIAL_VALUE</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0
or
1</td></tr>
</table>

(Correct)
<table>
<tr><td>BIT_OFFSET</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td>BIT_NAME</td><td>MID[15]</td><td>MID[14]</td><td>MID[13]</td><td>MID[12]</td><td>MID[11]</td><td>MID[10]</td><td>MID[9]</td><td>MID[8]</td></tr>
<tr><td>ACCESS_TYPE</td><td>R0,WX</td><td>R0,WX</td><td>R0,WX</td><td>R0,WX</td><td>R0,WX</td><td>R0,WX</td><td>R0,WX
or
R1,WX</td><td>R0,WX
or
R1,WX</td></tr>
<tr><td>PROT_TYPE</td><td colspan="8">-</td></tr>
<tr><td>INITIAL_VALUE</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0
or
1</td><td>0
or
1</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER52: DDR High Speed SPI Controller
5.Registers
5.23.DDRHSSPI Module ID Register (DDRHSSPIn_MID)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
The unique module ID number identifies the version of the DDRHSSPI used in the MCU.
<table>
<tr><th>MID[31:0]</th><th>Description</th></tr>
<tr><td>0x00000001</td><td>This version has limitations as below.
– Octal protocol (Dual Quad and Dual Legacy) is not supported.
➢ DDRHSSPIn_DMTRP.TRP[1:0] shall not be "11".
➢ DDRHSSPIn_CSCFG.MBM[1:0] shall not be "11".
– Sample Point Control function is not supported.
➢ DDRHSSPIn_SDATASAMPLEPTCNT0-7.SDATASMPTCNT shall be fixed to 0.
➢ DDRHSSPIn_SDATASAMPLEPTLFT0-7.SDATASMPTLFT shall be fixed to 0.
➢ DDRHSSPIn_SDATASAMPLEPTRGH0-7.SDATASMPTRGH shall be fixed to 0.
– DLP function is not supported.
➢ DDRHSSPIn_MCTRL.DLPEN shall be fixed to "0".</td></tr>
</table>

(Correct)
The unique module ID number identifies the version of the DDRHSSPI used in the MCU.
<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>0x00000001</td><td>This version has limitations as below.
– Octal protocol (Dual Quad and Dual Legacy) is not supported.
➢ DDRHSSPIn_DMTRP.TRP[1:0] shall not be 0b11.
➢ DDRHSSPIn_CSCFG.MBM[1:0] shall not be 0b11.
– Sample Point Control function is not supported.
➢ DDRHSSPIn_SDATASAMPLEPTCNT0-7.SDATASMPTCNT shall be fixed to 0.
➢ DDRHSSPIn_SDATASAMPLEPTLFT0-7.SDATASMPTLFT shall be fixed to 0.
➢ DDRHSSPIn_SDATASAMPLEPTRGH0-7.SDATASMPTRGH shall be fixed to 0.
– DLP function is not supported.
➢ DDRHSSPIn_MCTRL.DLPEN shall be fixed to "0".</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER52: DDR High Speed SPI Controller 5.Registers 5.23.DDRHSSPI Module ID Register (DDRHSSPIn_MID)</td><td>The item should be added as indicated by the shading below.

(Error)
The unique module ID number identifies the version of the DDRHSSPI used in the MCU.
<table><tr><td>0x00000100</td><td>This version supports the following features.
– Octal protocol (Dual Quad and Dual Legacy)
– Sample Point Control function
– DLP function</td></tr></table>
(Correct)
The unique module ID number identifies the version of the DDRHSSPI used in the MCU.
<table><tr><td>0x00000100</td><td>This version supports the following features.
– Octal protocol (Dual Quad and Dual Legacy)
– Sample Point Control function
– DLP function</td></tr><tr><td>0x00000300</td><td>The DDRHSSPIn_SDATASAMPLEPT* registers have been extended from six ([5:0]) to seven ([6:0]) bits,
All other features are exactly the same as MID = 0x00000100.</td></tr></table></td></tr>
<tr><td>CHAPTER52: DDR High Speed SPI Controller 5.Registers 5.25.DDRHSSPI SDATA Center Clock Sample Point Registers (DDRHSSPIn_SDATASAMPLEPTCNT0-7)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
This feature is available only when DDRHSSPIn_MID.MID is 0x00000100.

(Correct)
This feature is available only when DDRHSSPIn_MID.MID is 0x00000100 or 0x00000300.</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER52: DDR High Speed SPI Controller 5.Registers 5.25.DDRHSSPI SDATA Center Clock Sample Point Registers (DDRHSSPIn_SDATASAMPLEPTCNT0-7)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
<table>
<tr><td>BIT_OFFSET</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>BIT_NAME</td><td>Reserved</td><td>Reserved</td><td>SDATASMPTCNT[5]</td><td>SDATASMPTCNT[4]</td><td>SDATASMPTCNT[3]</td><td>SDATASMPTCNT[2]</td><td>SDATASMPTCNT[1]</td><td>SDATASMPTCNT[0]</td></tr>
<tr><td>ACCESS_TYPE</td><td>R0,W0</td><td>R0,W0</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td></tr>
<tr><td>PROT_TYPE</td><td colspan="8">-</td></tr>
<tr><td>INITIAL_VALUE</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
(Correct)
<table>
<tr><td>BIT_OFFSET</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>BIT_NAME</td><td>Reserved</td><td>SDATASMPTCNT[6]</td><td>SDATASMPTCNT[5]</td><td>SDATASMPTCNT[4]</td><td>SDATASMPTCNT[3]</td><td>SDATASMPTCNT[2]</td><td>SDATASMPTCNT[1]</td><td>SDATASMPTCNT[0]</td></tr>
<tr><td>ACCESS_TYPE</td><td>R0,W0</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td></tr>
<tr><td>PROT_TYPE</td><td colspan="8">-</td></tr>
<tr><td>INITIAL_VALUE</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table></td></tr>
<tr><td>CHAPTER52: DDR High Speed SPI Controller 5.Registers 5.25.DDRHSSPI SDATA Center Clock Sample Point Registers (DDRHSSPIn_SDATASAMPLEPTCNT0-7)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit7:6] Reserved : Reserved

[bit5:0] SDATASMPTCNT[5:0] : SDATA Sample Point Center Control

(Correct)
[bit7] Reserved: Reserved

[bit6:0] SDATASMPTCNT[6:0]: SDATA Sample Point Center Control</td></tr>
</table>

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER52: DDR High Speed SPI Controller
5.Registers
5.26.DDRHSSPI SDATA Left Clock Sample Point Registers (DDRHSSPIn_SDATASAMPLEPTLFT0-7)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
This feature is available only when DDRHSSPIn_MID.MID is 0x00000100.

(Correct)
This feature is available only when DDRHSSPIn_MID.MID is 0x00000100 or 0x00000300.</td></tr>
<tr><td>CHAPTER52: DDR High Speed SPI Controller
5.Registers
5.26.DDRHSSPI SDATA Left Clock Sample Point Registers (DDRHSSPIn_SDATASAMPLEPTLFT0-7)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
<table>
<tr><td>BIT_OFFSET</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>BIT_NAME</td><td>Reserved</td><td>Reserved</td><td>SDATASMPTLFT[5]</td><td>SDATASMPTLFT[4]</td><td>SDATASMPTLFT[3]</td><td>SDATASMPTLFT[2]</td><td>SDATASMPTLFT[1]</td><td>SDATASMPTLFT[0]</td></tr>
<tr><td>ACCESS_TYPE</td><td>R0,W0</td><td>R0,W0</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td></tr>
<tr><td>PROT_TYPE</td><td colspan="8">-</td></tr>
<tr><td>INITIAL_VALUE</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
(Correct)
<table>
<tr><td>BIT_OFFSET</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>BIT_NAME</td><td>Reserved</td><td>SDATASMPTLFT[6]</td><td>SDATASMPTLFT[5]</td><td>SDATASMPTLFT[4]</td><td>SDATASMPTLFT[3]</td><td>SDATASMPTLFT[2]</td><td>SDATASMPTLFT[1]</td><td>SDATASMPTLFT[0]</td></tr>
<tr><td>ACCESS_TYPE</td><td>R0,W0</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td></tr>
<tr><td>PROT_TYPE</td><td colspan="8">-</td></tr>
<tr><td>INITIAL_VALUE</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table></td></tr>
</table>

Section	Change Results
CHAPTER52: DDR High Speed SPI Controller 5.Registers 5.26.DDRHSSPI SDATA Left Clock Sample Point Registers (DDRHSSPIn_SDATASAMPLEPTLFT0-7)	Features of "For details" should be corrected as indicated by the shading below. (Error) [bit7:6] Reserved : Reserved [bit5:0] SDATASMPTLFT[5:0] : SDATA Left Clock Sample Point Control (Correct) [bit7] Reserved: Reserved [bit6:0] SDATASMPTLFT[6:0]: SDATA Left Clock Sample Point Control
CHAPTER52: DDR High Speed SPI Controller 5.Registers 5.27.DDRHSSPI SDATA Right Clock Sample Point Registers (DDRHSSPIn_SDATASAMPLEPTRGH0-7)	Features of "For details" should be corrected as indicated by the shading below. (Error) This feature is available only when DDRHSSPIn_MID.MID is 0x00000100. (Correct) This feature is available only when DDRHSSPIn_MID.MID is 0x00000100 or 0x00000300.

<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER52: DDR High Speed SPI Controller 5.Registers 5.27.DDRHSSPI SDATA Right Clock Sample Point Registers (DDRHSSPIn_SDATASAMPLEPTRGH0-7)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
<table>
<tr><td>BIT_OFFSET</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>BIT_NAME</td><td>Reserved</td><td>Reserved</td><td>SDATASMPTRGH[5]</td><td>SDATASMPTRGH[4]</td><td>SDATASMPTRGH[3]</td><td>SDATASMPTRGH[2]</td><td>SDATASMPTRGH[1]</td><td>SDATASMPTRGH[0]</td></tr>
<tr><td>ACCESS_TYPE</td><td>R0,W0</td><td>R0,W0</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td></tr>
<tr><td>PROT_TYPE</td><td colspan="8">-</td></tr>
<tr><td>INITIAL_VALUE</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
(Correct)
<table>
<tr><td>BIT_OFFSET</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>BIT_NAME</td><td>Reserved</td><td>SDATASMPTRGH[6]</td><td>SDATASMPTRGH[5]</td><td>SDATASMPTRGH[4]</td><td>SDATASMPTRGH[3]</td><td>SDATASMPTRGH[2]</td><td>SDATASMPTRGH[1]</td><td>SDATASMPTRGH[0]</td></tr>
<tr><td>ACCESS_TYPE</td><td>R0,W0</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td></tr>
<tr><td>PROT_TYPE</td><td colspan="8">-</td></tr>
<tr><td>INITIAL_VALUE</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table></td></tr>
<tr><td>CHAPTER52: DDR High Speed SPI Controller 5.Registers 5.27.DDRHSSPI SDATA Right Clock Sample Point Registers (DDRHSSPIn_SDATASAMPLEPTRGH0-7)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit7:6] Reserved : Reserved

[bit5:0] SDATASMPTRGH[5:0] : SDATA Right Clock Sample Point Control

(Correct)
[bit7] Reserved: Reserved

[bit6:0] SDATASMPTRGH[6:0]: SDATA Right Clock Sample Point Control</td></tr>
</table>

Section	Change Results
CHAPTER52: DDR High Speed SPI Controller 5.Registers 5.28.DDRHSSPI Data Learning Pattern Register (DDRHSSPIn_DLP)	Features of "For details" should be corrected as indicated by the shading below. (Error) This feature is available only when DDRHSSPIn_MID.MID is 0x00000100. (Correct) This feature is available only when DDRHSSPIn_MID.MID is 0x00000100 or 0x00000300.
CHAPTER52: DDR High Speed SPI Controller 5.Registers 5.29.DDRHSSPI Data Learning Pattern Sample Status Register (DDRHSSPIn_DLPSAMPLESTATUS)	Features of "For details" should be corrected as indicated by the shading below. (Error) This feature is available only when DDRHSSPIn_MID.MID is 0x00000100. (Correct) This feature is available only when DDRHSSPIn_MID.MID is 0x00000100 or 0x00000300.

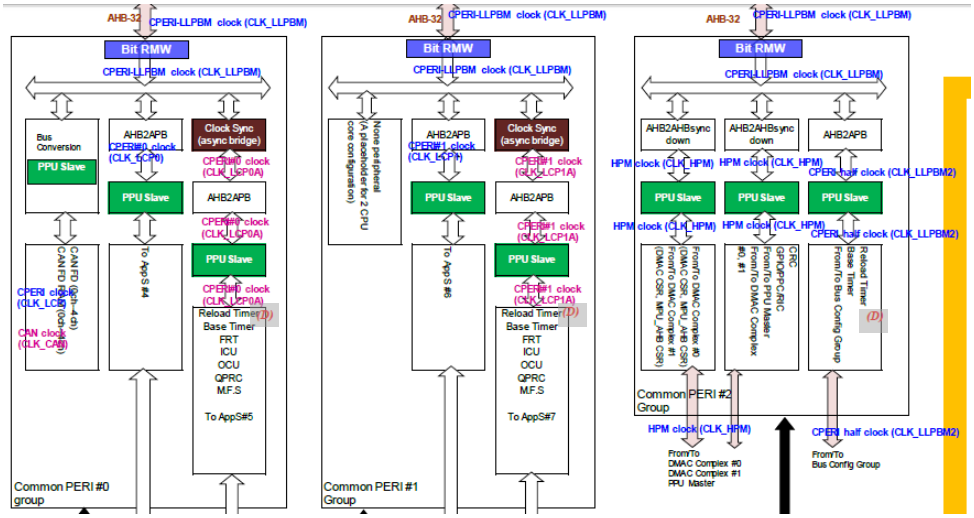
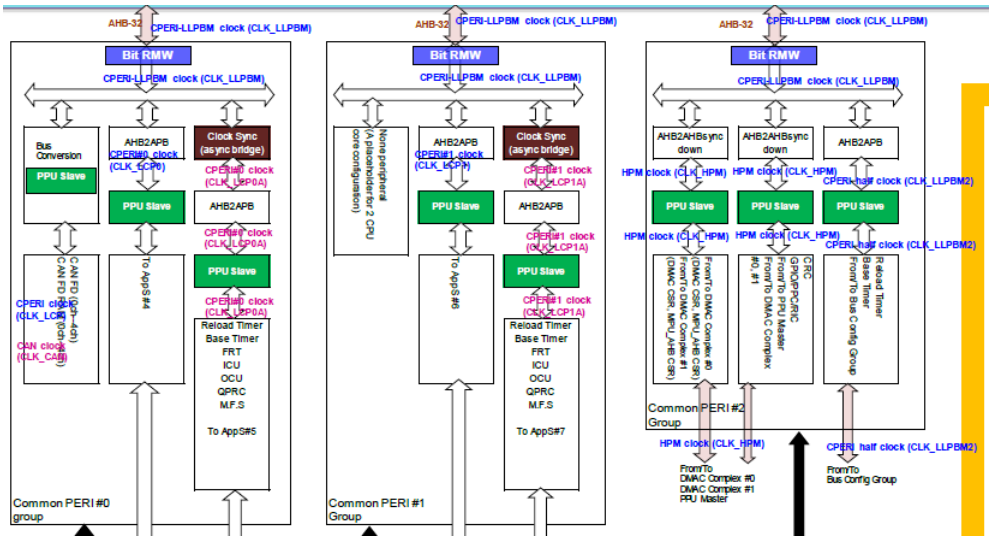
<table>
<tr><th>Section</th><th>Change Results</th></tr>
<tr><td>CHAPTER52: DDR High Speed SPI Controller 5.Registers 5.30.DDRHSSPI Command Sequencer Configuration Register (DDRHSSPIn_CSCFG)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
This field decides the bit width on the Serial Interface in Command Sequencer Mode.
<table>
<tr><th>MBM[1:0]</th><th>Description</th></tr>
<tr><td>00</td><td>This setting is prohibited.</td></tr>
<tr><td>01</td><td>This setting is prohibited.</td></tr>
<tr><td>10</td><td>Serial transfer works at Quad Protocol. Read data is sampled on SDATA[3:0]. Memory instruction, address, and other control information are transmitted on SDATA[3:0].</td></tr>
<tr><td>11</td><td>This value is available only when DDRHSSPIn_MID.MID is 0x00000100.
Serial transfer works at Dual Quad Protocol. Read data is sampled on SDATA[7:0]. Memory instruction, address, and other control information are transmitted on SDATA[7:0].</td></tr>
</table>

(Correct)
This field decides the bit width on the Serial Interface in Command Sequencer Mode.
<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>00</td><td>This setting is prohibited.</td></tr>
<tr><td>01</td><td>This setting is prohibited.</td></tr>
<tr><td>10</td><td>Serial transfer works at Quad Protocol. Read data is sampled on SDATA[3:0]. Memory instruction, address, and other control information are transmitted on SDATA[3:0].</td></tr>
<tr><td>11</td><td>This value is available only when DDRHSSPIn_MID.MID is 0x00000100 or 0x00000300.
Serial transfer works at Dual Quad Protocol. Read data is sampled on SDATA[7:0]. Memory instruction, address, and other control information are transmitted on SDATA[7:0].</td></tr>
</table>
</td></tr>
</table>

Page	Section	Change Results
002-04854 Rev. *C		
39	CHAPTER 1:Platform Overview 1.Overview	Error) ■Debug assist features Many of peripherals can be forcedly stopped by entering debug mode. Dedicated Reload Times can capture DMA control signal events like DREQ. EICU can capture pin status of peripherals. Correct) ■Debug assist features Following peripheral function can be forcedly stopped by entering debug mode. (1) DMA (2) Reload Timer in DMA Complex Subsystem (3) Real Time Clock (4) Hardware Watchdog Timer (5) Software Watchdog Timer (6) Time Protection Unit (7) Source Clock Timer Dedicated Reload Times can capture DMA control signal events like DREQ. EICU can capture pin status of peripherals.

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>41</td><td>CHAPTER 1:Platform Overview
1.Overview
1.1.Function Overview</td><td>
Error)
<table>
<tr><td>Main Flash memory (TCFlash)</td><td>1MB / 2MB / 4MB (configurable by option)
SECDED ECC by 8-byte via AXI
Read access via TCM port
Read and write access via AXI
TCM buffer for effective access via TCM port</td></tr>
<tr><td>Work Flash memory</td><td>64KB (when TCFlash is 1MB) / 128KB (when TCFlash is 2MB or 4MB)
SECDED ECC by 8-byte
Dedicated command sequencer for EEPROM emulation</td></tr>
<tr><td>Main SRAM</td><td>64KB / 128KB (configurable by option)
SECDED ECC</td></tr>
<tr><td>DMA Controller</td><td>Up to 2 instances
16 channels per instance
Up to 512 DMA client interfaces
Block / burst transfer
Fixed / dynamic / round-robin priority scheme</td></tr>
</table>
Correct)
<table>
<tr><td>Main Flash memory (TCFlash)</td><td>Up to 4MB (product specification)
SECDED ECC by 8-byte via AXI
Read access via TCM port
Read and write access via AXI
TCM buffer for effective access via TCM port</td></tr>
<tr><td>Work Flash memory</td><td>Up to 112KB (product specification)
SECDED ECC by 8-byte
Dedicated command sequencer for EEPROM emulation</td></tr>
<tr><td>TCRAM</td><td>Up to 128KB with SECDED ECC (product specification)
SECDED ECC</td></tr>
<tr><td>System RAM</td><td>Up to 384KB (product specification)
SECDED ECC</td></tr>
<tr><td>DMA Controller</td><td>Up to 2 instances
16 channels per instance
Up to 512 DMA client interfaces
Block / burst transfer
Fixed / dynamic / round-robin priority scheme</td></tr>
</table>
</td></tr>
<tr><td>43</td><td>CHAPTER 1:Platform Overview
2. Product Configuration</td><td>
Error)
Memories
<table>
<tr><th>Item</th><th>Size</th></tr>
<tr><td>Main Flash</td><td>2MB with SECDED ECC via AXI(configurable by option)</td></tr>
<tr><td>Work Flash</td><td>128KB with SECDED ECC (configurable by option)</td></tr>
<tr><td>TCRAM</td><td>128KB with SECDED ECC (configurable by option)</td></tr>
<tr><td>Scratch Pad RAM</td><td>128KB with SECDED ECC (configurable by option)</td></tr>
<tr><td>Backup RAM</td><td>8KB + 8KB with SECDED ECC (configurable by option)</td></tr>
<tr><td>BootROM</td><td>16KB (configurable by option)</td></tr>
</table>
Correct)
Memories
<table>
<tr><th>Item</th><th>Size</th></tr>
<tr><td>Main Flash</td><td>Up to 4MB with SECDED ECC via AXI(product specification)</td></tr>
<tr><td>Work Flash</td><td>Up to 112KB with SECDED ECC (product specification)</td></tr>
<tr><td>TCRAM</td><td>Up to 128KB with SECDED ECC (product specification)</td></tr>
<tr><td>System RAM</td><td>Up to 384KB with SECDED ECC (product specification)</td></tr>
<tr><td>Backup RAM</td><td>Up to 8KB + 8KB with SECDED ECC (product specification)</td></tr>
<tr><td>BootROM</td><td>Up to 16KB (product specification)</td></tr>
</table>
</td></tr>
</table>

Page	Section	Change Results
44	CHAPTER 1:Platform Overview 3. Configuration	Error) **Figure 3-1 Block Diagram**  Correct) **Figure 3-1 Block Diagram** 

Page	Section	Change Results
64	CHAPTER 2: CPU 2. Notes	Add the below: Correct) **Build Option** The build option of Cortex-R5 is shown below. **Table 2-1 Cortex-R5 Build Option**
79	CHAPTER 4:Reset 2. Configuration	Revised the below: Figure 2-1 Reset System Block Diagram
80	CHAPTER 4:Reset 2. Configuration	Revised the below: Figure 2-2 RST Manage Block Diagram
104	CHAPTER 4:Reset 3. Operational Description 3.1.21. Software Reset	Error) − *In case of generating the software reset, please write to SYSC_RSTCNTR0/1 after clearing the hardware watchdog counter in advance. After writing to SYSC_RSTCNTR0/1, please do not clear the watchdog counter until the reset is released.* Correct) − *In case of generating the software reset, please write to SYSC_RSTCNTR after clearing the hardware watchdog counter in advance. After writing to SYSC_RSTCNTR, please do not clear the watchdog counter until the reset is released.*

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>115</td><td>CHAPTER 4:Reset
3. Operational Description
3.2.2. I/O Reset</td><td>Error)
Notes:
− *1: It is controlled by IO3RSTCNT in system special setting register (SYSC0_SPECFGR).
− *2: It is controlled by IO35RSTCNT in system special setting register (SYSC0_SPECFGR)
− *3: It is activated by the transition to the PSS in case of PSSPADCTRL="1" in system special setting register (SYSC0_SPECFGR).

Correct)
Notes:
− *1: It is controlled by IO3RSTC in system special setting register (SYSC0_SPECFGR).
− *2: It is controlled by IO35RSTC in system special setting register (SYSC0_SPECFGR)
− *3: It is activated by the transition to the PSS in case of PSSPADCTRL="1" in system special setting register (SYSC0_SPECFGR).</td></tr>
<tr><td>116</td><td>CHAPTER 4:Reset
3. Operational Description
3.2.3. Reset of External Power Supply Control</td><td>Error)
Table 3-31 Reset of External Power Supply Control
<table>
<tr><th>Signal Name</th><th>Initialization Target</th><th>Reset Category</th><th>Reset Factor</th></tr>
<tr><td rowspan="2"></td><td rowspan="2">1,2V external power supply control</td><td>Hard reset</td><td></td></tr>
<tr><td>Standby transition reset</td><td></td></tr>
<tr><td rowspan="2">EX5VRST</td><td rowspan="2">5/3/3V external power supply control</td><td>Hard reset</td><td></td></tr>
<tr><td>Standby transition reset</td><td></td></tr>
</table>
Correct)
Table 3-31 Reset of External Power Supply Control
<table>
<tr><th>Signal Name</th><th>Initialization Target</th><th>Reset Category</th><th>Reset Factor</th></tr>
<tr><td rowspan="2"></td><td rowspan="2">1.2V external power supply control</td><td>Hard reset</td><td></td></tr>
<tr><td>Standby transition reset</td><td></td></tr>
<tr><td rowspan="2">EX5VRST</td><td rowspan="2">5/3.3V external power supply control</td><td>Hard reset</td><td></td></tr>
<tr><td>Standby transition reset</td><td></td></tr>
</table></td></tr>
</table>

Page	Section	Change Results
117	CHAPTER 4:Reset 3. Operational Description 3.3. Reset Sequence	Revised the below: Figure 3-1 Reset Sequence
118	CHAPTER 4:Reset 3. Operational Description 3.3. Reset Sequence	Error) 2. Waiting for the clock stop Before a reset, clock source is stopped for the resets which need RAM to be secured. Timeover will occur if the clock does not stop in the specified time. Correct) 2. Waiting for the clock stop Before a reset, clock source is stopped for the resets which need RAM(TCRAM, Backup RAM, System RAM) to be guaranteed. Timeover will occur if the clock does not stop in the specified time.

Page	Section	Change Results
119	CHAPTER 4:Reset 3. Operational Description 3.4. Operations after Reset Release	Revised the below:

Table 3-32 Operations after Reset Release

Error)

Hardware Reset	Clock stop wait timeout reset Main clock supervisor reset Sub-clock supervisor reset PLL clock supervisor reset SSCG clock supervisor reset	N/A	Yes	Yes	Fast CR	No
	Hardware watchdog reset Extended internal power supply low-voltage detection reset	FLASH 1.2V external power supply control	Yes	Yes	Fast CR	Yes*1
	RSTX pin input reset Software watchdog reset External power supply low-voltage detection reset Extended external Power supply low-voltage detection reset Profile error reset Software trigger hard reset	N/A	Yes	Yes	Fast CR	Yes
	nSRST pin input reset	N/A	No	No	No change.	Yes
Software Reset	Software reset	N/A	No	No	No change	Yes
Debugger reset	TRSTX pin input reset Software debugger reset	N/A	No	No	No change	Yes*2

Correct)

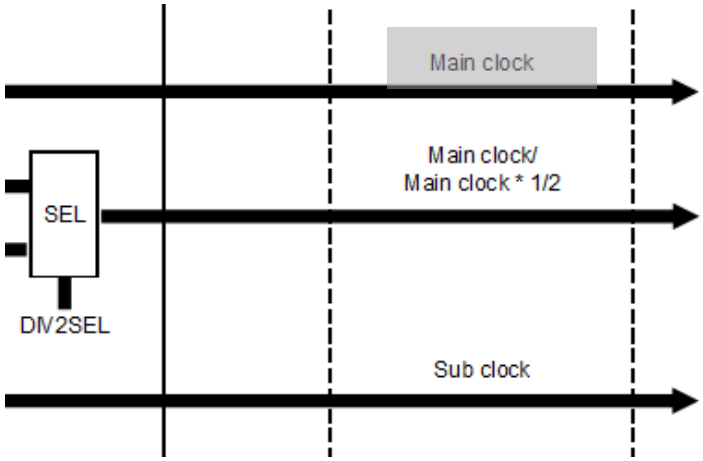
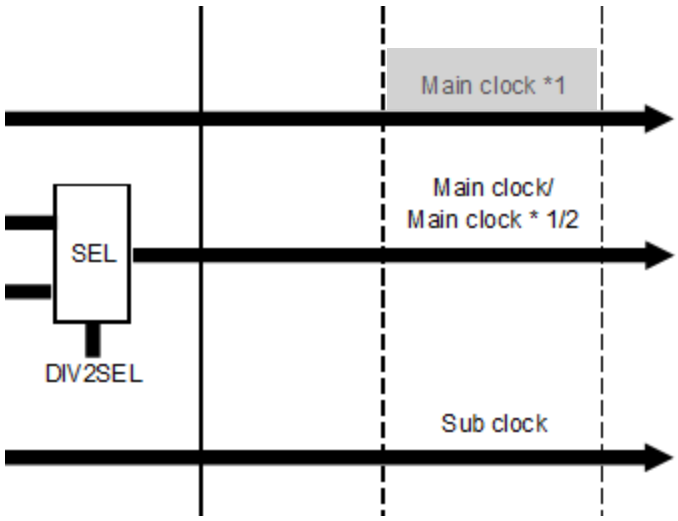
Hardware Reset	Clock stop wait timeout reset Main clock supervisor reset Sub-clock supervisor reset PLL clock supervisor reset SSCG clock supervisor reset	N/A	Yes	Yes	Fast CR	No
	Hardware watchdog reset Extended internal power supply low-voltage detection reset	FLASH 1.2V external power supply control	Yes	Yes	Fast CR	Yes*1
	RSTX pin input reset Software watchdog reset External power supply low-voltage detection reset Extended external Power supply low-voltage detection reset Profile error reset Software trigger hard reset	N/A	Yes	Yes	Fast CR	Yes*3
	nSRST pin input reset	N/A	No	No	No change.	Yes*3
Software Reset	Software reset	N/A	No	No	No change	Yes*3
Debugger reset	TRSTX pin input reset Software debugger reset	N/A	No	No	No change	Yes*2

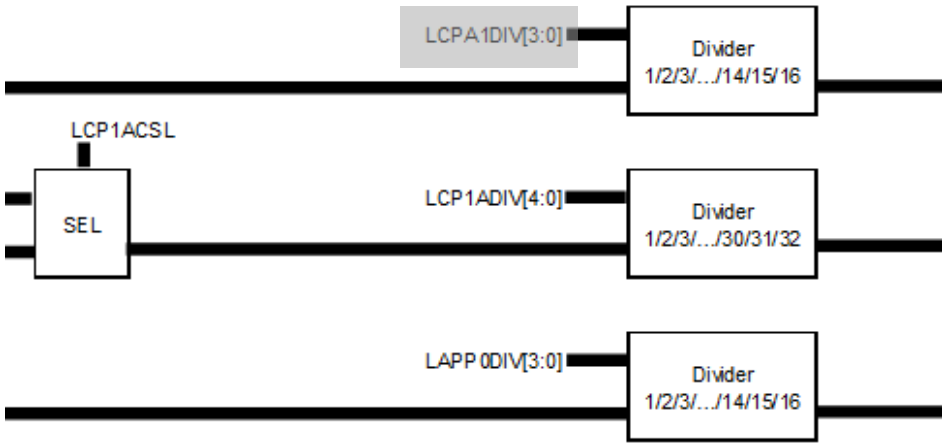
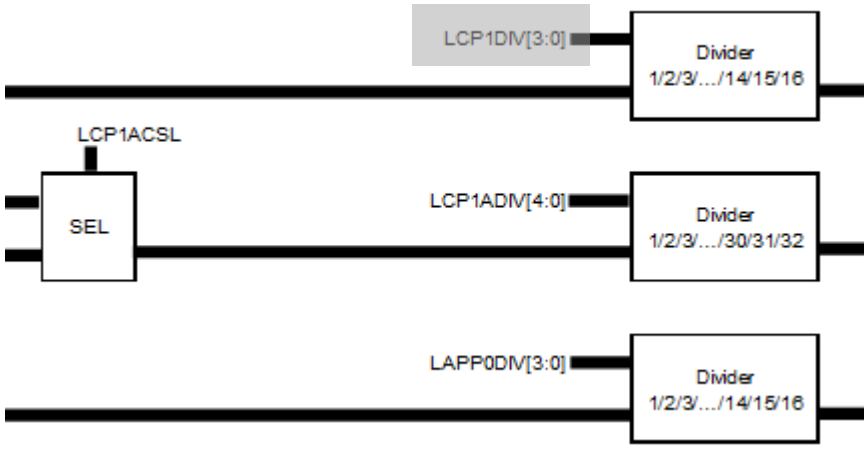
Page	Section	Change Results
120	CHAPTER 4:Reset 3. Operational Description 3.4. Operations after Reset Release	Error) *Notes:* *− *1: For a domain that operates with a 1.2 V external power supply, the RAM contents are not assured after a wait for stabilization of the 1.2 V external power supply. .* *− *2: After a software debugger reset is released, the ETB (Embedded Trace Buffer) RAM contents are not guaranteed.* *− nSRST pin input reset cannot be used. Therefore, please ignore the description about nSRST in RESET chapter.* Correct) *Notes:* *− *1: Contents of the Backup RAM are guaranteed. The other RAM contents are not guaranteed.* *− *2: The reset does not impact to RAM contents. But ETB (Embedded Trace Buffer) RAM contents are not guaranteed in case of Software debugger reset..* *− *3: Contents of TCRAM, System SRAM and Backup RAM are guaranteed. The other RAM contents are not guaranteed.* *− nSRST pin input reset cannot be used. Therefore, please ignore the description about nSRST in RESET chapter.*
124	CHAPTER 4:Reset 3. Operational Description 3.4.1. Hard Reset Operation	Error) Figure 4.3-5 Operation at Hardware Watchdog Reset or Extended Internal Power Supply Low-Voltage Reset Correct) Figure 3-5 Operation at Hardware Watchdog Reset
125	CHAPTER 4:Reset 3. Operational Description 3.4.1. Hard Reset Operation	Added the below: Figure 3-6 Extended Internal Power Supply Low-Voltage Reset
126	CHAPTER 4:Reset 3. Operational Description 3.4.1. Hard Reset Operation	Added the below: Figure 3-7 Operation at RSTX Pin Input Reset or External power supply low-voltage detection reset

Page	Section	Change Results
129	CHAPTER 4:Reset 4. Registers	Error) − *: When PD2 voltage is internal LDO, initial value is "1". When PD2 voltage is external LDO initial value is "0". Correct) − *: When PD2 voltage is internal LDO, initial value is "0". When PD2 voltage is external LDO initial value is "1".
137	CHAPTER 4:Reset 4. Registers 4.2. User Reset Factor Register (SYSC_RSTCAUSEUR)	Added the below: ***Notes:*** − Initial value of the bit is undefined when the electrical characteristics specification (both DC and AC) of the power supply is not kept.
138	CHAPTER 4:Reset 4. Registers 4.3. User Extended CSV Reset Factor Register (SYSC_EXCSVRSTCAUSEUR)	Error) [bit7] CSVSR 3: SSCG3 Clock Supervisor Reset Detection Bit [bit6] CSVSR 2: SSCG2 Clock Supervisor Reset Detection Bit [bit5] CSVSR 1: SSCG1 Clock Supervisor Reset Detection Bit Correct) [bit7] CSVSR3: SSCG3 Clock Supervisor Reset Detection Bit [bit6] CSVSR2: SSCG2 Clock Supervisor Reset Detection Bit [bit5] CSVSR1: SSCG1 Clock Supervisor Reset Detection Bit

Page	Section	Change Results
140 to 142	CHAPTER 4:Reset 4. Registers 4.4. User PowerDomain Reset Factor Register (SYSC_PDRSTCAUSEUR)	Error) [bit17] PR6R1: PowerDomain6_1 Reset Detection Bit [bit16] PR6R0: PowerDomain6_0 Reset Detection Bit [bit15] PR5R3: PowerDomain5_3 Reset Detection Bit [bit14] PR5R2: PowerDomain5_2 Reset Detection Bit [bit13] PR5R1: PowerDomain5_1 Reset Detection Bit [bit12] PR5R1: PowerDomain5_0 Reset Detection Bit [bit9] PR4R1: PowerDomain4_1 Reset Detection Bit [bit8] PR4R0: PowerDomain4_0 Reset Detection Bit [bit4] PR3R0: PowerDomain3_0 Reset Detection Bit Correct) [bit17] PD6R1: PowerDomain6_1 Reset Detection Bit [bit16] PD6R0: PowerDomain6_0 Reset Detection Bit [bit15] PD5R3: PowerDomain5_3 Reset Detection Bit [bit14] PD5R2: PowerDomain5_2 Reset Detection Bit [bit13] PD5R1: PowerDomain5_1 Reset Detection Bit [bit12] PD5R0: PowerDomain5_0 Reset Detection Bit [bit9] PD4R1: PowerDomain4_1 Reset Detection Bit [bit8] PD4R0: PowerDomain4_0 Reset Detection Bit [bit4] PD3R0: PowerDomain3_0 Reset Detection Bit
149	CHAPTER 4:Reset 4. Registers 4.5. BootROM Reset Factor Register (SYSC_RSTCAUSEBT)	Added the below: ***Notes:*** *− Initial value of the bit is undefined when the electrical characteristics specification (both DC an* *AC) of the power supply is not kept.* *− BootROM is used by this register. BootROM initializes this register in the startup process. Can* *not be read "1" set the status of each reset factor from the user program.*
152	CHAPTER 4:Reset 4. Registers 4.7. BootROM PowerDomain Reset Factor Register (SYSC_PDRSTCAUSEBT)	Error) *Note:* *− *: When PD2 voltage is internal LDO, initial value is "1". When PD2 voltage is external LDO initial value is "0".* Correct) *Note:* *− *: When PD2 voltage is internal LDO, initial value is "0". When PD2 voltage is external LDO initial value is "1".*

Page	Section	Change Results
152 to 154	CHAPTER 4:Reset 4. Registers 4.7. BootROM PowerDomain Reset Factor Register (SYSC_PDRSTCAUSEBT)	Error) [bit17] PR6R1: PowerDomain6_1 Reset Detection Bit [bit16] PR6R0: PowerDomain6_0 Reset Detection Bit [bit15] PR5R3: PowerDomain5_3 Reset Detection Bit [bit14] PR5R2: PowerDomain5_2 Reset Detection Bit [bit13] PR5R1: PowerDomain5_1 Reset Detection Bit [bit12] PR5R1: PowerDomain5_0 Reset Detection Bit [bit9] PR4R1: PowerDomain4_1 Reset Detection Bit [bit8] PR4R0: PowerDomain4_0 Reset Detection Bit [bit4] PR3R0: PowerDomain3_0 Reset Detection Bit Correct) [bit17] PD6R1: PowerDomain6_1 Reset Detection Bit [bit16] PD6R0: PowerDomain6_0 Reset Detection Bit [bit15] PD5R3: PowerDomain5_3 Reset Detection Bit [bit14] PD5R2: PowerDomain5_2 Reset Detection Bit [bit13] PD5R1: PowerDomain5_1 Reset Detection Bit [bit12] PD5R0: PowerDomain5_0 Reset Detection Bit [bit9] PD4R1: PowerDomain4_1 Reset Detection Bit [bit8] PD4R0: PowerDomain4_0 Reset Detection Bit [bit4] PD3R0: PowerDomain3_0 Reset Detection Bit
156	CHAPTER 4:Reset 4. Registers 4.8.PowerDomain Reset Status Register (SYSC_PDRSTSTATUS)	Error) [bit12] PD5RS1: PowerDomain5_0 Reset Detection Bit Correct) [bit12] PD5RS0: PowerDomain5_0 Reset Detection Bit

Page	Section	Change Results
165	CHAPTER 5:Clock System 2. Configuration and Block Diagram 2.2. Clock Generator Configuration and Block Diagram	Error) **Figure 2 2 Clock Generator Block Diagram**  Correct) **Figure 2 2 Clock Generator Block Diagram**  *1 : Real Time Clock and Software Watchdog use this as their Main clock source.

Page	Section	Change Results
168 To 169	CHAPTER 5:Clock System 2. Configuration and Block Diagram 2.3. Clock Distributer Configuration and Block Diagram	Error) **Figure 2 5 Clock domain0 Block Diagram**  Correct) **Figure 2 5 Clock domain0 Block Diagram**  ***Note:*** *− Do not use Sub clock for the domain clock.*

Page	Section	Change Results
169	CHAPTER 5:Clock System 2. Configuration and Block Diagram 2.3. Clock Distributer Configuration and Block Diagram	Added the below: Figure 2-7 Clock domainx(x=1,2,3,4,5) Block Diagram ***Note:*** − *Do not use Sub clock for the domain clock*
170	CHAPTER 5:Clock System 2. Configuration and Block Diagram 2.3. Clock Distributer Configuration and Block Diagram	Added the below: Figure 2-8 Clock Domain HSSPI Block Diagram ***Note:*** − *Do not use Sub clock for the domain clock.*
170	CHAPTER 5:Clock System 2. Configuration and Block Diagram 2.3. Clock Distributer Configuration and Block Diagram	Added the below: Figure 2-9 Clock Domain MCUC Block Diagram ***Note:*** − *Do not use Sub clock for the domain clock.*

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>171</td><td>CHAPTER 5:Clock System
2. Configuration and Block Diagram
2.3. Clock Distributer Configuration and Block Diagram</td><td>Added the below:

Figure 2-10 Clock Domain CLKO Block Diagram

Note:
− Do not use Sub clock for the domain clock.</td></tr>
<tr><td>173</td><td>CHAPTER 5:Clock System
3. Operations
3.1. Source Clock Generation</td><td>Error)
Table 3-1 Source Clock Enable/Disable

<table><tr><th>Source Clock</th><th>Initial State</th><th>RUN</th><th>PSS</th></tr><tr><td>Sub clock</td><td>Disable</td><td>Programmable</td><td>Programmable</td></tr></table>
Correct)
<table><tr><th>Source Clock</th><th>Initial State</th><th>RUN</th><th>PSS</th></tr><tr><td>Sub clock</td><td>(product specification)</td><td>Programmable</td><td>Programmable</td></tr></table></td></tr>
<tr><td>174</td><td>CHAPTER 5:Clock System
3. Operations
3.2. PLL/SSCG PLL Clock</td><td>Error)
2. Set the RUN/PSS profile register for PLLx configuration.(SYSC_RUNPLLxCNTR/SYSC_PSSPLLxCNTR)
3. Set the RUN/PSS profile register for PLLxEN (SYSC_RUNCKSRER/SYSC_PSSCKSRER)

Correct)
2. Set the RUN/PSS profile register for PLLx configuration.(SYSC0_RUNPLLxCNTR/SYSC0_PSSPLLxCNTR)
3. Set the RUN/PSS profile register for PLLxEN SYSC0_RUNCKSRER/SYSC0_PSSCKSRER)</td></tr>
</table>

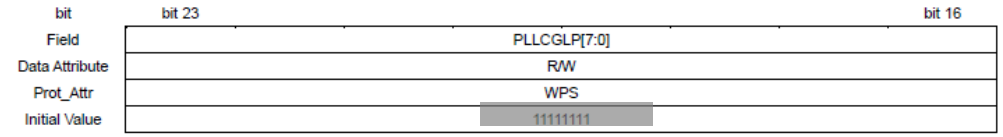
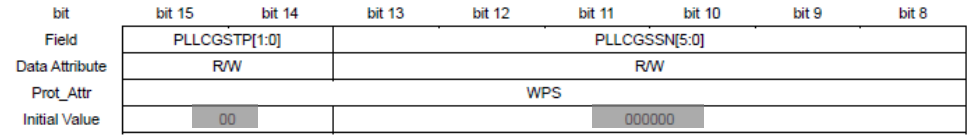
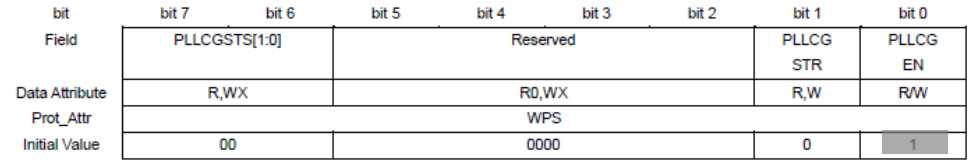
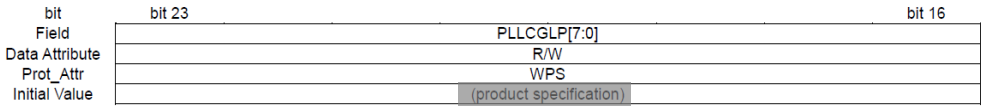
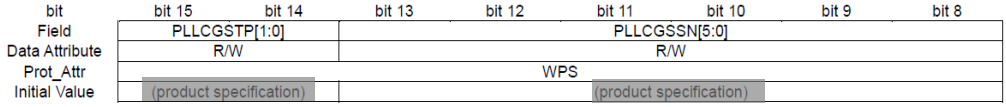
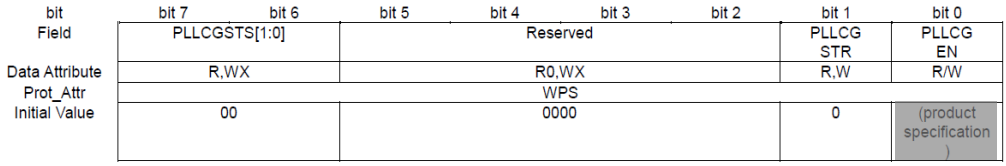
Page	Section	Change Results
174	CHAPTER 5:Clock System 3. Operations 3.2. PLL/SSCG PLL Clock	Error) 2. Set the RUN/PSS profile register for SSCG PLLx configuration.(SYSC_RUNSSCGxCNTR0 and SYSC_RUNSSCGxCNTR1/SYSC_PSSSSCGxCNTR and SYSC_PSSSSCGxCNTR) 3. Set the RUN/PSS profile register for SSCG PLLxEN (SYSC_RUNCKSRER/SYSC_PSSCKSRER) Correct) 2. Set the RUN/PSS profile register for SSCG PLLx configuration.(SYSC0_RUNSSCGxCNTR0 and SYSC0_RUNSSCGxCNTR1/SYSC0_PSSSSCGxCNTR0 and SYSC0_PSSSSCGxCNTR1) 3. Set the RUN/PSS profile register for SSCG PLLxEN (SYSC0_RUNCKSRER/SYSC0_PSSCKSRER)

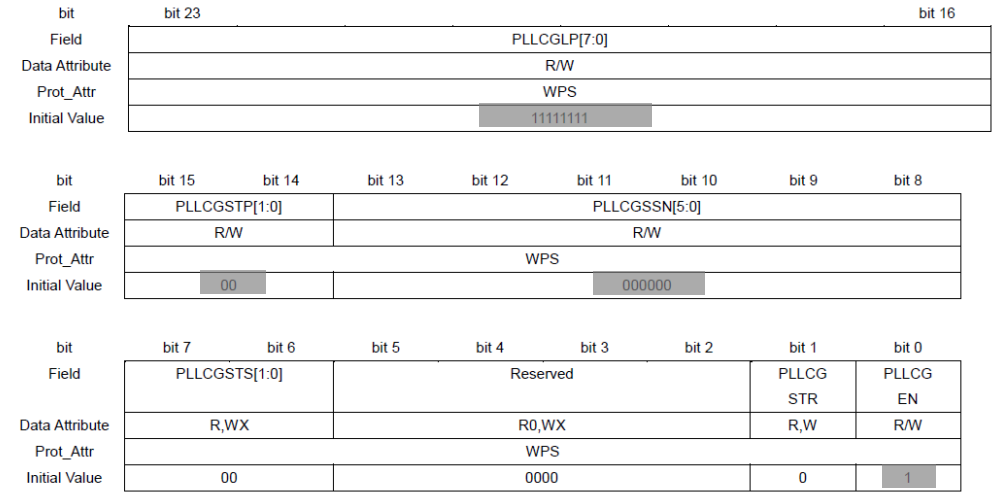
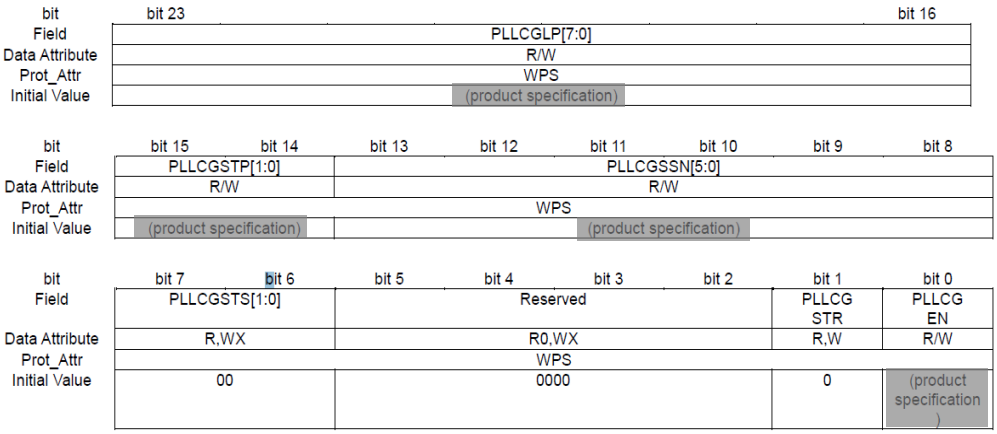
<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>179</td><td>CHAPTER 5:Clock System
3. Operations
3.3. Clock Distributer</td><td>
Error)
Table 3 5 Clock Divider Configuration and Maximum Frequency
<table>
<tr><th>Clock Domain</th><th>Internal Clock</th><th>Resources</th><th>Initial State</th><th>RUN/PSS</th><th>Maximum Frequency</th></tr>
<tr><td rowspan="20">CD0</td><td>CLK_HAPP0A0</td><td>Application Specific0 (HPM)</td><td>1</td><td>Programmable</td><td>120MHz</td></tr>
<tr><td>CLK_HAPP0A1</td><td>1</td><td>Programmable</td><td>60MHz</td></tr>
<tr><td>CLK_HAPP1B0</td><td>Application Specific1 (HPM)</td><td>1</td><td>Programmable</td><td>120MHz</td></tr>
<tr><td>CLK_HAPP1B1</td><td>1</td><td>Programmable</td><td>60MHz</td></tr>
<tr><td>CLK_LLPBM</td><td>Common Peripheral (LLPBM)</td><td>1</td><td>Programmable</td><td>240MHz</td></tr>
<tr><td>CLK_LLPBM2</td><td>Common Peripheral0/1/2</td><td>2</td><td>2</td><td>120MHz</td></tr>
<tr><td>CLK_LCP</td><td>Common Peripheral0/1</td><td>1</td><td>Programmable</td><td>120MHz</td></tr>
<tr><td>CLK_LCP0</td><td>Common Peripheral0</td><td>1</td><td>Programmable</td><td>120MHz</td></tr>
<tr><td>CLK_LCP0A</td><td>1</td><td>Programmable</td><td>120MHz</td></tr>
<tr><td>CLK_LCP1</td><td>Common Peripheral1</td><td>1</td><td>Programmable</td><td>120MHz</td></tr>
<tr><td>CLK_LCP1A</td><td>1</td><td>Programmable</td><td>120MHz</td></tr>
<tr><td>CLK_LAPP0</td><td>Application Specific0 (LLPPBM)</td><td>1</td><td>Programmable</td><td>120MHz</td></tr>
<tr><td>CLK_LAPP0A</td><td>1</td><td>Programmable</td><td>120MHz</td></tr>
<tr><td>CLK_LAPP1</td><td>Application Specific1 (LLPPBM)</td><td>1</td><td>Programmable</td><td>120MHz</td></tr>
<tr><td>CLK_LAPP1ACLK_ LAPP0A1</td><td>1</td><td>Programmable</td><td>120MHz</td></tr>
</table>
Correct)
Table 3 5 Clock Divider Configuration and Maximum Frequency
<table>
<tr><th>Clock Domain</th><th>Internal Clock</th><th>Resources</th><th>Initial State</th><th>RUN/PSS</th><th>Maximum Frequency</th></tr>
<tr><td rowspan="20">CD0</td><td>CLK_HAPP0A0</td><td>Application Specific0 (HPM)</td><td>1</td><td>Programmable</td><td>120MHz</td></tr>
<tr><td>CLK_HAPP0A1</td><td>1</td><td>Programmable</td><td>60MHz</td></tr>
<tr><td>CLK_HAPP1B0</td><td>Application Specific1 (HPM)</td><td>1</td><td>Programmable</td><td>120MHz</td></tr>
<tr><td>CLK_HAPP1B1</td><td>1</td><td>Programmable</td><td>60MHz</td></tr>
<tr><td>CLK_LLPBM</td><td>Common Peripheral (LLPBM)</td><td>1</td><td>Programmable</td><td>240MHz</td></tr>
<tr><td>CLK_LLPBM2</td><td>Common Peripheral0/1/2</td><td>2</td><td>2</td><td>120MHz</td></tr>
<tr><td>CLK_LCP</td><td>Common Peripheral0/1</td><td>1</td><td>Programmable</td><td>120MHz</td></tr>
<tr><td>CLK_LCP0</td><td>Common Peripheral0</td><td>1</td><td>Programmable</td><td>120MHz</td></tr>
<tr><td>CLK_LCP0A</td><td>1</td><td>Programmable</td><td>120MHz</td></tr>
<tr><td>CLK_LCP1</td><td>Common Peripheral1</td><td>1</td><td>Programmable</td><td>120MHz</td></tr>
<tr><td>CLK_LCP1A</td><td>1</td><td>Programmable</td><td>120MHz</td></tr>
<tr><td>CLK_LAPP0</td><td>Application Specific0 (LLPPBM)</td><td>1</td><td>Programmable</td><td>120MHz</td></tr>
<tr><td>CLK_LAPP0A</td><td>1</td><td>Programmable</td><td>120MHz</td></tr>
<tr><td>CLK_LAPP1</td><td>Application Specific1 (LLPPBM)</td><td>1</td><td>Programmable</td><td>120MHz</td></tr>
<tr><td>CLK_LAPP1A</td><td>1</td><td>Programmable</td><td>120MHz</td></tr>
</table>
</td></tr>
</table>

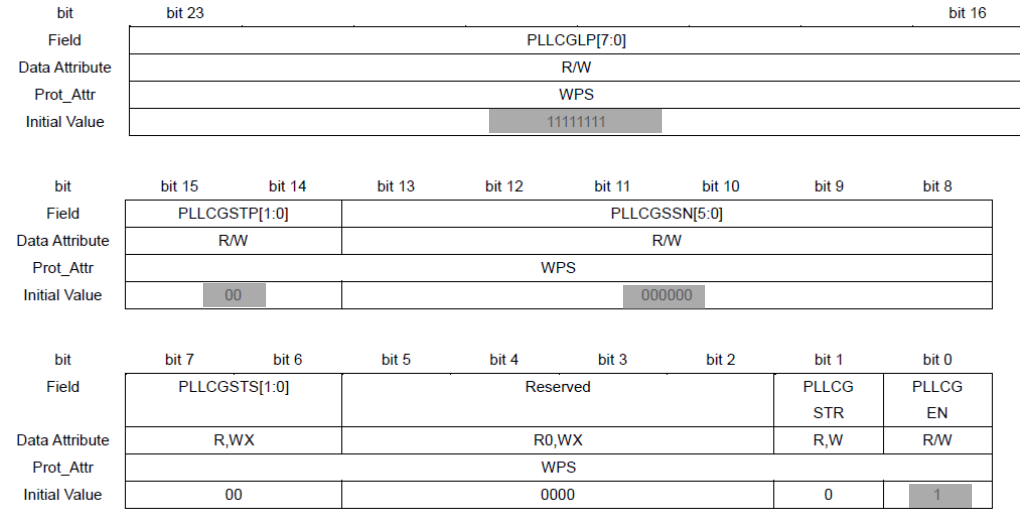
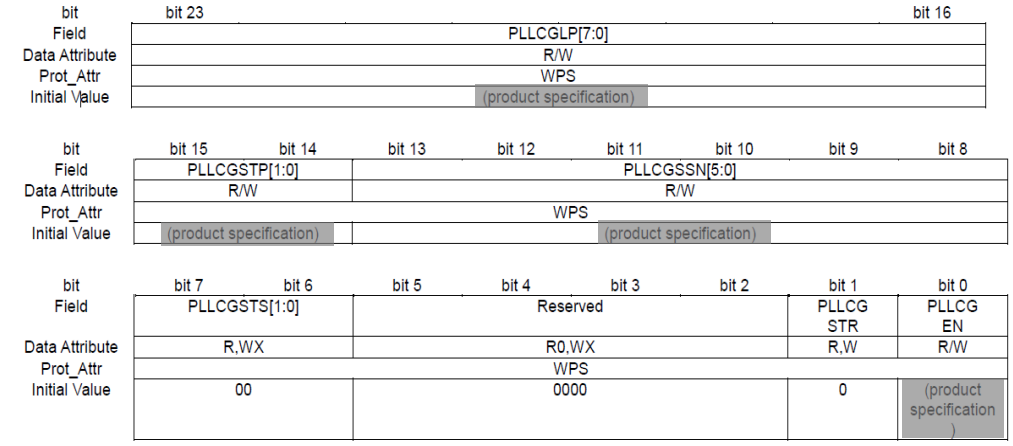
<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>188</td><td>CHAPTER 5:Clock System
4. Setting Procedure Example
4.1. RUN Configuration</td><td>Revised the Below:
Figure 4-2 Setting Procedure Example for Using PLL Clock</td></tr>
<tr><td>192</td><td>CHAPTER 5:Clock System
5. Registers
5.1. Common Configuration Registers</td><td>Error)
5.1.1. CR Clock Control Register(SYSC_CRCNTR)
<table>
<tr><td>bit</td><td>bit 15</td><td>bit 14</td><td>bit 13</td><td>bit 12</td><td>bit 11</td><td>bit 10</td><td>bit 9</td><td>bit 8</td></tr>
<tr><td>Field</td><td colspan="3">Reserved</td><td colspan="5">TRC[4:0]</td></tr>
<tr><td>Data Attribute</td><td colspan="3">R0,WX</td><td colspan="5">R/W</td></tr>
<tr><td>Prot_Attr</td><td colspan="8">WPS</td></tr>
<tr><td>Initial Value</td><td colspan="3">000</td><td colspan="5">00000</td></tr>
</table>
<table>
<tr><td>bit</td><td>bit 7</td><td>bit 6</td><td>bit 5</td><td>bit 4</td><td>bit 3</td><td>bit 2</td><td>bit 1</td><td>bit 0</td></tr>
<tr><td>Field</td><td colspan="3">Reserved</td><td colspan="5">TRF[4:0]</td></tr>
<tr><td>Data Attribute</td><td colspan="3">R0,WX</td><td colspan="5">R/W</td></tr>
<tr><td>Prot_Attr</td><td colspan="8">WPS</td></tr>
<tr><td>Initial Value</td><td colspan="3">000</td><td colspan="5">00000</td></tr>
</table>
Correct)
<table>
<tr><td>bit</td><td>bit 15</td><td>bit 14</td><td>bit 13</td><td>bit 12</td><td>bit 11</td><td>bit 10</td><td>bit 9</td><td>bit 8</td></tr>
<tr><td>Field</td><td colspan="3">Reserved</td><td colspan="5">TRC[4:0]</td></tr>
<tr><td>Data Attribute</td><td colspan="3">R0,WX</td><td colspan="5">R/W</td></tr>
<tr><td>Prot_Attr</td><td colspan="8">WPS</td></tr>
<tr><td>Initial Value</td><td colspan="3">000</td><td colspan="5">(product specification)</td></tr>
</table>
<table>
<tr><td>bit</td><td>bit 7</td><td>bit 6</td><td>bit 5</td><td>bit 4</td><td>bit 3</td><td>bit 2</td><td>bit 1</td><td>bit 0</td></tr>
<tr><td>Field</td><td colspan="3">Reserved</td><td colspan="5">TRF[4:0]</td></tr>
<tr><td>Data Attribute</td><td colspan="3">R0,WX</td><td colspan="5">R/W</td></tr>
<tr><td>Prot_Attr</td><td colspan="8">WPS</td></tr>
<tr><td>Initial Value</td><td colspan="3">000</td><td colspan="5">(product specification)</td></tr>
</table></td></tr>
<tr><td>193</td><td>CHAPTER 5:Clock System
5. Registers
5.1. Common Configuration Registers
5.1.2. Main Oscillator Control Register(SYSC_MOSCCNTR)</td><td>Error)
<table>
<tr><td>bit</td><td>bit 31</td><td>bit 30</td><td>bit 29</td><td>bit 28</td><td>bit 27</td><td>bit 26</td><td>bit 25</td><td>bit 24</td></tr>
<tr><td>Field</td><td>MCMODE</td><td colspan="3">Reserved</td><td colspan="2">MCGAIN[1:0]</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>Data Attribute</td><td>R/W</td><td colspan="3">R0,WX</td><td colspan="2">R/W</td><td>R0,WX</td><td>R/W0</td></tr>
<tr><td>Prot_Attr</td><td colspan="8">WPS</td></tr>
<tr><td>Initial Value</td><td>0</td><td colspan="3">000</td><td colspan="2">00</td><td>0</td><td>0</td></tr>
</table>
Correct)
<table>
<tr><td>bit</td><td>bit 31</td><td>bit 30</td><td>bit 29</td><td>bit 28</td><td>bit 27</td><td>bit 26</td><td>bit 25</td><td>bit 24</td></tr>
<tr><td>Field</td><td>MCMODE</td><td colspan="3">Reserved</td><td colspan="2">MCGAIN[1:0]</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>Data Attribute</td><td>R/W</td><td colspan="3">R0,WX</td><td colspan="2">R/W</td><td>R0,WX</td><td>R/W0</td></tr>
<tr><td>Prot_Attr</td><td colspan="8">WPS</td></tr>
<tr><td>Initial Value</td><td>(product specification)</td><td colspan="3">000</td><td colspan="2">(product specification)</td><td>0</td><td>0</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>196</td><td>CHAPTER 5:Clock System 5.Registers 5.1.Common Configuration Registers 5.1.4.PLL/SSCG Stabilization Time Control Register (SYSC_PLLSSCGSTCNTR)</td><td>Error)
PLL/SSCG stabilization time control register(SYSC_STCNTR) is used to control clock stabilization time.

Correct)
PLL/SSCG stabilization time control register(SYSC_PLLSSCGSTCNTR) is used to control clock stabilization time.</td></tr>
<tr><td>198</td><td>CHAPTER 5:Clock System 5.Registers 5.1.Common Configuration Registers 5.1.5. PLL0 Clock Gear Control Register (SYSC_PLL0CGCNTR)</td><td>Error)
<table>
<tr><td>bit</td><td>bit 23</td><td>bit 16</td></tr>
<tr><td>Field</td><td colspan="2">PLLCGLP[7:0]</td></tr>
<tr><td>Data Attribute</td><td colspan="2">R/W</td></tr>
<tr><td>Prot_Attr</td><td colspan="2">WPS</td></tr>
<tr><td>Initial Value</td><td colspan="2">11111111</td></tr>
</table>
<table>
<tr><td>bit</td><td>bit 15</td><td>bit 14</td><td>bit 13</td><td>bit 12</td><td>bit 11</td><td>bit 10</td><td>bit 9</td><td>bit 8</td></tr>
<tr><td>Field</td><td colspan="2">PLLCGSTP[1:0]</td><td colspan="6">PLLCGSSN[5:0]</td></tr>
<tr><td>Data Attribute</td><td colspan="2">R/W</td><td colspan="6">R/W</td></tr>
<tr><td>Prot_Attr</td><td colspan="8">WPS</td></tr>
<tr><td>Initial Value</td><td colspan="2">00</td><td colspan="6">000000</td></tr>
</table>
<table>
<tr><td>bit</td><td>bit 7</td><td>bit 6</td><td>bit 5</td><td>bit 4</td><td>bit 3</td><td>bit 2</td><td>bit 1</td><td>bit 0</td></tr>
<tr><td>Field</td><td colspan="2">PLLCGSTS[1:0]</td><td colspan="4">Reserved</td><td>PLLCG STR</td><td>PLLCG EN</td></tr>
<tr><td>Data Attribute</td><td colspan="2">R,WX</td><td colspan="4">R0,WX</td><td>R,W</td><td>R/W</td></tr>
<tr><td>Prot_Attr</td><td colspan="8">WPS</td></tr>
<tr><td>Initial Value</td><td colspan="2">00</td><td colspan="4">0000</td><td>0</td><td>1</td></tr>
</table>
Correct)
<table>
<tr><td>bit</td><td>bit 23</td><td>bit 16</td></tr>
<tr><td>Field</td><td colspan="2">PLLCGLP[7:0]</td></tr>
<tr><td>Data Attribute</td><td colspan="2">R/W</td></tr>
<tr><td>Prot_Attr</td><td colspan="2">WPS</td></tr>
<tr><td>Initial Value</td><td colspan="2">(product specification)</td></tr>
</table>
<table>
<tr><td>bit</td><td>bit 15</td><td>bit 14</td><td>bit 13</td><td>bit 12</td><td>bit 11</td><td>bit 10</td><td>bit 9</td><td>bit 8</td></tr>
<tr><td>Field</td><td colspan="2">PLLCGSTP[1:0]</td><td colspan="6">PLLCGSSN[5:0]</td></tr>
<tr><td>Data Attribute</td><td colspan="2">R/W</td><td colspan="6">R/W</td></tr>
<tr><td>Prot_Attr</td><td colspan="8">WPS</td></tr>
<tr><td>Initial Value</td><td colspan="2">(product specification)</td><td colspan="6">(product specification)</td></tr>
</table>
<table>
<tr><td>bit</td><td>bit 7</td><td>bit 6</td><td>bit 5</td><td>bit 4</td><td>bit 3</td><td>bit 2</td><td>bit 1</td><td>bit 0</td></tr>
<tr><td>Field</td><td colspan="2">PLLCGSTS[1:0]</td><td colspan="4">Reserved</td><td>PLLCG STR</td><td>PLLCG EN</td></tr>
<tr><td>Data Attribute</td><td colspan="2">R,WX</td><td colspan="4">R0,WX</td><td>R,W</td><td>R/W</td></tr>
<tr><td>Prot_Attr</td><td colspan="8">WPS</td></tr>
<tr><td>Initial Value</td><td colspan="2">00</td><td colspan="4">0000</td><td>0</td><td>(product specification)</td></tr>
</table></td></tr>
</table>

Page	Section	Change Results
201	CHAPTER 5:Clock System 5.Registers 5.1.Common Configuration Registers 5.1.6. PLL1 Clock Gear Control Register (SYSC_PLL1 CGCNTR)	Error)    Correct)   

Page	Section	Change Results
204	CHAPTER 5:Clock System 5.Registers 5.1.Common Configuration Registers 5.1.7. PLL2 Clock Gear Control Register (SYSC_PLL2 CGCNTR)	Error)  Correct) 

Page	Section	Change Results
207	CHAPTER 5:Clock System 5.Registers 5.1.Common Configuration Registers 5.1.8. PLL3 Clock Gear Control Register (SYSC_PLL3 CGCNTR)	Error)  Correct) 

bit	bit 23							bit 16
Field	PLLCGLP[7:0]							
Data Attribute	R/W							
Prot_Attr	WPS							
Initial Value	11111111							

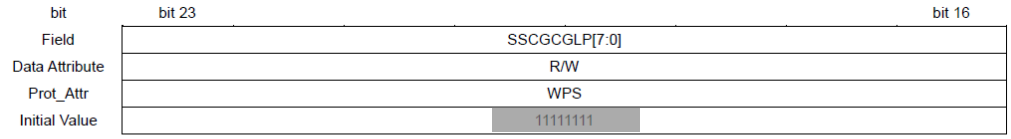
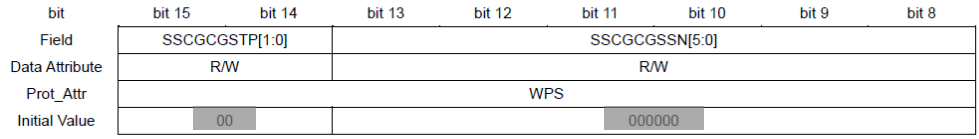
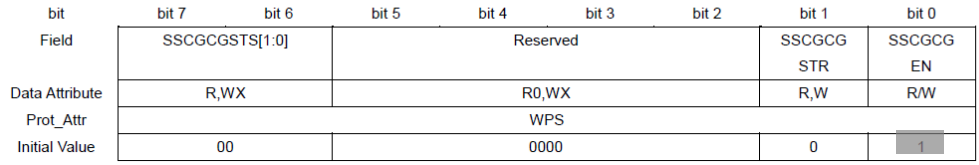
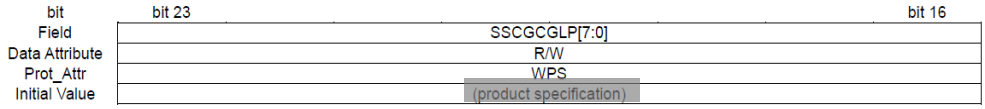
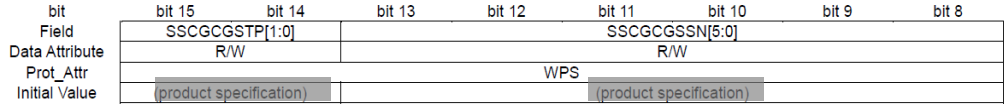
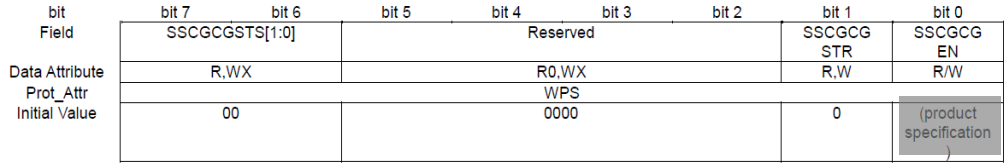
bit	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
Field	PLLCGSTP[1:0]		PLLCGSSN[5:0]					
Data Attribute	R/W		R/W					
Prot_Attr	WPS							
Initial Value	00		000000					

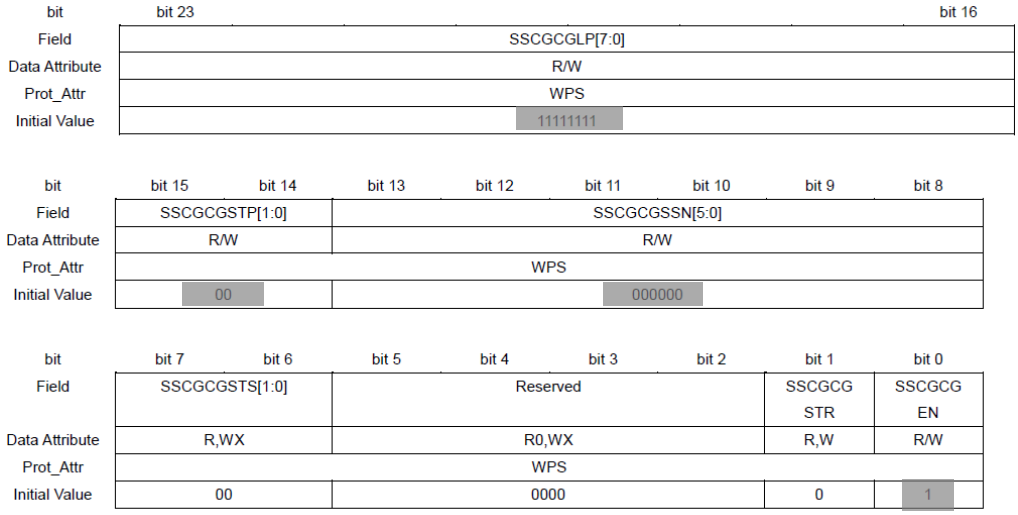
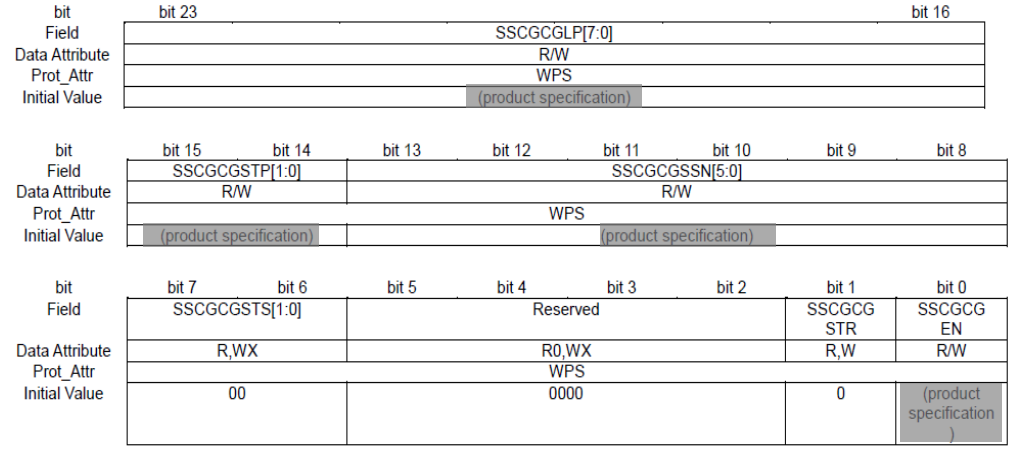
bit	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Field	PLLCGSTS[1:0]		Reserved				PLLCG STR	PLLCG EN
Data Attribute	R,WX		R0,WX				R,W	R/W
Prot_Attr	WPS							
Initial Value	00		0000				0	1

bit	bit 23							bit 16
Field	PLLCGLP[7:0]							
Data Attribute	R/W							
Prot_Attr	WPS							
Initial Value	(product specification)							

bit	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
Field	PLLCGSTP[1:0]		PLLCGSSN[5:0]					
Data Attribute	R/W		R/W					
Prot_Attr	WPS							
Initial Value	(product specification)		(product specification)					

bit	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Field	PLLCGSTS[1:0]		Reserved				PLLCG STR	PLLCG EN
Data Attribute	R,WX		R0,WX				R,W	R/W
Prot_Attr	WPS							
Initial Value	00		0000				0	(product specification)

Page	Section	Change Results
210	CHAPTER 5:Clock System 5.Registers 5.1.Common Configuration Registers 5.1.9. SSCG PLL0 Clock Gear Control Register (SYSC_SSCG0CGCNTR)	Error)  bit: bit 23 … bit 16 Field: SSCGCGLP[7:0] Data Attribute: R/W Prot_Attr: WPS Initial Value: 11111111  bit: bit 15 bit 14 bit 13 bit 12 bit 11 bit 10 bit 9 bit 8 Field: SSCGCGSTP[1:0] | SSCGCGSSN[5:0] Data Attribute: R/W | R/W Prot_Attr: WPS Initial Value: 00 | 000000  bit: bit 7 bit 6 bit 5 bit 4 bit 3 bit 2 bit 1 bit 0 Field: SSCGCGSTS[1:0] | Reserved | SSCGCGSTR | SSCGCGEN Data Attribute: R,WX | R0,WX | R,W | R/W Prot_Attr: WPS Initial Value: 00 | 0000 | 0 | 1 Correct)  bit: bit 23 … bit 16 Field: SSCGCGLP[7:0] Data Attribute: R/W Prot_Attr: WPS Initial Value: (product specification)  bit: bit 15 bit 14 bit 13 bit 12 bit 11 bit 10 bit 9 bit 8 Field: SSCGCGSTP[1:0] | SSCGCGSSN[5:0] Data Attribute: R/W | R/W Prot_Attr: WPS Initial Value: (product specification) | (product specification)  bit: bit 7 bit 6 bit 5 bit 4 bit 3 bit 2 bit 1 bit 0 Field: SSCGCGSTS[1:0] | Reserved | SSCGCGSTR | SSCGCGEN Data Attribute: R,WX | R0,WX | R,W | R/W Prot_Attr: WPS Initial Value: 00 | 0000 | 0 | (product specification)

Page	Section	Change Results
213	CHAPTER 5:Clock System 5.Registers 5.1.Common Configuration Registers 5.1.10. SSCG PLL1 Clock Gear Control Register (SYSC_SSCG1CGCNTR)	Error)  Correct) 

bit	bit 23							bit 16
Field	SSCGCGLP[7:0]							
Data Attribute	R/W							
Prot_Attr	WPS							
Initial Value	11111111							

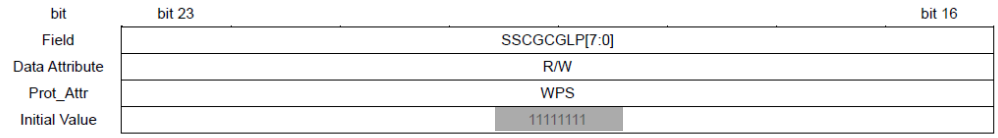
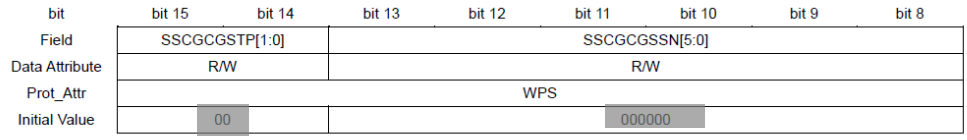
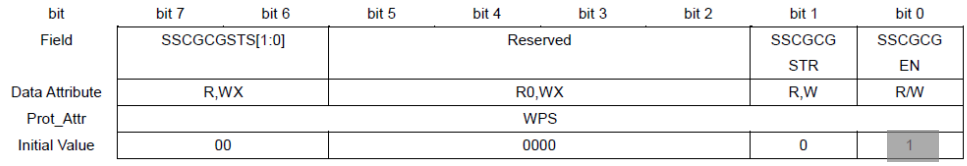
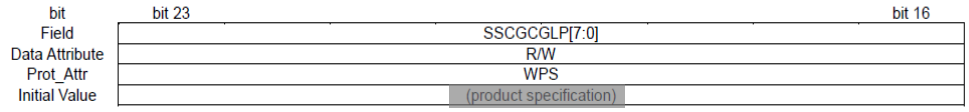
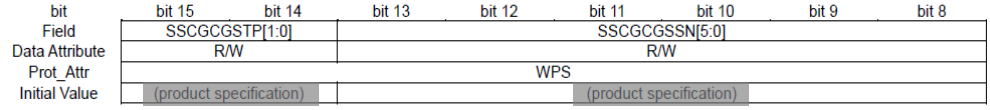
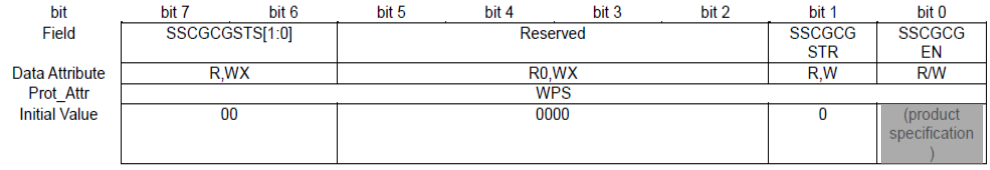
bit	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
Field	SSCGCGSTP[1:0]		SSCGCGSSN[5:0]					
Data Attribute	R/W		R/W					
Prot_Attr	WPS							
Initial Value	00		000000					

bit	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Field	SSCGCGSTS[1:0]		Reserved				SSCGCG STR	SSCGCG EN
Data Attribute	R,WX		R0,WX				R,W	R/W
Prot_Attr	WPS							
Initial Value	00		0000				0	1

bit	bit 23							bit 16
Field	SSCGCGLP[7:0]							
Data Attribute	R/W							
Prot_Attr	WPS							
Initial Value	(product specification)							

bit	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
Field	SSCGCGSTP[1:0]		SSCGCGSSN[5:0]					
Data Attribute	R/W		R/W					
Prot_Attr	WPS							
Initial Value	(product specification)		(product specification)					

bit	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Field	SSCGCGSTS[1:0]		Reserved				SSCGCG STR	SSCGCG EN
Data Attribute	R,WX		R0,WX				R,W	R/W
Prot_Attr	WPS							
Initial Value	00		0000				0	(product specification)

Page	Section	Change Results
216	CHAPTER 5:Clock System 5.Registers 5.1.Common Configuration Registers 5.1.11. SSCG PLL2 Clock Gear Control Register (SYSC_SSCG2CGCNTR)	Error)    Correct)   

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>219</td><td>CHAPTER 5:Clock System 5.Registers 5.1.Common Configuration Registers 5.1.12. SSCG PLL3 Clock Gear Control Register (SYSC_SSCG3CGCNTR)</td><td>Error)
<table>
<tr><td>bit</td><td>bit 23</td><td></td><td></td><td></td><td></td><td></td><td></td><td>bit 16</td></tr>
<tr><td>Field</td><td colspan="8">SSCGCGLP[7:0]</td></tr>
<tr><td>Data Attribute</td><td colspan="8">R/W</td></tr>
<tr><td>Prot_Attr</td><td colspan="8">WPS</td></tr>
<tr><td>Initial Value</td><td colspan="8">11111111</td></tr>
</table>
<table>
<tr><td>bit</td><td>bit 15</td><td>bit 14</td><td>bit 13</td><td>bit 12</td><td>bit 11</td><td>bit 10</td><td>bit 9</td><td>bit 8</td></tr>
<tr><td>Field</td><td colspan="2">SSCGCGSTP[1:0]</td><td colspan="6">SSCGCGSSN[5:0]</td></tr>
<tr><td>Data Attribute</td><td colspan="2">R/W</td><td colspan="6">R/W</td></tr>
<tr><td>Prot_Attr</td><td colspan="8">WPS</td></tr>
<tr><td>Initial Value</td><td colspan="2">00</td><td colspan="6">000000</td></tr>
</table>
<table>
<tr><td>bit</td><td>bit 7</td><td>bit 6</td><td>bit 5</td><td>bit 4</td><td>bit 3</td><td>bit 2</td><td>bit 1</td><td>bit 0</td></tr>
<tr><td>Field</td><td colspan="2">SSCGCGSTS[1:0]</td><td colspan="4">Reserved</td><td>SSCGCG STR</td><td>SSCGCG EN</td></tr>
<tr><td>Data Attribute</td><td colspan="2">R,WX</td><td colspan="4">R0,WX</td><td>R,W</td><td>R/W</td></tr>
<tr><td>Prot_Attr</td><td colspan="8">WPS</td></tr>
<tr><td>Initial Value</td><td colspan="2">00</td><td colspan="4">0000</td><td>0</td><td>1</td></tr>
</table>
Correct)
<table>
<tr><td>bit</td><td>bit 23</td><td></td><td></td><td></td><td></td><td></td><td></td><td>bit 16</td></tr>
<tr><td>Field</td><td colspan="8">SSCGCGLP[7:0]</td></tr>
<tr><td>Data Attribute</td><td colspan="8">R/W</td></tr>
<tr><td>Prot_Attr</td><td colspan="8">WPS</td></tr>
<tr><td>Initial Value</td><td colspan="8">(product specification)</td></tr>
</table>
<table>
<tr><td>bit</td><td>bit 15</td><td>bit 14</td><td>bit 13</td><td>bit 12</td><td>bit 11</td><td>bit 10</td><td>bit 9</td><td>bit 8</td></tr>
<tr><td>Field</td><td colspan="2">SSCGCGSTP[1:0]</td><td colspan="6">SSCGCGSSN[5:0]</td></tr>
<tr><td>Data Attribute</td><td colspan="2">R/W</td><td colspan="6">R/W</td></tr>
<tr><td>Prot_Attr</td><td colspan="8">WPS</td></tr>
<tr><td>Initial Value</td><td colspan="2">(product specification)</td><td colspan="6">(product specification)</td></tr>
</table>
<table>
<tr><td>bit</td><td>bit 7</td><td>bit 6</td><td>bit 5</td><td>bit 4</td><td>bit 3</td><td>bit 2</td><td>bit 1</td><td>bit 0</td></tr>
<tr><td>Field</td><td colspan="2">SSCGCGSTS[1:0]</td><td colspan="4">Reserved</td><td>SSCGCG STR</td><td>SSCGCG EN</td></tr>
<tr><td>Data Attribute</td><td colspan="2">R,WX</td><td colspan="4">R0,WX</td><td>R,W</td><td>R/W</td></tr>
<tr><td>Prot_Attr</td><td colspan="8">WPS</td></tr>
<tr><td>Initial Value</td><td colspan="2">00</td><td colspan="4">0000</td><td>0</td><td>(product specification)</td></tr>
</table></td></tr>
<tr><td>224</td><td>CHAPTER 5:Clock System 6. Usage Precautions</td><td>Added the Below:
■ The selection of Domain clock
Do not use Sub clock for the domain clock.</td></tr>
</table>

Page	Section	Change Results
233	CHAPTER 6:Low Power Consumption 3.2. Power Supply Domain	Deleted the Below: Supply of Power 3. Internal regulator supply area PD1, PD4_0, PD4_1, PD6_0, PD6_1, PD2, PD5_0, PD5_1, PD5_2, PD5_3
233	CHAPTER 6:Low Power Consumption 3.2. Power Supply Domain	Error) Startup and shutdown sequences The startup sequence is as follows. 1 Turn ON PSW of PD6_0 2 Turn ON PSW of PD2 or external power control 3 Turn ON PSW of PD4 4 Turn ON PSW of PD6_1 5 Turn ON PSW of PD5_0 6 Turn ON PSW of PD5_1 7 Turn ON PSW of PD5_2 8 Turn ON PSW of PD5_3 To shut down the power supply, turn OFF all PSWs simultaneously. To start up the power supply, turn ON all PSWs or external power simultaneously. To shut down the power supply, turn OFF all PSWs simultaneously. Correct) Startup and shutdown sequences To start up the power supply, turn ON all PSWs or external power simultaneously. To shut down the power supply, turn OFF all PSWs simultaneously.

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>237</td><td>CHAPTER 6:Low Power Consumption 3.3. Profile Setting Items</td><td>
Error)
Table 3-2 List of RUN/PSS Profile Setting Parameters
<table>
<tr><th>Category</th><th>Setting Item</th><th>RUN Setting (Initial Value)</th><th>RUN Setting</th><th>PSS Setting</th><th>Control Block</th></tr>
<tr><td rowspan="20">Clock</td><td>LLPBM clock oscillation enable</td><td>Enable</td><td>Enable</td><td>Enable control</td><td>SYSC1</td></tr>
<tr><td>LLPBM2 clock oscillation enable</td><td>Enable</td><td>Enable</td><td>Enable control</td><td>SYSC1</td></tr>
<tr><td>LCP clock oscillation enable</td><td>Enable</td><td>Enable</td><td>Enable control</td><td>SYSC1</td></tr>
<tr><td>LCP0 clock oscillation enable</td><td>Enable</td><td>Enable</td><td>Enable control</td><td>SYSC1</td></tr>
<tr><td>LCP1 clock oscillation enable</td><td>Enable</td><td>Enable</td><td>Enable control</td><td>SYSC1</td></tr>
<tr><td>LCPP0 clock oscillation enabled</td><td>Enable</td><td>Enable</td><td>Enable control</td><td>SYSC1</td></tr>
<tr><td>LCPP1 clock oscillation enable</td><td>Enable</td><td>Enable</td><td>Enable control</td><td>SYSC1</td></tr>
<tr><td>LCP0A clock oscillation enable</td><td>Enable</td><td>Enable</td><td>Enable control</td><td>SYSC1</td></tr>
<tr><td>LCP1A clock oscillation enable</td><td>Enable</td><td>Enable</td><td>Enable control</td><td>SYSC1</td></tr>
<tr><td>LCPP0A clock oscillation enable</td><td>Enable</td><td>Enable</td><td>Enable control</td><td>SYSC1</td></tr>
<tr><td>LCPP1A clock oscillation enable</td><td>Enable</td><td>Enable</td><td>Enable control</td><td>SYSC1</td></tr>
<tr><td>CD1 clock oscillation enable</td><td>Enable</td><td>Enable control</td><td>Enable control</td><td>SYSC1</td></tr>
<tr><td>CD1A0 clock oscillation enable</td><td>Enable</td><td>Enable control</td><td>Enable control</td><td>SYSC1</td></tr>
<tr><td>CD1A1 clock oscillation enable</td><td>Enable</td><td>Enable control</td><td>Enable control</td><td>SYSC1</td></tr>
<tr><td>CD1B0 clock oscillation enable</td><td>Enable</td><td>Enable control</td><td>Enable control</td><td>SYSC1</td></tr>
<tr><td>CD1B1 clock oscillation enable</td><td>Enable</td><td>Enable control</td><td>Enable control</td><td>SYSC1</td></tr>
</table>
Correct)
Table 3-2 List of RUN/PSS Profile Setting Parameters
<table>
<tr><th>Category</th><th>Setting Item</th><th>RUN Setting (Initial Value)</th><th>RUN Setting</th><th>PSS Setting</th><th>Control Block</th></tr>
<tr><td rowspan="20">Clock</td><td>LLPBM clock oscillation enable</td><td>Enable</td><td>Enable</td><td>Enable control</td><td>SYSC1</td></tr>
<tr><td>LLPBM2 clock oscillation enable</td><td>Enable</td><td>Enable</td><td>Enable control</td><td>SYSC1</td></tr>
<tr><td>LCP clock oscillation enable</td><td>Enable</td><td>Enable</td><td>Enable control</td><td>SYSC1</td></tr>
<tr><td>LCP0 clock oscillation enable</td><td>Enable</td><td>Enable</td><td>Enable control</td><td>SYSC1</td></tr>
<tr><td>LCP1 clock oscillation enable</td><td>Enable</td><td>Enable</td><td>Enable control</td><td>SYSC1</td></tr>
<tr><td>LAPP0 clock oscillation enabled</td><td>Enable</td><td>Enable</td><td>Enable control</td><td>SYSC1</td></tr>
<tr><td>LAPP1 clock oscillation enable</td><td>Enable</td><td>Enable</td><td>Enable control</td><td>SYSC1</td></tr>
<tr><td>LCP0A clock oscillation enable</td><td>Enable</td><td>Enable</td><td>Enable control</td><td>SYSC1</td></tr>
<tr><td>LCP1A clock oscillation enable</td><td>Enable</td><td>Enable</td><td>Enable control</td><td>SYSC1</td></tr>
<tr><td>LAPP0A clock oscillation enable</td><td>Enable</td><td>Enable</td><td>Enable control</td><td>SYSC1</td></tr>
<tr><td>LAPP1A clock oscillation enable</td><td>Enable</td><td>Enable</td><td>Enable control</td><td>SYSC1</td></tr>
<tr><td>CD1 clock oscillation enable</td><td>Enable</td><td>Enable control</td><td>Enable control</td><td>SYSC1</td></tr>
<tr><td>CD1A0 clock oscillation enable</td><td>Enable</td><td>Enable control</td><td>Enable control</td><td>SYSC1</td></tr>
<tr><td>CD1A1 clock oscillation enable</td><td>Enable</td><td>Enable control</td><td>Enable control</td><td>SYSC1</td></tr>
<tr><td>CD1B0 clock oscillation enable</td><td>Enable</td><td>Enable control</td><td>Enable control</td><td>SYSC1</td></tr>
<tr><td>CD1B1 clock oscillation enable</td><td>Enable</td><td>Enable control</td><td>Enable control</td><td>SYSC1</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>242</td><td>CHAPTER 6:Low Power Consumption 3.4. Profile</td><td>Error)
PSS<table><tr><td>SYSC0_PSSSSCGC0NTR0</td><td>SYSC0_PSSPLL0CNTR</td><td colspan="4">SYSC0_PSSCKSRER</td></tr><tr><td>SSCG0ISEL</td><td>PLL0ISEL</td><td>SSCG0EN</td><td>PLL0EN</td><td>MOSCEN</td><td>CROSCEN</td></tr><tr><td>X</td><td>1</td><td>X</td><td>1</td><td>X</td><td>0</td></tr><tr><td>1</td><td>X</td><td>1</td><td>X</td><td>X</td><td>0</td></tr><tr><td>X</td><td>0</td><td>X</td><td>1</td><td>0</td><td>X</td></tr><tr><td>0</td><td>X</td><td>1</td><td>X</td><td>0</td><td>X</td></tr></table>
Correct)
PSS<table><tr><td>SYSC0_PSSSSCG0CNTR0</td><td>SYSC0_PSSPLL0CNTR</td><td colspan="4">SYSC0_PSSCKSRER</td></tr><tr><td>SSCG0ISEL</td><td>PLL0ISEL</td><td>SSCG0EN</td><td>PLL0EN</td><td>MOSCEN</td><td>CROSCEN</td></tr><tr><td>X</td><td>1</td><td>X</td><td>1</td><td>X</td><td>0</td></tr><tr><td>1</td><td>X</td><td>1</td><td>X</td><td>X</td><td>0</td></tr><tr><td>X</td><td>0</td><td>X</td><td>1</td><td>0</td><td>X</td></tr><tr><td>0</td><td>X</td><td>1</td><td>X</td><td>0</td><td>X</td></tr></table></td></tr>
<tr><td>244</td><td>CHAPTER 6:Low Power Consumption 3.4. Profile</td><td>Added the Below:

- In the product which supports SSCG PLL0 clock, it doesn't include " 101 " of the set value in the profile error.
- In the product which supports Sub clock, it doesn't include " 011 " of the set value in the profile error.</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>250</td><td>CHAPTER 6:Low Power Consumption 3.4. Profile</td><td>
Error)
RUN
<table>
<tr><td colspan="4">SYSC0_RUNCKSRER</td><td colspan="5">SWDG0_CFG</td></tr>
<tr><td>SOSCEN</td><td>MOSC EN</td><td>SCROSC EN</td><td>CROSC EN</td><td>WDENRUN</td><td>WDENPSS</td><td>ALLOW STOPCLK</td><td colspan="2">CLKSEL[1:0]</td></tr>
<tr><td>X</td><td>0</td><td>X</td><td>X</td><td>1</td><td>X</td><td>X</td><td>1</td><td>1</td></tr>
<tr><td>0</td><td>X</td><td>X</td><td>X</td><td>1</td><td>X</td><td>X</td><td>1</td><td>0</td></tr>
</table>
PSS
<table>
<tr><td colspan="4">SYSC0_PSSCKSRER</td><td colspan="5">SWDG0_CFG</td></tr>
<tr><td>SOSCEN</td><td>MOSC EN</td><td>SCROSC EN</td><td>CROSC EN</td><td>WDENRUN</td><td>WDENPSS</td><td>ALLOW STOPCLK</td><td colspan="2">CLKSEL[1:0]</td></tr>
<tr><td>X</td><td>0</td><td>X</td><td>X</td><td>X</td><td>1</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>0</td><td>X</td><td>X</td><td>X</td><td>X</td><td>1</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>X</td><td>X</td><td>0</td><td>X</td><td>X</td><td>1</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>X</td><td>X</td><td>X</td><td>0</td><td>X</td><td>1</td><td>0</td><td>0</td><td>0</td></tr>
</table>
Correct)
RUN
<table>
<tr><td colspan="4">SYSC0_RUNCKSRER</td><td colspan="5">SWDG_CFG</td></tr>
<tr><td>SOSCEN</td><td>MOSC EN</td><td>SCROSC EN</td><td>CROSC EN</td><td>WDENRUN</td><td>WDENPSS</td><td>ALLOW STOPCLK</td><td colspan="2">CLKSEL[1:0]</td></tr>
<tr><td>X</td><td>0</td><td>X</td><td>X</td><td>1</td><td>X</td><td>X</td><td>1</td><td>1</td></tr>
<tr><td>0</td><td>X</td><td>X</td><td>X</td><td>1</td><td>X</td><td>X</td><td>1</td><td>0</td></tr>
</table>
PSS
<table>
<tr><td colspan="4">SYSC0_PSSCKSRER</td><td colspan="5">SWDG_CFG</td></tr>
<tr><td>SOSCEN</td><td>MOSC EN</td><td>SCROSC EN</td><td>CROSC EN</td><td>WDENRUN</td><td>WDENPSS</td><td>ALLOW STOPCLK</td><td colspan="2">CLKSEL[1:0]</td></tr>
<tr><td>X</td><td>0</td><td>X</td><td>X</td><td>X</td><td>1</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>0</td><td>X</td><td>X</td><td>X</td><td>X</td><td>1</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>X</td><td>X</td><td>0</td><td>X</td><td>X</td><td>1</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>X</td><td>X</td><td>X</td><td>0</td><td>X</td><td>1</td><td>0</td><td>0</td><td>0</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>266</td><td>CHAPTER 6:Low Power Consumption 5. Registers</td><td>Added the Below:

Table 5-6 System Register Group
<table><tr><th>Abbreviated Register Name</th><th>Register Name</th><th>Reference</th></tr><tr><td>SYSC0_SYSIDR</td><td>System ID register</td><td>5.6.1</td></tr><tr><td>SYSC0_SYSPFIDR</td><td>Platform ID register</td><td>5.6.2</td></tr><tr><td>SYSC0_SYSSTSR</td><td>System status register</td><td>5.6.3</td></tr><tr><td>SYSC0_SYSINTER</td><td>System status interrupt enable register</td><td>5.6.4</td></tr><tr><td>SYSC0_SYSICLR</td><td>System status flag interrupt clear register</td><td>5.6.5</td></tr><tr><td>SYSC0_SYSERRIR0</td><td>System error interrupt factor register 0</td><td>5.6.6</td></tr><tr><td>SYSC0_SYSERRIR1</td><td>System error interrupt factor register 1</td><td>5.6.7</td></tr><tr><td>SYSC0_SYSERRICLR0</td><td>System error interrupt factor clear register 0</td><td>5.6.8</td></tr><tr><td>SYSC0_SYSERRICLR1</td><td>System error interrupt factor clear register 1</td><td>5.6.9</td></tr><tr><td>SYSC0_SYSPROSTSR</td><td>Profile status register</td><td>5.6.10</td></tr><tr><td>SYSC0_SYSRUNPEFR</td><td>RUN profile error flag register</td><td>5.6.11</td></tr><tr><td>SYSC0_SYSPSSPEFR</td><td>PSS profile error flag register</td><td>5.6.12</td></tr></table></td></tr>
</table>

Page	Section	Change Results
274	CHAPTER 6:Low Power Consumption 5. Registers	Error)

Table 5-15 Low Power Consumption Register Address Map (SYSC0)

Offset	+3	+2	+1	+0	Area
0x0000_0248	SYSC0_STSCSVCFGR 00000000_00000000_00000000_00000000				STS profile register area
0x0000_024C	SYSC0_STSREGCFGR 00000000_00000000_00000000_0000000*				
0x0000_0250 - 0x0000_027C	Reserved				
0x0000_0280	Reserved				System register area
0x0000_0284	Reserved				
0x0000_0288	SYSC0_SYSSTSR 00000000_00000000_00000000_00000001				
0x0000_028C	SYSC0_SYSINTER 00000000_00000000_00000000_00000000				
0x0000_0290	SYSC0_SYSICLR 00000000_00000000_00000000_00000000				
0x0000_0294	SYSC0_SYSERRIR0 00000000_00000000_00000000_00000000				
0x0000_0298	SYSC0_SYSERRIR1 00000000_00000000_00000000_00000000				
0x0000_029C	SYSC0_SYSERRICLR0 00000000_00000000_00000000_00000000				
0x0000_02A0	SYSC0_SYSERRICLR1 00000000_00000000_00000000_00000000				
0x0000_02A4	SYSC0_SYSPROSTSR 00000000_00000000_00000000_00000000				
0x0000_02A8	SYSC0_SYSRUNPEFR 00000000_00000000_00000000_00000000				
0x0000_02AC	SYSC0_SYSPSSPEFR 00000000_00000000_00000000_00000000				
0x0000_02B0 - 0x0000_02FC	Reserved				
0x0000_0680	SYSC0_SPECFGR 00000000_01000000_00000000_00001110				Special register area
0x0000_0684	Reserved				
0x0000_0688					
0x0000_068C	Reserved				
0x0000_0690	Reserved				
0x0000_0694	Reserved				

Correct)

Table 5-15 Low Power Consumption Register Address Map (SYSC0)

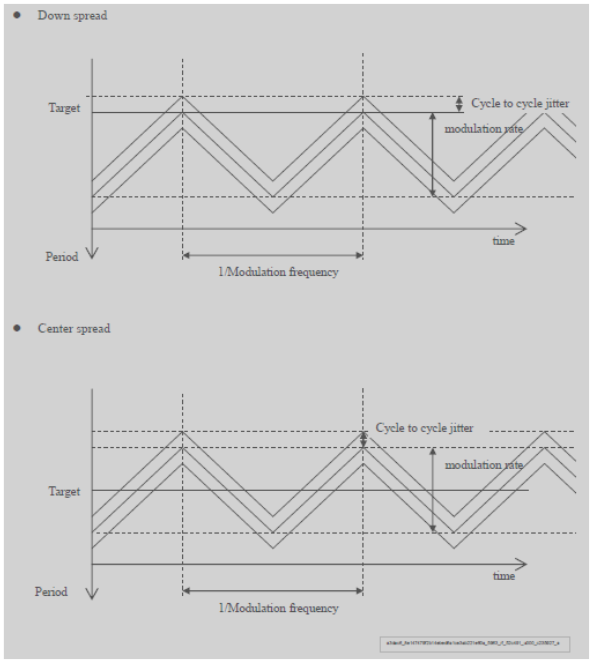
Offset	+3	+2	+1	+0	Area
0x0000_0248	SYSC0_STSCSVCFGR 00000000_00000000_00000000_00000000				STS profile register area
0x0000_024C	SYSC0_STSREGCFGR 00000000_00000000_00000000_0000000*				
0x0000_0250 - 0x0000_027C	Reserved				
0x0000_0280	SYSC0_SYSIDR				System register area
0x0000_0284	SYSC0_SYSPFIDR				
0x0000_0288	SYSC0_SYSSTSR 00000000_00000000_00000000_00000001				
0x0000_028C	SYSC0_SYSINTER 00000000_00000000_00000000_00000000				
0x0000_0290	SYSC0_SYSICLR 00000000_00000000_00000000_00000000				
0x0000_0294	SYSC0_SYSERRIR0 00000000_00000000_00000000_00000000				
0x0000_0298	SYSC0_SYSERRIR1 00000000_00000000_00000000_00000000				
0x0000_029C	SYSC0_SYSERRICLR0 00000000_00000000_00000000_00000000				
0x0000_02A0	SYSC0_SYSERRICLR1 00000000_00000000_00000000_00000000				
0x0000_02A4	SYSC0_SYSPROSTSR 00000000_00000000_00000000_00000000				
0x0000_02A8	SYSC0_SYSRUNPEFR 00000000_00000000_00000000_00000000				
0x0000_02AC	SYSC0_SYSPSSPEFR 00000000_00000000_00000000_00000000				
0x0000_02B0 - 0x0000_02FC	Reserved				
0x0000_0680	SYSC0_SPECFGR 00000000_01000000_00000000_00001110				Special register area
0x0000_0684	Reserved				
0x0000_0688	Reserved				
0x0000_068C	Reserved				
0x0000_0690	Reserved				
0x0000_0694	Reserved				

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>286</td><td>CHAPTER 6:Low Power Consumption 5. Registers 5.2.2. RUN Clock Source Enable Register (SYSC0_RUNCKSRER)</td><td>Error)
<table>
<tr><td>bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Field</td><td colspan="4">Reserved</td><td>SOSCEN</td><td>MOSC EN</td><td>SCROSC EN</td><td>CROSC EN</td></tr>
<tr><td>R/W Attribute</td><td colspan="4">R0,WX</td><td>R/W</td><td>R/W</td><td>R1,WX</td><td>R1,WX</td></tr>
<tr><td>Protection Attribute</td><td colspan="8">WPS</td></tr>
<tr><td>Initial Value</td><td colspan="4">0000</td><td>(product specification)</td><td>1</td><td>1</td><td>1</td></tr>
</table>
Correct)
<table>
<tr><td>bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Field</td><td colspan="4">Reserved</td><td>SOSCEN</td><td>MOSC EN</td><td>SCROSC EN</td><td>CROSC EN</td></tr>
<tr><td>R/W Attribute</td><td colspan="4">R0,WX</td><td>R/W</td><td>R/W</td><td>R1,WX</td><td>R1,WX</td></tr>
<tr><td>Protection Attribute</td><td colspan="8">WPS</td></tr>
<tr><td>Initial Value</td><td colspan="4">0000</td><td>0</td><td>1</td><td>1</td><td>1</td></tr>
</table></td></tr>
<tr><td>289</td><td>CHAPTER 6:Low Power Consumption 5. Registers 5.2.3. RUN Clock Selection Register (SYSC0_RUNCKSELR)</td><td>Added the Below:

[bit2:0] CDMCUCCSL: Clock Domain MCUC Clock Selection Bits
<table>
<tr><th>bit2:0</th><th>Description</th></tr>
<tr><td>011</td><td>Sub clock (Do not use this setting)</td></tr>
</table></td></tr>
<tr><td>300</td><td>CHAPTER 6:Low Power Consumption 5. Registers 5.2.8. RUN SSCGx Control Register 1 (SYSC0_RUNSSCGxCNTR1)</td><td>Error)
[bit18:17] SSCGxFREQ: SSCG PLLx Clock modulation Frequency Selection Bits
These bits set the modulation frequency of the SSCG PLLx clock.
<table>
<tr><th>bit18:17</th><th>Description</th></tr>
<tr><td>00</td><td>Fmod = (1/1024) x Fin</td></tr>
<tr><td>01</td><td>Fmod = (1/2048) x Fin</td></tr>
<tr><td>10 or 11</td><td>Fmod = (1/4096) x Fin</td></tr>
</table>
* Fin: clock before modulation, Fmod: clock after modulation

Correct)
[bit18:17] SSCGxFREQ: SSCG PLLx Clock modulation Frequency Selection Bits
These bits set the modulation frequency of the SSCG PLLx clock.
<table>
<tr><th>bit18:17</th><th>Description</th></tr>
<tr><td>00</td><td>Fmod = (1/1024) x Fin</td></tr>
<tr><td>01</td><td>Fmod = (1/2048) x Fin</td></tr>
<tr><td>10 or 11</td><td>Fmod = (1/4096) x Fin</td></tr>
</table>
* Fin: SSCG reference clock frequency, Fmod: Modulation frequency</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>300</td><td>CHAPTER 6:Low Power Consumption 5. Registers 5.2.8. RUN SSCGx Control Register 1 (SYSC0_RUNSSCGxCNTR1)</td><td>Added the Below:

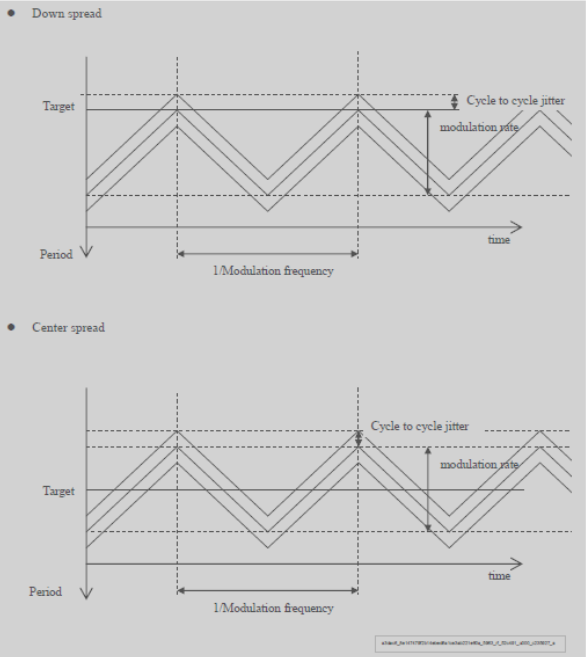
Correct)
[bit16] SSCGxMODE: SSCG PLLx modulation Mode Setting Bit
This bit sets the modulation mode of the SSCG PLLx clock.
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Set Down spread mode.</td></tr>
<tr><td>1</td><td>Set Center spread mode.</td></tr>
</table>

</td></tr>
<tr><td>316</td><td>CHAPTER 6:Low Power Consumption 5. Registers 5.3.2. PSS Clock Source Enable Register (SYSC0_PSSCKSRER)</td><td>Error)
<table>
<tr><td>bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Field</td><td colspan="4">Reserved</td><td>SOSCEN</td><td>MOSC EN</td><td>SCROSC EN</td><td>CROSC EN</td></tr>
<tr><td>R/W Attribute</td><td colspan="4">R0,WX</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td></tr>
<tr><td>Protection Attribute</td><td colspan="8">WPS</td></tr>
<tr><td>Initial Value</td><td colspan="4">0000</td><td>(product specification)</td><td>1</td><td>1</td><td>1</td></tr>
</table>
Correct)
<table>
<tr><td>bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Field</td><td colspan="4">Reserved</td><td>SOSCEN</td><td>MOSC EN</td><td>SCROSC EN</td><td>CROSC EN</td></tr>
<tr><td>R/W Attribute</td><td colspan="4">R0,WX</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td></tr>
<tr><td>Protection Attribute</td><td colspan="8">WPS</td></tr>
<tr><td>Initial Value</td><td colspan="4">0000</td><td>0</td><td>1</td><td>1</td><td>1</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>319</td><td>CHAPTER 6:Low Power Consumption 5. Registers 5.3.3. PSS Clock Selection Register (SYSC0_PSSCKSELR)</td><td>Added the Below:

[bit2:0] CDMCUCCSL: Clock Domain MCUC Clock Selection Bits
<table><tr><th>bit2:0</th><th>Description</th></tr><tr><td>011</td><td>Sub clock (Do not use this setting)</td></tr></table></td></tr>
<tr><td>328</td><td>CHAPTER 6:Low Power Consumption 5. Registers 5.3.8. PSS SSCGx Control Register 1 (SYSC0_PSSSSCGxCNTR1)</td><td>Error)
[bit18:17] SSCGxFREQ: SSCG PLLx Clock modulation Frequency Selection Bits
These bits set the modulation frequency of the SSCG PLL clock.
<table><tr><th>bit18:17</th><th>Description</th></tr><tr><td>00</td><td>Fmod = (1/1024) x Fin</td></tr><tr><td>01</td><td>Fmod = (1/2048) x Fin</td></tr><tr><td>10 or 11</td><td>Fmod = (1/4096) x Fin</td></tr></table>* Fin: clock before modulation, Fmod: clock after modulation

Correct)
[bit18:17] SSCGxFREQ: SSCG PLLx Clock modulation Frequency Selection Bits
These bits set the modulation frequency of the SSCG PLL clock.
<table><tr><th>bit18:17</th><th>Description</th></tr><tr><td>00</td><td>Fmod = (1/1024) x Fin</td></tr><tr><td>01</td><td>Fmod = (1/2048) x Fin</td></tr><tr><td>10 or 11</td><td>Fmod = (1/4096) x Fin</td></tr></table>* Fin: SSCG reference clock frequency, Fmod: Modulation frequency</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>328</td><td>CHAPTER 6:Low Power Consumption 5. Registers 5.3.8. PSS SSCGx Control Register 1 (SYSC0_PSS SSCGxCNTR 1)</td><td>Added the Below:

Correct)
[bit16] SSCGxMODE: SSCG PLLx modulation Mode Setting Bit
This bit sets the modulation mode of the SSCG PLL clock.
<table><tr><th>Bit</th><th>Description</th></tr><tr><td>0</td><td>Set Down spread mode.</td></tr><tr><td>1</td><td>Set Center spread mode.</td></tr></table>
</td></tr>
<tr><td>344</td><td>CHAPTER 6:Low Power Consumption 5. Registers 5.4.2. APP Clock Source Enable Register (SYSC0_APP CKSRER)</td><td>Error)
The SYSC0_APPCKSRER0 register indicates the value for setting whether to enable/disable oscillation of the internal operating clock to be updated.

Correct)
The SYSC0_APPCKSRER register indicates the value for setting whether to enable/disable oscillation of the internal operating clock to be updated.</td></tr>
</table>

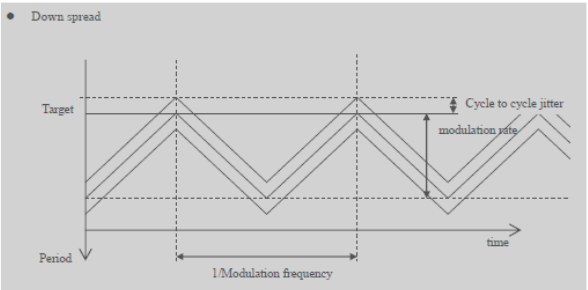
<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>344</td><td>CHAPTER 6:Low Power Consumption 5. Registers 5.4.2. APP Clock Source Enable Register (SYSC0_APPCKSRER)</td><td>Error)
<table>
<tr><td>bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Field</td><td colspan="4">Reserved</td><td>SOSCEN</td><td>MOSC EN</td><td>SCROSC EN</td><td>CROSC EN</td></tr>
<tr><td>R/W Attribute</td><td colspan="4">R0,WX</td><td>R,WX</td><td>R,WX</td><td>R,WX</td><td>R,WX</td></tr>
<tr><td>Protection Attribute</td><td colspan="8">WPS</td></tr>
<tr><td>Initial Value</td><td colspan="4">0000</td><td>(product specification)</td><td>1</td><td>1</td><td>1</td></tr>
</table>
Correct)
<table>
<tr><td>bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Field</td><td colspan="4">Reserved</td><td>SOSCEN</td><td>MOSC EN</td><td>SCROSC EN</td><td>CROSC EN</td></tr>
<tr><td>R/W Attribute</td><td colspan="4">R0,WX</td><td>R,WX</td><td>R,WX</td><td>R,WX</td><td>R,WX</td></tr>
<tr><td>Protection Attribute</td><td colspan="8">WPS</td></tr>
<tr><td>Initial Value</td><td colspan="4">0000</td><td>0</td><td>1</td><td>1</td><td>1</td></tr>
</table></td></tr>
<tr><td>347</td><td>CHAPTER 6:Low Power Consumption 5. Registers 5.4.3.APP Clock Selection Register (SYSC0_APPCKSELR)</td><td>Error)
The SYSC0_APPCKSELR0 register indicates the set values of the clock source for the clock domain to be updated.

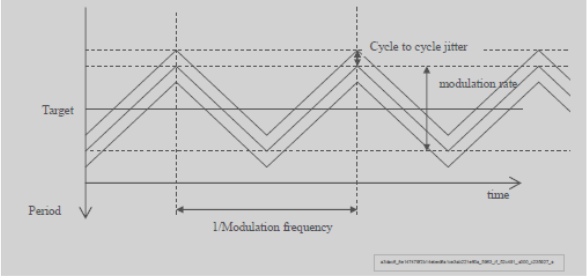
Correct)
The SYSC0_APPCKSELR register indicates the set values of the clock source for the clock domain to be updated.</td></tr>
<tr><td>347</td><td>CHAPTER 6:Low Power Consumption 5. Registers 5.4.3.APP Clock Selection Register (SYSC0_APPCKSELR)</td><td>Added the Below:

[bit2:0] CDMCUCCSL: Clock Domain MCUC Clock Selection Bits
<table>
<tr><th>bit2:0</th><th>Description</th></tr>
<tr><td>011</td><td>Sub clock (Do not use this setting)</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>358</td><td>CHAPTER 6:Low Power Consumption 5. Registers 5.4.8. APP SSCGx Control Register 1 (SYSC0_APP SSCGxCNTR 1)</td><td>Error)
[bit18:17] SSCGxFREQ: SSCG PLLx Clock modulation Frequency Selection Bits
These bits set the modulation frequency of the SSCG PLLx clock.
<table><tr><th>bit18:17</th><th>Description</th></tr><tr><td>00</td><td>Fmod = (1/1024) x Fin</td></tr><tr><td>01</td><td>Fmod = (1/2048) x Fin</td></tr><tr><td>10 or 11</td><td>Fmod = (1/4096) x Fin</td></tr></table>* Fin: clock before modulation, Fmod: clock after modulation

Correct)
[bit18:17] SSCGxFREQ: SSCG PLLx Clock modulation Frequency Selection Bits
These bits set the modulation frequency of the SSCG PLLx clock.
<table><tr><th>bit18:17</th><th>Description</th></tr><tr><td>00</td><td>Fmod = (1/1024) x Fin</td></tr><tr><td>01</td><td>Fmod = (1/2048) x Fin</td></tr><tr><td>10 or 11</td><td>Fmod = (1/4096) x Fin</td></tr></table>* Fin: SSCG reference clock frequency, Fmod: Modulation frequency</td></tr>
<tr><td>358</td><td>CHAPTER 6:Low Power Consumption 5. Registers 5.4.8. APP SSCGx Control Register 1 (SYSC0_APP SSCGxCNTR 1)</td><td>Added the Below:

Correct)
[bit16] SSCGxMODE: SSCG PLLx modulation Mode Setting Bit
This bit sets the modulation mode of the SSCG PLLx clock.
<table><tr><th>Bit</th><th>Description</th></tr><tr><td>0</td><td>Set Down spread mode.</td></tr><tr><td>1</td><td>Set Center spread mode.</td></tr></table>• Down spread


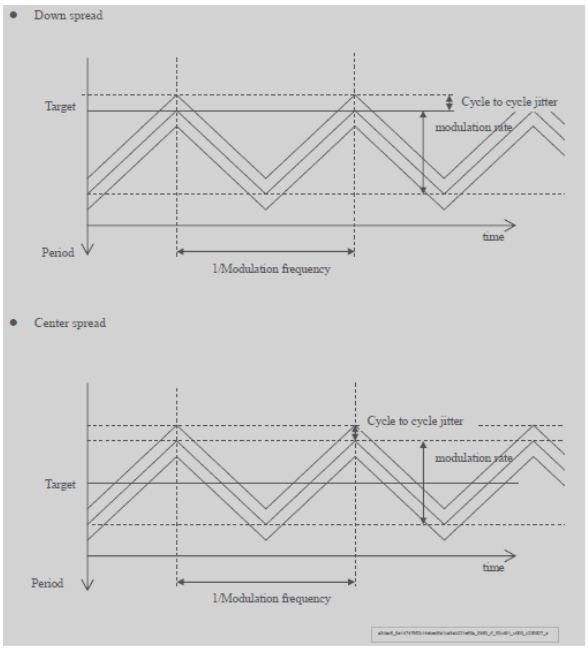
• Center spread

</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>373</td><td>CHAPTER 6:Low Power Consumption 5. Registers 5.5.2. STS Clock Source Enable Register (SYSC0_STSCKSRER)</td><td>Error)
<table>
<tr><td>bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Field</td><td>SOSC RDY</td><td>MOSC RDY</td><td>SCROSC RDY</td><td>CROSC RDY</td><td>SOSCEN</td><td>MOSC EN</td><td>SCROSC EN</td><td>CROSC EN</td></tr>
<tr><td>R/W Attribute</td><td>R,WX</td><td>R,WX</td><td>R,WX</td><td>R,WX</td><td>R,WX</td><td>R,WX</td><td>R,WX</td><td>R,WX</td></tr>
<tr><td>Protection Attribute</td><td colspan="8">WPS</td></tr>
<tr><td>Initial Value</td><td>0</td><td>0</td><td>0</td><td>0</td><td>(product specification)</td><td>1</td><td>1</td><td>1</td></tr>
</table>
Correct)
<table>
<tr><td>bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Field</td><td>SOSC RDY</td><td>MOSC RDY</td><td>SCROSC RDY</td><td>CROSC RDY</td><td>SOSCEN</td><td>MOSC EN</td><td>SCROSC EN</td><td>CROSC EN</td></tr>
<tr><td>R/W Attribute</td><td>R,WX</td><td>R,WX</td><td>R,WX</td><td>R,WX</td><td>R,WX</td><td>R,WX</td><td>R,WX</td><td>R,WX</td></tr>
<tr><td>Protection Attribute</td><td colspan="8">WPS</td></tr>
<tr><td>Initial Value</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td></tr>
</table></td></tr>
<tr><td>378</td><td>CHAPTER 6:Low Power Consumption 5. Registers 5.5.3. STS Clock Selection Register (SYSC0_STSCKSELR)</td><td>Added the Below:

[bit6:4] CDMCUCCM: Clock Domain MCUC Clock Selection Status Bits
<table>
<tr><th>Bit6:4</th><th>Description</th></tr>
<tr><td>011</td><td>Sub clock (Do not use this setting)</td></tr>
</table>
[bit2:0] CDMCUCCSL: Clock Domain MCUC Clock Selection Bits
<table>
<tr><th>Bit2:0</th><th>Description</th></tr>
<tr><td>011</td><td>Sub clock (Do not use this setting)</td></tr>
</table></td></tr>
<tr><td>389</td><td>CHAPTER 6:Low Power Consumption 5. Registers 5.5.8. STS SSCGx Control Register 1 (SYSC0_STSSSCGxCNTR1)</td><td>Error)
[bit18:17] SSCGxFREQ: SSCG PLLx Clock modulation Frequency Selection Bits
These bits indicate the set value of the modulation frequency of the SSCG PLLx clock.
<table>
<tr><th>bit18:17</th><th>Description</th></tr>
<tr><td>00</td><td>Fmod = (1/1024) x Fin</td></tr>
<tr><td>01</td><td>Fmod = (1/2048) x Fin</td></tr>
<tr><td>10 or 11</td><td>Fmod = (1/4096) x Fin</td></tr>
</table>
* Fin: clock before modulation, Fmod: clock after modulation

Correct)
[bit18:17] SSCGxFREQ: SSCG PLLx Clock modulation Frequency Selection Bits
These bits indicate the set value of the modulation frequency of the SSCG PLLx clock.
<table>
<tr><th>bit18:17</th><th>Description</th></tr>
<tr><td>00</td><td>Fmod = (1/1024) x Fin</td></tr>
<tr><td>01</td><td>Fmod = (1/2048) x Fin</td></tr>
<tr><td>10 or 11</td><td>Fmod = (1/4096) x Fin</td></tr>
</table>
* Fin: SSCG reference clock frequency, Fmod: Modulation frequency</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>389</td><td>CHAPTER 6:Low Power Consumption 5. Registers 5.5.8. STS SSCGx Control Register 1 (SYSC0_STS SSCGxCNTR 1)</td><td>Added the Below:

Correct)
[bit16] SSCGxMODE: SSCG PLLx modulation Mode Setting Bit
This bit indicates the set value for the modulation mode of the SSCG PLLx clock.
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Set Down spread mode.</td></tr>
<tr><td>1</td><td>Set Center spread mode.</td></tr>
</table>

</td></tr>
<tr><td>401</td><td>CHAPTER 6:Low Power Consumption 5. Registers 5.6.1. System ID Register (SYSC0_SYSIDR)</td><td>Added the Below:

Correct)
5.6.1. System ID Register (SYSC0_SYSIDR)
The SYSC0_SYSIDR register displays model information
<table>
<tr><td>bit</td><td>31</td><td>0</td></tr>
<tr><td>Field</td><td colspan="2">CHIPID</td></tr>
<tr><td>R/W Attribute</td><td colspan="2">R,WX</td></tr>
<tr><td>Protection Attribute</td><td colspan="2">WPS</td></tr>
<tr><td>Initial Value</td><td colspan="2">Product Specification</td></tr>
</table>
[bit31:0] CHIPID: Chip ID Display Bits
Refer to the product hardware manual</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>402</td><td>CHAPTER 6:Low Power Consumption 5. Registers 5.6.2. Platform ID Register (SYSC0_SYSPFIDR)</td><td>Added the Below:

Correct)
5.6.2. Platform ID Register (SYSC0_SYSPFIDR)
The SYSC0_SYSPFIDR register indicates platform information.
<table>
<tr><td>bit</td><td>31 0</td></tr>
<tr><td>Field</td><td>PFID</td></tr>
<tr><td>R/W attribute</td><td>R,WX</td></tr>
<tr><td>Protection Attribute</td><td>WPS</td></tr>
<tr><td>Initial Value</td><td>Product Specification</td></tr>
</table>
[bit31:0] PFID: Platform ID Display Bits
This bit is a specific fixed value. The value has no meaning.
Write access has no effect.</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>452</td><td>CHAPTER 6:Low Power Consumption 5. Registers 5.11.1. RUN Clock Selection Register 0 (SYSC1_RUN CKSELR0)</td><td>Added the Below:

[bit27:24] HSSPICSL: HSSPI Clock Selection Bits
<table><tr><th>Bit27:24</th><th>Description</th></tr><tr><td>0011</td><td>Sub clock (Do not use this setting)</td></tr></table></td></tr>
<tr><td>453</td><td>CHAPTER 6:Low Power Consumption 5. Registers 5.11.1. RUN Clock Selection Register 0 (SYSC1_RUN CKSELR0)</td><td>Added the Below:

[bit2:0] CD0CSL: Clock Domain 0 Clock Selection Bits
<table><tr><th>Bit2:0</th><th>Description</th></tr><tr><td>011</td><td>Sub clock (Do not use this setting)</td></tr></table></td></tr>
<tr><td>455</td><td>CHAPTER 6:Low Power Consumption 5. Registers 5.11.2. RUN Clock Selection Register 1 (SYSC1_RUN CKSELR1)</td><td>Added the Below:

[bit27:24] CD4CSL: Clock Domain 4 Clock Selection Bits
<table><tr><th>Bit27:24</th><th>Description</th></tr><tr><td>0011</td><td>Sub clock (Do not use this setting)</td></tr></table></td></tr>
<tr><td>455 to 456</td><td>CHAPTER 6:Low Power Consumption 5. Registers 5.11.2. RUN Clock Selection Register 1 (SYSC1_RUN CKSELR1)</td><td>Added the Below:

[bit19:16] CD3CSL: Clock Domain 3 Clock Selection Bits
<table><tr><th>Bit19:16</th><th>Description</th></tr><tr><td>0011</td><td>Sub clock (Do not use this setting)</td></tr></table>
[bit11:8] CD2CSL: Clock Domain 2 Clock Selection Bits
<table><tr><th>Bit11:8</th><th>Description</th></tr><tr><td>0011</td><td>Sub clock (Do not use this setting)</td></tr></table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>456</td><td>CHAPTER 6:Low Power Consumption 5. Registers 5.11.2. RUN Clock Selection Register 1 (SYSC1_RUNCKSELR1)</td><td>Added the Below:

[bit3:0] CD1CSL: Clock Domain 1 Clock Selection Bits
<table><tr><th>Bit3:0</th><th>Description</th></tr><tr><td>0011</td><td>Sub clock (Do not use this setting)</td></tr></table></td></tr>
<tr><td>457</td><td>CHAPTER 6:Low Power Consumption 5. Registers 5.11.3. RUN Clock Selection Register 2 (SYSC1_RUNCKSELR2)</td><td>Added the Below:

[bit10:8] TRCCSL: TRC Clock Selection Bits
<table><tr><th>Bit10:8</th><th>Description</th></tr><tr><td>011</td><td>Sub clock (Do not use this setting)</td></tr></table></td></tr>
<tr><td>458</td><td>CHAPTER 6:Low Power Consumption 5. Registers 5.11.3. RUN Clock Selection Register 2 (SYSC1_RUNCKSELR2)</td><td>Added the Below:

[bit3:0] CD5CSL: Clock Domain 5 Clock Selection Bits
<table><tr><th>Bit3:0</th><th>Description</th></tr><tr><td>0011</td><td>Sub clock (Do not use this setting)</td></tr></table></td></tr>
</table>

Page	Section	Change Results
464	CHAPTER 6:Low Power Consumption 5. Registers 5.11.5. RUN Clock Source Enable Register 1 (SYSC1_RUNCKER1)	Error)

bit 31 30 29 28 27 26 25 24

Field	Reserved	ENCLK CD3B1	ENCLK CD3B0	ENCLK CD3A1	ENCLK CD3A0	ENCLK CD3
R/W Attribute	R0,WX	R/W	R/W	R/W	R/W	R/W
Protection Attribute	WPS					
Initial Value	000	1	1	1	1	1

bit 23 22 21 20 19 18 17 16

Field	Reserved	ENCLK CD2B1	ENCLK CD2B0	ENCLK CD2A1	ENCLK CD2A0	ENCLK CD2
R/W Attribute	R0,WX	R/W	R/W	R/W	R/W	R/W
Protection Attribute	WPS					
Initial Value	000	1	1	1	1	1

bit 15 14 13 12 11 10 9 8

Field	Reserved	ENCLK CD1B1	ENCLK CD1B0	ENCLK CD1A1	ENCLK CD1A0	ENCLK CD1
R/W Attribute	R0,WX	R/W	R/W	R/W	R/W	R/W
Protection Attribute	WPS					
Initial Value	000	1	1	1	1	1

Correct)

bit 31 30 29 28 27 26 25 24

Field	Reserved	ENCLK CD3B1	ENCLK CD3B0	ENCLK CD3A1	ENCLK CD3A0	ENCLK CD3
R/W Attribute	R0,WX	R/W1	R/W1	R/W1	R/W1	R/W1
Protection Attribute	WPS					
Initial Value	000	1	1	1	1	1

bit 23 22 21 20 19 18 17 16

Field	Reserved	ENCLK CD2B1	ENCLK CD2B0	ENCLK CD2A1	ENCLK CD2A0	ENCLK CD2
R/W Attribute	R0,WX	R/W1	R/W1	R/W1	R/W1	R/W1
Protection Attribute	WPS					
Initial Value	000	1	1	1	1	1

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>464</td><td>CHAPTER 6:Low Power Consumption 5. Registers 5.11.5. RUN Clock Source Enable Register 1 (SYSC1_RUNCKER1)</td><td>Correct)
<table>
<tr><td>bit</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td>Field</td><td colspan="3">Reserved</td><td>ENCLKCD1B1</td><td>ENCLKCD1B0</td><td>ENCLKCD1A1</td><td>ENCLKCD1A0</td><td>ENCLKCD1</td></tr>
<tr><td>R/W Attribute</td><td colspan="3">R0,WX</td><td>R/W1</td><td>R/W1</td><td>R/W1</td><td>R/W1</td><td>R/W1</td></tr>
<tr><td>Protection Attribute</td><td colspan="8">WPS</td></tr>
<tr><td>Initial Value</td><td colspan="3">000</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr>
</table></td></tr>
<tr><td>464 to 465</td><td>CHAPTER 6:Low Power Consumption 5. Registers 5.11.5. RUN Clock Source Enable Register 1 (SYSC1_RUNCKER1)</td><td>Error)
[bit28] ENCLKCD3B1: CD3B1 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Disable clock oscillation.</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
[bit27] ENCLKCD3B0: CD3B0 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Disable clock oscillation.</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
[bit26] ENCLKCD3A1: CD3A1 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Disable clock oscillation.</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
Correct)
[bit28] ENCLKCD3B1: CD3B1 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Setting '0' is prohibited</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
[bit27] ENCLKCD3B0: CD3B0 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Setting '0' is prohibited</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
[bit26] ENCLKCD3A1: CD3A1 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Setting '0' is prohibited</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>465
to
466</td><td>CHAPTER 6:Low Power Consumption
5. Registers
5.11.5. RUN Clock Source Enable Register 1 (SYSC1_RUNCKER1)</td><td>
Error)
[bit25] ENCLKCD3A0: CD3A0 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Disable clock oscillation.</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
[bit24] ENCLKCD3: CD3 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Disable clock oscillation.</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
[bit20] ENCLKCD2B1: CD2B1 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Disable clock oscillation.</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
[bit19] ENCLKCD2B0: CD2B0 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Disable clock oscillation.</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
[bit18] ENCLKCD2A1: CD2A1 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Disable clock oscillation.</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
[bit17] ENCLKCD2A0: CD2A0 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Disable clock oscillation.</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
[bit16] ENCLKCD2: CD2 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Disable clock oscillation.</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
Correct)
[bit25] ENCLKCD3A0: CD3A0 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Setting '0' is prohibited</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>465
to
466</td><td>CHAPTER 6:Low Power Consumption
5. Registers
5.11.5. RUN Clock Source Enable Register 1 (SYSC1_RUNCKER1)</td><td>Correct)
[bit24] ENCLKCD3: CD3 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Setting '0' is prohibited</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
[bit20] ENCLKCD2B1: CD2B1 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Setting '0' is prohibited</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
[bit19] ENCLKCD2B0: CD2B0 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Setting '0' is prohibited</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
[bit18] ENCLKCD2A1: CD2A1 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Setting '0' is prohibited</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
[bit17] ENCLKCD2A0: CD2A0 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Setting '0' is prohibited</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
[bit16] ENCLKCD2: CD2 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Setting '0' is prohibited</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>466
to
467</td><td>CHAPTER 6:Low Power Consumption
5. Registers
5.11.5. RUN Clock Source Enable Register 1 (SYSC1_RUNCKER1)</td><td>
Error)
[bit12] ENCLKCD1B1: CD1B1 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Disable clock oscillation.</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
[bit11] ENCLKCD1B0: CD1B0 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Disable clock oscillation.</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
[bit10] ENCLKCD1A1: CD1A1 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Disable clock oscillation.</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
[bit9] ENCLKCD1A0: CD1A0 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Disable clock oscillation.</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
[bit8] ENCLKCD1: CD1 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Disable clock oscillation.</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
Correct)
[bit12] ENCLKCD1B1: CD1B1 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Setting '0' is prohibited</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
[bit11] ENCLKCD1B0: CD1B0 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Setting '0' is prohibited</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
[bit10] ENCLKCD1A1: CD1A1 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Setting '0' is prohibited</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>466
to
467</td><td>CHAPTER 6:Low Power Consumption
5. Registers
5.11.5. RUN Clock Source Enable Register 1 (SYSC1_RUNCKER1)</td><td>Correct)
[bit9] ENCLKCD1A0: CD1A0 Clock Oscillation Enable Bit
<table><tr><th>Bit</th><th>Description</th></tr><tr><td>0</td><td>Setting '0' is prohibited</td></tr><tr><td>1</td><td>Enable clock oscillation.</td></tr></table>
[bit8] ENCLKCD1: CD1 Clock Oscillation Enable Bit
<table><tr><th>Bit</th><th>Description</th></tr><tr><td>0</td><td>Setting '0' is prohibited</td></tr><tr><td>1</td><td>Enable clock oscillation.</td></tr></table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>468</td><td>CHAPTER 6:Low Power Consumption 5. Registers 5.11.6. RUN Clock Source Enable Register 2 (SYSC1_RUNCKER2)</td><td>
Error)
<table>
<tr><td>bit</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td>Field</td><td colspan="3">Reserved</td><td>ENCLKCD5B1</td><td>ENCLKCD5B0</td><td>ENCLKCD5A1</td><td>ENCLKCD5A0</td><td>ENCLKCD5</td></tr>
<tr><td>R/W Attribute</td><td colspan="3">R0,WX</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td></tr>
<tr><td>Protection Attribute</td><td colspan="8">WPS</td></tr>
<tr><td>Initial Value</td><td colspan="3">000</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr>
</table>
<table>
<tr><td>bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Field</td><td colspan="3">Reserved</td><td>ENCLKCD4B1</td><td>ENCLKCD4B0</td><td>ENCLKCD4A1</td><td>ENCLKCD4A0</td><td>ENCLKCD4</td></tr>
<tr><td>R/W Attribute</td><td colspan="3">R0,WX</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td></tr>
<tr><td>Protection Attribute</td><td colspan="8">WPS</td></tr>
<tr><td>Initial Value</td><td colspan="3">000</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr>
</table>
Correct)
<table>
<tr><td>bit</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td>Field</td><td colspan="3">Reserved</td><td>ENCLKCD5B1</td><td>ENCLKCD5B0</td><td>ENCLKCD5A1</td><td>ENCLKCD5A0</td><td>ENCLKCD5</td></tr>
<tr><td>R/W Attribute</td><td colspan="3">R0,WX</td><td>R/W1</td><td>R/W1</td><td>R/W1</td><td>R/W1</td><td>R/W1</td></tr>
<tr><td>Protection Attribute</td><td colspan="8">WPS</td></tr>
<tr><td>Initial Value</td><td colspan="3">000</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr>
</table>
<table>
<tr><td>bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Field</td><td colspan="3">Reserved</td><td>ENCLKCD4B1</td><td>ENCLKCD4B0</td><td>ENCLKCD4A1</td><td>ENCLKCD4A0</td><td>ENCLKCD4</td></tr>
<tr><td>R/W Attribute</td><td colspan="3">R0,WX</td><td>R/W1</td><td>R/W1</td><td>R/W1</td><td>R/W1</td><td>R/W1</td></tr>
<tr><td>Protection Attribute</td><td colspan="8">WPS</td></tr>
<tr><td>Initial Value</td><td colspan="3">000</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>468
to
469</td><td>CHAPTER 6:Low Power Consumption
5. Registers
5.11.6. RUN Clock Source Enable Register 2 (SYSC1_RUNCKER2)</td><td>
Error)
[bit12] ENCLKCD5B1: CD5B1 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Disable clock oscillation.</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
[bit11] ENCLKCD5B0: CD5B0 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Disable clock oscillation.</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
[bit10] ENCLKCD5A1: CD5A1 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Disable clock oscillation.</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
[bit9] ENCLKCD5A0: CD5A0 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Disable clock oscillation.</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
Correct)
[bit12] ENCLKCD5B1: CD5B1 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Setting '0' is prohibited</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
[bit11] ENCLKCD5B0: CD5B0 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Setting '0' is prohibited</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
[bit10] ENCLKCD5A1: CD5A1 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Setting '0' is prohibited</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
[bit9] ENCLKCD5A0: CD5A0 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Setting '0' is prohibited</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>469
to
470</td><td>CHAPTER 6:Low Power Consumption 5. Registers 5.11.6. RUN Clock Source Enable Register 2 (SYSC1_RUNCKER2)</td><td>
Error)
[bit8] ENCLKCD5: CD5 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Disable clock oscillation.</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
[bit4] ENCLKCD4B1: CD4B1 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Disable clock oscillation.</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
[bit3] ENCLKCD4B0: CD4B0 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Disable clock oscillation.</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
[bit2] ENCLKCD4A1: CD4A1 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Disable clock oscillation.</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
[bit1] ENCLKCD4A0: CD4A0 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Disable clock oscillation.</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
[bit0] ENCLKCD4: CD4 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Disable clock oscillation.</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
Correct)
[bit8] ENCLKCD5: CD5 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Setting '0' is prohibited</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
[bit4] ENCLKCD4B1: CD4B1 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Setting '0' is prohibited</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>469
to
470</td><td>CHAPTER 6:Low Power Consumption
5. Registers
5.11.6. RUN Clock Source Enable Register 2 (SYSC1_RUNCKER2)</td><td>Correct)
[bit3] ENCLKCD4B0: CD4B0 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Setting '0' is prohibited</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
[bit2] ENCLKCD4A1: CD4A1 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Setting '0' is prohibited</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
[bit1] ENCLKCD4A0: CD4A0 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Setting '0' is prohibited</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
[bit0] ENCLKCD4: CD4 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Setting '0' is prohibited</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>472</td><td>CHAPTER 6:Low Power Consumption 5. Registers 5.11.7. RUN Clock Divider Register 0 (SYSC1_RUNCKDIVR0)</td><td>Error)

[bit20:16] TRCDIV: TRC Clock Divider Setting Bits
These bits set the division ratio of the TRC clock from the source clock (clock domain TRC).
<table>
<tr><th>bit20:16</th><th>Description</th></tr>
<tr><td>0000</td><td>No division</td></tr>
<tr><td>0001</td><td>Divided by 2</td></tr>
<tr><td>0010</td><td>Divided by 3</td></tr>
<tr><td>0011</td><td>Divided by 4</td></tr>
<tr><td>. . .</td><td>. . .</td></tr>
<tr><td>1101</td><td>Divided by 14</td></tr>
<tr><td>1110</td><td>Divided by 15</td></tr>
<tr><td>1111</td><td>Divided by 16</td></tr>
</table>

Correct)

[bit20:16] TRCDIV: TRC Clock Divider Setting Bits
These bits set the division ratio of the TRC clock from the source clock (clock domain TRC).
<table>
<tr><th>bit20:16</th><th>Description</th></tr>
<tr><td>00000</td><td>No division</td></tr>
<tr><td>00001</td><td>Divided by 2</td></tr>
<tr><td>00010</td><td>Divided by 3</td></tr>
<tr><td>00011</td><td>Divided by 4</td></tr>
<tr><td>. . .</td><td>. . .</td></tr>
<tr><td>11101</td><td>Divided by 30</td></tr>
<tr><td>11110</td><td>Divided by 31</td></tr>
<tr><td>11111</td><td>Divided by 32</td></tr>
</table></td></tr>
<tr><td>500</td><td>CHAPTER 6:Low Power Consumption 5. Registers 5.12.1. PSS Clock Selection Register 0 (SYSC1_PSSCKSELR0)</td><td>Added the Below:

[bit27:24] HSSPICSL: HSSPI Clock Selection Bits
<table>
<tr><th>bit27:24</th><th>Description</th></tr>
<tr><td>0011</td><td>Sub clock (Do not use this setting)</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>502</td><td>CHAPTER 6:Low Power Consumption 5. Registers 5.12.1. PSS Clock Selection Register 0 (SYSC1_PSSCKSELR0)</td><td>Added the Below:

[bit2:0] CD0CSL: Clock Domain 0 Clock Selection Bits
<table><tr><th>bit2:0</th><th>Description</th></tr><tr><td>011</td><td>Sub clock (Do not use this setting)</td></tr></table></td></tr>
<tr><td>503</td><td>CHAPTER 6:Low Power Consumption 5. Registers 5.12.2. PSS Clock Selection Register 1 (SYSC1_PSSCKSELR1)</td><td>Added the Below:

[bit27:24] CD4CSL: Clock Domain 4 Clock Selection Bits
<table><tr><th>bit27:24</th><th>Description</th></tr><tr><td>0011</td><td>Sub clock (Do not use this setting)</td></tr></table></td></tr>
<tr><td>504</td><td>CHAPTER 6:Low Power Consumption 5. Registers 5.12.2. PSS Clock Selection Register 1 (SYSC1_PSSCKSELR1)</td><td>Added the Below:

[bit19:16] CD3CSL: Clock Domain 3 Clock Selection Bits
<table><tr><th>bit19:16</th><th>Description</th></tr><tr><td>0011</td><td>Sub clock (Do not use this setting)</td></tr></table>
[bit11:8] CD2CSL: Clock Domain 2 Clock Selection Bits
<table><tr><th>bit11:8</th><th>Description</th></tr><tr><td>0011</td><td>Sub clock (Do not use this setting)</td></tr></table></td></tr>
<tr><td>505</td><td>CHAPTER 6:Low Power Consumption 5. Registers 5.12.2. PSS Clock Selection Register 1 (SYSC1_PSSCKSELR1)</td><td>Added the Below:

[bit3:0] CD1CSL: Clock Domain 1 Clock Selection Bits
<table><tr><th>bit3:0</th><th>Description</th></tr><tr><td>0011</td><td>Sub clock (Do not use this setting)</td></tr></table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>506</td><td>CHAPTER 6:Low Power Consumption 5. Registers 5.12.3. PSS Clock Selection Register 2 (SYSC1_PSSCKSELR2)</td><td>Added the Below:
[bit10:8] TRCCSL: TRC Clock Selection Bits
<table>
<tr><th>bit10:8</th><th>Description</th></tr>
<tr><td>011</td><td>Sub clock (Do not use this setting)</td></tr>
</table></td></tr>
<tr><td>507</td><td>CHAPTER 6:Low Power Consumption 5. Registers 5.12.3. PSS Clock Selection Register 2 (SYSC1_PSSCKSELR2)</td><td>Added the Below:

[bit3:0] CD5CSL: Clock Domain 5 Clock Selection Bits
<table>
<tr><th>bit3:0</th><th>Description</th></tr>
<tr><td>0011</td><td>Sub clock (Do not use this setting)</td></tr>
</table></td></tr>
<tr><td>508</td><td>CHAPTER 6:Low Power Consumption 5. Registers 5.12.4. PSS Clock Source Enable Register 0 (SYSC1_PSSCKER0)</td><td>Error)
<table>
<tr><td>bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Field</td><td>ENCLK HPM2</td><td>ENCLK TRC</td><td>ENCLK DBG</td><td>ENCLK ATB</td><td colspan="3">Reserved</td><td>ENCLK CPU0</td></tr>
<tr><td>R/W Attribute</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td colspan="3">R0,WX</td><td>R0/W</td></tr>
<tr><td>Protection Attribute</td><td colspan="8">WPS</td></tr>
<tr><td>Initial Value</td><td>1</td><td>1</td><td>1</td><td>1</td><td colspan="3">000</td><td>0</td></tr>
</table>
Correct)
<table>
<tr><td>bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Field</td><td>ENCLK HPM2</td><td>ENCLK TRC</td><td>ENCLK DBG</td><td>ENCLK ATB</td><td colspan="3">Reserved</td><td>ENCLK CPU0</td></tr>
<tr><td>R/W Attribute</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td colspan="3">R0,WX</td><td>R0,WX</td></tr>
<tr><td>Protection Attribute</td><td colspan="8">WPS</td></tr>
<tr><td>Initial Value</td><td>1</td><td>1</td><td>1</td><td>1</td><td colspan="3">000</td><td>0</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>513</td><td>CHAPTER 6:Low Power Consumption 5. Registers 5.12.5. PSS Clock Source Enable Register 1 (SYSC1_PSSCKER1)</td><td>
Error)
<table>
<tr><td>bit</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td>Field</td><td colspan="3">Reserved</td><td>ENCLK CD3B1</td><td>ENCLK CD3B0</td><td>ENCLK CD3A1</td><td>ENCLK CD3A0</td><td>ENCLK CD3</td></tr>
<tr><td>R/W Attribute</td><td colspan="3">R0,WX</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td></tr>
<tr><td>Protection Attribute</td><td colspan="8">WPS</td></tr>
<tr><td>Initial Value</td><td colspan="3">000</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr>
</table>
<table>
<tr><td>bit</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td>Field</td><td colspan="3">Reserved</td><td>ENCLK CD2B1</td><td>ENCLK CD2B0</td><td>ENCLK CD2A1</td><td>ENCLK CD2A0</td><td>ENCLK CD2</td></tr>
<tr><td>R/W Attribute</td><td colspan="3">R0,WX</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td></tr>
<tr><td>Protection Attribute</td><td colspan="8">WPS</td></tr>
<tr><td>Initial Value</td><td colspan="3">000</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr>
</table>
<table>
<tr><td>bit</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td>Field</td><td colspan="3">Reserved</td><td>ENCLK CD1B1</td><td>ENCLK CD1B0</td><td>ENCLK CD1A1</td><td>ENCLK CD1A0</td><td>ENCLK CD1</td></tr>
<tr><td>R/W Attribute</td><td colspan="3">R0,WX</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td></tr>
<tr><td>Protection Attribute</td><td colspan="8">WPS</td></tr>
<tr><td>Initial Value</td><td colspan="3">000</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr>
</table>
Correct)
<table>
<tr><td>bit</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td>Field</td><td colspan="3">Reserved</td><td>ENCLK CD3B1</td><td>ENCLK CD3B0</td><td>ENCLK CD3A1</td><td>ENCLK CD3A0</td><td>ENCLK CD3</td></tr>
<tr><td>R/W Attribute</td><td colspan="3">R0,WX</td><td>R/W1</td><td>R/W1</td><td>R/W1</td><td>R/W1</td><td>R/W1</td></tr>
<tr><td>Protection Attribute</td><td colspan="8">WPS</td></tr>
<tr><td>Initial Value</td><td colspan="3">000</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr>
</table>
<table>
<tr><td>bit</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td>Field</td><td colspan="3">Reserved</td><td>ENCLK CD2B1</td><td>ENCLK CD2B0</td><td>ENCLK CD2A1</td><td>ENCLK CD2A0</td><td>ENCLK CD2</td></tr>
<tr><td>R/W Attribute</td><td colspan="3">R0,WX</td><td>R/W1</td><td>R/W1</td><td>R/W1</td><td>R/W1</td><td>R/W1</td></tr>
<tr><td>Protection Attribute</td><td colspan="8">WPS</td></tr>
<tr><td>Initial Value</td><td colspan="3">000</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>513</td><td>CHAPTER 6:Low Power Consumption 5. Registers 5.12.5. PSS Clock Source Enable Register 1 (SYSC1_PSSCKER1)</td><td>Correct)
<table>
<tr><td>bit</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td>Field</td><td colspan="3">Reserved</td><td>ENCLKCD1B1</td><td>ENCLKCD1B0</td><td>ENCLKCD1A1</td><td>ENCLKCD1A0</td><td>ENCLKCD1</td></tr>
<tr><td>R/W Attribute</td><td colspan="3">R0,WX</td><td>R/W1</td><td>R/W1</td><td>R/W1</td><td>R/W1</td><td>R/W1</td></tr>
<tr><td>Protection Attribute</td><td colspan="8">WPS</td></tr>
<tr><td>Initial Value</td><td colspan="3">000</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr>
</table></td></tr>
<tr><td>513 to 514</td><td>CHAPTER 6:Low Power Consumption 5. Registers 5.12.5. PSS Clock Source Enable Register 1 (SYSC1_PSSCKER1)</td><td>Error)
[bit28] ENCLKCD3B1: CD3B1 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Disable clock oscillation.</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
[bit27] ENCLKCD3B0: CD3B0 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Disable clock oscillation.</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
[bit26] ENCLKCD3A1: CD3A1 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Disable clock oscillation.</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
Correct)
[bit28] ENCLKCD3B1: CD3B1 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Setting '0' is prohibited</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
[bit27] ENCLKCD3B0: CD3B0 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Setting '0' is prohibited</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>513
to
514</td><td>CHAPTER 6:Low Power Consumption
5. Registers
5.12.5. PSS Clock Source Enable Register 1 (SYSC1_PSSCKER1)</td><td>Correct)
[bit26] ENCLKCD3A1: CD3A1 Clock Oscillation Enable Bit
<table><tr><th>Bit</th><th>Description</th></tr><tr><td>0</td><td>Setting '0' is prohibited</td></tr><tr><td>1</td><td>Enable clock oscillation.</td></tr></table></td></tr>
<tr><td>514
to
515</td><td>CHAPTER 6:Low Power Consumption
5. Registers
5.12.5. PSS Clock Source Enable Register 1 (SYSC1_PSSCKER1)</td><td>Error)
[bit25]ENCLKCD3A0: CD3A0 Clock Oscillation Enable Bit
<table><tr><th>Bit</th><th>Description</th></tr><tr><td>0</td><td>Disable clock oscillation.</td></tr><tr><td>1</td><td>Enable clock oscillation.</td></tr></table>
[bit24] ENCLKCD3: CD3 Clock Oscillation Enable Bit
<table><tr><th>Bit</th><th>Description</th></tr><tr><td>0</td><td>Disable clock oscillation.</td></tr><tr><td>1</td><td>Enable clock oscillation.</td></tr></table>
[bit20] ENCLKCD2B1: CD2B1 Clock Oscillation Enable Bit
<table><tr><th>Bit</th><th>Description</th></tr><tr><td>0</td><td>Disable clock oscillation.</td></tr><tr><td>1</td><td>Enable clock oscillation.</td></tr></table>
[bit19] ENCLKCD2B0: CD2B0 Clock Oscillation Enable Bit
<table><tr><th>Bit</th><th>Description</th></tr><tr><td>0</td><td>Disable clock oscillation.</td></tr><tr><td>1</td><td>Enable clock oscillation.</td></tr></table>
[bit18] ENCLKCD2A1: CD2A1 Clock Oscillation Enable Bit
<table><tr><th>Bit</th><th>Description</th></tr><tr><td>0</td><td>Disable clock oscillation.</td></tr><tr><td>1</td><td>Enable clock oscillation.</td></tr></table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>514
to
515</td><td>CHAPTER 6:Low Power Consumption 5. Registers 5.12.5. PSS Clock Source Enable Register 1 (SYSC1_PSSCKER1)</td><td>
Error)
[bit17] ENCLKCD2A0: CD2A0 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Disable clock oscillation.</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
[bit16] ENCLKCD2: CD2 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Disable clock oscillation.</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
Correct)
[bit25]ENCLKCD3A0: CD3A0 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Setting '0' is prohibited</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
[bit24] ENCLKCD3: CD3 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Setting '0' is prohibited</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
[bit20] ENCLKCD2B1: CD2B1 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Setting '0' is prohibited</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
[bit19] ENCLKCD2B0: CD2B0 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Setting '0' is prohibited</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
[bit18] ENCLKCD2A1: CD2A1 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Setting '0' is prohibited</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>514
to
515</td><td>CHAPTER 6:Low Power Consumption
5. Registers
5.12.5. PSS Clock Source Enable Register 1 (SYSC1_PSSCKER1)</td><td>Correct)
[bit17] ENCLKCD2A0: CD2A0 Clock Oscillation Enable Bit
<table><tr><th>Bit</th><th>Description</th></tr><tr><td>0</td><td>Setting '0' is prohibited</td></tr><tr><td>1</td><td>Enable clock oscillation.</td></tr></table>
[bit16] ENCLKCD2: CD2 Clock Oscillation Enable Bit
<table><tr><th>Bit</th><th>Description</th></tr><tr><td>0</td><td>Setting '0' is prohibited</td></tr><tr><td>1</td><td>Enable clock oscillation.</td></tr></table></td></tr>
<tr><td>515
to
516</td><td>CHAPTER 6:Low Power Consumption
5. Registers
5.12.5. PSS Clock Source Enable Register 1 (SYSC1_PSSCKER1)</td><td>Error)
[bit12] ENCLKCD1B1: CD1B1 Clock Oscillation Enable Bit
<table><tr><th>Bit</th><th>Description</th></tr><tr><td>0</td><td>Disable clock oscillation.</td></tr><tr><td>1</td><td>Enable clock oscillation.</td></tr></table>
[bit11] ENCLKCD1B0: CD1B0 Clock Oscillation Enable Bit
<table><tr><th>Bit</th><th>Description</th></tr><tr><td>0</td><td>Disable clock oscillation.</td></tr><tr><td>1</td><td>Enable clock oscillation.</td></tr></table>
[bit10] ENCLKCD1A1: CD1A1 Clock Oscillation Enable Bit
<table><tr><th>Bit</th><th>Description</th></tr><tr><td>0</td><td>Disable clock oscillation.</td></tr><tr><td>1</td><td>Enable clock oscillation.</td></tr></table>
[bit9] ENCLKCD1A0: CD1A0 Clock Oscillation Enable Bit
<table><tr><th>Bit</th><th>Description</th></tr><tr><td>0</td><td>Disable clock oscillation.</td></tr><tr><td>1</td><td>Enable clock oscillation.</td></tr></table>
[bit8] ENCLKCD1: CD1 Clock Oscillation Enable Bit
<table><tr><th>Bit</th><th>Description</th></tr><tr><td>0</td><td>Disable clock oscillation.</td></tr><tr><td>1</td><td>Enable clock oscillation.</td></tr></table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>515
to
516</td><td>CHAPTER 6:Low Power Consumption
5. Registers
5.12.5. PSS Clock Source Enable Register 1 (SYSC1_PSSCKER1)</td><td>Correct)
[bit12] ENCLKCD1B1: CD1B1 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Setting '0' is prohibited</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
[bit11] ENCLKCD1B0: CD1B0 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Setting '0' is prohibited</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
[bit10] ENCLKCD1A1: CD1A1 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Setting '0' is prohibited</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
[bit9] ENCLKCD1A0: CD1A0 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Setting '0' is prohibited</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
[bit8] ENCLKCD1: CD1 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Setting '0' is prohibited</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>517</td><td>CHAPTER 6:Low Power Consumption 5. Registers 5.12.6. PSS Clock Source Enable Register 2(SYSC1_PSSCKER2)</td><td>
Error)
<table>
<tr><td>bit</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td>Field</td><td colspan="3">Reserved</td><td>ENCLKCD5B1</td><td>ENCLKCD5B0</td><td>ENCLKCD5A1</td><td>ENCLKCD5A0</td><td>ENCLKCD5</td></tr>
<tr><td>R/W Attribute</td><td colspan="3">R0,WX</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td></tr>
<tr><td>Protection Attribute</td><td colspan="8">WPS</td></tr>
<tr><td>Initial Value</td><td colspan="3">000</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr>
</table>
<table>
<tr><td>bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Field</td><td colspan="3">Reserved</td><td>ENCLKCD4B1</td><td>ENCLKCD4B0</td><td>ENCLKCD4A1</td><td>ENCLKCD4A0</td><td>ENCLKCD4</td></tr>
<tr><td>R/W Attribute</td><td colspan="3">R0,WX</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td></tr>
<tr><td>Protection Attribute</td><td colspan="8">WPS</td></tr>
<tr><td>Initial Value</td><td colspan="3">000</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr>
</table>
Correct)
<table>
<tr><td>bit</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td>Field</td><td colspan="3">Reserved</td><td>ENCLKCD5B1</td><td>ENCLKCD5B0</td><td>ENCLKCD5A1</td><td>ENCLKCD5A0</td><td>ENCLKCD5</td></tr>
<tr><td>R/W Attribute</td><td colspan="3">R0,WX</td><td>R/W1</td><td>R/W1</td><td>R/W1</td><td>R/W1</td><td>R/W1</td></tr>
<tr><td>Protection Attribute</td><td colspan="8">WPS</td></tr>
<tr><td>Initial Value</td><td colspan="3">000</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr>
</table>
<table>
<tr><td>bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Field</td><td colspan="3">Reserved</td><td>ENCLKCD4B1</td><td>ENCLKCD4B0</td><td>ENCLKCD4A1</td><td>ENCLKCD4A0</td><td>ENCLKCD4</td></tr>
<tr><td>R/W Attribute</td><td colspan="3">R0,WX</td><td>R/W1</td><td>R/W1</td><td>R/W1</td><td>R/W1</td><td>R/W1</td></tr>
<tr><td>Protection Attribute</td><td colspan="8">WPS</td></tr>
<tr><td>Initial Value</td><td colspan="3">000</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>517
to
518</td><td>CHAPTER 6:Low Power Consumption 5. Registers 5.12.6. PSS Clock Source Enable Register 2(SYSC1_PSSCKER2)</td><td>
Error)
[bit12] ENCLKCD5B1: CD5B1 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Disable clock oscillation.</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
[bit11] ENCLKCD5B0: CD5B0 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Disable clock oscillation.</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
[bit10] ENCLKCD5A1: CD5A1 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Disable clock oscillation.</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
[bit9] ENCLKCD5A0: CD5A0 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Disable clock oscillation.</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
Correct)
[bit12] ENCLKCD5B1: CD5B1 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Setting '0' is prohibited</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
[bit11] ENCLKCD5B0: CD5B0 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Setting '0' is prohibited</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
[bit10] ENCLKCD5A1: CD5A1 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Setting '0' is prohibited</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
[bit9] ENCLKCD5A0: CD5A0 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Setting '0' is prohibited</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>518
to
519</td><td>CHAPTER 6:Low Power Consumption
5. Registers
5.12.6. PSS Clock Source Enable Register 2(SYSC1_PSSCKER2)</td><td>
Error)
[bit8] ENCLKCD5: CD5 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Disable clock oscillation.</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
[bit4] ENCLKCD4B1: CD4B1 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Disable clock oscillation.</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
[bit3] ENCLKCD4B0: CD4B0 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Disable clock oscillation.</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
[bit2] ENCLKCD4A1: CD4A1 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Disable clock oscillation.</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
[bit1] ENCLKCD4A0: CD4A0 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Disable clock oscillation.</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
[bit0] ENCLKCD4: CD4 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Disable clock oscillation.</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
Correct)
[bit8] ENCLKCD5: CD5 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Setting '0' is prohibited</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
[bit4] ENCLKCD4B1: CD4B1 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Setting '0' is prohibited</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>518
to
519</td><td>CHAPTER 6:Low Power Consumption 5. Registers 5.12.6. PSS Clock Source Enable Register 2(SYSC1_PSSCKER2)</td><td>Correct)
[bit3] ENCLKCD4B0: CD4B0 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Setting '0' is prohibited</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
[bit2] ENCLKCD4A1: CD4A1 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Setting '0' is prohibited</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
[bit1] ENCLKCD4A0: CD4A0 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Setting '0' is prohibited</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table>
[bit0] ENCLKCD4: CD4 Clock Oscillation Enable Bit
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Setting '0' is prohibited</td></tr>
<tr><td>1</td><td>Enable clock oscillation.</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>521</td><td>CHAPTER 6:Low Power Consumption 5. Registers 5.12.7. PSS Clock Divider Register 0 (SYSC1_PSSCKDIVR0)</td><td>Error)

[bit20:16] TRCDIV: TRC Clock Divider Setting Bits
These bits set the division ratio of the TRC clock from the source clock (clock domain TRC).
<table>
<tr><th>bit20:16</th><th>Description</th></tr>
<tr><td>0000</td><td>No division</td></tr>
<tr><td>0001</td><td>Divided by 2</td></tr>
<tr><td>0010</td><td>Divided by 3</td></tr>
<tr><td>0011</td><td>Divided by 4</td></tr>
<tr><td>. . .</td><td>. . .</td></tr>
<tr><td>1101</td><td>Divided by 14</td></tr>
<tr><td>1110</td><td>Divided by 15</td></tr>
<tr><td>1111</td><td>Divided by 16</td></tr>
</table>
Correct)

[bit20:16] TRCDIV: TRC Clock Divider Setting Bits
These bits set the division ratio of the TRC clock from the source clock (clock domain TRC).
<table>
<tr><th>bit20:16</th><th>Description</th></tr>
<tr><td>00000</td><td>No division</td></tr>
<tr><td>00001</td><td>Divided by 2</td></tr>
<tr><td>00010</td><td>Divided by 3</td></tr>
<tr><td>00011</td><td>Divided by 4</td></tr>
<tr><td>. . .</td><td>. . .</td></tr>
<tr><td>11101</td><td>Divided by 30</td></tr>
<tr><td>11110</td><td>Divided by 31</td></tr>
<tr><td>11111</td><td>Divided by 32</td></tr>
</table></td></tr>
<tr><td>549</td><td>CHAPTER 6:Low Power Consumption 5. Registers 5.13.1. APP Clock Selection Register 0 (SYSC1_APPCKSELR0)</td><td>Added the Below:

[bit27:24] HSSPICSL: HSSPI Clock Selection Bits
<table>
<tr><th>bit27:24</th><th>Description</th></tr>
<tr><td>0011</td><td>Sub clock (Do not use this setting)</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>551</td><td>CHAPTER 6:Low Power Consumption 5. Registers 5.13.1. APP Clock Selection Register 0 (SYSC1_APPCKSELR0)</td><td>Added the Below:

[bit2:0] CD0CSL: Clock Domain 0 Clock Selection Bits
<table><tr><th>bit2:0</th><th>Description</th></tr><tr><td>011</td><td>Sub clock (Do not use this setting)</td></tr></table></td></tr>
<tr><td>552</td><td>CHAPTER 6:Low Power Consumption 5. Registers 5.13.2. APP Clock Selection Register 1 (SYSC1_APPCKSELR1)</td><td>Added the Below:

[bit27:24] CD4CSL: Clock Domain 4 Clock Selection Bits
<table><tr><th>bit27:24</th><th>Description</th></tr><tr><td>0011</td><td>Sub clock (Do not use this setting)</td></tr></table></td></tr>
<tr><td>553</td><td>CHAPTER 6:Low Power Consumption 5. Registers 5.13.2. APP Clock Selection Register 1 (SYSC1_APPCKSELR1)</td><td>Added the Below:

[bit19:16] CD3CSL: Clock Domain 3 Clock Selection Bits
<table><tr><th>bit19:16</th><th>Description</th></tr><tr><td>0011</td><td>Sub clock (Do not use this setting)</td></tr></table>
[bit11:8] CD2CSL: Clock Domain 2 Clock Selection Bits
<table><tr><th>bit11:8</th><th>Description</th></tr><tr><td>0011</td><td>Sub clock (Do not use this setting)</td></tr></table></td></tr>
<tr><td>554</td><td>CHAPTER 6:Low Power Consumption 5. Registers 5.13.2. APP Clock Selection Register 1 (SYSC1_APPCKSELR1)</td><td>Added the Below:

[bit3:0] CD1CSL: Clock Domain 1 Clock Selection Bits
<table><tr><th>bit3:0</th><th>Description</th></tr><tr><td>0011</td><td>Sub clock (Do not use this setting)</td></tr></table></td></tr>
<tr><td>555</td><td>CHAPTER 6:Low Power Consumption 5. Registers 5.13.3. APP Clock Selection Register 2 (SYSC1_APPCKSELR2)</td><td>Added the Below:

[bit10:8] TRCCSL: TRC Clock Selection Bits
<table><tr><th>bit10:8</th><th>Description</th></tr><tr><td>011</td><td>Sub clock (Do not use this setting)</td></tr></table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>556</td><td>CHAPTER 6:Low Power Consumption
5. Registers
5.13. APP Profile Register Group (SYSC1)
5.13.3. APP Clock Selection Register 2 (SYSC1_APPCKSELR2)</td><td>Added the Below:

[bit3:0] CD5CSL: Clock Domain 5 Clock Selection Bits
<table><tr><th>bit3:0</th><th>Description</th></tr><tr><td>0011</td><td>Sub clock (Do not use this setting)</td></tr></table></td></tr>
<tr><td>557</td><td>CHAPTER 6:Low Power Consumption
5. Registers
5.13.4. APP Clock Source Enable Register0 (SYSC1_APPCKER0)</td><td>Deleted the below:
Error)
The SYSC1_APPCKSRER0 register indicates the value for setting whether to enable/disable oscillation of the internal operating clock to be updated.

Correct)
The SYSC1_APPCKER0 register indicates the value for setting whether to enable/disable oscillation of the internal operating clock to be updated.</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>570</td><td>CHAPTER 6:Low Power Consumption 5. Registers 5.13.7. APP Clock Divider Register 0 (SYSC1_APPCKDIVR0)</td><td>Error)

[bit20:16] TRCDIV: TRC Clock Divider Setting Bits
These bits indicate the set value of the division ratio of the TRC clock from the source clock (clock domain TRC) to be updated.
<table>
<tr><th>bit20:16</th><th>Description</th></tr>
<tr><td>0000</td><td>No division</td></tr>
<tr><td>0001</td><td>Divided by 2</td></tr>
<tr><td>0010</td><td>Divided by 3</td></tr>
<tr><td>0011</td><td>Divided by 4</td></tr>
<tr><td>. . .</td><td>. . .</td></tr>
<tr><td>1101</td><td>Divided by 14</td></tr>
<tr><td>1110</td><td>Divided by 15</td></tr>
<tr><td>1111</td><td>Divided by 16</td></tr>
</table>

Correct)

[bit20:16] TRCDIV: TRC Clock Divider Setting Bits
These bits indicate the set value of the division ratio of the TRC clock from the source clock (clock domain TRC) to be updated.
<table>
<tr><th>bit20:16</th><th>Description</th></tr>
<tr><td>00000</td><td>No division</td></tr>
<tr><td>00001</td><td>Divided by 2</td></tr>
<tr><td>00010</td><td>Divided by 3</td></tr>
<tr><td>00011</td><td>Divided by 4</td></tr>
<tr><td>. . .</td><td>. . .</td></tr>
<tr><td>11101</td><td>Divided by 30</td></tr>
<tr><td>11110</td><td>Divided by 31</td></tr>
<tr><td>11111</td><td>Divided by 32</td></tr>
</table></td></tr>
<tr><td>598</td><td>CHAPTER 6:Low Power Consumption 5. Registers 5.14.1. STS Clock Selection Register 0 (SYSC1_STSCKSELR0)</td><td>Added the Below:

[bit31:28] HSSPICM: HSSPI Clock Selection Status Bits
<table>
<tr><th>bit31:28</th><th>Description</th></tr>
<tr><td>0011</td><td>Sub clock (Do not use this setting)</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>599</td><td>CHAPTER 6:Low Power Consumption 5. Registers 5.14.1. STS Clock Selection Register 0 (SYSC1_STSCKSELR0)</td><td>Added the Below:

[bit27:24] HSSPICSL: HSSPI Clock Selection Bits
<table><tr><th>bit27:24</th><th>Description</th></tr><tr><td>0011</td><td>Sub clock (Do not use this setting)</td></tr></table></td></tr>
<tr><td>601</td><td>CHAPTER 6:Low Power Consumption 5. Registers 5.14.1. STS Clock Selection Register 0 (SYSC1_STSCKSELR0)</td><td>Added the Below:

[bit6:4] CD0CM: Clock Domain 0 Clock Selection Status Bits
<table><tr><th>Bit6:4</th><th>Description</th></tr><tr><td>011</td><td>Sub clock (Do not use this setting)</td></tr></table></td></tr>
<tr><td>601</td><td>CHAPTER 6:Low Power Consumption 5. Registers 5.14.1. STS Clock Selection Register 0 (SYSC1_STSCKSELR0)</td><td>Added the Below:

[bit2:0] CD0CSL: Clock Domain 0 Clock Selection Bits
<table><tr><th>Bit2:0</th><th>Description</th></tr><tr><td>011</td><td>Sub clock (Do not use this setting)</td></tr></table></td></tr>
<tr><td>602</td><td>CHAPTER 6:Low Power Consumption 5. Registers 5.14.2. STS Clock Selection Register 1 (SYSC1_STSCKSELR1)</td><td>Added the Below:

[bit31:28] CD4CM: Clock Domain 4 Clock Selection Status Bits
<table><tr><th>bit31:28</th><th>Description</th></tr><tr><td>0011</td><td>Sub clock (Do not use this setting)</td></tr></table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>603</td><td>CHAPTER 6:Low Power Consumption 5. Registers 5.14.2. STS Clock Selection Register 1 (SYSC1_STSCKSELR1)</td><td>Added the Below:

[bit27:24] CD4CSL: Clock Domain 4 Clock Selection Bits
<table><tr><th>bit27:24</th><th>Description</th></tr><tr><td>0011</td><td>Sub clock (Do not use this setting)</td></tr></table>
[bit23:20] CD3CM: Clock Domain 3 Clock Selection Status Bits
<table><tr><th>bit23:20</th><th>Description</th></tr><tr><td>0011</td><td>Sub clock (Do not use this setting)</td></tr></table></td></tr>
<tr><td>604</td><td>CHAPTER 6:Low Power Consumption 5. Registers 5.14.2. STS Clock Selection Register 1 (SYSC1_STSCKSELR1)</td><td>Added the Below:

[bit19:16] CD3CSL: Clock Domain 3 Clock Selection Bits
<table><tr><th>bit19:16</th><th>Description</th></tr><tr><td>0011</td><td>Sub clock (Do not use this setting)</td></tr></table>
[bit15:12] CD2CM: Clock Domain 2 Clock Selection Status Bits
<table><tr><th>bit15:12</th><th>Description</th></tr><tr><td>0011</td><td>Sub clock (Do not use this setting)</td></tr></table></td></tr>
<tr><td>605</td><td>CHAPTER 6:Low Power Consumption 5. Registers 5.14.2. STS Clock Selection Register 1 (SYSC1_STSCKSELR1)</td><td>Added the Below:

[bit11:8] CD2CSL: Clock Domain 2 Clock Selection Bits
<table><tr><th>bit11:18</th><th>Description</th></tr><tr><td>0011</td><td>Sub clock (Do not use this setting)</td></tr></table>
[bit7:4] CD1CM: Clock Domain 1 Clock Selection Status Bits
<table><tr><th>bit7:4</th><th>Description</th></tr><tr><td>0011</td><td>Sub clock (Do not use this setting)</td></tr></table></td></tr>
<tr><td>606</td><td>CHAPTER 6:Low Power Consumption 5. Registers 5.14.2. STS Clock Selection Register 1 (SYSC1_STSCKSELR1)</td><td>Added the Below:

[bit3:0] CD1CSL: Clock Domain 1 Clock Selection Bits
<table><tr><th>bit3:0</th><th>Description</th></tr><tr><td>0011</td><td>Sub clock (Do not use this setting)</td></tr></table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>607
to
608</td><td>CHAPTER 6:Low Power Consumption
5. Registers
5.14.3. STS Clock Selection Register 2 (SYSC1_STSCKSELR2)</td><td>Added the Below:

[bit14:12] TRCCM: TRC Clock Selection Status Bits
<table><tr><th>bit14:12</th><th>Description</th></tr><tr><td>011</td><td>Sub clock (Do not use this setting)</td></tr></table>
[bit10:8] TRCCSL: TRC Clock Selection Bits
<table><tr><th>bit10:8</th><th>Description</th></tr><tr><td>011</td><td>Sub clock (Do not use this setting)</td></tr></table></td></tr>
<tr><td>608
to
609</td><td>CHAPTER 6:Low Power Consumption
5. Registers
5.14.3. STS Clock Selection Register 2 (SYSC1_STSCKSELR2)</td><td>Added the Below:

[bit7:4] CD5CM: Clock Domain 5 Clock Selection Status Bits
<table><tr><th>bit7:4</th><th>Description</th></tr><tr><td>0011</td><td>Sub clock (Do not use this setting)</td></tr></table>
[bit3:0] CD5CSL: Clock Domain 5 Clock Selection Bits
<table><tr><th>bit3:0</th><th>Description</th></tr><tr><td>0011</td><td>Sub clock (Do not use this setting)</td></tr></table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>623</td><td>CHAPTER 6:Low Power Consumption 5. Registers 5.14.7. STS Clock Divider Register 0 (SYSC1_STSCKDIVR0)</td><td>
Error)

[bit20:16] TRCDIV: TRC Clock Divider Setting Bits
These bits indicate the set value of the division ratio of the TRC clock from the source clock (clock domain TRC).
<table>
<tr><th>bit20:16</th><th>Description</th></tr>
<tr><td>0000</td><td>No division</td></tr>
<tr><td>0001</td><td>Divided by 2</td></tr>
<tr><td>0010</td><td>Divided by 3</td></tr>
<tr><td>0011</td><td>Divided by 4</td></tr>
<tr><td>・・・</td><td>・・・</td></tr>
<tr><td>1101</td><td>Divided by 14</td></tr>
<tr><td>1110</td><td>Divided by 15</td></tr>
<tr><td>1111</td><td>Divided by 16</td></tr>
</table>

Correct)

[bit20:16] TRCDIV: TRC Clock Divider Setting Bits
These bits indicate the set value of the division ratio of the TRC clock from the source clock (clock domain TRC).
<table>
<tr><th>bit20:16</th><th>Description</th></tr>
<tr><td>00000</td><td>No division</td></tr>
<tr><td>00001</td><td>Divided by 2</td></tr>
<tr><td>00010</td><td>Divided by 3</td></tr>
<tr><td>00011</td><td>Divided by 4</td></tr>
<tr><td>・・・</td><td>・・・</td></tr>
<tr><td>11101</td><td>Divided by 30</td></tr>
<tr><td>11110</td><td>Divided by 31</td></tr>
<tr><td>11111</td><td>Divided by 32</td></tr>
</table>
</td></tr>
<tr><td>650</td><td>CHAPTER 6:Low Power Consumption 6. Other 6.1. Restrictions on Transitioning to the PSS</td><td>
Added the Below:

2. When using in the following condition, Fast CR clock must be enabled.
- Use PSS mode without PD2 shutdown
- Use peripherals in PD6 at PSS mode shutdown
Fast CR clock is enabled when it sets SYSC0_PSSCKSRER. CROSCEN=1.
</td></tr>
</table>

Page	Section	Change Results
654	CHAPTER 7:Low-voltage Detection 1.1. Features	Error) Internal Low-Voltage Detection This section explains the features of internal low-voltage detection. ■ Operation: Always active. The RAM contents after a power-on reset cannot be guaranteed. 1.2 V Power Supply Low-Voltage Detection The 1.2 V power supply low-voltage detection circuit has two channels. This section explains the features of 1.2 V power supply low-voltage detection. ■ Operation: Always active. The RAM contents after a 1.2 V power supply low-voltage detection reset cannot be guaranteed. Correct) **Internal Low-Voltage Detection** ■ Function, detection voltage: Please see (Product specification). ■ Operation: Always active. The RAM contents after a power-on reset cannot be guaranteed. **Core Power Supply Low-Voltage Detection** The Core power supply low-voltage detection circuit has two channels. ■ Function, detection voltage: Please see (Product specification). ■ Operation: Always active. The RAM contents after a Core power supply low-voltage detection reset cannot be guaranteed.
654	CHAPTER 7:Low-voltage Detection 1.1. Features	Deleted the Below: 3.3/5.0 V Power Supply Low-Voltage Detection This section explains the features of 3.3/5.0 V power supply low-voltage detection

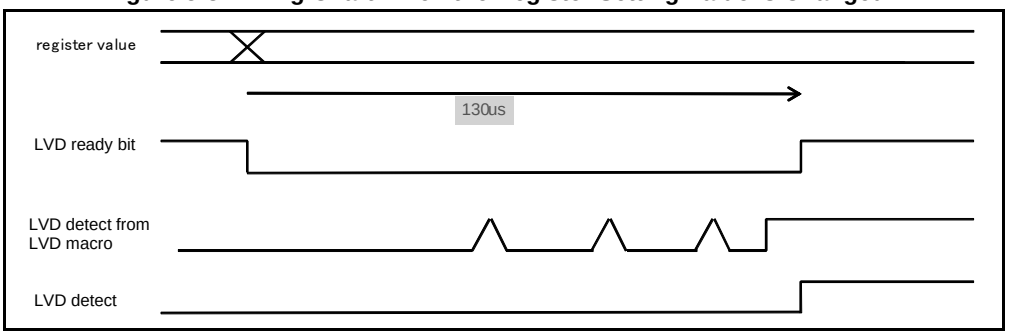
<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>655</td><td>CHAPTER 7:Low-voltage Detection
1.1. Features</td><td>Error)
Table 7-2 LVD Operation Select Bits<table><tr><td>LVDL1S</td><td>1.2 V power supply low-voltage detection operation select bit ch1</td></tr><tr><td>LVDL2S</td><td>1.2 V power supply low-voltage detection operation select bit ch2</td></tr><tr><td>LVDH1S</td><td>3.3/5.0 V power supply low-voltage detection operation select bit ch1</td></tr><tr><td>LVDH2S</td><td>3.3/5.0 V power supply low-voltage detection operation select bit ch2</td></tr></table>
Correct)
Table 7-2 LVD Operation Select Bits<table><tr><td>LVDL1S</td><td>Core power supply low-voltage detection operation select bit ch1</td></tr><tr><td>LVDL2S</td><td>Core power supply low-voltage detection operation select bit ch2</td></tr><tr><td>LVDH1S</td><td>3.3/5.0 V power supply low-voltage detection operation select bit ch1</td></tr><tr><td>LVDH2S</td><td>3.3/5.0 V power supply low-voltage detection operation select bit ch2</td></tr></table></td></tr>
<tr><td>655</td><td>CHAPTER 7:Low-voltage Detection
1. Overview
1.1. Features</td><td>Error)
Table 7-3 LVD Ready Bits<table><tr><td>LVDL1RDY</td><td>1.2 V power supply low-voltage detection ready bit ch1</td></tr><tr><td>LVDL2RDY</td><td>1.2 V power supply low-voltage detection ready bit ch2</td></tr><tr><td>LVDH1RDY</td><td>3.3/5.0 V power supply low-voltage detection ready bit ch1</td></tr><tr><td>LVDH2RDY</td><td>3.3/5.0 V power supply low-voltage detection ready bit ch2</td></tr></table>
Correct)
Table 7-3 LVD Ready Bits<table><tr><td>LVDL1R</td><td>Core power supply low-voltage detection ready bit ch1</td></tr><tr><td>LVDL2R</td><td>Core power supply low-voltage detection ready bit ch2</td></tr><tr><td>LVDH1R</td><td>3.3/5.0 V power supply low-voltage detection ready bit ch1</td></tr><tr><td>LVDH2R</td><td>3.3/5.0 V power supply low-voltage detection ready bit ch2</td></tr></table></td></tr>
</table>

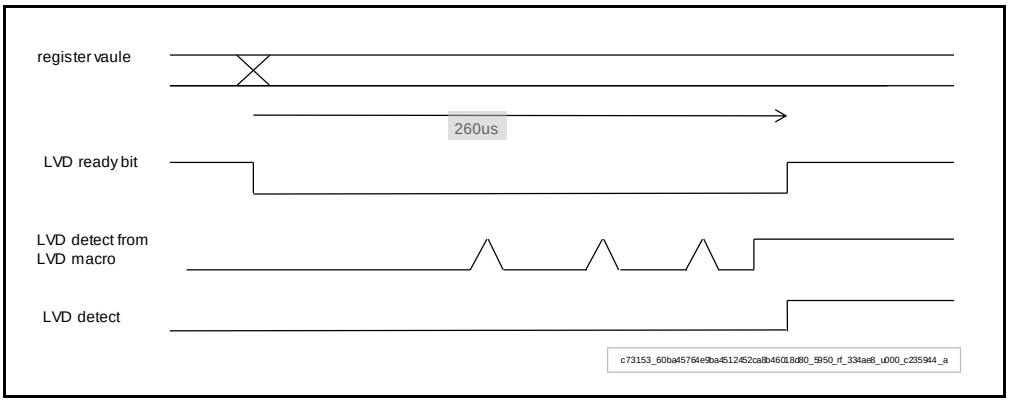
<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>655</td><td>CHAPTER 7:Low-voltage Detection
1. Overview
1.1. Features</td><td>Error)
Table 7-4 LVD Reset Source Bits<table><tr><td>LVDL1E</td><td>1.2 V power supply low-voltage detection operation enable bit ch1</td></tr><tr><td>LVDL2E</td><td>1.2 V power supply low-voltage detection operation enable bit ch2</td></tr><tr><td>LVDH1E</td><td>3.3/5.0 V power supply low-voltage detection operation enable bit ch1</td></tr><tr><td>LVDL2E</td><td>3.3/5.0 V power supply low-voltage detection operation enable bit ch2</td></tr></table>Correct)
Table 7-4 LVD Reset Source Bits<table><tr><td>LVDL1E</td><td>Core power supply low-voltage detection operation enable bit ch1</td></tr><tr><td>LVDL2E</td><td>Core power supply low-voltage detection operation enable bit ch2</td></tr><tr><td>LVDH1E</td><td>3.3/5.0 V power supply low-voltage detection operation enable bit ch1</td></tr><tr><td>LVDH2E</td><td>3.3/5.0 V power supply low-voltage detection operation enable bit ch2</td></tr></table></td></tr>
<tr><td>655</td><td>CHAPTER 7:Low-voltage Detection
1. Overview
1.1. Features</td><td>Error)
Notes:
■ In the PF-Cluster series, functions are mapped as follows:
■ 1.2 V power supply low-voltage detection ch1: 1.2 V internal regulator voltage monitoring
■ 1.2 V power supply low-voltage detection ch2: 1.2 V external power supply monitoring
■ In the PF-Cluster series, a low-voltage detection interrupt corresponds to an NMI.

Correct)
Notes:
− In the Traveo Family Hardware Manual Platform Part series, functions are mapped as follows:
Core power supply low-voltage detection ch1: 1.2V internal regulator voltage monitoring
Core power supply low-voltage detection ch2: 1.2V external power supply monitoring
− In the Traveo Family Hardware Manual Platform Part series, a low-voltage detection interrupt corresponds to an NMI.</td></tr>
<tr><td>657</td><td>CHAPTER 7:Low-voltage Detection
2. onfiguration and Block Diagrams</td><td>Error)
Figure 7.2-3 1.2 V Power Supply Low-Voltage Detection, 3.3/5.0 V Power Supply Low-Voltage Detection

Correct)
Figure 2-3 Core Power Supply Low-Voltage Detection, 3.3/5.0 V Power Supply Low-Voltage Detection</td></tr>
</table>

Page	Section	Change Results
658	CHAPTER 7:Low-voltage Detection 3.Explanation of Operation	Error) 1.2 V Power Supply Low-Voltage Detection Circuit, 3.3/5.0 V Power Supply Low-Voltage Detection Circuit The 3.3/5.0 V power supply low-voltage detection circuit operates in the same way as the 1.2 V power supply low-voltage detection circuit. Correct) Core Power Supply Low-Voltage Detection Circuit, 3.3/5.0 V Power Supply Low-Voltage Detection Circuit The 3.3/5.0 V power supply low-voltage detection circuit operates in the same way as the core power supply low-voltage detection circuit.
658	CHAPTER 7:Low-voltage Detection 3. Explanation of Operation	Error) After power-on or a change in its settings, the low-voltage detection circuit is given a stabilization wait time (about 130 µs). Correct) After power-on or a change in its settings, the low-voltage detection circuit is given a stabilization wait time (about 260 µs).

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>664</td><td>CHAPTER 7:Low-voltage Detection
5. Operation Examples</td><td>Error)
Figure 5-3 Timing Chart When the Register Setting Value Is Changed


Correct)
Figure 5-3 Timing Chart When the Register Setting Value Is Changed

</td></tr>
<tr><td>664</td><td>CHAPTER 7:Low-voltage Detection
5. Operation Examples</td><td>Error)
<table>
<tr><td>LVDL1E</td><td>low voltage detect enable for 1.2V power supply, ch1</td></tr>
<tr><td>LVDL2E</td><td>low voltage detect enable for 1.2V power supply, ch2</td></tr>
<tr><td>LVDH1E :</td><td>low voltage detect enable for 3.3V or 5.5V power supply, ch1</td></tr>
<tr><td>LVDH2E :</td><td>low voltage detect enable for 3.3V or 5.5V power supply, ch2</td></tr>
</table>
Correct)
<table>
<tr><td>LVDL1E</td><td>Core power supply low-voltage detection operation enable bit ch1</td></tr>
<tr><td>LVDL2E</td><td>Core power supply low-voltage detection operation enable bit ch2</td></tr>
<tr><td>LVDH1E :</td><td>3.3V/5.0V power supply low-voltage detection operation enable bit ch1</td></tr>
<tr><td>LVDH2E :</td><td>3.3V/5.0V power supply low-voltage detection operation enable bit ch2</td></tr>
</table></td></tr>
</table>

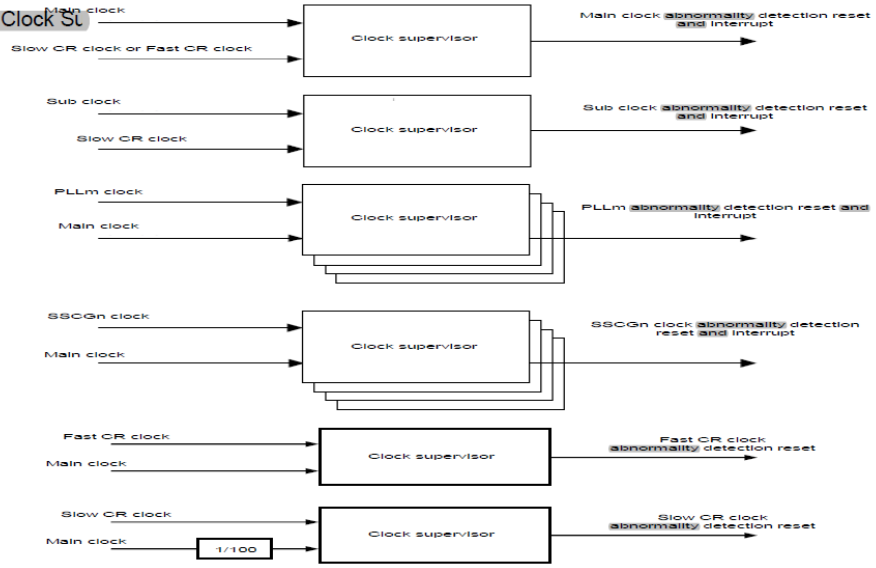
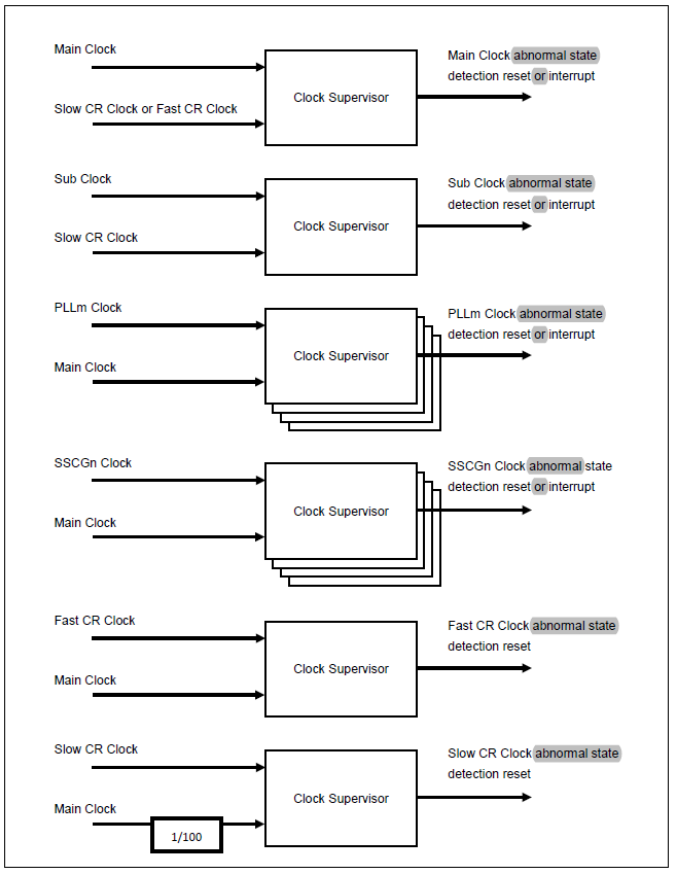
Page	Section	Change Results
665	CHAPTER 7:Low-voltage Detection 6. Precautions for Using This Device	Error) Masking of a Detection Result ■Following any low-voltage setting change, the detection result is masked for about 130 us. Correct) ■Following any low-voltage setting change, the detection result is masked for about 260 us.

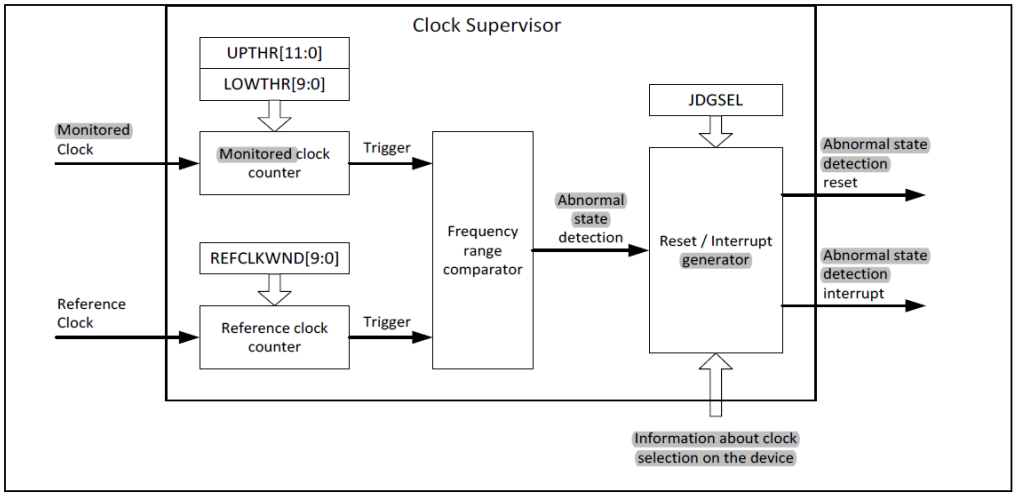
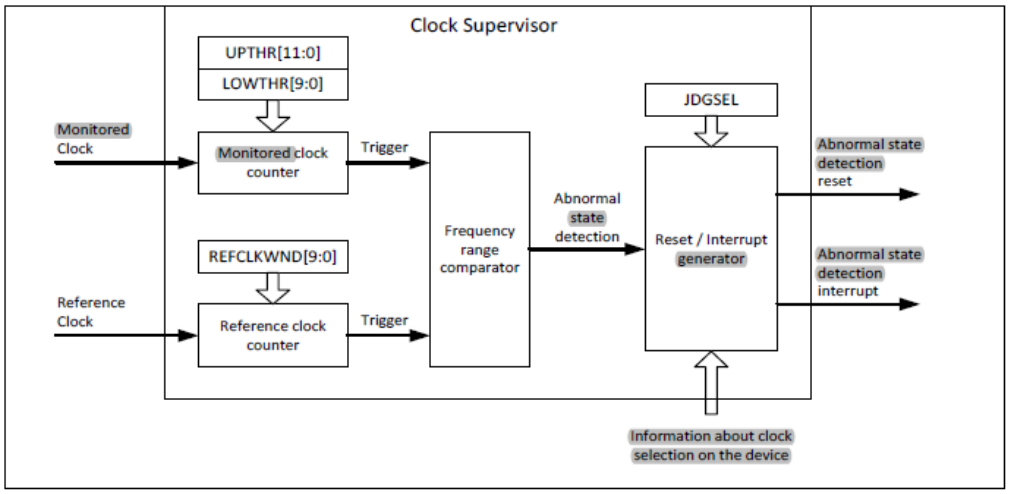
Page	Section	Change Results
-	Deleted the section.	8.5.7 PLL1 Clock Supervisor Setting Register 0 (SYSC_CSVPLL1CFGR0) 8.5.8 PLL1 Clock Supervisor Setting Register 1 (SYSC_CSVPLL1CFGR1) 8.5.9 PLL2 Clock Supervisor Setting Register 0 (SYSC_CSVPLL2CFGR0) 8.5.10 PLL2 Clock Supervisor Setting Register 1 (SYSC_CSVPLL2CFGR1) 8.5.11 PLL3 Clock Supervisor Setting Register 0 (SYSC_CSVPLL3CFGR0) 8.5.12 PLL3 Clock Supervisor Setting Register 1 (SYSC_CSVPLL3CFGR1) 8.5.15 SSCG PLL1 Clock Supervisor Setting Register 0 (SYSC_CSVSP1CFGR0) 8.5.16 SSCG PLL1 Clock Supervisor Setting Register 1 (SYSC_CSVSP1CFGR1) 8.5.17 SSCG PLL2 Clock Supervisor Setting Register 0 (SYSC_CSVSP2CFGR0) 8.5.18 SSCG PLL2 Clock Supervisor Setting Register 1 (SYSC_CSVSP2CFGR1) 8.5.19 SSCG PLL3 Clock Supervisor Setting Register 0 (SYSC_CSVSP3CFGR0) 8.5.20 SSCG PLL3 Clock Supervisor Setting Register 1 (SYSC_CSVSP3CFGR1)
668	CHAPTER 8:Clock Supervisor 1. Overview	Error) Two types of clock supervisors are equipped with this device. The clock supervisor to be used by the clock to be monitored is different. Correct) This device is equipped with several instances of two different types of clock supervisors. The used type of clock supervisor depends on the clock signal which is to be monitored.
668		Error) 1. Detection type of stop or frequency range abnormality of the clock. It monitors for stop or frequency range abnormality of the monitoring clock. Parameters for reference clock frequency, upper limit of the monitoring clock frequency and lower limit of the monitoring clock frequency are set to the monitoring clock counter and reference clock counter. Frequency range comparator can detect abnormalities from each counter output. Either reset or interrupt is generated when the comparator detect abnormalities. Correct) 1. Detector for stopped clock or abnormal frequency of the monitored clock. This type of clock supervisor detects two possible errors in clock operation: a stopped clock or a clock operating in an abnormal frequency range. There are clock counters for both the monitored clock and the reference clock. Parameters for each counter define the frequency of the reference clock as well as the upper and lower limit for the frequency of the monitored clock. In case the dedicated frequency range comparator detects a stopped clock or a clock outside the frequency range, an abnormal state is signalled. Then, it depends on register settings and on the way, the monitored clock is used on the device, whether a reset or an interrupt is generated.

Page	Section	Change Results
668	CHAPTER 8:Clock Supervisor 1. Overview	Error) 2. Detection type of stop of the clock. It monitors for stop of the monitoring clock. When the rising edge of the monitoring clock is not detected, the clock supervisor detects the abnormalities of oscillator. Reset is generated when the comparator detect abnormalities. Correct) 2. Detector for stopped clock only. This type of clock supervisor detects a stopped clock operation for the monitored clock. When the rising edge of the monitored clock does not occur, the clock supervisor signals an abnormal state. In this case, a reset is generated.

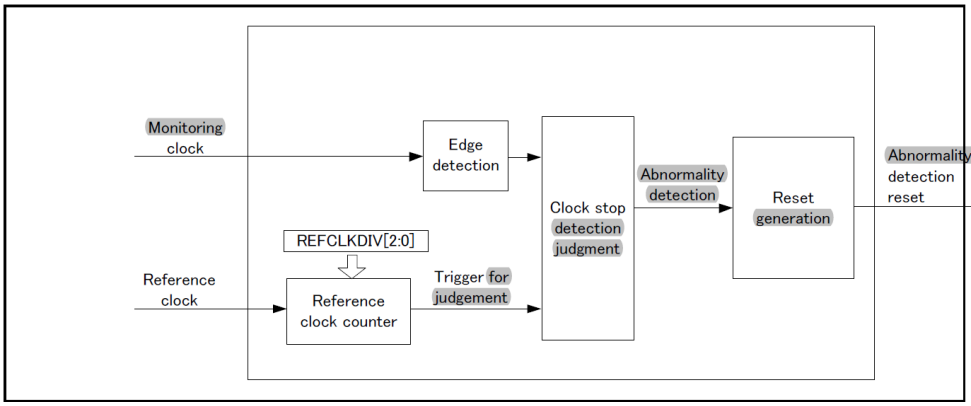
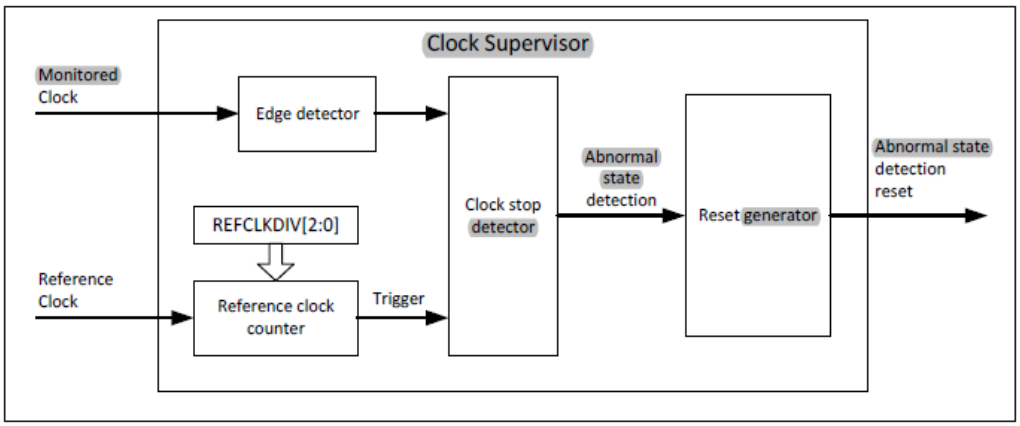
<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>668</td><td>CHAPTER 8:Clock Supervisor 1. Overview</td><td>
Error)
Table 8.1-1 Overview of Each Clock Supervisor Instances
<table>
<tr><th>Monitoring Clock</th><th>Reference Clock</th><th>Clock Supervisor Feature</th></tr>
<tr><td>Main clock</td><td>Selection from Fast CR clock or Slow CR clock</td><td rowspan="9">Detection of stop or frequency range abnormality of the clock.</td></tr>
<tr><td>Sub clock</td><td>Slow CR clock</td></tr>
<tr><td>PLL0 clock</td><td>Main clock</td></tr>
<tr><td>PLL1 clock</td><td>Main clock</td></tr>
<tr><td>PLL2 clock</td><td>Main clock</td></tr>
<tr><td>PLL3 clock</td><td>Main clock</td></tr>
<tr><td>SSCG0 clock</td><td>Main clock</td></tr>
<tr><td>SSCG1 clock</td><td>Main clock</td></tr>
<tr><td>SSCG2clock</td><td>Main clock</td></tr>
<tr><td>SSCG3 clock</td><td>Main clock</td><td rowspan="2">Detection of stop of the clock.</td></tr>
<tr><td>Fast CR clock</td><td>Main clock</td></tr>
<tr><td>Slow CR clock</td><td>Main clock
(Use the clock divided by 100 of the main clock)</td></tr>
</table>

Correct)
Table 1-1 Overview of Clock Supervisor Instances
<table>
<tr><th>Monitored Clock</th><th>Reference Clock</th><th>Clock Supervisor Type</th></tr>
<tr><td>Main clock</td><td>Selection from Fast CR clock or Slow CR clock</td><td rowspan="10">Detector for stopped clock or abnormal frequency of the monitored clock</td></tr>
<tr><td>Sub clock</td><td>Slow CR clock</td></tr>
<tr><td>PLL0 clock</td><td>Main clock</td></tr>
<tr><td>PLL1 clock</td><td>Main clock</td></tr>
<tr><td>PLL2 clock</td><td>Main clock</td></tr>
<tr><td>PLL3 clock</td><td>Main clock</td></tr>
<tr><td>SSCG0 clock</td><td>Main clock</td></tr>
<tr><td>SSCG1 clock</td><td>Main clock</td></tr>
<tr><td>SSCG2 clock</td><td>Main clock</td></tr>
<tr><td>SSCG3 clock</td><td>Main clock</td></tr>
<tr><td>Fast CR clock</td><td>Main clock</td><td rowspan="2">Detector for stopped clock only.</td></tr>
<tr><td>Slow CR clock</td><td>Main clock divided by 100</td></tr>
</table>
</td></tr>
</table>

Page	Section	Change Results
669	CHAPTER 8:Clock Supervisor 2.Configuration	Error) Figure 8.2-1 Clock Su  Correct) **Figure 2-1: Clock Supervisor Configuration** 

Page	Section	Change Results
671	CHAPTER 8:Clock Supervisor 2.1.Detector for Stopped Clock or Abnormal Frequency of the Monitored Clock	Error) **Detection Type of Stop or Frequency Range Abnormality of the Clock** Figure 8.2-2 Block Diagram of Clock Supervisor Detection Type of Stop or Frequency Range Abnormality of the Clock  Correct) **2.1.Detector for Stopped Clock or Abnormal Frequency of the Monitored Clock** Figure **2-2**: Block Diagram of Clock Supervisor for stopped clock or abnormal frequency of the monitored clock 
671	CHAPTER 8:Clock Supervisor 2.1.Detector for Stopped Clock or Abnormal Frequency of	Error) **Monitoring Clock Counter** There are 2 down counters for counting with the monitoring clock. Correct) **Monitored Clock Counter** There are 2 down counters for counting with the monitored clock.

Page	Section	Change Results
671	the Monitored Clock	Error) **Frequency Range Comparator** The frequency range comparator compares frequency ranges according to triggers from the monitoring clock counters and reference clock counter. The lower-limit threshold value bits (LOWTHR[9:0]) and upper-limit threshold value bits (UPTHR[11:0]) of the monitoring clock counters set a frequency range. Correct) **Frequency Range Comparator** The frequency range comparator compares frequency ranges according to triggers from the monitored clock counters and reference clock counter. The lower-limit threshold value bits (LOWTHR[9:0]) and upper-limit threshold value bits (UPTHR[11:0]) of the monitored clock counters set a frequency range.
671		Error) **Reset/Interrupt Generator** Upon detecting an abnormality of a clock, the reset/interrupt generator determines whether to generate a reset or interrupt according to the use status of the monitoring clock. Correct) **Reset/Interrupt Generator** Upon detecting an abnormality of a clock, the reset/interrupt generator determines whether to generate a reset or interrupt according to the use status of the monitored clock.
672	CHAPTER 8:Clock Supervisor 2.2.Detector for Stopped Clock Only	Error) **Detection Type of Stop of the Clock** Correct) **2.2. Detector for Stopped Clock Only**

Page	Section	Change Results
672	CHAPTER 8:Clock Supervisor 2.2.Detector for Stopped Clock Only	Error) Figure 8.2-3 Block Diagram of Detection Type of Stop of the Clock  Correct) Figure 2-3: Block Diagram of Detector for Stopped Clock Only 
672		Error) **Edge Detection** The edge detection detects the rising edge of the monitoring clock. Correct) **Edge detection** The edge detection detects the rising edge of the monitored clock.

Page	Section	Change Results
674	CHAPTER 8:Clock Supervisor 3.Explanation of Operation	Error) **PLL Clock Supervisor** ■ PLLm clock supervisor setting register 0 (SYSC_CSVPLLCFGR0) and PLLm clock supervisor setting register 1 (SYSC_CSVPLLCFGR1) are used for settings. Correct) **PLL Clock Supervisor** ■ PLLm clock supervisor setting register 0 (SYSC_CSVPLLmCFGR0) and PLLm clock supervisor setting register 1 (SYSC_CSVPLLmCFGR1) are used for settings.
674		Error) **SSCG PLL Clock Supervisor** ■ SSCG PLLn clock supervisor setting register 0 (SYSC_CSVSPCFGR0) and SSCG PLLn clock supervisor setting register 1 (SYSC_CSVSPCFGR1) are used for settings. Correct) **SSCG PLL Clock Supervisor** ■ SSCG PLLn clock supervisor setting register 0 (SYSC_CSVSPnCFGR0) and SSCG PLLn clock supervisor setting register 1 (SYSC_CSVSPnCFGR1) are used for settings.
674		Error) **SSCG PLL Clock Supervisor** ■ Upon detecting an abnormality of the SSCG PLL clock and generating a reset, the SSCG PLL clock supervisor can confirm it with the SSCG PLL clock supervisor reset detection bit (CSVSPR) in the user reset factor register (SYSC_RSTCAUSEUR). Correct) **SSCG PLL Clock Supervisor** ■ Upon detecting an abnormality of the SSCG PLL clock and generating a reset, the SSCG PLL clock supervisor can confirm it with the SSCG PLL clock supervisor reset detection bit (CSVSR) in the user reset factor register (SYSC_RSTCAUSEUR).
676	CHAPTER 8:Clock Supervisor 4.Setting Procedure Example	Error) ■ The clock supervisor does not operate until monitoring clock and reference clock oscillation has stabilized. Correct) ■The clock supervisor does not operate until monitored clock and reference clock oscillation have stabilized.
676	CHAPTER 8:Clock Supervisor 4.Setting Procedure Example	Error) ■ The clock supervisor can detect artificially the clock abnormality detection state by gating the monitoring clock for self-testing. Correct) ■The clock supervisor can detect artificially the clock abnormality detection state by gating the monitored clock for self-testing.

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>677</td><td>CHAPTER 8:Clock Supervisor 5.Registers</td><td>Error)
Table 8.5-1 List of Clock Supervisor Registers (CSV Configuration Registers)
<table>
<tr><th>Abbreviated Register Name</th><th>Register Name</th><th>Reference</th></tr>
<tr><td>SYSC_CSVMOCFGR0</td><td>Main clock supervisor setting register 0</td><td>8.5.1</td></tr>
<tr><td>SYSC_CSVMOCFGR1</td><td>Main clock supervisor setting register 1</td><td>8.5.2</td></tr>
<tr><td>SYSC_CSVSOCFGR0</td><td>Sub clock supervisor setting register 0</td><td>8.5.3</td></tr>
<tr><td>SYSC_CSVSOCFGR1</td><td>Sub clock supervisor setting register 1</td><td>8.5.4</td></tr>
<tr><td>SYSC_CSVPLL0CFGR0</td><td>PLL0 clock supervisor setting register 0</td><td>8.5.5</td></tr>
<tr><td>SYSC_CSVPLL0CFGR1</td><td>PLL0 clock supervisor setting register 1</td><td>8.5.6</td></tr>
<tr><td>SYSC_CSVPLL1CFGR0</td><td>PLL1 clock supervisor setting register 0</td><td>8.5.7</td></tr>
<tr><td>SYSC_CSVPLL1CFGR1</td><td>PLL1 clock supervisor setting register 1</td><td>8.5.8</td></tr>
<tr><td>SYSC_CSVPLL2CFGR0</td><td>PLL2 clock supervisor setting register 0</td><td>8.5.9</td></tr>
<tr><td>SYSC_CSVPLL2CFGR1</td><td>PLL2 clock supervisor setting register 1</td><td>8.5.10</td></tr>
<tr><td>SYSC_CSVPLL3CFGR0</td><td>PLL3 clock supervisor setting register 0</td><td>8.5.11</td></tr>
<tr><td>SYSC_CSVPLL3CFGR1</td><td>PLL3 clock supervisor setting register 1</td><td>8.5.12</td></tr>
<tr><td>SYSC_CSVSP0CFGR0</td><td>SSCG PLL0 clock supervisor setting register 0</td><td>8.5.13</td></tr>
<tr><td>SYSC_CSVSP0CFGR1</td><td>SSCG PLL0 clock supervisor setting register 1</td><td>8.5.14</td></tr>
<tr><td>SYSC_CSVSP1CFGR0</td><td>SSCG PLL1 clock supervisor setting register 0</td><td>8.5.15</td></tr>
<tr><td>SYSC_CSVSP1CFGR1</td><td>SSCG PLL1 clock supervisor setting register 1</td><td>8.5.16</td></tr>
</table>
<table>
<tr><th>Abbreviated Register Name</th><th>Register Name</th><th>Reference</th></tr>
<tr><td>SYSC_CSVSP2CFGR0</td><td>SSCG PLL2 clock supervisor setting register 0</td><td>8.5.17</td></tr>
<tr><td>SYSC_CSVSP2CFGR1</td><td>SSCG PLL2 clock supervisor setting register 1</td><td>8.5.18</td></tr>
<tr><td>SYSC_CSVSP3CFGR0</td><td>SSCG PLL3 clock supervisor setting register 0</td><td>8.5.19</td></tr>
<tr><td>SYSC_CSVSP3CFGR1</td><td>SSCG PLL3 clock supervisor setting register 1</td><td>8.5.20</td></tr>
<tr><td>SYSC_CSVFCRCFGR</td><td>Fast CR clock supervisor setting register</td><td>8.5.21</td></tr>
<tr><td>SYSC_CSVSCRCFGR</td><td>Slow CR clock supervisor setting register</td><td>8.5.22</td></tr>
<tr><td>SYSC_CSVOUTER</td><td>Clock supervisor output enable register</td><td>8.5.23</td></tr>
<tr><td>SYSC_CSVTESTR</td><td>Clock supervisor test register</td><td>8.5.24</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>677</td><td>CHAPTER 8:Clock Supervisor 5.Registers</td><td>Correct)
Table 5-1 List of Clock Supervisor Registers (CSV Configuration Registers)
<table>
<tr><th>Abbreviated Register Name</th><th>Register Name</th><th>Reference</th></tr>
<tr><td>SYSC_CSVMOCFGR0</td><td>Main clock supervisor setting register 0</td><td>5.1</td></tr>
<tr><td>SYSC_CSVMOCFGR1</td><td>Main clock supervisor setting register 1</td><td>5.2</td></tr>
<tr><td>SYSC_CSVSOCFGR0</td><td>Sub clock supervisor setting register 0</td><td>5.3</td></tr>
<tr><td>SYSC_CSVSOCFGR1</td><td>Sub clock supervisor setting register 1</td><td>5.4</td></tr>
<tr><td>SYSC_CSVPLL0CFGR0</td><td>PLL0 clock supervisor setting register 0</td><td>5.5</td></tr>
<tr><td>SYSC_CSVPLL0CFGR1</td><td>PLL0 clock supervisor setting register 1</td><td>5.6</td></tr>
<tr><td>SYSC_CSVPLL1CFGR0</td><td>PLL1 clock supervisor setting register 0</td><td>5.5</td></tr>
<tr><td>SYSC_CSVPLL1CFGR1</td><td>PLL1 clock supervisor setting register 1</td><td>5.6</td></tr>
<tr><td>SYSC_CSVPLL2CFGR0</td><td>PLL2 clock supervisor setting register 0</td><td>5.5</td></tr>
<tr><td>SYSC_CSVPLL2CFGR1</td><td>PLL2 clock supervisor setting register 1</td><td>5.6</td></tr>
<tr><td>SYSC_CSVPLL3CFGR0</td><td>PLL3 clock supervisor setting register 0</td><td>5.5</td></tr>
<tr><td>SYSC_CSVPLL3CFGR1</td><td>PLL3 clock supervisor setting register 1</td><td>5.6</td></tr>
<tr><td>SYSC_CSVSP0CFGR0</td><td>SSCG PLL0 clock supervisor setting register 0</td><td>5.7</td></tr>
<tr><td>SYSC_CSVSP0CFGR1</td><td>SSCG PLL0 clock supervisor setting register 1</td><td>5.8</td></tr>
<tr><td>SYSC_CSVSP1CFGR0</td><td>SSCG PLL1 clock supervisor setting register 0</td><td>5.7</td></tr>
<tr><td>SYSC_CSVSP1CFGR1</td><td>SSCG PLL1 clock supervisor setting register 1</td><td>5.8</td></tr>
<tr><td>SYSC_CSVSP2CFGR0</td><td>SSCG PLL2 clock supervisor setting register 0</td><td>5.7</td></tr>
<tr><td>SYSC_CSVSP2CFGR1</td><td>SSCG PLL2 clock supervisor setting register 1</td><td>5.8</td></tr>
<tr><td>SYSC_CSVSP3CFGR0</td><td>SSCG PLL3 clock supervisor setting register 0</td><td>5.7</td></tr>
<tr><td>SYSC_CSVSP3CFGR1</td><td>SSCG PLL3 clock supervisor setting register 1</td><td>5.8</td></tr>
<tr><td>SYSC_CSVFCRCFGR</td><td>Fast CR clock supervisor setting register</td><td>5.9</td></tr>
<tr><td>SYSC_CSVSCRCFGR</td><td>Slow CR clock supervisor setting register</td><td>5.10</td></tr>
<tr><td>SYSC_CSVOUTER</td><td>Clock supervisor output enable register</td><td>5.11</td></tr>
<tr><td>SYSC_CSVTESTR</td><td>Clock supervisor test register</td><td>5.12</td></tr>
</table></td></tr>
<tr><td>678 to 679</td><td>CHAPTER 8:Clock Supervisor 5.Registers</td><td>Error)
Table 8.5-2 Clock Supervisor Register Map (B/B-PG)
<table>
<tr><th>Offset</th><th>Register Name/Initial Value</th></tr>
<tr><td>0x0000_0058～
0x0000_005C</td><td>Reserved
00000000_00000000_00000000_00000000</td></tr>
</table>
Correct)
Table 5-2 Clock Supervisor Register Map (B/B)
<table>
<tr><th>Offset</th><th>Register Name/Initial Value</th></tr>
<tr><td>0x0000_0058 to
0x0000_005C</td><td>Reserved
00000000_00000000_00000000_00000000</td></tr>
</table></td></tr>
<tr><td>680</td><td>CHAPTER 8:Clock Supervisor 5.1.Main Clock Supervisor Setting Register 0 (SYSC_CSVMOCFGR0)</td><td>Error)
[bit27:16] UPTHR[11:0]: Upper-Limit Threshold Value Bits
These bits set the clock upper-limit threshold value for comparison with the monitoring clock count value. If the monitoring clock counter value exceeds the upper-limit threshold value, the state is judged as abnormal.

Correct)
[bit27:16] UPTHR[11:0]: Upper-limit threshold value bits
These bits set the clock upper-limit threshold value for comparison with the monitored clock count value. If the monitored clock counter value exceeds the upper-limit threshold value, the state is judged as abnormal.</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>680
to
681</td><td>CHAPTER 8:Clock Supervisor 5.1.Main Clock Supervisor Setting Register 0 (SYSC_CSVMOCFGR0)</td><td>Error)
[bit9:0] LOWTHR[9:0]: Lower-Limit Threshold Value Bits
These bits set the clock lower-limit threshold value for comparison with the monitoring clock count value. If the monitoring clock counter value falls below the lower-limit threshold value, the state is judged as abnormal.

Correct)
[bit9:0] LOWTHR[9:0]: Lower-limit threshold value bits
These bits set the clock lower-limit threshold value for comparison with the monitored clock count value. If the monitored clock counter value falls below the lower-limit threshold value, the state is judged as abnormal.</td></tr>
<tr><td>681</td><td></td><td>Error)
Notes:
For the lower-limit threshold value bits (LOWTHR), set a value larger than that obtained from dividing the reference clock cycle by the monitoring clock cycle.

Correct)
Notes:
− For the lower-limit threshold value bits (LOWTHR), set a value larger than that obtained from dividing the reference clock cycle by the monitored clock cycle.</td></tr>
<tr><td>683</td><td>CHAPTER 8:Clock Supervisor 5.2.Main Clock Supervisor Setting Register 1 (SYSC_CSVMOCFGR1)</td><td>Error)
[bit16] JDGSEL: Judgment Selection Bit
Reset generation condition for when an abnormality is detected can be selected with this bit.
An interrupt can be generated when reset generation condition is not correspondence.
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Monitoring clock is selected as the clock for clock domain 0 or MCUC clock domain or software watchdog timer.</td></tr>
<tr><td>1</td><td>Monitoring clock is selected as the clock for clock domain 0 or MCUC clock domain.</td></tr>
</table>
Correct)
[bit16] JDGSEL: Judgment selection bit
This bit selects an additional condition under which a detected abnormal state (by CSV) leads to reset generation. Without the signalling of an abnormal state by the CSV, neither reset nor interrupt are generated. If the signal of an abnormal state occurs, the reaction depends on the condition which is selected by JDGSEL. If the condition as defined by the bit value description is true, the abnormal state leads to a reset, if it is false, an interrupt will be generated.
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Main clock is selected for clock domain 0 or for MCUC clock domain or for software watchdog timer.</td></tr>
<tr><td>1</td><td>Main clock is selected for clock domain 0 or for MCUC clock domain.</td></tr>
</table>
− Example: When JDGSEL = 1 and main clock is neither selected for clock domain 0 nor for MCUC, the selected condition is false and an abnormal state leads to interrupt generation.</td></tr>
</table>

Page	Section	Change Results
684	CHAPTER 8:Clock Supervisor 5.3.Sub Clock Supervisor Setting Register 0 (SYSC_CSVSOCFGR0)	Error) **[bit27:16] UPTHR[11:0]: Upper-Limit Threshold Value Bits** These bits set the clock upper-limit threshold value for comparison with the monitoring clock count value. If the monitoring clock counter value exceeds the upper-limit threshold value, the state is judged as abnormal. Correct) **[bit27:16] UPTHR[11:0]: Upper-limit threshold value bits** These bits set the clock upper-limit threshold value for comparison with the monitored clock count value. If the monitored clock counter value exceeds the upper-limit threshold value, the state is judged as abnormal.
684 to 685	CHAPTER 8:Clock Supervisor 5.3.Sub Clock Supervisor Setting Register 0 (SYSC_CSVSOCFGR0)	Error) **[bit9:0] LOWTHR[9:0]: Lower-Limit Threshold Value Bits** These bits set the clock lower-limit threshold value for comparison with the monitoring clock count value. If the monitoring clock counter value falls below the lower-limit threshold value, the state is judged as abnormal. Correct) **[bit9:0] LOWTHR[9:0]: Lower-limit threshold value bits** These bits set the clock lower-limit threshold value for comparison with the monitored clock count value. If the monitored clock counter value falls below the lower-limit threshold value, the state is judged as abnormal.
685	CHAPTER 8:Clock Supervisor 5.3.Sub Clock Supervisor Setting Register 0 (SYSC_CSVSOCFGR0)	Error) **Notes:** ■ For the lower-limit threshold value bits (LOWTHR), set a value larger than that obtained from dividing the reference clock cycle by the monitoring clock cycle. Correct) ***Notes:*** – *For the lower-limit threshold value bits (LOWTHR), set a value larger than that obtained from dividing the reference clock cycle by the monitored clock cycle.*

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>686
to
687</td><td>CHAPTER 8:Clock Supervisor 5.4.Sub Clock Supervisor Setting Register 1 (SYSC_CSVSOCFGR1)</td><td>Error)
[bit16] JDGSEL: Judgment Selection Bit
Reset generation condition for when an abnormality is detected can be selected with this bit.
An interrupt can be generated when reset generation condition is not correspondence.
<table><tr><th>Bit</th><th>Description</th></tr><tr><td>0</td><td>Monitoring clock is selected as the clock for clock domain 0 or MCUC clock domain or software watchdog timer.</td></tr><tr><td>1</td><td>Monitoring clock is selected as the clock for clock domain 0 or MCUC clock domain.</td></tr></table>
Correct)
[bit16] JDGSEL: Judgment selection bit
This bit selects an additional condition under which a detected abnormal state (by CSV) leads to reset generation. Without the signalling of an abnormal state by the CSV, neither reset nor interrupt are generated. If the signal of an abnormal state occurs, the reaction depends on the condition which is selected by JDGSEL. If the condition as defined by the bit value description is true, the abnormal state leads to a reset, if it is false, an interrupt will be generated.
<table><tr><th>Bit</th><th>Description</th></tr><tr><td>0</td><td>Sub clock is selected for clock domain 0 or for MCUC clock domain or for software watchdog timer.</td></tr><tr><td>1</td><td>Sub clock is selected for clock domain 0 or for MCUC clock domain.</td></tr></table>
– Example: When JDGSEL = 1 and sub clock is neither selected for clock domain 0 nor for MCUC, the selected condition is false and an abnormal state leads to interrupt generation.</td></tr>
<tr><td>688</td><td>CHAPTER 8:Clock Supervisor 5.5.PLLm Clock Supervisor Setting Register 0 (SYSC_CSVPLLmCFGR0)</td><td>Error)
8.5.5 PLL0 Clock Supervisor Setting Register 0 (SYSC_CSVPLL0CFGR0)
PLL0 clock supervisor setting register 0 (SYSC_CSVPLL0CFGR0) sets the upper-limit threshold value and lower-limit threshold value of a frequency range.

Correct)
5.5 PLLm Clock Supervisor Setting Register 0 (SYSC_CSVPLLmCFGR0)
PLLm clock supervisor setting register 0 (SYSC_CSVPLLmCFGR0) sets the upper-limit threshold value and lower-limit threshold value of a frequency range.</td></tr>
<tr><td>688</td><td></td><td>Error)
[bit27:16] UPTHR[11:0]: Upper-Limit Threshold Value Bits
These bits set the clock upper-limit threshold value for comparison with the monitoring clock count value. If the monitoring clock counter value exceeds the upper-limit threshold value, the state is judged as abnormal.

Correct)
[bit27:16] UPTHR[11:0]: Upper-limit threshold value bits
These bits set the clock upper-limit threshold value for comparison with the monitored clock count value. If the monitored clock counter value exceeds the upper-limit threshold value, the state is judged as abnormal.</td></tr>
</table>

Page	Section	Change Results
688 to 689	CHAPTER 8:Clock Supervisor 5.5.PLLm Clock Supervisor Setting Register 0 (SYSC_CSVPLLmCFGR0)	Error) **[bit9:0] LOWTHR[9:0]: Lower-Limit Threshold Value Bits** These bits set the clock lower-limit threshold value for comparison with the monitoring clock count value. If the monitoring clock counter value falls below the lower-limit threshold value, the state is judged as abnormal. Correct) **[bit9:0] LOWTHR[9:0]: Lower-limit threshold value bits** These bits set the clock lower-limit threshold value for comparison with the monitored clock count value. If the monitored clock counter value falls below the lower-limit threshold value, the state is judged as abnormal.
689		Error) **Notes:** ■ For the lower-limit threshold value bits (LOWTHR), set a value larger than that obtained from dividing the reference clock cycle by the monitoring clock cycle. Correct) − *For the lower-limit threshold value bits (LOWTHR), set a value larger than that obtained from dividing the reference clock cycle by the monitored clock cycle.*
690	CHAPTER 8:Clock Supervisor 5.6.PLLm Clock Supervisor Setting Register 1 (SYSC_CSVPLLmCFGR1)	Error) **8.5.6 PLL0 Clock Supervisor Setting Register 1 (SYSC_CSVPLL0CFGR1)** PLL0 clock supervisor setting register 1 (SYSC_CSVPLL0CFGR1) sets a reference clock count period. Correct) **5.6.PLLm Clock Supervisor Setting Register 1 (SYSC_CSVPLLmCFGR1)** PLLm clock supervisor setting register 1 (SYSC_CSVPLLmCFGR1) sets a reference clock count period.

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>690 to 691</td><td>CHAPTER 8:Clock Supervisor 5.6.PLLm Clock Supervisor Setting Register 1 (SYSC_CSVPLLmCFGR1)</td><td>Error)
[bit16] JDGSEL: Judgment Selection Bit
Reset generation condition for when an abnormality is detected can be selected with this bit.
An interrupt can be generated when reset generation condition is not correspondence.
<table><tr><th>Bit</th><th>Description</th></tr><tr><td>0</td><td>Monitoring clock is selected as the clock for clock domain 0 or MCUC clock domain.</td></tr><tr><td>1</td><td>- (same condition)</td></tr></table>
Correct)
[bit16] JDGSEL: Judgment selection bit
This bit selects an additional condition under which a detected abnormal state (by CSV) leads to reset generation. Without the signalling of an abnormal state by the CSV, neither reset nor interrupt are generated. If the signal of an abnormal state occurs, the reaction depends on the condition which is selected by JDGSEL. If the condition as defined by the bit value description is true, the abnormal state leads to a reset, if it is false, an interrupt will be generated.
<table><tr><th>Bit</th><th>Description</th></tr><tr><td>0</td><td>PLLm clock is selected for clock domain 0 or for MCUC clock domain.</td></tr><tr><td>1</td><td>- (same condition)</td></tr></table>
– Example: When JDGSEL = 1 and sub clock is neither selected for clock domain 0 nor for MCUC, the selected condition is false and an abnormal state leads to interrupt generation.</td></tr>
<tr><td>692</td><td>CHAPTER 8:Clock Supervisor 5.7.SSCG PLLn Clock Supervisor Setting Register 0 (SYSC_CSVSPnCFGR0)</td><td>Error)
8.5.7 PLL1 Clock Supervisor Setting Register 0 (SYSC_CSVPLL1CFGR0)
PLL1 clock supervisor setting register 0 (SYSC_CSVPLL1CFGR0) sets the upper-limit threshold value and lower-limit threshold value of a frequency range.

Correct)
5.7.SSCG PLLn Clock Supervisor Setting Register 0 (SYSC_CSVSPnCFGR0)
SSCG PLLn clock supervisor setting register 0 (SYSC_CSVSPnCFGR0) sets the upper-limit threshold value and lower-limit threshold value of a frequency range.</td></tr>
<tr><td>692</td><td>CHAPTER 8:Clock Supervisor 5.7.SSCG PLLn Clock Supervisor Setting Register 0 (SYSC_CSVSPnCFGR0)</td><td>Error)
[bit27:16] UPTHR[11:0]: Upper-Limit Threshold Value Bits
These bits set the clock upper-limit threshold value for comparison with the monitoring clock count value. If the monitoring clock counter value exceeds the upper-limit threshold value, the state is judged as abnormal.

Correct)
[bit27:16] UPTHR[11:0]: Upper-limit threshold value bits
These bits set the clock upper-limit threshold value for comparison with the monitored clock count value. If the monitored clock counter value exceeds the upper-limit threshold value, the state is judged as abnormal.</td></tr>
</table>

Page	Section	Change Results
692 to 693		Error) **[bit9:0] LOWTHR[9:0]: Lower-Limit Threshold Value Bits** These bits set the clock lower-limit threshold value for comparison with the monitoring clock count value. If the monitoring clock counter value falls below the lower-limit threshold value, the state is judged as abnormal. Correct) **[bit9:0] LOWTHR[9:0]: Lower-limit threshold value bits** These bits set the clock lower-limit threshold value for comparison with the monitored clock count value. If the monitored clock counter value falls below the lower-limit threshold value, the state is judged as abnormal.
693		Error) **Notes:** ■ For the lower-limit threshold value bits (LOWTHR), set a value larger than that obtained from dividing the reference clock cycle by the monitoring clock cycle. Correct) − *For the lower-limit threshold value bits (LOWTHR), set a value larger than that obtained from dividing the reference clock cycle by the monitored clock cycle.*
694	CHAPTER 8:Clock Supervisor 5.8.SSCG PLLn Clock Supervisor Setting Register 1 (SYSC_CSVSPnCFGR1)	Error) **8.5.8 PLL1 Clock Supervisor Setting Register 1 (SYSC_CSVPLL1CFGR1)** PLL clock supervisor setting register 1 (SYSC_CSVPLLCFGR1) sets a reference clock count period. Correct) **5.8.SSCG PLLn Clock Supervisor Setting Register 1 (SYSC_CSVSPnCFGR1)** SSCG PLLn clock supervisor setting register 1 (SYSC_CSVSPnCFGR1) can set a reference clock count period. It can also select whether to generate a reset or interrupt at the abnormal state detection time.

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>694 to 695</td><td>CHAPTER 8:Clock Supervisor 5.8.SSCG PLLn Clock Supervisor Setting Register 1 (SYSC_CSVSPnCFGR1)</td><td>Error)
[bit16] JDGSEL: Judgment Selection Bit
Reset generation condition for when an abnormality is detected can be selected with this bit.
An interrupt can be generated when reset generation condition is not correspondence.
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Generate a reset.</td></tr>
<tr><td>1</td><td>No conditions (Generate an interrupt)</td></tr>
</table>
Correct)
[bit16] JDGSEL: Judgment selection bit
This bit selects an additional condition under which a detected abnormal state (by CSV) leads to reset generation. Without the signalling of an abnormal state by the CSV, neither reset nor interrupt are generated. If the signal of an abnormal state occurs, the reaction depends on the condition which is selected by JDGSEL. If the condition as defined by the bit value description is true, the abnormal state leads to a reset, if it is false, an interrupt will be generated.
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>SSCG PLLn clock is selected for clock domain 0 or for MCUC clock domain.</td></tr>
<tr><td>1</td><td>- (same condition)</td></tr>
</table>
– Example: When JDGSEL = 1 and SSCG PLL0 clock is neither selected for clock domain 0 nor for MCUC, the selected condition is false and an abnormal state leads to interrupt generation.</td></tr>
<tr><td>696</td><td>CHAPTER 8:Clock Supervisor 5.9.Fast CR Clock Supervisor Setting Register(SYSC_CSVFCRCFGR)</td><td>Error)
8.5.21 Fast CR Clock Supervisor Setting Register (SYSC_CSVFCRCFGR)
Fast CR clock supervisor setting register (SYSC_ CSVFCRCFGR) can select a monitoring clock division setting. It can also select a reference clock division setting.

Correct)
5.9. Fast CR Clock Supervisor Setting Register (SYSC_CSVFCRCFGR)
Fast CR clock supervisor setting register (SYSC_CSVFCRCFGR) can select a monitored clock division setting. It can also select a reference clock division setting.</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>696
to
697</td><td>CHAPTER 8:Clock Supervisor 5.9.Fast CR Clock Supervisor Setting Register(SYSC_CSVFCRCFGR)</td><td>Error)
[bit2:0] REFCLKDIV[2:0]: Reference Clock Division Setting Bits
<table><tr><th>bit2:0</th><th>Description</th></tr><tr><td>000</td><td>Divided by 4</td></tr><tr><td>001</td><td>Divided by 8</td></tr><tr><td>010</td><td>Divided by 16 (initial value)</td></tr><tr><td>011</td><td>Divided by 32</td></tr><tr><td>100</td><td>Divided by 64</td></tr><tr><td>others</td><td>Divided by 128</td></tr></table>
Correct)
[bit2:0] REFCLKDIV[2:0]: Reference clock division setting bits
<table><tr><th>Bits</th><th>Description</th></tr><tr><td>000</td><td>Divided by 4</td></tr><tr><td>001</td><td>Divided by 8</td></tr><tr><td>010</td><td>Divided by 16 (initial value)</td></tr><tr><td>011</td><td>Divided by 32</td></tr><tr><td>100</td><td>Divided by 64</td></tr><tr><td>others</td><td>Divided by 128</td></tr></table></td></tr>
<tr><td>698</td><td>CHAPTER 8:Clock Supervisor 5.10.Slow CR Clock Supervisor Setting Register(SYSC_CSVSCRCFGR)</td><td>Error)
8.5.22 Slow CR Clock Supervisor Setting Register (SYSC_CSVSCRCFGR)
Slow CR clock supervisor setting register (SYSC_ CSVSCRCFGR) can select a monitoring clock division setting. It can also select a reference clock division setting.

Correct)
5.10. Slow CR Clock Supervisor Setting Register (SYSC_CSVSCRCFGR)
Slow CR clock supervisor setting register (SYSC_CSVSCRCFGR) can select a monitored clock division setting. It can also select a reference clock division setting.</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>698
to
699</td><td>CHAPTER 8:Clock Supervisor 5.10.Slow CR Clock Supervisor Setting Register(SYSC_CSVSCRCFGR)</td><td>Error)
[bit2:0] REFCLKDIV[2:0]: Reference Clock Division Setting Bits
<table><tr><th>bit2:0</th><th>Description</th></tr><tr><td>000</td><td>Divided by 4</td></tr><tr><td>001</td><td>Divided by 8</td></tr><tr><td>010</td><td>Divided by 16 (initial value)</td></tr><tr><td>011</td><td>Divided by 32</td></tr><tr><td>100</td><td>Divided by 64</td></tr><tr><td>others</td><td>Divided by 128</td></tr></table>
Correct)
[bit2:0] REFCLKDIV[2:0]: Reference clock division setting bits
<table><tr><th>Bits</th><th>Description</th></tr><tr><td>000</td><td>Divided by 4</td></tr><tr><td>001</td><td>Divided by 8</td></tr><tr><td>010</td><td>Divided by 16 (initial value)</td></tr><tr><td>011</td><td>Divided by 32</td></tr><tr><td>100</td><td>Divided by 64</td></tr><tr><td>others</td><td>Divided by 128</td></tr></table></td></tr>
<tr><td>700</td><td>CHAPTER 8:Clock Supervisor 5.11.Clock Supervisor Output Enable Register(SYSC_CSVOUTER)</td><td>Error)
8.5.23 Clock Supervisor Output Enable Register (SYSC_CSVOUTER)

Correct)
5.11. Clock Supervisor Output Enable Register (SYSC_CSVOUTER)</td></tr>
<tr><td>701</td><td>CHAPTER 8:Clock Supervisor 5.12.Clock Supervisor Test Register(SYSC_CSVTESTER)</td><td>Error)
8.5.24 Clock Supervisor Test Register (SYSC_CSVTESTR)

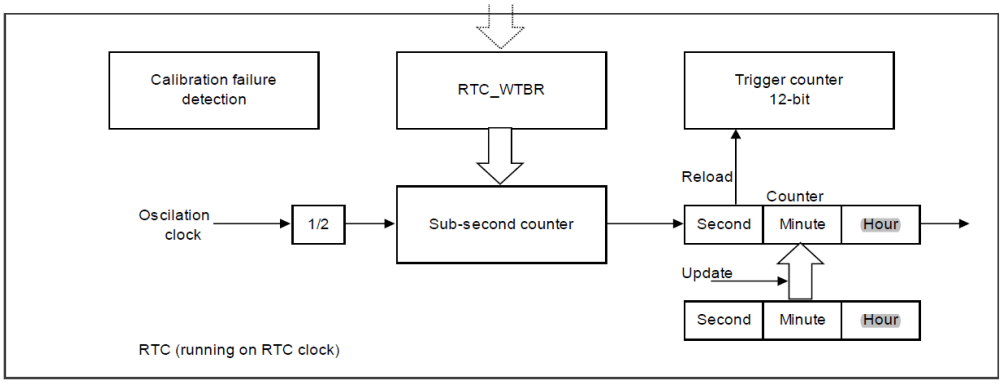
Correct)
5.12. Clock Supervisor Test Register (SYSC_CSVTESTR)</td></tr>
</table>

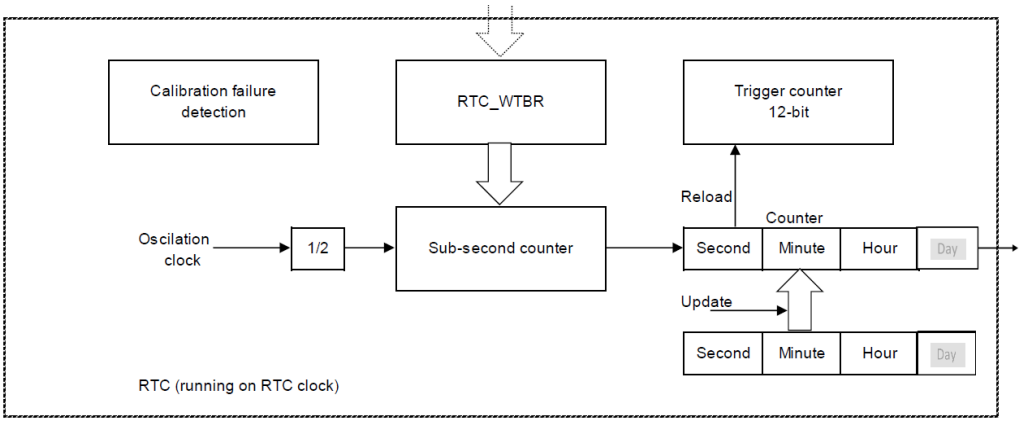
Page	Section	Change Results
705	CHAPTER 8:Clock Supervisor 6.Precautions for Using This Device	Error) ■ To operate the clock supervisor, enable oscillation of the monitoring clock and reference clock. ■ To stop the clock supervisor, simultaneously stop oscillation of the monitoring clock. Correct) ■ To operate the clock supervisor, enable oscillation of the monitored clock and reference clock. ■To stop the clock supervisor, simultaneously stop oscillation of the monitored clock.

Page	Section	Change Results
711	CHAPTER 9:Source Clock Timer 3. Explanation of Operation	Error) **Oscillation Stabilization Wait Operation** ■ The source clock timer starts the oscillation stabilization wait operation when a hard reset is released or oscillation is enabled. The source clock timer gates the clock output until the oscillation stabilization wait time ends. ■ If data is written to a trigger register, timer control register, timer compare prescaler register, interrupt enable register, or interrupt clear register during the oscillation stabilization wait operation, release the sequence protection in privileged mode. If the above is not complied with, a bus error is returned. ■ During the oscillation stabilization wait operation, the source clock timer operates in single-shot mode. Correct) **Oscillation Stabilization Wait Operation** ■ Main, Fast-CR and Slow-CR source clock timers start the oscillation stabilization wait operation when a hard reset is released or oscillation is enabled. The source clock timers gate the clock output until the oscillation stabilization wait time ends. ■ Sub source clock timer starts the oscillation stabilization wait operation when oscillation is enabled. The source clock timer gates the clock output until the oscillation stabilization wait time ends. ■ If data is written to a trigger register, timer control register, timer compare prescaler register, interrupt enable register, or interrupt clear register during the oscillation stabilization wait operation, release the sequence protection in privileged mode. If the above is not complied with, a bus error is returned. ■ During the oscillation stabilization wait operation, the source clock timer operates in single-shot mode.
-	CHAPTER 9:Source Clock Timer 5. Registers	Deleted as below: Error) About register attributes There is no register attribute "p" because the base version does not have a PPU.

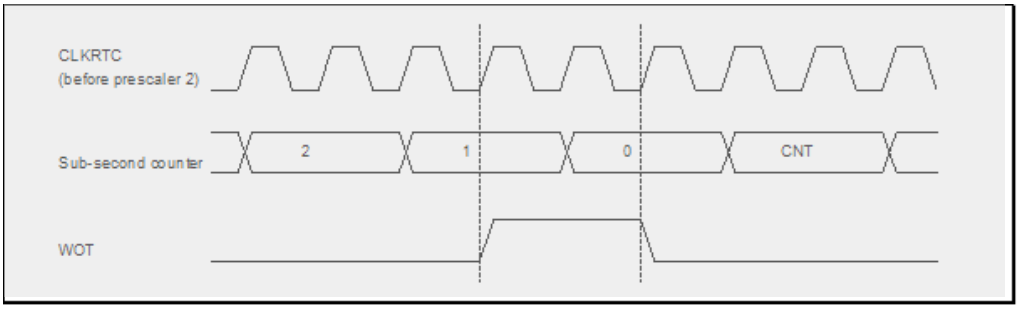
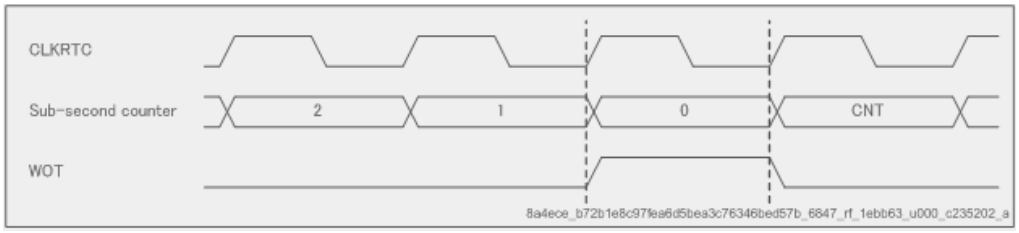
<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>721</td><td>CHAPTER 9:Source Clock Timer 5. Registers 5.3. High-Speed CR Clock Timer Compare Prescaler Register (SYSC_FCRCTCPR)</td><td>Error)
Oscillation stabilization wait time of the high-speed CR clock
<table><tr><th>Prescale
PSCL[3:0]</th><th>Compare Value
CMPR[15:0]</th><th>Oscillation Stabilization Wait Time
(High-Speed CR Clock Cycle x Prescale Value x Compare Value)</th></tr><tr><td>0110 (divided by 64)</td><td>0x001E(30)</td><td>0.24 to 0.48 [ms]</td></tr></table>
Correct)
Oscillation stabilization wait time of the high-speed CR clock
<table><tr><th>Prescale
PSCL[3:0]</th><th>Compare Value
CMPR[15:0]</th><th>Oscillation Stabilization Wait Time
(High-Speed CR Clock Cycle x Prescale Value x Compare Value)</th></tr><tr><td>0110 (divided by 64)</td><td>0x001E(30)</td><td>Product specification</td></tr></table></td></tr>
<tr><td>731</td><td>CHAPTER 9:Source Clock Timer 5. Registers 5.9. Low-Speed CR Clock Timer Compare Prescaler Register (SYSC_SCRCTCPR)</td><td>Error)
Oscillation stabilization wait time of the low-speed CR clock
<table><tr><th>Prescale
PSCL[3:0]</th><th>Compare Value
CMPR[15:0]</th><th>Oscillation Stabilization Wait Time
(Low-Speed CR Clock Cycle x Prescale Value x Compare Value)</th></tr><tr><td>0110 (divided by 64)</td><td>0x0001(1)</td><td>0.43 to 1.28 [ms]</td></tr></table>
Correct)
Oscillation stabilization wait time of the low-speed CR clock
<table><tr><th>Prescale
PSCL[3:0]</th><th>Compare Value
CMPR[15:0]</th><th>Oscillation Stabilization Wait Time
(Low-Speed CR Clock Cycle x Prescale Value x Compare Value)</th></tr><tr><td>0110 (divided by 64)</td><td>0x0001(1)</td><td>Product specification</td></tr></table></td></tr>
</table>

Page	Section	Change Results
758	CHAPTER 10 Real Time Clock 1.Overview	Error) The Real Time Clock (RTC) Timer module provides the current real time (HH/MM/SS) and the Calibration module calibrates Sub clock or Slow RC clock with respect to the Main oscillator clock. Correct) The Real Time Clock (RTC) Timer module provides the current real time (HH/MM/SS) and the Calibration module calibrates Sub clock or Slow CR clock with respect to the Main oscillator clock.
		Error) **Features of the Calibration Module** ■The automatic cyclic calibration/measurement of Sub clock (CLK_SUB) or Slow RC clock (CLK_SRC) against Main clock (CLK_MAIN) Correct) ■The automatic cyclic calibration/measurement of Sub clock (CLK_SUB) or Slow CR clock (CLK_SRC) against Main clock (CLK_MAIN)

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>759</td><td>CHAPTER 10 Real Time Clock 2.Configuration and Block Diagram</td><td>Error)
Figure 10.2-1 Block Diagram of RTC


Correct)
Figure 2-1 Block Diagram of RTC

</td></tr>
<tr><td>762</td><td>CHAPTER 10 Real Time Clock 3.Operation of the Real Time Clock</td><td>Error)
RTC Timer Module Operation
10. The Sub-second counter begins counting on the selected CLKRTC divided by 2. This could be the Main clock (CLK_MAIN) divided by 2 (4/2 MHz), Sub clock (CLK_SUB) divided by 2 (32.768/2 kHz) or Slow RC clock (CLK_SRC) divided by 2 (100/2 kHz).

Correct)
RTC Timer Module Operation
10. The Sub-second counter begins counting on the CLKRTC. This could be the Main clock (CLK_MAIN) divided by 2 (4/2 MHz), Sub clock (CLK_SUB) divided by 2 (32.768/2 kHz) or Slow CR clock (CLK_SRC) divided by 2 (100/2 kHz).</td></tr>
</table>

Page	Section	Change Results
764	CHAPTER 10 Real Time Clock 3.Operation of the Real Time Clock	Error) **Figure 10.3-2 WOT Pin Operation**  Correct) **Figure 3-2 WOT Pin Operation** 

Page	Section	Change Results
765	CHAPTER 10 Real Time Clock 3.Operation of the Real Time Clock	Error) Automatic Calibration: 4. Write the Calibration trigger counter value to RTC_CALTRG to configure the time interval between two calibration cycles. For low-power mode, this also defines the intervals to wake up the Main oscillator. 5. Set RTC_WTCR:ENUP to '1' to enable the automatic transfer of the measured calibration value to the Sub-Second Register (RTC_WTBR) at the end of every calibration cycle. 6. Set RTC_WTCR:ACAL to '1', to enable Automatic Calibration. When the RTC is operating (RTC_WTSR:RUN) = '1', the Calibration trigger counter will run and generate cyclic calibration trigger events. 7. Set RTC_WINE:CFDE to '1' to enable interrupt generation (and MCU wake-up) in case of a calibration failure. An interrupt request will be generated when a calibration failure is detected if RTC_WTBR is not updated within one second after the hardware trigger. This happens when the Main clock is missing. Correct) Automatic Calibration: 1. Write the Calibration trigger counter value to RTC_CALTRG to configure the time interval between two calibration cycles. For low-power mode, this also defines the intervals to wake up the Main oscillator. 2. Set RTC_WTCR:ENUP to '1' to enable the automatic transfer of the measured calibration value to the Sub-Second Register (RTC_WTBR) at the end of every calibration cycle. 3. Set RTC_WTCR:ACAL to '1', to enable Automatic Calibration. When the RTC is operating (RTC_WTSR:RUN) = '1', the Calibration trigger counter will run and generate cyclic calibration trigger events. 4. Set RTC_WINE:CFDE to '1' to enable interrupt generation (and MCU wake-up) in case of a calibration failure. An interrupt request will be generated when a calibration failure is detected if RTC_WTBR is not updated within one second after the hardware trigger. This happens when the Main clock is missing.
765	CHAPTER 10 Real Time Clock 3.Operation of the Real Time Clock	Error) **Notes:** ■ For Automatic Calibration in low-power mode, ensure a valid stabilization time for CLK_MAIN ■ oscillation is configured. Information about stabilization time can be found in CHAPTER 2. Correct) ***Note:*** − *For Automatic Calibration in low-power mode, ensure a valid stabilization time for CLK_MAIN oscillation is configured. Information about stabilization time can be found in CHAPTER 9.*

Page	Section	Change Results
767	CHAPTER 10 Real Time Clock 3.Operation of the Real Time Clock	Error) **Cyclic Calibration** The Calibration Trigger Register (RTC_CALTRIG) value defines the duration between two Calibrations. Correct) **Cyclic Calibration** The Calibration Trigger Register (RTC_CALTRG) value defines the duration between two calibrations.

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>768</td><td>CHAPTER 10
Real Time Clock
4.Registers</td><td>
Error)
Table 10.4-1 Registers of Real Time Clock
<table>
<tr><th>Abbreviated expressions</th><th>Register Name</th><th>Reference</th></tr>
<tr><td>RTC_WTCR</td><td>Timer Control Register</td><td>10.4.1</td></tr>
<tr><td>RTC_WTSR</td><td>Timer Status Register</td><td>10.4.2</td></tr>
<tr><td>RTC_WINS</td><td>Interrupt Status Register</td><td>10.4.3</td></tr>
<tr><td>RTC_WINE</td><td>Interrupt Enable Register</td><td>10.4.4</td></tr>
<tr><td>RTC_WINC</td><td>Interrupt Clear Register</td><td>10.4.5</td></tr>
<tr><td>RTC_WTBR</td><td>Sub-Second Register</td><td>10.4.6</td></tr>
<tr><td>RTC_WRT</td><td>Real Time Register</td><td>10.4.7</td></tr>
<tr><td>RTC_RTR1</td><td>Real Time Register1</td><td>10.4.8</td></tr>
<tr><td>RTC_CNTCAL</td><td>Calibration Clock Counter Register</td><td>10.4.9</td></tr>
<tr><td>RTC_CNTPCAL</td><td>Calibration Clock Period Counter Register</td><td>10.4.10</td></tr>
<tr><td>RTC_DURMW</td><td>Calibration Duration Register</td><td>10.4.11</td></tr>
<tr><td>RTC_CALTRG</td><td>Calibration Trigger Register</td><td>10.4.12</td></tr>
<tr><td>RTC_DEBUG</td><td>Debug Register</td><td>10.4.13</td></tr>
<tr><td>RTC_PWUTRGCR</td><td>Partial Wake Up Trigger Control Register</td><td>10.4.14</td></tr>
<tr><td>RTC_PWUTRGSR</td><td>Partial Wake Up Trigger Status Register</td><td>10.4.15</td></tr>
</table>
Register map is not required because it can be extracted from each register descriptions.
For the customers, they should refer DS for the I/O map.

Correct)
Table 4-1 Registers of Real Time Clock
<table>
<tr><th>Abbreviated expressions</th><th>Register Name</th><th>Reference</th></tr>
<tr><td>RTC_WTCR</td><td>Timer Control Register</td><td>4.1</td></tr>
<tr><td>RTC_WTSR</td><td>Timer Status Register</td><td>4.2</td></tr>
<tr><td>RTC_WINS</td><td>Interrupt Status Register</td><td>4.3</td></tr>
<tr><td>RTC_WINE</td><td>Interrupt Enable Register</td><td>4.4</td></tr>
<tr><td>RTC_WINC</td><td>Interrupt Clear Register</td><td>4.5</td></tr>
<tr><td>RTC_WTBR</td><td>Sub-Second Register</td><td>4.6</td></tr>
<tr><td>RTC_WRT</td><td>Real Time Register</td><td>4.7</td></tr>
<tr><td>RTC_RTR1</td><td>Real Time Register1</td><td>4.8</td></tr>
<tr><td>RTC_CNTCAL</td><td>Calibration Clock Counter Register</td><td>4.9</td></tr>
<tr><td>RTC_CNTPCAL</td><td>Calibration Clock Period Counter Register</td><td>4.10</td></tr>
<tr><td>RTC_DURMW</td><td>Calibration Duration Register</td><td>4.11</td></tr>
<tr><td>RTC_CALTRG</td><td>Calibration Trigger Register</td><td>4.12</td></tr>
<tr><td>RTC_DEBUG</td><td>Debug Register</td><td>4.13</td></tr>
<tr><td>RTC_PWUTRGCR</td><td>Partial Wake Up Trigger Control Register</td><td>4.14</td></tr>
<tr><td>RTC_PWUTRGSR</td><td>Partial Wake Up Trigger Status Register</td><td>4.15</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>770</td><td>CHAPTER 10 Real Time Clock 4.Registers 4.1.Timer Control Register (RTC_WTCR)</td><td>
Error)
10.4.1 Timer Control Register (RTC_WTCR)
<table>
<tr><td>BIT_OFFSET</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td>BIT_NAME</td><td>UPCAL</td><td colspan="3">SCAL[2:0]</td><td>CCKSEL</td><td>ENUP</td><td>MTRG</td><td>ACAL</td></tr>
<tr><td>ACCESS_TYPE</td><td>R,W</td><td colspan="3">R/W</td><td>R/W</td><td>R/W</td><td>R0,W</td><td>R/W</td></tr>
<tr><td>PROT_TYPE</td><td colspan="8">-</td></tr>
<tr><td>INITIAL_VALUE</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
Correct)
4.1. Timer Control Register (RTC_WTCR)
<table>
<tr><td>BIT_OFFSET</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td>BIT_NAME</td><td>UPCAL</td><td colspan="3">SCAL[2:0]</td><td>CCKSEL</td><td>ENUP</td><td>MTRG</td><td>ACAL</td></tr>
<tr><td>ACCESS_TYPE</td><td>R,W</td><td colspan="3">R,W</td><td>R/W</td><td>R,W</td><td>R0,W</td><td>R,W</td></tr>
<tr><td>PROT_TYPE</td><td colspan="8">-</td></tr>
<tr><td>INITIAL_VALUE</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
</td></tr>
<tr><td>771</td><td>CHAPTER 10 Real Time Clock 4.Registers 4.1.Timer Control Register (RTC_WTCR)</td><td>
Error)
[bit11] CCKSEL: Clock select for calibration
This bit is used to select the Calibration clock (CLKCAL).
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Sub clock (CLK_SUB) is used as CLKCAL</td></tr>
<tr><td>1</td><td>Slow RC clock (CLKSRC) is used as CLKCAL</td></tr>
</table>
Correct)
[bit11] CCKSEL: Clock select for calibration
This bit is used to select the Calibration clock (CLKCAL).
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Sub clock (CLK_SUB) is used as CLKCAL</td></tr>
<tr><td>1</td><td>Slow CR clock (CLK_SRC) is used as CLKCAL</td></tr>
</table>
</td></tr>
<tr><td>772</td><td>CHAPTER 10 Real Time Clock 4.Registers 4.1.Timer Control Register (RTC_WTCR)</td><td>
Error)
[bit8] ACAL: Automatic calibration
Note:
For a detailed description of the Automatic RTC Calibration, refer to Section 3 'Operation of the Real Time Clock' and Section 5 'Cautions'.

Correct)
[bit8] ACAL: Automatic calibration
Notes:

For a detailed description of the Automatic RTC Calibration, refer to Section 3 'Operation of the Real Time Clock' and Section 5 'Usage Precaution'
Because of the internal clock synchronization, it takes certain period (several Main clock period and several bus clock period) that the written value is reflected to the bit.

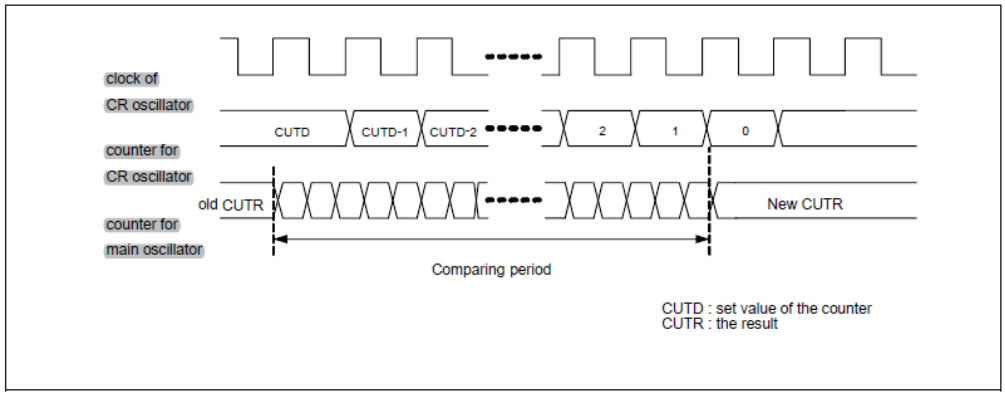
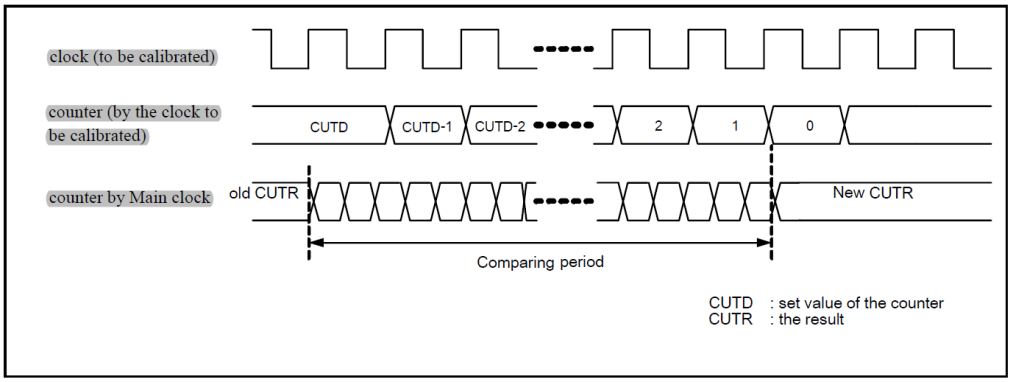
</td></tr>
</table>

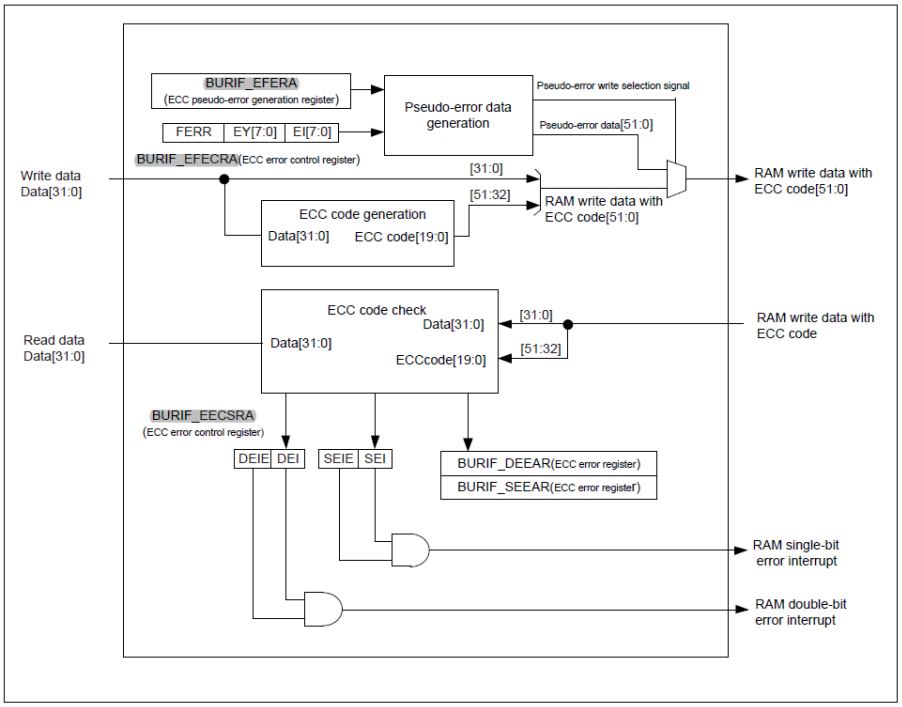
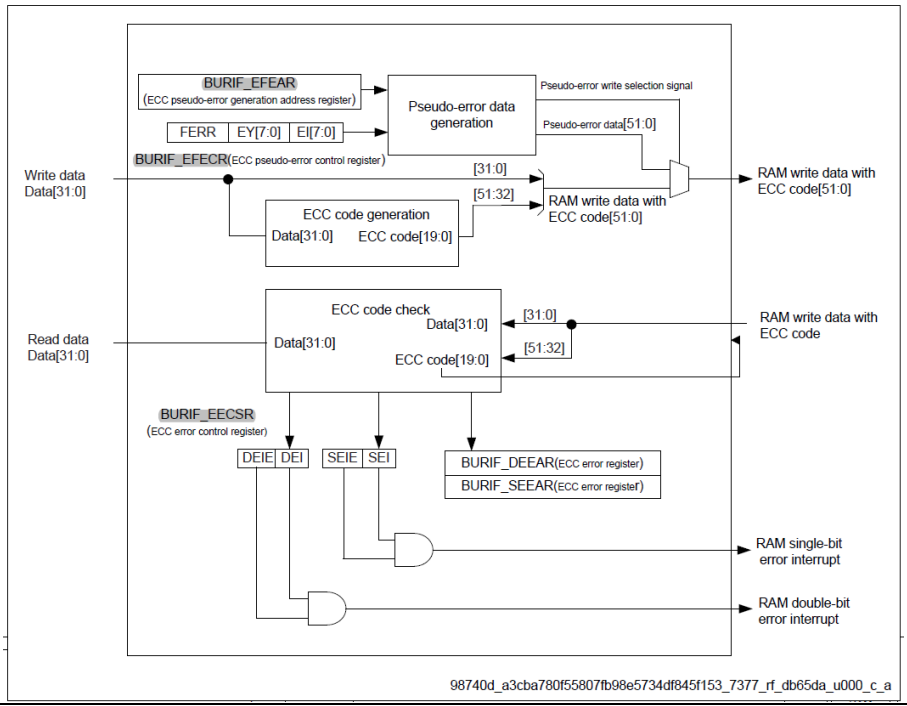
<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>772
to
773</td><td>CHAPTER 10
Real Time Clock
4.Registers
4.1.Timer Control Register (RTC_WTCR)</td><td>Error)
[bit5:4] RCKSEL[1:0]: Clock select for RTC
These bits are used to select the input clock for the RTC (CLKRTC).
<table><tr><th>Bits</th><th>Description</th></tr><tr><td>00</td><td>Main clock (CLK_MAIN/2) is selected as CLKRTC</td></tr><tr><td>01</td><td>Sub clock (CLK_SUB/2) is selected as CLKRTC</td></tr><tr><td>10</td><td>Slow RC clock (CLK_SRC/2) is selected as CLKRTC</td></tr><tr><td>11</td><td>Prohibited</td></tr></table>
Correct)
[bit5:4] RCKSEL[1:0]: Clock select for RTC
These bits are used to select the input clock for the RTC (CLKRTC).
<table><tr><th>Bits</th><th>Description</th></tr><tr><td>00</td><td>Main clock (CLK_MAIN/2) is selected as CLKRTC</td></tr><tr><td>01</td><td>Sub clock (CLK_SUB/2) is selected as CLKRTC</td></tr><tr><td>10</td><td>Slow CR clock (CLK_SRC/2) is selected as CLKRTC</td></tr><tr><td>11</td><td>Prohibited</td></tr></table></td></tr>
<tr><td>773</td><td>CHAPTER 10
Real Time Clock
4.Registers
4.1.Timer Control Register (RTC_WTCR)</td><td>Error)
[bit2] UPDT: Update
Updating by the CPU has a higher priority than updating by hardware (in case both updates come at the same time). Refer to section 5 'Cautions'.

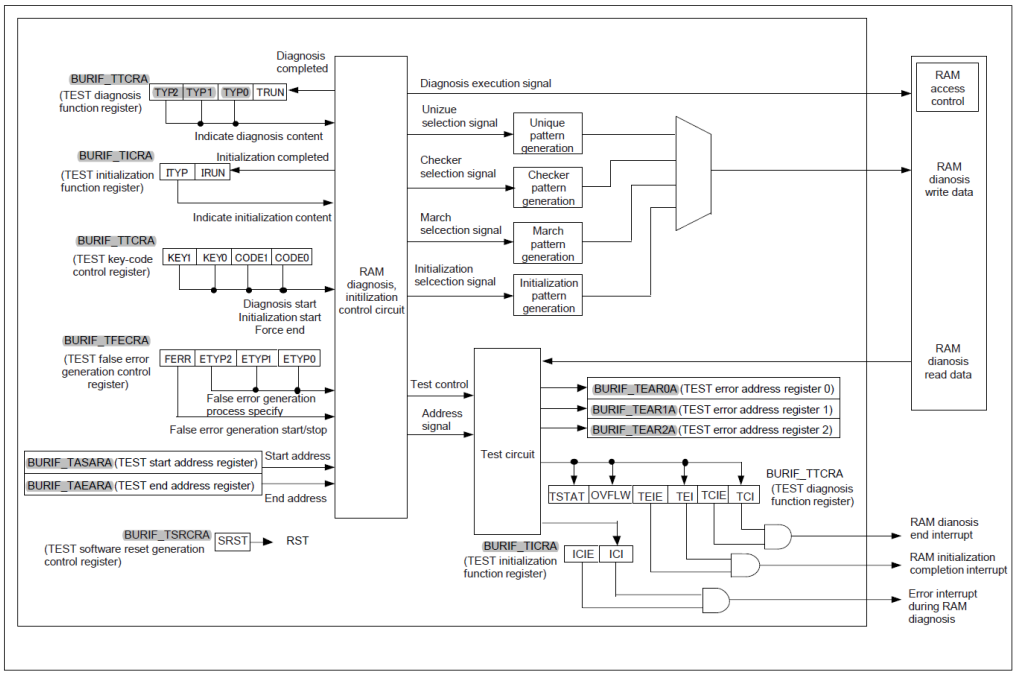
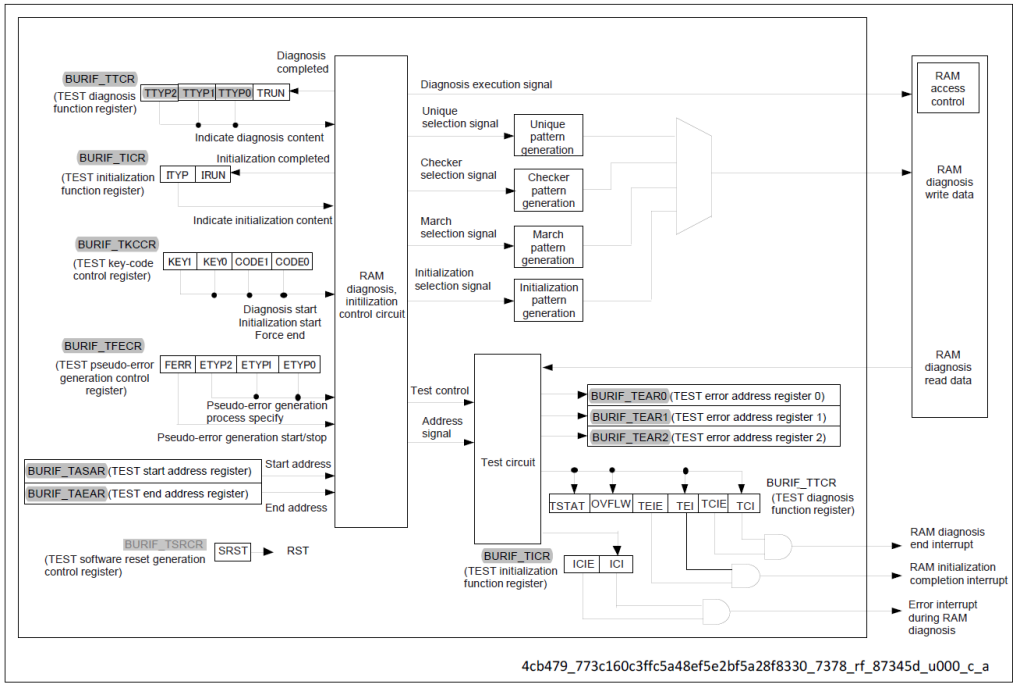
Correct)
[bit2] UPDT: Update
Updating by the CPU has a higher priority than updating by hardware (in case both updates come at the same time). Refer to section 5 'Usage Precaution'.</td></tr>
<tr><td>786</td><td>CHAPTER 10
Real Time Clock
Real Time Clock
4.Registers
4.6.Sub-Second Register (RTC_WTBR)</td><td>Error)
Table 10.4-3 Example Configuration of RTC_WTBR Register for Different Clock Configurations
Calculation formula:
RTC_WTBR:WTBR[23:0] = fCLKRTC/2 x 0.5 s – 1
fCLKRTC: CLKRTC frequency [Hz]

Correct)
Table 4-3 Example Configuration of RTC_WTBR Register for Different Clock Configurations
Calculation formula:
RTC_WTBR:WTBR[23:0] = fCLKRTC x 0.5 s – 1
fCLKRTC: CLKRTC frequency [Hz]</td></tr>
</table>

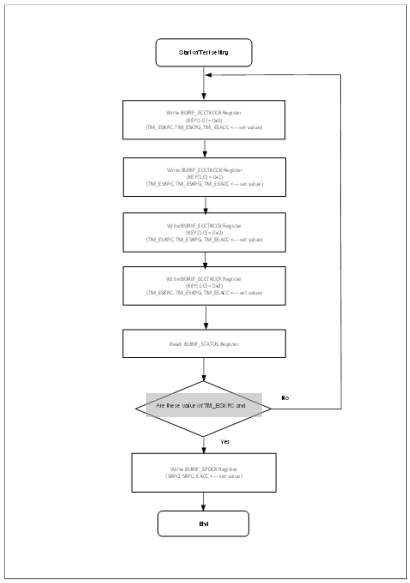
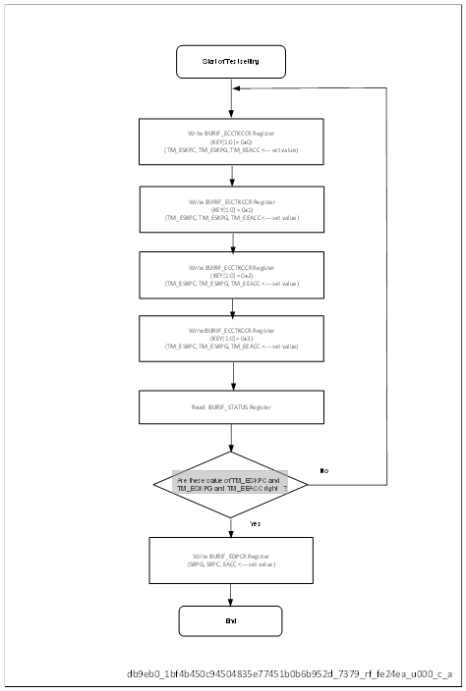
Page	Section	Change Results
802	CHAPTER 10 Real Time Clock 5.Usage Precaution	Error) **Cautions** ■ When updating the Hour/Minute/Second registers using the RTC_WTCR:ST bit, the following must be taken into account: The new value is written into the registers with the rising edge of the RUN bit. This RUN bit is clocked by the CLKRTC/2. Correct) **Cautions** ■ When updating the Hour/Minute/Second registers using the RTC_WTCR:ST bit, the following must be taken into account: The new value is written into the registers with the rising edge of the RUN bit. This RUN bit is clocked by the CLKRTC.

Page	Section	Change Results
806	CHAPTER11: CR Calibration 1.Overview	Error) **Figure 11.1-1 Comparison of Counters Driven by Each Clock**  Correct) **Figure 1-1 Comparison of Counters Driven by Each Clock** 

Page	Section	Change Results
821	CHAPTER 12:BURAM 2.Configuration	Error) **Figure 12.2-2 RAMECC Function Block Diagram**  Correct) **Figure 2-2 RAMECC Function Block Diagram** 

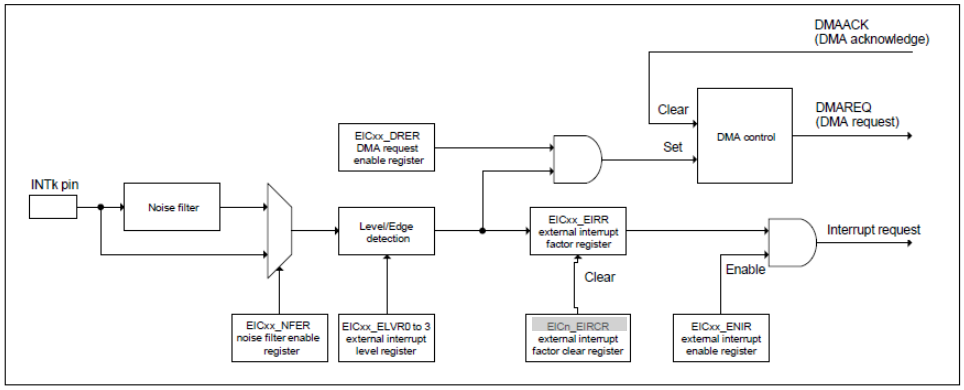
Page	Section	Change Results
822	CHAPTER 12:BURAM 2.Configuration	Error) **Figure 12.2-3 RAMECC Diagnosis Function Block Diagram**  Correct) **Figure 2-3 RAMECC Diagnosis Function Block Diagram** 

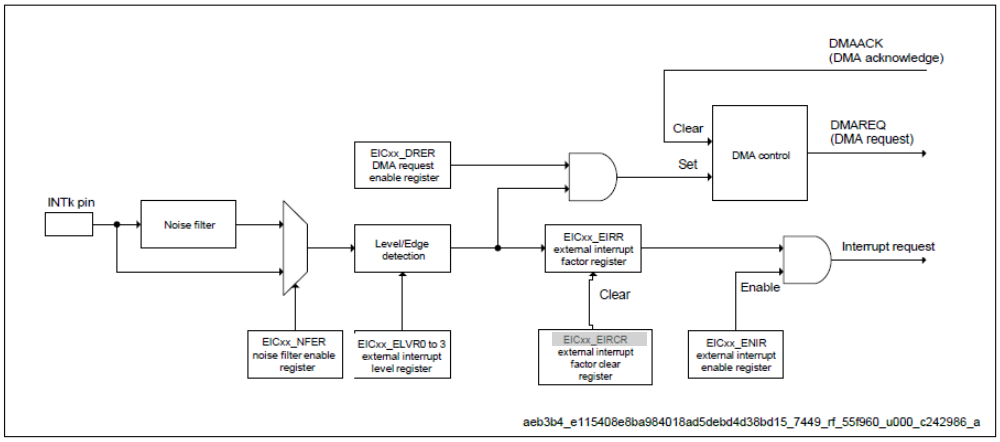
Page	Section	Change Results
828	CHAPTER 12:BURAM 3.2. RAM Diagnosis	Error) 2. Set the start of diagnosis processing in the TEST diagnosis function register (BURIF_TTCR). (B) In the BURIF_TEST key code control register (BURIF_TKCCR), consecutively write "02h", "42h", "82h", and "C2h," in that order, to set the diagnosis pattern using the operation start procedure. Correct) 2. Set the start of diagnosis processing in the TEST diagnosis function register (BURIF_TTCR). (B) In the TEST key code control register (BURIF_TKCCR), consecutively write "02h", "42h", "82h", and "C2h," in that order, to set the diagnosis pattern using the operation start procedure.

Page	Section	Change Results
834	CHAPTER 12:BURAM 4. Setting Procedure	Error) **Figure 12.4-2 Setting Flow of ECC Test Mode Key Code Control Register**  Correct) **Figure 4-2 Setting Flow of ECC Test Mode Key Code Control Register** 

Page	Section	Change Results
847	CHAPTER 12:BURAM 5. Registers 5.7. BURAMIF ECC Pseudo-Error Generation Control Register (BURIF_EFECR)	Error) The ECC pseudo-error generation control register (EFECR) specifies the content of the Backup RAM pseudo-error to be generated in the form of a generation byte and generation bit. Correct) The ECC pseudo-error generation control register specifies the content of the Backup RAM pseudo-error to be generated in the form of a generation byte and generation bit.
851	CHAPTER 12:BURAM 5. Registers 5.9. BURAMIF ECC Test Mode Key Code Control Register (BURIF_ECCTKCCR)	Error) **Note:** Set the same values in the TM_ESKPG, TM_ESKPGC and TM_EEACC bits while KEY bit operation is being performed. For details on the procedure for setting these bits, see "Figure 4 2 Setting Flow of ECC Test Mode Key Code Control Register" Correct) ***Note:*** *- Set the same values in the TM_ESKPG, TM_ESKPC and TM_EEACC bits while KEY bit operation is being performed. For details on the procedure for setting these bits, see "Figure 4 2 Setting Flow of ECC Test Mode Key Code Control Register"*
858	CHAPTER 12:BURAM 5. Registers 5.13. BURAMIF TEST Start Address Register (BURIF_TASAR)	Error) **Notes:** ■ SADDR is an offset address in units of words (RAM address). ■ To calculate the absolute address, add {SADDR, 2'b00} to the base address. ■ A value outside the Backup RAM area and a value such that BURIF_TASARX.SADDR > BURIF_TAEAR.EADDR cannot be set. Correct) ***Notes:*** *- SADDR is an offset address in units of words (RAM address).* *- To calculate the absolute address, add {SADDR, 2'b00} to the base address.* *- A value outside the Backup RAM area and a value such that BURIF_TASAR.SADDR > BURIF_TAEAR.EADDR cannot be set.*

Page	Section	Change Results
859	CHAPTER 12:BURAM 5. Registers 5.14. BURAMIF TEST End Address Register (BURIF_TAEAR)	Error) **Notes:** ■ EADDR is an offset in word units (RAM address). ■ Obtain the absolute address by adding {EADDR, 2'b00} to the base address. ■ A value outside the Backup RAM area and a value such that BURIF_TASARX.SADDR > BURIF_TAEAR.EADDR cannot be set. Correct) ***Notes:*** - *EADDR is an offset in word units (RAM address).* - *Obtain the absolute address by adding {EADDR, 2'b00} to the base address.* - *A value outside the Backup RAM area and a value such that BURIF_TASAR.SADDR > BURIF_TAEAR.EADDR cannot be set.*

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>871</td><td>CHAPTER 13: External Interrupt
2. Configuration
2.1. Block Diagrams</td><td>Error)
Figure 2-1 External Interrupt Block Diagram


Correct)
Figure 2-1 External Interrupt Block Diagram

</td></tr>
<tr><td>872</td><td>CHAPTER 13: External Interrupt
3.Explanation of Operation</td><td>Error)
If edge detection is specified as an event input and the noise filter is enabled, a rule applies to the pulse width of an input signal to be recognized as an input edge. For the minimum value of the pulse width, see the PF-Cluster design sheet.

Correct)
If edge detection is specified as an event input and the noise filter is enabled, a rule applies to the pulse width of an input signal to be recognized as an input edge. For the minimum value of the pulse width, see the Traveo Family Hardware Manual Platform Part design sheet.</td></tr>
</table>

Page	Section	Change Results
898	CHAPTER 14:Hardware Watchdog Timer 1.Overview	Error) **Hardware Watchdog Timer Features** ■ It provides the function for outputting the watchdog counter monitor bit for the MCU output pin so that the watchdog timer counter can be observed from the outside. (Support target of this function is different depends on product specification. For details, see "3 Explanation of Operation For this product, output of the watchdog counter monitor bit of MCU output pin is supported by the Hardware watchdog timer.") Correct) **Hardware Watchdog Timer Features** ■ It provides the function for outputting the watchdog counter monitor bit for the MCU output pin so that the watchdog timer counter can be observed from the outside. (Support target of this function is different depends on product specification. For details, see "Output of the Watchdog Counter Monitor Bit for MCU Output Pin" in 3 Explanation of Operation.)
905 to 906	CHAPTER 14:Hardware Watchdog Timer 3. Explanation of Operation	Error) **Generation of Watchdog Reset Request or Watchdog Interrupt Request (NMI)** **Notes:** When you generate the prior warning interrupt request, set the Hardware watchdog reset delay counter register (HWDG_RSTDLY) to a value other than 0x0000. The use of the watchdog interrupt request (NMI) is prohibited except when used as a test function. (This is prohibited because if such use is allowed, security will be compromised during the execution of an application.) The following figure shows the operation of the test function of the Hardware watchdog. ■ Insertion of a number of cycles equal to that for the delay time set in the Hardware watchdog reset delay counter register (HWDG_RSTDLY) is valid only for 1 watchdog reset request or watchdog interrupt request (NMI). (It remains invalid until the next hard reset occurs.) Correct) **Generation of Watchdog Reset Request or Watchdog Interrupt Request (NMI)** ***Notes:*** − *When you generate the prior warning interrupt request, set the Hardware watchdog reset delay counter register (HWDG_RSTDLY) to a value other than 0x0000.* − *The use of the watchdog interrupt request (NMI) is prohibited except when used as a test function. (This is prohibited because if such use is allowed, security will be compromised during the execution of an application.) Figure 3-6 shows the operation of the test function of the Hardware watchdog.* − *Insertion of a number of cycles equal to that for the delay time set in the Hardware watchdog reset delay counter register (HWDG_RSTDLY) is valid only for 1 watchdog reset request or watchdog interrupt request (NMI). (It remains invalid until the next hard reset occurs.)*

Page	Section	Change Results
906	CHAPTER 14:Hardware Watchdog Timer 3. Explanation of Operation	Error) **Output of the Watchdog Counter Monitor Bit for MCU Output Pin** For This Product, Output of the Watchdog Counter Monitor Bit of MCU Output Pin is Supported by the Hardware Watchdog Timer. The Hardware watchdog timer outputs the watchdog counter monitor bit to the outside so that watchdog counter operation can be monitored from the outside. You can select any 1 bit in the watchdog counter (32 bits) by setting the OBSSEL[4:0] bits in the Hardware watchdog configuration register (HWDG_CFG). Correct) **Output of the Watchdog Counter Monitor Bit for MCU Output Pin** The Hardware watchdog timer outputs the watchdog counter monitor bit to the outside so that watchdog counter operation can be monitored from the outside. You can select any 1 bit in the watchdog counter (32 bits) by setting the OBSSEL[4:0] bits in the Hardware watchdog configuration register (HWDG_CFG). Support of this function is different depends on product specification.
908	CHAPTER 14:Hardware Watchdog Timer 3. Explanation of Operation	Error) **Notes:** ■ The following register setting values are determined by a marker in the flash memory (BootROM marker). □ HWDT_INT □ HWDT_TRG0CFG □ HWDT_TRG1CFG □ HWDT_RUNLLS □ HWDT_RUNULS □ HWDT_PSSLLS □ HWDT_PSSULS □ HWDT_RSTDLY □ HWDT_CFG Correct) ***Notes:*** − *The following register setting values are determined by a marker in the flash memory (BootROM marker).* − HWDG_INT − HWDG_TRG0CFG − HWDG_TRG1CFG − HWDG_RUNLLS − HWDG_RUNULS − HWDG_PSSLLS − HWDG_PSSULS − HWDG_RSTDLY − HWDG_CFG

Page	Section	Change Results
913	CHAPTER 14:Hardware Watchdog Timer 3. Explanation of Operation	Error) The watchdog counter in RUN always operates. When the device state transitions from RUN to PSS, the processing transitions to the operation of the watchdog in PSS. For details, see The following figure shows the Hardware watchdog operation in PSS. During the operation of the watchdog counter, execute the watchdog counter clear protection trigger sequence within the range between the window upper limit value and lower limit value that have been specified beforehand. Do this to clear the watchdog counter periodically. 6. If a runaway state caused by a user program occurs and prevents periodic clearing, the watchdog counter reaches the window upper limit value. Then, processing transitions to the flow that generates the watchdog reset request or watchdog interrupt request (NMI). 7. According to the value of the IRQEN bit in the Hardware watchdog interrupt configuration register (HWDG_INT), the prior warning interrupt request is generated. At the same time, as many cycles as are set for the delay time in the Hardware watchdog reset delay counter register (HWDG_RSTDLY) are inserted. 8. After the elapse of the time corresponding to the number of cycles for the delay time, the watchdog reset request or watchdog interrupt request (NMI) is generated according to the value of the RSTEN bit in the Hardware watchdog interrupt configuration register (HWDG_INT). When the watchdog interrupt request (NMI) is generated, the processing transitions to the operation of the test function of the watchdog. For details, see Figure 3-6. Correct) 1. The watchdog counter in RUN always operates. 2. When the device state transitions from RUN to PSS, the processing transitions to the operation of the watchdog in PSS. For details, see Figure 3-5. 3. During the operation of the watchdog counter, execute the watchdog counter clear protection trigger sequence within the range between the window upper limit value and lower limit value that have been specified beforehand. Do this to clear the watchdog counter periodically. 4. If a runaway state caused by a user program occurs and prevents periodic clearing, the watchdog counter reaches the window upper limit value. Then, processing transitions to the flow that generates the watchdog reset request or watchdog interrupt request (NMI). 5. According to the value of the IRQEN bit in the Hardware watchdog interrupt configuration register (HWDG_INT), the prior warning interrupt request is generated. At the same time, as many cycles as are set for the delay time in the Hardware watchdog reset delay counter register (HWDG_RSTDLY) are inserted. 6. After the elapse of the time corresponding to the number of cycles for the delay time, the watchdog reset request or watchdog interrupt request (NMI) is generated according to the value of the RSTEN bit in the Hardware watchdog interrupt configuration register (HWDG_INT). When the watchdog interrupt request (NMI) is generated, the processing transitions to the operation of the test function of the watchdog. For details, see Figure 3-6.
916	CHAPTER 14:Hardware Watchdog Timer 3. Explanation of Operation	Error) **Note:** The notation * in The following figure shows the operation of the test function of the Hardware watchdog. Correct) ***Note:*** *The notation * in Figure 3-6 means "RUN" or "PSS".*

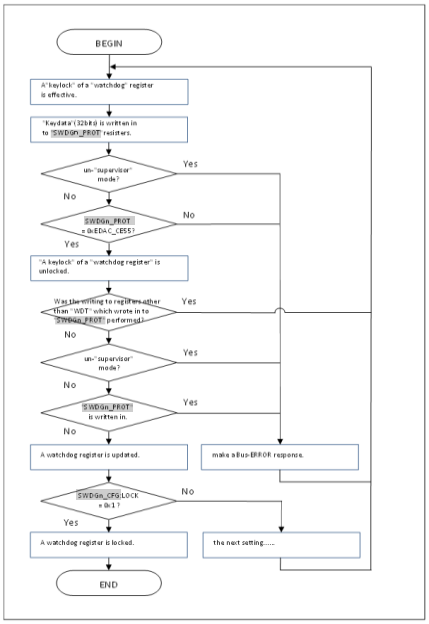
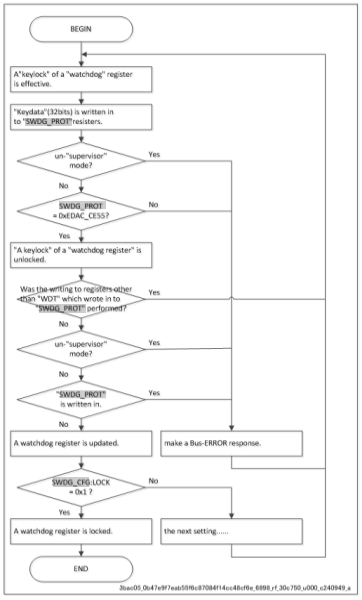
Page	Section	Change Results
917 to 918	CHAPTER 14:Hardware Watchdog Timer 4. Setting Procedure Example	Error) **Note:** In Figure 4-1, the setting order can be changed except for the step of setting the LOCK bit in the Hardware watchdog configuration register (HWDG_CFG) to "1". Correct) ***Note:*** − *In Figure 4-1, the setting order can be changed except for the step of setting the LOCK bit in the Hardware watchdog configuration register (HWDG_CFG) to "1".* − *Hardware watchdog timer is configured by boot program, and this is determined by value of BootROM marker. For detail of boot program and BootROM marker, see BootROM Software Interface chapter.* *Therefore, user program has to execute only "write 0x0 to HWDG_RSTCAUSE".*
942	CHAPTER 14:Hardware Watchdog Timer 6.Register List	Error) **[bit20:16] OBSSEL[4:0]: Watchdog Counter Monitor Bit Output Selection Bits** These bits are used for selecting any 1 bit in the watchdog counter (32 bits) as the output of the watchdog counter monitor bit.(Support target of this function is different depends on product specification. For details, see "3 Explanation of Operation For this product, output of the watchdog counter monitor bit of MCU output pin is supported by the Hardware watchdog timer.") Correct) **[bit20:16] OBSSEL[4:0]: Watchdog Counter Monitor Bit Output Selection Bits** These bits are used for selecting any 1 bit in the watchdog counter (32 bits) as the output of the watchdog counter monitor bit.(Support target of this function is different depends on product specification. For details, see "Output of the Watchdog Counter Monitor Bit for MCU Output Pin" in 3 Explanation of Operation.)
948	CHAPTER 14:Hardware Watchdog Timer 7. Precautions for Using This Device	Error) **Switching of Watchdog Counter Source Clock** 7. Writing 0xAB to the RUN profile update trigger register of the system controller (SYSC_TRGRUNCNTR) executes RUN profile update. At this point in time, the source clock of the watchdog counter is switched to the new source clock selected by the CLKSEL bit of the Hardware watchdog configuration register (HWDG_CFG). Correct) **Switching of Watchdog Counter Source Clock** 7. Writing 0xAB to the RUN profile update trigger register of the system controller (SYSC0_TRGRUNCNTR) executes RUN profile update. At this point in time, the source clock of the watchdog counter is switched to the new source clock selected by the CLKSEL bit of the Hardware watchdog configuration register (HWDG_CFG).

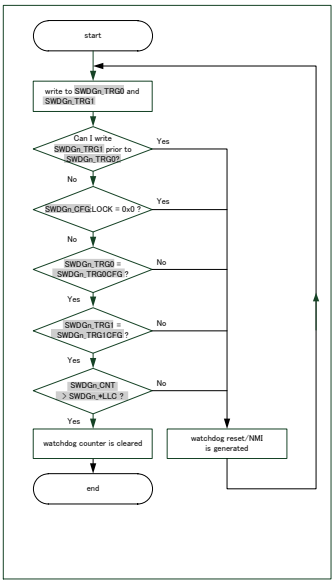
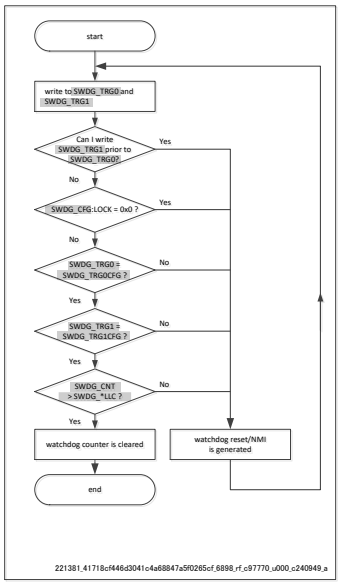
Page	Section	Change Results
949	CHAPTER 14:Hardware Watchdog Timer 7. Precautions for Using This Device	Error) **Watchdog Timer Operation in Standby State of Processor** ■ Without a valid key (0xBA or 0xBB) being written to the PSS profile update enable register (SYSC_PSSENR.PSSEN), the processor transitions to the standby state by executing a wait for interrupt (WFI) instruction. Correct) **Watchdog Timer Operation in Standby State of Processor** ■ Without a valid key being written to the PSS profile update enable register (SYSC0_PSSENR.PSSEN0 and SYSC1_PSSENR.PSSEN1), the processor transitions to the standby state by executing a wait for interrupt (WFI) instruction.

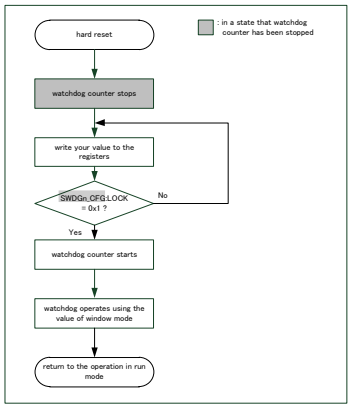
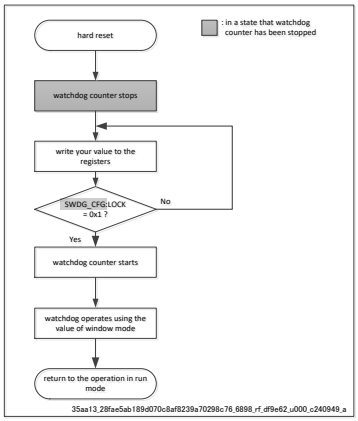
Page	Section	Change Results
952	CHAPTER 15:Software Watchdog Timer 1. Overview	Error) The Software watchdog timer is positioned in the SYS1 group, used for detecting a runaway state caused by a user program, also the Software watchdog timer monitors user program by the CPU. Correct) The Software watchdog timer is positioned in the SYSC1 group, used for detecting a runaway state caused by a user program, also the Software watchdog timer monitors user program by the CPU.
952	CHAPTER 15:Software Watchdog Timer 1. Overview	Error) ■ It provides the function for outputting the watchdog counter monitor bit for the MCU output pin so that the watchdog timer counter can be observed from the outside. (Support target of this function is different depends on product specification. For details, see "3 Explanation of Operation For this product, output of the watchdog counter monitor bit of MCU output pin is supported by the Software watchdog timer".) Correct) ■ It provides the function for outputting the watchdog counter monitor bit for the MCU output pin so that the watchdog timer counter can be observed from the outside. (Support target of this function is different depends on product specification. For details, see “Output of the Watchdog Counter Monitor Bit for MCU Output Pin” in 3 Explanation of Operation.

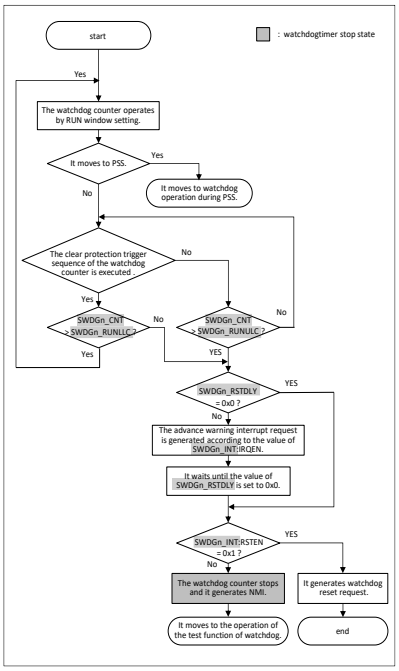
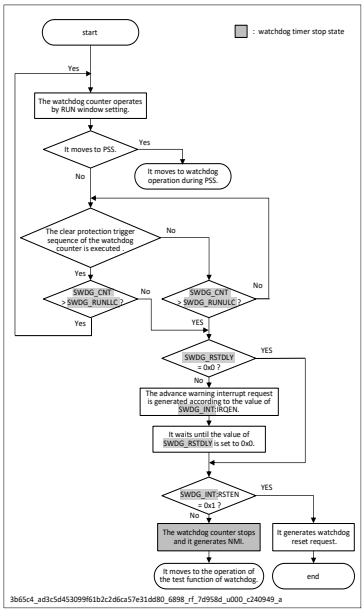
Page	Section	Change Results
954	CHAPTER 15:Software Watchdog Timer 3. Explanation of Operation	Error) **Software Watchdog Timer Start by User Program** A user program starts the software watchdog timer. Specifically, operation is started by setting the LOCK bit in the software watchdog configuration register (SWDGn_CFG). **Watchdog Counter Source Clock Selection (4 Types)** The software watchdog timer sets values in the CLKSEL[1:0] bits in the software watchdog configuration register (SWDGn_CFG) to select a source clock for the watchdog counter from among 4 types of source clocks. Specifically, it selects a source clock from the high-speed CR clock, low-speed CR clock, sub clock, or main clock. (The initial setting is for the high-speed CR clock.) Correct) **Software Watchdog Timer Start by User Program** A user program starts the software watchdog timer. Specifically, operation is started by setting the LOCK bit in the software watchdog configuration register (SWDG_CFG). **Watchdog Counter Source Clock Selection (4 Types)** The software watchdog timer sets values in the CLKSEL[1:0] bits in the software watchdog configuration register (SWDG_CFG) to select a source clock for the watchdog counter from among 4 types of source clocks. Specifically, it selects a source clock from the high-speed CR clock, low-speed CR clock, sub clock, or main clock. (The initial setting is for the high-speed CR clock.)
958	CHAPTER 15:Software Watchdog Timer 3. Explanation of Operation	Error) **Notes:** □ Software watchdog interrupt configuration register (SWDGn_INT) □ Software watchdog trigger 0 configuration register (SWDGn_TRG0CFG) □ Software watchdog trigger 1 configuration register (SWDGn_TRG1CFG) □ Software watchdog reset delay counter register (SWDGn_RSTDLY) □ Software watchdog configuration register (SWDGn_CFG) Correct) ***Notes:*** − Software watchdog interrupt configuration register (SWDG_INT) − Software watchdog trigger 0 configuration register (SWDG_TRG0CFG) − Software watchdog trigger 1 configuration register (SWDG_TRG1CFG) − Software watchdog reset delay counter register (SWDG_RSTDLY) − Software watchdog configuration register (SWDG_CFG)

Page	Section	Change Results
960	CHAPTER 15:Software Watchdog Timer 3. Explanation of Operation	Error) **Generation of Watchdog Reset Request or Watchdog Interrupt Request (NMI)** **Notes:** ■ When you generate the prior warning interrupt request, set the Software watchdog reset delay counter register (SWDG_RSTDLY) to a value other than 0x0000. The use of the watchdog interrupt request (NMI) is prohibited except when used as a test function. (This is prohibited because if such use is allowed, security will be compromised during the execution of an application.) The following figure shows the operation of the test function of the Hardware watchdog. Correct) **Generation of Watchdog Reset Request or Watchdog Interrupt Request (NMI)** ***Notes:*** — *When you generate the prior warning interrupt request, set the Software watchdog reset delay counter register (SWDG_RSTDLY) to a value other than 0x0000.* — *The use of the watchdog interrupt request (NMI) is prohibited except when used as a test function. (This is prohibited because if such use is allowed, security will be compromised during the execution of an application.) Figure 3-6 shows the operation of the test function of the Software watchdog.*
960	CHAPTER 15:Software Watchdog Timer 3. Explanation of Operation	Error) **Output of the Watchdog Counter Monitor Bit for MCU Output Pin** For This Product, Output of the Watchdog Counter Monitor Bit of MCU Output Pin is Supported by the Software Watchdog Timer. The Software watchdog timer outputs the watchdog counter monitor bit to the outside so that watchdog counter operation can be monitored from the outside. You can select any 1 bit in the watchdog counter (32 bits) by setting the OBSSEL[4:0] bits in the Software watchdog configuration register (SWDG_CFG). Correct) **Output of the Watchdog Counter Monitor Bit for MCU Output Pin** The Software watchdog timer outputs the watchdog counter monitor bit to the outside so that watchdog counter operation can be monitored from the outside. You can select any 1 bit in the watchdog counter (32 bits) by setting the OBSSEL[4:0] bits in the Software watchdog configuration register (SWDG_CFG). Support of this function is different depends on product specification.

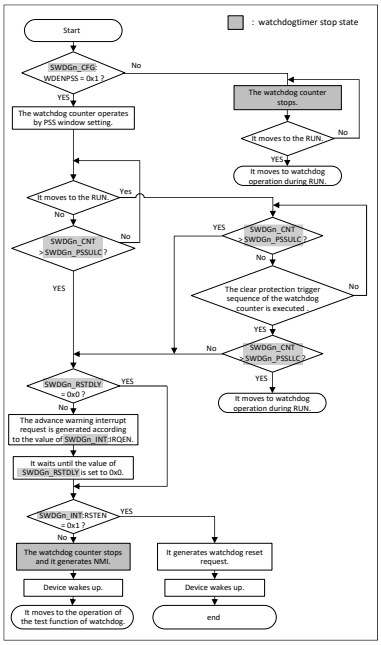
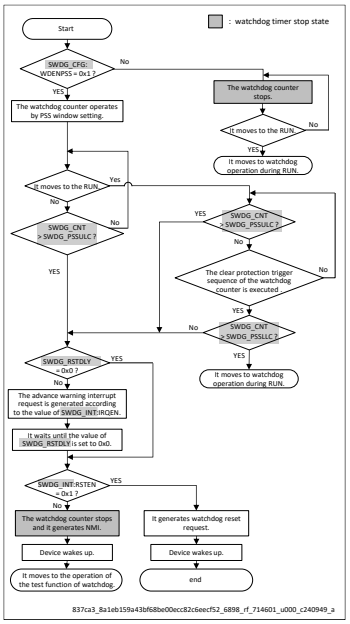
Page	Section	Change Results
964	CHAPTER 15:Software Watchdog Timer 3. Explanation of Operation	Error) Figure 15.3-1 Setting Procedure for Watchdog Register Write Protection Sequence  Correct) Figure 3-1 Setting Procedure for Watchdog Register Write Protection Sequence 

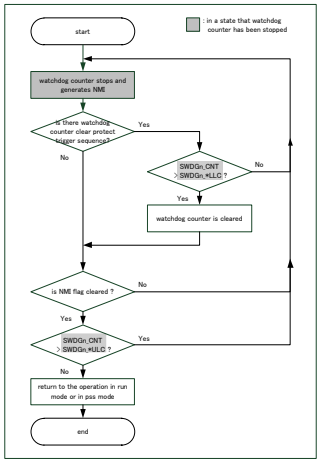
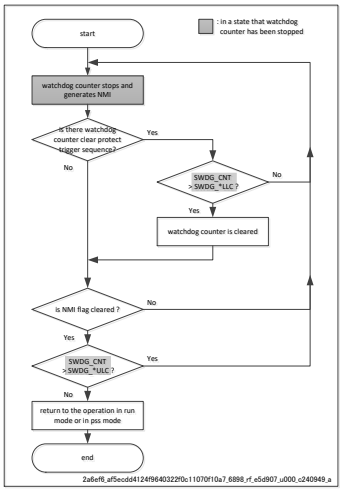
Page	Section	Change Results
965	CHAPTER 15:Software Watchdog Timer 3. Explanation of Operation	Error) Figure 15.3-2 Setting Procedure for Watchdog Counter Clear Protection Trigger Sequence  Correct) Figure 3-2 Setting Procedure for Watchdog Counter Clear Protection Trigger Sequence 

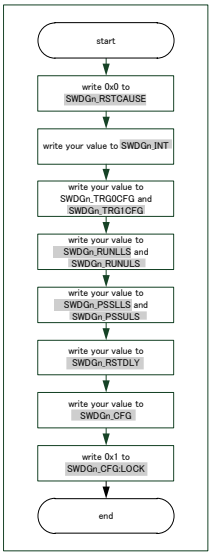
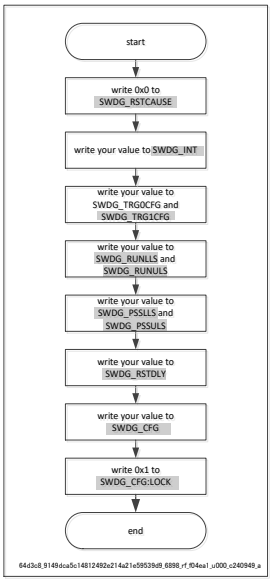
Page	Section	Change Results
966	CHAPTER 15:Software Watchdog Timer 3. Explanation of Operation	Error) Figure 15.3-3 Startup of Software Watchdog Timer  1. The hard reset initializes the software watchdog timer. Then, immediately after the hard reset is cleared, the stop state is indicated. 2. You can change the register setting as long as the LOCK bit in the software watchdog configuration register (SWDGn_CFG) is "0". 3. Setting "1" in the LOCK bit in the software watchdog configuration register (SWDGn_CFG) causes the watchdog counter to start up-counting. Correct) Figure 3-3 Startup of Software Watchdog Timer  1. The hard reset initializes the software watchdog timer. Then, immediately after the hard reset is cleared, the stop state is indicated. 2. You can change the register setting as long as the LOCK bit in the software watchdog configuration register (SWDG_CFG) is "0". 3. Setting "1" in the LOCK bit in the software watchdog configuration register (SWDG_CFG) causes the watchdog counter to start up-counting.

Page	Section	Change Results
967	CHAPTER 15:Software Watchdog Timer 3. Explanation of Operation	Error) Figure 15.3-4 Software Watchdog Operation in RUN  Correct) Figure 3-4 Software Watchdog Operation in RUN 

Page	Section	Change Results
968	CHAPTER 15:Software Watchdog Timer 3. Explanation of Operation	Error) The watchdog counter in RUN always operates. When the device state transitions from RUN to PSS, the processing transitions to the operation of the watchdog in PSS. For details, see The following figure shows the Software watchdog operation in PSS. During the operation of the watchdog counter, execute the watchdog counter clear protection trigger sequence within the range between the window upper limit value and lower limit value that have been specified beforehand. Do this to clear the watchdog counter periodically. 5. If a runaway state of the software occurs and prevents periodic clearing, the watchdog counter reaches the window upper limit value. Then, processing transitions to the flow that generates the watchdog reset request or watchdog interrupt request (NMI). 6. According to the value of the IRQEN bit in the Software watchdog interrupt configuration register (SWDG_INT), the prior warning interrupt request is generated. At the same time, as many cycles as are set for the delay time in the Software watchdog reset delay counter register (SWDG_RSTDLY) are inserted. 7. After the elapse of the time corresponding to the number of cycles for the delay time, the watchdog reset request or watchdog interrupt request (NMI) is generated according to the value of the RSTEN bit in the Software watchdog interrupt configuration register (SWDG_INT). When the watchdog interrupt request (NMI) is generated, the processing transitions to the operation of the test function of the watchdog. For details, see Figure 3-6. 8. Correct) 1. The watchdog counter in RUN always operates. 2. When the device state transitions from RUN to PSS, the processing transitions to the operation of the watchdog in PSS. For details, see Figure 3-5. 3. During the operation of the watchdog counter, execute the watchdog counter clear protection trigger sequence within the range between the window upper limit value and lower limit value that have been specified beforehand. Do this to clear the watchdog counter periodically. 4. If a runaway state of the software occurs and prevents periodic clearing, the watchdog counter reaches the window upper limit value. Then, processing transitions to the flow that generates the watchdog reset request or watchdog interrupt request (NMI). 5. According to the value of the IRQEN bit in the Software watchdog interrupt configuration register (SWDG_INT), the prior warning interrupt request is generated. At the same time, as many cycles as are set for the delay time in the Software watchdog reset delay counter register (SWDG_RSTDLY) are inserted. 6. After the elapse of the time corresponding to the number of cycles for the delay time, the watchdog reset request or watchdog interrupt request (NMI) is generated according to the value of the RSTEN bit in the Software watchdog interrupt configuration register (SWDG_INT). When the watchdog interrupt request (NMI) is generated, the processing transitions to the operation of the test function of the watchdog. For details, see Figure 3-6.

Page	Section	Change Results
969	CHAPTER 15:Software Watchdog Timer 3. Explanation of Operation	Error) Figure 15.3-5 Software Watchdog Operation in PSS  Correct) Figure 3-5 Software Watchdog Operation in PSS 

Page	Section	Change Results
971	CHAPTER 15:Software Watchdog Timer 3. Explanation of Operation	Error) Figure 15.3-6 Operation of Software Watchdog Test Function  : (omit) Note: The notation * in The following figure shows the operation of the test function of the Software watchdog. Correct) Figure 3-6 Operation of Software Watchdog Test Function  : (omit) Note: The notation * in Figure 3-6 means "RUN" or "PSS".

Page	Section	Change Results
972	CHAPTER 15:Software Watchdog Timer 4. Setting Procedure Example	Error) Figure 15.4-1 Example of Software Watchdog Timer Register Setting Procedure  Correct) Figure 4-1 Example of Software Watchdog Timer Register Setting Procedure 

Page	Section	Change Results
973	CHAPTER 15:Software Watchdog Timer 5. Operation Examples	(Under Figure 5-1) Error) 1. Before starting the software watchdog timer, the setting values of the registers are written. 2. The LOCK bit in the software watchdog configuration register (SWDGn_CFG) is set to "1" to start the software watchdog timer. At this time, it starts up-counting from 0x00000000. 3. The watchdog counter clear protection trigger sequence is executed, and the watchdog counter is cleared. Correct) 1. Before starting the software watchdog timer, the setting values of the registers are written. 2. The LOCK bit in the software watchdog configuration register (SWDG_CFG) is set to "1" to start the software watchdog timer. At this time, it starts up-counting from 0x00000000. 3. The watchdog counter clear protection trigger sequence is executed, and the watchdog counter is cleared.
974	CHAPTER 15:Software Watchdog Timer 5. Operation Examples	(Under figure 5-2) Error) 1. Before starting the software watchdog timer, the setting values of the registers are written. 2. The LOCK bit in the software watchdog configuration register (SWDGn_CFG) is set to "1" to start the software watchdog timer. At this time, it starts up-counting from 0x00000000. 3. The watchdog counter clear protection trigger sequence is executed, and the watchdog counter is cleared. Correct) 1. Before starting the software watchdog timer, the setting values of the registers are written. 2. The LOCK bit in the software watchdog configuration register (SWDG_CFG) is set to "1" to start the software watchdog timer. At this time, it starts up-counting from 0x00000000. 3. The watchdog counter clear protection trigger sequence is executed, and the watchdog counter is cleared.
978	CHAPTER 15:Software Watchdog Timer 6. Register List 6.1. Software Watchdog Protection Register (SWDTG_PROT)	Error) 6.1. Software Watchdog Protection Register (SWDT_PROT) Correct) 6.1. Software Watchdog Protection Register (SWDG_PROT)

Page	Section	Change Results
978	CHAPTER 15:Software Watchdog Timer 6. Register List 6.1. Software Watchdog Protection Register (SWDTG_PROT)	Error) **Notes:** ■ An AHB transfer error response is generated when write access to ■ A protect key will be locked again by writing to the address where is in the same group area(SYS1 Group), however protect key will not be locked by writing to no protect target register. Correct) ***Notes:*** — *An AHB transfer error response is generated when write access to* — *A protect key will be locked again by writing to the address where is in the same group area(SYSC1 Group), however protect key will not be locked by writing to no protect target register.*
995	CHAPTER 15:Software Watchdog Timer 6. Register List 6.15. Software Watchdog Configuration Register (SWDG_CFG)	Error) **[bit20:16] OBSSEL[4:0]: Watchdog Counter Monitor Bit Output Selection Bits** These bits are used for selecting any 1 bit in the watchdog counter (32 bits) as the output of the watchdog counter monitor bit.(Support target of this function is different depends on product specification. For details, see "3 Explanation of Operation For this product, output of the watchdog counter monitor bit of MCU output pin is supported by the Software watchdog timer.") Correct) **[bit20:16] OBSSEL[4:0]: Watchdog Counter Monitor Bit Output Selection Bits** These bits are used for selecting any 1 bit in the watchdog counter (32 bits) as the output of the watchdog counter monitor bit.(Support target of this function is different depends on product specification. For details, see “Output of the Watchdog Counter Monitor Bit for MCU Output Pin” in 3 Explanation of Operation.
997	CHAPTER 15:Software Watchdog Timer 6. Register List 6.16. Software Watchdog Lower Limit RUN Current Register (SWDG_RUNLLC)	Error) You can change the set value of this register even after the LOCK bit in the Software watchdog configuration register (SWDG_CFG) is set. In this case, this register fetches the set value of the software watchdog lower limit RUN setting register (SWDGn_RUNLLS) when the watchdog counter clear protection sequence is executed. Correct) You can change the set value of this register even after the LOCK bit in the Software watchdog configuration register (SWDG_CFG) is set. In this case, this register fetches the set value of the software watchdog lower limit RUN setting register (SWDG_RUNLLS) when the watchdog counter clear protection sequence is executed.

Page	Section	Change Results
998	CHAPTER 15:Software Watchdog Timer 6. Register List 6.17. Software Watchdog Upper Limit RUN Current Register (SWDG_RUNULC)	Error) You can change the set value of this register even after the LOCK bit in the Software watchdog configuration register (SWDG_CFG) is set. In this case, this register fetches the set value of the software watchdog upper limit RUN setting register (SWDGn_RUNULS) when the watchdog counter clear protection trigger sequence is executed. Correct) You can change the set value of this register even after the LOCK bit in the Software watchdog configuration register (SWDG_CFG) is set. In this case, this register fetches the set value of the software watchdog upper limit RUN setting register (SWDG_RUNULS) when the watchdog counter clear protection trigger sequence is executed.
999	CHAPTER 15:Software Watchdog Timer 6. Register List 6.18. Software Watchdog Lower Limit PSS Current Register (SWDG_PSSLLC)	Error) You can change the set value of this register even after the LOCK bit in the Software watchdog configuration register (SWDG_CFG) is set. In this case, this register fetches the set value of the software watchdog lower limit PSS setting register (SWDGn_PSSLLS) when the watchdog counter clear protection trigger sequence is executed. Correct) You can change the set value of this register even after the LOCK bit in the Software watchdog configuration register (SWDG_CFG) is set. In this case, this register fetches the set value of the software watchdog lower limit PSS setting register (SWDG_PSSLLS) when the watchdog counter clear protection trigger sequence is executed.
1000	CHAPTER 15:Software Watchdog Timer 6. Register List 6.19. Software Watchdog Upper Limit PSS Current Register (SWDG_PSSULC)	Error) You can change the set value of this register even after the LOCK bit in the Software watchdog configuration register (SWDG_CFG) is set. In this case, this register fetches the set value of the software watchdog upper limit PSS setting register (SWDGn_PSSULS) when the watchdog counter clear protection trigger sequence is executed. Correct) You can change the set value of this register even after the LOCK bit in the Software watchdog configuration register (SWDG_CFG) is set. In this case, this register fetches the set value of the software watchdog upper limit PSS setting register (SWDG_PSSULS) when the watchdog counter clear protection trigger sequence is executed.

Page	Section	Change Results
1001	CHAPTER 15:Software Watchdog Timer 7. Precautions for Using This Device	Error) **Switching of Watchdog Counter Source Clock** The system controller controls the switching of the watchdog counter source clock since the source clock control of the watchdog counter is a part of the system setting information (profile). To change the watchdog counter source clock, follow the sequence below. 1.Writing data to the RUN profile source clock enable register of the system controller (SYSC_RUNCKSRER) enables oscillation of the new source clock of the watchdog. Then, writing 0xAB to the RUN profile update trigger register of the system controller (SYSC_TRGRUNCNTR) executes RUN profile update. 8.Writing 0xAB to the RUN profile update trigger register of the system controller (SYSC_TRGRUNCNTR) executes RUN profile update. At this point in time, the source clock of the watchdog counter is switched to the new source clock selected by the CLKSEL bit of the Software watchdog configuration register (SWDG_CFG). Correct) **Switching of Watchdog Counter Source Clock** The system controller controls the switching of the watchdog counter source clock since the source clock control of the watchdog counter is a part of the system setting information (profile). To change the watchdog counter source clock, follow the sequence below. 1.Writing data to the RUN profile source clock enable register of the system controller (SYSC0_RUNCKSRER) enables oscillation of the new source clock of the watchdog. Then, writing 0xAB to the RUN profile update trigger register of the system controller (SYSC0_TRGRUNCNTR) executes RUN profile update. 8.Writing 0xAB to the RUN profile update trigger register of the system controller (SYSC0_TRGRUNCNTR) executes RUN profile update. At this point in time, the source clock of the watchdog counter is switched to the new source clock selected by the CLKSEL bit of the Software watchdog configuration register (SWDG_CFG).
1002	CHAPTER 15:Software Watchdog Timer 7. Precautions for Using This Device	Error) **Watchdog Timer Operation in Standby State of Processor** The Software watchdog timer operates using the RUN window setting in the following case. ■ Without a valid key (0xBA or 0xBB) being written to the PSS profile update enable register (SYSC_PSSENR.PSSEN), the processor transitions to the standby state by executing a wait for interrupt(WFI) instruction. Correct) **Watchdog Timer Operation in Standby State of Processor** The Software watchdog timer operates using the RUN window setting in the following case. ■ Without a valid key being written to the PSS profile update enable register (SYSC0_PSSENR.PSSEN0 and SYSC1_PSSENR.PSSEN1), the processor transitions to the standby state by executing a wait for interrupt(WFI) instruction.

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1009</td><td>CHAPTER 16:TCRAM Interface
3. Explanation of Operation</td><td>Error)
■ When TRCFGn_ECCDEN:DEN is "1", 0b10110101 is written in bit[31:24] in the TRCFGn_ECCDWEN register (DWEN, DWENUNLOCK) in a state in which privilege access is possible.

Correct)
■ When TRCFGn_ECCDEN:DEN is "1", 0b10110101 is written in bit[31:24] in the TRCFGn_ECCDW register (DWEN, DWENUNLOCK) in a state in which privilege access is possible.</td></tr>
<tr><td>1009</td><td>CHAPTER 16:TCRAM Interface
3. Explanation of Operation</td><td>Error)
■ A value that is not 0b10110101 is written in bit[31:24] in the TRCFGn_ECCDWEN register (DWEN, DWENUNLOCK) in a state in which privilege access is possible.

Correct)
■ A value that is not 0b10110101 is written in bit[31:24] in the TRCFGn_ECCDW register (DWEN, DWENUNLOCK) in a state in which privilege access is possible.</td></tr>
<tr><td>1040</td><td>CHAPTER 16:TCRAM Interface
5. Registers
5.16. TCRAM IF TEST Diagnosis Function Register (TRCFGn_TTCR)</td><td>Error)
[bit7] TEIE: Enable Error Occurrence Interrupt during Diagnosis<table><tr><th>TEIE</th><th>Diagnosis-Time Error Generation Interrupt Enable Bit</th></tr><tr><td>During Write Operation</td><td>0: Disable interrupts triggered by RAM diagnosis errors.
1: Enable interrupts triggered by RAM diagnosis errors. When TTCR:TEI is "1", "H" is output to the interrupt signal.</td></tr><tr><td>During Read Operation</td><td>You can read the setting for enabling or disabling the error interrupts during diagnosis.</td></tr></table>
Correct)
[bit7] TEIE: Enable Error Occurrence Interrupt during Diagnosis<table><tr><th>TEIE</th><th>Diagnosis-Time Error Generation Interrupt Enable Bit</th></tr><tr><td>During Write Operation</td><td>0: Disable interrupts triggered by RAM diagnosis errors.
1: Enable interrupts triggered by RAM diagnosis errors. When TRCFGn_TTCR:TEI is "1", "H" is output to the interrupt signal.</td></tr><tr><td>During Read Operation</td><td>You can read the setting for enabling or disabling the error interrupts during diagnosis.</td></tr></table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1041</td><td>CHAPTER 16:TCRAM Interface
5. Registers
5.16. TCRAM IF TEST Diagnosis Function Register (TRCFGn_TTCR)</td><td>Error)
[bit5] TCIE: Diagnosis End Source Interrupt Enable
<table><tr><th>TCIE</th><th>Diagnosis End Source Interrupt Enable Bit</th></tr><tr><td>During Write Operation</td><td>0: Disable interrupts triggered by RAM diagnosis end.
1: Enable interrupts triggered by RAM diagnosis end. When TTCR:TCI is "1", "H" is output to the interrupt signal.</td></tr><tr><td>During Read Operation</td><td>You can read the setting for enabling or disabling the diagnosis end source interrupt.</td></tr></table>
Correct)
[bit5] TCIE: Diagnosis End Source Interrupt Enable
<table><tr><th>TCIE</th><th>Diagnosis End Source Interrupt Enable Bit</th></tr><tr><td>During Write Operation</td><td>0: Disable interrupts triggered by RAM diagnosis end.
1: Enable interrupts triggered by RAM diagnosis end. When TRCFGn_TTCR:TCI is "1", "H" is output to the interrupt signal.</td></tr><tr><td>During Read Operation</td><td>You can read the setting for enabling or disabling the diagnosis end source interrupt.</td></tr></table></td></tr>
<tr><td>1044</td><td>CHAPTER 16:TCRAM Interface
5. Registers
5.18. TCRAM IF TEST Key Code Control Register (TRCFGn_TKCCR)</td><td>Add

Correct)
[bit7:6] KEY: Key Code Control
<table><tr><th>KEY</th><th>Key Code Control Bits</th></tr><tr><td>During Write Operation</td><td>These bits control the key code.
The operation specified in the CODE bits is performed by writing in the order of 0b00, 0b01, 0b10, and 0b11.
In addition, if you have accessed another RAMTEST (TRCFGn_TEAR0, TRCFGn_TEAR1, TRCFGn_TEAR2, TRCFGn_TAEAR, TRCFGn_TASAR, TRCFGn_TFECR, TRCFGn_TICR, TRCFGn_TTCR, TRCFGn_TSRCR, TRCFGn_TSCR) register or have performed an operation other than that described above (read operation or a series of write operations other than the above) during key code operation, start the operation over again.
For the detailed setting procedure of these bits, see Figure 4-4.</td></tr><tr><td>During Read Operation</td><td>"0" is always read.</td></tr></table></td></tr>
</table>

Page	Section	Change Results
1048	CHAPTER 17 TCFLASH 1.Overview	Error) ■ Bus error response is sent when access is to the reserved area. Correct) ■ Bus error response is sent when access is to the reserved area or detect 1-bit ECC error via TCM region(*1). *1:Partially ECC supported. For more detail, See " CHAPTER17, 6. Others, 1-bit ECC error is detected".
1048, 1049	CHAPTER 17 TCFLASH 1.Overview	Add Correct) ■ ECC support for the AXI ports for FLASH 1-bit error correction and 2-bit error detection (SEC-DED). ■ ECC support for the TCM ports for FLASH 1-bit error detection(*1) and 2-bit error detection (SED-DED) at default setting. *1 For more detail, See "CHAPTER17, 6. Others, 1-bit ECCerror is detected"
1070	CHAPTER 17 TCFLASH 3.Explanation of Operation 3.9. Interrupt	Error) TCFLASH can issue an interrupt request in the following cases. ■ When a program or erase operation is completed and flash memory is ready to start the next operation The RDYINT bit in the TCFCFG_FSTAT register reads "1". In this case, if the RDYIE bit of the TCFCFG_FICTRL register is "1", TCFLASH generates an interrupt request. ■ When a hang-up state is detected When a timeout is detected during the execution of an automatic algorithm, etc., the hang-up state is detected, causing the HANGINT bit in the TCFCFG_FSTAT register to show "1". In this case, if the HANGIE bit of the TCFCFG_FICTRL register is "1", TCFLASH generates an interrupt request. Correct) TCFLASH can generate an interrupt request in the following case.

Page	Section	Change Results
1088	CHAPTER 17 TCFLASH 5.Register 5.1.TCFLASH Configuration Protection Key Register: TCFCFG_FCPROTKEY (TCFLASH Configuration Protection Key Register)	Error) TCFLASH ECC control register (TCFFCFG_FECCCTRL) Correct) TCFLASH ECC control register (TCFCFG_FECCCTRL)
1095	CHAPTER 17 TCFLASH 5.Register 5.6. TCFLASH Interrupt Control Register: TCFCFG_FICTRL (TCFLASH Interrupt Control Register)	Add Correct) This register is not supported. Always write "0" to RDYIE and HANGIE bit.

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1095</td><td>CHAPTER 17 TCFLASH 5.Register 5.6. TCFLASH Interrupt Control Register: TCFCFG_FICTRL (TCFLASH Interrupt Control Register)</td><td>Error)
<table>
<tr><td>bit</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td>Field</td><td colspan="6">Reserved</td><td>HANGIC</td><td>RDYIC</td></tr>
<tr><td>R/W Attribute</td><td colspan="6">R0, WX</td><td>R0, W</td><td>R0, W</td></tr>
<tr><td>Protection Attribute</td><td colspan="8">WP</td></tr>
<tr><td>Initial Value</td><td colspan="6">000000</td><td>0</td><td>0</td></tr>
</table>
<table>
<tr><td>bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Field</td><td colspan="6">Reserved</td><td>HANGIE</td><td>RDYIE</td></tr>
<tr><td>R/W Attribute</td><td colspan="6">R0, WX</td><td>R/W</td><td>R/W</td></tr>
<tr><td>Protection Attribute</td><td colspan="8">WP</td></tr>
<tr><td>Initial Value</td><td colspan="6">000000</td><td>0</td><td>0</td></tr>
</table>
Correct)
<table>
<tr><td>bit</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td>Field</td><td colspan="6">Reserved</td><td>HANGIC</td><td>RDYIC</td></tr>
<tr><td>R/W Attribute</td><td colspan="6">R0, WX</td><td>R0, W0</td><td>R0, W0</td></tr>
<tr><td>Protection Attribute</td><td colspan="8">WP</td></tr>
<tr><td>Initial Value</td><td colspan="6">000000</td><td>0</td><td>0</td></tr>
</table>
<table>
<tr><td>bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Field</td><td colspan="6">Reserved</td><td>HANGIE</td><td>RDYIE</td></tr>
<tr><td>R/W Attribute</td><td colspan="6">R0, WX</td><td>R/W0</td><td>R/W0</td></tr>
<tr><td>Protection Attribute</td><td colspan="8">WP</td></tr>
<tr><td>Initial Value</td><td colspan="6">000000</td><td>0</td><td>0</td></tr>
</table></td></tr>
<tr><td>1095</td><td>CHAPTER 17 TCFLASH 5.Register 5.6. TCFLASH Interrupt Control Register: TCFCFG_FICTRL (TCFLASH Interrupt Control Register)</td><td>Add
Correct)
[bit 9] HANGIC: Hang Interrupt Clear
<table>
<tr><th>HANGIC</th><th>Description</th></tr>
<tr><td>0</td><td>Writing "0" to this bit has no meaning.</td></tr>
<tr><td>1</td><td>Writing "1" to this bit clears the HANGINT bit in the TCFCFG_FSTAT register.</td></tr>
</table>
Note:
− Do not use this function.</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1095</td><td>CHAPTER 17 TCFLASH 5.Register 5.6. TCFLASH Interrupt Control Register: TCFCFG_FICTRL (TCFLASH Interrupt Control Register)</td><td>Add

Correct)
[bit 8]RDYIC: Ready Interrupt Clear
<table><tr><th>RDYIC</th><th>Description</th></tr><tr><td>0</td><td>Writing "0" to this bit has no meaning.</td></tr><tr><td>1</td><td>Writing "1" to this bit clears the RDYINT bit in the TCFCFG_FSTAT register.</td></tr></table>Note:
− Do not use this function.</td></tr>
<tr><td>1096</td><td>CHAPTER 17 TCFLASH 5.Register 5.6. TCFLASH Interrupt Control Register: TCFCFG_FICTRL (TCFLASH Interrupt Control Register)</td><td>Add

Correct)
[bit 1] HANGIE: Hang Interrupt Enable
<table><tr><th>HANGIE</th><th>Description</th></tr><tr><td>0</td><td>Disable the generation of hang interrupt requests.</td></tr><tr><td>1</td><td>Enable the generation of hang interrupt requests.</td></tr></table>Note:
− Do not use this function.</td></tr>
<tr><td>1096</td><td>CHAPTER 17 TCFLASH 5.Register 5.6. TCFLASH Interrupt Control Register: TCFCFG_FICTRL (TCFLASH Interrupt Control Register)</td><td>Add

Correct)
[bit 0] RDYIE: Programming/Erasing Ready Interrupt Enable
<table><tr><th>RDYIE</th><th>Description</th></tr><tr><td>0</td><td>Disable the generation of programming/erasing ready interrupt requests.</td></tr><tr><td>1</td><td>Enable the generation of programming/erasing ready interrupt requests.</td></tr></table>Note:
− Do not use this function.</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1097</td><td>CHAPTER 17 TCFLASH 5.Register 5.7. TCFLASH Status Register: TCFCFG_FSTAT (TCFLASH Status Register)</td><td>Add

Correct)
[bit 9] HANGINT: Hang Up Interrupt
This bit indicates that a hung up interrupt request is generated because transition of the flash memory to the hung up 1 state is detected.
This bit is cleared by writing "1" to the HANGIC bit in the TCFCFG_FICTRL register.
<table><tr><th>HANGINT</th><th>Description</th></tr><tr><td>0</td><td>No hang up interrupt request is generated.</td></tr><tr><td>1</td><td>A hang up interrupt request is generated.</td></tr></table>
Note:
− Do not use this function. Use polling instead of interrupt. Use HANG(bit-1) bit.</td></tr>
<tr><td>1097</td><td>CHAPTER 17 TCFLASH 5.Register 5.7. TCFLASH Status Register: TCFCFG_FSTAT (TCFLASH Status Register)</td><td>Add

Correct)
[bit 8] RDYINT: Ready INTerrupt
This bit indicates that a ready interrupt request has been generated because a change of the flash memory to the ready state has been detected. This bit is cleared by writing "1" to the RDYIC bit in the TCFCFG_FICTRL register for the same memory in the same unit.
<table><tr><th>RDYINT</th><th>Description</th></tr><tr><td>0</td><td>No programming/erasing ready interrupt request is generated.</td></tr><tr><td>1</td><td>A programming/erasing ready interrupt request is generated.</td></tr></table>
Notes:
− Initial value depends on the boot configuration.
− Do not use this function. Use polling instead of interrupt. Use RDY(bit-0) bit.</td></tr>
</table>

Page	Section	Change Results
1100	CHAPTER 17 TCFLASH 5.Register 5.8.TCFLASH SEC Interrupt Register: TCFCFG_FSECIR (TCFLASH SEC Interrupt Register)	Error) **[bit31:24] SYN: Syndrome** This bit holds syndrome when a 1-bit error is detected. If a 1-bit error is detected in both the upper 64 bits and lower 64 bits when 128-bit read access is performed, syndrome for the error detected in the lower 64 bits is stored. Use the value of bit3 in the TCFCG_FECCEAR register to determine whether the syndrome stored in this field is detected in the upper 64-bits or lower 64 bits. Correct) **[bit31:24] SYN: Syndrome** This bit holds syndrome when a 1-bit error is detected. If a 1-bit error is detected in both the upper 64 bits and lower 64 bits when 128-bit read access is performed, syndrome for the error detected in the lower 64 bits is stored. Use the value of bit3 in the TCFCFG_FECCEAR register to determine whether the syndrome stored in this field is detected in the upper 64-bits or lower 64 bits.
1103	CHAPTER 17 TCFLASH 5.Register 5.10 TCFLASH Uncorrectable Error Detection Interrupt Register: TCFCFG_FUCEDIR (TCFLASH Un-Correctable Error Detection Interrupt Register)	Error) **[bit31:24] SYN: Syndrome** This bit holds syndrome when a 2-bit error is detected. If an uncorrectable error is detected in both the upper 64 bits and lower 64 bits when 128-bit read access is performed, syndrome for the error detected in the lower 64 bits is stored. Use the value of bit3 in the TCFCG_FUCEAR register to determine whether the stored syndrome is for an error that occurred in the upper 64 bits or an error that occurred in the lower 64 bits. Correct) **[bit31:24] SYN: Syndrome** This bit holds syndrome when a 2-bit error is detected. If an uncorrectable error is detected in both the upper 64 bits and lower 64 bits when 128-bit read access is performed, syndrome for the error detected in the lower 64 bits is stored. Use the value of bit3 in the TCFCFG_FUCEAR register to determine whether the stored syndrome is for an error that occurred in the upper 64 bits or an error that occurred in the lower 64 bits.

Page	Section	Change Results
1106	CHAPTER 17 TCFLASH 5.Register 5.12.TCFLASH TCM Buffer Region Configuration Register: TCFCFG_BRCFG (TCFLASH Buffer Region Configuration Register)	Error) **[bit31:16] REGION_END_BUF[15:0]: TCM Buffer Word Number of Region End** These bits set the TCM buffer word number of this region end. The first of this region end is set to "0". If REGION_END_BUF is set to "8", it uses up to TCMBUF0 to TCMBUF7. Correct) **[bit31:16] REGION_END_BUF[15:0]: TCM Buffer Word Number of Region End** These bits set the TCM buffer word number of this region end. The first of this region end is set to "0". If REGION_END_BUF is set to "8", it uses up to TCFCFG_TCMBUF0 to TCFCFG_TCMBUF7.
1109	CHAPTER 17 TCFLASH 5.Register 5.14.TCFLASH TCM Buffer Register: TCFCFG_TCMBUF0 to 15 (TCFLASH Buffer0 to 15)	Error) **[bit127:0] BUFDATA[127:0]: Data Value of TCM Buffer** These bits are data to be buffered. If TCF_BRAT::AM is in lock mode, these bits do not function unless they are intentionally set. It is necessary to rewrite when TCM is not read. In addition, do not set the same address to multiple buffers. Correct) **[bit127:0] BUFDATA[127:0]: Data Value of TCM Buffer** These bits are data to be buffered. If TCFCFG_BRAT:AM is in lock mode, these bits do not function unless they are intentionally set. It is necessary to rewrite when TCM is not read. In addition, do not set the same address to multiple buffers.
1111	CHAPTER 17 TCFLASH 6.Others	Error) **Wait for Reset Completion from Software** Flash memory is reset by writing "1" to the SWFRST bit in the TCFCFG_FCGCR register. If the flash memory is reset, be sure to wait for reset completion by monitoring the RDY bit in the TCFCFG_FSTAT register in the unit that executes reset before accessing the memory. Correct) **Wait for Reset Completion from Software** Flash memory is reset by writing "1" to the SWFRST bit in the TCFCFG_FCFGR register. If the flash memory is reset, be sure to wait for reset completion by monitoring the RDY bit in the TCFCFG_FSTAT register in the unit that executes reset before accessing the memory.

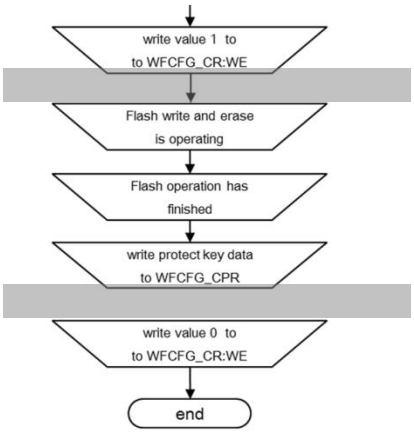
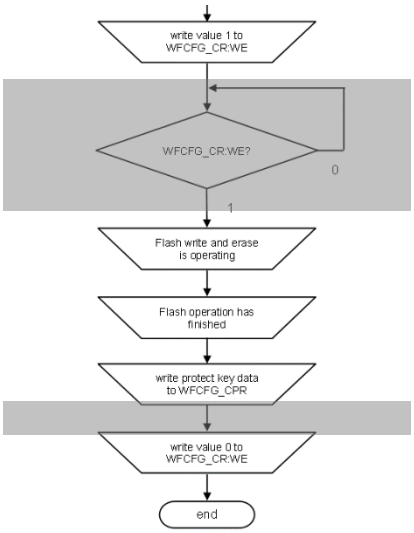
Page	Section	Change Results
1111	CHAPTER 17 TCFLASH 6.Others	Error) **Suspend Command** After the suspend command is issued, do not read flash memory until the RDY bit in the status register becomes "1". Correct) **Suspend Command** After the suspend command is issued, do not read flash memory until the RDY bit in the status register becomes "1". When 1-bit ECC error is detected When the CPU accesses the TCFLASH via TCM and detected 1-bit ECC error with the default setting, it tries to re-write the 1-bit-error-corrected data to the same address by the hardware. But TCFlash cannot be written. So it results to the data abort.

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1130</td><td>CHAPTER 18 WorkFLASH 3.Explanation of Operation 3.7.Bus Error Response</td><td>Error)
<table><tr><td>- Writing to the mirror area 4
(however, if the reserved area is written, RESA flag is set instead of this bit).</td><td>9</td><td>NWTM</td></tr></table>
Correct)
<table><tr><td>- Writing to the mirror area 4
(refer next table for detail condition when NWTM flag is set).</td><td>9</td><td>NWTM</td></tr></table></td></tr>
<tr><td>1130</td><td>CHAPTER 18 WorkFLASH 3.Explanation of Operation 3.7.Bus Error Response</td><td>Error)
<table><tr><td>- Command sequencer writes to WorkFLASH or perform a command from WFCFG_SEQCM register (after this action, write operations/commands are ignored) unless the command sequencer is idle.
- Performing the sector erase suspend command or read/reset command while Flash is in the sector erase suspend state.
- Performing write or read operation to the sector to be erased while Flash is in the sector erase suspend state.
- Performing the sector erase suspend command while Flash is in the normal state.</td><td>8</td><td>ACCIGN</td></tr></table>
Correct)
<table><tr><td>- Command sequencer writes to WorkFLASH or perform a command from WFCFG_SEQCM register (after this action, write operations/commands are ignored) unless the command sequencer is idle.
- Performing the sector erase suspend command or read/reset command while Flash is in the sector erase suspend state.
- Performing write or read operation to the sector to be erased while Flash is in the sector erase suspend state.
- Performing the sector erase suspend command while Flash is in the normal state.
- Reading WorkFLASH while the command sequencer is operating.</td><td>8</td><td>ACCIGN</td></tr></table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1130</td><td>CHAPTER 18 WorkFLASH 3.Explanation of Operation 3.7.Bus Error Response</td><td>Error)
<table><tr><td>- Performing write operation to WFCFG_ECR more than once.
- Attempting to write to a protected register by performing an invalid register write unlock sequence. The invalid sequences are described specifically below.
- Writing to WFCFG_CPR two times in a row
- Writing a wrong key value to WFCFG_CPR</td><td>7</td><td>ECRWL</td></tr></table>
Correct)
<table><tr><td>- Performing write operation to WFCFG_ECR more than once.
- Attempting to write to a protected register by performing an invalid register write unlock sequence. The invalid sequences are described specifically below.
- Writing to WFCFG_CPR two times in a row
- Writing a wrong key value to WFCFG_CPR
- Writing to following protected registers in the locked state.
- WFCFG_CR
- WFCFG_ECR
- WFCFG_DBEIR
- WFCFG_EEIR</td><td>7</td><td>ECRWL</td></tr></table></td></tr>
<tr><td>1130</td><td>CHAPTER 18 WorkFLASH 3.Explanation of Operation 3.7.Bus Error Response</td><td>Error)
<table><tr><td>- Accessing the reserved area of WorkFLASH
(the reserved area in Flash and registers)</td><td>5</td><td>RESA</td></tr></table>
Correct)
<table><tr><td>- Accessing the reserved area of WorkFLASH
(reserved area in Flash and registers, refer next table for detail condition when RESA flag is set in access to WorkFLASH memory map)</td><td>5</td><td>RESA</td></tr></table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1130</td><td>CHAPTER 18 WorkFLASH 3.Explanation of Operation 3.7.Bus Error Response</td><td>Error)
<table><tr><td>- Performing write or erase operation to the mirror area 1 or mirror area 3 while WE bit of WFCFG_CR is “0”.
- Performing software reset while WE bit of WFCFG_CR is “0”.</td><td>1</td><td>CRWE</td></tr></table>
Correct)
<table><tr><td>- Performing write or erase operation to the mirror area 1 or mirror area 3 while WE bit of WFCFG_CR is “0”.
- Performing read/reset command while WE bit of WFCFG_CR is “0”.
- Performing software reset while WE bit of WFCFG_CR is “0”.</td><td>1</td><td>CRWE</td></tr></table></td></tr>
<tr><td>1130</td><td>CHAPTER 18 WorkFLASH 3.Explanation of Operation 3.7.Bus Error Response</td><td>Correct)
add the table

Following table shows occurrence of bus error response in access to each mirror area addresses, with relevant field state in WFCFG_BERR (RESA bit and NWTM bit).</td></tr>
</table>

Page	Section	Change Results
1134	CHAPTER 18 WorkFLASH 4.Setting Procedure of Operation 4.2.Enabling Writing	Error) Figure 4-1 Procedure for Setting Flash Write Enable  Correct) Figure 4-1 Procedure for Setting Flash Write Enable 

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1160</td><td>CHAPTER 18 WorkFLASH 5.Registers 5.13.WorkFLASH Sequencer Command Register: WFCFG_SEQCM</td><td>Error)
OPC[1:0] is reset to the initial value when the command specified by OPC[1:0] is complete.

Correct)
OPC[1:0] is reset to the initial value when the command specified by OPC[1:0] is complete.
While WE bit of WFCFG_CR is "0", writing OPC other than "00" causes bus error.</td></tr>
<tr><td>1164</td><td>CHAPTER 18 WorkFLASH 5.Registers 5.16.WorkFLASH Bus Error Response Status Register: WFCFG_BERR</td><td>Error)
[bit7] ECRWL: Protection Sequence Violation
<table>
<tr><th>ECRWL</th><th>Description</th></tr>
<tr><td>0</td><td>Writing to the protected register in violation of the protection sequence has not been detected.</td></tr>
<tr><td>1</td><td>WorkFLASH has detected a violation of the protection sequence and made a bus error response, in the following cases:
・Writing twice in a row to the protection key register WFCFG##_CPR
・Incorrect key value written to the protection key register
・Attempt to write to a protected register in the locked state
・Attempt to write to the ECCOFF bit in the WFCFG_ECR register a second or subsequent time</td></tr>
</table>
Correct)
[bit7] ECRWL: Protection Sequence Violation
<table>
<tr><th>ECRWL</th><th>Description</th></tr>
<tr><td>0</td><td>Writing to the protected register in violation of the protection sequence has not been detected.</td></tr>
<tr><td>1</td><td>WorkFLASH has detected a violation of the protection sequence and made a bus error response, in the following cases:
・Writing twice in a row to the protection key register WFCFG_CPR
・Incorrect key value written to the protection key register
・Attempt to write to a protected register in the locked state
・Attempt to write to the ECCOFF bit in the WFCFG_ECR register a second or subsequent time</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1164,
1165</td><td>CHAPTER 18 WorkFLASH
5.Registers
5.16.WorkFLASH Bus Error Response Status Register: WFCFG_BERR</td><td>Error)
[bit4] RWE: Reserved Area Access
<table>
<tr><th>RWE</th><th>Description</th></tr>
<tr><td>0</td><td>- Access to a protected sector has not been detected.
- No ERS writing violation has been detected.</td></tr>
<tr><td>1</td><td>- This indicates that a bus error response was made because access to a protected sector was detected.
- A sector assigned to SHE has been written to the ERS[7.0] bits in WFCFG_SEQCM.
- The sector protection function specifies the protected sectors.</td></tr>
</table>
Correct)
[bit4] RWE: Write protected sector access error
<table>
<tr><th>RWE</th><th>Description</th></tr>
<tr><td>0</td><td>- Access to a protected sector has not been detected.
- No ERS writing violation has been detected.</td></tr>
<tr><td>1</td><td>- This indicates that a bus error response was made because access to a protected sector was detected.
- A sector assigned to SHE has been written to the ERS[7.0] bits in WFCFG_SEQCM.</td></tr>
</table></td></tr>
<tr><td>1183</td><td>CHAPTER 19: BootROM Hardware Interface
5.Registers</td><td>Error)
<table>
<tr><td>0x000003D0</td><td>read0</td><td>EXCFG_DABORTACT
11111111_11111111_00000000_00110000</td></tr>
</table>
Correct)
<table>
<tr><td>0x000003D0</td><td>Reserved</td><td>EXCFG_DABORTACT
11111111_11111111_00000000_00110000</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1199</td><td>CHAPTER 20:BootROM Software Interface
2. BootROM Markers</td><td>Error)
2.4 Debugger Connection Enable Marker (DDR_DSM)

Correct)
2.4 Debugger Key Enable Marker (DDR_DSM)</td></tr>
<tr><td>1201</td><td>CHAPTER 20:BootROM Software Interface
2. BootROM Markers
2.2. Register Lists</td><td>Error)
List of the BootROM Marker Registers (BDR Registers)
Table 2-3 List of the BootROM Marker Registers (BDR Registers)
<table>
<tr><th>Abbreviated Register Name</th><th>Register Name</th><th>Reference</th></tr>
<tr><td>DDR_DSM</td><td>Debugger Connection Enable Marker</td><td>2.4</td></tr>
<tr><td>DDR_DSKM0</td><td>Debugger Security Key Marker 0</td><td>2.5</td></tr>
<tr><td>DDR_DSKM1</td><td>Debugger Security Key Marker 1</td><td>2.6</td></tr>
<tr><td>DDR_DSKM2</td><td>Debugger Security Key Marker 2</td><td>2.7</td></tr>
<tr><td>DDR_DSKM3</td><td>Debugger Security Key Marker 3</td><td>2.8</td></tr>
</table>
Correct)
List of the BootROM Marker Registers (BDR Registers)
Table 2-3 List of the BootROM Marker Registers (BDR Registers)
<table>
<tr><th>Abbreviated Register Name</th><th>Register Name</th><th>Reference</th></tr>
<tr><td>DDR_DSM</td><td>Debugger Key Enable Marker</td><td>2.4</td></tr>
<tr><td>DDR_DSKM0</td><td>Debugger Security Key Marker 0</td><td>2.5</td></tr>
<tr><td>DDR_DSKM1</td><td>Debugger Security Key Marker 1</td><td>2.6</td></tr>
<tr><td>DDR_DSKM2</td><td>Debugger Security Key Marker 2</td><td>2.7</td></tr>
<tr><td>DDR_DSKM3</td><td>Debugger Security Key Marker 3</td><td>2.8</td></tr>
</table></td></tr>
<tr><td>1207</td><td>CHAPTER 20:BootROM Software Interface
2. BootROM Markers
2.4. Debugger Key Enable Marker (DDR_DSM)</td><td>Error)
2.4. Debugger Connection Enable Marker (DDR_DSM)
This marker is used to enable debugger connection.

Correct)
2.4. Debugger Key Enable Marker (DDR_DSM)
This marker is used to enable the debugger connection key.</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1207</td><td>CHAPTER 20:BootROM Software Interface
2. BootROM Markers
2.4. Debugger Key Enable Marker (DDR_DSM)</td><td>
Error)

[bit31:0] DSEM: Debugger Connection Enable Marker
These bits are used to enable debugger connection. The following are the setting values.

<table>
<tr><th>bit31:0</th><th>Description</th></tr>
<tr><td>0x59F71234</td><td>Debugger connection enabled</td></tr>
<tr><td>Other than above</td><td>Debugger connection disabled (initial value)</td></tr>
</table>

If debugger connection is enabled with these bits, a 128-bit authentication key (DDR_DSKM0 to 3 registers) is required when a debugger is connected.

If security is disabled with the MK_SER register, an SR register, a 128-bit authentication key is not required when a debugger is connected.

<table>
<tr><th>MK_SER</th><th>DDR_DSM</th><th>Description</th></tr>
<tr><td>Security enabled</td><td>Debugger connection enabled</td><td>A 128-bit authentication key is required when a debugger is connected.</td></tr>
<tr><td>Security disabled</td><td>Debugger connection enabled</td><td>A 128-bit authentication key is not required when a debugger is connected.</td></tr>
<tr><td>Security enabled</td><td>Debugger connection disabled</td><td>A debugger cannot be connected.</td></tr>
<tr><td>Security disabled</td><td>Debugger connection disabled</td><td>A 128-bit authentication key is not required when a debugger is connected.</td></tr>
</table>

Correct)

[bit31:0] DSEM: Debugger Key Enable Marker
These bits are used to enable the debugger security key. The following are the setting values.

<table>
<tr><th>bit31:0</th><th>Description</th></tr>
<tr><td>0x59F71234</td><td>Debugger key enabled</td></tr>
<tr><td>Other than above</td><td>Debugger connection disabled (initial value)</td></tr>
</table>

If debugger key is enabled with these bits, a 128-bit authentication key (DDR_DSKM0 to 3 registers) can be used to open a debugger connection in case the device is in the security enabled state.

If security is disabled with the MK_SER register, an SR register, a 128-bit authentication key is not required when a debugger is connected.

<table>
<tr><th>MK_SER</th><th>DDR_DSEM</th><th>Description</th></tr>
<tr><td>Security disabled</td><td>Debugger key disabled</td><td>The debugger can be connected without a key.</td></tr>
<tr><td>Security disabled</td><td>Debugger key enabled</td><td>The debugger can be connected without a key.</td></tr>
<tr><td>Security enabled</td><td>Debugger key disabled</td><td>Either an application based authentication scheme has to be used or a debugger cannot be connected.</td></tr>
<tr><td>Security enabled</td><td>Debugger key enabled</td><td>A 128-bit authentication key can be used to open the debugger connection.</td></tr>
</table>
</td></tr>
</table>

Page	Section	Change Results
1208	CHAPTER 20:BootROM Software Interface 2. BootROM Markers 2.5. Debugger Security Key Marker 0 (DDR_DSKM0)	Error) **[bit31:0] DSKM (bit127:96): Debugger Security Key Marker (bit127:96)** These bits are a 128-bit authentication key for debugger connection when debugger connection is enabled by Debugger Connection Enable Marker (DSEM [31:0] bits of the DDR_DSM register). These bits are bit127:96 of the key. Correct) **[bit31:0] DSKM (bit127:96): Debugger Security Key Marker (bit127:96)** These bits are a 128-bit authentication key for debugger connection when debugger connection is enabled by Debugger Key Enable Marker (DSEM [31:0] bits of the DDR_DSM register). These bits are bit127:96 of the key. A trivial key DSKM[127:0] equal to all-0 or all-1 cannot be used to authenticate the debugger connection.
1209	CHAPTER 20:BootROM Software Interface 2. BootROM Markers 2.6. Debugger Security Key Marker 1 (DDR_DSKM1)	Error) **[bit31:0] DSKM (bit95:64): Debugger Security Key Marker (bit95:64)** These bits are a 128-bit authentication key for debugger connection when debugger connection is enabled by Debugger Connection Enable Marker (DSEM [31:0] bits of the DDR_DSM register). These bits are bit95:64 of the key. Correct) **[bit31:0] DSKM (bit95:64): Debugger Security Key Marker (bit95:64)** These bits are a 128-bit authentication key for debugger connection when debugger connection is enabled by Debugger Key Enable Marker (DSEM [31:0] bits of the DDR_DSM register). These bits are bit95:64 of the key. A trivial key DSKM[127:0] equal to all-0 or all-1 cannot be used to authenticate the debugger connection.
1210	CHAPTER 20:BootROM Software Interface 2. BootROM Markers 2.7. Debugger Security Key Marker 2 (DDR_DSKM2)	Error) **[bit31:0] DSKM (bit63:32): Debugger Security Key Marker (bit63:32)** These bits are a 128-bit authentication key for debugger connection when debugger connection is enabled by Debugger Connection Enable Marker (DSEM [31:0] bits of the DDR_DSM register). These bits are bit63:32 of the key. Correct) **[bit31:0] DSKM (bit63:32): Debugger Security Key Marker (bit63:32)** These bits are a 128-bit authentication key for debugger connection when debugger connection is enabled by Debugger Key Enable Marker (DSEM [31:0] bits of the DDR_DSM register). These bits are bit63:32 of the key. A trivial key DSKM[127:0] equal to all-0 or all-1 cannot be used to authenticate the debugger connection.

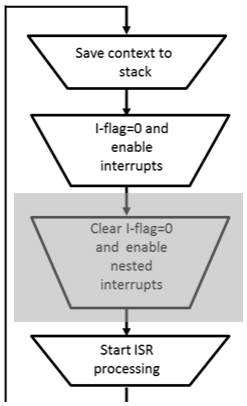
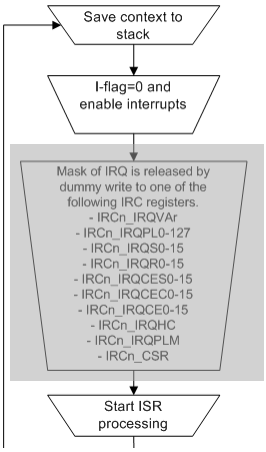
Page	Section	Change Results
1211	CHAPTER 20:BootROM Software Interface 2. BootROM Markers 2.8. Debugger Security Key Marker 3 (DDR_DSKM3)	Error) **[bit31:0] DSKM (bit31:0): Debugger Security Key Marker (bit31:0)** These bits are a 128-bit authentication key for debugger connection when debugger connection is enabled by Debugger Connection Enable Marker (DSEM [31:0] bits of the DDR_DSM register). These bits are bit31:0 of the key. Correct) **[bit31:0] DSKM (bit31:0): Debugger Security Key Marker (bit31:0)** These bits are a 128-bit authentication key for debugger connection when debugger connection is enabled by Debugger Key Enable Marker (DSEM [31:0] bits of the DDR_DSM register). These bits are bit31:0 of the key. A trivial key DSKM[127:0] equal to all-0 or all-1 cannot be used to authenticate the debugger connection.
1232	CHAPTER 20:BootROM Software Interface 3. BootROM Operation	Add Correct) **Core Initial Setting** During the core initial setting, the following settings are made. ■ Initialization of the general-purpose registers ■ VIC port enable setting ■ FPU enable setting ■ The settings of the stack pointers for the following modes, which are used by the BootROM software 1. System mode 2. ABORT 3. Undefined instruction exception

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1232</td><td>CHAPTER 20:BootROM Software Interface
3. BootROM Operation</td><td>Error)
Table 3-1 BootROM Software CPU Initial Settings
<table>
<tr><th>Function</th><th>BootROM Software Processing</th></tr>
<tr><td>System control</td><td>VIC port enable setting
Access right setting of the coprocessor for the FPU</td></tr>
<tr><td>MPU control</td><td>Not set</td></tr>
<tr><td>Cache control</td><td>Not set</td></tr>
<tr><td>TCM control</td><td>Not set</td></tr>
<tr><td>System performance</td><td>Cycle Counter is used for debugger connection wait (it is initialized after being used).</td></tr>
<tr><td>FPU control</td><td>Invalid because of FPU is not mounted</td></tr>
<tr><td>Debugging reset</td><td>Not set</td></tr>
</table>
Correct)
Table 3-1 BootROM Software CPU Initial Settings
<table>
<tr><th>Function</th><th>BootROM Software Processing</th></tr>
<tr><td>System control</td><td>VIC port enable setting
Access right setting of the coprocessor for the FPU</td></tr>
<tr><td>MPU control</td><td>Not set</td></tr>
<tr><td>Cache control</td><td>Not set</td></tr>
<tr><td>TCM control</td><td>Not set</td></tr>
<tr><td>System performance</td><td>Cycle Counter is used for debugger connection wait (it is initialized after being used).</td></tr>
<tr><td>FPU control</td><td>FPU enable setting</td></tr>
<tr><td>Debugging reset</td><td>Not set</td></tr>
</table></td></tr>
<tr><td>1235</td><td>CHAPTER 20:BootROM Software Interface
3. BootROM Operation</td><td>Error)
The value of the Debugger Connection Enable Marker (DDR_DSM) is 0x59F71234.

Correct)
The value of the Debugger Key Enable Marker (DDR_DSM) is 0x59F71234.</td></tr>
<tr><td>1237</td><td>CHAPTER 20:BootROM Software Interface
3. BootROM Operation</td><td>Error)
Before jumping to the user application, the BootROM software clears the stack area and general-purpose registers R1 to R13 it has used.

Correct)
Before jumping to the user application, the BootROM software clears the stack area and general-purpose registers R0 to R12 it has used.</td></tr>
</table>

Page	Section	Change Results
1238	CHAPTER 20:BootROM Software Interface 4. Notes	Add Correct) **ARM Core Setting** 1. R0-R12 are cleared to 0. 2. Bit24(VE) of System Control Register (SCTLR) is set to 1. 3. Bit23:22(CP11) and bit21:20(CP10) of Coprocessor Access Control Register (CPACR) are set to 0b11 (Privileged and User mode access). 4. Bit30(EN) of Floating-Point Exception Register (FPEXC) is set to 1.

Page	Section	Change Results
1262	CHAPTER 22:Interrupt Controller 1. Overview	Error) - *For details on the interrupt factor assignment to each NMI/IRQ channel, see "APPENDIXES: LIST OF INTERRUPT FACTOR AND DMA ACTIVATION FACTOR."* Correct) - *For details on the interrupt factor assignment to each NMI/IRQ channel, see product specification.*
1267	CHAPTER 22:Interrupt Controller 3. Explanation of Operation	Error) During the execution of the ISR for IRQ50, IRQ20 occurs. IRQ20 will wait because IRS for IRQ50 has not ended. Correct) During the execution of the ISR for IRQ50, IRQ20 occurs. IRQ20 will wait because ISR for IRQ50 has not ended.
1270	CHAPTER 22:Interrupt Controller 3. Explanation of Operation	Error) Figure 3 4 Nest Supported IRQ Processing Flow  Correct) Figure 3 4 Nest Supported IRQ Processing Flow 

Page	Section	Change Results
1274	CHAPTER 22:Interrupt Controller 3. Explanation of Operation	Error) The IRCn_EEI:EEIS bit is set and the IRQ is generated for both IRQVAr register reading and reading from the IRQ processing stage. Correct) The IRCn_EEI:EEIS bit is set and the IRQ is generated for both IRCn_IRQVAr register reading and reading from the IRQ processing stage.
1274	CHAPTER 22:Interrupt Controller 3. Explanation of Operation	Error) If an error of 2 or more bits (uncorrectable) occurs, the IRCn_EEI:EENS bit is set and the NMI is generated at IRQVAr register reading. Correct) If an error of 2 or more bits (uncorrectable) occurs, the IRCn_EEI:EENS bit is set and the NMI is generated at IRCn_IRQVAr register reading.
1322	CHAPTER 22:Interrupt Controller 5. Registers 5.30. IRC ECC Error Interrupt Register (IRCn_EEI)	Error) **[bit8] EENS: ECC Error NMI Status Bit** This bit indicates the status of the ECC error NMI. When data is read from SRAM installed in the interrupt controller through the IRQVAr register, this bit becomes "1" if a double-bit ECC error occurs. Correct) **[bit8] EENS: ECC Error NMI Status Bit** This bit indicates the status of the ECC error NMI. When data is read from SRAM installed in the interrupt controller through the IRCn_IRQVAr register, this bit becomes "1" if a double-bit ECC error occurs.
1334	CHAPTER 23:TPU 3. Explanation of Operation	Error) The lock must be released by writing the lock release value in PUn_UNLOCK before the timer setting can be updated Correct) The lock must be released by writing the lock release value in TPU0_UNLOCK before the timer setting can be updated

Page	Section	Change Results
1353	CHAPTER 24:Security	Added Error) 1. Overview 2. Security Scope and Access Restriction by the Security 3. Using and Configuring the Security 4. Registers Correct) 1. Overview 2. Security Scope and Access Restriction by the Security 3. Using and Configuring the Security 4. Registers In this chapter, there is a place with two specifications, "SWP_NOMAL_type" and "SWP_LITE_type". Basically please refer to "SWP_NORMAL_type" specification, but when SWP_LITE_type is used, please refer to "SWP_LITE_type" specification and see the Product's HWM.
1355	CHAPTER 24:Security 2. Security Scope and Access Restriction by the Security 2.1. Security Scope	Added Error) The ability to execute “chip erase” always exists. The marker MK_CEER does not have any functionality. Correct) Refer to datasheet of product. If being written with "MK_CEER = Selectable", below is spec. The ability to execute “chip erase” function depends on the permission set by the marker MK_CEER. The possibility of doing chip erase can be removed, when the user intents to forbid the usage of other parties software on his system. However this removes any possibility for a recovery of a device when not implementing an unlock mechanism or forgetting the key for a device unlock. To protect the software IP only from being disassembled, debugged or copied – the normal security is enough and chip erase functionality should stay enabled. Other spec is below. The ability to execute “chip erase” always exists. The marker MK_CEER does not have any functionality.

Page	Section	Change Results
1362	CHAPTER 24:Security 3. Using and Configuring the Security 3.3. Security Marker Definition	Added Error) This section describes details of the Flash Marker which is used to configure the security setting. The security configuration is stored in the Flash Marker which is located to head of the SA0 area in small sector area of TCFlash. Correct) This section describes details of the Flash Marker which is used to configure the security setting. The security configuration is stored in the Flash Marker which is located to head of the SA0 area in small sector area of TCFlash#0.
1362	CHAPTER 24:Security 3. Using and Configuring the Security 3.3. Security Marker Definition	Error) The security markers are transferred after each RSTX_PD2H into the appropriate registers containing the security state of the device. This process is called “Security Fetch”. There is no error signaling during that hardware mechanism, since the CPU is not involved into that mechanism (no bus transfer). No errors will succeed in loading the marker values from Flash to registers. Single bit errors get corrected and will succeed in loading the marker values from Flash to registers. Fatal errors will not succeed in loading the marker values from Flash to registers. The according security register will keep its previous initial value. To check the integrity of ECC of the above marker section, it is recommended, to read the markers by the application. The normal ECC error detection by the TCFLASH interface can be utilized for that. Correct) The security markers are transferred after each RSTX_PD2H into the appropriate registers containing the security state of the device. This process is called "Security Fetch". No errors will succeed in loading the marker values from Flash to registers. Single bit errors get corrected and will succeed in loading the marker values from Flash to registers. Fatal errors will not succeed in loading the marker values from Flash to registers. The according security register will keep its previous initial value. To check detail of the integrity of ECC of the above marker section, it is recommended, to read the markers by the application. The normal ECC error detection by the TCFLASH interface can be utilized for that.

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1364</td><td>CHAPTER 24:Security 3. Using and Configuring the Security 3.3. Security Marker Definition 3.3.3. Chip Erase Enable (MK_CEER)</td><td>Added
Correct)

Refer to datasheet of product.
If being written with "MK_CEER = Selectable", below is spec.

[31:0] CEER: Chip Erase Enable
The value controls the ability to execute the “chip erase” function.

<table><tr><th>Value</th><th>Description</th></tr><tr><td>0xFFFFFFFF</td><td>The MCU will support the “chip erase” command. As an effect a secured device can be unlocked by erasing all its internal secret content, except SHE.</td></tr><tr><td>Other</td><td>The MCU has disabled the “chip erase” command. Without knowing authentication scheme, the device cannot be erased anymore (independent from its operation mode).</td></tr></table>
The marker value is automatically copied to TCFCFG0_CEER after each RSTX_PD2H.

Other spec is below.</td></tr>
<tr><td>1369</td><td>CHAPTER 24:Security 3. Using and Configuring the Security 3.4. Gain Access in USERMODE 3.4.1. Gaining program / erase permission to Flash Sectors</td><td>Error)

Each of the bits in the above registers are controlling erase and program permissions of single Flash sectors. Lower bits reflect the permission of the lower sector number. Higher bits reflect the setting of the higher sector number.
The sector permission registers are defining accessibility of Flash during USERMODE.

Correct)

Each of the bits in the above registers control write permissions (erase and program) of single Flash sectors. Lower bits reflect the permission of the lower sector number. Higher bits reflect the setting of the higher sector number. Depending on the Flash size equipped with the device only such sector write permission registers or bits have an effect, for which the according Flash sector exists.
The sector permission registers are defining accessibility of Flash during USERMODE.</td></tr>
</table>

Page	Section	Change Results
1370	CHAPTER 24:Security 3. Using and Configuring the Security 3.4. Gain Access in USERMODE 3.4.1. Gaining program / erase permission to Flash Sectors	Added Error) The macro erase command is not supported by the WORKFLASH. Correct) ■The macro erase command is not supported by the WORKFLASH. ■ ■ **SWP_LITE_type** Following procedure is required to ensure the operation of Sector Write Permissions. When programing to TCFLASH, all of following Sector Write Permissions register bits must be set to “1” to permit the TCFLASH programing. - Code Flash 0 Sector Write Permissions of the Small Sectors (TCFCFG0_C0SWP) - Code Flash 0 Sector Write Permissions of the Large Sectors (TCFCFG0_C1SWP) - Code Flash 1 Sector Write Permissions of the Large Sectors (TCFCFG0_C2SWP) - Code Flash 2 Sector Write Permissions of the Large Sectors (TCFCFG0_C3SWP) After programing, set the original values of Sector Write Permissions again to protect the TCFLASH data. ■**SWP_NORMAL_type** No additional prodecure is required. ■**end_of_type_listing**

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1374</td><td>CHAPTER 24:Security 4. Registers 4.1. Security Status Register (TCFCFG0_SECSTAT)</td><td>
Error)
<table>
<tr><th>BIT_OFFSET</th><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th></tr>
<tr><td>BIT_NAME</td><td colspan="8">Reserved</td></tr>
<tr><td>ACCESS_TYPE</td><td>R,WX</td><td>R,WX</td><td>R,WX</td><td>R,WX</td><td>R,WX</td><td>R,WX</td><td>R,WX</td><td>R,WX</td></tr>
<tr><td>PROT_TYPE</td><td colspan="8">-</td></tr>
<tr><td>INITIAL_VALUE</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
<table>
<tr><th>BIT_OFFSET</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th></tr>
<tr><td>BIT_NAME</td><td colspan="8">Reserved</td></tr>
<tr><td>ACCESS_TYPE</td><td>R,WX</td><td>R,WX</td><td>R,WX</td><td>R,WX</td><td>R,WX</td><td>R,WX</td><td>R,WX</td><td>R,WX</td></tr>
<tr><td>PROT_TYPE</td><td colspan="8">-</td></tr>
<tr><td>INITIAL_VALUE</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
<table>
<tr><th>BIT_OFFSET</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th></tr>
<tr><td>BIT_NAME</td><td colspan="7">Reserved</td><td>UMV</td></tr>
<tr><td>ACCESS_TYPE</td><td>R,WX</td><td>R,WX</td><td>R,WX</td><td>R,WX</td><td>R,WX</td><td>R,WX</td><td>R,WX</td><td>R,WX</td></tr>
<tr><td>PROT_TYPE</td><td colspan="8">-</td></tr>
<tr><td>INITIAL_VALUE</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr>
</table>
Correct)
<table>
<tr><td>BIT_OFFSET</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td>BIT_NAME</td><td colspan="8">Reserved</td></tr>
<tr><td>ACCESS_TYPE</td><td>R,WX</td><td>R,WX</td><td>R,WX</td><td>R,WX</td><td>R,WX</td><td>R,WX</td><td>R,WX</td><td>R,WX</td></tr>
<tr><td>PROT_TYPE</td><td colspan="8">-</td></tr>
<tr><td>INITIAL_VALUE</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>BIT_OFFSET</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td>BIT_NAME</td><td colspan="8">Reserved</td></tr>
<tr><td>ACCESS_TYPE</td><td>R,WX</td><td>R,WX</td><td>R,WX</td><td>R,WX</td><td>R,WX</td><td>R,WX</td><td>R,WX</td><td>R,WX</td></tr>
<tr><td>PROT_TYPE</td><td colspan="8">-</td></tr>
<tr><td>INITIAL_VALUE</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>NEUTRAL_TYPE</td><td>*</td><td>*</td><td>*</td><td>*</td><td>*</td><td>*</td><td>*</td><td>*</td></tr>
<tr><td>BIT_OFFSET</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td>BIT_NAME</td><td colspan="7">Reserved</td><td>UMV</td></tr>
<tr><td>ACCESS_TYPE</td><td>R,WX</td><td>R,WX</td><td>R,WX</td><td>R,WX</td><td>R,WX</td><td>R,WX</td><td>R,WX</td><td>R,WX</td></tr>
<tr><td>PROT_TYPE</td><td colspan="8">-</td></tr>
<tr><td>INITIAL_VALUE</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1374</td><td>CHAPTER 24:Security 4. Registers 4.1. Security Status Register (TCFCFG0_SECSTAT)</td><td>Error)
<table><tr><th>BIT_OFFSET</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr><tr><td>BIT_NAME</td><td>Reserved</td><td>Reserved</td><td>SFDONE</td><td>SWPOE</td><td>SECOE</td><td>CEEN</td><td>SECSC</td><td>SECEN</td></tr><tr><td>ACCESS_TYPE</td><td>RX,WX</td><td>RX,WX</td><td>R,WX</td><td>R,WX</td><td>R,WX</td><td>R,WX</td><td>R,WX</td><td>R,WX</td></tr><tr><td>PROT_TYPE</td><td colspan="8">-</td></tr><tr><td>INITIAL_VALUE</td><td>X</td><td>X</td><td>1</td><td>0 or 1</td><td>0 or 1</td><td>1</td><td>0 or 1</td><td>0 or 1</td></tr></table>
Correct)
<table><tr><td>BIT_OFFSET</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr><tr><td>BIT_NAME</td><td>Reserved</td><td>Reserved</td><td>SFDONE</td><td>SWPOE</td><td>SECOE</td><td>CEEN</td><td>SECSC</td><td>SECEN</td></tr><tr><td>ACCESS_TYPE</td><td>RX,WX</td><td>RX,WX</td><td>R,WX</td><td>R,WX</td><td>R,WX</td><td>R,WX</td><td>R,WX</td><td>R,WX</td></tr><tr><td>PROT_TYPE</td><td colspan="8">-</td></tr><tr><td>INITIAL_VALUE</td><td>X</td><td>X</td><td>1</td><td>0 or 1</td><td>0 or 1</td><td>0 or 1</td><td>0 or 1</td><td>0 or 1</td></tr></table></td></tr>
<tr><td>1374</td><td>CHAPTER 24:Security 4. Registers 4.1. Security Status Register (TCFCFG0_SECSTAT)</td><td>Error)
[31:9] Reserved: These Bits are Not Used. Writing has No effect.

Correct)

[bit31:16] Reserved: Reserved bits
These bits are not used. Writing has no effect.

[bit15:9] Reserved: Reserved bits
These bits are not used. Writing has no effect.</td></tr>
<tr><td>1374</td><td>CHAPTER 24:Security 4. Registers 4.1. Security Status Register (TCFCFG0_SECSTAT)</td><td>Error)
[8] UMV: Specification of the Unlock Marker Value
“1” is always read.
UMV is a read-only bit. Any write attempt will cause a bus error.

Correct)

[bit8] UMV: Specification of the unlock marker value
<table><tr><th>Bit</th><th>Description</th></tr><tr><td>0</td><td>The device supports the non-trivial unlock marker value 0x692E3A7B.</td></tr><tr><td>1</td><td>The device supports the trivial unlock marker value 0xFFFFFFFF.</td></tr></table>
The register bit UMV indicates what value has to be written to the markers for the purpose of unlocking or to configure the less secure option. It has effect on the interpretation of MK_SER, MK_SSR, MK_CEER, MK_SOER and MK_SWPOER.
"1" is always read.
UMV is a read-only bit. Any write attempt will cause a bus error.</td></tr>
</table>

Page	Section	Change Results
1375	CHAPTER 24:Security 4. Registers 4.1. Security Status Register (TCFCFG0_SECSTAT)	Error) **[7:6] Reserved: These Bits are Not Used. Any Write attempt will Cause a Bus Error.** Correct) **[bit7:6] Reserved: Reserved bits** These bits are not used. Any write attempt will cause a bus error.
1376	CHAPTER 24:Security 4. Registers 4.1. Security Status Register (TCFCFG0_SECSTAT)	Added Error) The register bit CEEN indicates the permission to erase the full device. Dedicated memory of the SHE module cannot be erased, regardless of that setting. CEEN is a read-only bit. Any write attempt will cause a bus error. In this product CEEN is always ‘1’ (fixed, because MK_CEER marker is not supported). Correct) The register bit CEEN indicates the permission to erase the full device. Dedicated memory of the SHE module cannot be erased, regardless of that setting. CEEN is a read-only bit. Any write attempt will cause a bus error. Refer to datasheet of product. If being written with "MK_CEER != Selectable", below is spec. In this product CEEN is always ‘1’ (fixed, because MK_CEER marker is not supported).

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1381</td><td>CHAPTER 24:Security 4. Registers 4.4. Chip Erase Enable Register (TCFCFG0_CEER)</td><td>Added Error)

[31:0] CEER: Chip Erase Enable Register
Disabling chip erase is not possible by this product. Writing to that register has no effect.

Correct)

[bit31:0] CEER: Chip erase enable register
Refer to datasheet of product.
If being written with "MK_CEER = Selectable", below is spec.
Full-chip erase can be forbidden using this marker.

<table><tr><th>Value</th><th>Description</th></tr><tr><td>0xFFFFFFFF</td><td>The chip erase of a secured device is enabled.</td></tr><tr><td>Other</td><td>Any other value disables the chip erase in non-USERMODE.</td></tr></table>
Without enabling the security by TCFCFG0_SER, the Chip Erase Enable Register has no meaning.

Other spec is below.
Disabling chip erase is not possible by this product. Writing to that register has no effect.</td></tr>
</table>

Page	Section	Change Results
1382	CHAPTER 24:Security 4. Registers 4.4. Chip Erase Enable Register (TCFCFG0_CEER)	Added Error) TCFCFG0_CEER can be written by the application, when TCFCFG0_SOER indicates that overwrite is enabled. Otherwise the write access will cause a bus error. Correct) TCFCFG0_CEER can be written by the application, when TCFCFG0_SOER indicates that overwrite is enabled. Otherwise the write access will cause a bus error. Refer to datasheet of product. If being written with "MK_CEER = Selectable", below is spec. The chip erase can be triggered by JTAG equipment, writing the chip erase sequence to the general purpose registers of the security checker or transferring the chip erase sequence through serial interface when using Serial Loader equipment. It erases all sectors of the WORKFLASH and TCFLASH. Sectors used for SHE are not erased. ***Notes:*** – *A secured device without an unlock mechanism implemented by the application and without the ability to erase the full device cannot be updated or tested anymore. Not programming the unlock marker value to CEEM should be used with absolute care!* – *Single ECC errors are corrected. When double ECC errors are detected, the register is not updated from Flash and the chip erase will be disabled.*

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1420</td><td>CHAPTER 25:Memory Protection Unit for AXI of SHE 4. Registers 4.1. MPU AXI Control Register (MPUXSHE_CTRL0)</td><td>
Error)
■ MPUXSHE_CTRL0
<table>
<tr><td>bit</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td>Field</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>AP[2]</td><td>AP[1]</td><td>AP[0]</td></tr>
<tr><td>R/W Attribute</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td><td>R/W</td><td>R/W</td><td>R/W</td></tr>
<tr><td>Protection Attribute</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>WP</td><td>WP</td><td>WP</td></tr>
<tr><td>Initial Value</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
<table>
<tr><td>bit</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td>Field</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>MPUENC</td><td>MPUEN</td></tr>
<tr><td>R/W Attribute</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td><td>R/W</td><td>R</td></tr>
<tr><td>Protection Attribute</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>WP</td><td>-</td></tr>
<tr><td>Initial Value</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
<table>
<tr><td>bit</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td>Field</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>PROT</td><td>POEN</td><td>MPUSTOPEN</td><td>MPUSTOP</td><td>LST</td></tr>
<tr><td>R/W Attribute</td><td>R0</td><td>R0</td><td>R0</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R</td><td>R</td></tr>
<tr><td>Protection Attribute</td><td>-</td><td>-</td><td>-</td><td>WP</td><td>WP</td><td>WP</td><td>-</td><td>-</td></tr>
<tr><td>Initial Value</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr>
</table>
<table>
<tr><td>bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Field</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>NMICL</td><td>NMI</td></tr>
<tr><td>R/W Attribute</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td><td>R0,W</td><td>R</td></tr>
<tr><td>Protection Attribute</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>WP</td><td>-</td></tr>
<tr><td>Initial Value</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>

Correct)
■ MPUXSHE_CTRL0
<table>
<tr><td>bit</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td>Field</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>AP[2]</td><td>AP[1]</td><td>AP[0]</td></tr>
<tr><td>R/W Attribute</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td><td>R/W</td><td>R/W</td><td>R/W</td></tr>
<tr><td>Protection Attribute</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>WPS</td><td>WPS</td><td>WPS</td></tr>
<tr><td>Initial Value</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
<table>
<tr><td>bit</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td>Field</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>MPUENC</td><td>MPUEN</td></tr>
<tr><td>R/W Attribute</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td><td>R/W</td><td>R</td></tr>
<tr><td>Protection Attribute</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>WPS</td><td>-</td></tr>
<tr><td>Initial Value</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
<table>
<tr><td>bit</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td>Field</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>PROT</td><td>POEN</td><td>MPUSTOPEN</td><td>MPUSTOP</td><td>LST</td></tr>
<tr><td>R/W Attribute</td><td>R0</td><td>R0</td><td>R0</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R</td><td>R</td></tr>
<tr><td>Protection Attribute</td><td>-</td><td>-</td><td>-</td><td>WPS</td><td>WPS</td><td>WPS</td><td>-</td><td>-</td></tr>
<tr><td>Initial Value</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr>
</table>
<table>
<tr><td>bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Field</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>NMICL</td><td>NMI</td></tr>
<tr><td>R/W Attribute</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td><td>R0,W</td><td>R</td></tr>
<tr><td>Protection Attribute</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>WPS</td><td>-</td></tr>
<tr><td>Initial Value</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1423</td><td>CHAPTER 25:Memory Protection Unit for AXI of SHE 4. Registers 4.2. MPU AXI NMI Enable Register (MPUXSHE_NMIEN)</td><td>
Error)
■ MPUXSHE_NMIEN
<table>
<tr><td>bit</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td>Field</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>R/W Attribute</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td></tr>
<tr><td>Protection Attribute</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>Initial Value</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
<table>
<tr><td>bit</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td>Field</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>R/W Attribute</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td></tr>
<tr><td>Protection Attribute</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>Initial Value</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
<table>
<tr><td>bit</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td>Field</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>R/W Attribute</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td></tr>
<tr><td>Protection Attribute</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>Initial Value</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
<table>
<tr><td>bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Field</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>NMIEN</td></tr>
<tr><td>R/W Attribute</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td><td>R/W</td></tr>
<tr><td>Protection Attribute</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>WP</td></tr>
<tr><td>Initial Value</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr>
</table>

Correct)
■ MPUXSHE_NMIEN
<table>
<tr><td>bit</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td>Field</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>R/W Attribute</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td></tr>
<tr><td>Protection Attribute</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>Initial Value</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
<table>
<tr><td>bit</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td>Field</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>R/W Attribute</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td></tr>
<tr><td>Protection Attribute</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>Initial Value</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
<table>
<tr><td>bit</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td>Field</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>R/W Attribute</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td></tr>
<tr><td>Protection Attribute</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>Initial Value</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
<table>
<tr><td>bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Field</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>NMIEN</td></tr>
<tr><td>R/W Attribute</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td><td>R/W</td></tr>
<tr><td>Protection Attribute</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>WPS</td></tr>
<tr><td>Initial Value</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1424</td><td>CHAPTER 25:Memory Protection Unit for AXI of SHE 4. Registers 4.3. MPU AXI Write Error Control Register (MPUXSHE_ WERRC)</td><td>
Error)
■ MPUXSHE_WERRC
<table>
<tr><td>bit</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td>Field</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>R/W Attribute</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td></tr>
<tr><td>Protection Attribute</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>Initial Value</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
<table>
<tr><td>bit</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td>Field</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>R/W Attribute</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td></tr>
<tr><td>Protection Attribute</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>Initial Value</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
<table>
<tr><td>bit</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td>Field</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>AWSIZE[2]</td><td>AWSIZE[1]</td><td>AWSIZE[0]</td></tr>
<tr><td>R/W Attribute</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td><td>R</td><td>R</td><td>R</td></tr>
<tr><td>Protection Attribute</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>Initial Value</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>X</td><td>X</td><td>X</td></tr>
</table>
<table>
<tr><td>bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Field</td><td>AWBURST[1]</td><td>AWBURST[0]</td><td>AWLEN[3]</td><td>AWLEN[2]</td><td>AWLEN[1]</td><td>AWLEN[0]</td><td>AWPLOTPRIV</td><td>AWMPV</td></tr>
<tr><td>R/W Attribute</td><td>R</td><td>R</td><td>R</td><td>R</td><td>R</td><td>R</td><td>R</td><td>R</td></tr>
<tr><td>Protection Attribute</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>Initial Value</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>0</td></tr>
</table>
Correct)
■ MPUXSHE_WERRC
<table>
<tr><td>bit</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td>Field</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>R/W Attribute</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td></tr>
<tr><td>Protection Attribute</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>Initial Value</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
<table>
<tr><td>bit</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td>Field</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>R/W Attribute</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td></tr>
<tr><td>Protection Attribute</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>Initial Value</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
<table>
<tr><td>bit</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td>Field</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>AWSIZE[2]</td><td>AWSIZE[1]</td><td>AWSIZE[0]</td></tr>
<tr><td>R/W Attribute</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td><td>R0</td><td>R</td><td>R</td><td>R</td></tr>
<tr><td>Protection Attribute</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>Initial Value</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>X</td><td>X</td><td>X</td></tr>
</table>
<table>
<tr><td>bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Field</td><td>AWBURST[1]</td><td>AWBURST[0]</td><td>AWLEN[3]</td><td>AWLEN[2]</td><td>AWLEN[1]</td><td>AWLEN[0]</td><td>AWPROTPRIV</td><td>AWMPV</td></tr>
<tr><td>R/W Attribute</td><td>R</td><td>R</td><td>R</td><td>R</td><td>R</td><td>R</td><td>R</td><td>R</td></tr>
<tr><td>Protection Attribute</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>Initial Value</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>0</td></tr>
</table>
</td></tr>
</table>

Page	Section	Change Results
1430	CHAPTER 25:Memory Protection Unit for AXI of SHE 4. Registers 4.7. MPU AXI Region Control Registers (MPUXSHE_CTRL1~8)	Error)

■ MPUXSHE_CTRL1

bit	31	30	29	28	27	26	25	24
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W Attribute	R0	R0	R0	R0	R0	R0	R0	R0
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W Attribute	R0	R0	R0	R0	R0	R0	R0	R0
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved	Reserved	Reserved	Reserved	AP[2]	AP[1]	AP[0]
R/W Attribute	R0	R0	R0	R0	R0	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	WP	WP	WP
Initial Value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	MPUENC	MPUEN
R/W Attribute	R0	R0	R0	R0	R0	R0	R/W	R
Protection Attribute	-	-	-	-	-	-	WP	-
Initial Value	0	0	0	0	0	0	0	0

Correct)

■ MPUXSHE_CTRL1

bit	31	30	29	28	27	26	25	24
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W Attribute	R0	R0	R0	R0	R0	R0	R0	R0
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

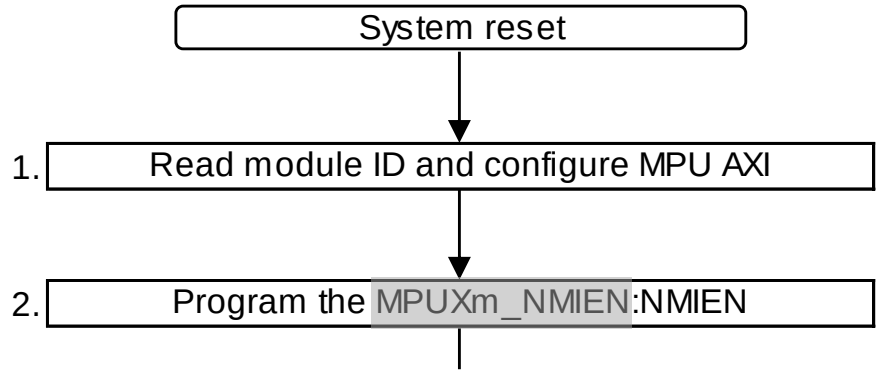
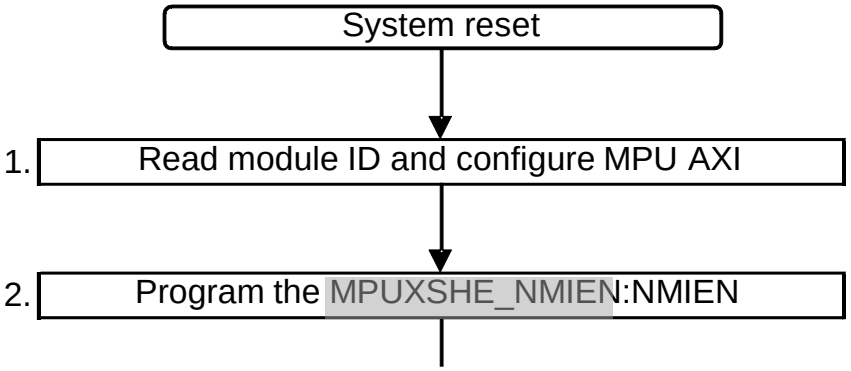
bit	23	22	21	20	19	18	17	16
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W Attribute	R0	R0	R0	R0	R0	R0	R0	R0
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved	Reserved	Reserved	Reserved	AP[2]	AP[1]	AP[0]
R/W Attribute	R0	R0	R0	R0	R0	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	WP	WP	WP
Initial Value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	MPUENC	MPUEN
R/W Attribute	R0	R0	R0	R0	R0	R0	R/W	R
Protection Attribute	-	-	-	-	-	-	WPS	-
Initial Value	0	0	0	0	0	0	0	0

Page	Section	Change Results
1432	CHAPTER 25:Memory Protection Unit for AXI of SHE 4. Registers 4.8. MPU AXI Start Address Registers (MPUXSHE_SADDR1~8)	Error) ■ MPUXSHE_SADDR1 bit 31 30 29 28 27 26 25 24 Field SADDR[31] SADDR[30] SADDR[29] SADDR[28] SADDR[27] SADDR[26] SADDR[25] SADDR[24] R/W Attribute R/W R/W R/W R/W R/W R/W R/W R/W Protection Attribute WP WP WP WP WP WP WP WP Initial Value 0 0 0 0 0 0 0 0 bit 23 22 21 20 19 18 17 16 Field SADDR[23] SADDR[22] SADDR[21] SADDR[20] SADDR[19] SADDR[18] SADDR[17] SADDR[16] R/W Attribute R/W R/W R/W R/W R/W R/W R/W R/W Protection Attribute WP WP WP WP WP WP WP WP Initial Value 0 0 0 0 0 0 0 0 bit 15 14 13 12 11 10 9 8 Field SADDR[15] SADDR[14] SADDR[13] SADDR[12] SADDR[11] SADDR[10] SADDR[9] SADDR[8] R/W Attribute R/W R/W R/W R/W R/W R/W R/W R/W Protection Attribute WP WP WP WP WP WP WP WP Initial Value 0 0 0 0 0 0 0 0 bit 7 6 5 4 3 2 1 0 Field SADDR[7] SADDR[6] SADDR[5] SADDR[4] SADDR[3] SADDR[2] SADDR[1] SADDR[0] R/W Attribute R/W R/W R/W R/W R/W R/W R/W R/W Protection Attribute WP WP WP WP WP WP WP WP Initial Value 0 0 0 0 0 0 0 0 Correct) ■ MPUXSHE_SADDR1 bit 31 30 29 28 27 26 25 24 Field SADDR[31] SADDR[30] SADDR[29] SADDR[28] SADDR[27] SADDR[26] SADDR[25] SADDR[24] R/W Attribute R/W R/W R/W R/W R/W R/W R/W R/W Protection Attribute WPS WPS WPS WPS WPS WPS WPS WPS Initial Value 0 0 0 0 0 0 0 0 bit 23 22 21 20 19 18 17 16 Field SADDR[23] SADDR[22] SADDR[21] SADDR[20] SADDR[19] SADDR[18] SADDR[17] SADDR[16] R/W Attribute R/W R/W R/W R/W R/W R/W R/W R/W Protection Attribute WPS WPS WPS WPS WPS WPS WPS WPS Initial Value 0 0 0 0 0 0 0 0 bit 15 14 13 12 11 10 9 8 Field SADDR[15] SADDR[14] SADDR[13] SADDR[12] SADDR[11] SADDR[10] SADDR[9] SADDR[8] R/W Attribute R/W R/W R/W R/W R/W R/W R/W R/W Protection Attribute WPS WPS WPS WPS WPS WPS WPS WPS Initial Value 0 0 0 0 0 0 0 0 bit 7 6 5 4 3 2 1 0 Field SADDR[7] SADDR[6] SADDR[5] SADDR[4] SADDR[3] SADDR[2] SADDR[1] SADDR[0] R/W Attribute R/W R/W R/W R/W R/W R/W R/W R/W Protection Attribute WPS WPS WPS WPS WPS WPS WPS WPS Initial Value 0 0 0 0 0 0 0 0

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1433</td><td>CHAPTER 25:Memory Protection Unit for AXI of SHE 4. Registers 4.9. MPU AXI End Address Registers (MPUXSHE_EADDR1~8)</td><td>
Error)
■ MPUXSHE_EADDR1
<table>
<tr><td>bit</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td>Field</td><td>EADDR[31]</td><td>EADDR[30]</td><td>EADDR[29]</td><td>EADDR[28]</td><td>EADDR[27]</td><td>EADDR[26]</td><td>EADDR[25]</td><td>EADDR[24]</td></tr>
<tr><td>R/W Attribute</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td></tr>
<tr><td>Protection Attribute</td><td>WP</td><td>WP</td><td>WP</td><td>WP</td><td>WP</td><td>WP</td><td>WP</td><td>WP</td></tr>
<tr><td>Initial Value</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
<table>
<tr><td>bit</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td>Field</td><td>EADDR[23]</td><td>EADDR[22]</td><td>EADDR[21]</td><td>EADDR[20]</td><td>EADDR[19]</td><td>EADDR[18]</td><td>EADDR[17]</td><td>EADDR[16]</td></tr>
<tr><td>R/W Attribute</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td></tr>
<tr><td>Protection Attribute</td><td>WP</td><td>WP</td><td>WP</td><td>WP</td><td>WP</td><td>WP</td><td>WP</td><td>WP</td></tr>
<tr><td>Initial Value</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
<table>
<tr><td>bit</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td>Field</td><td>EADDR[15]</td><td>EADDR[14]</td><td>EADDR[13]</td><td>EADDR[12]</td><td>EADDR[11]</td><td>EADDR[10]</td><td>EADDR[9]</td><td>EADDR[8]</td></tr>
<tr><td>R/W Attribute</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td></tr>
<tr><td>Protection Attribute</td><td>WP</td><td>WP</td><td>WP</td><td>WP</td><td>WP</td><td>WP</td><td>WP</td><td>WP</td></tr>
<tr><td>Initial Value</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
<table>
<tr><td>bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Field</td><td>EADDR[7]</td><td>EADDR[6]</td><td>EADDR[5]</td><td>EADDR[4]</td><td>EADDR[3]</td><td>EADDR[2]</td><td>EADDR[1]</td><td>EADDR[0]</td></tr>
<tr><td>R/W Attribute</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td></tr>
<tr><td>Protection Attribute</td><td>WP</td><td>WP</td><td>WP</td><td>WP</td><td>WP</td><td>WP</td><td>WP</td><td>WP</td></tr>
<tr><td>Initial Value</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr>
</table>
Correct)
■ MPUXSHE_EADDR1
<table>
<tr><td>bit</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td>Field</td><td>EADDR[31]</td><td>EADDR[30]</td><td>EADDR[29]</td><td>EADDR[28]</td><td>EADDR[27]</td><td>EADDR[26]</td><td>EADDR[25]</td><td>EADDR[24]</td></tr>
<tr><td>R/W Attribute</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td></tr>
<tr><td>Protection Attribute</td><td>WPS</td><td>WPS</td><td>WPS</td><td>WPS</td><td>WPS</td><td>WPS</td><td>WPS</td><td>WPS</td></tr>
<tr><td>Initial Value</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
<table>
<tr><td>bit</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td>Field</td><td>EADDR[23]</td><td>EADDR[22]</td><td>EADDR[21]</td><td>EADDR[20]</td><td>EADDR[19]</td><td>EADDR[18]</td><td>EADDR[17]</td><td>EADDR[16]</td></tr>
<tr><td>R/W Attribute</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td></tr>
<tr><td>Protection Attribute</td><td>WPS</td><td>WPS</td><td>WPS</td><td>WPS</td><td>WPS</td><td>WPS</td><td>WPS</td><td>WPS</td></tr>
<tr><td>Initial Value</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
<table>
<tr><td>bit</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td>Field</td><td>EADDR[15]</td><td>EADDR[14]</td><td>EADDR[13]</td><td>EADDR[12]</td><td>EADDR[11]</td><td>EADDR[10]</td><td>EADDR[9]</td><td>EADDR[8]</td></tr>
<tr><td>R/W Attribute</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td></tr>
<tr><td>Protection Attribute</td><td>WPS</td><td>WPS</td><td>WPS</td><td>WPS</td><td>WPS</td><td>WPS</td><td>WPS</td><td>WPS</td></tr>
<tr><td>Initial Value</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
<table>
<tr><td>bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Field</td><td>EADDR[7]</td><td>EADDR[6]</td><td>EADDR[5]</td><td>EADDR[4]</td><td>EADDR[3]</td><td>EADDR[2]</td><td>EADDR[1]</td><td>EADDR[0]</td></tr>
<tr><td>R/W Attribute</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td></tr>
<tr><td>Protection Attribute</td><td>WPS</td><td>WPS</td><td>WPS</td><td>WPS</td><td>WPS</td><td>WPS</td><td>WPS</td><td>WPS</td></tr>
<tr><td>Initial Value</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr>
</table>
</td></tr>
</table>

Page	Section	Change Results
1436	CHAPTER 25:Memory Protection Unit for AXI of SHE 5. Notes on Using MPU AXI	Changed Error) **Figure 5-1 Programmer's Flowchart**  Correct) **Figure 5-1 Programmer's Flowchart** 

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1450</td><td>CHAPTER 26:Secure Hardware Extension(SHE)
2. Secure Hardware Extension Registers
2.1.1. SHE Command Register (SHE_CMD)</td><td>Added
Error)
Note:
– Initial value depends on the boot configuration.
Correct)
Notes:
– Initial value depends on the boot configuration.
– In the case of the other values except above ones, it is not in the normal status. Please take appropriate measures for the system characteristic.</td></tr>
<tr><td>1458</td><td>CHAPTER 26:Secure Hardware Extension(SHE)
2. Secure Hardware Extension Registers
2.2.1. SHE Status Register (SHE_STATUS)</td><td>Added
Error)
Note:
– Initial value depends on the boot configuration.
Correct)
Notes:
– Initial value depends on the boot configuration.
– In the case of the other values except above ones, it isnot in the normal staus. Please take appropriate measures for the system characteristic.</td></tr>
<tr><td>1507</td><td>CHAPTER 26:Secure Hardware Extension
3. Operation of the Secure Hardware Extension
3.1. Operation of the Command Interface</td><td>Changed
Error)
Table 3-2 Sequential Order for Parameter and Result Transfer
<table>
<tr><td rowspan="5">CMD_EXPORT_
RAM_KEY</td><td rowspan="5"></td><td rowspan="5">N/A</td><td rowspan="5">N/A</td><td>1</td><td>M1</td><td>128</td></tr>
<tr><td>2</td><td>M2</td><td>256</td></tr>
<tr><td>3</td><td>M3</td><td>128</td></tr>
<tr><td>4</td><td>M4</td><td>256</td></tr>
<tr><td>5</td><td>M5</td><td>128</td></tr>
</table>
Correct)
Table 3-2 Sequential Order for Parameter and Result Transfer
<table>
<tr><td rowspan="5">CMD_EXPORT_
RAM_KEY</td><td rowspan="5"></td><td rowspan="5">N/A</td><td rowspan="5">N/A</td><td>1</td><td>M1</td><td>128</td></tr>
<tr><td>2</td><td>M2</td><td>256</td></tr>
<tr><td>3</td><td>M3</td><td>128</td></tr>
<tr><td>4</td><td>M4</td><td>256</td></tr>
<tr><td>5</td><td>M5</td><td>128</td></tr>
</table></td></tr>
</table>

Page	Section	Change Results
1520	CHAPTER 26:Secure Hardware Extension 3. Operation of the Secure Hardware Extension 3.2. Operation of the Data Interface	Changed Error) **Memory Protection** For safety reasons both masters have a Memory Protection Unit (MPU) in order to protect the memory areas of the MCU. For details on MPU configuration, refer to Chapter X (Memory Protection Unit for the AXI) of the Traveo Cluster Series Hardware Manual. Correct) **Memory Protection** For safety reasons both masters have a Memory Protection Unit (MPU) in order to protect the memory areas of the MCU. For details on MPU configuration, refer to Chapter: Memory Protection Unit for the AXI of SHE.
1528	CHAPTER 26:Secure Hardware Extension 4. Notes on Using SHE 4.1.1. Notes on Using SHE Commands	Changed Error) However in some rare cases such errors are accepted by CMD_LOAD_KEY and the memory slot is updated. In that case the command CMD_LOAD key will succeed (SHE_ERC:CMDMAIN contains ERC_NO_ERROR), but the user will be notified by the warning code (other than ERC_INVALID) in the extended error code register (SHE_ERC:CMDEXTD) and can react by repeating the memory update procedure. Correct) However in some rare cases such errors are accepted by CMD_LOAD_KEY and the memory slot is updated. In that case the command CMD_LOAD_KEY will succeed (SHE_ERC:CMDMAIN contains ERC_NO_ERROR), but the user will be notified by the warning code (other than ERC_INVALID) in the extended error code register (SHE_ERC:CMDEXTD) and can react by repeating the memory update procedure.

Page	Section	Change Results
1545	CHAPTER 27 DMA Cntroller 3.Operational Description 3.1.DMA Channels	Error) **Block of Data** A block of data is determined by the setting of Block Count (DMAi_An:BC[3:0]) and Transfer Width (DMAi_Bn:TW[1:0]). The DMA Controller will make DMAi_An:BC + 1 data transfers from source address range starting at Source Address (DMAi_SAn:SA) to destination address range starting at Destination Address (DMAi_DAn:DA). If DMAi_An:BC is set to "0" a single data transfer from Source Address (DMAi_SAn:SA) to Destination Address (DMAi_DAn:DA) will be done. The settings of Block Count (DMAi_AN:BC[3:0]), Beat Limit (DMAi_An:BL[1:0]), Alternate (DMAi_An:AL), and Transfer Width (DMAi_Bn:TW[1:0]) define how the AHB master interface issues the DMAi_An:BC + 1 data transfers. The following table shows all possible combinations between DMAi_An:BC, DMAi_An:BL, and DMAi_An:AL. Only these three influence the sequence of AHB transfers, whereas DMAi_Bn:TW only affects the data size which will be transported. Correct) **Block of Data** A block of data is determined by the setting of Block Count (DMAi_An:BC[3:0]) and Transfer Width (DMAi_Bn:TW[1:0]). The DMA Controller will make DMAi_An:BC + 1 data transfers from source address range starting at Source Address (DMAi_SAn:SA) to destination address range starting at Destination Address (DMAi_DAn:DA). If DMAi_An:BC is set to "0" a single data transfer from Source Address (DMAi_SAn:SA) to Destination Address (DMAi_DAn:DA) will be done. The settings of Block Count (DMAi_An:BC[3:0]), Beat Limit (DMAi_An:BL[1:0]), Alternate (DMAi_An:AL), and Transfer Width (DMAi_Bn:TW[1:0]) define how the AHB master interface issues the DMAi_An:BC + 1 data transfers. The following table shows all possible combinations between DMAi_An:BC, DMAi_An:BL, and DMAi_An:AL. Only these three influence the sequence of AHB transfers, whereas DMAi_Bn:TW only affects the data size which will be transported.
1553	CHAPTER 27 DMA Cntroller 3.Operational Description 3.1.DMA Channels DMA Transfer Completion and Error Handling	Error) DMAi_Bn:SS = 0b000) if DMAi_Bn:DQ or DMAi_Bn:EQ is set to "1". **Source and Destination Protection** Correct) DMAi_Bn:SS = 0b000) if DMAi_Bn:DQ or DMAi_Bn:EQ is set to "1". ***Note:*** – *For new transfer request accepted during on-going DMA transfer or after previous DMA transfer completion, this DMA controller starts the next transfer by clearing the interrupt flag. Therefore, in case it is unnecessary to start DMA transfer for next transfer request, it is required to disable DMA channel and clear transfer request of DMA client, and then clear the interrupt flag.* **Source and Destination Protection**

Page	Section	Change Results
1580	CHAPTER 27 DMA Cntroller 4.Registers 4.8.DMA Controller Channel Configuration B Register Channel 'n' (DMAi_Bn)	Error) ■DEBUG == 1 and DMAi_R:DF == 1 and DMAi_R:DB == 0b01 or DMAi_R:DB == 0b10, debug event is pending while debug flag is set and Debug Behavior is set to HALT or STOP Correct) ■DEBUG == 1 and DMAi_R:DE == 1 and DMAi_R:DB == 0b01 or DMAi_R:DB == 0b10, debug event is pending while debug flag is set and Debug Behavior is set to HALT or STOP
1580	CHAPTER 27 DMA Cntroller 4.Registers 4.8.DMA Controller Channel Configuration B Register Channel 'n' (DMAi_Bn)	Error) *: Refer the explanation of DMAiAn:ST bit in DMA Controller Channel Configuration A Register Channel 'n' (DMAi_An) for the rest of the conditions. Correct) *: Refer the explanation of DMAi_An:ST bit in DMA Controller Channel Configuration A Register Channel 'n' (DMAi_An) for the rest of the conditions.
1583	CHAPTER 27 DMA Cntroller 4.Registers 4.9.DMA Controller Channel Configuration Source Address Register Channel 'n' (DMAi_SAn)	Error) ***Note:*** – *Disable the channel (i.e. DMAi_R:DE=0 or DMAiAn:EB = 0) before configuring this register.* Correct) ***Note:*** – *Disable the channel (i.e. DMAi_R:DE=0 or DMAi_An:EB = 0) before configuring this register.*
1584	CHAPTER 27 DMA Cntroller 4.Registers 4.10.DMA Controller Channel Configuration Destination Address Register Channel 'n' (DMAi_DAn)	Error) ***Note:*** – *Disable the channel (i.e. DMAi_R:DE=0 or DMAiAn:EB = 0) before configuring this register.* Correct) ***Note:*** – *Disable the channel (i.e. DMAi_R:DE=0 or DMAi_An:EB = 0) before configuring this register.*

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1592</td><td>CHAPTER 27 DMA Cntroller 4.Registers 4.16. DMA Controller Client Matrix Internal Client Interface Configuration Register 'm' (DMAi_CMICICm)</td><td>Error)
Note:
– Attribute of 27th bit (BEHSTPACK) is "R/W" in DMAi_CMICIC8 and DMAi_CMICIC9, is "R0,WX" in DMAi_CMICIC10, DMAi_CMICIC11, ... and DMAi_CMICIC(product specification).
Correct)
Notes:
– Attribute of the 27th bit (BEHSTPACK) of DMAi_CMICICm has "R/W" and "R0,WX" by difference of product specifications.
– DMAi_CMICICm registers are not implemeted depending on the product specification. In case of register access to those DMAi_CMICICm registers, response of bus error will occur
– About the implementation of these registers, please see the product's HWM.</td></tr>
<tr><td>1593</td><td>CHAPTER 27 DMA Cntroller 4.Registers 4.16. DMA Controller Client Matrix Internal Client Interface Configuration Register 'm' (DMAi_CMICICm)</td><td>Error)
[bit27] BEHSTPACK : Behavior stop acknowledge
BEHSTPACK sets the behavior of the Internal DMA Client Interface 'm' output signal DSTP_ACK[m] if the client interface is not selected in any of the Channel Configuration Registers (DMAi_CMCHICn).
Note:
– This configuration bit is read0 in DMAi_CMICICx (x=10, 11, … , (product specification))
<table><tr><th>Bit</th><th>Description</th></tr><tr><td>0</td><td>DSTP_ACK[m] outputs inactive logic level</td></tr><tr><td>1</td><td>DSTP[m] connects directly to DSTP_ACK[m]</td></tr></table>
Correct)
[bit27] BEHSTPACK : Behavior stop acknowledge
BEHSTPACK sets the behavior of the Internal DMA Client Interface 'm' output signal DSTP_ACK[m] if the client interface is not selected in any of the Channel Configuration Registers (DMAi_CMCHICn).
<table><tr><th>Bit</th><th>Description</th></tr><tr><td>0</td><td>DSTP_ACK[m] outputs inactive logic level</td></tr><tr><td>1</td><td>DSTP[m] connects directly to DSTP_ACK[m]</td></tr></table></td></tr>
</table>

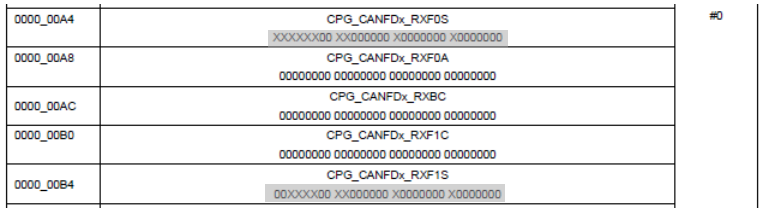
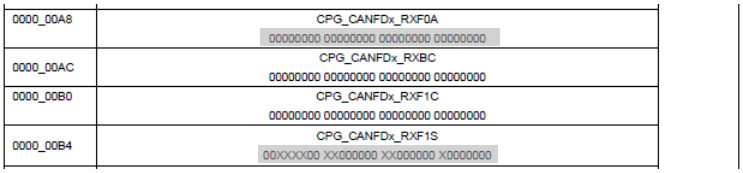
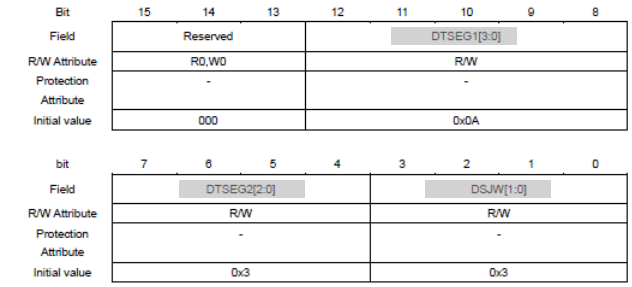
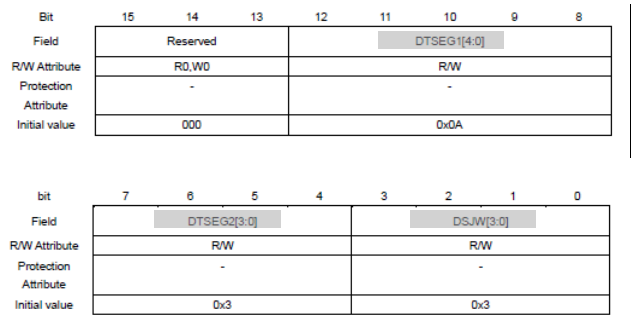
<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1605</td><td>CHAPTER 28:DMA COMPLEX SUBSYSTEM 3.Operation of the DMA COMPLEX SUBSYSTEM 3.3.Reload Timer in DMA Additional Control 3.3.1.Reload Timer Trigger Inputs</td><td>Error)
Table 3 1 Reload Timer TIN Input Selection<table><tr><td>HBUSREQ</td><td>HBUSREQ is asserted to "1" at transfer starts. HBUSREQ is de-asserted to "0" at transfer completion of current block. By referring to HBUSREQ, "transfer gap" can be observed. Please refer to "CHAPTER : DMA Controller" for the "transfer gap".</td></tr><tr><td>Software trigger</td><td>When bit 'j' of DMAn_RTSSSR is written, a "0"→"1"→"0" pulse will be given to TIN of ReloadTimer channel j. Using the DMAAn_RTSSSR, all of four Reload Timer can be triggered at the same time.
Software trigger is always able to be used regardless DMAAn_RTTSR setting.</td></tr></table>Correct)
Table 3 1 Reload Timer TIN Input Selection<table><tr><td>HBUSREQ</td><td>HBUSREQ is asserted to "1" at transfer starts. HBUSREQ is de-asserted to "0" at transfer completion of current block. By referring to HBUSREQ, "transfer gap" can be observed. Please refer to "CHAPTER : DMA Controller" for the "transfer gap".</td></tr><tr><td>Software trigger</td><td>When bit 'j' of DMAAn_RTSSSR is written, a "0"→"1"→"0" pulse will be given to TIN of ReloadTimer channel j. Using the DMAAn_RTSSSR, all of four Reload Timer can be triggered at the same time.
Software trigger is always able to be used regardless DMAAn_RTTSR setting.</td></tr></table></td></tr>
<tr><td>1636</td><td>CHAPTER 28:DMA COMPLEX SUBSYSTEM 8.Registers in DMA COMPLEX SUBSYSTEM 8.2.Timer Control Status Register (DMAAn_RLTm_TMCSR)</td><td>Error)
– This bit is enable at
✧ from ch. 0 to ch. 3 of CPER#2
✧ from ch.0 to ch. 1 of MCU_CONFIG_GROUP.
Correct)
– This bit is enable at
✧ from ch. 0 to ch. 3 of CPERI#2
✧ from ch.0 to ch. 1 of MCU_CONFIG_GROUP.</td></tr>
</table>

Page	Section	Change Results
1649	CHAPTER 29:MPU16 AHB 2. MPU16 AHB Registers 2.1. MPU16 AHB Control Register (MPUHn_CTRL0)	Error) - *The register bits MPUHn_CTRL0:AP[2:0], MPUHm_CTRL0:POEN and MPUHn_CTRL0:PROT can only be written when the MPU is disabled (MPUHn_CTRL0:MPUEN = 0).* Correct) - *The register bits MPUHn_CTRL0:AP[2:0], MPUHn_CTRL0:POEN and MPUHn_CTRL0:PROT can only be written when the MPU is disabled (MPUHn_CTRL0:MPUEN = 0).*
1655	CHAPTER 29:MPU16 AHB 2. MPU16 AHB Registers 2.6. MPU16 AHB Start Address Registers (MPUHn_SADDR1 to 16)	Error) - *the MPUHn_SADDR1 to 16 registers can only be written when the corresponding region is disabled (MPUHm_CTRL1 to 16:MPUEN = 0) or the MPU is disabled (MPUHn_CTRL0:MPUEN = 0).* Correct) - *the MPUHn_SADDR1 to 16 registers can only be written when the corresponding region is disabled (MPUHn_CTRL1 to 16:MPUEN = 0) or the MPU is disabled (MPUHn_CTRL0:MPUEN = 0).*
1656	CHAPTER 29:MPU16 AHB 2. MPU16 AHB Registers 2.7. MPU16 AHB End Address Registers (MPUHn_EADDR1 to 16)	Error) **Table 2-8 MPU16 AHB End Address Register for Region 1 (MPUHm_EADDR1) Bits** Correct) **Table 2-8 MPU16 AHB End Address Register for Region 1 (MPUHn_EADDR1) Bits**

Page	Section	Change Results
1657	CHAPTER 29:MPU16 AHB 2. MPU16 AHB Registers 2.8. MPU16 AHB Unlock Register (MPUHn_UNLOCK)	Error) Unlock value:(product specification) Lock value: (product specification) Correct) Unlock value: 0xACCABB56 Lock value: 0x112ABB56
1659	CHAPTER 29:MPU16 AHB 3. Operation of the MPU16 AHB	Error) End Addresses are specified by registers MPUHn_EADDR1 to MPUHn_EADDR16n for region 1 to region 16 respectively. Correct) End Addresses are specified by registers MPUHn_EADDR1 to MPUHn_EADDR16 for region 1 to region 16 respectively.
1686	CHAPTER 30:CAN FD Controller(MCAN3.0.1) 3. Explanation of Operations 3.4. Rx Handling 3.4.2. Rx FIFOs 3.4.2.2 Rx FIFO Overwrite Mode	Error) After reading from the Rx FIFO, the number of the last element read has to be written to the Rx FIFO Acknowledge Index (RXFnA.FnA[5:0]). Correct) After reading from the Rx FIFO, the number of the last element read has to be written to the Rx FIFO Acknowledge Index (RXFnA.FnAI[5:0]).

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1865</td><td>CHAPTER 31:CAN FD Controller(MCAN3.2)
3. Explanation of Operations
3.7. Configuring the CAN Bit Timing
3.7.2. CAN Bit Rates</td><td>
Error)
Table 3 8 Example Configurations for Nominal Bit Rates
<table>
<tr><td>CAN clock frequency</td><td colspan="2">8MHz</td><td colspan="2">10MHz</td><td colspan="2">16MHz</td><td colspan="2">20MHz</td><td colspan="2">32MHz</td><td colspan="2">40MHz</td></tr>
<tr><td>configuration
nominal bit rate</td><td>Number of tqs per bit time</td><td>NBTP.NBRP + 1</td><td>Number of tqs per bit time</td><td>NBTP.NBRP + 1</td><td>Number of tqs per bit time</td><td>NBTP.NBRP + 1</td><td>Number of tqs per bit time</td><td>NBTP.NBRP + 1</td><td>Number of tqs per bit time</td><td>NBTP.NBRP + 1</td><td>Number of tqs per bit time</td><td>NBTP.NBRP + 1</td></tr>
<tr><td>125Kbps</td><td>64tq
32tq
16tq
8tq</td><td>1
2
4
8</td><td>80tq
40tq
20tq
10tq</td><td>1
2
4
8</td><td>128tq
64tq
32tq
16tq
8tq</td><td>1
2
4
8
16</td><td>160tq
80tq
40tq
20tq
10tq</td><td>1
2
4
8
16</td><td>256tq
128tq
64tq
32tq
16tq
8tq</td><td>1
2
4
8
16
32</td><td>320tq
160tq
80tq
40tq
20tq
10tq</td><td>1
2
4
8
16
32</td></tr>
<tr><td>250Kbps</td><td>32tq
16tq
8tq</td><td>1
2
4</td><td>40tq
20tq
10tq</td><td>1
2
4</td><td>64tq
32tq
16tq
8tq</td><td>1
2
4
8</td><td>80tq
40tq
20tq
10tq</td><td>1
2
4
8</td><td>128tq
64tq
32tq
16tq
8tq</td><td>1
2
4
8
16</td><td>160tq
80tq
40tq
20tq
10tq</td><td>1
2
4
8
16</td></tr>
<tr><td>500Kbps</td><td>16tq
8tq</td><td>1
2</td><td>20tq
10tq</td><td>1
2</td><td>32tq
16tq
8tq</td><td>1
2
4</td><td>40tq
20tq
10tq</td><td>1
2
4</td><td>64tq
32tq
16tq
8tq</td><td>1
2
4
8</td><td>80tq
40tq
20tq
10tq</td><td>1
2
4
8</td></tr>
<tr><td>1Mbps</td><td>16tq</td><td>1</td><td>20tq</td><td>1</td><td>16tq
8tq</td><td>1
2</td><td>20tq
10tq</td><td>1
2</td><td>32tq
16tq
8tq</td><td>1
2
4</td><td>40tq
20tq
10tq</td><td>1
2
4</td></tr>
</table>
Correct)
Table 3 8 Example Configurations for Nominal Bit Rates
<table>
<tr><td>CAN clock frequency</td><td colspan="2">8MHz</td><td colspan="2">10MHz</td><td colspan="2">16MHz</td><td colspan="2">20MHz</td><td colspan="2">32MHz</td><td colspan="2">40MHz</td></tr>
<tr><td>configuration
nominal bit rate</td><td>Number of tqs per bit time</td><td>NBTP.NBRP + 1</td><td>Number of tqs per bit time</td><td>NBTP.NBRP + 1</td><td>Number of tqs per bit time</td><td>NBTP.NBRP + 1</td><td>Number of tqs per bit time</td><td>NBTP.NBRP + 1</td><td>Number of tqs per bit time</td><td>NBTP.NBRP + 1</td><td>Number of tqs per bit time</td><td>NBTP.NBRP + 1</td></tr>
<tr><td>125Kbps</td><td>64tq
32tq
16tq
8tq</td><td>1
2
4
8</td><td>80tq
40tq
20tq
10tq</td><td>1
2
4
8</td><td>128tq
64tq
32tq
16tq
8tq</td><td>1
2
4
8
16</td><td>160tq
80tq
40tq
20tq
10tq</td><td>1
2
4
8
16</td><td>256tq
128tq
64tq
32tq
16tq
8tq</td><td>1
2
4
8
16
32</td><td>320tq
160tq
80tq
40tq
20tq
10tq</td><td>1
2
4
8
16
32</td></tr>
<tr><td>250Kbps</td><td>32tq
16tq
8tq</td><td>1
2
4</td><td>40tq
20tq
10tq</td><td>1
2
4</td><td>64tq
32tq
16tq
8tq</td><td>1
2
4
8</td><td>80tq
40tq
20tq
10tq</td><td>1
2
4
8</td><td>128tq
64tq
32tq
16tq
8tq</td><td>1
2
4
8
16</td><td>160tq
80tq
40tq
20tq
10tq</td><td>1
2
4
8
16</td></tr>
<tr><td>500Kbps</td><td>16tq
8tq</td><td>1
2</td><td>20tq
10tq</td><td>1
2</td><td>32tq
16tq
8tq</td><td>1
2
4</td><td>40tq
20tq
10tq</td><td>1
2
4</td><td>64tq
32tq
16tq
8tq</td><td>1
2
4
8</td><td>80tq
40tq
20tq
10tq</td><td>1
2
4
8</td></tr>
<tr><td>1Mbps</td><td>8tq</td><td>1</td><td>10tq</td><td>1</td><td>16tq
8tq</td><td>1
2</td><td>20tq
10tq</td><td>1
2</td><td>32tq
16tq
8tq</td><td>1
2
4</td><td>40tq
20tq
10tq</td><td>1
2
4</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1891</td><td>CHAPTER 31:CAN FD Controller(MCAN3.2)
5. Registers</td><td>Error)
Common Peripheral #0 Group (Channel No.: 0, 1, 2, 3, and 4)
<table>
<tr><th rowspan="2">Offset</th><th colspan="4">Register Name / Initial Value</th><th rowspan="2">Group Name</th></tr>
<tr><th>+3</th><th>+2</th><th>+1</th><th>+0</th></tr>
<tr><td>0000_0000</td><td colspan="4">CPG_CANFDx_CREL
00110000 00010011 00000101 00000110</td><td rowspan="17">Common Peripheral #0</td></tr>
<tr><td>0000_0004</td><td colspan="4">CPG_CANFDx_ENDN
10000111 01100101 01000011 00100001</td></tr>
<tr><td>0000_0008</td><td colspan="4">-</td></tr>
<tr><td>0000_000C</td><td colspan="4">CPG_CANFDx_FBTP
00000000 00000000 00001010 00110011</td></tr>
<tr><td>0000_0010</td><td colspan="4">CPG_CANFDx_TEST
00000000 00000000 00000000 *0000000</td></tr>
<tr><td>0000_0014</td><td colspan="4">CPG_CANFDx_RWD
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_0018</td><td colspan="4">CPG_CANFDx_CCCR
00000000 00000000 00000000 00000001</td></tr>
<tr><td>0000_001C</td><td colspan="4">CPG_CANFDx_BTP
00000000 00000000 00001010 00110011</td></tr>
<tr><td>0000_0020</td><td colspan="4">CPG_CANFDx_TSCC
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_0024</td><td colspan="4">CPG_CANFDx_TSCV
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_0028</td><td colspan="4">CPG_CANFDx_TOCC
11111111 11111111 00000000 00000000</td></tr>
<tr><td>0000_002C</td><td colspan="4">CPG_CANFDx_TOCV
00000000 00000000 11111111 11111111</td></tr>
<tr><td>0000_0030
|
0000_003C</td><td colspan="4">-</td></tr>
<tr><td>0000_0040</td><td colspan="4">CPG_CANFDx_ECR
XXXXXXXX 00000000 00000000 00000000</td></tr>
<tr><td>0000_0044</td><td colspan="4">CPG_CANFDx_PSR
XXXXXXXX XXXXXXXX XX000111 00000111</td></tr>
<tr><td>0000_0048</td><td colspan="4">-</td></tr>
<tr><td>0000_004C</td><td colspan="4"></td></tr>
</table>
Correct)
Common Peripheral #0 Group (Channel No.: 0, 1, 2, 3, and 4)
<table>
<tr><th rowspan="2">Offset</th><th colspan="4">Register Name / Initial Value</th><th rowspan="2">Group Name</th></tr>
<tr><th>+3</th><th>+2</th><th>+1</th><th>+0</th></tr>
<tr><td>0000_0000</td><td colspan="4">CPG_CANFDx_CREL
00110010 00000100 00010010 00011000</td><td rowspan="17">Common Peripheral #0</td></tr>
<tr><td>0000_0004</td><td colspan="4">CPG_CANFDx_ENDN
10000111 01100101 01000011 00100001</td></tr>
<tr><td>0000_0008</td><td colspan="4">-</td></tr>
<tr><td>0000_000C</td><td colspan="4">CPG_CANFDx_DBTP
00000000 00000000 00001010 00110011</td></tr>
<tr><td>0000_0010</td><td colspan="4">CPG_CANFDx_TEST
00000000 00000000 00000000 *0000000</td></tr>
<tr><td>0000_0014</td><td colspan="4">CPG_CANFDx_RWD
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_0018</td><td colspan="4">CPG_CANFDx_CCCR
00000000 00000000 00000000 00000001</td></tr>
<tr><td>0000_001C</td><td colspan="4">CPG_CANFDx_NBTP
00000110 00000000 00001010 00000011</td></tr>
<tr><td>0000_0020</td><td colspan="4">CPG_CANFDx_TSCC
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_0024</td><td colspan="4">CPG_CANFDx_TSCV
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_0028</td><td colspan="4">CPG_CANFDx_TOCC
11111111 11111111 00000000 00000000</td></tr>
<tr><td>0000_002C</td><td colspan="4">CPG_CANFDx_TOCV
00000000 00000000 11111111 11111111</td></tr>
<tr><td>0000_0030
|
0000_003C</td><td colspan="4">-</td></tr>
<tr><td>0000_0040</td><td colspan="4">CPG_CANFDx_ECR
XXXXXXXX 00000000 00000000 00000000</td></tr>
<tr><td>0000_0044</td><td colspan="4">CPG_CANFDx_PSR
XXXXXXXX XXXXXXXX XX000111 00000111</td></tr>
<tr><td>0000_0048</td><td colspan="4">CPG_CANFDx_TDCR
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_004C</td><td colspan="4">-</td></tr>
</table>
</td></tr>
</table>

Page	Section	Change Results
1892	CHAPTER 31:CAN FD Controller(MCAN3.2) 5. Registers	Error)  Correct) 
1897	CHAPTER 31:CAN FD Controller(MCAN3.2) 5. Registers 5.3. Data Bit Timing & Prescaler Register (MCG_CANFDx_DBTP, CPG_CANFDx_DBTP)	Error)  Correct) 

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1898</td><td>CHAPTER 31:CAN FD Controller(MCAN3.2) 5. Registers 5.3. Data Bit Timing & Prescaler Register (MCG_CANFDx_DBTP, CPG_CANFDx_DBTP)</td><td>Error)
[bit12:8] DTSEG1[3:0]: Data Time Segment before sample Point
<table><tr><th>DTSEG1[4:0]</th><th>Description</th></tr><tr><td>0x00-0x1F</td><td>Valid values are 0 to 31. The actual interpretation by the hardware of this value is such that one more than the programmed value is used.</td></tr></table>
Correct)
[bit12:8] DTSEG1[4:0]: Data Time Segment before sample Point
<table><tr><th>DTSEG1[4:0]</th><th>Description</th></tr><tr><td>0x00-0x1F</td><td>Valid values are 0 to 31. The actual interpretation by the hardware of this value is such that one more than the programmed value is used.</td></tr></table></td></tr>
<tr><td>1903</td><td>CHAPTER 31:CAN FD Controller(MCAN3.2) 5. Registers 5.6. CC Control Register (MCG_CANFDx_CCCR, CPG_CANFDx_CCCR)</td><td>Error)
<table><tr><td>bit</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr><tr><td>Field</td><td>NISO</td><td>TXP</td><td>EFBI</td><td>PXHD</td><td colspan="2">Reserved</td><td>BRSE</td><td>FDOE</td></tr><tr><td>R/W Attribute</td><td>R0,W0</td><td>R/W</td><td>R,WX</td><td>R,WX</td><td colspan="2">R/W</td><td>R/W</td><td>R/W</td></tr><tr><td>Protection Attribute</td><td>-</td><td>-</td><td>-</td><td>-</td><td colspan="2">-</td><td>-</td><td>-</td></tr><tr><td>Initial value</td><td>0</td><td>0</td><td>0</td><td>0</td><td colspan="2">00</td><td>0</td><td>0</td></tr></table>
Correct)
<table><tr><td>bit</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr><tr><td>Field</td><td>NISO</td><td>TXP</td><td>EFBI</td><td>PXHD</td><td colspan="2">Reserved</td><td>BRSE</td><td>FDOE</td></tr><tr><td>R/W Attribute</td><td>R0,W0</td><td>R/W</td><td>R,WX</td><td>R,WX</td><td colspan="2">R0,W0</td><td>R/W</td><td>R/W</td></tr><tr><td>Protection Attribute</td><td>-</td><td>-</td><td>-</td><td>-</td><td colspan="2">-</td><td>-</td><td>-</td></tr><tr><td>Initial value</td><td>0</td><td>0</td><td>0</td><td>0</td><td colspan="2">00</td><td>0</td><td>0</td></tr></table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1904</td><td>CHAPTER 31:CAN FD Controller(MCAN3.2)
5. Registers
5.6. CC Control Register (MCG_CANFDx_CCCR, CPG_CANFDx_CCCR)</td><td>Error)
[bit13] EFBI: Edge Filtering during Bus Integration
Write access to this bit is possible only when the Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) bits are both "1".
<table><tr><th>Bit</th><th>Description</th></tr><tr><td>0</td><td>Edge filtering disabled.</td></tr><tr><td>1</td><td>Two consecutive dominant tq required to detect an edge for hard synchronization.</td></tr></table>
Correct)
[bit13] EFBI: Edge Filtering during Bus Integration
Write access to this bit is possible only when the Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) bits are both "1".
<table><tr><th>Bit</th><th>Description</th></tr><tr><td>0</td><td>Edge filtering disabled.</td></tr><tr><td>1</td><td>Two consecutive dominant tq required to detect an edge for hard synchronization.</td></tr></table>
Note:
– In case that all of the following conditions are fulfilled, it may happen that the CAN FD Controller synchronizes itself wrongly. And it will rate the received CAN FD frame as faulty and error frame will be send. When that frame is retransmitted, Bus integration has finished and the frame is received correctly.
– when edge filtering is activated and
– when the end of Bus integration coincides with a falling edge (or dominant) at pin canfd_rx, and a CAN FD frame reception is started</td></tr>
<tr><td>1907</td><td>CHAPTER 31:CAN FD Controller(MCAN3.2)
5. Registers
5.7. Nominal Bit Timing & Prescaler Register (MCG_CANFDx_NBTP, CPG_CANFDx_NBTP)</td><td>Error)
<table><tr><td>Bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr><tr><td>Field</td><td>Reserved</td><td colspan="7">NTSEG2[6:0]</td></tr><tr><td>R/W Attribute</td><td>R0,WX</td><td colspan="7">R/W</td></tr><tr><td>Protection Attribute</td><td>-</td><td colspan="7">-</td></tr><tr><td>Initial value</td><td>0</td><td colspan="7">0x03</td></tr></table>
Correct)
<table><tr><td>Bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr><tr><td>Field</td><td>Reserved</td><td colspan="7">NTSEG2[6:0]</td></tr><tr><td>R/W Attribute</td><td>R0,W0</td><td colspan="7">R/W</td></tr><tr><td>Protection Attribute</td><td>-</td><td colspan="7">-</td></tr><tr><td>Initial value</td><td>0</td><td colspan="7">0x03</td></tr></table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1908</td><td>CHAPTER 31:CAN FD Controller(MCAN3.2) 5. Registers 5.7. Nominal Bit Timing & Prescaler Register (MCG_CANFDx_NBTP, CPG_CANFDx_NBTP)</td><td>Error)
[bit6:0] NTSEG2[6:0]: Nominal Time Segment after sample Point
<table><tr><th>NTSEG2[6:0]</th><th>Description</th></tr><tr><td>0x00-0x7F</td><td>Valid values are 0 to 127. The actual interpretation by the hardware of this value is such that one more than the programmed value is used.</td></tr></table>
Correct)
[bit6:0] NTSEG2[6:0]: Nominal Time Segment after sample Point
<table><tr><th>NTSEG2[6:0]</th><th>Description</th></tr><tr><td>0x01-0x7F</td><td>Valid values are 1 to 127. The value 0 must not be used. The actual interpretation by the hardware of this value is such that one more than the programmed value is used.</td></tr></table></td></tr>
<tr><td>1926</td><td>CHAPTER 31:CAN FD Controller(MCAN3.2) 5. Registers 5.15. Interrupt Register (MCG_CANFDx_IR, CPG_CANFDx_IR)</td><td>Error)
[bit28] PED: Protocol Error in Data Phase
<table><tr><th>Bit</th><th>Description</th></tr><tr><td>0</td><td>No protocol error in data phase.</td></tr><tr><td>1</td><td>Protocol error in data phase detected (Data Phase Last Error Code PSR.DLEC[2:0] ≠ 0, 7).</td></tr></table>
Correct)
[bit28] PED: Protocol Error in Data Phase (Data Bit Time is Used)
<table><tr><th>Bit</th><th>Description</th></tr><tr><td>0</td><td>No protocol error in data phase.</td></tr><tr><td>1</td><td>Protocol error in data phase detected (Data Phase Last Error Code PSR.DLEC[2:0] ≠ 0, 7).</td></tr></table></td></tr>
<tr><td>1926</td><td>CHAPTER 31:CAN FD Controller(MCAN3.2) 5. Registers 5.15. Interrupt Register (MCG_CANFDx_IR, CPG_CANFDx_IR)</td><td>Error)
[bit27] PEA: Protocol Error in Arbitration Phase
<table><tr><th>Bit</th><th>Description</th></tr><tr><td>0</td><td>No protocol error in arbitration phase.</td></tr><tr><td>1</td><td>Protocol error in arbitration phase detected (Last Error Code PSR.LEC[2:0] ≠ 0, 7).</td></tr></table>
Correct)
[bit27] PEA: Protocol Error in Arbitration Phase (Nominal Bit Time is Used)
<table><tr><th>Bit</th><th>Description</th></tr><tr><td>0</td><td>No protocol error in arbitration phase.</td></tr><tr><td>1</td><td>Protocol error in arbitration phase detected (Last Error Code PSR.LEC[2:0] ≠ 0, 7).</td></tr></table></td></tr>
</table>

Page	Section	Change Results
2006	CHAPTER 32:External Time Stamp Counter For CAN FD 5. Register 5.1. Time Stamp Control Register (MCG_CANFDx_TSCNTR, CPG_CANFDx_TSCNTR)	Changed Error) **5.1. Time Stamp Control Register (MCG_CANFDx_TSCNTR, CPG_CANFDx_TSCNTR)** (MCG_CANFD0_TSCNTR, MCG_CANFD1_TSCNTR, MCGG_CANFD2_TSCNTR, CPG_CANFD0_TSCNTR, CPG_CANFD1_TSCNTR, CPG_CANFD2_TSCNTR, CPG_CANFD3_TSCNTR, CPG_CANFD4_TSCNTR) Correct) **5.1. Time Stamp Control Register (MCG_CANFDx_TSCNTR, CPG_CANFDx_TSCNTR)** (MCG_CANFD0_TSCNTR, MCG_CANFD1_TSCNTR, MCG_CANFD2_TSCNTR, CPG_CANFD0_TSCNTR, CPG_CANFD1_TSCNTR, CPG_CANFD2_TSCNTR, CPG_CANFD3_TSCNTR, CPG_CANFD4_TSCNTR)
2007	CHAPTER 32:External Time Stamp Counter For CAN FD 5. Register 5.2. Time Stamp Mode Register(MCG_CANFDx_TSMDR, CPG_CANFDx_TSMDR)	Changed Error) **5.2. Time Stamp Mode Register(MCG_CANFDx_TSMDR, MCG_CANFDx_TSMDR)** Correct) **5.2. Time Stamp Mode Register(MCG_CANFDx_TSMDR, CPG_CANFDx_TSMDR)**

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>2069</td><td>CHAPTER 35:Overview of Multi-function Serial Interface
6. Registers of the Multi-Function Serial Interface</td><td>
Error)
<table>
<tr><th>Registers</th><th>Mode</th><th>bit15</th><th>bit14</th><th>bit13</th><th>bit12</th><th>bit11</th><th>bit10</th><th>bit9</th><th>bit8</th><th>bit7</th><th>bit6</th><th>bit5</th><th>bit4</th><th>bit3</th><th>bit2</th><th>bit1</th><th>bit0</th></tr>
<tr><td rowspan="3">SCR /SMR</td><td>0/1</td><td>UPCL</td><td>-</td><td>-</td><td>RIE</td><td>TIE</td><td>TBIE</td><td>RXE</td><td>TXE</td><td>MD2</td><td>MD1</td><td>MD0</td><td>WUCR</td><td>SBL</td><td>BDS</td><td>-</td><td>SOE</td></tr>
<tr><td>2</td><td>UPCL</td><td>MS</td><td>SPI</td><td>RIE</td><td>TIE</td><td>TBIE</td><td>RXE</td><td>TXE</td><td>MD2</td><td>MD1</td><td>MD0</td><td>WUCR</td><td>SCINV</td><td>BDS</td><td>SCKE</td><td>SOE</td></tr>
<tr><td>3</td><td>UPCL</td><td>MS</td><td>LBR</td><td>RIE</td><td>TIE</td><td>TBIE</td><td>RXE</td><td>TXE</td><td>MD2</td><td>MD1</td><td>MD0</td><td>WUCR</td><td>SBL</td><td>-</td><td>-</td><td>SOE</td></tr>
<tr><td>IBCR /SMR</td><td>4</td><td>MSS</td><td>ACT/ SCC</td><td>SCKE</td><td>WSEL</td><td>CNDE</td><td>INTE</td><td>BER</td><td>INT</td><td>MD2</td><td>MD1</td><td>MD0</td><td>WUCR</td><td>RIE</td><td>TIE</td><td>-</td><td>-</td></tr>
<tr><td rowspan="3">SSR /ESCR</td><td>0/1</td><td>REC</td><td>-</td><td>PE</td><td>FRE</td><td>ORE</td><td>RDRF</td><td>TDRE</td><td>TBI</td><td>FLWEN</td><td>ESBL</td><td>INV</td><td>PEN</td><td>P</td><td>L2</td><td>L1</td><td>L0</td></tr>
<tr><td>2</td><td>REC</td><td>-</td><td>ES</td><td>AWC</td><td>ORE</td><td>RDRF</td><td>TDRE</td><td>TBI</td><td>SOP</td><td>L3</td><td>CSFE</td><td>WT1</td><td>WT0</td><td>L2</td><td>L1</td><td>L0</td></tr>
<tr><td>3</td><td>REC</td><td>-</td><td>LBD</td><td>FRE</td><td>ORE</td><td>RDRF</td><td>TDRE</td><td>TBI</td><td>-</td><td>ESBL</td><td>LBL2</td><td>LBIE</td><td>LBL1</td><td>LBL0</td><td>DEL1</td><td>DEL0</td></tr>
<tr><td>SSR/IBSR</td><td>4</td><td>REC</td><td>TEST</td><td>DMA</td><td>TBIE</td><td>ORE</td><td>RDRF</td><td>TDRE</td><td>TBI</td><td>FBT</td><td>RACK</td><td>RSA</td><td>TRX</td><td>AL</td><td>RSC</td><td>SPC</td><td>BB</td></tr>
<tr><td rowspan="4">TDR1/0 (RDR1/0)</td><td>0/1</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>D8</td><td>D7</td><td>D6</td><td>D5</td><td>D4</td><td>D3</td><td>D2</td><td>D1</td><td>D0</td></tr>
<tr><td>2</td><td>D15</td><td>D14</td><td>D13</td><td>D12</td><td>D11</td><td>D10</td><td>D9</td><td>D8</td><td>D7</td><td>D6</td><td>D5</td><td>D4</td><td>D3</td><td>D2</td><td>D1</td><td>D0</td></tr>
<tr><td>3</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>D7</td><td>D6</td><td>D5</td><td>D4</td><td>D3</td><td>D2</td><td>D1</td><td>D0</td></tr>
<tr><td>4</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>D7</td><td>D6</td><td>D5</td><td>D4</td><td>D3</td><td>D2</td><td>D1</td><td>D0</td></tr>
<tr><td>-</td><td>0/1</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>TDR3/2 (RDR3/2)</td><td>2</td><td>D31</td><td>D30</td><td>D29</td><td>D28</td><td>D27</td><td>D26</td><td>D25</td><td>D24</td><td>D23</td><td>D22</td><td>D21</td><td>D20</td><td>D19</td><td>D18</td><td>D17</td><td>D16</td></tr>
<tr><td rowspan="2">-</td><td>3</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>4</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td rowspan="4">SACSR 1/0</td><td>0/1</td><td>STST</td><td>-</td><td>-</td><td>-</td><td>-</td><td>TRG1</td><td>TRG0</td><td>TINT</td><td>TINTE</td><td>TSYNE</td><td>TRGE</td><td>TDIV3</td><td>TDIV2</td><td>TDIV1</td><td>TDIV0</td><td>TMRE</td></tr>
<tr><td>2</td><td>STST</td><td>-</td><td>TBEEN</td><td>CSEIE</td><td>CSE</td><td>TRG1</td><td>TRG0</td><td>TINT</td><td>TINTE</td><td>TSYNE</td><td>TRGE</td><td>TDIV3</td><td>TDIV2</td><td>TDIV1</td><td>TDIV0</td><td>TMRE</td></tr>
<tr><td>3</td><td>STST</td><td>BST</td><td>SFD</td><td>SFDE</td><td>AUTE</td><td>TRG1</td><td>TRG0</td><td>TINT</td><td>TINTE</td><td>-</td><td>TRGE</td><td>TDIV3</td><td>TDIV2</td><td>TDIV1</td><td>TDIV0</td><td>TMRE</td></tr>
<tr><td>4</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>TRG1</td><td>TRG0</td><td>TINT</td><td>TINTE</td><td>-</td><td>TRGE</td><td>TDIV3</td><td>TDIV2</td><td>TDIV1</td><td>TDIV0</td><td>TMRE</td></tr>
<tr><td rowspan="4">STMR 1/0</td><td>0/1</td><td>TM15</td><td>TM14</td><td>TM13</td><td>TM12</td><td>TM11</td><td>TM10</td><td>TM9</td><td>TM8</td><td>TM7</td><td>TM6</td><td>TM5</td><td>TM4</td><td>TM3</td><td>TM2</td><td>TM1</td><td>TM0</td></tr>
<tr><td>2</td><td>TM15</td><td>TM14</td><td>TM13</td><td>TM12</td><td>TM11</td><td>TM10</td><td>TM9</td><td>TM8</td><td>TM7</td><td>TM6</td><td>TM5</td><td>TM4</td><td>TM3</td><td>TM2</td><td>TM1</td><td>TM0</td></tr>
<tr><td>3</td><td>TM15</td><td>TM14</td><td>TM13</td><td>TM12</td><td>TM11</td><td>TM10</td><td>TM9</td><td>TM8</td><td>TM7</td><td>TM6</td><td>TM5</td><td>TM4</td><td>TM3</td><td>TM2</td><td>TM1</td><td>TM0</td></tr>
<tr><td>4</td><td>TM15</td><td>TM14</td><td>TM13</td><td>TM12</td><td>TM11</td><td>TM10</td><td>TM9</td><td>TM8</td><td>TM7</td><td>TM6</td><td>TM5</td><td>TM4</td><td>TM3</td><td>TM2</td><td>TM1</td><td>TM0</td></tr>
<tr><td rowspan="4">STMCR 1/0</td><td>0/1</td><td>TC15</td><td>TC14</td><td>TC13</td><td>TC12</td><td>TC11</td><td>TC10</td><td>TC9</td><td>TC8</td><td>TC7</td><td>TC6</td><td>TC5</td><td>TC4</td><td>TC3</td><td>TC2</td><td>TC1</td><td>TC0</td></tr>
<tr><td>2</td><td>TC15</td><td>TC14</td><td>TC13</td><td>TC12</td><td>TC11</td><td>TC10</td><td>TC9</td><td>TC8</td><td>TC7</td><td>TC6</td><td>TC5</td><td>TC4</td><td>TC3</td><td>TC2</td><td>TC1</td><td>TC0</td></tr>
<tr><td>3</td><td>TC15</td><td>TC14</td><td>TC13</td><td>TC12</td><td>TC11</td><td>TC10</td><td>TC9</td><td>TC8</td><td>TC7</td><td>TC6</td><td>TC5</td><td>TC4</td><td>TC3</td><td>TC2</td><td>TC1</td><td>TC0</td></tr>
<tr><td>4</td><td>TC15</td><td>TC14</td><td>TC13</td><td>TC12</td><td>TC11</td><td>TC10</td><td>TC9</td><td>TC8</td><td>TC7</td><td>TC6</td><td>TC5</td><td>TC4</td><td>TC3</td><td>TC2</td><td>TC1</td><td>TC0</td></tr>
<tr><td>-</td><td>0/1</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>SCSCR 1/0</td><td>2</td><td>SST1</td><td>SST0</td><td>SED1</td><td>SED0</td><td>SCD1</td><td>SCD0</td><td>-</td><td>CDIV2</td><td>CDIV1</td><td>CDIV0</td><td>CSLVL</td><td>CSEN3</td><td>CSEN2</td><td>CSEN1</td><td>CSEN0</td><td>CSOE</td></tr>
</table>
Correct)
<table>
<tr><th>Registers</th><th>Mode</th><th>bit15</th><th>bit14</th><th>bit13</th><th>bit12</th><th>bit11</th><th>bit10</th><th>bit9</th><th>bit8</th><th>bit7</th><th>bit6</th><th>bit5</th><th>bit4</th><th>bit3</th><th>bit2</th><th>bit1</th><th>bit0</th></tr>
<tr><td rowspan="3">SCR /SMR</td><td>0/1</td><td>UPCL</td><td>-</td><td>-</td><td>RIE</td><td>TIE</td><td>TBIE</td><td>RXE</td><td>TXE</td><td>MD2</td><td>MD1</td><td>MD0</td><td>WUCR</td><td>SBL</td><td>BDS</td><td>-</td><td>SOE</td></tr>
<tr><td>2</td><td>UPCL</td><td>MS</td><td>SPI</td><td>RIE</td><td>TIE</td><td>TBIE</td><td>RXE</td><td>TXE</td><td>MD2</td><td>MD1</td><td>MD0</td><td>WUCR</td><td>SCINV</td><td>BDS</td><td>SCKE</td><td>SOE</td></tr>
<tr><td>3</td><td>UPCL</td><td>MS</td><td>LBR</td><td>RIE</td><td>TIE</td><td>TBIE</td><td>RXE</td><td>TXE</td><td>MD2</td><td>MD1</td><td>MD0</td><td>WUCR</td><td>SBL</td><td>-</td><td>-</td><td>SOE</td></tr>
<tr><td>IBCR /SMR</td><td>4</td><td>MSS</td><td>ACT/ SCC</td><td>SCKE</td><td>WSEL</td><td>CNDE</td><td>INTE</td><td>BER</td><td>INT</td><td>MD2</td><td>MD1</td><td>MD0</td><td>WUCR</td><td>RIE</td><td>TIE</td><td>-</td><td>-</td></tr>
<tr><td rowspan="3">SSR /ESCR</td><td>0/1</td><td>REC</td><td>-</td><td>PE</td><td>FRE</td><td>ORE</td><td>RDRF</td><td>TDRE</td><td>TBI</td><td>FLWEN</td><td>ESBL</td><td>INV</td><td>PEN</td><td>P</td><td>L2</td><td>L1</td><td>L0</td></tr>
<tr><td>2</td><td>REC</td><td>-</td><td>-</td><td>AWC</td><td>ORE</td><td>RDRF</td><td>TDRE</td><td>TBI</td><td>SOP</td><td>L3</td><td>CSFE</td><td>WT1</td><td>WT0</td><td>L2</td><td>L1</td><td>L0</td></tr>
<tr><td>3</td><td>REC</td><td>-</td><td>LBD</td><td>FRE</td><td>ORE</td><td>RDRF</td><td>TDRE</td><td>TBI</td><td>-</td><td>ESBL</td><td>LBL2</td><td>LBIE</td><td>LBL1</td><td>LBL0</td><td>DEL1</td><td>DEL0</td></tr>
<tr><td>SSR/IBSR</td><td>4</td><td>REC</td><td>TEST</td><td>DMA</td><td>TBIE</td><td>ORE</td><td>RDRF</td><td>TDRE</td><td>TBI</td><td>FBT</td><td>RACK</td><td>RSA</td><td>TRX</td><td>AL</td><td>RSC</td><td>SPC</td><td>BB</td></tr>
<tr><td rowspan="4">TDR1/0 (RDR1/0)</td><td>0/1</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>D8</td><td>D7</td><td>D6</td><td>D5</td><td>D4</td><td>D3</td><td>D2</td><td>D1</td><td>D0</td></tr>
<tr><td>2</td><td>D15</td><td>D14</td><td>D13</td><td>D12</td><td>D11</td><td>D10</td><td>D9</td><td>D8</td><td>D7</td><td>D6</td><td>D5</td><td>D4</td><td>D3</td><td>D2</td><td>D1</td><td>D0</td></tr>
<tr><td>3</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>D7</td><td>D6</td><td>D5</td><td>D4</td><td>D3</td><td>D2</td><td>D1</td><td>D0</td></tr>
<tr><td>4</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>D7</td><td>D6</td><td>D5</td><td>D4</td><td>D3</td><td>D2</td><td>D1</td><td>D0</td></tr>
<tr><td>-</td><td>0/1</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>TDR3/2 (RDR3/2)</td><td>2</td><td>D31</td><td>D30</td><td>D29</td><td>D28</td><td>D27</td><td>D26</td><td>D25</td><td>D24</td><td>D23</td><td>D22</td><td>D21</td><td>D20</td><td>D19</td><td>D18</td><td>D17</td><td>D16</td></tr>
<tr><td rowspan="2">-</td><td>3</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>4</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td rowspan="4">SACSR 1/0</td><td>0/1</td><td>STST</td><td>-</td><td>-</td><td>-</td><td>-</td><td>TRG1</td><td>TRG0</td><td>TINT</td><td>TINTE</td><td>TSYNE</td><td>TRGE</td><td>TDIV3</td><td>TDIV2</td><td>TDIV1</td><td>TDIV0</td><td>TMRE</td></tr>
<tr><td>2</td><td>STST</td><td>-</td><td>TBEEN</td><td>CSEIE</td><td>CSE</td><td>TRG1</td><td>TRG0</td><td>TINT</td><td>TINTE</td><td>TSYNE</td><td>TRGE</td><td>TDIV3</td><td>TDIV2</td><td>TDIV1</td><td>TDIV0</td><td>TMRE</td></tr>
<tr><td>3</td><td>STST</td><td>BST</td><td>SFD</td><td>SFDE</td><td>AUTE</td><td>TRG1</td><td>TRG0</td><td>TINT</td><td>TINTE</td><td>-</td><td>TRGE</td><td>TDIV3</td><td>TDIV2</td><td>TDIV1</td><td>TDIV0</td><td>TMRE</td></tr>
<tr><td>4</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>TRG1</td><td>TRG0</td><td>TINT</td><td>TINTE</td><td>-</td><td>TRGE</td><td>TDIV3</td><td>TDIV2</td><td>TDIV1</td><td>TDIV0</td><td>TMRE</td></tr>
<tr><td rowspan="4">STMR 1/0</td><td>0/1</td><td>TM15</td><td>TM14</td><td>TM13</td><td>TM12</td><td>TM11</td><td>TM10</td><td>TM9</td><td>TM8</td><td>TM7</td><td>TM6</td><td>TM5</td><td>TM4</td><td>TM3</td><td>TM2</td><td>TM1</td><td>TM0</td></tr>
<tr><td>2</td><td>TM15</td><td>TM14</td><td>TM13</td><td>TM12</td><td>TM11</td><td>TM10</td><td>TM9</td><td>TM8</td><td>TM7</td><td>TM6</td><td>TM5</td><td>TM4</td><td>TM3</td><td>TM2</td><td>TM1</td><td>TM0</td></tr>
<tr><td>3</td><td>TM15</td><td>TM14</td><td>TM13</td><td>TM12</td><td>TM11</td><td>TM10</td><td>TM9</td><td>TM8</td><td>TM7</td><td>TM6</td><td>TM5</td><td>TM4</td><td>TM3</td><td>TM2</td><td>TM1</td><td>TM0</td></tr>
<tr><td>4</td><td>TM15</td><td>TM14</td><td>TM13</td><td>TM12</td><td>TM11</td><td>TM10</td><td>TM9</td><td>TM8</td><td>TM7</td><td>TM6</td><td>TM5</td><td>TM4</td><td>TM3</td><td>TM2</td><td>TM1</td><td>TM0</td></tr>
<tr><td rowspan="4">STMCR 1/0</td><td>0/1</td><td>TC15</td><td>TC14</td><td>TC13</td><td>TC12</td><td>TC11</td><td>TC10</td><td>TC9</td><td>TC8</td><td>TC7</td><td>TC6</td><td>TC5</td><td>TC4</td><td>TC3</td><td>TC2</td><td>TC1</td><td>TC0</td></tr>
<tr><td>2</td><td>TC15</td><td>TC14</td><td>TC13</td><td>TC12</td><td>TC11</td><td>TC10</td><td>TC9</td><td>TC8</td><td>TC7</td><td>TC6</td><td>TC5</td><td>TC4</td><td>TC3</td><td>TC2</td><td>TC1</td><td>TC0</td></tr>
<tr><td>3</td><td>TC15</td><td>TC14</td><td>TC13</td><td>TC12</td><td>TC11</td><td>TC10</td><td>TC9</td><td>TC8</td><td>TC7</td><td>TC6</td><td>TC5</td><td>TC4</td><td>TC3</td><td>TC2</td><td>TC1</td><td>TC0</td></tr>
<tr><td>4</td><td>TC15</td><td>TC14</td><td>TC13</td><td>TC12</td><td>TC11</td><td>TC10</td><td>TC9</td><td>TC8</td><td>TC7</td><td>TC6</td><td>TC5</td><td>TC4</td><td>TC3</td><td>TC2</td><td>TC1</td><td>TC0</td></tr>
<tr><td>-</td><td>0/1</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>SCSCR 1/0</td><td>2</td><td>SST1</td><td>SST0</td><td>SED1</td><td>SED0</td><td>SCD1</td><td>SCD0</td><td>SCAM</td><td>CDIV2</td><td>CDIV1</td><td>CDIV0</td><td>CSLVL</td><td>CSEN3</td><td>CSEN2</td><td>CSEN1</td><td>CSEN0</td><td>CSOE</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>2070</td><td>CHAPTER 35:Overview of Multi-function Serial Interface
6. Registers of the Multi-Function Serial Interface</td><td>
Error)
<table>
<tr><td>SCSTR2</td><td>2</td><td>CSDS15</td><td>CSDS14</td><td>CSDS13</td><td>CSDS12</td><td>CSDS11</td><td>CSDS10</td><td>CSDS9</td><td>CSDS8</td><td>CSDS7</td><td>CSDS6</td><td>CSDS5</td><td>CSDS4</td><td>CSDS3</td><td>CSDS2</td><td>CSDS1</td><td>CSDS0</td></tr>
<tr><td>LAMSR/LAMCR</td><td>3</td><td>LER</td><td>SER</td><td>RDRF</td><td>TDRE</td><td>TBI</td><td>LCSC</td><td>-</td><td>LAHC</td><td>LDL3</td><td>LDL2</td><td>LDL1</td><td>LDL0</td><td>LTDRCL</td><td>LCSTYP</td><td>LIDEN</td><td>LAMEN</td></tr>
<tr><td>EIBCR/NFCR</td><td>4</td><td>-</td><td>-</td><td>SDLS</td><td>SCLS</td><td>SDLC</td><td>SCLC</td><td>SOCE</td><td>BEC</td><td>-</td><td>-</td><td>-</td><td>NFT4</td><td>NFT3</td><td>NFT2</td><td>NFT1</td><td>NFT0</td></tr>
</table>
Correct)
<table>
<tr><td>SCSTR2</td><td>2</td><td>CSDS15</td><td>CSDS14</td><td>CSDS13</td><td>CSDS12</td><td>CSDS11</td><td>CSDS10</td><td>CSDS9</td><td>CSDS8</td><td>CSDS7</td><td>CSDS6</td><td>CSDS5</td><td>CSDS4</td><td>CSDS3</td><td>CSDS2</td><td>CSDS1</td><td>CSDS0</td></tr>
<tr><td>LAMSR/LAMCR</td><td>3</td><td>LER</td><td>SER</td><td>RDRF</td><td>TDRE</td><td>TBI</td><td>LCSC</td><td>-</td><td>LAHC</td><td>LDL3</td><td>LDL2</td><td>LDL1</td><td>LDL0</td><td>LTDRCL</td><td>LCSTYP</td><td>LIDEN</td><td>LAMEN</td></tr>
<tr><td>EIBCR/NFCR</td><td>4</td><td>-</td><td>-</td><td>SDAS</td><td>SCLS</td><td>SDAC</td><td>SCLC</td><td>SOCE</td><td>BEC</td><td>-</td><td>-</td><td>-</td><td>NFT4</td><td>NFT3</td><td>NFT2</td><td>NFT1</td><td>NFT0</td></tr>
</table>
</td></tr>
<tr><td>2070</td><td>CHAPTER 35:Overview of Multi-function Serial Interface
6. Registers of the Multi-Function Serial Interface</td><td>
Error)
<table>
<tr><td>-</td><td>0/1</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>TD7</td><td>TD6</td><td>TD5</td><td>TD4</td><td>TD3</td><td>TD2</td><td>TD1</td><td>TD0</td></tr>
<tr><td>TBYTE1/0</td><td>2</td><td>CS1TD7</td><td>CS1TD6</td><td>CS1TD5</td><td>CS1TD4</td><td>CS1TD3</td><td>CS1TD2</td><td>CS1TD1</td><td>CS1TD0</td><td>CS0TD7</td><td>CS0TD6</td><td>CS0TD5</td><td>CS0TD4</td><td>CS0TD3</td><td>CS0TD2</td><td>CS0TD1</td><td>CS0TD0</td></tr>
<tr><td>LAMIER/LAMTID(LAMRID)</td><td>3</td><td>-</td><td>LCSERIE</td><td>LPTERIE</td><td>LSFERIE</td><td>LBSERIE</td><td>LCSCIE</td><td>-</td><td>LAHCIE</td><td>P1</td><td>P0</td><td>LID5</td><td>LID4</td><td>LID3</td><td>LID2</td><td>LID1</td><td>LID0</td></tr>
<tr><td>-</td><td>4</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
</table>
Correct)
<table>
<tr><td>TBYTE0</td><td>0/1</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>TD7</td><td>TD6</td><td>TD5</td><td>TD4</td><td>TD3</td><td>TD2</td><td>TD1</td><td>TD0</td></tr>
<tr><td>TBYTE1/0</td><td>2</td><td>CS1TD7</td><td>CS1TD6</td><td>CS1TD5</td><td>CS1TD4</td><td>CS1TD3</td><td>CS1TD2</td><td>CS1TD1</td><td>CS1TD0</td><td>CS0TD7</td><td>CS0TD6</td><td>CS0TD5</td><td>CS0TD4</td><td>CS0TD3</td><td>CS0TD2</td><td>CS0TD1</td><td>CS0TD0</td></tr>
<tr><td>LAMIER/LAMTID(LAMRID)</td><td>3</td><td>-</td><td>LCSERIE</td><td>LPTERIE</td><td>LSFERIE</td><td>LBSERIE</td><td>LCSCIE</td><td>-</td><td>LAHCIE</td><td>P1</td><td>P0</td><td>LID5</td><td>LID4</td><td>LID3</td><td>LID2</td><td>LID1</td><td>LID0</td></tr>
<tr><td>-</td><td>4</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>2070</td><td>CHAPTER 35:Overview of Multi-function Serial Interface
6. Registers of the Multi-Function Serial Interface</td><td>Error)
<table><tr><td>FTICR
2/1</td><td>0 to 4</td><td>TXFTL
15</td><td>TXFTL
14</td><td>TXFTL
13</td><td>TXFTL
12</td><td>TXFTL
11</td><td>TXFTL
10</td><td>TXFTL
9</td><td>TXFTL
8</td><td>TXFTL
7</td><td>TXFTL
6</td><td>TXFTL
5</td><td>TXFTL
4</td><td>TXFTL
3</td><td>TXFTL
2</td><td>TXFTL
1</td><td>TXFTL
0</td></tr></table>
Correct)
<table><tr><td>FTICR
2/1</td><td>0 to 4</td><td>TXFTL
15</td><td>TXFTL
14</td><td>TXFTL
13</td><td>TXFTL
12</td><td>TXFTL
11</td><td>TXFTL
10</td><td>TXFTL
9</td><td>TXFTL
8</td><td>TXFTL
7</td><td>TXFTL
6</td><td>TXFTL
5</td><td>TXFTL
4</td><td>TXFTL
3</td><td>TXFTL
2</td><td>TXFTL
1</td><td>TXFTL
0</td></tr><tr><td>ESR/ECR</td><td>0 to 4</td><td>-</td><td>-</td><td>-</td><td>-</td><td>RXUD
R</td><td>TXOVR</td><td>RBER
R</td><td>TBERR</td><td>-</td><td>-</td><td>-</td><td>EISEL</td><td>REIR</td><td>TEIE</td><td>RXBLK
EN</td><td>TXBLK
EN</td></tr><tr><td>TBSIZE</td><td>0 to 4</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>TBSIZ
E7</td><td>TBSIZ
E6</td><td>TBSIZ
E5</td><td>TBSIZ
E4</td><td>TBSIZ
E3</td><td>TBSIZ
E2</td><td>TBSIZ
E1</td><td>TBSIZ
E0</td></tr></table></td></tr>
<tr><td>2071</td><td>CHAPTER 35:Overview of Multi-function Serial Interface
6. Registers of the Multi-Function Serial Interface</td><td>Error)
<table><tr><td>FCR1C/
FCR0C</td><td>0 to 4</td><td>-</td><td>-</td><td>-</td><td>FLSTE
C</td><td>FRIIEC</td><td>FDRQ
C</td><td>FTIEC</td><td>FSELC</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>FE2C</td><td>FE1C</td></tr></table>
Correct)
<table><tr><td>FCR1C/
FCR0C</td><td>0 to 4</td><td>-</td><td>-</td><td>-</td><td>FLSTE
C</td><td>FRIIEC</td><td>FDRQ
C</td><td>FTIEC</td><td>FSELC</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>FE2C</td><td>FE1C</td></tr><tr><td>ESRC/-</td><td>0 to 4</td><td>-</td><td>-</td><td>-</td><td>-</td><td>RXUD
RC</td><td>TXOVR
C</td><td>RBER
RC</td><td>TBERR
C</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr></table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>2076</td><td>CHAPTER 36:UART (Asynchronous Serial Interface)
2. Interrupts from the UART</td><td>Changed
Error)
<table>
<tr><th rowspan="2">Interrupt Types</th><th rowspan="2">Interrupt Request Flag Bit</th><th rowspan="2">Flag Register</th><th colspan="2">Operation Mode</th><th rowspan="2">Interrupt Factor</th><th rowspan="2">Interrupt Factor Enable Bit</th><th rowspan="2">Interrupt Request Flag Clearing</th></tr>
<tr><th>0</th><th>1</th></tr>
<tr><td rowspan="7">Reception</td><td rowspan="3">RDRF</td><td rowspan="3">SSR</td><td rowspan="3">Appl</td><td rowspan="3">Appl</td><td>1-byte reception</td><td rowspan="3">SCR:RIE</td><td>Reading of reception data (RDR)</td></tr>
<tr><td>Reception of as much data as the amount set in FBYTE</td><td rowspan="2">Reading of reception data (RDR) until the reception FIFO is empty</td></tr>
<tr><td>Detection of reception idle for the 8-bit time or longer while the FRIIE bit is "1" and the reception FIFO contains valid data</td></tr>
<tr><td>ORE</td><td>SSR</td><td>Appl</td><td>Appl</td><td>Overrun error</td><td rowspan="2">SCR:RIE
ECR:REIE</td><td rowspan="2">Writing "1" to the reception error flag clear bit (SSR:REC)</td></tr>
<tr><td>FRE</td><td>SSR</td><td>Appl</td><td>Appl</td><td>Framing error</td></tr>
<tr><td>PE</td><td>SSR</td><td>Appl</td><td>NA</td><td>Parity error</td><td></td></tr>
<tr><td>RXUDR</td><td>ESR</td><td>Appl</td><td>Appl</td><td>Reception FIFO under run</td><td>ECR:REIE</td><td>Writing "1" to the reception FIFO under run flag clear bit (ESRC:RXUDRC)</td></tr>
<tr><td>RBERR</td><td>ESR</td><td>Appl</td><td>Appl</td><td>Reception block transfer error</td><td>ECR:REIE</td><td>Writing "1" to the reception block transfer error clear bit (ESRC:RBERRC)</td></tr>
</table>
Correct)
<table>
<tr><th rowspan="2">Interrupt Types</th><th rowspan="2">Interrupt Request Flag Bit</th><th rowspan="2">Flag Register</th><th colspan="2">Operation Mode</th><th rowspan="2">Interrupt Factor</th><th rowspan="2">Interrupt Factor Enable Bit</th><th rowspan="2">Interrupt Request Flag Clearing</th></tr>
<tr><th>0</th><th>1</th></tr>
<tr><td rowspan="7">Reception</td><td rowspan="3">RDRF</td><td rowspan="3">SSR</td><td rowspan="3">Appl</td><td rowspan="3">Appl</td><td>1-byte reception</td><td rowspan="3">SCR:RIE</td><td>- Reading of reception data (RDR)
- Software reset (SCR:UPCL=1)</td></tr>
<tr><td>Reception of as much data as the amount set in FBYTE</td><td rowspan="2">- Reading of reception data (RDR) until the reception FIFO is empty
- Software reset (SCR:UPCL=1)</td></tr>
<tr><td>Detection of reception idle for the 8-bit time or longer while the FRIIE bit is "1" and the reception FIFO contains valid data</td></tr>
<tr><td>ORE</td><td>SSR</td><td>Appl</td><td>Appl</td><td>Overrun error</td><td rowspan="2">SCR:RIE
ECR:REIE</td><td rowspan="2">- Writing "1" to the reception error flag clear bit (SSR:REC)
- Software reset (SCR:UPCL=1)</td></tr>
<tr><td>FRE</td><td>SSR</td><td>Appl</td><td>Appl</td><td>Framing error</td></tr>
<tr><td>PE</td><td>SSR</td><td>Appl</td><td>NA</td><td>Parity error</td><td></td></tr>
<tr><td>RXUDR</td><td>ESR</td><td>Appl</td><td>Appl</td><td>Reception FIFO under run</td><td>ECR:REIE</td><td>- Writing "1" to the reception FIFO under run flag clear bit (ESRC:RXUDRC)
- Software reset (SCR:UPCL=1)
- Disable reception FIFO
(1) When FCR1:FSEL=0, set FCR0.FE2=0
(2) When FCR1:FSEL=1, set FCR0.FE1=0</td></tr>
<tr><td>RBERR</td><td>ESR</td><td>Appl</td><td>Appl</td><td>Reception block transfer error</td><td>ECR:REIE</td><td>- Writing "1" to the reception block transfer error clear bit (ESRC:RBERRC)
- Software reset(SCR:UPCL=1)
- Disable reception block transfer mode (ECR:RXBLKEN=0)</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>2077</td><td>CHAPTER 36:UART (Asynchronous Serial Interface)
2. Interrupts from the UART</td><td>Error)
<table>
<tr><td></td><td>TXOVR</td><td>ESFR</td><td>Appl</td><td>Appl</td><td>Transmission FIFO overrun</td><td>ECR:TEIE</td><td>Writing "1" to the transmission FIFO overrun flag clear bit (ESRC:TXOVRC)</td></tr>
<tr><td></td><td>TBERR</td><td>ESFR</td><td>Appl</td><td>Appl</td><td>Transmission block transfer error</td><td>ECR:TEIE</td><td>Writing "1" to the transmission block transfer error clear bit (ESRC:TBERRC)</td></tr>
<tr><td>Status</td><td>TINT</td><td>SACSR</td><td>Appl</td><td>Appl</td><td>Coincidence of the serial timer register (STMR) and the serial timer comparison register (STMCR) values</td><td>SACSR: TINTE</td><td>Writing "1" to the timer interrupt flag clear bit (SACSRC:TINTC)</td></tr>
</table>

Correct)
<table>
<tr><td></td><td>TXOVR</td><td>ESR</td><td>Appl</td><td>Appl</td><td>Transmission FIFO overrun</td><td>ECR:TEIE</td><td>- Writing "1" to the transmission FIFO overrun flag clear bit (ESRC:TXOVRC)
- Software reset(SCR:UPCL=1)
- Disable transmission FIFO
(1) When FCR1:FSEL=0, set FCR0:FE1=0
(2) When FCR1.FSEL=1, set FCR0:FE2=0</td></tr>
<tr><td></td><td>TBERR</td><td>ESR</td><td>Appl</td><td>Appl</td><td>Transmission block transfer error</td><td>ECR:TEIE</td><td>- Writing "1" to the transmission block transfer error clear bit (ESRC:TBERRC)
- Software reset(SCR:UPCL=1)
- Disable transmission block transfer (ECR.TXBLKEN=0)</td></tr>
<tr><td>Status</td><td>TINT</td><td>SACSR</td><td>Appl</td><td>Appl</td><td>Coincidence of the serial timer register (STMR) and the serial timer comparison register (STMCR) values</td><td>SACSR: TINTE</td><td>- Writing "1" to the timer interrupt flag clear bit (SACSRC:TINTC)
- Software reset(SCR:UPCL=1)</td></tr>
</table>
</td></tr>
<tr><td>2079</td><td>CHAPTER 36:UART (Asynchronous Serial Interface)
2. Interrupts from the UART
2.1. Occurrence of Reception Interrupts and Flag Set Timing</td><td>Error)
At that time, a reception interrupt occurs if the reception interrupt is enabled (SSR:RIE = 1).

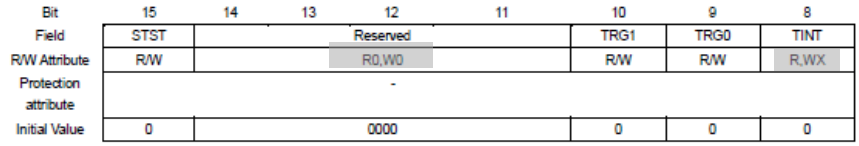
Correct)
At that time, a reception interrupt occurs if the reception interrupt is enabled (SCR:RIE = 1).</td></tr>
</table>

Page	Section	Change Results
2096	CHAPTER 36:UART (Asynchronous Serial Interface) 4. Serial Timer Operation	Error) ***Note:*** − *In the state in which synchronous transmission is disabled (SACSR:TSYNE = 0) and 0x0000 is set in the timer comparison register (STMCR), if the timer is operating and the division value (SACSR:TDIV) of the timer operation clock is set to 0b0000, the timer interrupt flag (SACSR:TINT) is fixed to "1".* Correct) ***Note:*** − *In the state in which synchronous transmission is disabled (SACSR:TSYNE = 0) and 0x0000 is set in the timer comparison register (STMCR), if the timer is operating and the division value (SACSR:TDIV3-0) of the timer operation clock is set to 0b0000, the timer interrupt flag (SACSR:TINT) is fixed to "1".*
2110	CHAPTER 36:UART (Asynchronous Serial Interface) 7. Block Transfer	Error) The block size can be set with the transmission block size register and the reception block size register. Correct) The block size can be set with the transmission block size register and the FBYTE register (reception side).
2113	CHAPTER 36:UART (Asynchronous Serial Interface) 7. Block Transfer 7.1 ACK Mode	Error) **Transmission Block Transfer Error** If, while the Write counter is "0", an acknowledgment is input before the next transmission block transfer request (TIRQ) is asserted, a transmission block transfer error occurs, and the FSR:TBERR bit is asserted. Correct) **Transmission Block Transfer Error** If, while the Write counter is "0", an acknowledgment is input before the next transmission block transfer request (TIRQ) is asserted, a transmission block transfer error occurs, and the ESR:TBERR bit is asserted.

Page	Section	Change Results
2113	CHAPTER 36:UART (Asynchronous Serial Interface) 7. Block Transfer 7.2 RW Access Mode	Error) **Transmission Block Transfer Error** If, while the Write counter is "0", a write access occurs before the next transmission block transfer request (TIRQ) is asserted, a transmission block transfer error occurs, and the FSR:TBERR bit is asserted. Correct) **Transmission Block Transfer Error** If, while the Write counter is "0", a write access occurs before the next transmission block transfer request (TIRQ) is asserted, a transmission block transfer error occurs, and the ESR:TBERR bit is asserted.
2132	CHAPTER 36:UART (Asynchronous Serial Interface) 10. Registers of UART (Asynchronous Serial Interface) 10.2 Serial Mode Register (SMR)	Error) **[bit7:5] MD[2:0]: Operation Mode Setting Bits** These bits set the operation mode of the Asynchronous Serial Interface. 0b000: Set operation mode 0 (asynchronous normal mode). 0b001: Set operation mode 1 (asynchronous multi-processor mode). 0b010: Set operation mode 2 (clock synchronous mode). 0b011: Set operation mode 3 (LIN communication mode). 0b100: Set operation mode 4 (I2C mode). This section describes the registers and their operations in operation mode 0 (asynchronous normal mode) and operation mode 1 (asynchronous multi-processor mode). Correct) **[bit7:5] MD[2:0]: Operation Mode Setting Bits** These bits set the operation mode of the Asynchronous Serial Interface.
2132	CHAPTER 36:UART (Asynchronous Serial Interface) 10. Registers of UART (Asynchronous Serial Interface) 10.2 Serial Mode Register (SMR)	Error) **[bit4] WUCR: WAKE UP Control Bit** Correct) **[bit4] WUCR: WAKE UP Control Bit** ***Note:*** *− The WAKE UP function is not supported.*

Page	Section	Change Results
2134	CHAPTER 36:UART (Asynchronous Serial Interface) 10. Registers of UART (Asynchronous Serial Interface) 10.3 Serial Status Register (SSR)	Error) **[bit13] PE: Parity Error Flag Bit (Effective Only in Operation Mode 0)** Correct) **[bit13] PE: Parity Error Flag Bit (Effective Only in Operation Mode 0)** ***Note:*** − *This bit is reset by the Software reset (SCR:UPCL=1).*
2135	CHAPTER 36:UART (Asynchronous Serial Interface) 10. Registers of UART (Asynchronous Serial Interface) 10.3 Serial Status Register (SSR)	Error) **[bit12] FRE: Framing Error Flag Bit** Correct) **[bit12] FRE: Framing Error Flag Bit** ***Note:*** − *This bit is reset by the Software reset (SCR:UPCL=1).*
2135	CHAPTER 36:UART (Asynchronous Serial Interface) 10. Registers of UART (Asynchronous Serial Interface) 10.3 Serial Status Register (SSR)	Error) **[bit11] ORE: Overrun Error Flag Bit** Correct) **[bit11] ORE: Overrun Error Flag Bit** ***Note:*** − *This bit is reset by the Software reset (SCR:UPCL=1).*

Page	Section	Change Results
2136	CHAPTER 36:UART (Asynchronous Serial Interface) 10. Registers of UART (Asynchronous Serial Interface) 10.3 Serial Status Register (SSR)	Error) **[bit10] RDRF: Reception Data Full Flag Bit** Correct) **[bit10] RDRF: Reception Data Full Flag Bit** ***Note:*** − *This bit is reset by the Software reset (SCR:UPCL=1).*
2136	CHAPTER 36:UART (Asynchronous Serial Interface) 10. Registers of UART (Asynchronous Serial Interface) 10.3 Serial Status Register (SSR)	Error) **[bit9] TDRE: Transmission Data Empty Flag Bit** The TDRE bit is set to "1" when the UPCL bit in the serial control register (SCR) is set to "1". Correct) **[bit9] TDRE: Transmission Data Empty Flag Bit** The TDRE bit is set to "1" by the programmable clear (SCR:UPCL=1) is executed.
2136	CHAPTER 36:UART (Asynchronous Serial Interface) 10. Registers of UART (Asynchronous Serial Interface) 10.3 Serial Status Register (SSR)	Error) **bit8] TBI: Transmission Bus Idle Flag Bit** The TBI bit is set to "1" when the UPCL bit in the serial control register (SCR) is set to "1". Correct) **bit8] TBI: Transmission Bus Idle Flag Bit** The TBI bit is set to "1" by the programmable clear (SCR:UPCL=1) is executed.

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>2144</td><td>CHAPTER 36:UART (Asynchronous Serial Interface)
10. Registers of UART (Asynchronous Serial Interface)
10.6 Serial Auxiliary Control Status Register (SACSR)</td><td>Error)
Figure 36.10-7 Bit Configuration of Serial Auxiliary Control Status Register (SACSR)

Correct)

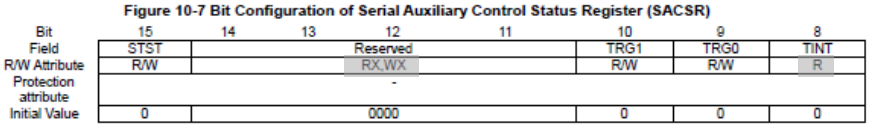
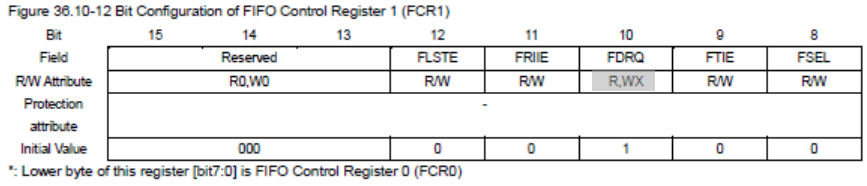



Figure 10-7 Bit Configuration of Serial Auxiliary Control Status Register (SACSR)</td></tr>
<tr><td>2151</td><td>CHAPTER 36:UART (Asynchronous Serial Interface)
10. Registers of UART (Asynchronous Serial Interface)
10.9. Transfer Byte Register (TBYTE0)</td><td>Error)
The transfer byte (TBYTE0) represents the transfer data amount when each serial chip select pin is active.

Correct)
The transfer byte (TBYTE0) represents the transfer data amount when synchronous transmission or external trigger transmission starts.</td></tr>
<tr><td>2154</td><td>CHAPTER 36:UART (Asynchronous Serial Interface)
10. Registers of UART (Asynchronous Serial Interface)
10.11. FIFO Control Register 1 (FCR1)</td><td>Error)
Figure 36.10-12 Bit Configuration of FIFO Control Register 1 (FCR1)

*: Lower byte of this register [bit7:0] is FIFO Control Register 0 (FCR0)
Correct)

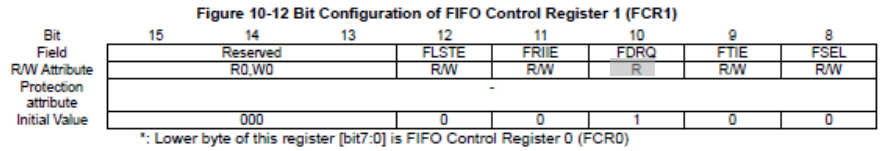



Figure 10-12 Bit Configuration of FIFO Control Register 1 (FCR1)
*: Lower byte of this register [bit7:0] is FIFO Control Register 0 (FCR0)</td></tr>
</table>

Page	Section	Change Results
2155	CHAPTER 36:UART (Asynchronous Serial Interface) 10. Registers of UART (Asynchronous Serial Interface) 10.11. FIFO Control Register 1 (FCR1)	Error) **[bit10] FDRQ: Transmission FIFO Data Request Bit** FDRQ reset conditions ■ "1" is written to FRC1C:FDRQC. Correct) **[bit10] FDRQ: Transmission FIFO Data Request Bit** FDRQ reset conditions ■ "1" is written to FCR1C:FDRQC.
2156	CHAPTER 36:UART (Asynchronous Serial Interface) 10. Registers of UART (Asynchronous Serial Interface) 10.11. FIFO Control Register 1 (FCR1)	Error) **[bit8] FSEL: FIFO Selection Bit** ***Notes:*** − *This bit is not cleared by a FIFO reset (FCR0:FCL2, FCL1 = 1).* − *To change the value of this bit, disable the FIFO operation (FCR0:FE2, FE1 = 0) first.* − *This bit cannot be changed when FDRQ = 0.* Correct) **[bit8] FSEL: FIFO Selection Bit** ***Notes:*** − *This bit is not cleared by a FIFO reset (FCR0:FCL2, FCL1 = 1).* − *To change the value of this bit, disable the FIFO operation (FCR0:FE2, FE1 = 0) first.* − *This bit cannot be changed when FDRQ = 0.* − *Set this bit before setting the FIFO byte register(FBYTE) and the transmission FIFO interrupt control register(FTICR).* − *This bit and the FIFO byte register(FBYTE) cannot be accessed at the same time.*

Page	Section	Change Results
2157	CHAPTER 36:UART (Asynchronous Serial Interface) 10. Registers of UART (Asynchronous Serial Interface) 10.12. FIFO Control Register 0 (FCR0)	Error) FLST reset conditions ■ FIFO reset (writing "1" to FCL) Correct) FLST reset conditions ■ The transmision FIFO reset − set "1" to FCR0:FCL1 when FCR1:FSEL=0 − set "1" to FCR0:FCL2 when FCR1:FSEL=1
2159	CHAPTER 36:UART (Asynchronous Serial Interface) 10. Registers of UART (Asynchronous Serial Interface) 10.12. FIFO Control Register 0 (FCR0)	Error) **[bit3] FCL2: FIFO2 Reset Bit** ***Notes:*** − *Disable transmission/reception first and then execute a FIFO2 reset.* − *Execute the reset after setting the transmission FIFO interrupt enable bit to "0".* − *The valid data count of the FBYTE2 register is set to 0.* Correct) **[bit3] FCL2: FIFO2 Reset Bit** ***Notes:*** − *Disable transmission/reception first and then execute a FIFO2 reset.* − *Execute the reset after setting the transmission FIFO interrupt enable bit to "0".* − *The valid data count of the FBYTE2 register is set to 0.* − *The TDR register and the RDR register are not initialized.*
2159	CHAPTER 36:UART (Asynchronous Serial Interface) 10. Registers of UART (Asynchronous Serial Interface) 10.12. FIFO Control Register 0 (FCR0)	Error) **[bit2] FCL1: FIFO1 Reset Bit** ***Notes:*** − *Disable transmission/reception first and then execute a FIFO1 reset.* − *Execute the reset after setting the transmission FIFO interrupt enable bit to "0".* − *The valid data count of the FBYTE1 register is set to 0.* Correct) **[bit2] FCL1: FIFO1 Reset Bit** ***Notes:*** − *Disable transmission/reception first and then execute a FIFO1 reset.* − *Execute the reset after setting the transmission FIFO interrupt enable bit to "0".* − *The valid data count of the FBYTE1 register is set to 0.* − *The TDR register and the RDR register are not initialized.*

Page	Section	Change Results
2160	CHAPTER 36:UART (Asynchronous Serial Interface) 10. Registers of UART (Asynchronous Serial Interface) 10.12. FIFO Control Register 0 (FCR0)	Error) **[bit1] FE2: FIFO2 Operation Enable Bit** Correct) **[bit1] FE2: FIFO2 Operation Enable Bit** ■ To use FIFO2 as the transmission or reception FIFO, set this bit to "1" or "0" when the transmission or reception buffer is empty ((SSR:TDRE = 1) or (SSR:RDRF = 0)), respectively.
2161	CHAPTER 36:UART (Asynchronous Serial Interface) 10. Registers of UART (Asynchronous Serial Interface) 10.13. FIFO Byte Register (FBYTE)	Error) Correct) ■ Whenever transmission data is written to TDR, the FBYTE of the transmission FIFO is incremented by 1. ■ Whenever reception data is read from RDR, the FBYTE of the reception FIFO is decremented by 1.

Page	Section	Change Results
2162	CHAPTER 36:UART (Asynchronous Serial Interface) 10. Registers of UART (Asynchronous Serial Interface) 10.13. FIFO Byte Register (FBYTE)	Error) **[bit15:8] FBYTE2: FIFO2 Data Count Indication byte** **[bit7:0] FBYTE1: FIFO1 Data Count Indication Byte** ■ Change the value after disabling the reception operation. ■ Any setting that exceeds the FIFO capacity is prohibited. Correct) **[bit15:8] FBYTE2: FIFO2 Data Count Indication byte** **[bit7:0] FBYTE1: FIFO1 Data Count Indication Byte** ***Notes:*** *− Change the value of the FBYTE of the reception FIFO after disabling the reception operation.* *− Any setting that exceeds the FIFO capacity is prohibited.* *− Set the FIFO byte register (FBYTE) after setting the FIFO selection bit (FCR1:FSEL).* *− The FIFO selection bit (FCR1:FSEL) and the FIFO byte register (FBYTE) cannot be set at the same time.* *− As the FIFO data count for transmission, the transmission data count that has been written, minus 1, is displayed as the valid data count. This is because an attempt to write transmission data to the TDR register stores the data in the transmission FIFO if the TDR register contains data that has not been transmitted yet. When the data in the TDR register is transmitted, the data in the transmission FIFO that has not been transmitted is transferred to the TDR register.* *− The FIFO data count for reception represents the data count received by the reception FIFO but which has not yet been read. The data count does not include the data being received by the RDR register.*
2167	CHAPTER 36:UART (Asynchronous Serial Interface) 10. Registers of UART (Asynchronous Serial Interface) 10.16. Extended Status Register (ESR)	Error) **[bit11] RXUDR: Reception FIFO under Run Flag Bit** ***Notes:*** Correct) **[bit11] RXUDR: Reception FIFO under Run Flag Bit** ***Notes:*** *− If Reception FIFO is disabled, this bit is set to "0".*

Page	Section	Change Results
2168	CHAPTER 36:UART (Asynchronous Serial Interface) 10. Registers of UART (Asynchronous Serial Interface) 10.16. Extended Status Register (ESR)	Error) **[bit10] TXOVR: Transmission FIFO Overrun Flag Bit** ***Notes:*** Correct) **[bit10] TXOVR: Transmission FIFO Overrun Flag Bit** ***Notes:*** *− If Transmission FIFO is disabled, this bit is set to "0".*
2168	CHAPTER 36:UART (Asynchronous Serial Interface) 10. Registers of UART (Asynchronous Serial Interface) 10.16. Extended Status Register (ESR)	Error) **[bit9] RBERR: Reception Block Transfer Error Bit** This bit indicates that a block transfer error has occurred during reception. If block transfer is executed with a value greater than the threshold set in the FBYTE register, "1" is set in this bit as a block transfer error. A reception interrupt occurs if reception interrupts are enabled (SSR:RIE = 1). Correct) **[bit9] RBERR: Reception Block Transfer Error Bit** This bit indicates that a block transfer error has occurred during reception. If block transfer is executed with a value greater than the threshold set in the FBYTE register, "1" is set in this bit as a block transfer error. A reception interrupt occurs if reception interrupts are enabled (SCR:RIE = 1).

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>2183</td><td>CHAPTER 37:CSIO (Clock Synchronous Serial Interface)
2. Interrupts of the CSIO (Clock Synchronous Serial Interface)</td><td>
Error)
<table>
<tr><th>Interrupt Type</th><th>Interrupt Request Flag Bit</th><th>Flag Register</th><th>Interrupt Factor</th><th>Interrupt Factor Enable Bit</th><th>Interrupt Request Flag Clearing</th></tr>
<tr><td rowspan="4">Reception</td><td rowspan="3">RDRF</td><td rowspan="3">SSR</td><td>1-byte reception</td><td rowspan="3">SCR:RIE</td><td>Reading of reception data (RDR)</td></tr>
<tr><td>Reception of as much data as the value set in FBYTE</td><td rowspan="2">Reading of reception data until the reception FIFO is empty</td></tr>
<tr><td>Detection of reception idle for the 8-bit time or longer while the FRIIE bit is "1" and the reception FIFO contains valid data</td></tr>
<tr><td>ORE</td><td>SSR</td><td>Overrun error</td><td>SCR:RIE
ECR:REIE</td><td>Writing "1" to the reception error flag clear bit (SSR:REC)</td></tr>
<tr><td>RXUDR</td><td>ESR</td><td>Reception FIFO under run</td><td>ECR:REIE</td><td>Writing "1" to the transmission FIFO under run flag clear bit (ESRC:RXUDRC)</td></tr>
<tr><td>RBERR</td><td>ESR</td><td>Reception block transfer error</td><td>ECR:REIE</td><td>Writing "1" to the reception block transfer error clear bit (ESRC:RBERRC)</td></tr>
</table>
Correct)
<table>
<tr><th>Interrupt Type</th><th>Interrupt Request Flag Bit</th><th>Flag Register</th><th>Interrupt Factor</th><th>Interrupt Factor Enable Bit</th><th>Interrupt Request Flag Clearing</th></tr>
<tr><td rowspan="5">Reception</td><td rowspan="3">RDRF</td><td rowspan="3">SSR</td><td>1-byte reception</td><td rowspan="3">SCR:RIE</td><td>- Reading of reception data (RDR)
- Software reset (SCR:UPCL=1)</td></tr>
<tr><td>Reception of as much data as the value set in FBYTE</td><td rowspan="2">- Reading of reception data until the reception FIFO is empty
- Software reset (SCR:UPCL=1)</td></tr>
<tr><td>Detection of reception idle for the 8-bit time or longer while the FRIIE bit is "1" and the reception FIFO contains valid data</td></tr>
<tr><td>ORE</td><td>SSR</td><td>Overrun error</td><td>SCR:RIE
ECR:REIE</td><td>- Writing "1" to the reception error flag clear bit (SSR:REC)
- Software reset (SCR:UPCL=1)</td></tr>
<tr><td>RXUDR</td><td>ESR</td><td>Reception FIFO under run</td><td>ECR:REIE</td><td>- Writing "1" to the transmission FIFO under run flag clear bit (ESRC:RXUDRC)
- Software reset (SCR:UPCL=1)
-Disable reception FIFO
(1) When FCR1:FSEL=0, set FCR0:FE2=0
(2) When FCR1:FSEL=1, set FCR0:FE1=0</td></tr>
<tr><td>RBERR</td><td>ESR</td><td>Reception block transfer error</td><td>ECR:REIE</td><td>- Writing "1" to the reception block transfer error clear bit (ESRC:RBERRC)
- Software reset(SCR:UPCL=1)
- Disable reception block transfer mode (ECR:RXBLKEN=0)</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>2184</td><td>CHAPTER 37:CSIO (Clock Synchronous Serial Interface)
2. Interrupts of the CSIO (Clock Synchronous Serial Interface)</td><td>Error)
<table>
<tr><td rowspan="3">Transmission</td><td>CSE</td><td>SACSR</td><td>In slave mode (SCR:MS = 1), the serial chip select pin is inactive during transmission.
In master mode (SCR:MS = 0), the transmission count is equal to or less than the TBYTE setting and the next transmission data is not written to TDR (SSR:TDRE = 1).</td><td>SACSR:CSEIE</td><td>Writing "1" to the serial chip select flag clear bit (SACSRC:CSEC)</td></tr>
<tr><td>TXOVR</td><td>ESR</td><td>Transmission FIFO overrun</td><td>ECR:TEIE</td><td>Writing "1" to the transmission FIFO overrun flag clear bit (ESRC:TXOVRC)</td></tr>
<tr><td>TBERR</td><td>ESR</td><td>Transmission block transfer error</td><td>ECR:TEIE</td><td>Writing "1" to the transmission block transfer error clear bit (ESRC:TBERRC)</td></tr>
<tr><td>Status</td><td>TINT</td><td>SACSR</td><td>Coincidence of the serial timer register (STMR) and the serial timer comparison register (STMCR) values</td><td>SACSR:TINTE</td><td>Writing "1" to the timer interrupt flag clear bit (SACSRC:TINTC)</td></tr>
</table>
Correct)
<table>
<tr><td rowspan="3">Transmission</td><td>CSE</td><td>SACSR</td><td>In slave mode (SCR:MS = 1), the serial chip select pin is inactive during transmission.
In master mode (SCR:MS = 0), the transmission count is equal to or less than the TBYTE setting and the next transmission data is not written to TDR (SSR:TDRE = 1).</td><td>SACSR:CSEIE</td><td>- Writing "1" to the serial chip select flag clear bit (SACSRC:CSEC)
- Software reset (SCR:UPCL=1)</td></tr>
<tr><td>TXOVR</td><td>ESR</td><td>Transmission FIFO overrun</td><td>ECR:TEIE</td><td>- Writing "1" to the transmission FIFO overrun flag clear bit (ESRC:TXOVRC)
- Software reset(SCR:UPCL=1)
- Disable transmission FIFO
(1) When FCR1:FSEL=0, set FCR0:FE1=0
(2) When FCR1:FSEL=1, set FCR0:FE2=0</td></tr>
<tr><td>TBERR</td><td>ESR</td><td>Transmission block transfer error</td><td>ECR:TEIE</td><td>- Writing "1" to the transmission block transfer error clear bit (ESRC:TBERRC)
- Software reset(SCR:UPCL=1)
- Disable transmission block transfer(ECR:TXBLKEN=0)</td></tr>
<tr><td>Status</td><td>TINT</td><td>SACSR</td><td>Coincidence of the serial timer register (STMR) and the serial timer comparison register (STMCR) values</td><td>SACSR:TINTE</td><td>- Writing "1" to the timer interrupt flag clear bit (SACSRC:TINTC)
- Software reset(SCR:UPCL=1)</td></tr>
</table></td></tr>
</table>

Page	Section	Change Results
2187	CHAPTER 37:CSIO (Clock Synchronous Serial Interface) 2. Interrupts of the CSIO (Clock Synchronous Serial Interface)	2.1. Reception Interrupt Generation and Flag Set Timing Error) At this time, if reception interrupts are enabled (SSR:RIE = 1), a reception interrupt is generated. Correct) At this time, if reception interrupts are enabled (SCR:RIE = 1), a reception interrupt is generated.
2243	CHAPTER 37:CSIO (Clock Synchronous Serial Interface) 5. Operation of Serial Chip Select	Error) This can be adjusted with the chip select setup delay bits (SCSTR1:CSSU7 to CSSU70). Correct) This can be adjusted with the chip select setup delay bits (SCSTR1:CSSU7 to CSSU0).

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>2251</td><td>CHAPTER 37:CSIO (Clock Synchronous Serial Interface)
5. Operation of Serial Chip Select</td><td>
Error)
Table 5-2 Serial Chip Select Pin Format Setting
<table>
<tr><th colspan="2">Conditions</th><th>Chip Select Active Level</th><th>Serial Clock Inversion</th><th>SPI Setting</th><th>Data Direction</th><th>Data Length</th></tr>
<tr><td rowspan="4">Chip select format is enabled (SCR:CSFE = 1) and master mode (SCR:MS = 0)</td><td>Serial chip select pin 0 output</td><td>SCSCR0:SCLVL</td><td>SMR:SCINV</td><td>SCR:SPI</td><td>SMR:BDS</td><td>ESCR:L3 to L0</td></tr>
<tr><td>Serial chip select pin 1 output</td><td>SCSFR0: CS1SCLVL</td><td>SCSFR0: CS1SCINV</td><td>SCSFR0: CS1SPI</td><td>SCSFR0: CS1BDS</td><td>SCSFR0: CS1L3 to CS1L0</td></tr>
<tr><td>Serial chip select pin 2 output</td><td>SCSFR1: CS2SCLVL</td><td>SCSFR1: CS2SCINV</td><td>SCSFR1: CS2SPI</td><td>SCSFR1: CS2BDS</td><td>SCSFR1: CS2L3 to CS2L0</td></tr>
<tr><td>Serial chip select pin 3 output</td><td>SCSFR2: CS3SCLVL</td><td>SCSFR2: CS3SCINV</td><td>SCSFR2: CS3SPI</td><td>SCSFR2: CS3BDS</td><td>SCSFR2: CS3L3 to CS3L0</td></tr>
<tr><td colspan="2">Chip select format disable (SCR:CSFE = 0)</td><td rowspan="3">SCSCR0:SCLVL</td><td rowspan="3">SMR:SCINV</td><td rowspan="3">SCR:SPI</td><td rowspan="3">SMR:BDS</td><td rowspan="3">ESCR:L3 to L0</td></tr>
<tr><td colspan="2">Slave mode (MS = 1)</td></tr>
<tr><td colspan="2">When chip select is not used (CSEN3 to CSEN0 = 0b0000)</td></tr>
</table>
Correct)
Table 5-2 Serial Chip Select Pin Format Setting
<table>
<tr><th colspan="2">Conditions</th><th>Chip Select Active Level</th><th>Serial Clock Inversion</th><th>SPI Setting</th><th>Data Direction</th><th>Data Length</th></tr>
<tr><td rowspan="4">Chip select format is enabled (SCR:CSFE = 1) and master mode (SCR:MS = 0)</td><td>Serial chip select pin 0 output</td><td>SCSCR0:CSLVL</td><td>SMR:SCINV</td><td>SCR:SPI</td><td>SMR:BDS</td><td>ESCR:L3 to L0</td></tr>
<tr><td>Serial chip select pin 1 output</td><td>SCSFR0: CS1CSLVL</td><td>SCSFR0: CS1SCINV</td><td>SCSFR0: CS1SPI</td><td>SCSFR0: CS1BDS</td><td>SCSFR0: CS1L3 to CS1L0</td></tr>
<tr><td>Serial chip select pin 2 output</td><td>SCSFR1: CS2CSLVL</td><td>SCSFR1: CS2SCINV</td><td>SCSFR1: CS2SPI</td><td>SCSFR1: CS2BDS</td><td>SCSFR1: CS2L3 to CS2L0</td></tr>
<tr><td>Serial chip select pin 3 output</td><td>SCSFR2: CS3CSLVL</td><td>SCSFR2: CS3SCINV</td><td>SCSFR2: CS3SPI</td><td>SCSFR2: CS3BDS</td><td>SCSFR2: CS3L3 to CS3L0</td></tr>
<tr><td colspan="2">Chip select format disable (SCR:CSFE = 0)</td><td rowspan="3">SCSCR0:CSLVL</td><td rowspan="3">SMR:SCINV</td><td rowspan="3">SCR:SPI</td><td rowspan="3">SMR:BDS</td><td rowspan="3">ESCR:L3 to L0</td></tr>
<tr><td colspan="2">Slave mode (MS = 1)</td></tr>
<tr><td colspan="2">When chip select is not used (CSEN3 to CSEN0 = 0b0000)</td></tr>
</table>
</td></tr>
</table>

Page	Section	Change Results
2255	CHAPTER 37:CSIO (Clock Synchronous Serial Interface) 7. Dedicated Baud Rate Generator 7.1. Setting the Baud Rate	Error) (2) Calculation example When bus clock ls 16MHz, using internal clock, baud rate is 19200 bps, the reload value can be calculated as follows. reload value : V = (16 * 1000000)/19200 - 1 = 832 and then baud rate will be b = (16 * 1000000)/(832 + 1) = 19208 bps Correct) (2) Calculation example When bus clock is 16MHz, using internal clock, baud rate is 19200 bps, the reload value can be calculated as follows. reload value : V = (16 * 1000000)/19200 - 1 = 832 and then baud rate will be b = (16 * 1000000)/(832 + 1) = 19208 bps 6e9f8f_ef8b95fa570a507b8b91fc169c501b19_7320_rf_96ff61_u000_c242742_a
2261	CHAPTER 37:CSIO (Clock Synchronous Serial Interface) 8. Block Transfer	Error) The block size can be set with the transmission block size register and the reception block size register. Correct) The block size can be set with the transmission block size register and the FBYTE register (for reception).
2264	CHAPTER 37:CSIO (Clock Synchronous Serial Interface) 8. Block Transfer 8.1 ACK MODE	Error) **Transmission Block Transfer Error** If, while the Write counter is "0", an acknowledgment is input before the next transmission block transfer request (TIRQ) is asserted, a transmission block transfer error occurs, and the FSR:TBERR bit is asserted. Correct) **Transmission Block Transfer Error** If, while the Write counter is "0", an acknowledgment is input before the next transmission block transfer request (TIRQ) is asserted, a transmission block transfer error occurs, and the ESR:TBERR bit is asserted.

Page	Section	Change Results
2274	CHAPTER 37:CSIO (Clock Synchronous Serial Interface) 9. Registers of the CSIO (Clock Synchronous Serial Interface) 9.1. Serial Control Register (SCR)	Error) **[bit11] TIE: Transmission Interrupt Enable Bit** ■ For ECR:EISEL = 0, a transmission interrupt request is issued when the TIE bit and the SSR:TDRE bit are both "1". are set to "1" or when one of the error flag bits (ESR:TXOVR, TBERR) is set to "1". ■ For ECR:EISEL = 1, a reception interrupt request is issued when the TIE bit and the SSR:TDRE bit are both "1" or when one of the error flag bits (ESR:TXOVR, TBERR) is "1". Correct) **[bit11] TIE: Transmission Interrupt Enable Bit** ■ For ECR:EISEL = 0, a transmission interrupt request is issued when the TIE bit and the SSR:TDRE bit are set to "1" or when one of the error flag bits (ESR:TXOVR, TBERR) is set to "1". ■ For ECR:EISEL = 1, a transmission interrupt request is issued when the TIE bit and the SSR:TDRE bit are both "1" or when one of the error flag bits (ESR:TXOVR, TBERR) is "1".
2277	CHAPTER 37:CSIO (Clock Synchronous Serial Interface) 9. Registers of the CSIO (Clock Synchronous Serial Interface) 9.2. Serial Mode Register (SMR)	Error) **[bit1] SCKE: Serial Clock Input/Output Enable Bit** *Notes:* *− Set this bit to "0" in slave mode (SCR:MS = 1) and set this bit to "1" in master mode (SCR_MS = 0).* Correct) **[bit1] SCKE: Serial Clock Input/Output Enable Bit** *Notes:* *− Set this bit to "0" in slave mode (SCR:MS = 1) and set this bit to "1" in master mode (SCR:MS = 0).*

Page	Section	Change Results
2279	CHAPTER 37:CSIO (Clock Synchronous Serial Interface) 9. Registers of the CSIO (Clock Synchronous Serial Interface) 9.3. Serial Status Register (SSR)	Error) **[bit11] ORE: Overrun Error Flag Bit** Correct) **[bit11] ORE: Overrun Error Flag Bit** *Note:* − *This bit is cleared by the Software reset (SCR:UPCL=1).*
2279	CHAPTER 37:CSIO (Clock Synchronous Serial Interface) 9. Registers of the CSIO (Clock Synchronous Serial Interface) 9.3. Serial Status Register (SSR)	Error) **[bit10] RDRF: Reception Data Full Flag Bit** Correct) **[bit10] RDRF: Reception Data Full Flag Bit** *Note:* − *This bit is cleared by the Software reset (SCR:UPCL=1).*

Page	Section	Change Results
2280	CHAPTER 37:CSIO (Clock Synchronous Serial Interface) 9. Registers of the CSIO (Clock Synchronous Serial Interface) 9.3. Serial Status Register (SSR)	Error) **[bit9] TDRE: Transmission Data Empty Flag Bit** ■ If the UPCL bit of the serial control register (SCR) is set to "1", the TDRE bit is set to "1". Correct) **[bit9] TDRE: Transmission Data Empty Flag Bit** ■ The TDRE bit is set to "1" when programmable reset is executed (SCR:UPCL=1).
2280	CHAPTER 37:CSIO (Clock Synchronous Serial Interface) 9. Registers of the CSIO (Clock Synchronous Serial Interface) 9.3. Serial Status Register (SSR)	Error) **[bit8] TBI: Transmission Bus Idle Flag Bit** ■ If the UPCL bit of the serial control register (SCR) is set to "1", the TDRE bit is set to "1". Correct) **[bit8] TBI: Transmission Bus Idle Flag Bit** ■ The TBI bit is set to "1" when programmable reset is executed (SCR:UPCL=1).

Page	Section	Change Results
2281	CHAPTER 37:CSIO (Clock Synchronous Serial Interface) 9. Registers of the CSIO (Clock Synchronous Serial Interface) 9.4. Extended Communication Control Register (ESCR)	Error) **[bit7] SOP: Serial Output Pin Set Bit** **Note:** Correct) **[bit7] SOP: Serial Output Pin Set Bit** **Note:** *– To set Serial output (SOT) pin to "High", follow one of the below mentioned methods.* *1. Execute programmable clear. (Set SCR.UPCL="1")* *- After disabling interrupt, execute programmable clear. (Set SCR.UPCL="1")* *- When FIFOs are used, execute programmable clear after disabling the FIFOs (FCR0:FE2, FE1 = 0).* *2. Disable transmission. (Set SCR.TXE="0")* *- When serial chip select is used (SCSCR:CSEN = 1) in master mode (SCR:MS = 0), perform a programmable reset (SCR:UPCL = 1) after disabling transmission.* *3. Use Bit Band Unit to set ESCR.SOP bit.*
2284	CHAPTER 37:CSIO (Clock Synchronous Serial Interface) 9. Registers of the CSIO (Clock Synchronous Serial Interface) 9.5. Reception Data Register/Transmission Data Register (RDR/TDR)	Error) The reception data and transmission data registers are placed at the same address. If read, the register functions as a reception data register, while if written to, it functions as a transmission data register. Correct) The reception data register and transmission data register are placed at the same address. If read, the register functions as a reception data register, while if written to, it functions as a transmission data register. When FIFO operation is enabled, the RDR/TDR address is the FIFO read/write address.

Page	Section	Change Results
2297	CHAPTER 37:CSIO (Clock Synchronous Serial Interface) 9. Registers of the CSIO (Clock Synchronous Serial Interface) 9.9 Serial Chip Select Control Status Register (SCSCR)	Error) **[bit15:14] SST[1:0]: Serial chip select start bits** **Notes:** ■ Only those serial chip select pins for which serial chip select is enabled (CSEN = 1) become active. ■ When using serial chip select in master mode (SCR:MS = 0), enable serial chip select (CSEN = 1) for the serial chip select pin that is set with these bits. Correct) **[bit15:14] SST[1:0]: Serial chip select start bits** **Notes:** − *Only those serial chip select pins for which serial chip select is enabled (SCSCR:CSEN3, CSEN2, CSEN1, CSEN0=1) become active.* − *When using serial chip select in master mode (SCR:MS = 0), enable serial chip select (SCSCR:CSEN3, CSEN2, CSEN1, CSEN0="1") for the serial chip select pin that is set with these bits.*
2298	CHAPTER 37:CSIO (Clock Synchronous Serial Interface) 9. Registers of the CSIO (Clock Synchronous Serial Interface) 9.9 Serial Chip Select Control Status Register (SCSCR)	Error) **[bit13:12] SED[1:0]: Serial chip select end bits** ■ Only those serial chip select pins for which serial chip select is enabled (CSEN = 1) become active. ■ When serial chip select is used in master mode (SCR:MS = 0), enable serial chip select (CSEN = 1) for the serial chip select pin that is set with these bits. Correct) **[bit13:12] SED[1:0]: Serial chip select end bits** − *Only those serial chip select pins for which serial chip select is enabled (SCSCR:CSEN3, CSEN2, CSEN1, CSEN0="1") become active.* − *When serial chip select is used in master mode (SCR:MS = 0), enable serial chip select (SCSCR:CSEN3, CSEN2, CSEN1, CSEN0="1") for the serial chip select pin that is set with these bits.*

Page	Section	Change Results
2306	CHAPTER 37:CSIO (Clock Synchronous Serial Interface) 9. Registers of the CSIO (Clock Synchronous Serial Interface) 9.11 Serial Chip Select Format Registers (SCSFR2 to SCSFR0)	Error) **[bit12] CS2BDS: Bit for Selecting the Transfer Direction of Chip Select Pin 2** When the chip select data format is enabled (ESCR:CSFE = 1), this bit selects the transfer of transfer serial data when serial chip select pin 2 is active, setting either transfer of the lowest bit first (LSB first, BDS = 0) or transfer of the highest bit first (MSB first, BDS = 1). Correct) **[bit12] CS2BDS: Bit for Selecting the Transfer Direction of Chip Select Pin 2** When the chip select data format is enabled (ESCR:CSFE = 1), this bit selects the transfer of transfer serial data when serial chip select pin 2 is active, setting either transfer of the lowest bit first (LSB first, SCSFR1:CS2BDS = 0) or transfer of the highest bit first (MSB first, SCSFR1:CS2BDS = 1).
2309	CHAPTER 37:CSIO (Clock Synchronous Serial Interface) 9. Registers of the CSIO (Clock Synchronous Serial Interface) 9.11 Serial Chip Select Format Registers (SCSFR2 to SCSFR0)	Error) **[bit4] CS1BDS: Bit for Selecting the Transfer Direction of Chip Select Pin 1** When the chip select data format is enabled (ESCR:CSFE = 1), this bit selects the transfer of transfer serial data when serial chip select pin 1 is active, setting either transfer of the lowest bit first (LSB first, BDS = 0) or transfer of the highest bit first (MSB first, BDS = 1). Correct) **[bit4] CS1BDS: Bit for Selecting the Transfer Direction of Chip Select Pin 1** When the chip select data format is enabled (ESCR:CSFE = 1), this bit selects the transfer of transfer serial data when serial chip select pin 1 is active, setting either transfer of the lowest bit first (LSB first, SCSFR0:CS1BDS = 0) or transfer of the highest bit first (MSB first, SCSFR0:CS1BDS = 1).

Page	Section	Change Results
2311	CHAPTER 37:CSIO (Clock Synchronous Serial Interface) 9. Registers of the CSIO (Clock Synchronous Serial Interface) 9.11 Serial Chip Select Format Registers (SCSFR2 to SCSFR0)	Error) **[bit4] CS3BDS: Bit for Selecting the Transfer Direction of Chip Select Pin 3** When the chip select data format is enabled (ESCR:CSFE = 1), this bit selects the transfer of transfer serial data when serial chip select pin 3 is active, setting either transfer of the lowest bit first (LSB first, BDS = 0) or transfer of the highest bit first (MSB first, BDS = 1). Correct) **[bit4] CS3BDS: Bit for Selecting the Transfer Direction of Chip Select Pin 3** When the chip select data format is enabled (ESCR:CSFE = 1), this bit selects the transfer of transfer serial data when serial chip select pin 3 is active, setting either transfer of the lowest bit first (LSB first, SCSFR2.CS3BDS = 0) or transfer of the highest bit first (MSB first, SCSFR2:CS3BDS = 1).
2314	CHAPTER 37:CSIO (Clock Synchronous Serial Interface) 9. Registers of the CSIO (Clock Synchronous Serial Interface) 9.13 Baud Rate Generator Registers 1,0 (BGR1,BGR0)	Error) ■ Set values in baud rate generator registers 1, 0 (BGR1, BGR0). Correct) ■ These registers are set the value of the division ratio of the serial clock.

Page	Section	Change Results
2315	CHAPTER 37:CSIO (Clock Synchronous Serial Interface) 9. Registers of the CSIO (Clock Synchronous Serial Interface) 9.14 FIFO Control Register 1 (FCR1)	Error) **[bit12] FLSTE: Retransmission Data Lost Detection Enable Bit** This bit enables FLST bit detection. Correct) **[bit12] FLSTE: Retransmission Data Lost Detection Enable Bit** This bit enables FIFO Retransmission Data Lost Flag Bit (FCR0:FLST) detection.
2316	CHAPTER 37:CSIO (Clock Synchronous Serial Interface) 9. Registers of the CSIO (Clock Synchronous Serial Interface) 9.14 FIFO Control Register 1 (FCR1)	Error) **[bit10] FDRQ: Transmission FIFO Data Request Bit** ■ When transmission FIFO interrupt control is used □ FTICR setting value ≥ FTICR read value (The amount of data in the transmission FIFO is at the interrupt trigger level or lower.) Correct) **[bit10] FDRQ: Transmission FIFO Data Request Bit** ■ When transmission FIFO interrupt control is used − For ECR:TXBLKEN=0 − FTICR setting value ≥ FTICR read value (The amount of data in the transmission FIFO is at the interrupt trigger level or lower.) − For ECR:TXBLKEN=1 − FTICR setting value ≦ number of vacant data items in the transmission FIFO

Page	Section	Change Results
2317	CHAPTER 37:CSIO (Clock Synchronous Serial Interface) 9. Registers of the CSIO (Clock Synchronous Serial Interface) 9.14 FIFO Control Register 1 (FCR1)	Error) **[bit8] FSEL: FIFO Selection Bit** **Notes:** Correct) **[bit8] FSEL: FIFO Selection Bit** **Notes:** − *Set this bit before setting the FIFO Byte Register(FBYTE) and the Transmission FIFO Interrupt Control Register(FTICR).* − *This bit can't be accessed with the FIFO Byte Register(FBYTE) at the same time.*
2318	CHAPTER 37:CSIO (Clock Synchronous Serial Interface) 9. Registers of the CSIO (Clock Synchronous Serial Interface) 9.15 FIFO Control Register 0 (FCR0)	Error) **[bit6] FLST: FIFO Retransmission Data Lost Flag Bit** FLST reset conditions ■ FIFO reset (writing "1" to FCL) Correct) **[bit6] FLST: FIFO Retransmission Data Lost Flag Bit** FLST reset conditions ■ The transmission FIFO reset(When FCR1:FSEL=0, set "1" to the FCR0:FCL1. When FCR1:FSEL=1, set "1" to the FCR0:FCL2.)

Page	Section	Change Results
2319	CHAPTER 37:CSIO (Clock Synchronous Serial Interface) 9. Registers of the CSIO (Clock Synchronous Serial Interface) 9.15 FIFO Control Register 0 (FCR0)	Error) **[bit3] FCL2: FIFO2 Reset Bit** **Notes:** Correct) **[bit3] FCL2: FIFO2 Reset Bit** **Notes:** − *Neither TDR register nor the RDR register is initialized.*
2320	CHAPTER 37:CSIO (Clock Synchronous Serial Interface) 9. Registers of the CSIO (Clock Synchronous Serial Interface) 9.15 FIFO Control Register 0 (FCR0)	Error) **[bit2] FCL1: FIFO1 Reset Bit** **Notes:** Correct) **[bit2] FCL1: FIFO1 Reset Bit** **Notes:** − *Neither TDR register nor the RDR register is initialized.*

Page	Section	Change Results
2322	CHAPTER 37:CSIO (Clock Synchronous Serial Interface) 9. Registers of the CSIO (Clock Synchronous Serial Interface) 9.16 FIFO Byte Register (FBYTE)	Error) ■ To receive data with the master operation (master reception), set the SCR:TIE bit and the SCR:TBIE bit to "0", set the number of reception data items in the FBYTE register of the transmission FIFO, and write "0" to the FCR1:FDRQ bit. Correct) ■ To receive data with the master operation (master reception), set the SCR:TIE bit and the SCR:TBIE bit to "0", set the number of reception data items in the FBYTE register of the transmission FIFO, and write "1" to the FCR1C:FDRQC bit.
2323	CHAPTER 37:CSIO (Clock Synchronous Serial Interface) 9. Registers of the CSIO (Clock Synchronous Serial Interface) 9.16 FIFO Byte Register (FBYTE)	Error) **[bit15:8] FBYTE2: FIFO2 Data Count Indication Bit** **[bit7:0] FBYTE1: FIFO1 Data Count Indication Bit** **Notes:** Correct) **[bit15:8] FBYTE2: FIFO2 Data Count Indication Bit** **[bit7:0] FBYTE1: FIFO1 Data Count Indication Bit** **Notes:** − *Set the FIFO byte register (FBYTE) after setting the FIFO selection bit (FCR1:FSEL).* − *The FIFO selection bit (FCR1:FSEL) and the FIFO byte register (FBYTE) cannot be set at the same time.* − *When transmitting, FBYTE shows the value of the number of writing the TDR register times minus 1. This is the reason that the data to transmit written to the TDR register is stored to the FIFO when a previous data to transmit is remained in the TDR register. When the data in the TDR register is transmitted, the data not transmitted in the FIFO is transferred to the TDR register.* − *When receiving, FBYTE shows the value of the number of the received data in the FIFO not read from the RDR register. The data that has not received yet to the RDR register is not included.*

Page	Section	Change Results
2329	CHAPTER 37:CSIO (Clock Synchronous Serial Interface) 9. Registers of the CSIO (Clock Synchronous Serial Interface) 9.19 Extended Status Register (ESR)	Error) **[bit11] RXUDR: Reception FIFO under Run Flag Bit** **Notes:** Correct) **[bit11] RXUDR: Reception FIFO under Run Flag Bit** **Notes:** − *This bit is set to "0" when Reception FIFO is disabled.*
2329	CHAPTER 37:CSIO (Clock Synchronous Serial Interface) 9. Registers of the CSIO (Clock Synchronous Serial Interface) 9.19 Extended Status Register (ESR)	Error) **[bit10] TXOVR: Transmission FIFO Overrun Flag Bit** **Notes:** Correct) **[bit10] TXOVR: Transmission FIFO Overrun Flag Bit** **Notes:** − *This bit is set to "0" when Transmission FIFO is disabled.*

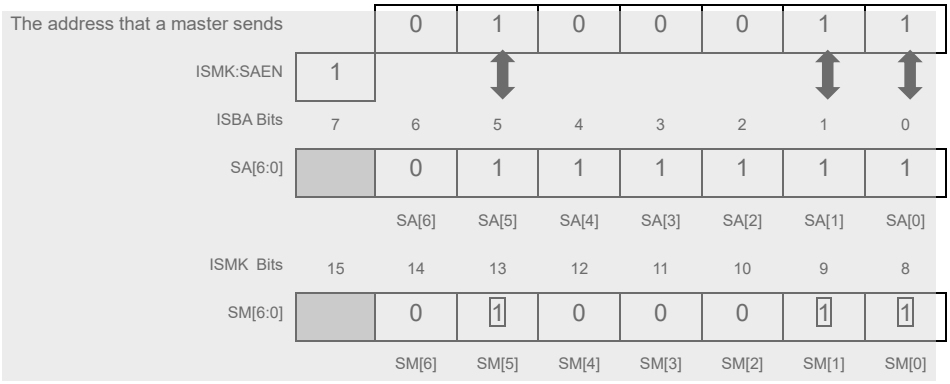
Page	Section	Change Results
2330	CHAPTER 37:CSIO (Clock Synchronous Serial Interface) 9. Registers of the CSIO (Clock Synchronous Serial Interface) 9.19 Extended Status Register (ESR)	Error) **[bit9] RBERR: Reception Block Transfer Error Bit** A reception interrupt occurs if reception interrupts are enabled (SSR:RIE = 1). Notes: ■ If reception block transfer enable (ECR:RXBLKEN) is "0", this bit is set to "0". ■ If reception FIFO enable is "0"(FE2 = 0 for FSEL = 0, FE1 = 0 for FSEL = 1), this bit is set to "0". ■ If the reception FIFO is reset (FCL2 = 1 FSEL = 0, FCL1 = 1 for FSEL = 1), this bit is set to "0". Correct) **[bit9] RBERR: Reception Block Transfer Error Bit** A reception interrupt occurs if reception interrupts are enabled (SCR:RIE = 1). Notes: − *If reception block transfer enable (ECR:RXBLKEN) is "0", or if reception interrupt enable (SCR:RIE) is "0", this bit is set to "0".*
2330	CHAPTER 37:CSIO (Clock Synchronous Serial Interface) 9. Registers of the CSIO (Clock Synchronous Serial Interface) 9.19 Extended Status Register (ESR)	Error) **[bit8] TBERR: Transmission Block Transfer Error Bit** Transmission-side DMA transfer stops. At that time, a transmission interrupt occurs if transmission interrupt is enabled (SSR:TIE = 1). Notes: ■ If transmission FIFO enable is "0" (FE1 = 0 for FSEL = 0,FE2 = 0 for FSEL = 1), this bit is set to "0". ■ If transmission FIFO is reset (FCL1 = 1 for FSEL = 0, FCL2 = 1 for FSEL = 1), this bit is set to "0". Correct) **[bit8] TBERR: Transmission Block Transfer Error Bit** At that time, a transmission interrupt occurs if transmission interrupt is enabled (SCR:TIE = 1). Notes:

Page	Section	Change Results
2336	CHAPTER 37:CSIO (Clock Synchronous Serial Interface) 9. Registers of the CSIO (Clock Synchronous Serial Interface) 9.24 Extended Status Clear Register (ESRC)	Error) **[bit9] TXOVRC: Transmission FIFO overrun flag clear bit** Correct) **[bit10] TXOVRC: Transmission FIFO overrun flag clear bit**

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>2344</td><td>CHAPTER 38: I2C Interface (I2C Communication Control Interface)
1. Overview of the I2C</td><td>Error)
<table><tr><td>7</td><td>Interrupt request</td><td>– Reception interrupt
– Transmission interrupt
– Status interrupt (INT interrupt, stop condition interrupt, repeated start detection interrupt, serial timer interrupt)
– Transmission FIFO interrupt (A transmission FIFO overrun error occurs when the transmission FIFO is not higher than the interrupt trigger level or when the transmission FIFO is empty.)
– Reception FIFO interrupt (reception FIFO under run)
– A DMA transfer support function is provided for both transmission and reception.</td></tr></table>
Correct)
<table><tr><td>7</td><td>Interrupt request</td><td>– Reception interrupt (reception complete, a reserved address reception, overrun error, reception block transfer error)
– Transmission interrupt (transmission data empty, bus idle transmission, transmission block transfer error)
– Status interrupt (INT interrupt, stop condition interrupt, repeated start detection interrupt, serial timer interrupt)
– Transmission FIFO interrupt (A transmission FIFO overrun error occurs when the transmission FIFO is not higher than the interrupt trigger level or when the transmission FIFO is empty.)
– Reception FIFO interrupt (reception FIFO under run)
– A DMA transfer support function is provided for both transmission and reception.</td></tr></table></td></tr>
<tr><td>2345</td><td>CHAPTER 38: I2C Interface (I2C Communication Control Interface)
2. I2C Interface Interrupts</td><td>Error)
<table><tr><td>Status</td><td>TINT</td><td>SACSR</td><td>When serial timer register (STMR) matches serial timer comparison register (STMCR)</td><td>SACSR:TINTE</td><td>Writing "0" to timer interrupt flag bit (SACSR:TINT)
Writing "1" to timer interrupt flag clear bit (SACSRC:TINTC)</td></tr></table>
Correct)
<table><tr><td>Status</td><td>TINT</td><td>SACSR</td><td>When serial timer register (STMR) matches serial timer comparison register (STMCR)</td><td>SACSR:TINTE</td><td>Writing "1" to timer interrupt flag clear bit (SACSRC:TINTC)</td></tr></table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>2346</td><td>CHAPTER 38: I2C Interface (I2C Communication Control Interface)
2. I2C Interface Interrupts</td><td>
Error)
<table>
<tr><td rowspan="2">Recption</td><td>RXUDR</td><td>ESR</td><td>Reception FIFO under run</td><td>ECR:REIE</td><td>- Writing "1" to reception FIFO under run flag clear bit (ESRC:RXUDRC)</td></tr>
<tr><td>RBERR</td><td>ESR</td><td>Reception block transfer error</td><td>ECR:REIE</td><td>- Writing "1" to reception block transfer error clear bit (ESRC:RBERRC)</td></tr>
</table>
Correct)
<table>
<tr><td rowspan="2">Recption</td><td>RXUDR</td><td>ESR</td><td>Reception FIFO under run</td><td>ECR:REIE</td><td>- Writing "1" to reception FIFO under run flag clear bit (ESRC:RXUDRC)
- Disable reception FIFO
1)When FCR1:FSEL=0, FCR0:FE2=0
2)When FCR1:FSEL=1, FCR0:FE1=0</td></tr>
<tr><td>RBERR</td><td>ESR</td><td>Reception block transfer error</td><td>ECR:REIE</td><td>- Writing "1" to reception block transfer error clear bit (ESRC:RBERRC)
- Disable Block transfer mode (ECR:RXBLKEN=0)</td></tr>
</table>
</td></tr>
<tr><td>2346</td><td>CHAPTER 38: I2C Interface (I2C Communication Control Interface)
2. I2C Interface Interrupts</td><td>
Error)
<table>
<tr><td rowspan="2">Transmission</td><td>TXOVR</td><td>ESR</td><td>Transmission FIFO overrun</td><td>ECR:TEIE</td><td>- Writing "1" to transmission FIFO overrun flag clear bit (ESRC:TXOVRC)</td></tr>
<tr><td>TBERR</td><td>ESR</td><td>Transmission block transfer error</td><td>ECR:TEIE</td><td>- Writing "1" to transmission block transfer error clear bit (ESRC:TBERRC)</td></tr>
</table>
Correct)
<table>
<tr><td rowspan="2">Transmission</td><td>TXOVR</td><td>ESR</td><td>Transmission FIFO overrun</td><td>ECR:TEIE</td><td>- Writing "1" to transmission FIFO overrun flag clear bit (ESRC:TXOVRC)
- Disable transmission FIFO
1)When FCR1:FSEL=0, FCR0:FE1=0
2)When FCR1:FSEL=1, FCR0:FE2=0</td></tr>
<tr><td>TBERR</td><td>ESR</td><td>Transmission block transfer error</td><td>ECR:TEIE</td><td>- Writing "1" to transmission block transfer error clear bit (ESRC:TBERRC)
- Disable Block transfer mode (ECR:TXBLKEN=0)</td></tr>
</table>
</td></tr>
</table>

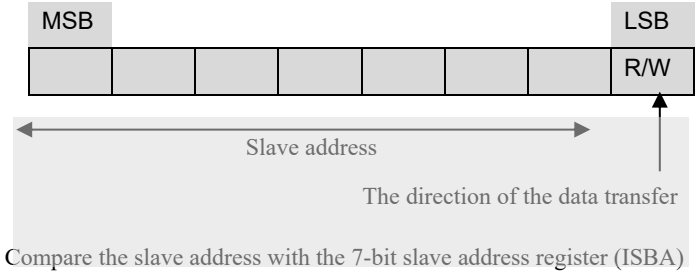
Page	Section	Change Results
2356	CHAPTER 38: I2C Interface (I2C Communication Control Interface) 3 I2C Interface Operation 3.1 Master Mode	Error) **When DMA Mode is Enabled (SSR:DMA = 1)** ■ If the IBSR:RSA bit is "1", the interrupt flag (IBCR:INT) is set to "1" after the reserved address is received (before an acknowledgment), the SCL remains "L", and a wait is performed. After the RDR register is read, the IBCR:ACKE bit and transmission data are set, and "0" is written to the interrupt flag. This sets the interrupt flag to "0", which releases the wait. Correct) **When DMA Mode is Enabled (SSR:DMA = 1)** If the IBSR:RSA bit is "1", the interrupt flag (IBCR:INT) is set to "1" after the reserved address is received (before an acknowledgment), the SCL remains "L", and a wait is performed. After the RDR register is read, the IBCR:ACKE bit and transmission data are set, the writing "1" to the IBCRC:INTC sets the interrupt flag to "0", which releases the wait.
2390	CHAPTER 38: I2C Interface (I2C Communication Control Interface) 3 I2C Interface Operation 3.2. Slave Mode	Error) Operation is in slave mode after the (repeated) start condition is detected and an ACK response is returned when the combination of the ISBA register and the ISMK register matches the received address. **Notes:** ■ When EIBCR:BEC = 0 during address data transfer after the start condition has been detected or during transfer of bit2 to bit9 (acknowledgment bits), the start condition may be detected again. If so, a bus error is detected (IBCR:BER = 1) and reception is stopped, so the next data cannot be received. Correct) Operation is in slave mode after the start condition or the repeated start condition is detected and an ACK response is returned when the combination of the ISBA register and the ISMK register matches the received address. ***Notes:*** *− When EIBCR:BEC=0, if 2nd start condition is detected after 1st start condition is detected (during transfer of bit2 to bit9), a bus error is detected (IBCR:BER=1) and reception is stopped.*

Page	Section	Change Results
2390	CHAPTER 38:I2C Interface (I2C Communication Control Interface) 3 I2C Interface Operation 3.2. Slave Mode	Error) **Slave Address Match Detection** After the (repeated) start condition is detected, the next data is received as a 7-bit address. With regard to the bits that are set to "1" in the ISMK register, the bits in the ISBA register are compared with the bits of the received address. If they match, ACK is output. Correct) **Slave Address Match Detection** The 1st receiving data after the start condition or the repeated start condition is detected contains 7-bit slave address and the bit that shows the direction of the data transmission. The ISMK register is set the value that masks the slave address of the ISBA register. The bit value "0" means "don't care", and "1" means that it needs that the bit value of address corresponds with the data of received slave address. That is, the bit whose the value is set to "0" in the ISMK register is not compared to the bit of the address. When SAEN is set to "1", it enables the slave address detection. The data as the address sent by a master is compared with some slave address bits(SA[6:0]) that mask bits(SM[6:0]) corresponded to is set to "1". If they match, ACK is output. If not, ACK is not output. Example of the address detection For example, a master send the slave address 0x23. **Figure 3-39 The example of the slave address detection**  Just SA5, SA1, SA0 are compared the data as the address sent by the master. Because SM[6], SM[4:2] are set to "0", those bits are "don't care". As a result, the Multi-function serial interface outputs the ACK response.

Page	Section	Change Results
2393	CHAPTER 38:I2C Interface (I2C Communication Control Interface) 3 I2C Interface Operation 3.2. Slave Mode	Error) **When DMA Mode is Disabled (SSR:DMA = 0)** ■ When the reception FIFO is enabled 2. If the interrupt flag (IBCR:INT) was set to "1", read the received data from the RDR register. After reading all the data, write "0" to the interrupt flag to release the wait for the I2C bus. If the stop condition or the repeated start condition was detected, read all the received data from the RDR register, and clear the IBSR:SPC bit or the IBSR:RSC bit to "0". Correct) **When DMA Mode is Disabled (SSR:DMA = 0)** ■ When the reception FIFO is enabled 2. If the interrupt flag (IBCR:INT) was set to "1", read the received data from the RDR register. After reading all the data, write "1" to the interrupt flag clear bit (IBCRC:INTC) to release the wait for the I2C bus. If the stop condition or the repeated start condition was detected, read all the received data from the RDR register, and clear the IBSR:SPC bit or the IBSR:RSC bit to "0".
2400	CHAPTER 38:I2C Interface (I2C Communication Control Interface) 3 I2C Interface Operation 3.3. Bus Error	Error) Check the IBCR:BER bit when the interrupt flag (IBCR:INT) is set to "1" according to the status. If the IBCR:BER bit is "1", take action against the error. The IBCR:BER bit is cleared by the writing of "0" to the IBCR:INT bit. Correct) **When EIBCR:BEC=0** Check the IBCR:BER bit when the interrupt flag (IBCR:INT) is set to "1" according to the status. If the IBCR:BER bit is "1", take action against the error. The IBCR:BER bit is cleared by writing "1" to the interrupt flag clear bit (IBCRC:INTC).
2406	CHAPTER 38:I2C Interface (I2C Communication Control Interface) 6. Block Transfer	Error) The block size is set in the transmission/reception block size register. Correct) The block size is set in the transmission block size register or FBYTE register(reception side).

Page	Section	Change Results
2411	CHAPTER 38:I2C Interface (I2C Communication Control Interface) 6. Block Transfer 6.2. RW Access Mode	Error) After the reception block transfer is requested, the reception block transfer request (RIRQ) is cleared when read access to the reception FIFO occurs. The read access to the reception FIFO decrements the Read counter. Correct) After the reception block transfer is requested, the reception block transfer request (RIRQ) is cleared when a read access to the reception data occurs. The read access to the reception data decrements the Read counter.
2433	CHAPTER 38:I2C Interface (I2C Communication Control Interface) 8. I2C Interface Registers 8.2. Serial Mode Register (SMR)	Error) **[bit7:5] MD[2:0] : Operation Mode Setting Bits** These bits set the operation mode. "0b000": Set operation mode 0 (asynchronous normal mode). "0b001": Set operation mode 1 (asynchronous multi-processor mode). "0b010": Set operation mode 2 (clock synchronous mode). "0b011": Set operation mode 3 (LIN communication mode). "0b100": Set operation mode 4 (I^2C mode). This section explains the register and its operations in operation mode 4 (I^2C mode). Correct) **[bit7:5] MD[2:0] : Operation Mode Setting Bits** These bits set the operation mode.
2435	CHAPTER 38:I2C Interface (I2C Communication Control Interface) 8. I2C Interface Registers 8.3. I2C Bus Status Register (IBSR)	Error) **[bit7] FBT : First Byte Bit** 1. The 2nd byte is transmitted and received. Correct) **[bit7] FBT : First Byte Bit** 1. The 2nd byte starts to be transmitted and received.

Page	Section	Change Results
2442	CHAPTER 38:I2C Interface (I2C Communication Control Interface) 8. I2C Interface Registers 8.4. Serial Status Register (SSR)	Error) **[bit10] RDRF : Reception Data Full Flag Bit** ■ When reading of the reception FIFO empties it during use, this bit is cleared to "0". **Notes:** ■ After RDRF becomes "1" when the reception FIFO is used, resetting the reception FIFO (FCR0:FCL2,FCL1= 1) does not cause RDRF to become "0". Therefore, to set RDRF to "0" after resetting the reception FIFO, perform a dummy read on the reception data register while reception is disabled (SCR:RXE= 0). Correct) **[bit10] RDRF : Reception Data Full Flag Bit** ■ When ECR:RXBLKEN=0 and reading of the reception FIFO empties it during use, this bit is cleared to "0". ■ When ECR:RXBLKEN=1 and using reception FIFO, this bit is clear to "0" when the number of the data in the reception FIFO is less than or equal to the value of the FBYTE register. **Notes:** − *After RDRF becomes "1" when the reception FIFO is used, resetting the reception FIFO (FCR0:FCL2,FCL1= 1) does not cause RDRF to become "0". Therefore, to set RDRF to "0" after* *resetting the reception FIFO, perform a dummy read on the reception data register while I2C mode is disabled (ISMK.I2CEN=0).*
2445	CHAPTER 38:I2C Interface (I2C Communication Control Interface) 8. I2C Interface Registers 8.5. Reception Data Register/Transmission Data Register (RDR/TDR)	Error) Correct) When the FIFO mode is enabled, RDR/TDR register address is the address of reading/writing of the FIFO.

Page	Section	Change Results
2452	CHAPTER 38:I2C Interface (I2C Communication Control Interface) 8. I2C Interface Registers 8.9. Noise Filter Control Register (NFCR)	Error) ***Notes:*** − *Set these bits when the ISMK:EN bit in the ISMK register is "0".* Correct) ***Notes:*** − *Set these bits when the EN bit in the ISMK register is "0".*
2457	CHAPTER 38:I2C Interface (I2C Communication Control Interface) 8. I2C Interface Registers 8.12. 7-Bit Slave Address Register (ISBA)	Error) **[bit6:0] SA[6:0] : 7-Bit Slave Address** Correct) **[bit6:0] SA[6:0] : 7-Bit Slave Address** ■The 1st received data after detecting the (repeated) start condition consists of 7-bit slave address and the bit that specifies the direction of data transfer. It compares the slave address included in the received data to these bits. **Figure 8-14 The 1st byte format after detecting the (repeated) start condition** 

Page	Section	Change Results
2459	CHAPTER 38:I2C Interface (I2C Communication Control Interface) 8. I2C Interface Registers 8.13. Baud Rate Generator Registers 1, 0 (BGR1, BGR0)	Error) BGR1 represents the upper bits, and BGR0 represents the lower bits. The reload value for counting can be written to these registers, and the BGR1/0 setting value can be read. Correct) BGR1 represents the upper bits, and BGR0 represents the lower bits. The reload value for counting can be written to these registers, and the setting value can be read.
2460	CHAPTER 38:I2C Interface (I2C Communication Control Interface) 8. I2C Interface Registers 8.14. FIFO Control Register 1 (FCR1)	Error) **[bit12] FLSTE : Retransmission Data Lost Detection Enable Bit** This bit enables FCR0:FLST bit detection. Correct) **[bit12] FLSTE : Retransmission Data Lost Detection Enable Bit** This bit enables the detection of the FIFO Retransmission Data Lost Flag Bit (FCR0:FLST).
2462	CHAPTER 38:I2C Interface (I2C Communication Control Interface) 8. I2C Interface Registers 8.14. FIFO Control Register 1 (FCR1)	Error) **[bit10] FDRQ : Transmission FIFO Data Request Bit** FDRQ reset conditions ■ "1" is written to the FRC1C:FDRQC bit. ***Notes:*** Correct) **[bit10] FDRQ : Transmission FIFO Data Request Bit** FDRQ reset conditions ■ "1" is written to the FCR1C:FDRQC bit. ***Notes:*** *− Writing "0" to this bit is inhibited when FBYTE(for transmission) = "0".*

Page	Section	Change Results
2462	CHAPTER 38:I2C Interface (I2C Communication Control Interface) 8. I2C Interface Registers 8.14. FIFO Control Register 1 (FCR1)	Error) **[bit8] FSEL : FIFO Selection Bit** ***Notes:*** Correct) **[bit8] FSEL : FIFO Selection Bit** ***Notes:*** − *Set the FIFO Selection Bit (FCR1:FSEL) before setting the FIFO byte register (FBYTE) and the Transmission FIFO Interrupt Control Register (FTICR).* − *It can not access the FIFO byte register (FBYTE) at the same access.*
2463	CHAPTER 38:I2C Interface (I2C Communication Control Interface) 8. I2C Interface Registers 8.15. FIFO Control Register 0 (FCR0)	Error) **[bit6] FLST : FIFO Retransmission Data Lost Flag Bit** FLST reset conditions ■ There is a FIFO reset (writing "1" to FCL). Correct) **[bit6] FLST : FIFO Retransmission Data Lost Flag Bit** FLST reset conditions ■ Transmission FIFO reset. − When FCR1:FSEL="0", set "1" to the FCR0:FCL1. − When FCR1:FSEL="1", set "1" to the FCR0:FCL2.
2465	CHAPTER 38:I2C Interface (I2C Communication Control Interface) 8. I2C Interface Registers 8.15. FIFO Control Register 0 (FCR0)	Error) **[bit3] FCL2 : FIFO2 Reset Bit** ***Notes:*** − *The valid data count of the FBYTE1 register is set to 0.* Correct) **[bit3] FCL2 : FIFO2 Reset Bit** ***Notes:*** − *The valid data count of the FBYTE2 register is set to 0.* − *TDR register and RDR register are not initialized.*

Page	Section	Change Results
2465	CHAPTER 38:I2C Interface (I2C Communication Control Interface) 8. I2C Interface Registers 8.15. FIFO Control Register 0 (FCR0)	Error) **[bit2] FCL1 : FIFO1 Reset Bit** ***Notes:*** Correct) **[bit2] FCL1 : FIFO1 Reset Bit** ***Notes:*** − *TDR register and RDR register are not initialized*
2467	CHAPTER 38:I2C Interface (I2C Communication Control Interface) 8. I2C Interface Registers 8.16. FIFO Byte Register (FBYTE)	Error) Correct) ■ When writing a transmission data to the TDR register one time, the FBYTE of the transmission FIFO is incremented by one ("+1"). ■ When reading a reception data from the RDR register one time, the FBYTE of the reception FIFO is decremeted by one ("-1").
2468	CHAPTER 38:I2C Interface (I2C Communication Control Interface) 8. I2C Interface Registers 8.16. FIFO Byte Register (FBYTE)	Error) Correct) ***Notes:*** − *After setting the FIFO Selection Bit (FCR1:FSEL), set the FIFO byte register (FBYTE).* − *The FIFO Selection Bit and the FIFO byte register can not be set at the same time.* − *The indication of transmission FIFO data number is indicated as the valid value that is the number of writing times of the transmission data decrimented by one. This is the reason that a transmission data written is transferred to transmission FIFO when TDR register has a data not transmitted. When the data in TDR register is transmitted, the data not transmitted in transmission FIFO is transferred to TDR register.* − *The indication of reception FIFO data number is indicated the number of the data received to reception FIFO and not read from reception FIFO. The data receiving in RDR register is not included.*

Page	Section	Change Results
2470	CHAPTER 38:I2C Interface (I2C Communication Control Interface) 8. I2C Interface Registers 8.17. Transmission FIFO Interrupt Control Register (FTICR)	Error) The FTICR register sets the interrupt trigger level by the valid transmission data count (remaining amount) of the transmission FIFO. The following table lists settings with the FCR1:FSEL bit. ■ Set the data count that generates a transmission interrupt in FTICR of the transmission FIFO. The interrupt flag (FDRQ) is set to "1" when the set data count matches or is smaller than the valid data count of the transmission FIFO (FTICR or FBYTE). Correct) The function of this bit is changed by the value of ECR:TXBLKEN. − ECR:TXBLKEN="0" The FTICR register sets the interrupt trigger level by the valid transmission data count (remaining amount) of the transmission FIFO. − ECR:TXBLKEN="1" The FTICR register sets the interrupt trigger level by the number of the empty data in transmission FIFO. ■ Set the data count that generates a transmission interrupt in FTICR of the transmission FIFO. ■ When ECR:TXBLKEN="0", the interrupt flag (FDRQ) is set to "1" when the set data count matches or is smaller than the valid data count of the transmission FIFO (FTICR or FBYTE). ■ When ECR:TXBLKEN="1", the interrupt flag (FDRQ) is set to "1" when the set data count matches or is smaller than the number of the empty data in transmission FIFO. When the number of data set to this register is bigger than the number of the empty data in transmission FIFO, the interrupt flag (FCR1:FDRQ) is set to "0".
2474	CHAPTER 38:I2C Interface (I2C Communication Control Interface) 8. I2C Interface Registers 8.19. Extended Status Register (ESR)	Error) **[bit11] RXUDR : Reception FIFO under Run Flag Bit** ***Notes:*** −*A software reset (SCR:UPCL= 1) resets this bit to "0".* Correct) **[bit11] RXUDR : Reception FIFO under Run Flag Bit** ***Notes:*** − *This bit is "0" when the reception FIFO is disabled.*

Page	Section	Change Results
2475	CHAPTER 38:I2C Interface (I2C Communication Control Interface) 8. I2C Interface Registers 8.19. Extended Status Register (ESR)	Error) **[bit10] TXOVR : Transmission FIFO Overrun Flag Bit** ***Notes:*** −*A software reset (SCR:UPCL= 1) resets this bit to "0".* Correct) **[bit10] TXOVR : Transmission FIFO Overrun Flag Bit** ***Notes:*** − *This bit is "0" when the transmission FIFO is disabled.*
2475	HAPTER 38:I2C Interface (I2C Communication Control Interface) 8. I2C Interface Registers 8.19. Extended Status Register (ESR)	Error) **[bit9] RBERR : Reception Block Transfer Error Bit** If reception interrupts are enabled (SSR:RIE= 1), a reception interrupt is generated. **Notes:** − A software reset (SCR:UPCL= 1) resets this bit to "0". − This bit is "0" when the reception block transfer enable (ECR:RXBLKEN) is "0". Correct) **[bit9] RBERR : Reception Block Transfer Error Bit** If reception interrupts are enabled (SMR:RIE= 1), a reception interrupt is generated. ***Notes:*** − *This bit is "0" when the reception block transfer enable (ECR:RXBLKEN) is "0" or when the reception interrupt enable bit is "0".*

Page	Section	Change Results
2475	HAPTER 38:I2C Interface (I2C Communication Control Interface) 8. I2C Interface Registers 8.19. Extended Status Register (ESR)	Error) **[bit8] TBERR : Transmission Block Transfer Error Bit** If a block transfer is performed using a value larger than the set threshold value in the FTICR register, it is handled as a block transfer error. Then, this bit is set to "1", and transmission DMA transfer is stopped. If transmission interrupts are enabled (SSR:TIE= 1), a transmission interrupt is generated. ***Notes:*** − A software reset (SCR:UPCL= 1) resets this bit to "0". Correct) **[bit8] TBERR : Transmission Block Transfer Error Bit** If a block transfer is performed using a value larger than the set threshold value in the FTICR register, it is handled as a block transfer error. Then, this bit is set to "1". If transmission interrupts are enabled (SMR:TIE= 1), a transmission interrupt is generated. ***Notes:*** − *This bit is set "1" as the block transfer error when the transmission FIFO is OFF and the block* *transfer is executed under either the condition:* − The transmission interrupt is enabled (SMR:TIE="1") and the TDR register has a data (SSR:TDRE="0") − The transmission interrupt is disabled(SMR:TIE="0") and the transmission bus idle interrupt is enabled (SSR:TBIE="1") and in transmitting (SSR:TBI="0") − *When the transmission interrupt is enabled (SMR:TIE="1") and the transmission bus idle interrupt is enabled (SSR:TBIE="1"), the condition of the transmission interrupt enabled (SMR:TIE="1") is prior.*

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>2502</td><td>CHAPTER 39: LIN Interface (V2.1)
2. LIN Interface (v2.1) Interrupts
2.1 Manual Mode</td><td>Error)
Table 39.2-1 Interrupt Control Bits and Interrupt Factors of LIN Interface (v2.1) (Manual Mode)
<table>
<tr><th>Interrupt Type</th><th>Interrupt Request Flag Bit</th><th>Flag Register</th><th>Interrupt Factor</th><th>Interrupt Factor Enable Bit</th><th>Clearing of Interrupt Request Flag</th></tr>
<tr><td rowspan="2">Reception</td><td rowspan="2">RDRF</td><td rowspan="2">SSR</td><td>1-byte reception</td><td rowspan="2">SCR:RIE</td><td>Reading of reception data (RDR)</td></tr>
<tr><td>Reception of as much data as the amount set in FBYTE</td><td>Reading of reception data until the reception FIFO becomes empty</td></tr>
</table>
<table>
<tr><th>Interrupt Type</th><th>Interrupt Request Flag Bit</th><th>Flag Register</th><th>Interrupt Factor</th><th>Interrupt Factor Enable Bit</th><th>Clearing of Interrupt Request Flag</th></tr>
<tr><td rowspan="6">Transmission</td><td></td><td></td><td>Detection of reception idle for 8-bit time or longer while the FRIIE bit is "1" and the reception FIFO contains valid data</td><td></td><td></td></tr>
<tr><td>ORE</td><td>SSR</td><td>Overrun error</td><td rowspan="2">SCR:RIE
ECR:REIE</td><td rowspan="2">Writing "1" to the reception error flag clear bit (SSR:REC)</td></tr>
<tr><td>FRE</td><td>SSR</td><td>Framing error</td></tr>
<tr><td>RXUDR</td><td>ESR</td><td>Reception FIFO underrun</td><td>ECR:REIE</td><td>Writing "1" to the reception FIFO underrun clear flag bit (ESRC:RXUDRC)</td></tr>
<tr><td>RBERR</td><td>ESR</td><td>Reception block transfer error</td><td>ECR:REIE</td><td>Writing "1" to the reception block transfer error clear bit (ESRC:RBERRC)</td></tr>
<tr><td>TDRE</td><td>SSR</td><td>The transmission register is empty.</td><td>SCR:TIE</td><td>Writing to the transmission data (TDR),
or writing "1" to the transmission FIFO operation enable bit (FCR0:FE1 or FCR0:FE2) (retransmission) while the bit value is "0" and the transmission FIFO contains valid data
*1</td></tr>
<tr><td>TBI</td><td>SSR</td><td>No transmission operation</td><td>SCR:TBIE</td><td>Writing to the transmission data (TDR), writing "1" to the LIN Break Field setting bit (LBR),
or writing "1" to the transmission FIFO operation enable bit (FCR0:FE1 or FCR0:FE2)
(retransmission) while the bit value is "0" and the transmission FIFO contains valid data*1</td></tr>
<tr><td>FDRQ</td><td>FCR1</td><td>The amount of data in the transmission FIFO is equal to or smaller than the FTICR setting value or
it is empty.</td><td>FCR1:FTIE</td><td>Writing "1" to the FIFO transmission data request clear bit (FCR1C:FDRQC)
, or transmission FIFO full</td></tr>
<tr><td>TXOVR</td><td>ESR</td><td>Transmission FIFO overrun</td><td>ECR:TEIE</td><td>Writing "1" to the transmission FIFO overrun flag clear bit (ESRC:TXOVRC)</td></tr>
</table>
<table>
<tr><th>Interrupt Type</th><th>Interrupt Request Flag Bit</th><th>Flag Register</th><th>Interrupt Factor</th><th>Interrupt Factor Enable Bit</th><th>Clearing of Interrupt Request Flag</th></tr>
<tr><td></td><td>TBERR</td><td>ESR</td><td>Transmission block transfer error</td><td>ECR:TEIE</td><td>Writing "1" to the transmission block transfer error clear bit (FSRC:TBERRC)</td></tr>
<tr><td rowspan="3">Status (Manual Mode)</td><td>LBD</td><td>SSR</td><td>LIN Break Field detection</td><td>ESCR:LBIE</td><td>Writing "0" to the SSR:LBD bit
Writing "1" to the SSRC:LBDC bit</td></tr>
<tr><td>SFD</td><td>SACSR</td><td>Sync Field detection</td><td>SACSR:SFDE</td><td>Writing "1" to the Sync Field detection flag clear bit (SACSRC:SFDC)</td></tr>
<tr><td>TINT</td><td>SACSR</td><td>Coincidence of the serial timer register (STMR) and the serial timer comparison register (STMCR) values</td><td>SACSR:TINTE</td><td>Writing "1" to the timer interrupt flag clear bit (SACSRC:TINTC)</td></tr>
<tr><td rowspan="2">Input Capture</td><td>ICP0</td><td>ICS0</td><td>1st falling edge of the LIN Sync Field</td><td rowspan="2">ICS0:ICE0</td><td rowspan="2">Disabling the ICP0</td></tr>
<tr><td>ICP0</td><td>ICS0</td><td>5th falling edge of the LIN Sync Field</td></tr>
</table></td></tr>
</table>

Page	Section	Change Results
-	-	Correct)

Table 2-1 Interrupt Control Bits and Interrupt Factors of LIN Interface (v2.1) (Manual Mode)

Interrupt Type	Interrupt Request Flag Bit	Flag Register	Interrupt Factor	Interrupt Factor Enable Bit	Clearing of Interrupt Request Flag
Reception	RDRF	SSR	1-byte reception	SCR:RIE	- Reading of reception data (RDR) - Software reset (SCR:UPCL = 1)
			Reception of as much data as the amount set in FBYTE		- Reading of reception data until the reception FIFO becomes empty - Software reset (SCR:UPCL = 1)
			Detection of reception idle for 8-bit time or longer while the FRIIE bit is "1" and the reception FIFO contains valid data		
	ORE	SSR	Overrun error	SCR:RIE ECR:REIE	- Writing "1" to the reception error flag clear bit (SSR:REC) - Software reset (SCR:UPCL = 1)
	FRE	SSR	Framing error		
	RXUDR	ESR	Reception FIFO underrun	ECR:REIE	- Writing "1" to the reception FIFO underrun clear flag bit (ESRC:RXUDRC) - Software reset (SCR:UPCL = 1) - Disable reception FIFO - When FCR1:FSEL=0, set FCR0:FE2=0 - When FCR1:FSEL=1, set FCR0:FE1=0
	RBERR	ESR	Reception block transfer error	ECR:REIE	- Writing "1" to the reception block transfer error clear bit (ESRC:RBERRC) - Software reset(SCR:UPCL=1) - Disable Block transfer mode (ECR:RXBLKEN=0)
Transmission	TDRE	SSR	The transmission register is empty.	SCR:TIE	Writing to the transmission data (TDR), or writing "1" to the transmission FIFO operation enable bit (FCR0:FE1 or FCR0:FE2) (retransmission) while the bit value is "0" and the transmission FIFO contains valid data *1
	TBI	SSR	No transmission operation	SCR:TBIE	Writing to the transmission data (TDR), writing "1" to the LIN Break Field setting bit (LBR), or writing "1" to the transmission FIFO operation enable bit (FCR0:FE1 or FCR0:FE2) (retransmission) while the bit value is "0" and the transmission FIFO contains valid data*1
	FDRQ	FCR1	The amount of data in the transmission FIFO is equal to or smaller than the FTICR setting value or it is empty.	FCR1:FTIE	Writing "1" to the FIFO transmission data request clear bit (FCR1C:FDRQC) , or transmission FIFO full

Interrupt Type	Interrupt Request Flag Bit	Flag Register	Interrupt Factor	Interrupt Factor Enable Bit	Clearing of Interrupt Request Flag
	TXOVR	ESR	Transmission FIFO overrun	ECR:TEIE	- Writing "1" to the transmission FIFO overrun flag clear bit (ESRC:TXOVRC) - Software reset (SCR:UPCL = 1) - Disable transmission FIFO - When FCR1:FSEL=0, set FCR0:FE1=0 - When FCR1:FSEL=1, set FCR0:FE2=0
	TBERR	ESR	Transmission block transfer error	ECR:TEIE	- Writing "1" to the transmission block transfer error clear bit (FSRC:TBERRC) - Software reset (SCR:UPCL=1) - Disable Block transfer mode (ECR:TXBLKEN=0)
Status (Manual Mode)	LBD	SSR	LIN Break Field detection	ESCR:LBIE	- Writing "1" to the SSRC:LBDC bit - Software reset (SCR:UPCL = 1)
	SFD	SACSR	Sync Field detection	SACSR:SFDE	- Writing "1" to the Sync Field detection flag clear bit (SACSRC:SFDC) - Software reset (SCR:UPCL=1)
	TINT	SACSR	Coincidence of the serial timer register (STMR) and the serial timer comparison register (STMCR) values	SACSR:TINTE	- Writing "1" to the timer interrupt flag clear bit (SACSRC:TINTC) - Software reset (SCR:UPCL=1)
Input Capture	ICP0	ICS0	1st falling edge of the LIN Sync Field	ICS0:ICE0	Disabling the ICP0
	ICP0	ICS0	5th falling edge of the LIN Sync Field		

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>2512</td><td>CHAPTER 39: LIN Interface (V2.1)
2. LIN Interface (v2.1) Interrupts
2.2 Assist Mode</td><td>Error)
Table 39.2-4 LIN Interface (v2.1) Interrupt Control Bits and Interrupt Factors (Assist Mode)
<table>
<tr><th>Interrupt Type</th><th>Interrupt Request Flag Bit</th><th>Flag Register</th><th>Interrupt Factor</th><th>Interrupt Factor Enable Bit</th><th>Clearing of Interrupt Request Flag</th></tr>
<tr><td rowspan="6">Reception</td><td rowspan="3">RDRF</td><td rowspan="3">SSR</td><td>1-byte reception</td><td rowspan="3">SCR:RIE</td><td>Reading of reception data (RDR)</td></tr>
<tr><td>Reception of as much data as the amount set in FBYTE</td><td rowspan="2">Reading of reception data (RDR) until the reception FIFO becomes empty</td></tr>
<tr><td>Detection of reception idle for the 8-bit time or longer while the FRIIE bit is "1" and the reception FIFO contains valid data</td></tr>
<tr><td>ORE</td><td>SSR</td><td>Overrun error</td><td>SCR:RIE</td><td rowspan="2">Writing "1" to the reception error flag clear bit (SSR:REC)</td></tr>
<tr><td>FRE</td><td>SSR</td><td>Framing error</td><td>ECR:REIE</td></tr>
<tr><td>LBSER</td><td>LAMESR</td><td>LIN bus error detection</td><td>LAMIER: LBSERIE</td><td>Writing "1" to LAMESRC:LBSERC</td></tr>
<tr><td>LSFER</td><td>LAMESR</td><td>LIN Sync Data error detection</td><td>LAMIER: LSFERIE</td><td>Writing "1" to LAMESRC:LSFERC</td></tr>
<tr><td>LPTER</td><td>LAMESR</td><td>LIN ID parity error detection</td><td>LAMIER: LPTERIE</td><td>Writing "1" to LAMESRC:LPTERC</td></tr>
</table>
<table>
<tr><th>Interrupt Type</th><th>Interrupt Request Flag Bit</th><th>Flag Register</th><th>Interrupt Factor</th><th>Interrupt Factor Enable Bit</th><th>Clearing of Interrupt Request Flag</th></tr>
<tr><td></td><td>LCSER</td><td>LAMESR</td><td>LIN checksum error detection</td><td>LAMIER: LCSERIE</td><td>Writing "1" to LAMESRC:LCSERC</td></tr>
<tr><td></td><td>RXUDR</td><td>ESR</td><td>Reception FIFO underrun</td><td>ECR:REIE</td><td>Writing "1" to the reception FIFO underrun flag clear bit (ESRC:RXUDRC)</td></tr>
<tr><td></td><td>RBERR</td><td>ESR</td><td>Reception block transfer error</td><td>ECR:REIE</td><td>Writing "1" to the reception block transfer error clear bit</td></tr>
<tr><td rowspan="5">Transmission</td><td>TDRE</td><td>SSR</td><td>The transmission register is empty.</td><td>SCR:TIE</td><td>Writing to the transmission data (TDR) , or writing "1" to the transmission FIFO operation enable bit (FCR0:FE1 or FCR0:FE2) (retransmission) when the bit value is "0" and the transmission FIFO contains valid data*1</td></tr>
<tr><td>TBI</td><td>SSR</td><td>No transmission operation</td><td>SCR:TBIE</td><td>Writing to the transmission data (TDR), writing "1" to the LIN Break Field setting bit (LBR), or writing "1" to the transmission FIFO operation enable bit (retransmission) when the bit value is "0" and the transmission FIFO contains valid data *1</td></tr>
<tr><td>FDRQ</td><td>FCR1</td><td>The amount of data in the transmission FIFO is equal to or less than the FTICR setting value or it is empty.</td><td>FCR1:FTIE</td><td>Writing "1" to the FIFO transmission data request clear bit (FCR1C:FDRQC) , or the transmission FIFO is full</td></tr>
<tr><td>TXOVR</td><td>ESR</td><td>Transmission FIFO overrun</td><td>ECR:TEIE</td><td>Writing "1" to the transmission FIFO overrun flag clear bit (ESRC:TVOVRC)</td></tr>
<tr><td>TBERR</td><td>ESR</td><td>Transmission block transfer error</td><td>ESR:TEIE</td><td>Writing "1" to the transmission block transfer error clear bit (ESRC:TBERRC)</td></tr>
<tr><td rowspan="3">Status (Assist Mode)</td><td>LBD</td><td>SSR</td><td>LIN Break Field detection</td><td>ESCR:LBIE</td><td>Writing "1" to the SSRC:LBDC bit</td></tr>
<tr><td>SFD</td><td>SACSR</td><td>Sync Field detection</td><td>SACSR: SFDE</td><td>Writing "1" to the Sync Field detection flag clear bit (SACSRC:SFDC)</td></tr>
<tr><td>TINT</td><td>SACSR</td><td>Coincidence of the value in the serial timer register (STMR) and that in the serial timer comparison register (STMCR)</td><td>SACSR: TINTE</td><td>Writing "1" to the timer interrupt flag clear bit (SACSRC:TINTC)</td></tr>
</table>
<table>
<tr><th>Interrupt Type</th><th>Interrupt Request Flag Bit</th><th>Flag Register</th><th>Interrupt Factor</th><th>Interrupt Factor Enable Bit</th><th>Clearing of Interrupt Request Flag</th></tr>
<tr><td></td><td>LAHC</td><td>LAMSR</td><td>Completion of auto header</td><td>LAMIER: LAHCIE</td><td>Writing "1" to LAMSRC:LAHCC
or
reading from the ID register (LAMRID)</td></tr>
<tr><td></td><td>LCSC</td><td>LAMSR</td><td>Completion of checksum calculation</td><td>LAMIER: LCSCIE</td><td>Writing "1" to LAMSRC:LCSCC</td></tr>
</table></td></tr>
</table>

Page	Section	Change Results
-	-	Correct)

Table 2-4 LIN Interface (v2.1) Interrupt Control Bits and Interrupt Factors (Assist Mode)

Interrupt Type	Interrupt Request Flag Bit	Flag Register	Interrupt Factor	Interrupt Factor Enable Bit	Clearing of Interrupt Request Flag
Reception	RDRF	SSR	1-byte reception	SCR:RIE	- Reading of reception data (RDR) - Software reset (SCR:UPCL = 1)
			Reception of as much data as the amount set in FBYTE		- Reading of reception data (RDR) until the reception FIFO becomes empty - Software reset (SCR:UPCL = 1)
			Detection of reception idle for the 8-bit time or longer while the FRIIE bit is "1" and the reception FIFO contains valid data		
	ORE	SSR	Overrun error	SCR:RIE ECR:REIE	- Writing "1" to the reception error flag clear bit (SSR:REC) - Software reset (SCr:UPCL = 1)
	FRE	SSR	Framing error		
	LBSER	LAMESR	LIN bus error detection	LAMIER: LBSERIE	- Writing "1" to LAMESRC:LBSERC - Software reset (SCr:UPCL = 1)
	LSFER	LAMESR	LIN Sync Data error detection	LAMIER: LSFERIE	- Writing "1" to LAMESRC:LSFERC - Software reset (SCr:UPCL = 1)
	LPTER	LAMESR	LIN ID parity error detection	LAMIER: LPTERIE	- Writing "1" to LAMESRC:LPTERC - Software reset (SCr:UPCL = 1)
	LCSER	LAMESR	LIN checksum error detection	LAMIER: LCSERIE	- Writing "1" to LAMESRC:LCSERC - Software reset (SCr:UPCL = 1)
	RXUDR	ESR	Reception FIFO underrun	ECR:REIE	- Writing "1" to the reception FIFO underrun flag clear bit (ESRC:RXUDRC) - Software reset (SCr:UPCL = 1) - Disable reception FIFO - When FCR1:FSEL=0, set FCR0:FE2=0 - When FCR1:FSEL=1, set FCR0:FE1=0
	RBERR	ESR	Reception block transfer error	ECR:REIE	- Writing "1" to the reception block transfer error clear bit - Software reset (SCR:UPCL=1) - Disable Block transfer mode (ECR:RXBLKEN=0)
Transmission	TDRE	SSR	The transmission register is empty.	SCR:TIE	Writing to the transmission data (TDR) , or writing "1" to the transmission FIFO operation enable bit (FCR0:FE1 or FCR0:FE2) (retransmission) when the bit value is "0" and the transmission FIFO contains valid data*1
	TBI	SSR	No transmission operation	SCR:TBIE	Writing to the transmission data (TDR), writing "1" to the LIN Break Field setting bit (LBR), or writing "1" to the transmission FIFO operation enable bit (retransmission) when the bit value is "0" and the transmission FIFO contains valid data *1
	FDRQ	FCR1	The amount of data in the transmission FIFO is equal to or less than the FTICR setting value or it is empty.	FCR1:FTIE	Writing "1" to the FIFO transmission data request clear bit (FCR1C:FDRQC) , or the transmission FIFO is full

Interrupt Type	Interrupt Request Flag Bit	Flag Register	Interrupt Factor	Interrupt Factor Enable Bit	Clearing of Interrupt Request Flag
	TXOVR	ESR	Transmission FIFO overrun	ECR:TEIE	- Writing "1" to the transmission FIFO overrun flag clear bit (ESRC:TXOVRC) - Software reset (SCR:UPCL=1) - Disable transmission FIFO - When FCR1:FSEL=0, set FCR0:FE1=0 - When FCR1:FSEL=1, set FCR0:FE2=0
	TBERR	ESR	Transmission block transfer error	ECR:TEIE	- Writing "1" to the transmission block transfer error clear bit (ESRC:TBERRC) - Software reset (SCR:UPCL=1) - Disable Block transfer (ECR:TXBLKEN=0)
Status (Assist Mode)	LBD	SSR	LIN Break Field detection	ESCR:LBIE	- Writing "1" to the SSRC:LBDC bit - Software reset (SCR:UPCL=1)
	SFD	SACSR	Sync Field detection	SACSR: SFDE	- Writing "1" to the Sync Field detection flag clear bit (SACSRC:SFDC) - Software reset (SCR:UPCL=1)
	TINT	SACSR	Coincidence of the value in the serial timer register (STMR) and that in the serial timer comparison register (STMCR)	SACSR: TINTE	- Writing "1" to the timer interrupt flag clear bit (SACSRC:TINTC) - Software reset (SCR:UPCL=1)
	LAHC	LAMSR	Completion of auto header	LAMIER: LAHCIE	- Writing "1" to LAMSRC:LAHCC - Reading from the ID register (LAMRID) - Software reset (SCR:UPCL = 1)
	LCSC	LAMSR	Completion of checksum calculation	LAMIER: LCSCIE	- Writing "1" to LAMSRC:LCSCC - Software reset (SCR:UPCL = 1)

Page	Section	Change Results
2522	CHAPTER 39: LIN Interface (V2.1) 2. LIN Interface (v2.1) Interrupts	Error) LIN data length (LAMCR:LDL) Correct) LIN data length (LAMCR:LDL3-0) Error) LIN data length (LAMCR:LDL) Correct) LIN data length (LAMCR:LDL3-0)
2523	CHAPTER 39: LIN Interface (V2.1) 2. LIN Interface (v2.1) Interrupts	Error) LIN data length (LAMCR:LDL) Correct) LIN data length (LAMCR:LDL3-0) Error) LIN data length (LAMCR:LDL) Correct) LIN data length (LAMCR:LDL3-0)
2547	CHAPTER 39: LIN Interface (V2.1) 6. Block Transfer	Error) The block size can be set with the transmission block size register and the reception block size register. Correct) The block size can be set with the transmission block size register and FBYTE register (for the reception side).
2550	CHAPTER 39: LIN Interface (V2.1) 6. Block Transfer 6.1 ACK Mode	Error) **Transmission Block Transfer Error** If, while the Write counter is "0", an acknowledgment is input before the next transmission block transfer request (TIRQ) is asserted, a transmission block transfer error occurs, and the FSR:TBERR bit is asserted. Correct) **Transmission Block Transfer Error** If, while the Write counter is "0", an acknowledgment is input before the next transmission block transfer request (TIRQ) is asserted, a transmission block transfer error occurs, and the ESR:TBERR bit is asserted.
2553	CHAPTER 39: LIN Interface (V2.1) 6. Block Transfer 6.2 RW Access Mode	Error) After the reception block transfer request, if a read access to the reception FIFO occurs, the reception block transfer request (RIRQ) is cleared. With a read access to the reception FIFO, the Read counter is decremented. Correct) After the reception block transfer request, if reading a reception data occurs, the reception block transfer request (RIRQ) is cleared. With reading a reception data, the Read counter is decremented.

Page	Section	Change Results
2554	CHAPTER 39: LIN Interface (V2.1) 6. Block Transfer 6.2 RW Access Mode	Error) **Transmission Block Transfer** After the transmission block transfer request, if a write access to the transmission FIFO occurs, the transmission block transfer request (TIRQ) is cleared. With a write access to the transmission FIFO, the Write counter is decremented. Correct) **Transmission Block Transfer** After the transmission block transfer request, if writing a transmission data occurs, the transmission block transfer request (TIRQ) is cleared. By writing a transmission data, the Write counter is decremented.
2555	CHAPTER 39: LIN Interface (V2.1) 6. Block Transfer 6.2 RW Access Mode	Error) **Transmission Block Transfer Error** If, while the Write counter is "0", a write access occurs before the next transmission block transfer request (TIRQ) is asserted, a transmission block transfer error occurs, and the FSR:TBERR bit is asserted. Correct) **Transmission Block Transfer Error** If, while the Write counter is "0", a write access occurs before the next transmission block transfer request (TIRQ) is asserted, a transmission block transfer error occurs, and the ESR:TBERR bit is asserted.
2558	CHAPTER 39: LIN Interface (V2.1) 7. LIN Interface (V2.1) Operation 7.1 Manual Mode	Error) □□When the DATA Field is received, SSR:RDRF is set to 1. At this time, a reception interrupt occurs if the reception interrupt is enabled (SSR:RIE = 1). Correct) □□When the DATA Field is received, SSR:RDRF is set to 1. At this time, a reception interrupt occurs if the reception interrupt is enabled (SCR:RIE = 1).
2565	CHAPTER 39: LIN Interface (V2.1) 7. LIN Interface (V2.1) Operation 7.1 Manual Mode	Error) ■ When the value of serial timer register (STMR) is between that of the Sync Field lower limit register (SFLR) and Sync Field upper limit register (SFUR), the value of the serial timer register (STMR) is set in the baud rate generator register (BGR), and the baud rate setting flag (SACSR:BST) is set to "1". ■ When the value of serial timer register (STMR) is below that of the Sync Field lower limit register (SFLR) or exceeds that of the Sync Field upper limit register (SFUR), the baud rate generator register (BGR) will not be changed and the baud rate setting flag (SACSR:BST) is reset to "0". Correct) − When the value of serial timer register (STMR) is between that of the Sync Field lower limit register (SFLR) and Sync Field upper limit register (SFUR), the value of the serial timer register (STMR) is set in the baud rate generator register (BGR1, BGR0), and the baud rate setting flag (SACSR:BST) is set to "1". − When the value of serial timer register (STMR) is below that of the Sync Field lower limit register (SFLR) or exceeds that of the Sync Field upper limit register (SFUR), the baud rate generator register (BGR1, BGR0) will not be changed and the baud rate setting flag (SACSR:BST) is reset to "0".

Page	Section	Change Results
2574	CHAPTER 39: LIN Interface (V2.1) 7. LIN Interface (V2.1) Operation 7.2 Assist Mode	Error) ■ When the LIN data length is set to 0 bytes (LAMCR:LDL3 to LD0 = 0000) for response transmission, write dummy data into the TDR register to automatically perform the checksum calculation and then transmit the data. (Any value can be written.) The TDR setting value does not affect the checksum calculation. ■ During the transmission of response data, do not change the transmission data length (LAMCR:LDL[3:0]). Correct) – *When the LIN data length is set to 0 bytes (LAMCR:LDL3 to LDL0 = 0000) for response transmission, write dummy data into the TDR register to automatically perform the checksum calculation and then transmit the data. (Any value can be written.) The TDR setting value does not affect the checksum calculation.* – *During the transmission of response data, do not change the transmission data length (LAMCR:LDL3 to LDL0).*
2575	CHAPTER 39: LIN Interface (V2.1) 7. LIN Interface (V2.1) Operation 7.2 Assist Mode	Error) ■ When the DATA Field is received, SSR:RDRF is set to 1. At this time, a reception interrupt occurs if a reception interrupt is enabled (SSR:RIE = 1). ■ During the reception of response data, do not change the reception data length (LAMCR:LDL[3:0]). Correct) ■ When the DATA Field is received, SSR:RDRF is set to 1. At this time, a reception interrupt occurs if a reception interrupt is enabled (SCR:RIE = 1). – *During the reception of response data, do not change the reception data length (LAMCR:LDL3 to LDL0).*

Page	Section	Change Results
2582	CHAPTER 39: LIN Interface (V2.1) 7. LIN Interface (V2.1) Operation 7.2 Assist Mode	Error) ■ When the value of the serial timer register (STMR) is between that of the Sync Field lower limit register (SFLR) and Sync Field upper limit register (SFUR), the value of the serial timer register (STMR) is set in the baud rate generator register (BGR), and the baud rate setting flag (SACSR:BST) is set to "1". ■ When the value of the serial timer register (STMR) is below that of the Sync Field lower limit register (SFLR) or exceeds that of the Sync Field upper limit register (SFUR), the baud rate generator register (BGR) will not be changed and the baud rate setting flag (SACSR:BST) is reset to "0". Correct) ■ When the value of the serial timer register (STMR) is between that of the Sync Field lower limit register (SFLR) and Sync Field upper limit register (SFUR), the value of the serial timer register (STMR) is set in the baud rate generator register (BGR1, BGR0), and the baud rate setting flag (SACSR:BST) is set to "1". ■ When the value of the serial timer register (STMR) is below that of the Sync Field lower limit register (SFLR) or exceeds that of the Sync Field upper limit register (SFUR), the baud rate generator register (BGR1, BGR0) will not be changed and the baud rate setting flag (SACSR:BST) is reset to "0".
2587	CHAPTER 39: LIN Interface (V2.1) 7. LIN Interface (V2.1) Operation 7.2 Assist Mode	Error) Notes: ■ For slave operation, when the LIN communication speed is 19.2 kbps, within about 25 µs after the auto header completion flag is set (LAMSR:LAHC = 1), set the checksum type (LAMCR:LCSTYP) and the LIN data length (LAMCR:LDL[3:0]), and disable the transmission (SCR:TXE), the transmission interrupt (SCR:TIE), the reception (SCR:RXE) and the reception interrupt (SCR:RIE) . (Set to a value less than half of one cycle time for LIN communication.) Correct) Notes: *− For slave operation, when the LIN communication speed is 19.2 kbps, within about 25 µs after the auto header completion flag is set (LAMSR:LAHC = 1), set the checksum type (LAMCR:LCSTYP) and the LIN data length (LAMCR:LDL3 to LDL0), and disable the transmission (SCR:TXE), the transmission interrupt (SCR:TIE), the reception (SCR:RXE) and the reception interrupt (SCR:RIE) . (Set to a value less than half of one cycle time for LIN communication.)*

Page	Section	Change Results
2610	CHAPTER 39: LIN Interface (V2.1) 9. Interface (V2.1) Registers 9.2 Serial Mode Register (SMR)	Error) **[bit7:5] MD[2:0]: Operation Mode Setting Bits** These bits set the operation mode. 0b000: Set operation mode 0 (asynchronous normal mode). 0b001: Set operation mode 1 (asynchronous multi-processor mode). 0b010: Set operation mode 2 (clock synchronous mode). 0b011: Set operation mode 3 (LIN communication mode). 0b100: Set operation mode 4 (I2C mode). This chapter describes the register and its operations in operation mode 3 (LIN communication mode). Correct) **[bit7:5] MD[2:0]: Operation Mode Setting Bits** These bits set the operation mode.
2613	CHAPTER 39: LIN Interface (V2.1) 9. Interface (V2.1) Registers 9.3 Serial Status Register (SSR)	Error) **[bit13] LBD: LIN Break Field Detection Flag Bit** Correct) **[bit13] LBD: LIN Break Field Detection Flag Bit** ***Note:*** *− This bit is set "0" when Software reset (SCR:UPCL=1) is done.*
2613	CHAPTER 39: LIN Interface (V2.1) 9. Interface (V2.1) Registers 9.3 Serial Status Register (SSR)	Error) **[bit12] FRE: Framing Error Flag Bit** ***Notes:*** Correct) **[bit12] FRE: Framing Error Flag Bit** ***Notes:*** *− This bit is set "0" when Software reset (SCR:UPCL=1) is done.*
2614	CHAPTER 39: LIN Interface (V2.1) 9. Interface (V2.1) Registers 9.3 Serial Status Register (SSR)	Error) **[bit11] ORE: Overrun Error Flag Bit** ***Notes:*** Correct) **[bit11] ORE: Overrun Error Flag Bit** ***Notes:*** *− This bit is set "0" when Software reset (SCR:UPCL=1) is done.*

Page	Section	Change Results
2614	CHAPTER 39: LIN Interface (V2.1) 9. Interface (V2.1) Registers 9.3 Serial Status Register (SSR)	Error) **[bit10] RDRF: Reception Data Full Flag Bit** Correct) **[bit10] RDRF: Reception Data Full Flag Bit** ■ When reception FIFO available, SSR:RDRF is set "1" when conditions under below and the condition that the reception idle state continues for at least 8 clock pulses of the baud rate clock are satisfied. – The reception FIFO idle detection enable bit (FCR1:FRIIE) is set to "1". – The reception FIFO has not received the prescribed number of data items, and data remains in the reception FIFO. During an 8-clock count, the counter is reset to 0 when RDR is read, and starts to count the 8clocks again. Notes: – *This bit is set "0" when Software reset (SCR:UPCL = 1) is done.*
2615	CHAPTER 39: LIN Interface (V2.1) 9. Interface (V2.1) Registers 9.3 Serial Status Register (SSR)	Error) **[bit9] TDRE: Transmission Data Empty Flag Bit** ■ The TDRE bit is set to "1" when "1" is written to the UPCL bit in the serial control register (SCR). Correct) **[bit9] TDRE: Transmission Data Empty Flag Bit** ■ The TDRE bit is set to "1" when programmable clear (SCR:UPCL=1) is done.
2617	CHAPTER 39: LIN Interface (V2.1) 9. Interface (V2.1) Registers 9.4 Extended Communication Control Register (ESCR)	Error) **[bit5, bit3:2] LBL[2:0]: LIN Break Field Length Selection Bits (Effective Only for Master Operation)** ■ Set these bits before setting the LBR bit in the serial control register (SCR) to "1" (LIN Break Field transmission). Correct) **[bit5, bit3:2] LBL[2:0]: LIN Break Field Length Selection Bits (Effective Only for Master Operation)** ■ Set these bits before setting the LIN Break Field setting bit (SCR:LBR) to "1".

Page	Section	Change Results
2617	CHAPTER 39: LIN Interface (V2.1) 9. Interface (V2.1) Registers 9.4 Extended Communication Control Register (ESCR)	Error) **[bit1:0] DEL[1:0]: LIN Break Delimiter Length Selection Bits (Effective Only for Master Operation)** ■ Set these bits before setting the LBR bit in the serial control register (SCR) to "1" (LIN Break Field transmission). Correct) **[bit1:0] DEL[1:0]: LIN Break Delimiter Length Selection Bits (Effective Only for Master Operation)** ■ Set these bits before setting the LIN Break Field setting bit(SCR:LBR) to "1".
2626	CHAPTER 39: LIN Interface (V2.1) 9. Interface (V2.1) Registers 9.8 Serial Timer Comparison Register (STMCR)	Error) **[bit15:0] TC15-0: Compare Bits** **Note:** ■ The timer interrupt flag (SACSR:TINT) is fixed to "1" when this register is set to 0x0000, the timer is operating, and the division value (SACSR:TDIV) of the timer operation clock is set to 0b0000. Correct) **[bit15:0] TC15-0: Compare Bits** The period below is (STMCR:TC+1) X The timer clock (set by SACSR:TDIV3-0) ・SACSR:TINT is set "1" ***Notes:*** − *The timer interrupt flag (SACSR:TINT) is fixed to "1" when all of the following conditions are satisfied.* − Synchronous transmission is disabled (SACSR:TSYNE = 0). − 0x0000 is set in this register. − The timer is operating. − 0b0000 is set as the timer operating clock division value (SACSR:TDIV3-0).
2627	CHAPTER 39: LIN Interface (V2.1) 9. Interface (V2.1) Registers 9.9 Sync Field Upper Limit Register (SFUR)	Error) **[bit14:0] TU14-0: Upper Limit Bits** These bits set the upper limit on the values that can be set in the baud rate generator register (BGR) for auto baud rate adjustment. Then, the baud rate generator register (BGR) is set to the value of the serial timer register (STMR). Correct) **[bit14:0] TU14-0: Upper Limit Bits** These bits set the upper limit on the values that can be set in the baud rate generator register (BGR1,BGR0) for auto baud rate adjustment. Then, the baud rate generator register (BGR1, BGR0) is set to the value of the serial timer register (STMR).

Page	Section	Change Results
2628	CHAPTER 39: LIN Interface (V2.1) 9. Interface (V2.1) Registers 9.10 Sync Field Lower Limit Register (SFLR)	Error) **[bit14:0] TL14-0: Lower Limit Bits** These bits set the lower limit on the values that can be set in the baud rate generator register (BGR) for auto baud rate adjustment. Then, the baud rate generator register (BGR) is set to the value of the serial timer register (STMR). Correct) **[bit14:0] TL14-0: Lower Limit Bits** These bits set the lower limit on the values that can be set in the baud rate generator register (BGR1,BGR0) for auto baud rate adjustment. Then, the baud rate generator register (BGR1, BGR0) is set to the value of the serial timer register (STMR).
2632	CHAPTER 39: LIN Interface (V2.1) 9. Interface (V2.1) Registers 9.12 LIN Assist Mode Status Register (LAMSR)	Error) **[bit10] LCSC: LIN Checksum Calculation Completion Flag Bit** ***Notes:*** Correct) **[bit10] LCSC: LIN Checksum Calculation Completion Flag Bit** ***Notes:*** − *This bit is set "0" when Software reset is done (SCR:UPCL=1).*
2633	CHAPTER 39: LIN Interface (V2.1) 9. Interface (V2.1) Registers 9.12 LIN Assist Mode Status Register (LAMSR)	Error) **[bit8] LAHC: LIN Auto Header Completion Flag Bit** ***Notes:*** Correct) **[bit8] LAHC: LIN Auto Header Completion Flag Bit** ***Notes:*** − *This bit is set "0" when Software reset is done (SCR:UPCL=1).*

Page	Section	Change Results
2641	CHAPTER 39: LIN Interface (V2.1) 9. Interface (V2.1) Registers 9.16 LIN Assist Mode Error Status Register (LAMESR)	Error) **[bit14] LCSER: LIN Checksum Error Flag Bit** ***Notes:*** Correct) **[bit14] LCSER: LIN Checksum Error Flag Bit** ***Notes:*** − *This bit is set "0" when Software reset is done (SCR:UPCL=1).*
2642	CHAPTER 39: LIN Interface (V2.1) 9. Interface (V2.1) Registers 9.16 LIN Assist Mode Error Status Register (LAMESR)	Error) **[bit13] LPTER: LIN ID Parity Error Flag Bit** ***Notes:*** Correct) **[bit13] LPTER: LIN ID Parity Error Flag Bit** ***Notes:*** − *This bit is set "0" when Software reset is done (SCR:UPCL=1).*
2642	CHAPTER 39: LIN Interface (V2.1) 9. Interface (V2.1) Registers 9.16 LIN Assist Mode Error Status Register (LAMESR)	Error) **[bit12] LSFER: LIN Sync Data Error Flag Bit** ***Notes:*** Correct) **[bit12] LSFER: LIN Sync Data Error Flag Bit** ***Notes:*** − *This bit is set "0" when Software reset is done (SCR:UPCL=1).*
2642	CHAPTER 39: LIN Interface (V2.1) 9. Interface (V2.1) Registers 9.16 LIN Assist Mode Error Status Register (LAMESR)	Error) **[bit11] LBSER: LIN Bus Error Flag Bit** ***Notes:*** Correct) **[bit11] LBSER: LIN Bus Error Flag Bit** ***Notes:*** − *This bit is set "0" when Software reset is done (SCR:UPCL=1).*

Page	Section	Change Results
2646	CHAPTER 39: LIN Interface (V2.1) 9. Interface (V2.1) Registers 9.18 FIFO Control Register 1 (FCR1)	Error) **[bit12] FLSTE: Retransmission Data Lost Detection Enable Bit** This bit enables FLST bit detection. Correct) **[bit12] FLSTE: Retransmission Data Lost Detection Enable Bit** This bit enables the detection of FIFO retransmission Data Lost Flag Bit (FCR0:FLST).
2647	CHAPTER 39: LIN Interface (V2.1) 9. Interface (V2.1) Registers 9.18 FIFO Control Register 1 (FCR1)	Error) **[bit10] FDRQ: Transmission FIFO Data Request Bit** ■ When transmission FIFO interrupt control is used ☐ FTICR setting value ≥ FTICR read value (The amount of data in the transmission FIFO is equal to the interrupt trigger level or lower.) ☐ Reset of the transmission FIFO FDRQ clearing conditions ■ "1" is written to FRC1C:FDRQC. Correct) **[bit10] FDRQ: Transmission FIFO Data Request Bit** ■ When transmission FIFO interrupt control is used − When ECR:TXBLKEN=0 − FTICR setting value ≥ FTICR read value (The amount of data in the transmission FIFO is equal to the interrupt trigger level or lower.) − When ECR:TXBLKEN=1 − FTICR setting value ≦ the number of transmission FIFO empty data − Reset of the transmission FIFO FDRQ clearing conditions ■ "1" is written to FCR1C:FDRQC.
2649	CHAPTER 39: LIN Interface (V2.1) 9. Interface (V2.1) Registers 9.19. FIFO Control Register 0 (FCR0)	Error) **[bit7] Reserved: Reserved Bit** Correct) **[bit7] Reserved: Bit** When read, always it can be read "0". When written, always it is set "0".

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>2649</td><td>CHAPTER 39: LIN Interface (V2.1)
9. Interface (V2.1) Registers
9.19. FIFO Control Register 0 (FCR0)</td><td>Error)
[bit6] FLST: FIFO Retransmission Data Lost Flag Bit
FLST clearing conditions
■ FIFO reset (writing "1" to FCL)

Correct)
[bit6] FLST: FIFO Retransmission Data Lost Flag Bit
FLST clearing conditions
■ transmission FIFO reset
− When FCR1:FSEL=0, set "1" to the FCR0:FCL1
− When FCR1:FSEL=1, set "1" to the FCR0:FCL2</td></tr>
<tr><td>2650</td><td>CHAPTER 39: LIN Interface (V2.1)
9. Interface (V2.1) Registers
9.19. FIFO Control Register 0 (FCR0)</td><td>Error)
[bit4] FSET: FIFO Pointer Saving Bit

<table><tr><th>Bit</th><th>FIFO Pointer Saving Bit</th></tr><tr><td>0</td><td>Do not save.</td></tr><tr><td>1</td><td>Save the read pointer value.</td></tr></table>
Correct)
[bit4] FSET: FIFO Pointer Saving Bit

<table><tr><th rowspan="2">Bit</th><th colspan="2">FIFO Pointer Saving Bit</th></tr><tr><th>Write</th><th>Read</th></tr><tr><td>0</td><td>Do not save.</td><td rowspan="2">Always read “0”</td></tr><tr><td>1</td><td>Save the read pointer value.</td></tr></table></td></tr>
<tr><td>2651</td><td>CHAPTER 39: LIN Interface (V2.1)
9. Interface (V2.1) Registers
9.19. FIFO Control Register 0 (FCR0)</td><td>Error)
[bit1] FE2: FIFO2 Operation Enable Bit
■ When FIFO2 is specified as the transmission FIFO and "1" is written to this bit, transmission starts immediately if FIFO2 contains data and LIN interface (v2.1) transmission is enabled (SCR:TXE = 1). At this time, to set the TIE and TBIE bits to "1", first set them to "0" and write "1" to this bit, and then set the TIE and TBIE bits to "1".

Correct)
[bit1] FE2: FIFO2 Operation Enable Bit
■ When FIFO2 is specified as the transmission FIFO (FCR1:FSEL=1) and "1" is written to this bit, transmission starts immediately if FIFO2 contains data and LIN interface (v2.1) transmission is enabled (SCR:TXE = 1). At this time, to set the TIE and TBIE bits to "1", first set them to "0" and write "1" to this bit, and then set the TIE and TBIE bits to "1".</td></tr>
</table>

Page	Section	Change Results
2653	CHAPTER 39: LIN Interface (V2.1) 9. Interface (V2.1) Registers 9.20. FIFO Byte Register (FBYTE)	Error) ■ The initial transfer count value for the FBYTE register is 0x08. Therefore, set 0x00 as the transfer count of the FBYTE to the transmission FIFO selected by the FSEL. ■ When both of the following conditions are satisfied, the continuation of the reception idle status for 8 baud rate clocks or longer sets the interrupt flag (RDRF) to "1". ☐ Reception FIFO idle detection enable bit (FRIIE) is "1". ☐ The number of data items in the reception FIFO does not reach the transfer count. During an 8-clock count, the counter is reset to 0 when RDR is read, and the system starts counting the 8 clocks again. The counter is reset to 0 when the reception FIFO is disabled. If the reception FIFO is enabled when data remains in the reception FIFO, counting restarts. Correct) ■ The initial transfer count value for the FBYTE register is 0x08. ■ When both of the following conditions are satisfied, the continuation of the reception idle status for 8 baud rate clocks or longer sets the interrupt flag (RDRF) to "1". − - Reception FIFO idle detection enable bit (FRIIE) is "1". − - The number of data items in the reception FIFO does not reach the transfer count. During an 8-clock count, the counter is reset to 0 when RDR is read, and the system starts counting the 8 clocks again. The counter is reset to 0 when the reception FIFO is disabled. If the reception FIFO is enabled when data remains in the reception FIFO, counting restarts. − The value of FBYTE of transmission FIFO is incremented by 1 when a transmission data is written to TDR 1 time. − The value of FBYTE of reception FIFO is decremented by 1 when a reception data is read from RDR 1 time.
2655	CHAPTER 39: LIN Interface (V2.1) 9. Interface (V2.1) Registers 9.21. Transmission FIFO Interrupt Control Register (FTICR)	Error) ■ ECR:TXBLKEN = 0 The FTICR register sets the threshold for interrupts according to the valid transmission data count (remaining amount) of the transmission FIFO. The table below shows the setting conditions for different FCR1:FSEL bit values. ■ ECR:TXBLKEN = 1 The FTICR register sets the trigger level for interrupts according to the vacant data count of the transmission FIFO. Correct) ■ ECR:TXBLKEN = 0 ■ The FTICR is set the trigger level to interrupt for transmission FIFO's transmission valid data number (remained data number). ■ ECR:TXBLKEN = 1 The FTICR register sets the trigger level for interrupts according to the vacant data count of the transmission FIFO. The table below shows setting values of the FCR1:FSEL bit.

Page	Section	Change Results
2659	CHAPTER 39: LIN Interface (V2.1) 9. Interface (V2.1) Registers 9.23. Extended Status Register (ESR)	Error) **[bit11] RXUDR: Reception FIFO Underrun Flag Bit** ***Notes:*** Correct) **[bit11] RXUDR: Reception FIFO Underrun Flag Bit** ***Notes:*** − *This bit is "0" when reception FIFO is not available.*
2660	CHAPTER 39: LIN Interface (V2.1) 9. Interface (V2.1) Registers 9.23. Extended Status Register (ESR)	Error) **[bit10] TXOVR: Transmission FIFO Overrun Flag Bit** ***Notes:*** ■ When the transmission FIFO is OFF, and block transfer is performed under any one of the following conditions, this bit is set to "1" to indicate a block transfer error. ☐ A transmission interrupt is enabled (SCR:TIE = 1) and the transmission data register contains data (SSR:TDRE = 0). ☐ A transmission interrupt is disabled (SCR:TIE = 0), and a transmission bus idle interrupt is enabled (SCR:TBIE = 1) and transmission is in progress (SSR:TBI = 0). If a transmission interrupt is enabled (SCR:TIE = 1) and a transmission bus idle interrupt is enabled (SCR:TBIE = 1), the condition for enabling transmission interrupt (SCR:TIE = 1) is given priority. Correct) **[bit10] TXOVR: Transmission FIFO Overrun Flag Bit** ***Notes:*** − *This bit is "0" when transmission FIFO is not available.*

Page	Section	Change Results
2660	CHAPTER 39: LIN Interface (V2.1) 9. Interface (V2.1) Registers 9.23. Extended Status Register (ESR)	Error) **[bit9] RBERR: Reception Block Transfer Error Bit** Suppose that block transfer is performed with a value larger than the threshold specified in the FBYTE register, this bit is set to "1" to indicate a block transfer error. A reception interrupt occurs if reception interrupt is enabled (SSR:RIE = 1). ***Notes:*** − *This bit is "0" when the reception block transfer enable bit (ECR:RXBLKEN) is "0".* Correct) **[bit9] RBERR: Reception Block Transfer Error Bit** Suppose that block transfer is performed with a value larger than the threshold specified in the FBYTE register, this bit is set to "1" to indicate a block transfer error. A reception interrupt occurs if reception interrupt is enabled (SCR:RIE = 1). ***Notes:*** − *This bit is "0" when the reception block transfer enable bit (ECR:RXBLKEN) is "0" or when the reception interrupt enable bit (SCR:RIE) is "0".*
2667	CHAPTER 39: LIN Interface (V2.1) 9. Interface (V2.1) Registers 9.29. Serial Auxiliary Control Status Clear Register (SACSRC)	Error) **[bit13] SFDC: Clearing the Sync Field Detection Flag** Writing "1" to this bit resets SACSR:CSEIE to "0". **[bit12] SFDEC: Clearing the Sync Field Detection Interrupt Enable Bit** Writing "1" to this bit resets SACSR:CSE to "0". Correct) **[bit13] SFDC: Clearing the Sync Field Detection Flag** Writing "1" to this bit resets SACSR:SFD to "0". **[bit12] SFDEC: Clearing the Sync Field Detection Interrupt Enable Bit** Writing "1" to this bit resets SACSR:SFDE to "0".

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>2671</td><td>CHAPTER 39: LIN Interface (V2.1) 9. Interface (V2.1) Registers 9.32. LIN Assist Mode Interrupt Enable Clear Register (LAMIERC)</td><td>Error)
<table>
<tr><td>Bit</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7 ... 0</td></tr>
<tr><td>Field</td><td>Reserved</td><td>LCSERIEC</td><td>LPTERIEC</td><td>LSFERIEC</td><td>LBSERIEC</td><td>LCSCIEC</td><td>Reserved</td><td>LAHCIEC</td><td>Reserved</td></tr>
<tr><td>R/W Attribute</td><td>R0, W0</td><td>R0,W</td><td>R0,W</td><td>R0,W</td><td>R0,W</td><td>R0,W</td><td>R0, W0</td><td>R0,W</td><td></td></tr>
<tr><td>Protection Attribute</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td></td></tr>
<tr><td>Initial Value</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td></td></tr>
</table>
Correct)
<table>
<tr><td>Bit</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7 ... 0</td></tr>
<tr><td>Field</td><td>Reserved</td><td>LCSERIEC</td><td>LPTERIEC</td><td>LSFERIEC</td><td>LBSERIEC</td><td>LCSCIEC</td><td>Reserved</td><td>LAHCIEC</td><td>Reserved</td></tr>
<tr><td>R/W Attribute</td><td>R0, W0</td><td>R0,W</td><td>R0,W</td><td>R0,W</td><td>R0,W</td><td>R0,W</td><td>R0, W0</td><td>R0,W</td><td>R0,W0</td></tr>
<tr><td>Protection Attribute</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>Initial Value</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>00000000</td></tr>
</table></td></tr>
<tr><td>2672</td><td>CHAPTER 39: LIN Interface (V2.1) 9. Interface (V2.1) Registers 9.32. LIN Assist Mode Interrupt Enable Clear Register (LAMIERC)</td><td>Error)
[bit7:0] Reserved: Reserved Bits

Correct)
[bit7:0] Reserved: Reserved Bits
All "0" are always read from these bits.</td></tr>
<tr><td>2673</td><td>CHAPTER 39: LIN Interface (V2.1) 9. Interface (V2.1) Registers 9.33. LIN Assist Mode Error Status Clear Register (LAMESRC)</td><td>Error)
<table>
<tr><td>Bit</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7 ... 0</td></tr>
<tr><td>Field</td><td>Reserved</td><td>LCSERC</td><td>LPTERC</td><td>LSFERC</td><td>LBSERC</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>R/W Attribute</td><td>R0, W0</td><td>R0,W</td><td>R0,W</td><td>R0,W</td><td>R0,W</td><td>R0, W0</td><td>R0, W0</td><td>R0, W0</td><td></td></tr>
<tr><td>Protection Attribute</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td></td></tr>
<tr><td>Initial Value</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td></td></tr>
</table>
Correct)
<table>
<tr><td>Bit</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7 ... 0</td></tr>
<tr><td>Field</td><td>Reserved</td><td>LCSERC</td><td>LPTERC</td><td>LSFERC</td><td>LBSERC</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>R/W Attribute</td><td>R0, W0</td><td>R0,W</td><td>R0,W</td><td>R0,W</td><td>R0,W</td><td>R0, W0</td><td>R0, W0</td><td>R0, W0</td><td>R0,W0</td></tr>
<tr><td>Protection Attribute</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>Initial Value</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>00000000</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>2673</td><td>CHAPTER 39: LIN Interface (V2.1) 9. Interface (V2.1) Registers 9.33. LIN Assist Mode Error Status Clear Register (LAMESRC)</td><td>Error)
[bit10:0] Reserved Bits

Correct)
[bit10:0] Reserved Bits
All "0" are always read from these bits.
All "0" are always written to these bits.</td></tr>
<tr><td>2684</td><td>CHAPTER 39: LIN Interface (V2.1) 9. Interface (V2.1) Registers 9.42. LIN Assist Mode Control Set Register (LAMCRS)</td><td>Error)
<table>
<tr><td>Bit</td><td>15 ... 8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Field</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>LTDRCLS</td><td>LCSTYPS</td><td>LIDENS</td><td>LAMENS</td></tr>
<tr><td>R/W Attribute</td><td></td><td>R0, W0</td><td>R0, W0</td><td>R0, W0</td><td>R0, W0</td><td>R0,W</td><td>R0,W</td><td>R0,W</td><td>R0,W</td></tr>
<tr><td>Protection Attribute</td><td></td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>Initial Value</td><td></td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
Correct)
<table>
<tr><td>Bit</td><td>15 ... 8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Field</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>LTDRCLS</td><td>LCSTYPS</td><td>LIDENS</td><td>LAMENS</td></tr>
<tr><td>R/W Attribute</td><td>R0,W0</td><td>R0, W0</td><td>R0, W0</td><td>R0, W0</td><td>R0, W0</td><td>R0,W</td><td>R0,W</td><td>R0,W</td><td>R0,W</td></tr>
<tr><td>Protection Attribute</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>Initial Value</td><td>00000000</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table></td></tr>
<tr><td>2685</td><td>CHAPTER 39: LIN Interface (V2.1) 9. Interface (V2.1) Registers 9.43. LIN Assist Mode Interrupt Enable Set Register (LAMIERS)</td><td>Error)
<table>
<tr><td>Bit</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7 ... 0</td></tr>
<tr><td>Field</td><td>Reserved</td><td>LCSERIES</td><td>LPTERIES</td><td>LSFERIES</td><td>LBSERIES</td><td>LCSCIES</td><td>Reserved</td><td>LAHCIES</td><td>Reserved</td></tr>
<tr><td>R/W Attribute</td><td>R0, W0</td><td>R0,W</td><td>R0,W</td><td>R0,W</td><td>R0,W</td><td>R0,W</td><td>R0, W0</td><td>R0,W</td><td></td></tr>
<tr><td>Protection Attribute</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td></td></tr>
<tr><td>Initial Value</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td></td></tr>
</table>
Correct)
<table>
<tr><td>Bit</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7 ... 0</td></tr>
<tr><td>Field</td><td>Reserved</td><td>LCSERIES</td><td>LPTERIES</td><td>LSFERIES</td><td>LBSERIES</td><td>LCSCIES</td><td>Reserved</td><td>LAHCIES</td><td>Reserved</td></tr>
<tr><td>R/W Attribute</td><td>R0, W0</td><td>R0,W</td><td>R0,W</td><td>R0,W</td><td>R0,W</td><td>R0,W</td><td>R0, W0</td><td>R0,W</td><td>R0,W0</td></tr>
<tr><td>Protection Attribute</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>Initial Value</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>00000000</td></tr>
</table></td></tr>
</table>

Page	Section	Change Results
2703	CHAPTER 40:Base Timer 6. Start of DMA Controller (DMAC)	Changed Error) Configure the DMAC before starting it with the base timer. For details on DMAC settings, see the following chapters: "DMA Controller" and "INTERRUPTS." Correct) Configure the DMAC before starting it with the base timer. For details on DMAC settings, see the following chapter: "DMA Controller".
2705	CHAPTER 40:Base Timer 7. Registers of the Base Timer	Changed Error) Tables 7-18, 7-19, 7-20 show the register map of each mode. reload/PMW/PPG/PWC timer, respectively. Correct) Tables 7-18, 7-19, 7-20 show the register map of each mode. reload/PWM/PPG/PWC timer, respectively.

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>2706</td><td>CHAPTER 40:Base Timer 7. Registers of the Base Timer</td><td>
Changed Error)
Table 7-6 Register Map (Channel No.: 0) (12) (24)
<table>
<tr><th rowspan="2">Offset</th><th colspan="4">Register Name/Initial Value</th></tr>
<tr><th>+3</th><th>+2</th><th>+1</th><th>+0</th></tr>
<tr><td>0x0000_0000</td><td colspan="2">Reserved
00000000_00000000</td><td colspan="2">BT00_PCSR/BT00_PCSR/BT00_PRLL
/Reserved
XXXXXXXX_XXXXXXXX</td></tr>
<tr><td>0x0000_0004</td><td colspan="2">Reserved
00000000_00000000</td><td colspan="2">Reserved/BT00_PDUT/ BT00_PRLH/
BT00_DTBF
XXXXXXXX_XXXXXXXX
/00000000_00000000 (BT0_DTBF)</td></tr>
<tr><td>0x0000_0008</td><td colspan="2">Reserved
00000000_00000000</td><td colspan="2">BT00_TMR/ BT00_TMR/
BT00_TMR/Reserved
00000000_00000000
XXXXXXXX_XXXXXXXX*</td></tr>
<tr><td>0x0000_000C</td><td colspan="2">Reserved
00000000_00000000</td><td colspan="2">BT00_TMCR
00000000_00000000</td></tr>
<tr><td>0x0000_0010</td><td colspan="2">Reserved
00000000_00000000</td><td>BT00_TMCR2
00000000</td><td>BT00_STC
00000000</td></tr>
<tr><td>0x0000_0014</td><td colspan="2">Reserved
00000000_00000000</td><td>Reserved
00000000</td><td>BT00_STCC
00000000</td></tr>
<tr><td>0x0000_0018</td><td colspan="2">Reserved
00000000_00000000</td><td>Reserved
00000000</td><td>BT00_STCS
00000000</td></tr>
<tr><td>0x0000_0030</td><td colspan="2">Reserved
11111111_11111111</td><td>Reserved
11111111</td><td>BT_BTSEL01
1111_0000</td></tr>
<tr><td>0x0000_0034</td><td colspan="2">Reserved
11111111_11111111</td><td colspan="2">BT_BTSSSR0/12/24
11111111_11111111</td></tr>
<tr><td>0x0000_0038</td><td colspan="2">Reserved
11111111_11111111</td><td colspan="2">BT_BTTRR0/12/24
11110000_00000000</td></tr>
</table>
Correct)
Table 7-6 Register Map (Channel No.: 0) (12) (24)
<table>
<tr><th rowspan="2">Offset</th><th colspan="4">Register Name/Initial Value</th></tr>
<tr><th>+3</th><th>+2</th><th>+1</th><th>+0</th></tr>
<tr><td>0x0000_0000</td><td colspan="2">Reserved
00000000_00000000</td><td colspan="2">BTxx_PCSR/BTxx_PCSR/BTxx_PRLL
/Reserved
XXXXXXXX_XXXXXXXX</td></tr>
<tr><td>0x0000_0004</td><td colspan="2">Reserved
00000000_00000000</td><td colspan="2">Reserved/BTxx_PDUT/ BTxx_PRLH/ BTxx_DTBF
XXXXXXXX_XXXXXXXX
/00000000_00000000 (BTxx_DTBF)</td></tr>
<tr><td>0x0000_0008</td><td colspan="2">Reserved
00000000_00000000</td><td colspan="2">BTxx_TMR/ BTxx_TMR/ BTxx_TMR/Reserved
00000000_00000000
XXXXXXXX_XXXXXXXX*</td></tr>
<tr><td>0x0000_000C</td><td colspan="2">Reserved
00000000_00000000</td><td colspan="2">BTxx_TMCR
00000000_00000000</td></tr>
<tr><td>0x0000_0010</td><td colspan="2">Reserved
00000000_00000000</td><td>BTxx_TMCR2
00000000</td><td>BTxx_STC
00000000</td></tr>
<tr><td>0x0000_0014</td><td colspan="2">Reserved
00000000_00000000</td><td>Reserved
00000000</td><td>BTxx_STCC
00000000</td></tr>
<tr><td>0x0000_0018</td><td colspan="2">Reserved
00000000_00000000</td><td>Reserved
00000000</td><td>BTxx_STCS
00000000</td></tr>
<tr><td>0x0000_001C</td><td colspan="2">Reserved
00000000_00000000</td><td colspan="2">Reserved/BTxx_PSDR/Reserved/Reserved
00000000_00000000</td></tr>
<tr><td>0x0000_0020</td><td colspan="2">Reserved
00000000_00000000</td><td colspan="2">Reserved/BTxx_ADTR/Reserved/Reserved
00000000_00000000</td></tr>
<tr><td>0x0000_0030</td><td colspan="2">Reserved
11111111_11111111</td><td>Reserved
11111111</td><td>BT_BTSEL01/1213/24
25
1111_0000</td></tr>
<tr><td>0x0000_0034</td><td colspan="2">Reserved
11111111_11111111</td><td colspan="2">BT_BTSSSR0/12/24
11111111_11111111</td></tr>
<tr><td>0x0000_0038</td><td colspan="2">Reserved
11111111_11111111</td><td colspan="2">BT_BTTRR0/12/24
11110000_00000000</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>2708</td><td>CHAPTER 40:Base Timer 7. Registers of the Base Timer</td><td>Changed Error)
Table 7-8 Register Map (Channel No.: 2, 4, 6, 8, and 10) (14, 16, 18, 20, and 22) (26, 28, 30, 32, and 34)
<table>
<tr><td>0x0000_0018</td><td>Reserved
00000000_00000000</td><td>Reserved
00000000</td><td>BTxx_STCS
00000000</td></tr>
<tr><td>0x0000_0030</td><td>Reserved
11111111_11111111</td><td>Reserved
11111111</td><td>BT_BTSEL23/45
/67/89/1011
1111_0000</td></tr>
</table>
Correct)
Table 7-8 Register Map (Channel No.: 2, 4, 6, 8, and 10) (14, 16, 18, 20, and 22) (26, 28, 30, 32, and 34)
<table>
<tr><td>0x0000_0018</td><td>Reserved
00000000_00000000</td><td>Reserved
00000000</td><td>BTxx_STCS
00000000</td></tr>
<tr><td>0x0000_001C</td><td>Reserved
00000000_00000000</td><td colspan="2">Reserved/BTxx_PSDR/Reserved/Reserved
00000000_00000000</td></tr>
<tr><td>0x0000_0020</td><td>Reserved
00000000_00000000</td><td colspan="2">Reserved/BTxx_ADTR/Reserved/Reserved
00000000_00000000</td></tr>
<tr><td>0x0000_0030</td><td>Reserved
11111111_11111111</td><td>Reserved
11111111</td><td>BT_BTSEL23/45/67/89/1011/
1415/1617/1819/
2021/2223/2627/2829/
3031/3233/3435
1111_0000</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>2802</td><td>CHAPTER 41:Base Timer I/O Selection Function 4. Registers</td><td>
Added Error)
Table 4-1 List of Base Timer I/O Selection Registers
<table>
<tr><td>BT_BTSEL1011</td><td>I/O selection register (channel 10, 11)</td><td>4.1</td></tr>
<tr><td>BT_BTSSSR0</td><td>Simultaneous soft start register (channel 0)</td><td>4.2</td></tr>
</table>
Correct)
Table 4-1 List of Base Timer I/O Selection Registers
<table>
<tr><td>BT_BTSEL1011</td><td>I/O selection register (channel 10, 11)</td><td>4.1</td></tr>
<tr><td>BT_BTSEL1213</td><td>I/O selection register (channel 12, 13)</td><td>4.1</td></tr>
<tr><td>BT_BTSEL1415</td><td>I/O selection register (channel 14, 15)</td><td>4.1</td></tr>
<tr><td>BT_BTSEL1617</td><td>I/O selection register (channel 16, 17)</td><td>4.1</td></tr>
<tr><td>BT_BTSEL1819</td><td>I/O selection register (channel 18, 19)</td><td>4.1</td></tr>
<tr><td>BT_BTSEL2021</td><td>I/O selection register (channel 20, 21)</td><td>4.1</td></tr>
<tr><td>BT_BTSEL2223</td><td>I/O selection register (channel 22, 23)</td><td>4.1</td></tr>
<tr><td>BT_BTSEL2425</td><td>I/O selection register (channel 24, 25)</td><td>4.1</td></tr>
<tr><td>BT_BTSEL2627</td><td>I/O selection register (channel 26, 27)</td><td>4.1</td></tr>
<tr><td>BT_BTSEL2829</td><td>I/O selection register (channel 28, 29)</td><td>4.1</td></tr>
<tr><td>BT_BTSEL3031</td><td>I/O selection register (channel 30, 31)</td><td>4.1</td></tr>
<tr><td>BT_BTSEL3233</td><td>I/O selection register (channel 32, 33)</td><td>4.1</td></tr>
<tr><td>BT_BTSEL3435</td><td>I/O selection register (channel 34, 35)</td><td>4.1</td></tr>
<tr><td>BT_BTSSSR0</td><td>Simultaneous soft start register (channel 0)</td><td>4.2</td></tr>
</table>
</td></tr>
<tr><td>2802</td><td>CHAPTER 41:Base Timer I/O Selection Function 4. Registers</td><td>
Changed Error)
Tables 4.7, 4.8, and 4.9 show the register map of each mode. The "***/***/***/***" expression in each table corresponds to the reload/PMW/PPG/PWC timer, respectively

Correct)
Tables 4.2, 4.3, and 4.4 show the register map of each mode. The "***/***/***/***" expression in each table corresponds to the reload/PWM/PPG/PWC timer, respectively
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>2803</td><td>CHAPTER 41:Base Timer I/O Selection Function 4. Registers</td><td>Changed
Error)
Table 4-2 Register Map (Channel No.: 0) (12) (24)
<table>
<tr><td>0x0000_0004</td><td>Reserved
00000000_00000000</td><td>Reserved/BTxx_PDUT/ BTxx_PRLH/ BTxx_DTBF
XXXXXXXX_XXXXXXXX
/00000000_00000000 (BT0_DTBF)</td></tr>
</table>
Correct)
Table 4-2 Register Map (Channel No.: 0) (12) (24)
<table>
<tr><td>0x0000_0004</td><td>Reserved
00000000_00000000</td><td>Reserved/BTxx_PDUT/ BTxx_PRLH/ BTxx_DTBF
XXXXXXXX_XXXXXXXX
/00000000_00000000 (BTxx_DTBF)</td></tr>
</table></td></tr>
<tr><td>2803</td><td>CHAPTER 41:Base Timer I/O Selection Function 4. Registers</td><td>Changed
Error)
Table 4-2 Register Map (Channel No.: 0) (12) (24)
<table>
<tr><td>0x0000_0018</td><td>Reserved
00000000_00000000</td><td>Reserved
00000000</td><td>BTxx_STCS
00000000</td></tr>
<tr><td>0x0000_0030</td><td>Reserved
11111111_11111111</td><td>Reserved
11111111</td><td>BT_BTSEL01
1111_0000</td></tr>
<tr><td>0x0000_0034</td><td>Reserved
11111111_11111111</td><td colspan="2">BT_BTSSSR0/12/24
11111111_11111111</td></tr>
</table>
Correct)
Table 4-2 Register Map (Channel No.: 0) (12) (24)
<table>
<tr><td>0x0000_0018</td><td>Reserved
00000000_00000000</td><td>Reserved
00000000</td><td>BTxx_STCS
00000000</td></tr>
<tr><td>0x0000_001C</td><td>Reserved
00000000_00000000</td><td colspan="2">Reserved/BTxx_PSDR/Reserved/Reserved
00000000_00000000</td></tr>
<tr><td>0x0000_0020</td><td>Reserved
00000000_00000000</td><td colspan="2">Reserved/BTxx_ADTR/Reserved/Reserved
00000000_00000000</td></tr>
<tr><td>0x0000_0030</td><td>Reserved
11111111_11111111</td><td>Reserved
11111111</td><td>BT_BTSEL01/1213/2425
1111_0000</td></tr>
<tr><td>0x0000_0034</td><td>Reserved
11111111_11111111</td><td colspan="2">BT_BTSSSR0/12/24
11111111_11111111</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>2805</td><td>CHAPTER 41:Base Timer I/O Selection Function 4. Registers</td><td>Changed
Error)
Table 4-4 Register Map (Channel No.: 2, 4, 6, 8, and 10) (14, 16, 18, 20, and 22) (26, 28, 30, 32, and 34)
<table>
<tr><td>0x0000_0018</td><td>Reserved
00000000_00000000</td><td>Reserved
00000000</td><td>BTxx_STCS
00000000</td></tr>
<tr><td>0x0000_0030</td><td>Reserved
11111111_11111111</td><td>Reserved
11111111</td><td>BT_BTSEL23/45
/67/89/1011
1111_0000</td></tr>
</table>
Correct)
Table 4-4 Register Map (Channel No.: 2, 4, 6, 8, and 10) (14, 16, 18, 20, and 22) (26, 28, 30, 32, and 34)
<table>
<tr><td>0x0000_0018</td><td>Reserved
00000000_00000000</td><td>Reserved
00000000</td><td>BTxx_STCS
00000000</td></tr>
<tr><td>0x0000_001C</td><td>Reserved
00000000_00000000</td><td colspan="2">Reserved/BTxx_PSDR/Reserved/Reserved
00000000_00000000</td></tr>
<tr><td>0x0000_0020</td><td>Reserved
00000000_00000000</td><td colspan="2">Reserved/BTxx_ADTR/Reserved/Reserved
00000000_00000000</td></tr>
<tr><td>0x0000_0030</td><td>Reserved
11111111_11111111</td><td>Reserved
11111111</td><td>BT_BTSEL23/45/67/89/
1011/1415/1617/1819/
2021/2223/2627/2829/
3031/3233/3435
1111_0000</td></tr>
</table></td></tr>
<tr><td>2806</td><td>CHAPTER 41:Base Timer I/O Selection Function 4. Registers 4.1. I/O Selection Registers (BT_BTSELmn)</td><td>Changed
Error)
4.1. I/O Selection Registers (BT_BTSEL01, BT_BTSEL23, BT_BTSEL45, BT_BTSEL67, BT_BTSEL_89, BT_BTSEL1011)
Correct)
4.1. I/O Selection Registers (BT_BTSELmn)</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>2806</td><td>CHAPTER 41:Base Timer I/O Selection Function 4. Registers 4.1. I/O Selection Registers (BT_BTSELmn)</td><td>Changed Error)
<table>
<tr><td>Bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Field</td><td colspan="4">Reserved</td><td colspan="4">BTSEL01</td></tr>
<tr><td>R/W Attribute</td><td colspan="4">R1,WX</td><td colspan="4">R/W</td></tr>
<tr><td>Protection Attribute</td><td colspan="8">-</td></tr>
<tr><td>Initial Value</td><td colspan="4">1111</td><td colspan="4">0000</td></tr>
</table>
Correct)
<table>
<tr><td>Bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Field</td><td colspan="4">Reserved</td><td colspan="4">BTSEL01</td></tr>
<tr><td>R/W Attribute</td><td colspan="4">R1,WX</td><td>R/W0</td><td colspan="3">R/W</td></tr>
<tr><td>Protection Attribute</td><td colspan="8">-</td></tr>
<tr><td>Initial Value</td><td colspan="4">1111</td><td colspan="4">0000</td></tr>
</table></td></tr>
<tr><td>2806</td><td>CHAPTER 41:Base Timer I/O Selection Function 4. Registers 4.1. I/O Selection Registers (BT_BTSELmn)</td><td>Changed Error)

[bit3:0] BTSEL01: I/O Mode Selection Bits

Correct)

[bit3:0] BTSELmn: I/O Mode Selection Bits</td></tr>
</table>

Page	Section	Change Results
2816	CHAPTER 42:32-bit Free-Run Timer 2. Explanation of the 32-Bit Free-Run Timer Operation	Error) ***Notes:*** – *The 0 detection interrupt is not generated when the timer is cleared (SCLR: bit4=1 of the timer state control register (TCCS)).* Correct) ***Notes:*** – *The 0 detection interrupt is not generated when the timer is cleared (SCLR: bit4=1 of the timer state control register (TCCS)).* – *If Interrupt mask function "by setting TCCS.MSI[2:0] Register" is used, please write TCCS[15:8], TCCSC[15:8] and TCCSS[15:8] after 32bit free-run timer stopped.*
2830	CHAPTER 42:32-bit Free-Run Timer 3. Registers of the 32-Bit Free-Run Timer	Error) ***Notes:*** The value read is a mask counter value. The mask counter is a decrement counter. – *The written data is written to the mask register.* – *While the free-run timer is operating (STOP: bit6=0 of the timer enable bit), the value written to the mask register is reloaded to the counter only when the mask counter becomes 0.* – *When the free-run timer is stopped (STOP: bit6=1 of the timer enable bit), the value written to the mask register is immediately reloaded to the mask counter.* Correct) ***Notes:*** The value read is a mask counter value. The mask counter is a decrement counter. – *The written data is written to the mask register.* – *If Interrupt mask function "by setting TCCS.MSI[2:0] Register" is used, please write TCCS[15:8], TCCSC[15:8] and TCCSS[15:8] after 32bit free-run timer stopped.* –

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>2831</td><td>CHAPTER 42:32-bit Free-Run Timer
3. Registers of the 32-Bit Free-Run Timer</td><td>Error)
<table><tr><th>Bit</th><th>Description</th></tr><tr><td>0</td><td>0 is not detected.</td></tr><tr><td>1</td><td>0 is detected.</td></tr></table>
Correct)
<table><tr><th>Bit</th><th>Description</th></tr><tr><td>0</td><td>No compare clear match.</td></tr><tr><td>1</td><td>Compare clear match.</td></tr></table></td></tr>
<tr><td>2837</td><td>CHAPTER 42:32-bit Free-Run Timer
3. Registers of the 32-Bit Free-Run Timer</td><td>Error)
The timer state control clear register (TCCSC) is a register that is used to clear bits of the timer state control register (TCCS).

Correct)
The timer state control clear register (TCCSC) is a register that is used to clear bits of the timer state control register (TCCS).
For details on writing to this register, see "4. Precautions for Using This Device"</td></tr>
<tr><td>2840</td><td>CHAPTER 42:32-bit Free-Run Timer
3. Registers of the 32-Bit Free-Run Timer</td><td>Error)
The timer state control set register (TCCSC) is a register that is used to set bits of the timer state control register (TCCS).

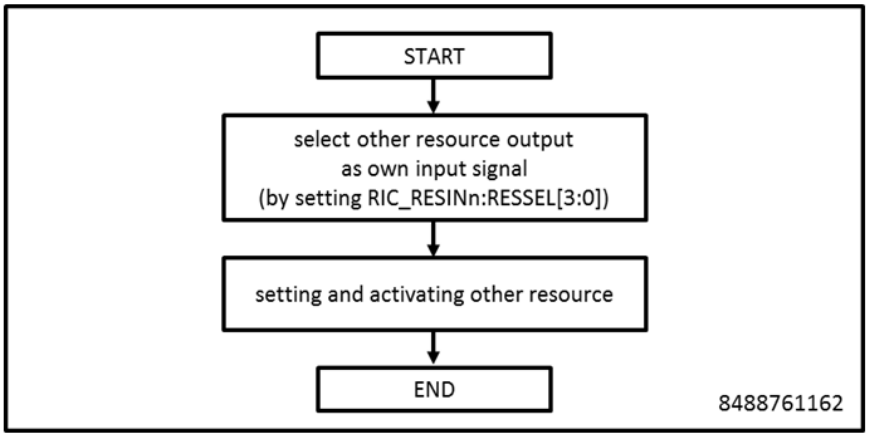
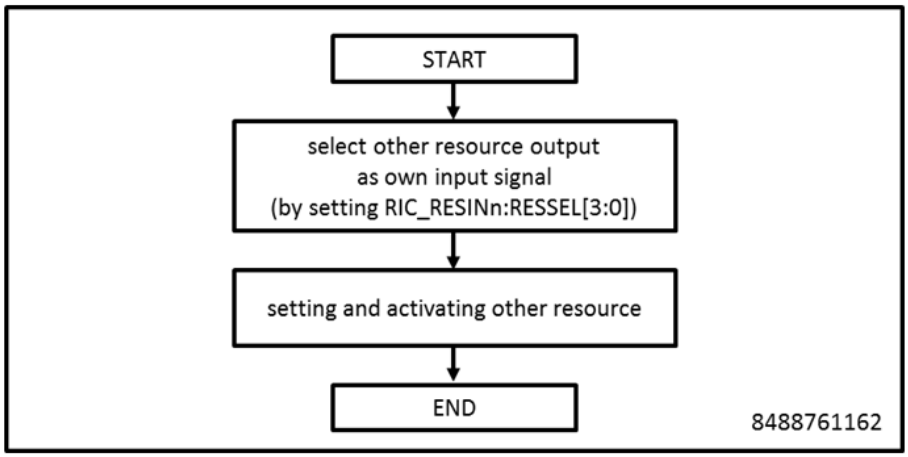
Correct)
The timer state control set register (TCCSC) is a register that is used to set bits of the timer state control register (TCCS).
For details on writing to this register, see "4. Precautions for Using This Device"</td></tr>
</table>

Page	Section	Change Results
2843		Error) ■Data can be written directly to this register only when writing to all bits. ■In the normal reading mode, the interrupt mask counter value is read from MSI2 to MSI0. Correct) ■Data can be written directly to this register only when writing to all bits. ■In the normal reading mode, the interrupt mask counter value is read from Interrupt Mask Selection bits(MSI[2:0]) in the Timer State Control Register (TCCS). ■If Interrupt mask function "by setting TCCS.MSI[2:0] Register" is used, please write TCCS[15:8], TCCSC[15:8] and TCCSS[15:8] after 32bit free-run timer stopped.
2904	CHAPTER 44: 32-bit Output Compare 5. Registers 5.4. Compare Control Set Register (OCSS)	Error) ■When "1" is written, the buffer transfer selection bit (CH.1) (BTS1) in the compare control register (OCS) is set to "1". ■"0" is always read from the BST1S bit. Correct) ■When "1" is written, the buffer transfer selection bit (CH.1) (BTS1) in the compare control register (OCS) is set to "1". ■"0" is always read from the BTS1S bit.
2932	CHAPTER 46:Reload Timer Simultaneous Soft Start 3. Explanation of Operation	Error) The RLT_TSEL register enables selections of the external trigger and software trigger for each channel. Correct) The RLTx_TSEL(x=0, 1, 2) register enables selections of the external trigger and software trigger for each channel.
2932	CHAPTER 46:Reload Timer Simultaneous Soft Start 3. Explanation of Operation	Error) A write access to the RLT_SSSR register can activate a software trigger to start multiple reload timers simultaneously. Correct) A write access to the RLTx_SSSR(x=0, 1, 2) register can activate a software trigger to start multiple reload timers simultaneously.

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>2949</td><td>CHAPTER 47:QPRC (Quadrature Position/Revolution Counter) 3. Operations 3.4. QPRC(Quadrature Position/Revolution Counter) Interrupts</td><td>
Error)
Table 3-6 Generation Conditions of QPRC(Quadrature Position/Revolution Counter) Interrupt Requests
<table>
<tr><th>Interrupt Request</th><th>Interrupt Request Flag</th><th>Interrupt Request is Enabled If</th><th>Interrupt Request is Cleared If</th></tr>
<tr><td>Count inversion interrupt request</td><td>QICR.CDCF= 1</td><td>QICR.CDCIE= 1</td><td>"0" is written to QICR.CDCF</td></tr>
<tr><td>Zero index interrupt request</td><td>QICR.ZIIF= 1</td><td rowspan="3">QICR.OUZIE= 1</td><td>"0" is written to QICR.ZIIF</td></tr>
<tr><td>Overflow interrupt request</td><td>QICR.OFDF= 1</td><td>"0" is written to QICR.OFDF</td></tr>
<tr><td>Underflow interrupt request</td><td>QICR.UFDF= 1</td><td>"0" is written to QICR.UFDF</td></tr>
<tr><td>PC and RC match interrupt request</td><td>QICR.QPRCMF= 1</td><td>QICR.QPRCMIE= 1</td><td>"0" is written to QICR.QPRCMF</td></tr>
<tr><td>PC match interrupt request</td><td>QICR.QPCMF= 1</td><td>QICR.QPCMIE= 1</td><td>"0" is written to QICR.QPCMF</td></tr>
<tr><td>PC match and RC match interrupt request</td><td>QICR.QPCNRCMF= 1</td><td>QICR.QPCNRCMIE= 1</td><td>"0" is written to QICR.QPCNRCMF</td></tr>
<tr><td>Outrange interrupt request</td><td>QECR.ORNGF= 1</td><td>QICR.ORNGIE= 1</td><td>"0" is written to QECR.QRNGF</td></tr>
</table>
Correct)
Table 3-6 Generation Conditions of QPRC(Quadrature Position/Revolution Counter) Interrupt Requests
<table>
<tr><th>Interrupt Request</th><th>Interrupt Request Flag</th><th>Interrupt Request is Enabled If</th><th>Interrupt Request is Cleared If</th></tr>
<tr><td>Count inversion interrupt request</td><td>QICR.CDCF= 1</td><td>QICR.CDCIE= 1</td><td>"0" is written to QICR.CDCF</td></tr>
<tr><td>Zero index interrupt request</td><td>QICR.ZIIF= 1</td><td rowspan="3">QICR.OUZIE= 1</td><td>"0" is written to QICR.ZIIF</td></tr>
<tr><td>Overflow interrupt request</td><td>QICR.OFDF= 1</td><td>"0" is written to QICR.OFDF</td></tr>
<tr><td>Underflow interrupt request</td><td>QICR.UFDF= 1</td><td>"0" is written to QICR.UFDF</td></tr>
<tr><td>PC and RC match interrupt request</td><td>QICR.QPRCMF= 1</td><td>QICR.QPRCMIE= 1</td><td>"0" is written to QICR.QPRCMF</td></tr>
<tr><td>PC match interrupt request</td><td>QICR.QPCMF= 1</td><td>QICR.QPCMIE= 1</td><td>"0" is written to QICR.QPCMF</td></tr>
<tr><td>PC match and RC match interrupt request</td><td>QICR.QPCNRCMF= 1</td><td>QICR.QPCNRCMIE= 1</td><td>"0" is written to QICR.QPCNRCMF</td></tr>
<tr><td>Outrange interrupt request</td><td>QECR.ORNGF= 1</td><td>QICR.ORNGIE= 1</td><td>"0" is written to QECR.ORNGF</td></tr>
</table>
</td></tr>
</table>

Page	Section	Change Results
2961	CHAPTER 47:QPRC (Quadrature Position/Revolution Counter) 4. Registers 4.5. QPRC Control Register (QCR)	Error) When the counter clear function is enabled(QGSC= 0), the ZIN pin clears the position counter if the revolution count mode is set to RC_Mode0(RCM[1:0]= 0b00), RC_Mode1(RCM[1:0]= 0b01), or RC_Mode3(RCM[1:0]= 0b11). The CGE1 and CGE0 bits of the QCR register clear the position counter by selecting a valid edge of the ZIN pin and detecting the selected edge. Correct) When the counter clear function is enabled(QGSC= 0), the ZIN pin clears the position counter if the revolution count mode is set to RC_Mode0(RCM[1:0]= 0b00), RC_Mode1(RCM[1:0]= 0b01), or RC_Mode3(RCM[1:0]= 0b11). The CGE[1] and CGE[0] bits of the QCR register clear the position counter by selecting a valid edge of the ZIN pin and detecting the selected edge.
2961	CHAPTER 47:QPRC (Quadrature Position/Revolution Counter) 4. Registers 4.5. QPRC Control Register (QCR)	Error) When the gate function is enabled(QGSC= 1), the ZIN pin controls the count operation of the position counter. The CGE1 and CGE0 bits of the QCR register count the position counter at the valid level of the ZIN pin. Correct) When the gate function is enabled(QGSC= 1), the ZIN pin controls the count operation of the position counter. The CGE[1] and CGE[0] bits of the QCR register count the position counter at the valid level of the ZIN pin.
2961	CHAPTER 47:QPRC (Quadrature Position/Revolution Counter) 4. Registers 4.5. QPRC Control Register (QCR)	Error) These bits are used to select the count mode of the revolution counter and the reset mode of the position counter. For the effect on the position counter, see "Operation of Revolution Counter 3.2". Correct) These bits are used to select the count mode of the revolution counter and the reset mode of the position counter. For the effect on the position counter, see"3.2 Operation of Revolution Counter".

Page	Section	Change Results
2912	CHAPTER 45:32-bit Reload Timer 3. Operation of the 32-Bit Reload Timer	Error) 3.3 Output Functions of 32-Bit Reload Timer 3.4 Counter Operation 3.5 DMA Operation Correct) 3.3 Output Functions of 32-Bit Reload Timer 3.4 Counter Operation State 3.5 DMA Operation
2924	CHAPTER 45:32-bit Reload Timer 4. Registers 4.2. Timer Control Status Register (RLTn_TMCSR)	Error) ***Notes:*** – *When DBGE is set to "1" and the processor is in debug state, the timer counter operation is paused, and writing to the RLTn_TMRLR register directly updates the timer counter (RLTn_TMR register). When the processor leaves debug state or DBGE is set to "0", the timer counter operation is resumed.* – *This bit is enable at* ✧ *from ch. 0 to ch. 3 of CPER#2* ✧ *from ch.0 to ch. 1 of MCU_CONFIG_GROUP.* Correct) ***Notes:*** – *When DBGE is set to "1" and the processor is in debug state, the timer counter operation is paused, and writing to the RLTn_TMRLR register directly updates the timer counter (RLTn_TMR register). When the processor leaves debug state or DBGE is set to "0", the timer counter operation is resumed.* – *This bit is enable at* ✧ *from ch. 0 to ch. 3 of CPERI#2* ✧ *from ch.0 to ch. 1 of MCU_CONFIG_GROUP.*

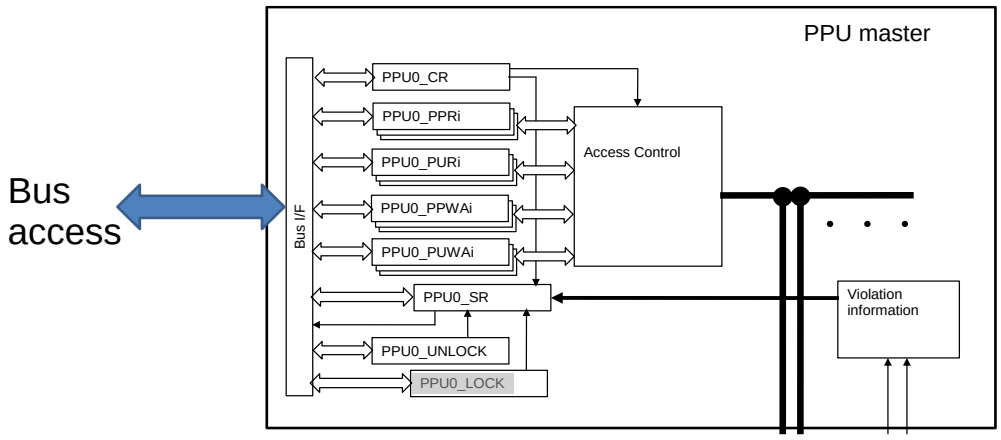
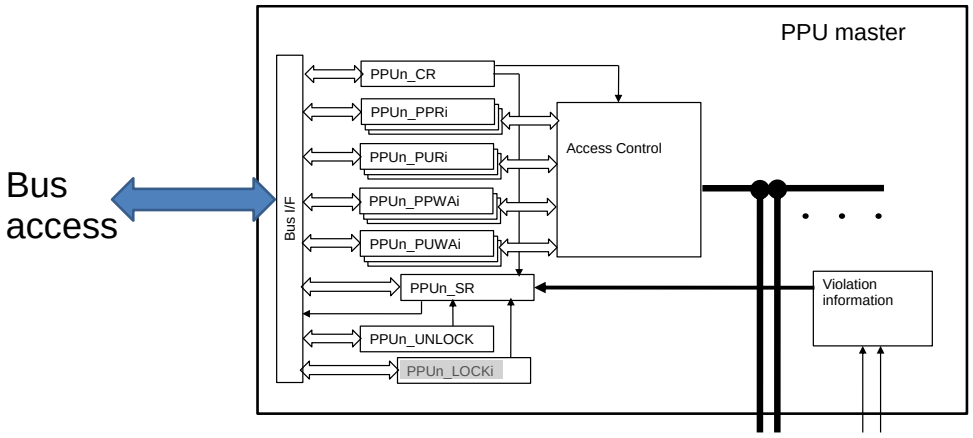
Page	Section	Change Results
3017	CHAPTER 50:I/O Port 3. Setting Procedure Examples	Error) **Resource Input Selection (Selecting Other Resource Output)** **Figure 3-7 Setting Procedure**  ***Note:*** – *For details on resource input selection, see "RESOURCE INPUT SELECTION of I/O PORT VARIOUS SETTINGS "* Correct) **Resource Input Selection (Selecting Other Resource Output)** **Figure 3-7 Setting Procedure**  ***Note:*** – *For details on resource input selection, see product specification.*

Page	Section	Change Results
3036	CHAPTER 50:I/O Port 4. Register List 4.9. GPIO Key Code Register (GPIO_KEYCDR)	Error) – *There is no effect on the key code protection release process even if registers other than this register and the applicable key code register that is in the release processing (the register set by the RADA) are accessed while writing the key code 0b00, 0b01, 0b10, and 0b11.* Correct) – *There is no effect on the key code protection release process even if registers other than this register and the applicable key code register that is in the release processing (the register set by the RADR) are accessed while writing the key code 0b00, 0b01, 0b10, and 0b11.*

Page	Section	Change Results
3037 ~ 3039	CHAPTER 50:I/O Port 4. Register List 4.10. Port Setting Register (PPC_PCFGRijj) (i = 0 to (product specification), jj = 00 to 31)	Error) **[bit14] POD: Port Output Data Bit** This bit indicates the value outputted to a pin. This bit is enabled only when the port output enable bit (POE) is "1". **[bit13] PID: Port Input Data Bit** This bit indicates the value inputted to the pin selected by the input level bit (PIL[1:0]). This bit is indefinite when the global input enable bit (GPIO_GPORTEN) is "0". **[bit11:10] PIL: Input Level Bit** For details, see " INPUT LEVEL SETTING of I/O PORT VARIOUS SETTINGS." **[bit9] PUE: Pull Up Enable Bit** This bit sets whether pull-up is performed or not in input status. **[bit8] PDE: Pull Down Enable Bit** This bit sets whether pull-down is performed in input status. **[bit7:6] ODR: Port Output Drive Selection Bit** For details, see " OUTPUT DRIVE CAPACITY SETTING of I/O PORT VARIOUS SETTINGS." **[bit2:0] POF: Port Output Function Selection Bit** For details, see "OUTPUT RESOURCE SELECTION of I/O PORT VARIOUS SETTINGS." Correct) **[bit14] POD: Port Output Data Bit** This bit indicates the value outputted to a pin. **[bit13] PID: Port Input Data Bit** This bit indicates the value inputted to the pin selected by the input level bit (PIL[1:0]). This bit is indefinite when the global input enable bit (GPORTEN) is "0". **[bit11:10] PIL: Input Level Bit** For details, see product specification. **[bit9] PUE: Pull Up Enable Bit** This bit set to enable pull-up resistor, when the port is configured as input. **[bit8] PDE: Pull Down Enable Bit** This bit set to enable pull-down resistor, when the port is configured as input. **[bit7:6] ODR: Port Output Drive Selection Bit** For details, see product specification. **[bit2:0] POF: Port Output Function Selection Bit** For details, see product specification.

Page	Section	Change Results
3041	CHAPTER 50:I/O Port 4. Register List 4.11. PPC Key Code Register (PPC_KEYCDR)	Error) – *There is no effect on the key code protection release process even if registers other than this register and the applicable key code register that is in the release processing (the register set by the RADA) are accessed while writing the key code 0b00, 0b01, 0b10, and 0b11.* Correct) – *There is no effect on the key code protection release process even if registers other than this register and the applicable key code register that is in the release processing (the register set by the RADR) are accessed while writing the key code 0b00, 0b01, 0b10, and 0b11.*
3042 3043	CHAPTER 50:I/O Port 4. Register List 4.12. Resource Input Setting Register (RIC_RESINn) (n = 0 to (product specification))	Error) **[bit11:8] PORTSEL: Resource Selection Bit** For details, see "RESOURCE INPUT SELECTION of I/O PORT VARIOUS SETTINGS." **[bit3:0] RESSEL: Resource Selection Bit** For assignment of source, see "RESOURCE INPUT SELECTION of I/O PORT VARIOUS SETTINGS." Correct) **[bit11:8] PORTSEL: Resource Selection Bit** For details, see product specification. **[bit3:0] RESSEL: Resource Selection Bit** For assignment of source, see product specification.

Page	Section	Change Results
3045	CHAPTER 50:I/O Port 4. Register List 4.13. RIC Key Code Register (RIC_KEYCDR)	Error) – *There is no effect on the key code protection release process even if registers other than this register and the applicable key code register that is in the release processing (the register set by the RADA) are accessed while writing the key code 0b00, 0b01, 0b10, and 0b11.* Correct) – *There is no effect on the key code protection release process even if registers other than this register and the applicable key code register that is in the release processing (the register set by the RADR) are accessed while writing the key code 0b00, 0b01, 0b10, and 0b11.*

Page	Section	Change Results
3049	CHAPTER 51:Peripheral Protection Unit 2. Configuration and Block Diagram	Changed Error) **Figure 2-1 Block Diagram**  Correct) **Figure 2-1 Block Diagram** 

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>3059</td><td>CHAPTER 51:Peripheral Protection Unit 4. Registers</td><td>Changed
Error)

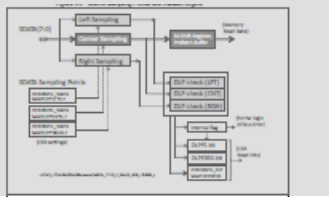
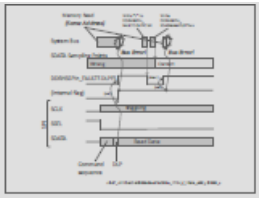
Table 4-1 Register Map
<table><tr><td>0x280

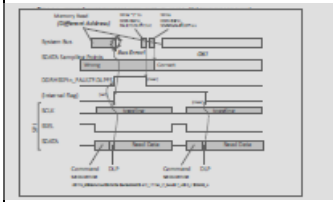
0x2F4</td><td>PPUn_PFEN0 - PPUn_PFEN29
00000000_00000000_00000000_00000000</td></tr><tr><td>0x2F8</td><td>PPUn_LOCK4
00000000_00000000_00000000_00000000</td></tr></table>
Correct)

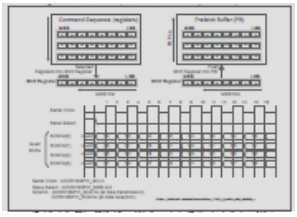
Table 4-1 Register Map
<table><tr><td>0x280

0x2F4</td><td>PPU0_PFEN0 - PPU0_PFEN29
00000000_00000000_00000000_00000000</td></tr><tr><td>0x2F8</td><td>PPU0_LOCK4
00000000_00000000_00000000_00000000</td></tr></table></td></tr>
<tr><td>3075</td><td>CHAPTER 51:Peripheral Protection Unit 4. Registers 4.13. PPU Privilege Mode Forced Change Function Enable Register (PPU0_PFENi)</td><td>Changed
Error)
<table><tr><td>BIT_OFFSET</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr><tr><td>BIT_NAME</td><td>PFEN[15]</td><td>PFEN[14]</td><td>PFEN[13]</td><td>PFEN[12]</td><td>PFEN11]</td><td>PFEN[10]</td><td>PFEN[9]</td><td>PFEN[8]</td></tr><tr><td>ACCESS_TYPE</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td></tr><tr><td>PROT_TYPE</td><td colspan="8">WPS</td></tr><tr><td>INITIAL_VALUE</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table>
Correct)
<table><tr><td>BIT_OFFSET</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr><tr><td>BIT_NAME</td><td>PFEN[15]</td><td>PFEN[14]</td><td>PFEN[13]</td><td>PFEN[12]</td><td>PFEN[11]</td><td>PFEN[10]</td><td>PFEN[9]</td><td>PFEN[8]</td></tr><tr><td>ACCESS_TYPE</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td></tr><tr><td>PROT_TYPE</td><td colspan="8">WPS</td></tr><tr><td>INITIAL_VALUE</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table></td></tr>
</table>

Page	Section	Change Results
3081	CHAPTER 52: DDR High Speed SPI Controller 1. Overview	Error) **Table 1-1 Terms and Abbreviations** Single Data Rate Correct) **Table 1-1 Terms and Abbreviations** Standard Data Rate
3089	CHAPTER 52: DDR High Speed SPI Controller 3. Serial Interface 3.3. SPI Input Data Sampling Point	Added the below: Any of DLP errors at CNT/LFT/RGH will cause an error response on the System Bus. For more details, please refer to Notes of these registers before using DLP.
3090	CHAPTER 52: DDR High Speed SPI Controller 3. Serial Interface 3.3. SPI Input Data Sampling Point	Error) **Figure 3-4SDATA Sampling Clock Settings** DDRHSSPIn_SDATASAMPLEPTLFT0-7. SDATASMPTLFT[4:0] DDRHSSPIn_SDATASAMPLEPTCNT0-7. SDATASMPTCNT[4:0] DDRHSSPIn_SDATASAMPLEPTRGH0-7. SDATASMPTRGH[4:0] Correct) **Figure 3-4SDATA Sampling Clock Settings** DDRHSSPIn_SDATASAMPLEPTLFT0-7. SDATASMPTLFT[6:0] DDRHSSPIn_SDATASAMPLEPTCNT0-7. SDATASMPTCNT[6:0] DDRHSSPIn_SDATASAMPLEPTRGH0-7. SDATASMPTRGH[6:0]

Page	Section	Change Results
3091, 3092	CHAPTER 52: DDR High Speed SPI Controller 3. Serial Interface 3.4. SPI Data Learning Pattern	Added the below: The logic related to SDATA Sampling Points and DLP is illustrated as below. The internal flag is directly invisible from the CPU, however, its status is reflected to the Bus Error and the system can detect it. **Figure 3-5SDATA Sampling Points and Related Logics**  Notes: − Once having detected a DLP error, the following Memory Read access is affected as below. (1)When a DLP error occurs, it is held on the internal flag along DDRHSSPIn_FAULTF.DLPFS bit. (2)The software clears DLP-related flags (DDRHSSPIn_FAULTF.DLPFS and DDRHSSPIn_RXF.DLPERR) and adjusts the SDATA Sampling Points to proper values. (3)This internal flag affects the next Memory Read access depending on the address. *(a)Same word-line addresses: "Error"If ("new address" >> 2) = ("previous address" >> 2), the result is as below.-Bus Error on the System Bus-Internal flag = "1"-DDRHSSPIn_FAULTF.DLPFS = "1"-DDRHSSPIn_RXF.DLPERR = "0"-DDRHSSPIn_DLPSAMPLESTATUS = 0x00000000(b)Different word-line addresses: "Success"If ("new address" >> 2) ≠("previous address" >> 2), the result is as below.-No Bus Error on the System Bus-Internal flag = "0"-DDRHSSPIn_FAULTF.DLPFS = "0"-DDRHSSPIn_RXF.DLPERR = "0"-DDRHSSPIn_DLPSAMPLESTATUS 0x00000000Figure 3-6and Figure 3-7illustrate these cases (a) and (b). Each diagram depicts two consecutive Memory Read accesses -the first one failing to sample SDATA, and the second one with successful sampling. The address of the second access determines the result.* *− The proposed Calibration sequence is described in the section 4.4.4 "Using the DDRHSSPI in Command Sequencer Mode of Operation".* **Figure 3-6Memory Read accesses from "FAIL" to "PASS" ((a) Same Word-line)** 

Page	Section	Change Results
3093	CHAPTER 52: DDR High Speed SPI Controller 3. Serial Interface 3.4. SPI Data Learning Pattern	Added the below: **Figure 3-7Memory Read accesses from "FAIL" to "PASS" ((b) Different Word-line)** 
3094	CHAPTER 52: DDR High Speed SPI Controller 3. Serial Interface 3.6. Shift Direction	Error) Figure 4-7 depicts the direction in which the data in the Shift Register is shifted to/from the serial data lines, under Quad Mode. The waveforms assume DDRHSSPIn_DMFIFOCFG.FWIDTH = 0b00. The Figures depict that the transmit data is loaded into the Shift Register from the TX-FIFO. However, the source of transmit data can be other registers, such as DDRHSSPIn_RDCSDC0-11.RDCSDATA. When DDRHSSPIn_MID.MID is 0x00000100 or 0x00000300, DDRHSSPI is available for Dual Legacy Mode and Dual Quad Mode. Figure 3-6 and Figure 3-7 depict the direction in which the data in the Shift Register is shifted to/from the serial data lines, under Dual Legacy Mode and Dual Quad Mode. Correct) The following figures Figure 4-7 depict the direction in which the data in the Shift Register is shifted to/from the serial data lines. The waveforms assume DDRHSSPIn_DMFIFOCFG.FWIDTH = 0b00 in Direct Mode. The Figures depict that the transmit data is loaded into the Shift Register from the TX-FIFO. However, the source of transmit data can be other registers, such as DDRHSSPIn_RDCSDC0-11.RDCSDATA. When DDRHSSPIn_MID.MID is 0x00000100 or 0x00000300, DDRHSSPI is available for Dual Legacy Mode and Dual Quad Mode. Figure 3-14 and Figure 3-15 depict the direction in which the data in the Shift Register is shifted to/from the serial data lines, under Dual Legacy Mode and Dual Quad Mode.

Page	Section	Change Results
3095, to 3102	CHAPTER 52: DDR High Speed SPI Controller 3. Serial Interface 3.6. Shift Direction	Error) **Figure 3-5 Shift Direction (Legacy Mode and Quad Mode)** **(Assumptions: DDRHSSPIn_DMFIFOCFG.FWIDTH= 0b00)** **Figure 3-6 Shift Direction (Dual Legacy Mode and Dual Quad Mode in Direct Mode)** **(Assumptions: DDRHSSPIn_DMFIFOCFG.FWIDTH= 0b00)** **Figure 3-7 Shift Direction (Dual Quad Mode in Command Sequencer Mode)** **(Assumptions: DDRHSSPIn_DMFIFOCFG.FWIDTH= 0b00)** Correct) **Figure 3-8Quad Mode / byte ordering -Command Sequencer Mode** **Figure 3-9Quad Mode / byte ordering -Command Sequencer Mode (RX Register -In Time)** **Figure 3-10Quad Mode / byte ordering -Direct Mode** **Figure 3-11Quad Mode / byte ordering -Direct Mode (RX Register -In Time)** **Figure 3-12Shift Direction(Legacy Mode and Quad Mode in Direct Mode)(Assumptions: DDRHSSPIn_DMFIFOCFG.FWIDTH=0b00)** **Figure 3-13 Shift Direction (Quad Mode in Command Sequencer Mode)**  (Each line in the PB has 32-bit width, and this diagram is simplified by showing the lower 8 bits.) **Figure 3-14 Shift Direction (Dual Legacy Mode and Dual Quad Mode in Direct Mode)** **(Assumptions: DDRHSSPIn_DMFIFOCFG.FWIDTH= 0b00)** *− The bit alignment is different between TX and RX in Dual Legacy Mode. (Please refer to Table 4-1, and its TXCTRL="1" & TXDATA[12]="1" condition -> SDR mode.)* **Figure 3-15Shift Direction(Dual Quad Mode in Command Sequencer Mode)** (Each line in the PB has 32-bit width, and this diagram is simplified by showing the lower 8 bits.)

Page	Section	Change Results
3105	CHAPTER 52: DDR High Speed SPI Controller 4.Operations of the DDRHSSPI 4.1.Direct Mode 4.1.3. TX-FIFO Control	Error) Table 4-1 Serial Data Output Control (1) DDRHSSPIn_DMTRP.TRP[1:0] ≠0b11 FWIDTH TRP[1:0] SDATA 0b00 0b00 TXDATA[7:0] will be sent on SDATA[0]. 0b10 TXDATA[7:0] will be sent on SDATA[3:0]. 0b01 0b00 TXDATA[15:8] and TXDATA[7:0] will be sequentially sent on SDATA[0]. 0b10 TXDATA[15:8] and TXDATA[7:0] will be sequentially sent on SDATA[3:0]. 0b11 0b00 TXDATA[31:24], TXDATA[23:16], TXDATA[15:8] and TXDATA[7:0] will be sequentially sent on SDATA[0]. 0b10 TXDATA[31:24], TXDATA[23:16], TXDATA[15:8] and TXDATA[7:0] will be sequentially sent on SDATA[3:0]. In TX-and-RX Mode, it supports only Legacy Mode. So, RX-FIFO receives data from SDATA[1]. Correct) Table 4-1 Serial Data Output Control (1) DDRHSSPIn_DMTRP.TRP[1:0]≠"11" FWIDTH TRP[1:0] SDATA 0b00 0b00 TXDATA[7:0] will be sent on SDATA[0]. 0b10 TXDATA[7:0] will be sent on SDATA[3:0]. 0b01 0b00 TXDATA[15:8][7:0] will be sent on SDATA[0]. 0b10 TXDATA[15:8][7:0] will be sent on SDATA[3:0]. (in the order of TXDATA[15:8]->[7:0], along the SCLK edge) 0b11 0b00 TXDATA[31:24][23:16][15:8][7:0] will be sent on SDATA[0]. 0b10 TXDATA[31:24][23:16][15:8][7:0] will be sent on SDATA[3:0]. (in the order of TXDATA[31:24]->[23:16]->[15:8]->[7:0] along the SCLK edge) In TX-and-RX Mode, it supports only Legacy Mode. So, RX-FIFO receives data from SDATA[1].

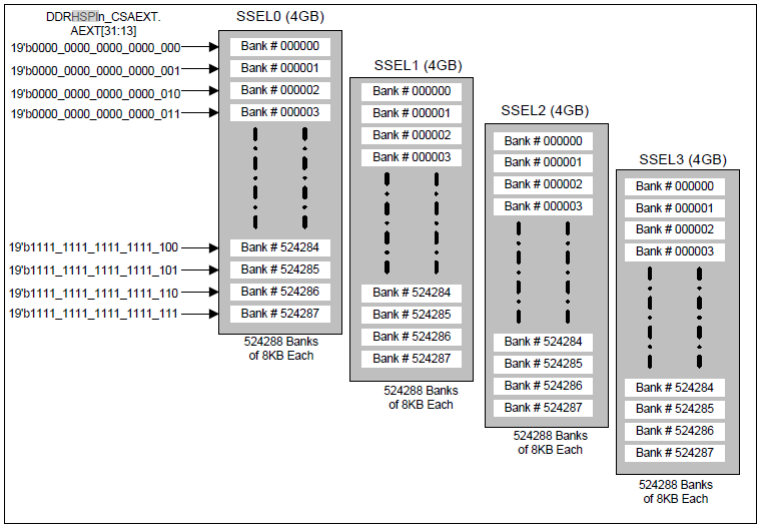
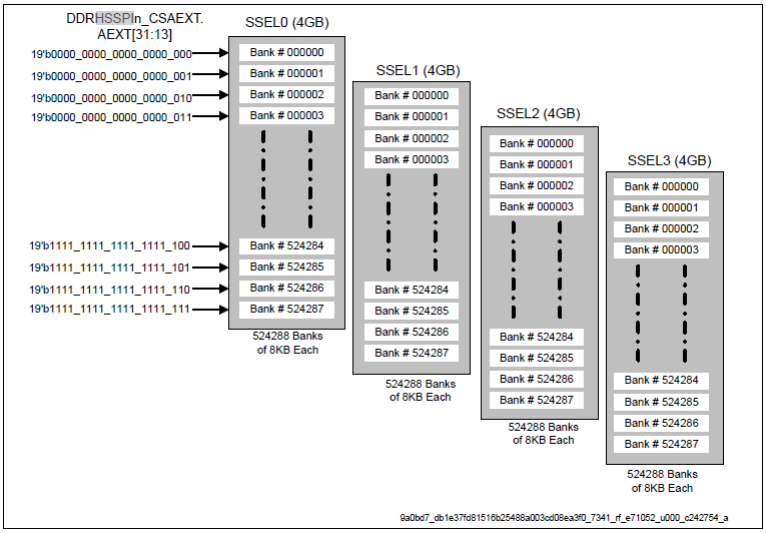
Page	Section	Change Results
3105	CHAPTER 52: DDR High Speed SPI Controller 4.Operations of the DDRHSSPI 4.1.Direct Mode 4.1.3. TX-FIFO Control	Error) Table 4-1 Serial Data Output Control (2) DDRHSSPIn_DMTRP.TRP[1:0]= 0b11 (Only when DDRHSSPIn_MID.MID is 0x00000100 or 0x00000300) FWIDTH TRP[1:0] SDATA 0b00 0b11 {TXDATA[7:4], TXDATA[3:0]} will be sent on {SDATA[7:4],SDATA[3:0]}. When in DDR Mode (DDRHSSPIn_DMFIFOCFG.DDRM= 1), TRP= 0b11 is not allowed. 0b01 0b11 {TXDATA[15:12], TXDATA[11:8]} and {TXDATA[7:4], TXDATA[3:0]} will be sequentially sent on {SDATA[7:4], SDATA[3:0]}. 0b11 0b11 {TXDATA[31:28], TXDATA[27:24]}, {TXDATA[23:20], TXDATA[19:16]}, {TXDATA[15:12], TXDATA[11:8]} and {TXDATA[7:4], TXDATA[3:0]} will be sequentially sent on {SDATA[7:4], SDATA[3:0]}. Correct) Table 4-1 Serial Data Output Control (2) DDRHSSPIn_DMTRP.TRP[1:0]= 0b11 (Only when DDRHSSPIn_MID.MID is 0x00000100 or 0x00000300) (2-1) SDR Mode: FWIDTH SDATA 0b00 TXDATA[7:4] will be sent on SDATA[7:4]. TXDATA[3:0] will be sent on SDATA[3:0]. 0b01 TXDATA[15:12][7:4] will be sent on SDATA[7:4]. TXDATA[11: 8][3: 0] will be sent on SDATA[3:0]. (in the order of TXDATA[15:8]->[7:0], along the SCLK edge) 0b11 TXDATA[31:28][23:20][15:12][7:4] will be sent on SDATA[7:4]. TXDATA[27:24][19:16][11: 8][3:0] will be sent on SDATA[3:0]. (in the order of TXDATA[31:24]->[23:16]->[15:8]->[7:0] along the SCLK edge). (2-2) DDR Mode: FWIDTH SDATA 0b00 Not allowed. 0b01 TXDATA[31:28][23:20] will be sent on SDATA[7:4]. TXDATA[27:24][19:16] will be sent on SDATA[3:0]. (in the order of TXDATA[31:24]->[23:16], along the SCLK edge) 0b11 TXDATA[31:28][23:20][15:12][7:4] will be sent on SDATA[7:4]. TXDATA[27:24][19:16][11: 8][3:0] will be sent on SDATA[3:0]. (in the order of TXDATA[31:24]->[23:16]->[15:8]->[7:0] along the SCLK edge)

Page	Section	Change Results
3106	CHAPTER 52: DDR High Speed SPI Controller 4.Operations of the DDRHSSPI 4.1.Direct Mode 4.1.3. TX-FIFO Control	Error) Table 4-1 Serial Data Output Control The length of dummy cycles is limited as below. [Max Dummy Cycles] = [Bit number of 1 FIFO entry] / [1 Byte Time (cycle number)] For example, under the following conditions: − Quad Mode (by DDRHSSPIn_DMTRP.TRP[1:0]) − SDR data rate − number of dummy cycles is 4 − DDRHSSPIn_DMFIFOCFG.FWIDTH is 0b01 (16 bit-wide). In this example, the number of dummy cycle is within the limit. If the number of dummy cycles is larger than 4 in this example, the output serial data will be tri-stated for only 4 SCLK cycles and the rest will be ignored. In TX-and-RX Mode, data on the input port iDDRHSSPIn_SDATA is sampled along the output of oDDRHSSPIn_SDATA. The number of received byte data is calculated as below. [Rx Byte Count] = [Dummy Cycles] / [1 Byte Time (cycle number)] The number of dummy cycles shall be a multiple of 1 Byte Time. RX-FIFO is being filled also during the dummy cycles in TX-and-RX Mode, so the software needs to read RX-FIFO to remove these received data. Correct) Table 4-1 Serial Data Output Control The length of dummy cycles is limited as below. [Max Dummy Cycles] = [Bits per 1 FIFO entry] / [Bits per 1 SCLK cycle] For example, under the following conditions: − Quad Mode (by DDRHSSPIn_DMTRP.TRP[1:0]) − SDR data rate − number of dummy cycles is 4 − DDRHSSPIn_DMFIFOCFG.FWIDTH is 0b01 (16 bit-wide). In this example, the number of dummy cycle is within the limit. If the number of dummy cycles is larger than 4 in this example, the output serial data will be tri-stated for only 4 SCLK cycles and the rest will be ignored.
3108	CHAPTER 52: DDR High Speed SPI Controller 4.Operations of the DDRHSSPI 4.1.Direct Mode 4.1.3. TX-FIFO Control	Added the below: (in the order of TXDATA[31:24]->[23:16] along the SCLK edge)

Page	Section	Change Results
3109	CHAPTER 52: DDR High Speed SPI Controller 4.Operations of the DDRHSSPI 4.1.Direct Mode 4.1.4. Available Command Formats of Serial Flash Memory	Error) Note: − When in writing TXDATA[12:8] for SPI control on-the-fly, it is allowing regardless the setting of DDRHSSPIn_DMFIFOCFG.FWIDTH. The available command formats of Serial Flash are different according to the value of DDRHSSPIn_MID. MID. Correct) Note: − When in writing TXDATA[12:8] for SPI control on-the-fly, it is allowed regardless the setting of DDRHSSPIn_DMFIFOCFG.FWIDTH. The available command formats of Serial Flash are different according to the value of DDRHSSPIn_MID. MID bits.
3112	CHAPTER 52: DDR High Speed SPI Controller 4.Operations of the DDRHSSPI 4.1.Direct Mode 4.1.5. FIFO Accesses in Direct Mode	Error) When in DDRHSSPIn_DMFIFOCFG.TXCTRL= 1, the setting of DDRHSSPIn_DMFIFOCFG.FWIDTH is ignored at writing into TX-FIFO, and up to 32 bits could be relevant. (please refer to the Table 4-1). Note: − In case that DDRHSSPI is configured in: − DDR Mode, AND − Dual Quad Mode setting DDRHSSPIn_DMFIFOCFG.FWIDTH to 0b00 (8-bit wide mode) is not allowed. Correct) When in DDRHSSPIn_DMFIFOCFG.TXCTRL= 1, the setting of DDRHSSPIn_DMFIFOCFG.FWIDTH is ignored at writing into TX-FIFO, and up to 32 bits could be relevant. (For details, please refer to the Table 4 1).
3112, 3113	CHAPTER 52: DDR High Speed SPI Controller 4.Operations of the DDRHSSPI 4.1.Direct Mode 4.1.6. Service Requests	Added the below: − The DDRHSSPIn_TXF.TFES is set if the DDRHSSPIn_DMFIFOSTATUS.TXFLEVEL is 0, i.e. TX-FIFO is empty. − The DDRHSSPIn_RXF.RFFS is set if the DDRHSSPIn_DMFIFOSTATUS.RXFLEVEL is 24, i.e. RX-FIFO is full.

Page	Section	Change Results
3113	CHAPTER 52: DDR High Speed SPI Controller 4.Operations of the DDRHSSPI 4.1.Direct Mode 4.1.6. Service Requests	Error) Each of these counters is a 5 bits down counter, which is reloaded with the DMA Block size (for the respective channel) whenever the DMA Service Request for that channel is asserted. The counters are decremented with every successful read or write accesses to the RX-FIFO or TX-FIFO. In case of the RX-FIFO accesses, the RX Block Counter is decremented only if the access was from a System Bus master other than the DAP Controller. Correct) Each of these counters is a 5 bits down counter, which is reloaded with the DMA Block size (for the respective channel) each time the DMA Service Request for that channel is asserted. The counters are decremented with every successful read or write accesses to the RX-FIFO or TX-FIFO. In case of the RX-FIFO accesses, the RX Block Counter is decremented unless the access was from the DAP Controller.
3115	CHAPTER 52: DDR High Speed SPI Controller 4.Operations of the DDRHSSPI 4.1.Direct Mode 4.1.7. SPI Transfers	Error) When the DDRHSSPI is working in Direct Mode, it can be interfaced with up to 4 Serial Flash Memories. Correct) When the DDRHSSPI is working in Direct Mode, it can be interfaced with up to 4 Serial Flash Memories at Legacy/Quad mode (8 at Dual Legacy/Dual Quad mode).
3116	CHAPTER 52: DDR High Speed SPI Controller 4.Operations of the DDRHSSPI 4.1.Direct Mode 4.1.7. SPI Transfers	Deleted the below: Note: − When in TX-and-RX Mode, please avoid halting the serial transfer.

Page	Section	Change Results
3117	CHAPTER 52: DDR High Speed SPI Controller 4.Operations of the DDRHSSPI 4.2. Command Sequencer Mode	Error) Since the first SPI transaction is controlled under TX-Only Mode, the read data from the Serial Flash are ignored. Valid reading of Serial Flash is run at the Command Sequencer Mode. In memory reading, the bit width on the SPI from the address field should be constantly Quad. Correct) Since the first SPI transaction is controlled under TX-Only Mode, the read data from the Serial Flash are ignored in that transaction. Valid reading of Serial Flash is run at the Command Sequencer Mode. In memory reading, the bit width on the SPI from the address field should be constantly Quad (or Dual Quad).
3118	CHAPTER 52: DDR High Speed SPI Controller 4.Operations of the DDRHSSPI 4.2. Command Sequencer Mode	Error) Generation of 32-Bit Memory Address The Address Extension Mechanism allows the mapping of 256MB of the MCU address space to 4GB of address space on a Slave Select line. Every Serial Flash Memory can be visualized as entity, consisting of several memory banks. The size of each bank can be programmed in the DDRHSSPIn_CSCFG.MSEL. Each bank can be selected by changing the value in the DDRHSSPIn_CSAEXT Register. By reprogramming the DDRHSSPIn_CSAEXT Register, each time a new bank in the selected Serial Flash Memory is accessed. Through different banks, it is possible to address a Serial Flash Memory of up to 4GB size. Correct) Generation of 32-Bit Memory Address The Address Extension Mechanism allows the MCU to access each Serial Flash Memory of a size of 4GB. Each Serial Flash Memory is accessible by dividing into several memory banks. The size of each bank can be programmed in the DDRHSSPIn_CSCFG.MSEL. Each bank can be selected by changing the value in the DDRHSSPIn_CSAEXT Register. To select a new bank, the software shall program a new value to the DDRHSSPIn_CSAEXT Register. By switching banks, it is possible to address a Serial Flash Memory of 4GB size.

Page	Section	Change Results
3119	CHAPTER 52: DDR High Speed SPI Controller 4. Operations of the DDRHSSPI 4.2. Command Sequencer Mode	Error) **Figure 4-6 Addressing 4GB Serial Flash Memories on Each Slave Select, Through Different Banks (DDRHSSPIn_CSCFG.MSEL= 0b0000)**  Correct) **Figure 4-6 Addressing 4GB Serial Flash Memories on Each Slave Select, Through Different Banks (DDRHSSPIn_CSCFG.MSEL= 0b0000)** 

Page	Section	Change Results
119	CHAPTER 52: DDR High Speed SPI Controller 4. Operations of the DDRHSSPI 4.2. Command Sequencer Mode	Error) The Least Significant Bits of the System Bus Address received by the DDRHSSPI on the System Bus are used as the offset within the bank selected by the Address Extension bits. The concatenation of the appropriate number of bits from the Address Extension Register with the appropriate number of bits from the address bus, determine the 32-bit address of the memory to be accessed on the Serial Interface. Please refer to Table 4-5. Correct) The Least Significant Bits of the System Bus Address received by the DDRHSSPI on the System Bus are used as the offset within the bank selected by the Address Extension bits. The final 32-bit address on the SPI is generated by the following concatenation: − Upper bits: Register bits DDRHSSPIn_CSAEXT.AEXT[31:m] − Lower bits: Address on the System Bus[m-1:0] (m = DDRHSSPIn_CSCFG.MSEL[3:0] + 13.) For details, please refer to Table 4-5.

Page	Section	Change Results
3120	CHAPTER 52: DDR High Speed SPI Controller 4. Operations of the DDRHSSPI 4.2. Command Sequencer Mode	Error) Last two columns in Table 4-5 indicate which bits from the DDRHSSPIn_CSAEXT.AEXT and the address bus (i.e. HADDR) are concatenated, to get the final 32-bit address of the Serial Flash Memory. Although, the final memory address generated in this way is a 32-bit address, it must be noted, that the software can choose the number of bytes (from this 32-bit address) to be sent to the Serial Flash Memory during the address phase of a memory read Command Sequence. Internal Prefetch Buffer In Command Sequencer Mode the DDRHSSPI internally has one Prefetch Buffer for temporary storage, of the data to be received (Prefetch Buffer). This Prefetch Buffer consists of 48 locations each 32-bit wide. Prefetch Buffer shall support 32 bits, 16 bits and 8 bits reads using the following mechanism: In generation of the Command Sequence the two least significant address bits shall be ignored. When a byte or halfword is read at a random (non-sequential) address, the Command Sequence shall be generated at the aligned word address which contains this byte or halfword. In any access scenario where the required data can be read from the top most position of the Prefetch Buffer, there will be no additional Command Sequence. When the Prefetch Buffer is read (as FIFO), only if the read includes the last byte within the actual Prefetch Buffer word, it will be popped. Following are typical read scenarios with explanation when the Prefetch Buffer pops: In order to obtain maximum throughput, 32-bit access is highly recommended. Correct) Last two columns in Table 4-5 indicate which bits from the DDRHSSPIn_CSAEXT.AEXT and the address bus (i.e. HADDR) are concatenated, to get the final 32-bit address of the Serial Flash Memory. The output memory address on the SPI can be 32 or 24 bits according to the Serial Flash Memory products, and this is controlled by the register settings of DDRHSSPIn_RDCSDC0-11 (for details, refer toTable 4-6). Internal Prefetch Buffer In Command Sequencer Mode the DDRHSSPI internally has one Prefetch Buffer for a temporary storage of received data. This Prefetch Buffer consists of 48 locations, 32-bit wide. Prefetch Buffer shall support 32 bits, 16 bits and 8 bits reads using the following mechanism: − When a byte or halfword is read at a random (non-sequential) address on the System Bus, the Command Sequence is generated with the address aligned to the word boundary, which covers the required data. − In the actual Command Sequence in transmission, the two least significant address bits are fixed to "00" (i.e. these two bits from the System Bus are ignored). As long as the software reads a series of continuous data from the top most position of the Prefetch Buffer, there will be no additional Command Sequence on the SPI. When the Prefetch Buffer is read, it internally works as a FIFO, and a word data in the next location pops out only when the current read data includes the last byte in the word line. The typical cases to pop the Prefetch Buffer are as below: In order to obtain the maximum throughput, word access (i.e. 32-bit access) is highly recommended.

Page	Section	Change Results
3121	CHAPTER 52: DDR High Speed SPI Controller 4. Operations of the DDRHSSPI 4.2. Command Sequencer Mode	Error) Initiation of Command Sequence Whenever the Command Sequencer receives a System Bus read access for the memory mapped Serial Flash Memory, it initiates a corresponding memory read command on one of the four Slave Select lines, assembles the data it has received, and responds with the memory read-data. Only Flash Memory read commands are supported in the Command Sequencer Mode. While the DDRHSSPI initiates a memory-read command and receives the read-data from the Serial Flash Memory, the System Bus Slave port of the DDRHSSPI inserts WAIT states on the System Bus. The DDRHSSPI keeps track of the previous address and the System Bus transfer type issued by the System Bus master. If the new transaction address is not contiguous, then a new serial transaction is issued on the Serial Interface. Correct) Initiation of Command Sequence Whenever the Command Sequencer detects a System Bus read access for the memory mapped Serial Flash Memory, it initiates a corresponding Memory Read Command Sequence on the SPI, assembles the received data, and responds to the System Bus with the Memory Read data. The Command Sequencer Mode supports only Memory Read commands of Serial Flash Memory. While the DDRHSSPI initiates a memory-read command and receives the read-data from the Serial Flash Memory, the DDRHSSPI inserts WAIT states on the System Bus. The DDRHSSPI keeps track of the previous address and the System Bus transfer type issued by the System Bus master. If the new transaction address is not contiguous, then a new serial transaction is issued on the Serial Interface.

Page	Section	Change Results
3122	CHAPTER 52: DDR High Speed SPI Controller 4. Operations of the DDRHSSPI 4.2. Command Sequencer Mode	Error) **Idle Timeout** After a Serial Flash Memory is accessed in the Command Sequencer Mode, the DDRHSSPI keeps asserting the Slave Select line even if the System Bus transaction is over. And DDRHSSPI can de-assert the Slave Select automatically after waiting for a certain period of time. To use this feature, it is necessary to set DDRHSSPIn_CSCFG.ITIMEREN bit to "1". If this feature is disabled, Slave Select remains asserted after any transfer on the Serial Flash Memory. If this feature is enabled, the following part explains the Idle Timer behavior: During idle cycles on the System Bus, prefetch reads from the Serial Flash Memories to the Prefetch Buffer are performed. As soon as the Prefetch Buffer is full, the Slave Select shall be kept asserted, and the Idle Timer shall start running. If there are no subsequent System Bus accesses to the consecutive memory address during the idle time, then after the Idle Timer expires, the DDRHSSPI de-asserts the Slave Select, indicating the termination of the transfer and flushes the Prefetch Buffer. Whenever a prefetch is performed before the Idle Timer expires, the Idle Timer shall be cleared again. Within the predefined time period, defined by the DDRHSSPIn_CSITIME.ITIME, the DDRHSSPI determines whether to extend the current serial transaction. If the following conditions are satisfied: Correct) **Idle Timeout** After a Serial Flash Memory is accessed in the Command Sequencer Mode, the DDRHSSPI keeps asserting the Slave Select line even if the System Bus transaction is over. And DDRHSSPI can deassert the Slave Select automatically after waiting for a certain period of time. To use this feature, it is necessary to set DDRHSSPIn_CSCFG.ITIMEREN bit to "1". If this feature is disabled, Slave Select remains asserted after any transfer on the Serial Flash Memory. If this feature is enabled, the following part explains the Idle Timer behavior: During idle cycles on the System Bus, DDRHSSPI continues reading the Serial Flash Memory until the prefetch Buffer reaches full. After reaching full, the Slave Select shall stay asserted, and the Idle Timer shall start running. If there are no subsequent System Bus accesses to the consecutive memory address during the programmed idle time, the DDRHSSPI deasserts the Slave Select after the Idle Timer expires. The deassertion of Slave Select indicates the termination of the transfer and flushes the Prefetch Buffer. If the consecutive memory read access resumes before the Idle Timer expires, the Idle Timer shall be cleared again. Within the predefined time period, defined by the DDRHSSPIn_CSITIME.ITIME, the DDRHSSPI determines whether to extend the current serial transaction. If the following conditions are satisfied:

Page	Section	Change Results
3124, 3125	CHAPTER 52: DDR High Speed SPI Controller 4. Operations of the DDRHSSPI 4.2. Command Sequencer Mode	Error) 5. Set the DDRHSSPIn_DMFIFOCFG.FWIDTH to 0b00. 11. Write "1" to the DDRHSSPI_TXC.TSSRC bit. 12. Set the DDRHSSPI_TXE.TSSRE bit to "1". 13. Select a Serial Flash Memory by setting the DDRHSSPI_DMPSEL.PSEL value. 14. Set the DDRHSSPIn_DMSTART.START bit to "1". 15. Wait for the TX Interrupt Request, or run a polling of DDRHSSPIn_TXF Register until the DDRHSSPI_TXF.TSSRS bit to be "1". 16. Write "1" to the DDRHSSPI_TXC.TSSRC bit. 17. Clear the DDRHSSPI_TXE.TSSRE bit to "0". 18. Switch to Command Sequencer Mode. Set DDRHSSPIn_MCTRL.CSEN bit to "1". 19. Set the DDRHSSPIn_CSCFG.MBM same as DDRHSSPIn_DMTRP.TRP[1:0] value. Set DDRHSSPIn_CSCFG.MSEL to appropriate value. Set at least one of DDRHSSPIn_CSCFG.SSEL0EN, DDRHSSPIn_CSCFG.SSEL1EN, DDRHSSPIn_CSCFG.SSEL2EN or DDRHSSPIn_CSCFG.SSEL3EN to "1", in order to contain the Serial Flash Memory selected by DDRHSSPI_DMPSEL.PSEL. Set DDRHSSPIn_CSCFG.DDRMODE bit same as DDRHSSPIn_DMTRP.DDRM bit. Correct) 10. Write "1" to the DDRHSSPIn_TXC.TSSRC bit. 11. Set the DDRHSSPIn_TXE.TSSRE bit to "1". 12. Select a Serial Flash Memory by setting the DDRHSSPIn_DMPSEL.PSEL value. 13. Set the DDRHSSPIn_DMSTART.START bit to "1". 14. Wait for the TX Interrupt Request, or run a polling of DDRHSSPIn_TXF Register until the DDRHSSPIn_TXF.TSSRS bit to be "1". 15. Write "1" to the DDRHSSPIn_TXC.TSSRC bit. 16. Clear the DDRHSSPIn_TXE.TSSRE bit to "0". 17. Switch to Command Sequencer Mode. Set DDRHSSPIn_MCTRL.CSEN bit to "1". 18. Set the DDRHSSPIn_CSCFG.MBM same as DDRHSSPIn_DMTRP.TRP[1:0] value. Set DDRHSSPIn_CSCFG.MSEL to appropriate value. Set at least one of DDRHSSPIn_CSCFG.SSEL0EN, DDRHSSPIn_CSCFG.SSEL1EN, DDRHSSPIn_CSCFG.SSEL2EN or DDRHSSPIn_CSCFG.SSEL3EN to "1", in order to contain the Serial Flash Memory selected by DDRHSSPIn_DMPSEL.PSEL. Set DDRHSSPIn_CSCFG.DDRMODE bit same as DDRHSSPIn_DMTRP.DDRM bit.

Page	Section	Change Results
3127	CHAPTER 52: DDR High Speed SPI Controller 4. Operations of the DDRHSSPI 4.3. Address Map of DDRHSSPI	Deleted the below: − Allocation for Serial Flash Memories The 256MB of memory space, starting from "DDRHSSPI Memory Base Address" is reserved for memory mapping of the external Serial Flash Memories onto the MCU's address space. This space is used in Command Sequencer Mode. − Allocation for CSRs The 1KB of memory space, starting from "DDRHSSPI CSR Base Address" is reserved for memory mapping of the Configuration and Status Registers of DDRHSSPI on to the MCU's address
3128	CHAPTER 52: DDR High Speed SPI Controller 4.4. General Use Case Guidelines for DDRHSSPI 4.4.1. Guidelines on Typical Use Cases of DDRHSSPI	Error) − Only one configurable FIFO is used, with different configuration settings, for each Direct Mode and Command Sequencer Mode. Hence, when switching from one mode to another, all FIFO data will be lost. − In Direct Mode, whenever the transfer ends, the data from the RX Shift Register is pushed into the RX-FIFO; provided that the RX-FIFO is not full. Correct) − Direct Mode and Command Sequencer Mode have different configurations of the FIFO, which is the common resource in DDRHSSPI. Hence, when switching from one mode to another, all FIFO data will be lost. − In Direct Mode with TX-and-RX Mode, whenever the transfer ends, the data from the RX Shift Register is pushed into the RX-FIFO; provided that the RX-FIFO is not full.

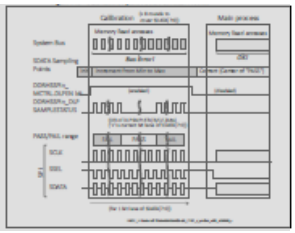
Page	Section	Change Results
3129	CHAPTER 52: DDR High Speed SPI Controller 4.4. General Use Case Guidelines for DDRHSSPI 4.4.2. Steps in Programming the DDRHSSPI Module	Error) 2. The next step is to configure the Attributes related to the Peripheral Communication with the Serial Flash Memory(Memories) connected with DDRHSSPI. DDRHSSPI can be interfaced with up to 4 Serial Flash Memories. Serial communication related attributes like Transfer Frequency (i.e. Clock Division Ratio bits), etc. shall be configured in the registers: DDRHSSPIn_PCC0-3. It is very important that these attributes shall be the same as the Serial Flash Memory, which is connected with DDRHSSPI. 3. DDRHSSPI can be configured either in Direct Mode or in Command Sequencer Mode, through the DDRHSSPIn_MCTRL.CSEN bit. Depending on which mode is to be used, the software shall configure the mode-specific registers. Correct) 2. The next step is to configure the Attributes related to the Peripheral Communication with the Serial Flash Memory(Memories) connected with DDRHSSPI. DDRHSSPI can be interfaced with up to 4 Serial Flash Memories at Legacy/Quad mode (8 at Dual Legacy/Dual Quad mode). Serial communication related attributes like Transfer Frequency (i.e. Clock Division Ratio bits), etc. shall be configured in the registers: DDRHSSPIn_PCC0-3. It is very important that these attributes shall be identical among the Serial Flash Memories connected with DDRHSSPI. 3. DDRHSSPI can be configured either in Direct Mode or in Command Sequencer Mode, by the DDRHSSPIn_MCTRL.CSEN bit. Depending on which mode is to be used, the software shall configure the mode-specific registers. When DDRHSSPIn_MID.MID is 0x00000300, there are additional registers (DDRHSSPIn_SDATASAMPLEPT{CNT/LFT/RGH}0-7, DDRHSSPIn_DLP) in Command Sequencer Mode.
3131	CHAPTER 52: DDR High Speed SPI Controller 4. Operations of the DDRHSSPI 4.4. General Use Case Guidelines for DDRHSSPI 4.4.3. Using the DDRHSSPI in Direct Mode of Operation	Error) 7. When DDRHSSPI is configured, setting the DDRHSSPIn_DMSTART.START bit triggers the start of the serial transaction. Once the serial transaction starts, if transmission is enabled in the DDRHSSPIn_DMTRP.TRP, the DDRHSSPI reads data from TX-FIFO and loads them to the Shift Register. The Shift Register is shifted left and the transmit data is shifted-out onto the Serial Interface. Correct) 7. When DDRHSSPI is configured, setting the DDRHSSPIn_DMSTART.START bit triggers the start of the serial transaction. Once the serial transaction starts, if the transmit data is available in the TX-FIFO, the DDRHSSPI reads data from TX-FIFO and loads them to the Shift Register. The Shift Register is shifted left and the transmit data is shifted-out onto the Serial Interface.

Page	Section	Change Results
3131	CHAPTER 52: DDR High Speed SPI Controller 4. Operations of the DDRHSSPI 4.4. General Use Case Guidelines for DDRHSSPI 4.4.3. Using the DDRHSSPI in Direct Mode of Operation	Error) 9. Service Requests are asserted by DDRHSSPI whenever the TX-FIFO level is below the threshold or whenever the DDRHSSPI RX-FIFO level is above the threshold. The software shall write TX-FIFO or read RX-FIFO, to ensure the serial data transfer of DDRHSSPI. After writing or reading the relevant FIFO, the software shall clear the Interrupt Service Requests by writing the DDRHSSPIn_TXC or the DDRHSSPIn_RXC Register. DMA Service Requests are cleared by the DMA Controller. 10. If reception is enabled in DDRHSSPIn_DMTRP Register, then the software fetches the received data from the RX-FIFO. Correct) 9. Service Requests are asserted by DDRHSSPI whenever the TX-FIFO level is below the threshold or whenever the DDRHSSPI RX-FIFO level is above the threshold. The software shall write TX-FIFO or read RX-FIFO, to ensure the serial data transfer of DDRHSSPI. After writing or reading the relevant FIFO, the software shall clear the Interrupt Service Requests by writing the DDRHSSPIn_TXC or the DDRHSSPIn_RXC Register. DMA Service Requests are cleared by the handshake responses from the DMA Controller. When the data transfer is not by DMA, the Service Requests can be asserted also by TX-FIFO's empty or RX-FIFO's full status. The software can select the conditions. 10. If reception is enabled in DDRHSSPIn_DMTRP Register (when in TX-and-RX Mode), then the software fetches the received data from the RX-FIFO.
3132	CHAPTER 52: DDR High Speed SPI Controller 4. Operations of the DDRHSSPI 4.4. General Use Case Guidelines for DDRHSSPI 4.4.4. Using the DDRHSSPI in Command Sequencer Mode of Operation	Error) Interfacing of one Serial Flash Memory which requires 32-bit addressing and other Serial Flash Memory (also of same family, but) which requires only 24-bit addressing in Command Sequencer Mode is not possible. − On-the-fly switching of programmed Data Rate Mode (SDR Mode or DDR Mode) is not allowed in Command Sequencer Mode while module is enabled. Correct) Interfacing Serial Flash Memories mixing 32-bit addressing and 24-bit addressing is not allowed in Command Sequencer Mode. − On-the-fly switching of Data Rate Mode (SDR/DDR Mode programmed in DDRHSSPIn_CSCFG.DDRMODE) is not allowed in Command Sequencer Mode while module is enabled.

Page	Section	Change Results
3133	CHAPTER 52: DDR High Speed SPI Controller 4. Operations of the DDRHSSPI 4.4. General Use Case Guidelines for DDRHSSPI 4.4.4. Using the DDRHSSPI in Command Sequencer Mode of Operation	Error) 3. Program the DDRHSSPIn_CSCFG.MSEL, with the size of the System Bus address space which must be used in selection of the Serial Flash Memory on which the serial transfer must be initiated. Please refer to Section 4.24.2for details of the Slave Select. 4. If the addresses generated for the memory-mapped accesses are to be virtually extended to cover a memory range of virtually 16GB, the DDRHSSPIn_CSAEXT Register value gives the upper bits of the address. Please refer to Section 4.2for details of address generation. Correct) 3. Program the DDRHSSPIn_CSCFG.MSEL, according to - The address space on the System Bus for mapping the Serial Flash Memory (Memories) - The number of Serial Flash Memories interfaced with DDRHSSPI Please refer to Section 4.2 for details of the Slave Select. 4. To cover the physical address space in the Serial Flash Memories (up to 16GB), the DDRHSSPIn_CSAEXT Register value gives the

Page	Section	Change Results
3134	CHAPTER 52: DDR High Speed SPI Controller 4. Operations of the DDRHSSPI 4.4. General Use Case Guidelines for DDRHSSPI 4.4.4. Using the DDRHSSPI in Command Sequencer Mode of Operation	Added the below: **Calibration of SDATA Sampling Points** Before running the main process, the software shall adjust the SDATA Sampling Points by setting up registers DDRHSSPIn_SDATASAMPLEPT{LFT/CNT/RGH}0-7. To determine the best sampling points on each bit lane of SDATA[7:0], the software has to run a flow (1)~(3) as below. This flow is an example. The overview of this flow is also illustrated in Figure 4-13. (1) Calibration sequence First of all, the software needs to prepare arrays to store the result of Calibration. Each array has a size corresponding to SDATA Sampling Points (=128). It is recommended to prepare 8 arrays to store results for each bit lane of SDATA[7:0]. The software runs the following sequence: − a) Set DDRHSSPIn_MCTRL.DLPEN bit to "1". − The Calibration sequence goes with causing Bus Errors, and these errors have to be ignored by the system. Please make the settings on the system for it. − b) Set all {R/C/L} of SDATA Sampling Points to 0. − The target registers are for current bit lane of SDATA[7:0]. − E.g. If current bit lane is SDATA[0], write 0x00 to registers DDRHSSPIn_SDATASAMPLEPT{LFT/CNT/RGH}0. − c) Run a Memory Read access at 32-bits width. − From the second round in the loop, the software shall increment the address at least +8. − If any of SDATA Sampling Points is not in the valid bit range, the DLP checking results in error and causes a Bus Error. − d) Check the status of DDRHSSPIn_DLPSAMPLESTATUS register. − The software shall check the bit fields for current bit lane of SDATA[7:0], and store the result to the corresponding array. − e) Clear the DLPFS bit by writing "1" to DDRHSSPIn_FAULTC.DLPFC. − f) Increment all {R/C/L} of SDATA Sampling Points (+1 on each). − g) Repeat (c)~(f) until the max value of SDATA Sampling Points (128 rounds). − h) Repeat (b)~(g) until completing all bit lanes of SDATA[7:0] (8 rounds). − i) At this moment, DDRHSSPI is expected to be in the status as below: − SDATA Sampling Points are the max value. − The last Memory Read access has ended in Bus Error. − DDRHSSPIn_FAULTF.DLPFS="1" − DDRHSSPIn_RXF.DLPERR="1" − The software holds all arrays of "PASS"/"FAIL" results for each bit lane of SDATA[7:0].

Page	Section	Change Results
3134, 3135	CHAPTER 52: DDR High Speed SPI Controller 4. Operations of the DDRHSSPI 4.4. General Use Case Guidelines for DDRHSSPI 4.4.4. Using the DDRHSSPI in Command Sequencer Mode of Operation	Added the below: (2) Setting SDATA Sampling Points Set the registers DDRHSSPIn_SDATASAMPLEPTCNT0-7 to the center positions in respective "PASS" ranges. If there is no "PASS" range in the array, the DDRHSSPI hereafter cannot serve Memory Read access. Else if the "PASS" range is divided by "FAIL(s)", choose the longest "PASS" range and take the center position of it. DDRHSSPIn_SDATASAMPLEPTLFT0-7 shall be set to less values than DDRHSSPIn_SDATASAMPLEPTCNT0-7 (for each bit lane of SDATA[7:0]). DDRHSSPIn_SDATASAMPLEPTRGH0-7 shall be set to greater values than DDRHSSPIn_SDATASAMPLEPTCNT0-7 (for each bit lane of SDATA[7:0]). These register settings for {"LFT", "RGH"} are actually not used in the data reception, and they are just used for detecting the violation of sampling. The degree of 'less' and 'greater' depends on the window size of "PASS" range, and these differences are used in the re-adjustment of SDATA Sampling Points afterward. (3) To the main process The software shall clear DDRHSSPIn_MCTRL.DLPEN bit to "0", if it is not allowed to lose the Memory Read data in the main process by the occurrence of DLP error. Now the DDRHSSPI is ready for the Memory Read access in the main process. During the main process, the software is required to run a periodical adjustment of SDATA Sampling Points, and this operation is described later in this section. ***Notes:*** *– If the Serial Flash Memory does not support the DLP function, please change following points.* *– DDRHSSPIn_MCTRL.DLPEN should be "0".* *– DLP-related status bits are not used.* *– All Serial Flash Memories have to store certain data in certain addresses in advance. These* *data are used for pattern matching instead of DLP, and shall have bit-toggling pattern on each bit* *lane of SDATA[7:0] as well as DLP. Please consider the "bit pattern" and "bit alignment" according to the intended SPI protocol (Quad or Dual-Quad).*

Page	Section	Change Results
3135, 3136	CHAPTER 52: DDR High Speed SPI Controller 4. Operations of the DDRHSSPI 4.4. General Use Case Guidelines for DDRHSSPI 4.4.4. Using the DDRHSSPI in Command Sequencer Mode of Operation	Added the below: *– The software has to do pattern-matching checks with the read data, since there is no automatic checking like DLP. Please note that the read data work instead of DLP by the process of software.* *Therefore, the software shall recognize how many data to read in the DLP-like operation, and gather byte data by re-arranging bits from the read data. [e.g. Memory Read at "DDR & Dual-Quad" protocol] For the pattern matching check, it is assumed to use 8 consecutive bits on each bit lane of SDATA[7:0] as the substitute for DLP. The sum of these pattern-matching bits becomes 64 bits. In this case, the software has to read 2 words (assuming 32-bits access) to check all of these bits.* *– Although the SDATA[7:0] is sampled in parallel, the software has to check the serial bits on each bit lane of SDATA[7:0] individually. The main loop flow is basically same as the case of DLP.* *– The SPI transaction goes on regardless the sampling condition of SDATA[7:0], different from the case using DLP.* The overview of Calibration sequence is shown in the following diagram. **Figure 4-13 Calibration Sequence before the Main Process**  Before starting the main process, please read carefully the Notes of 3.4 SPI Data Learning Pattern. **Re-adjustment of SDATA Sampling Points** After setting the best position of SDATA Sampling Points, these points can gradually drift and get close to the edge of SDATA as time passes (e.g. due to the fluctuation of temperature), and the sampling may begin to fail. To avoid the miss-sampling, the software shall periodically adjust the SDATA Sampling Points. The actual interval depends on the system, however, it is expected that the software does not have to run a full Calibration sequence each time. If the periodical adjustment is frequent enough, each adjustment can change the SDATA Sampling Points +/- 1 [tap] at a time. This operation will not take time compared to the full Calibration, and it is expected that the system can insert this re-adjustment operation in the moment while the main process does not access the Serial Flash Memories. Furthermore, the system can determine the degree of adjustment not only within +/- 1 [tap]. The software runs the following sequence: a) Set DDRHSSPIn_MCTRL.DLPEN bit to "1". The handling of Bus Error is same as Calibration. Please make sure that the main process will not access Serial Flash Memories during (a)~(f) in this flow. b) Run a Memory Read access (at 32-bits width). The software shall increment the address +8 each time.

Page	Section	Change Results
3137	CHAPTER 52: DDR High Speed SPI Controller 4. Operations of the DDRHSSPI 4.4. General Use Case Guidelines for DDRHSSPI 4.4.4. Using the DDRHSSPI in Command Sequencer Mode of Operation	Added the below: c) Check the status of following registers: DDRHSSPIn_FAULTF.DLPFS bit (optional) If this bit is "0", go to (f). DDRHSSPIn_RXF.DLPERR bit (optional) If this bit is "0", go to (f). DDRHSSPIn_DLPSAMPLESTATUS register (must) For each bit lane of SDATA[7:0], the software shall check the status of sampling and update the corresponding SDATA Sampling Point. These judgments are based on the register settings of DDRHSSPIn_SDATASAMPLEPT{LFT/CNT/RGH}0-7 in the Calibration (L < C < R). - DLPSMPLSTx{R/C/L} = "100": --> Decrease the SDATA Sampling Point. - DLPSMPLSTx{R/C/L} = "001": --> Increase the SDATA Sampling Point. - DLPSMPLSTx{R/C/L} = "000": --> No need to change the SDATA Sampling Point. If the value of DDRHSSPIn_DLPSAMPLESTATUS register is 0x00000000, go to (f). - Otherwise: --> Outside the re-adjustable range. This situation requires a full Calibration and the system shall stop the main process for a moment. When increasing/decreasing the SDATA Sampling Points, the "CNT" shall be adjusted to the center between "LFT" and "RGH", so as to ensure the PASS range as much as possible. e.g. If DLPSMPLST0{R/C/L} = "100", the software shall decrease only the value of DDRHSSPIn_SDATASAMPLEPTRGH0 registers, then calculate the "CNT" as the average value between "LFT" and "RGH". d) If DLPFS = "1", write "1" to DDRHSSPIn_FAULTC.DLPFC to clear it. If DLPERR = "1", write "1" to DDRHSSPIn_RXC.DLPERRC to clear it. e) Repeat (b)~(d) until DDRHSSPIn_DLPSAMPLESTATUS = 0x00000000. If the period to run this operation is frequent enough against the fluctuation of SDATA timing, this loop is not always necessary. f) Clear DDRHSSPIn_MCTRL.DLPEN bit to "0", if it is not allowed to lose the Memory Read data in the main process by the occurrence of DLP error. g) Main process can access Serial Flash Memories. Notes: − If the Serial Flash Memory does not support the DLP function, the software has to check the result of Memory Read access as well as Calibration sequence (please refer to its Notes). − The system has to ensure the timing for this Re-adjustment operation. Before starting the main process, please read carefully the Notes of 3.4 SPI Data Learning Pattern.

Page	Section	Change Results
3148	CHAPTER 52: DDR High Speed SPI Controller 5. Registers 5.1. DDRHSSPI Module Control Register (DDRHSSPIn_MCTRL)	Added the below: Notes: − Under the condition of DLPEN="1", any DLP error in the DDRHSSPIn_DLPSAMPLESTATUS register results in the system bus error, and this is common among {LFT/CNT/RGH}. In that case, the Memory Read data is invalid. − Even if the CNT is sampled correctly, any error on the LFT or RGH leads to the error.
3150	CHAPTER 52: DDR High Speed SPI Controller 5. Registers 5.2. DDRHSSPI Peripheral Communication Configuration Registers (DDRHSSPIn_PCC0-3)	Error) [bit6:5] SS2CD[1:0]: Slave Select to Clock Delay (SS2CD + 0.5) x SCLK cycle When the Slave Select becomes active, the Serial Flash Memory has to prepare data transfer within the delay time defined by SS2CD bits. Correct) [bit6:5] SS2CD[1:0]: Slave Select to Clock Delay (SS2CD + 0.50) x SCLK cycle [SDR mode] (SS2CD + 0.75) x SCLK cycle [DDR mode] When the Slave Select becomes active, the Serial Flash Memory has to prepare data transfer within the delay time defined by SS2CD bits. Note: − SS2CD[1:0] = "00" must not be used when all the following conditions are met: - DDRHSSPIn_DMTRP.TRP[3:0] = "1011" - DDRHSSPIn_DMTRP.DDRM = "1" - DDRHSSPIn_DMFIFOCFG.TXCTRL = "1" - DDRHSSPIn_TXFIFOx.TXDATA[12][10][9:8] = "1000"

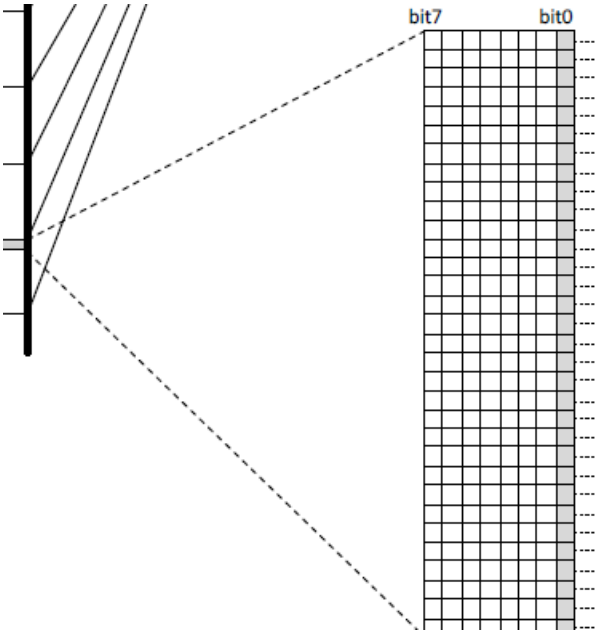
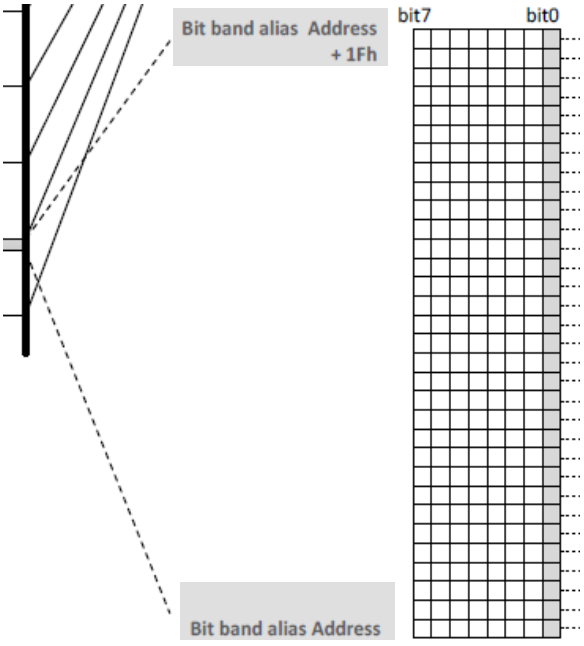
Page	Section	Change Results
3188	CHAPTER 52: DDR High Speed SPI Controller 5. Registers 5.25. DDRHSSPI SDATA Center Clock Sample Point Registers (DDRHSSPIn_SDATASAMPLEPTCNT0-7)	Added the below: Notes: − About the actual use cases, refer to the 4.4.4 Using the DDRHSSPI in Command Sequencer Mode of Operation -> "Calibration of SDATA Sampling Points", and "Re-adjustment of SDATA Sampling Points". − When in using
3189	CHAPTER 52: DDR High Speed SPI Controller 5. Registers 5.26. DDRHSSPI SDATA Left Clock Sample Point Registers (DDRHSSPIn_SDATASAMPLEPTLFT0-7)	Added the below: Notes: − About the actual use cases, refer to the 4.4.4 Using the DDRHSSPI in Command Sequencer Mode of Operation -> "Calibration of SDATA Sampling Points", and "Re-adjustment of SDATA Sampling Points". − When in using

Page	Section	Change Results
3190	CHAPTER 52: DDR High Speed SPI Controller 5. Registers 5.27. DDRHSSPI SDATA Right Clock Sample Point Registers (DDRHSSPIn_SDATASAMPLEPTRGH0-7)	Added the below: Notes: − About the actual use cases, refer to the 4.4.4 Using the DDRHSSPI in Command Sequencer Mode of Operation -> "Calibration of SDATA Sampling Points", and "Re-adjustment of SDATA Sampling Points". − When in using
3191	CHAPTER 52: DDR High Speed SPI Controller 5. Registers 5.28. DDRHSSPI Data Learning Pattern Register (DDRHSSPIn_DLP)	Added the below: ***Note:*** *− When in using the DLP check function, please read the Notes of DDRHSSPIn_MCTRL.DLPEN bit carefully.*

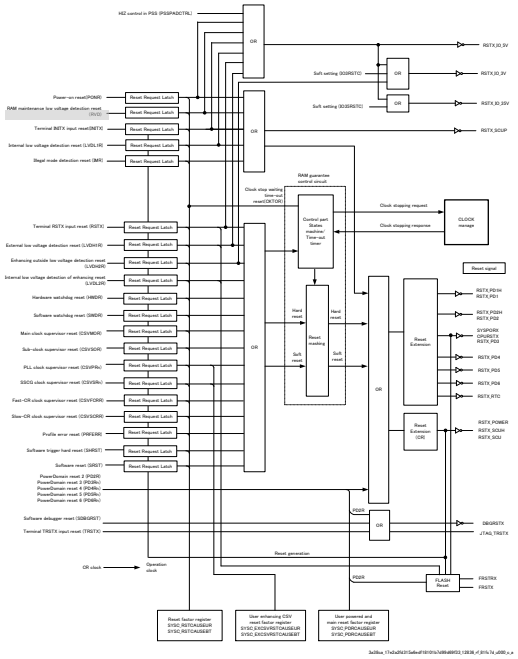
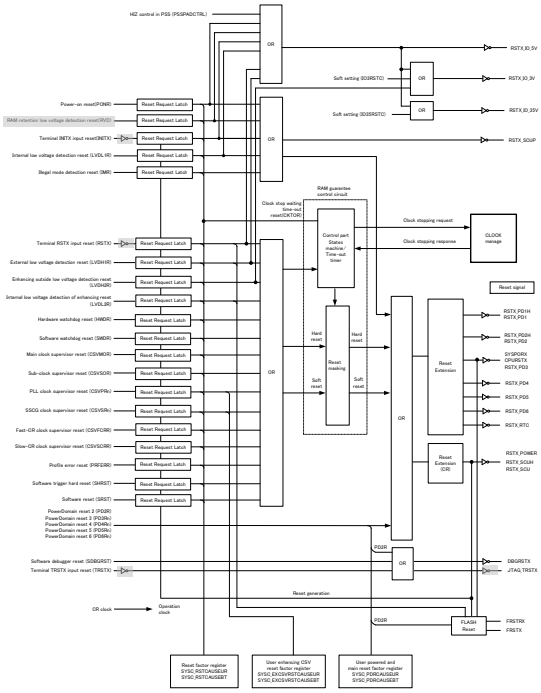
<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>3197</td><td>CHAPTER 52: DDR High Speed SPI Controller
5. Registers
5.29. DDRHSSPI Data Learning Pattern Sample Status Register (DDRHSSPIn_DLPSAMPLESTATUS)</td><td>Added the below:

Notes:
− When in using the DLP check function, please read the Notes of DDRHSSPIn_MCTRL.DLPEN bit carefully.
− These status bits are set along with DDRHSSPIn_FAULTF.DLPFS bit, however, the 'clear' conditions are different as below.
- DDRHSSPIn_FAULTF.DLPFS bit:
- Cleared by writing DDRHSSPIn_FAULTC.DLPFC="1".
- DDRHSSPIn_DLPSAMPLESTATUS DLPSMPLSTxx bits:
- Updated at the next Memory Read access (with discontinuous address).
- 'Set'/'Clear' depends on the latest result of the DLP check in the SPI transaction.</td></tr>
<tr><td>3206</td><td>CHAPTER 53 System SRAM module (SRAM_IF)
1.Outline of SRAM_IF</td><td>Error)
Table 1-1 Terms and Abbreviations<table><tr><th>Term</th><th>Meaning</th></tr><tr><td>SRAM</td><td>Static Random Access Memory</td></tr><tr><td>ECC</td><td>Error Checking and Correcting</td></tr><tr><td>ECU</td><td>Error Collection Unit</td></tr><tr><td>AXI</td><td>Advanced eXtensible Interface</td></tr></table>Correct)
Table 1-1 Terms and Abbreviations<table><tr><th>Term</th><th>Meaning</th></tr><tr><td>SRAM</td><td>Static Random Access Memory</td></tr><tr><td>ECC</td><td>Error Checking and Correcting</td></tr><tr><td>AXI</td><td>Advanced eXtensible Interface</td></tr></table></td></tr>
<tr><td>3209</td><td>CHAPTER 53 System SRAM module (SRAM_IF)
2.Operation of SRAM_IF</td><td>Error)
ECC Checker has detected a Single-bit error in the SRAM Read data (SRCFG_ERRFLG:SECFLAG is Set) and SRCFG_INTE:SEC_INT_EN bit is Set.
Correct)
ECC Checker has detected a Single-bit error in the SRAM Read data (SRCFG_ERRFLG:SECFLG is Set) and SRCFG_INTE:SEC_INT_EN bit is Set.</td></tr>
</table>

Page	Section	Change Results
3214	CHAPTER 53 System SRAM module (SRAM_IF) 3.SRAM_IF Register Set 3.3.SRAM_IF Configuration Register2(SRCFG_CFG2)	Error) – *Enabling RDB Bypass path (SRCFG_CFG2:BYPASSEN = 1) will reduce read latency since the read data from SRAM will be directly output to the Bus interface without buffering in RDB (Refer Figure 2-1).* Correct) – *Enabling RDB Bypass path (SRCFG_CFG2:BYPASSEN = 1) will reduce read latency since the read data from SRAM will be directly output to the Bus interface without buffering in RDB (Refer Figure 2-1).* – *Although the default value of this bit is 0, it is not required to configure this bit to 0.*

Page	Section	Change Results
3225	CHAPTER 54 Bit-Band Unit 2. Explanation of Operation	Error) **Figure 2-1 Bit-Band Alias Area and Bit-band Area**  Correct) **Figure 2-1 Bit-Band Alias Area and Bit-band Area** 

Page	Section	Change Results
002-04854 Rev. *D		

Page	Section	Change Results
80	CHAPTER 4:Reset 2. Configuration	Changed the Below **Figure 2-2 RST Manage Block Diagram** Error)  Correct) 

Page	Section	Change Results
101	CHAPTER 4:Reset 3.Operational Description	Correct) Added the below: 3.1.18. Slow CR Clock Supervisor Reset ***Notes:*** – Real Time Clock listed in the above table will be initialized if the operation clock of Real Time Clock is set to Slow CR clock.
115	CHAPTER 4:Reset 3.Operational Description	Changed the Below: 3.2.2. I/O Reset Table 3-30 I/O Reset Error) ***Notes:*** – **1: It is controlled by IO3RSTC in system special setting register (SYSC0_SPECFGR).* – **2: It is controlled by IO35RSTC in system special setting register (SYSC0_SPECFGR)* Correct) ***Notes:*** – **1: It is controlled by IO3RSTC in system special setting register (SYSC0_SPECFGR). RSTX_IO_3V becomes active by IO3RSTC, when IO3RSTC is initialized by hard reset or soft reset.* – **2: It is controlled by IO35RSTC in system special setting register (SYSC0_SPECFGR). RSTX_IO_35V becomes active by IO35RSTC, when IO35RSTC is initialized by hard reset or soft reset.*

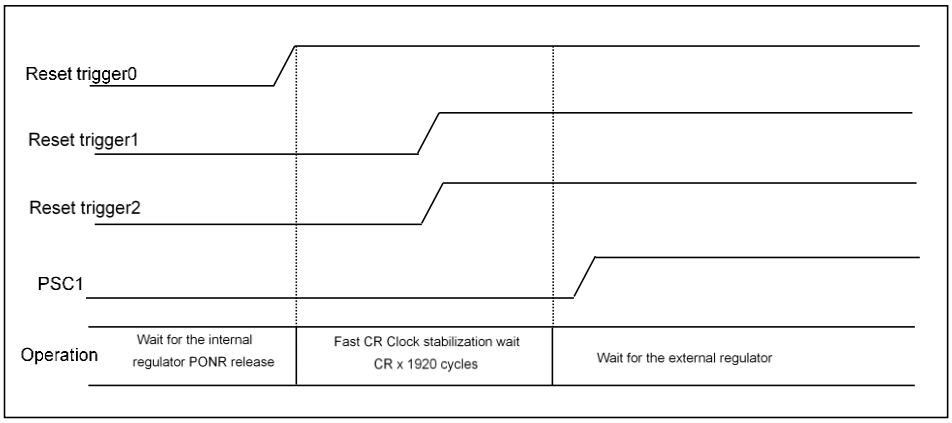
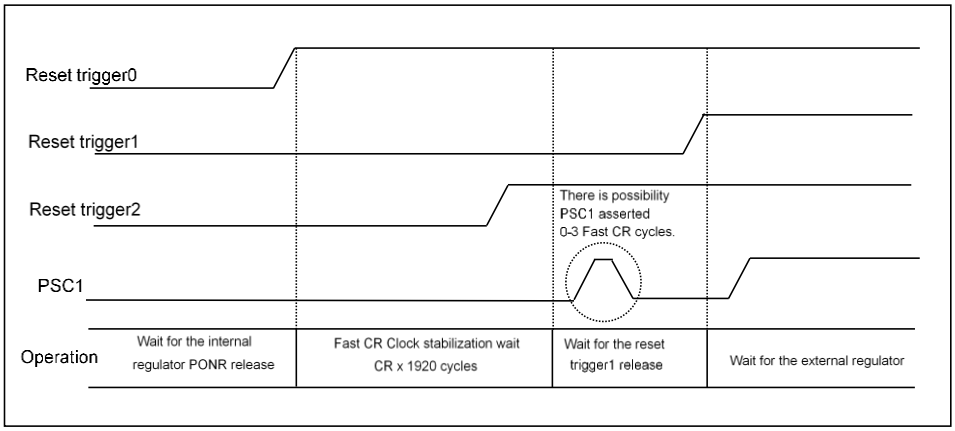
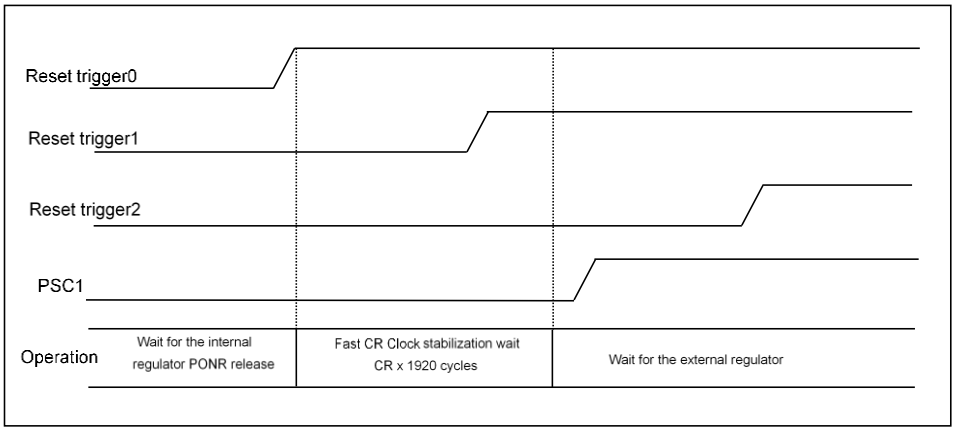
<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>117</td><td>CHAPTER 4: Reset

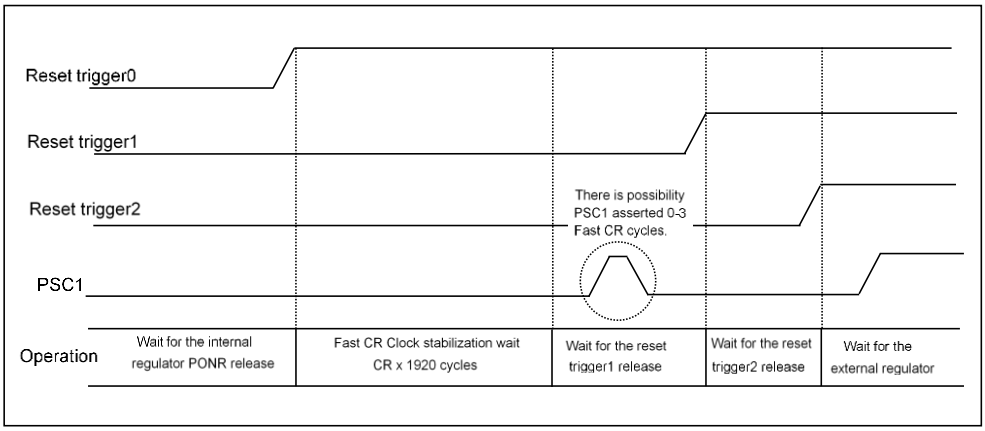
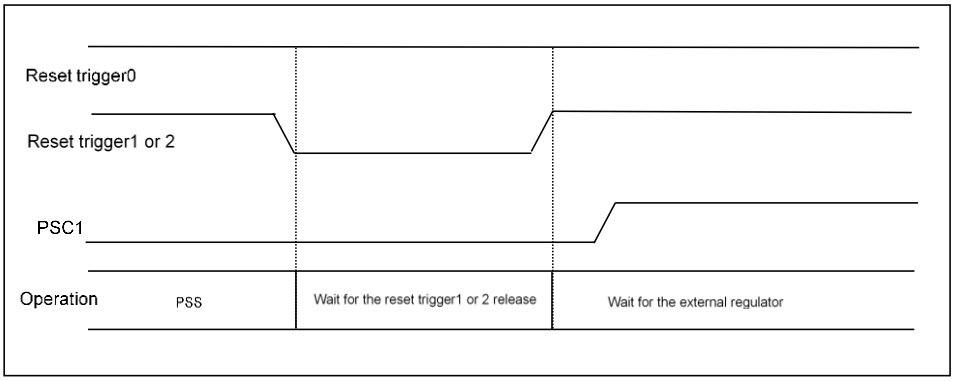
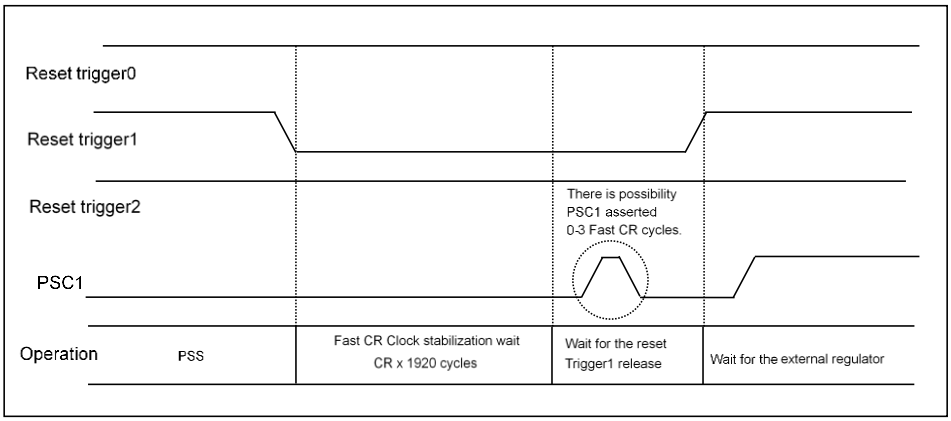
3. Operational Description</td><td>Added the below:

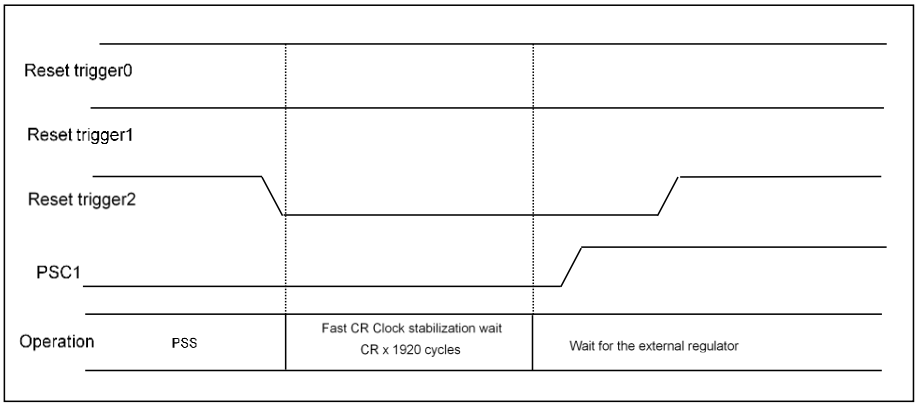
3.2.3. Reset of External Power Supply Control
There are some cases 1.2V external power supply control to be high.

Table 3-32 Behavior of External Power Supply Control
<table>
<tr><th>Condition</th><th>Reset trigger0</th><th>Reset trigger1</th><th>Reset trigger2</th><th>1.2V external power supply control</th></tr>
<tr><td>In power on sequence, Reset trigger1 and Reset trigger2 release during Fast-CR stabilization time</td><td>1</td><td>x</td><td>x</td><td>Case0</td></tr>
<tr><td rowspan="3">In power on sequence, Reset trigger1, Reset trigger2 release after Fast-CR stabilization time</td><td>0</td><td>1</td><td>x</td><td>Case1</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>Case2</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>Case3</td></tr>
<tr><td rowspan="2">In PSS mode with Fast-CR oscillating, Reset trigger0, Reset trigger1, Reset trigger2 occur</td><td>1</td><td>x</td><td>x</td><td>Case0</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>Case4</td></tr>
<tr><td rowspan="3">In PSS mode with Fast-CR stopping, Reset trigger0, Reset trigger1, Reset trigger2 occur</td><td>1</td><td>x</td><td>x</td><td>Case0</td></tr>
<tr><td>0</td><td>1</td><td>x</td><td>Case5</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>Case6</td></tr>
</table>
- Reset trigger0:
Power-on reset, RAM retention low-voltage detection reset, Internal power supply low-voltage detection reset,
Extended internal power supply low-voltage detection reset, INITX, Illegal mode detection reset,
Hardware watchdog reset
- Reset trigger1:
External power supply low-voltage detection reset
- Reset trigger2:
RSTX

1: reset active
0: reset inactive
x: reset inactive or reset inactive before completion of Fast-CR stabilization wait</td></tr>
</table>

Page	Section	Change Results
118 to 119	CHAPTER 4: Reset 3. Operational Description	Added the below Case0) In power on sequence/PSS, Reset trigger1 and 2 release during Fast-CR stabilization time  Case1) In power on sequence, Reset trigger1 release after Fast-CR stabilization time  Case2) In power on sequence, Reset trigger2 release after Fast-CR stabilization time 

Page	Section	Change Results
119 to 120	CHAPTER 4: Reset 3. Operational Description	Added the below Case3) In power on sequence, Reset trigger1 and Reset trigger2 release after Fast-CR stabilization time  Case4) In PSS mode with Fast-CR oscillating, Reset trigger1 or Reset trigger2 occur  Case5) In PSS mode with Fast-CR stopped, Reset trigger1 occur and release after Fast-CR stabilization time 

Page	Section	Change Results
120	CHAPTER 4: Reset 3. Operational Description	Added the below Case6) In PSS mode with Fast-CR stopped, Reset trigger2 occur and release after Fast-CR stabilization time 
260	CHAPTER 6:Low Power Consumption 4. Operation Procedure	Added the below 4.2. Transition from RUN to PSS *Notes:* − *When returning from PSS mode to Run mode, Main_OSC must be valid. Therefore, be sure to* *enable Main_OSC setting of the RUN mode profile before entering the PSS mode.*
293 294	CHAPTER 6:Low Power Consumption 5. Registers	5.2.5. RUN Clock Divider Register (SYSC0_RUNCKDIVR) [bit11:8] MCUCPDIV: MCUconfig APB Clock Divider Setting Bits [bit4:0] MCUCHDIV: MCUconfig AHB Clock Divider Setting Bits Correct) Added the below: ***Note:*** – *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>433</td><td>CHAPTER 6:Low Power Consumption

5.8. Debug Register Group (SYSC0)</td><td>
Changed the Below:
5.8.2. JTAG Setting Register (SYSC0_JTAGCNFG)
Error)
<table>
<tr><td>bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td></tr>
<tr><td>Field</td><td colspan="6">Reserved</td></tr>
<tr><td>R/W Attribute</td><td colspan="6">R0,WX</td></tr>
<tr><td>Protection Attribute</td><td colspan="6">-</td></tr>
<tr><td>Initial Value</td><td colspan="6">0000000</td></tr>
</table>
[bit0] DBGDONE: Debugger Status Bit
This bit indicates the debugger setting status.
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>The debugger is connected.</td></tr>
<tr><td>1</td><td>The debugger is not connected, or debugger setting has completed.</td></tr>
</table>
Correct)
<table>
<tr><td>bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td></tr>
<tr><td>Field</td><td colspan="6">Reserved</td></tr>
<tr><td>R/W Attribute</td><td colspan="6">R0,WX</td></tr>
<tr><td>Protection Attribute</td><td colspan="6">-</td></tr>
<tr><td>Initial Value</td><td colspan="6">0000000</td></tr>
</table>
[bit0] Reserved: Reserved Bit
Do not write any data to this bit.
</td></tr>
<tr><td>452
453</td><td>CHAPTER 6:Low Power Consumption

5. Registers</td><td>
5.11.1. RUN Clock Selection Register 0 (SYSC1_RUNCKSELR0)
[bit20] LAPP1ACSL：LAPP1A Clock Selection Bit
[bit16] LAPP0ACSL：LAPP0A Clock Selection Bit
[bit12] LCP1ACSL：LCP1A Clock Selection Bit
[bit8] LCP0ACSL：LCP0A Clock Selection Bit
Correct)
Added the below:
Note:
– When it is attempted to change the clock selection by this bit, both PLL0 clock and CD0 clock must be supplied.
</td></tr>
</table>

Page	Section	Change Results
473 474 475	CHAPTER 6:Low Power Consumption 5. Registers	5.11.7. RUN Clock Divider Register 0 (SYSC1_RUNCKDIVR0) [bit26:24] HPMDIV: HPM Clock Divider Setting Bits [bit20:16] TRCDIV: TRC Clock Divider Setting Bits [bit13:12] DBGDIV: DBG Clock Divider Setting Bits [bit9:8] ATBDIV [1:0]: ATB Clock Divider Setting Bits [bit4:0] SYSDIV: SYS Clock Divider Setting Bits Correct) Added the below: ***Note:*** – *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*
476 477 478 479	CHAPTER 6:Low Power Consumption 5. Registers	5.11.8. RUN Clock Divider Register 1 (SYSC1_RUNCKDIVR1) [bit31:28] HAPP1B1DIV: HAPP1B1 Clock Divider Setting Bits [bit27:24] HAPP1B0DIV: HAPP1B0 Clock Divider Setting Bits [bit23:20] HAPP0A1DIV: HAPP0A1 Clock Divider Setting Bits [bit19:16] HAPP0A0DIV: HAPP0A0 Clock Divider Setting Bits [bit7:4] SYSC1DIV: SYSC1 Clock Divider Setting Bits [bit2:0] EXTBUSDIV: EXTBUS Clock Divider Setting Bits Correct) Added the below: ***Note:*** – *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*
480 481 482	CHAPTER 6:Low Power Consumption 5. Registers	5.11.9. RUN Clock Divider Register 2 (SYSC1_RUNCKDIVR2) [bit27:24] LAPP1DIV: LAPP1 Clock Divider Setting Bits [bit19:16] LAPP0DIV: LAPP0 Clock Divider Setting Bits [bit11:8] LCP1DIV: LCP1 Clock Divider Setting Bits [bit7:4] LCP0DIV: LCP0 Clock Divider Setting Bits [bit1:0] LCPDIV: LCP Clock Divider Setting Bits Correct) Added the below: ***Note:*** – *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

Page	Section	Change Results
486	CHAPTER 6:Low Power Consumption 5. Registers	5.11.11. RUN Clock Divider Register 4 (SYSC1_RUNCKDIVR4) [bit4:0] HSSPIDIV: HSSPI Clock Divider Setting Bits Correct) Added the below: ***Note:*** – *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*
487 488 489	CHAPTER 6:Low Power Consumption 5. Registers	5.11.12. RUN Clock Divider Register 5 (SYSC1_RUNCKDIVR5) [bit23:20] CD1B1DIV: CD1B1 Clock Divider Setting Bits [bit19:16] CD1B0DIV: CD1B0 Clock Divider Setting Bits [bit15:12] CD1A1DIV: CD1A1 Clock Divider Setting Bits [bit11:8] CD1A0DIV: CD1A0 Clock Divider Setting Bits [bit4:0] CD1DIV: CD1 Clock Divider Setting Bits Correct) Added the below: ***Note:*** – *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*
490 491 492	CHAPTER 6:Low Power Consumption 5. Registers	5.11.13. RUN Clock Divider Register 6 (SYSC1_RUNCKDIVR6) [bit23:20] CD2B1DIV: CD2B1 Clock Divider Setting Bits [bit19:16] CD2B0DIV: CD2B0 Clock Divider Setting Bits [bit15:12] CD2A1DIV: CD2A1 Clock Divider Setting Bits [bit11:8] CD2A0DIV: CD2A0 Clock Divider Setting Bits [bit4:0] CD2DIV: CD2 Clock Divider Setting Bits Correct) Added the below: ***Note:*** – *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

Page	Section	Change Results
493 494 495	CHAPTER 6:Low Power Consumption 5. Registers	5.11.14. RUN Clock Divider Register 7 (SYSC1_RUNCKDIVR7) [bit23:20] CD3B1DIV: CD3B1 Clock Divider Setting Bits [bit19:16] CD3B0DIV: CD3B0 Clock Divider Setting Bits [bit15:12] CD3A1DIV: CD3A1 Clock Divider Setting Bits [bit11:8] CD3A0DIV: CD3A0 Clock Divider Setting Bits [bit4:0] CD3DIV: CD3 Clock Divider Setting Bits Correct) Added the below: ***Note:*** – *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*
496 497 498	CHAPTER 6:Low Power Consumption 5. Registers	5.11.15. RUN Clock Divider Register 8 (SYSC1_RUNCKDIVR8) [bit23:20] CD4B1DIV: CD4B1 Clock Divider Setting Bits [bit19:16] CD4B0DIV: CD4B0 Clock Divider Setting Bits [bit15:12] CD4A1DIV: CD4A1 Clock Divider Setting Bits [bit11:8] CD4A0DIV: CD4A0 Clock Divider Setting Bits [bit4:0] CD4DIV: CD4 Clock Divider Setting Bits Correct) Added the below: ***Note:*** – *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*
499 500 501	CHAPTER 6:Low Power Consumption 5. Registers	5.11.16. RUN Clock Divider Register 9 (SYSC1_RUNCKDIVR9) [bit23:20] CD5B1DIV: CD5B1 Clock Divider Setting Bits [bit19:16] CD5B0DIV: CD5B0 Clock Divider Setting Bits [bit15:12] CD5A1DIV: CD5A1 Clock Divider Setting Bits [bit11:8] CD5A0DIV: CD5A0 Clock Divider Setting Bits [bit4:0] CD5DIV: CD5 Clock Divider Setting Bits Correct) Added the below: ***Note:*** – *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

Page	Section	Change Results
505 506	CHAPTER 6:Low Power Consumption 5. Registers	5.12.1. PSS Clock Selection Register 0 (SYSC1_PSSCKSELR0) [bit20] LAPP1ACSL: LAPP1A Clock Selection Bit [bit16] LAPP0ACSL: LAPP0A Clock Selection Bit [bit12] LCP1ACSL: LCP1A Clock Selection Bit [bit8] LCP0ACSL: LCP0A Clock Selection Bit Correct) Added the below: ***Note:*** – *When it is attempted to change the clock selection by this bit, both PLL0 clock and CD0 clock must be supplied.* 5.12.1. PSS Clock Selection Register 0 (SYSC1_PSSCKSELR0) [bit2:0] CD0CSL: Clock domain 0 clock selection bits Correct) Added the below: ***Notes:*** – *To use the source clock selection "Clock fixed at "L"", following bit must be also set to '0'.* *SYSC1_PSSCKER0:ENCLKLAPP1A* *SYSC1_PSSCKER0:ENCLKLAPP0A* *SYSC1_PSSCKER0:ENCLKLCP1A* *SYSC1_PSSCKER0:ENCLKLCP0A* *Please refer to section 5.12.4 for the definition of the bits*
524 525 526	CHAPTER 6:Low Power Consumption 5. Registers	5.12.7. PSS Clock Divider Register 0 (SYSC1_PSSCKDIVR0) [bit26:24] HPMDIV: HPM Clock Divider Setting Bits [bit20:16] TRCDIV: TRC Clock Divider Setting Bits [bit13:12] DBGDIV: DBG Clock Divider Setting Bits [bit9:8] ATBDIV [1:0]: ATB Clock Divider Setting Bits [bit4:0] SYSDIV: SYS Clock Divider Setting Bits Correct) Added the below: ***Note:*** – *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*
535	CHAPTER 6:Low Power Consumption 5. Registers	5.12.11. PSS Clock Divider Register 4 (SYSC1_PSSCKDIVR4) [bit4:0] HSSPIDIV: HSSPI Clock Divider Setting Bits Correct) Added the below: ***Note:*** – *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

Page	Section	Change Results
536 537 538	CHAPTER 6:Low Power Consumption 5. Registers	5.12.12. PSS Clock Divider Register 5 (SYSC1_PSSCKDIVR5) [bit23:20] CD1B1DIV: CD1B1 Clock Divider Setting Bits [bit19:16] CD1B0DIV: CD1B0 Clock Divider Setting Bits [bit15:12] CD1A1DIV: CD1A1 Clock Divider Setting Bits [bit11:8] CD1A0DIV: CD1A0 Clock Divider Setting Bits [bit4:0] CD1DIV: CD1 Clock Divider Setting Bits Correct) Added the below: ***Note:*** – *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*
539 540 541	CHAPTER 6:Low Power Consumption 5. Registers	5.12.13. PSS Clock Divider Register 5 (SYSC1_PSSCKDIVR5) [bit23:20] CD2B1DIV: CD2B1 Clock Divider Setting Bits [bit19:16] CD2B0DIV: CD2B0 Clock Divider Setting Bits [bit15:12] CD2A1DIV: CD2A1 Clock Divider Setting Bits [bit11:8] CD2A0DIV: CD2A0 Clock Divider Setting Bits [bit4:0] CD2DIV: CD2 Clock Divider Setting Bits Correct) Added the below: ***Note:*** – *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*
542 543 544	CHAPTER 6:Low Power Consumption 5. Registers	5.12.14. PSS Clock Divider Register 5 (SYSC1_PSSCKDIVR5) [bit23:20] CD3B1DIV: CD3B1 Clock Divider Setting Bits [bit19:16] CD3B0DIV: CD3B0 Clock Divider Setting Bits [bit15:12] CD3A1DIV: CD3A1 Clock Divider Setting Bits [bit11:8] CD3A0DIV: CD3A0 Clock Divider Setting Bits [bit4:0] CD3DIV: CD3 Clock Divider Setting Bits Correct) Added the below: ***Note:*** – *When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.*

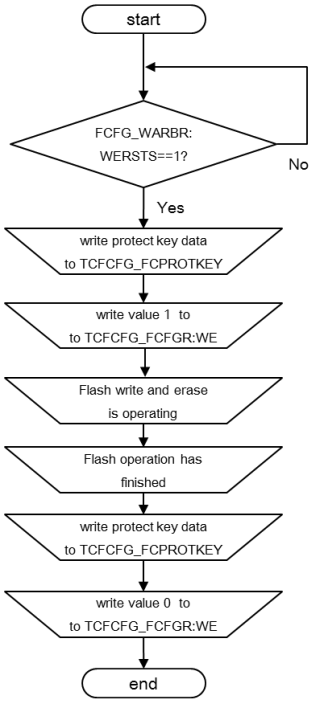
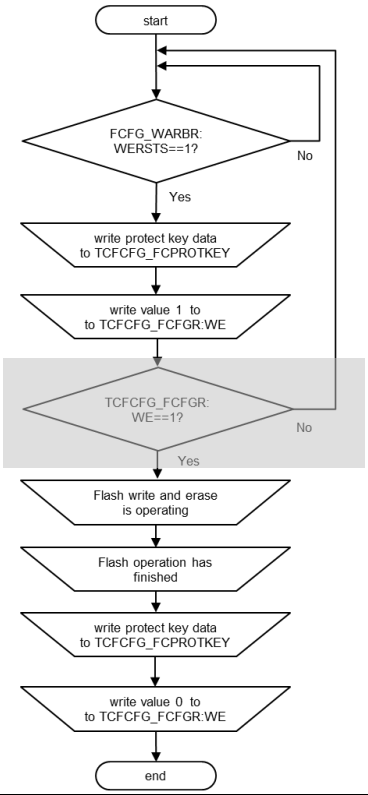
<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>545
546
547</td><td>CHAPTER 6:Low Power Consumption

5. Registers</td><td>5.12.15. PSS Clock Divider Register 5 (SYSC1_PSSCKDIVR5)
[bit23:20] CD4B1DIV: CD4B1 Clock Divider Setting Bits
[bit19:16] CD4B0DIV: CD4B0 Clock Divider Setting Bits
[bit15:12] CD4A1DIV: CD4A1 Clock Divider Setting Bits
[bit11:8] CD4A0DIV: CD4A0 Clock Divider Setting Bits
[bit4:0] CD4DIV: CD4 Clock Divider Setting Bits
Correct)
Added the below:
Note:
– When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.</td></tr>
<tr><td>548
549
550</td><td>CHAPTER 6:Low Power Consumption

5. Registers</td><td>5.12.16. PSS Clock Divider Register 5 (SYSC1_PSSCKDIVR5)
[bit23:20] CD5B1DIV: CD5B1 Clock Divider Setting Bits
[bit19:16] CD5B0DIV: CD5B0 Clock Divider Setting Bits
[bit15:12] CD5A1DIV: CD5A1 Clock Divider Setting Bits
[bit11:8] CD5A0DIV: CD5A0 Clock Divider Setting Bits
[bit4:0] CD5DIV: CD5 Clock Divider Setting Bits
Correct)
Added the below:
Note:
– When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.</td></tr>
<tr><td>656</td><td>CHAPTER 6:Low Power Consumption

6. Other</td><td>Correct)
Added the below:
6.3. Restriction to use “Clock fixed at L” and changing clock division ratio</td></tr>
<tr><td>683</td><td>CHAPTER 8:Clock Supervisor

5. Registers</td><td>Changed the Below:
Table 5-2 Clock Supervisor Register Map (B/B)
Error)
<table><tr><td>0x0000_0058 to 0x0000_005C</td><td>Reserved
00000000_00000000_00000000_00000000</td></tr></table>
Correct)
<table><tr><td>0x0000_0058 to 0x0000_005C</td><td>Reserved</td></tr></table></td></tr>
</table>

Page	Section	Change Results
719	CHAPTER 9:Source Clock Timer 5. Registers	Correct) Added the below: ***Notes:*** – *The registers are protected by protection key setting register (SYSC0_PROTKEYR). For details on the protection, see the following chapter: "Low-power Consumption."* – *Access to reserved address results bus error response.*
1078	CHAPTER 17:TCFLASH 4.3. Enable Programming	Changed the Below **Figure 4-1 Procedure to Set Flash Memory Program Enable Bit** Error)  Correct) 

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1361</td><td>CHAPTER 24:Security

2. Security Scope and Access Restriction by the Security</td><td>
Changed the Below:
2.2. Access Restriction by the Security
Access Restriction in Non-USERMODE

Error)
<table>
<tr><th>Operation</th><th>Non-USERMODE Security=Disabled</th><th>Non-USERMODE Security= Flash Protection</th><th>Non-USERMODE Security= Device Protection</th></tr>
<tr><td>TCFlash Macro Erase</td><td>Enabled</td><td>Restricted *1</td><td rowspan="8">Restricted *2</td></tr>
<tr><td>TCFlash Sector Erase</td><td>Enabled</td><td>Disabled</td></tr>
<tr><td>WorkFlash Sector Erase</td><td>Enabled</td><td>Restricted *1</td></tr>
<tr><td>Programming</td><td>Enabled</td><td>Restricted *1</td></tr>
<tr><td>Reading</td><td>Enabled</td><td>Disabled</td></tr>
<tr><td>Use of Serial Programmer</td><td>Enabled</td><td>Restricted *1</td></tr>
<tr><td>Use of Parallel Programmer</td><td>Enabled</td><td>Restricted *1</td></tr>
<tr><td>Debugger connection</td><td>Enabled</td><td>Enabled *3</td></tr>
</table>
*1: It has to follow the steps shown in Erase Order Restriction in Non-USERMODE to Devices with Flash Protection or Device Protection.
*2: Only a certain sequence of erase operations (full-chip erase) can be triggered by the writer equipment (Serial or JTAG).
*3: Access to Flash is restricted (only full-chip erase is possible)

Correct)
<table>
<tr><th>Operation</th><th>Non-USERMODE Security=Disabled</th><th>Non-USERMODE Security= Flash Protection Chip Erase Enabled = Enabled</th><th>Non-USERMODE Security= Flash Protection Chip Erase Enabled = Disabled</th><th>Non-USERMODE Security= Device Protection Chip Erase Enabled = Enabled</th><th>Non-USERMODE Security= Device Protection Chip Erase Enabled = Disabled</th></tr>
<tr><td>TCFlash Macro Erase</td><td>Enabled</td><td>Restricted *1</td><td rowspan="7">Disabled</td><td rowspan="8">Restricted *2</td><td rowspan="8">Disabled</td></tr>
<tr><td>TCFlash Sector Erase</td><td>Enabled</td><td>Disabled</td></tr>
<tr><td>WorkFlash Sector Erase</td><td>Enabled</td><td>Restricted *1</td></tr>
<tr><td>Programming</td><td>Enabled</td><td>Restricted *1</td></tr>
<tr><td>Reading</td><td>Enabled</td><td>Disabled</td></tr>
<tr><td>Use of Serial Programmer</td><td>Enabled</td><td>Restricted *1</td></tr>
<tr><td>Use of Parallel Programmer</td><td>Enabled</td><td>Restricted *1</td></tr>
<tr><td>Debugger connection</td><td>Enabled</td><td>Enabled *3</td><td>Enabled*4</td></tr>
</table>
*1: It has to follow the steps shown in Erase Order Restriction in Non-USERMODE to Devices with Flash Protection or Device Protection.
*2: Only a certain sequence of erase operations (full-chip erase) can be triggered by the writer equipment (Serial or JTAG).
*3: Access to Flash is restricted (only full-chip erase is possible).
*4: Access to Flash is restricted (Nothing can be done to Flash).
</td></tr>
</table>

Page	Section	Change Results
1585	CHAPTER 27:DMA Controller 4. Registers	Correct) Added the below: 4.8. DMA Controller Channel Configuration B Register Channel 'n' (DMAi_Bn) [bit15:12] SP[3:0] : Source protection ***Note:*** – *Configuration of SP[3] and SP[2] has no effect.* [bit11:8] DP[3:0] : Destination protection ***Note:*** – *Configuration of DP[3] has an effect.* – *DP[3]=0: Posted write buffer in HPM is used. Due to the posted write behavior, DMA cannot trap the error returned from the destination even if there is an error in the write access (e.g.: The destination register/memory requires Privilege level for writing, but DP[1] is "0").* – *DP[3]=0: Posted write buffer in HPM is unused. DMA always traps the error when there is an error in the write access. (DMAi_Bn:EQ)* – *Configuration of DP[2] has no effect.*

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1697</td><td>CHAPTER 30: CAN FD Controller(MCAN3.0.1)
3. Explanation of Operations
3.5. Tx Handling</td><td>Added the below:

3.5.5. Mixed Dedicated Tx Buffers/Tx FIFO
Note:
− Mixed Dedicated Tx Buffers/Tx FIFO isn't supported. Don't use combination Dedicated Tx Buffers and Tx FIFO.</td></tr>
<tr><td>1729</td><td>CHAPTER 30: CAN FD Controller(MCAN3.0.1)
5. Registers</td><td>Changed the below:

Table 5-2 Register Map CAN FD CONTROLLER
MCU Config Group (Channel No.: 0, 1, and 2)
Error)
<table>
<tr><th rowspan="2">Offset</th><th colspan="4">Register Name / Initial Value</th><th rowspan="2">Group Name</th></tr>
<tr><th>+3</th><th>+2</th><th>+1</th><th>+0</th></tr>
<tr><td>0000_0000</td><td colspan="4">MCG_CANFDx_CREL
00110010 00000100 00010010 00011000</td><td rowspan="28">MCU Config</td></tr>
<tr><td>0000_0004</td><td colspan="4">MCG_CANFDx_ENDN
10000111 01100101 01000011 00100001</td></tr>
<tr><td>0000_0008</td><td colspan="4">-</td></tr>
<tr><td>0000_000C</td><td colspan="4">MCG_CANFDx_DBTP
00000000 00000000 00001010 00110011</td></tr>
<tr><td>0000_0010</td><td colspan="4">MCG_CANFDx_TEST
00000000 00000000 00000000 *0000000</td></tr>
<tr><td>0000_0014</td><td colspan="4">MCG_CANFDx_RWD
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_0018</td><td colspan="4">MCG_CANFDx_CCCR
00000000 00000000 00000000 00000001</td></tr>
<tr><td>0000_001C</td><td colspan="4">MCG_CANFDx_NBTP
00000110 00000000 00001010 00000011</td></tr>
<tr><td>0000_0020</td><td colspan="4">MCG_CANFDx_TSCC
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_0024</td><td colspan="4">MCG_CANFDx_TSCV
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_0028</td><td colspan="4">MCG_CANFDx_TOCC
11111111 11111111 00000000 00000000</td></tr>
<tr><td>0000_002C</td><td colspan="4">MCG_CANFDx_TOCV
00000000 00000000 11111111 11111111</td></tr>
<tr><td>0000_0030
|
0000_003C</td><td colspan="4">-</td></tr>
<tr><td>0000_0040</td><td colspan="4">MCG_CANFDx_ECR
XXXXXXXX 00000000 00000000 00000000</td></tr>
<tr><td>0000_0044</td><td colspan="4">MCG_CANFDx_PSR
XXXXXXXX XXXXXXXX X0000111 00000111</td></tr>
<tr><td>0000_0048</td><td colspan="4">MCG_CANFDx_TDCR
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_004C</td><td colspan="4">-</td></tr>
<tr><td>0000_0050</td><td colspan="4">MCG_CANFDx_IR
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_0054</td><td colspan="4">MCG_CANFDx_IE
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_0058</td><td colspan="4">MCG_CANFDx_ILS
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_005C</td><td colspan="4">MCG_CANFDx_ILE
00000000 00000000 00000000 00000000</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1729</td><td>CHAPTER 30: CAN FD Controller(MCAN3.0.1)
5. Registers</td><td>Correct)
<table>
<tr><th rowspan="2">Offset</th><th colspan="4">Register Name / Initial Value</th></tr>
<tr><th>+3</th><th>+2</th><th>+1</th><th>+0</th></tr>
<tr><td>0000_0000</td><td colspan="4">MCG_CANFDx_CREL
00110010 00000100 00010010 00011000</td></tr>
<tr><td>0000_0004</td><td colspan="4">MCG_CANFDx_ENDN
10000111 01100101 01000011 00100001</td></tr>
<tr><td>0000_0008</td><td colspan="4">-</td></tr>
<tr><td>0000_000C</td><td colspan="4">MCG_CANFDx_DBTP
00000000 00000000 00001010 00110011</td></tr>
<tr><td>0000_0010</td><td colspan="4">MCG_CANFDx_TEST
00000000 00000000 00000000 *0000000</td></tr>
<tr><td>0000_0014</td><td colspan="4">MCG_CANFDx_RWD
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_0018</td><td colspan="4">MCG_CANFDx_CCCR
00000000 00000000 00000000 00000001</td></tr>
<tr><td>0000_001C</td><td colspan="4">MCG_CANFDx_NBTP
00000110 00000000 00001010 00000011</td></tr>
<tr><td>0000_0020</td><td colspan="4">MCG_CANFDx_TSCC
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_0024</td><td colspan="4">MCG_CANFDx_TSCV
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_0028</td><td colspan="4">MCG_CANFDx_TOCC
11111111 11111111 00000000 00000000</td></tr>
<tr><td>0000_002C</td><td colspan="4">MCG_CANFDx_TOCV
00000000 00000000 11111111 11111111</td></tr>
<tr><td>0000_0030
|
0000_003C</td><td colspan="4">-</td></tr>
<tr><td>0000_0040</td><td colspan="4">MCG_CANFDx_ECR
XXXXXXXX 00000000 00000000 00000000</td></tr>
<tr><td>0000_0044</td><td colspan="4">MCG_CANFDx_PSR
XXXXXXXX X0000000 X0000111 00000111</td></tr>
<tr><td>0000_0048</td><td colspan="4">MCG_CANFDx_TDCR
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_004C</td><td colspan="4">-</td></tr>
<tr><td>0000_0050</td><td colspan="4">MCG_CANFDx_IR
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_0054</td><td colspan="4">MCG_CANFDx_IE
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_0058</td><td colspan="4">MCG_CANFDx_ILS
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_005C</td><td colspan="4">MCG_CANFDx_ILE
00000000 00000000 00000000 00000000</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1730</td><td>CHAPTER 30: CAN FD Controller(MCAN3.0.1)
5. Registers</td><td>Changed the below:

Table 5-2 Register Map CAN FD CONTROLLER
MCU Config Group (Channel No.: 0, 1, and 2)

Error)
<table>
<tr><th rowspan="2">Offset</th><th colspan="4">Register Name / Initial Value</th><th rowspan="2">Group Name</th></tr>
<tr><th>+3</th><th>+2</th><th>+1</th><th>+0</th></tr>
<tr><td>0000_0060

0000_007C</td><td colspan="4">-</td><td rowspan="24">MCU Config</td></tr>
<tr><td>0000_0080</td><td colspan="4">MCG_CANFDx_GFC
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_0084</td><td colspan="4">MCG_CANFDx_SIDFC
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_0088</td><td colspan="4">MCG_CANFDx_XIDFC
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_008C</td><td colspan="4">-</td></tr>
<tr><td>0000_0090</td><td colspan="4">MCG_CANFDx_XIDAM
00011111 11111111 11111111 11111111</td></tr>
<tr><td>0000_0094</td><td colspan="4">MCG_CANFDx_HPMS
XXXXXXXX XXXXXXXX 00000000 00000000</td></tr>
<tr><td>0000_0098</td><td colspan="4">MCG_CANFDx_NDAT1
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_009C</td><td colspan="4">MCG_CANFDx_NDAT2
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_00A0</td><td colspan="4">MCG_CANFDx_RXF0C
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_00A4</td><td colspan="4">MCG_CANFDx_RXF0S
XXXXXX00 XX000000 XX000000 X0000000</td></tr>
<tr><td>0000_00A8</td><td colspan="4">MCG_CANFDx_RXF0A
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_00AC</td><td colspan="4">MCG_CANFDx_RXBC
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_00B0</td><td colspan="4">MCG_CANFDx_RXF1C
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_00B4</td><td colspan="4">MCG_CANFDx_RXF1S
00XXXX00 XX000000 XX000000 X0000000</td></tr>
<tr><td>0000_00B8</td><td colspan="4">MCG_CANFDx_RXF1A
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_00BC</td><td colspan="4">MCG_CANFDx_RXESC
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_00C0</td><td colspan="4">MCG_CANFDx_TXBC
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_00C4</td><td colspan="4">MCG_CANFDx_TXFQS
XXXXXXXX XX000000 XXX00000 XX000000</td></tr>
<tr><td>0000_00C8</td><td colspan="4">MCG_CANFDx_TXESC
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_00CC</td><td colspan="4">MCG_CANFDx_TXBRP
00000000 00000000 00000000 00000000</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1730</td><td>C CHAPTER 30: CAN FD Controller(MCAN3.0.1)
5. Registers</td><td>Correct)
<table>
<tr><th rowspan="2">Offset</th><th colspan="4">Register Name / Initial Value</th></tr>
<tr><th>+3</th><th>+2</th><th>+1</th><th>+0</th></tr>
<tr><td>0000_0060

0000_007C</td><td colspan="4">-</td></tr>
<tr><td>0000_0080</td><td colspan="4">MCG_CANFDx_GFC
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_0084</td><td colspan="4">MCG_CANFDx_SIDFC
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_0088</td><td colspan="4">MCG_CANFDx_XIDFC
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_008C</td><td colspan="4">-</td></tr>
<tr><td>0000_0090</td><td colspan="4">MCG_CANFDx_XIDAM
00011111 11111111 11111111 11111111</td></tr>
<tr><td>0000_0094</td><td colspan="4">MCG_CANFDx_HPMS
XXXXXXXX XXXXXXXX 00000000 00000000</td></tr>
<tr><td>0000_0098</td><td colspan="4">MCG_CANFDx_NDAT1
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_009C</td><td colspan="4">MCG_CANFDx_NDAT2
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_00A0</td><td colspan="4">MCG_CANFDx_RXF0C
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_00A4</td><td colspan="4">MCG_CANFDx_RXF0S
XXXXXX00 XX000000 XX000000 X0000000</td></tr>
<tr><td>0000_00A8</td><td colspan="4">MCG_CANFDx_RXF0A
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_00AC</td><td colspan="4">MCG_CANFDx_RXBC
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_00B0</td><td colspan="4">MCG_CANFDx_RXF1C
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_00B4</td><td colspan="4">MCG_CANFDx_RXF1S
00XXXX00 XX000000 XX000000 X0000000</td></tr>
<tr><td>0000_00B8</td><td colspan="4">MCG_CANFDx_RXF1A
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_00BC</td><td colspan="4">MCG_CANFDx_RXESC
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_00C0</td><td colspan="4">MCG_CANFDx_TXBC
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_00C4</td><td colspan="4">MCG_CANFDx_TXFQS
XXXXXXXX XX000000 XXX00000 XX000000</td></tr>
<tr><td>0000_00C8</td><td colspan="4">MCG_CANFDx_TXESC
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_00CC</td><td colspan="4">MCG_CANFDx_TXBRP
00000000 00000000 00000000 00000000</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1731</td><td>CHAPTER 30: CAN FD Controller(MCAN3.0.1)
5. Registers</td><td>
Changed the below:

Table 5-2 Register Map CAN FD CONTROLLER
MCU Config Group (Channel No.: 0, 1, and 2)

Error)
<table>
<tr><th rowspan="2">Offset</th><th colspan="4">Register Name / Initial Value</th><th rowspan="2">Group Name</th></tr>
<tr><th>+3</th><th>+2</th><th>+1</th><th>+0</th></tr>
<tr><td>0000_00D0</td><td colspan="4">MCG_CANFDx_TXBAR
00000000 00000000 00000000 00000000</td><td></td></tr>
<tr><td>0000_00D4</td><td colspan="4">MCG_CANFDx_TXBCR
00000000 00000000 00000000 00000000</td><td rowspan="14">MCU Config</td></tr>
<tr><td>0000_00D8</td><td colspan="4">MCG_CANFDx_TXBTO
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_00DC</td><td colspan="4">MCG_CANFDx_TXBCF
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_00E0</td><td colspan="4">MCG_CANFDx_TXBTIE
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_00E4</td><td colspan="4">MCG_CANFDx_TXBCIE
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_00E8
0000_00EC</td><td colspan="4">-</td></tr>
<tr><td>0000_00F0</td><td colspan="4">MCG_CANFDx_TXEFC
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_00F4</td><td colspan="4">MCG_CANFDx_TXEFS
XXXXXX00 XXX00000 XXX00000 XX000000</td></tr>
<tr><td>0000_00F8</td><td colspan="4">MCG_CANFDx_TXEFA
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_00FC
0000_01FC</td><td colspan="4">-</td></tr>
<tr><td>0000_0200</td><td colspan="2">MCG_CANFDx_FDSEAR
00000000 00000000</td><td>MCG_CANFDx_FDESR
00000000</td><td>MCG_CANFDx_FDECR
00000000</td></tr>
<tr><td>0000_0204</td><td colspan="2">MCG_CANFD0_FDDEAR
00000000 00000000</td><td>MCG_CANFDx_FDESCR
00000000</td><td>-</td></tr>
<tr><td>0000_0208
0000_02FC</td><td colspan="4">-</td></tr>
<tr><td>0000_0300
0000_7FFC</td><td colspan="4">-</td></tr>
<tr><td>0000_8000
0000_BFFC</td><td colspan="4">CAN_RAM ch0</td></tr>
<tr><td>0000_C000
0000_FFFC</td><td colspan="4">-</td></tr>
</table>

Notes:
– x is the channel number. (0 to 2)
– "*": Initial value "0" or "1" according to the setting (designed by higher layer)
– The number of circuits is designed by a higher layer.
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1731</td><td>CHAPTER 30: CAN FD Controller(MCAN3.0.1)
5. Registers</td><td>Correct)
<table>
<tr><th rowspan="2">Offset</th><th colspan="4">Register Name / Initial Value</th></tr>
<tr><th>+3</th><th>+2</th><th>+1</th><th>+0</th></tr>
<tr><td>0000_00D0</td><td colspan="4">MCG_CANFDx_TXBAR
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_00D4</td><td colspan="4">MCG_CANFDx_TXBCR
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_00D8</td><td colspan="4">MCG_CANFDx_TXBTO
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_00DC</td><td colspan="4">MCG_CANFDx_TXBCF
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_00E0</td><td colspan="4">MCG_CANFDx_TXBTIE
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_00E4</td><td colspan="4">MCG_CANFDx_TXBCIE
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_00E8
0000_00EC</td><td colspan="4">-</td></tr>
<tr><td>0000_00F0</td><td colspan="4">MCG_CANFDx_TXEFC
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_00F4</td><td colspan="4">MCG_CANFDx_TXEFS
XXXXXX00 XXX00000 XXX00000 XX000000</td></tr>
<tr><td>0000_00F8</td><td colspan="4">MCG_CANFDx_TXEFA
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_00FC
0000_01FC</td><td colspan="4">-</td></tr>
<tr><td>0000_0200</td><td colspan="2">MCG_CANFDx_FDSEAR
00000000 00000000</td><td>MCG_CANFDx_FDESR
00000000</td><td>MCG_CANFDx_FDECR
00000000</td></tr>
<tr><td>0000_0204</td><td colspan="2">MCG_CANFDx_FDDEAR
00000000 00000000</td><td>MCG_CANFDx_FDESCR
00000000</td><td>-</td></tr>
<tr><td>0000_0208
0000_02FC</td><td colspan="4">-</td></tr>
<tr><td>0000_0300
0000_7FFC</td><td colspan="4">-</td></tr>
<tr><td>0000_8000
0000_BFFC</td><td colspan="4">CAN_RAM chx</td></tr>
<tr><td>0000_C000
0000_FFFC</td><td colspan="4">-</td></tr>
</table>
Notes:
– x is the channel number. (0 to 2)
– "*": Initial value "0" or "1" according to the setting (designed by higher layer)
– The number of circuits is designed by a higher layer.</td></tr>
</table>

Page	Section	Change Results
1732	CHAPTER 30: CAN FD Controller(MCAN3.0.1) 5. Registers	Changed the below: **Table 5-2 Register Map CAN FD CONTROLLER** **Common Peripheral #0 Group (Channel No.: 0, 1, 2, 3, and 4)** Error)

Offset	Register Name / Initial Value				Group Name
	+3	+2	+1	+0	
0000_0000	CPG_CANFDx_CREL 00110010 00000100 00010010 00011000				Common Peripheral #0
0000_0004	CPG_CANFDx_ENDN 10000111 01100101 01000011 00100001				
0000_0008	-				
0000_000C	CPG_CANFDx_DBTP 00000000 00000000 00001010 00110011				
0000_0010	CPG_CANFDx_TEST 00000000 00000000 00000000 *0000000				
0000_0014	CPG_CANFDx_RWD 00000000 00000000 00000000 00000000				
0000_0018	CPG_CANFDx_CCCR 00000000 00000000 00000000 00000001				
0000_001C	CPG_CANFDx_NBTP 00000110 00000000 00001010 00000011				
0000_0020	CPG_CANFDx_TSCC 00000000 00000000 00000000 00000000				
0000_0024	CPG_CANFDx_TSCV 00000000 00000000 00000000 00000000				
0000_0028	CPG_CANFDx_TOCC 11111111 11111111 00000000 00000000				
0000_002C	CPG_CANFDx_TOCV 00000000 00000000 11111111 11111111				
0000_0030 | 0000_003C	-				
0000_0040	CPG_CANFDx_ECR XXXXXXXX 00000000 00000000 00000000				
0000_0044	CPG_CANFDx_PSR XXXXXXXX XXXXXXXX XX000111 00000111				
0000_0048	CPG_CANFDx_TDCR 00000000 00000000 00000000 00000000				
0000_004C	-				
0000_0050	CPG_CANFDx_IR 00000000 00000000 00000000 00000000				
0000_0054	CPG_CANFDx_IE 00000000 00000000 00000000 00000000				
0000_0058	CPG_CANFDx_ILS 00000000 00000000 00000000 00000000				
0000_005C	CPG_CANFDx_ILE 00000000 00000000 00000000 00000000				
0000_0060 0000_007C	-				

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1732</td><td>CHAPTER 30: CAN FD Controller(MCAN3.0.1)
5. Registers</td><td>Correct)
<table>
<tr><th rowspan="2">Offset</th><th colspan="4">Register Name / Initial Value</th></tr>
<tr><th>+3</th><th>+2</th><th>+1</th><th>+0</th></tr>
<tr><td>0000_0000</td><td colspan="4">CPG_CANFDx_CREL
00110010 00000100 00010010 00011000</td></tr>
<tr><td>0000_0004</td><td colspan="4">CPG_CANFDx_ENDN
10000111 01100101 01000011 00100001</td></tr>
<tr><td>0000_0008</td><td colspan="4">-</td></tr>
<tr><td>0000_000C</td><td colspan="4">CPG_CANFDx_DBTP
00000000 00000000 00001010 00110011</td></tr>
<tr><td>0000_0010</td><td colspan="4">CPG_CANFDx_TEST
00000000 00000000 00000000 *0000000</td></tr>
<tr><td>0000_0014</td><td colspan="4">CPG_CANFDx_RWD
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_0018</td><td colspan="4">CPG_CANFDx_CCCR
00000000 00000000 00000000 00000001</td></tr>
<tr><td>0000_001C</td><td colspan="4">CPG_CANFDx_NBTP
00000110 00000000 00001010 00000011</td></tr>
<tr><td>0000_0020</td><td colspan="4">CPG_CANFDx_TSCC
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_0024</td><td colspan="4">CPG_CANFDx_TSCV
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_0028</td><td colspan="4">CPG_CANFDx_TOCC
11111111 11111111 00000000 00000000</td></tr>
<tr><td>0000_002C</td><td colspan="4">CPG_CANFDx_TOCV
00000000 00000000 11111111 11111111</td></tr>
<tr><td>0000_0030
|
0000_003C</td><td colspan="4">-</td></tr>
<tr><td>0000_0040</td><td colspan="4">CPG_CANFDx_ECR
XXXXXXXX 00000000 00000000 00000000</td></tr>
<tr><td>0000_0044</td><td colspan="4">CPG_CANFDx_PSR
XXXXXXXX X0000000 X0000111 00000111</td></tr>
<tr><td>0000_0048</td><td colspan="4">CPG_CANFDx_TDCR
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_004C</td><td colspan="4">-</td></tr>
<tr><td>0000_0050</td><td colspan="4">CPG_CANFDx_IR
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_0054</td><td colspan="4">CPG_CANFDx_IE
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_0058</td><td colspan="4">CPG_CANFDx_ILS
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_005C</td><td colspan="4">CPG_CANFDx_ILE
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_0060

0000_007C</td><td colspan="4">-</td></tr>
</table></td></tr>
</table>

Page	Section	Change Results
1733	CHAPTER 30: CAN FD Controller(MCAN3.0.1) 5. Registers	Changed the below: **Table 5-2 Register Map CAN FD CONTROLLER** **Common Peripheral #0 Group (Channel No.: 0, 1, 2, 3, and 4)** Error)

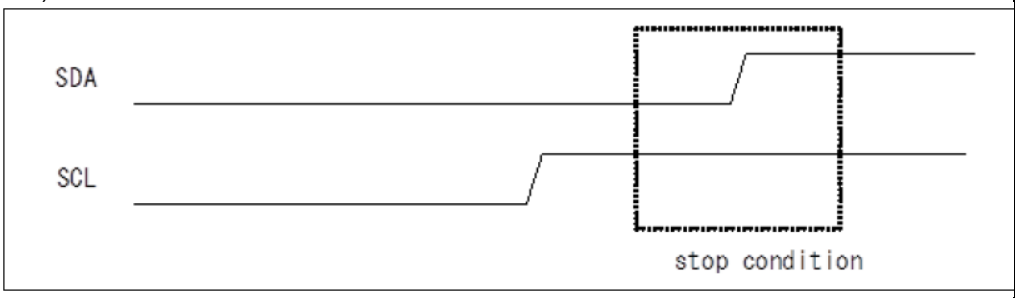
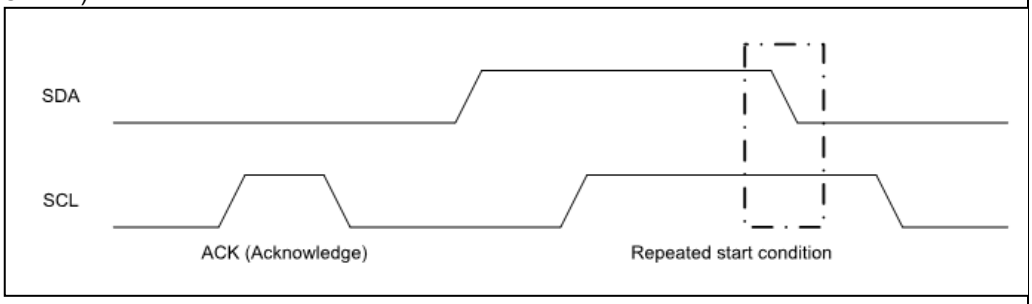
Offset	Register Name / Initial Value				
	+3	+2	+1	+0	
0000_0080	CPG_CANFDx_GFC 00000000 00000000 00000000 00000000				Common Peripheral #0
0000_0084	CPG_CANFDx_SIDFC 00000000 00000000 00000000 00000000				
0000_0088	CPG_CANFDx_XIDFC 00000000 00000000 00000000 00000000				
0000_008C	-				
0000_0090	CPG_CANFDx_XIDAM 00011111 11111111 11111111 11111111				
0000_0094	CPG_CANFDx_HPMS XXXXXXXX XXXXXXXX 00000000 00000000				
0000_0098	CPG_CANFDx_NDAT1 00000000 00000000 00000000 00000000				
0000_009C	CPG_CANFDx_NDAT2 00000000 00000000 00000000 00000000				
0000_00A0	CPG_CANFDx_RXF0C 00000000 00000000 00000000 00000000				
0000_00A4	CPG_CANFDx_RXF0S XXXXXX00 XX000000 XX000000 X0000000				
0000_00A8	CPG_CANFDx_RXF0A 00000000 00000000 00000000 00000000				
0000_00AC	CPG_CANFDx_RXBC 00000000 00000000 00000000 00000000				
0000_00B0	CPG_CANFDx_RXF1C 00000000 00000000 00000000 00000000				
0000_00B4	CPG_CANFDx_RXF1S 00XXXX00 XX000000 XX000000 X0000000				
0000_00B8	CPG_CANFDx_RXF1A 00000000 00000000 00000000 00000000				
0000_00BC	CPG_CANFDx_RXESC 00000000 00000000 00000000 00000000				
0000_00C0	CPG_CANFDx_TXBC 00000000 00000000 00000000 00000000				
0000_00C4	CPG_CANFDx_TXFQS XXXXXXXX XX000000 XXX00000 XX000000				
0000_00C8	CPG_CANFDx_TXESC 00000000 00000000 00000000 00000000				
0000_00CC	CPG_CANFDx_TXBRP 00000000 00000000 00000000 00000000				
0000_00D0	CPG_CANFDx_TXBAR 00000000 00000000 00000000 00000000				
0000_00D4	CPG_CANFDx_TXBCR 00000000 00000000 00000000 00000000				

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1733</td><td>CHAPTER 30: CAN FD Controller(MCAN3.0.1)
5. Registers</td><td>Correct)
<table>
<tr><th rowspan="2">Offset</th><th colspan="4">Register Name / Initial Value</th></tr>
<tr><th>+3</th><th>+2</th><th>+1</th><th>+0</th></tr>
<tr><td>0000_0080</td><td colspan="4">CPG_CANFDx_GFC
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_0084</td><td colspan="4">CPG_CANFDx_SIDFC
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_0088</td><td colspan="4">CPG_CANFDx_XIDFC
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_008C</td><td colspan="4">-</td></tr>
<tr><td>0000_0090</td><td colspan="4">CPG_CANFDx_XIDAM
00011111 11111111 11111111 11111111</td></tr>
<tr><td>0000_0094</td><td colspan="4">CPG_CANFDx_HPMS
XXXXXXXX XXXXXXXX 00000000 00000000</td></tr>
<tr><td>0000_0098</td><td colspan="4">CPG_CANFDx_NDAT1
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_009C</td><td colspan="4">CPG_CANFDx_NDAT2
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_00A0</td><td colspan="4">CPG_CANFDx_RXF0C
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_00A4</td><td colspan="4">CPG_CANFDx_RXF0S
XXXXXX00 XX000000 XX000000 X0000000</td></tr>
<tr><td>0000_00A8</td><td colspan="4">CPG_CANFDx_RXF0A
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_00AC</td><td colspan="4">CPG_CANFDx_RXBC
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_00B0</td><td colspan="4">CPG_CANFDx_RXF1C
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_00B4</td><td colspan="4">CPG_CANFDx_RXF1S
00XXXX00 XX000000 XX000000 X0000000</td></tr>
<tr><td>0000_00B8</td><td colspan="4">CPG_CANFDx_RXF1A
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_00BC</td><td colspan="4">CPG_CANFDx_RXESC
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_00C0</td><td colspan="4">CPG_CANFDx_TXBC
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_00C4</td><td colspan="4">CPG_CANFDx_TXFQS
XXXXXXXX XX000000 XXX00000 XX000000</td></tr>
<tr><td>0000_00C8</td><td colspan="4">CPG_CANFDx_TXESC
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_00CC</td><td colspan="4">CPG_CANFDx_TXBRP
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_00D0</td><td colspan="4">CPG_CANFDx_TXBAR
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_00D4</td><td colspan="4">CPG_CANFDx_TXBCR
00000000 00000000 00000000 00000000</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1734</td><td>CHAPTER 30: CAN FD Controller(MCAN3.0.1)
5. Registers</td><td>Changed the below:

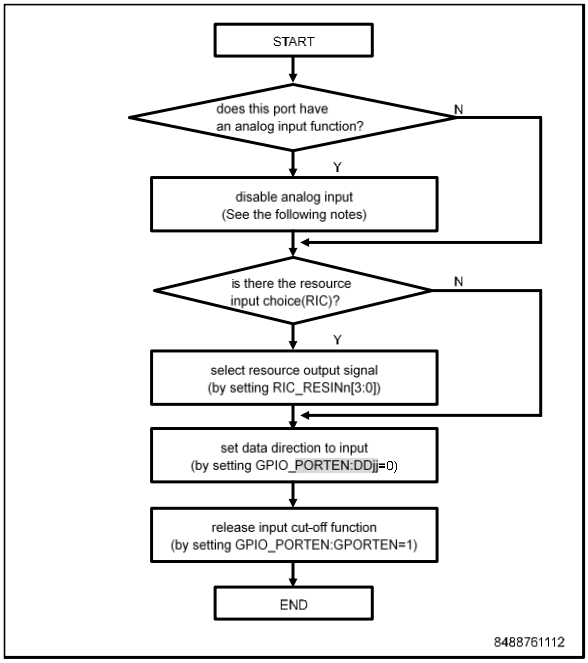
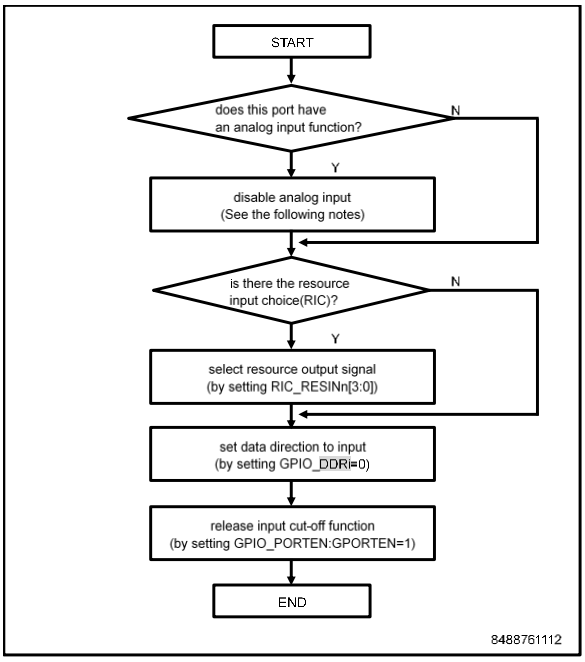
Table 5-3 Register Map CAN FD CONTROLLER
Common Peripheral #0 Group (Channel No.: 0, 1, 2, 3, and 4)
Error)
<table>
<tr><th rowspan="2">Offset</th><th colspan="4">Register Name / Initial Value</th><th rowspan="2"></th></tr>
<tr><th>+3</th><th>+2</th><th>+1</th><th>+0</th></tr>
<tr><td>0000_00D8</td><td colspan="4">CPG_CANFDx_TXBTO
00000000 00000000 00000000 00000000</td><td rowspan="16">Common Peripheral #0</td></tr>
<tr><td>0000_00DC</td><td colspan="4">CPG_CANFDx_TXBCF
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_00E0</td><td colspan="4">CPG_CANFDx_TXBTIE
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_00E4</td><td colspan="4">CPG_CANFDx_TXBCIE
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_00E8
0000_00EC</td><td colspan="4">-</td></tr>
<tr><td>0000_00F0</td><td colspan="4">CPG_CANFDx_TXEFC
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_00F4</td><td colspan="4">CPG_CANFDx_TXEFS
XXXXXX00 XXX00000 XXX00000 XX000000-</td></tr>
<tr><td>0000_00F8</td><td colspan="4">CPG_CANFDx_TXEFA
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_00FC
0000_01FC</td><td colspan="4">-</td></tr>
<tr><td>0000_0200</td><td colspan="2">CPG_CANFDx_FDSEAR
00000000 00000000</td><td>CPG_CANFDx_FDESR
00000000</td><td>CPG_CANFDx_FDECR
00000000</td></tr>
<tr><td>0000_0204</td><td colspan="2">CPG_CANFD0_FDDEAR
00000000 00000000</td><td>CPG_CANFDx_FDESCR
00000000</td><td>-</td></tr>
<tr><td>0000_0208
0000_02FC</td><td colspan="4">-</td></tr>
<tr><td>0000_0300
0000_7FFC</td><td colspan="4">-</td></tr>
<tr><td>0000_8000
0000_BFFC</td><td colspan="4">CAN_RAM ch0</td></tr>
<tr><td>0000_C000
0000_FFFC</td><td colspan="4">-</td></tr>
</table>
Notes:
– x is the channel number. (0 to 4)
– "*": Initial value "0" or "1" according to the setting (designed by higher layer)
– The number of circuits is designed by a higher layer.</td></tr>
</table>

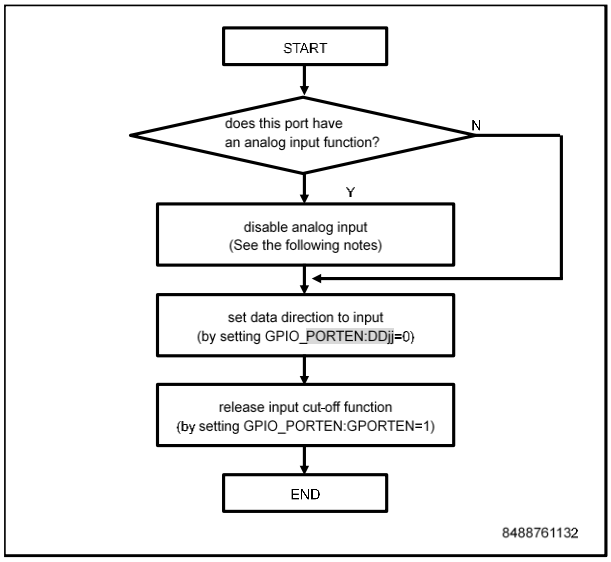
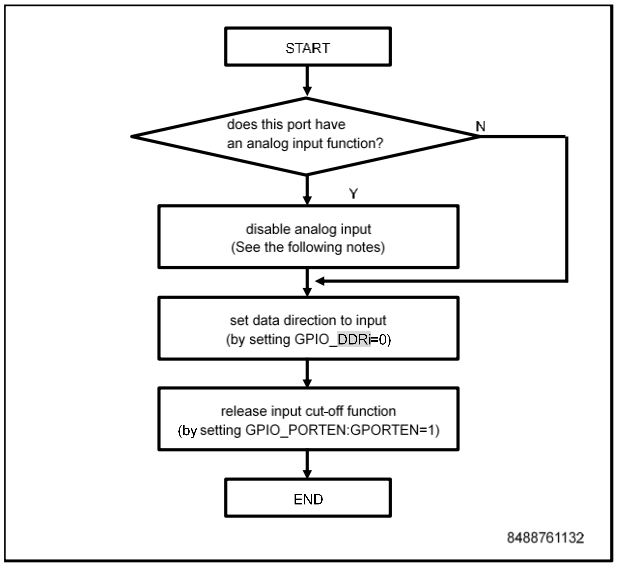
<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1734</td><td>CHAPTER 30: CAN FD Controller(MCAN3.0.1)
5. Registers</td><td>Correct)
<table>
<tr><th rowspan="2">Offset</th><th colspan="4">Register Name / Initial Value</th></tr>
<tr><th>+3</th><th>+2</th><th>+1</th><th>+0</th></tr>
<tr><td>0000_00D8</td><td colspan="4">CPG_CANFDx_TXBTO
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_00DC</td><td colspan="4">CPG_CANFDx_TXBCF
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_00E0</td><td colspan="4">CPG_CANFDx_TXBTIE
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_00E4</td><td colspan="4">CPG_CANFDx_TXBCIE
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_00E8
0000_00EC</td><td colspan="4">-</td></tr>
<tr><td>0000_00F0</td><td colspan="4">CPG_CANFDx_TXEFC
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_00F4</td><td colspan="4">CPG_CANFDx_TXEFS
XXXXXX00 XXX00000 XXX00000 XX000000</td></tr>
<tr><td>0000_00F8</td><td colspan="4">CPG_CANFDx_TXEFA
00000000 00000000 00000000 00000000</td></tr>
<tr><td>0000_00FC
0000_01FC</td><td colspan="4">-</td></tr>
<tr><td>0000_0200</td><td colspan="2">CPG_CANFDx_FDSEAR
00000000 00000000</td><td>CPG_CANFDx_FDESR
00000000</td><td>CPG_CANFDx_FDECR
00000000</td></tr>
<tr><td>0000_0204</td><td colspan="2">CPG_CANFDx_FDDEAR
00000000 00000000</td><td>CPG_CANFDx_FDESCR
00000000</td><td>-</td></tr>
<tr><td>0000_0208
0000_02FC</td><td colspan="4">-</td></tr>
<tr><td>0000_0300
0000_7FFC</td><td colspan="4">-</td></tr>
<tr><td>0000_8000
0000_BFFC</td><td colspan="4">CAN_RAM chx</td></tr>
<tr><td>0000_C000
0000_FFFC</td><td colspan="4">-</td></tr>
</table>
Notes:
– x is the channel number. (0 to 4)
– "*": Initial value "0" or "1" according to the setting (designed by higher layer)
– The number of circuits is designed by a higher layer.</td></tr>
</table>

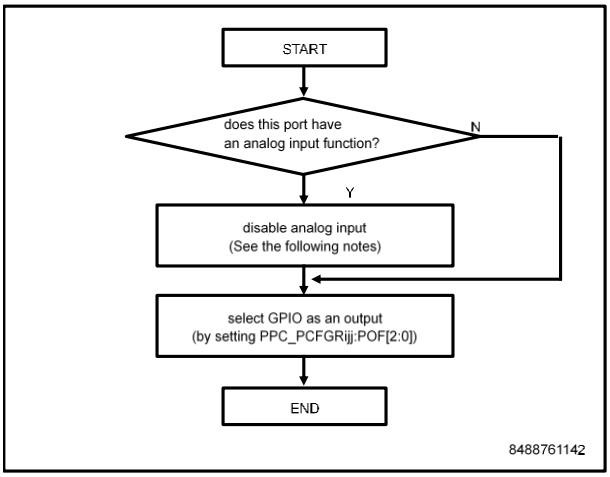
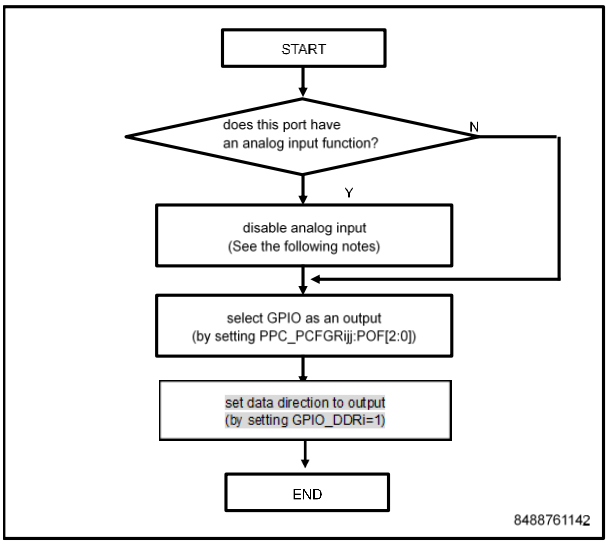
Page	Section	Change Results
2353	CHAPTER 38: I²C Interface (I²C Communication Control Interface 3. I²C Interface Operation	Changed the below: I²C Bus Repeated Start Condition Figure 3-3 Repeated Start Condition Error)  Correct) 

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>2427</td><td>CHAPTER38: I²C Interface (I²C Communication) 8. I²C Interface Registers</td><td>
Changed the below:

I2C Interface Registers
Table 8-1 I²C Interface Registers
Error)
<table>
<tr><th></th><th>bit15 bit8</th><th>bit7 bit0</th></tr>
<tr><td rowspan="16">I²C</td><td>IBCR (I²C bus control register)</td><td>SMR (serial mode register)</td></tr>
<tr><td>SSR (serial status register)</td><td>IBSR (I²C bus status register)</td></tr>
<tr><td>-</td><td>RDR/TDR (reception data register/transmission data register)</td></tr>
<tr><td>-</td><td>-</td></tr>
<tr><td>SACSR1
(serial auxiliary control status register 1)</td><td>SACSR0
(serial auxiliary control status register 0)</td></tr>
<tr><td>STMR1 (serial timer register 1)</td><td>STMR0 (serial timer register 0)</td></tr>
<tr><td>STMCR1 (serial timer comparison register 1)</td><td>STMCR0 (serial timer comparison register 0)</td></tr>
<tr><td>EIBCR (extended I²C bus control register)</td><td>NFCR (noise filter control register)</td></tr>
<tr><td>-</td><td>-</td></tr>
<tr><td>-</td><td>-</td></tr>
<tr><td>-</td><td>-</td></tr>
<tr><td>-</td><td>-</td></tr>
<tr><td>-</td><td>-</td></tr>
<tr><td></td><td></td></tr>
<tr><td>BGR1 (baud rate generator register 1)</td><td>BGR0 (baud rate generator register 0)</td></tr>
<tr><td>ISMK (7-bit slave address mask register)</td><td>ISBA (7-bit slave address register)</td></tr>
</table>
Correct)
<table>
<tr><th></th><th>bit15 bit8</th><th>bit7 bit0</th></tr>
<tr><td rowspan="16">I²C</td><td>IBCR (I²C bus control register)</td><td>SMR (serial mode register)</td></tr>
<tr><td>SSR (serial status register)</td><td>IBSR (I²C bus status register)</td></tr>
<tr><td>-</td><td>RDR/TDR (reception data register/transmission data register)</td></tr>
<tr><td>-</td><td>-</td></tr>
<tr><td>SACSR1
(serial auxiliary control status register 1)</td><td>SACSR0
(serial auxiliary control status register 0)</td></tr>
<tr><td>STMR1 (serial timer register 1)</td><td>STMR0 (serial timer register 0)</td></tr>
<tr><td>STMCR1 (serial timer comparison register 1)</td><td>STMCR0 (serial timer comparison register 0)</td></tr>
<tr><td>-</td><td>-</td></tr>
<tr><td>EIBCR (extended I²C bus control register)</td><td>NFCR (noise filter control register)</td></tr>
<tr><td>-</td><td>-</td></tr>
<tr><td>-</td><td>-</td></tr>
<tr><td>-</td><td>-</td></tr>
<tr><td>-</td><td>-</td></tr>
<tr><td>-</td><td>-</td></tr>
<tr><td>BGR1 (baud rate generator register 1)</td><td>BGR0 (baud rate generator register 0)</td></tr>
<tr><td>ISMK (7-bit slave address mask register)</td><td>ISBA (7-bit slave address register)</td></tr>
</table>
</td></tr>
</table>

Page	Section	Change Results
3018	CHAPTER 50:I/O Port 3.Setting Procedure Examples	Changed the below: **Assignment of Resource Input Pin (Selecting the Resource Input of an External Pin)** **Figure 3-2 Setting Procedure** Error)  Correct) 

Page	Section	Change Results
Page	Section	Change Results
3020	CHAPTER 50:I/O Port 3.Setting Procedure Examples	Changed the below: **Assignment of Resource Input Pin (Selecting the Resource Input of an External Pin)** **Figure 3-4 Setting Procedure** Error)  Correct) 

Page	Section	Change Results
3021	CHAPTER 50:I/O Port 3.Setting Procedure Examples	Changed the below: **Assignment of Port Function (Output)** **Figure 3-5 Setting Procedure** Error)  Correct) 

Page	Section	Change Results
3051	CHAPTER 50: I/O Port 4. Register List	Add the below: **4.13. RIC Key Code Register (RIC_KEYCDR)** **[bit14:0] RADR: Port Address Bits** ***Notes:*** − There is no effect on the key code protection release process even if registers other than this register and the applicable key code register that is in the release processing (the register set by the RADR) are accessed while writing the key code 0b00, 0b01, 0b10, and 0b11. When key code protection is released, register of protection target can be written once. If several registers are to be written, it is necessary to release the protection for each register writing.
3052	HAPTER 50: I/O Port 5. Precautions for Using This Device	Changed the below: Error) This section describes the precautions necessary when using the I/O port. A glitch may occur for a brief nanosecond (2 or 3 nanoseconds) when a general-purpose I/O port is switched (from input to output or from output to input). Correct) Notes on switching the I/O port function are shown below. A glitch may occur for a brief second (2 or 3 ns) when a general-purpose I/O port is switched (from input to output or from output to input or from port function to resource or from resource to port function). If this glitch may cause a problem for the system, please write a value to port output data register (GPIO_PODRi) in advance at a level that will not cause a problem. When a resource input is valid and its assignment is moved to another pin, a trigger which causes resource operation may occur if the pin levels before switching and after switching are different. Therefore, please switch input pins when the resource input function is stopped.
3064	CHAPTER 51: Peripheral Protection Unit 4. Registers	Deleted the below: Register map is not required because it can be extracted from each register descriptions. For the customers, they should refer DS for the I/O map.

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>3194</td><td>CHAPTER 52:DDR High Speed SPI Controller

5. Registers</td><td>Changed the below:
5.25. DDRHSSPI SDATA Center Clock Sample Point Registers (DDRHSSPIn_SDATASAMPLEPTCNT0-7)
Error)
<table>
<tr><td>BIT_OFFSET</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>BIT_NAME</td><td>Reserved</td><td>SDATASMP TCNT[6]</td><td>SDATASMP TCNT[5]</td><td>SDATASMP TCNT[4]</td><td>SDATASMP TCNT[3]</td><td>SDATASMP TCNT[2]</td><td>SDATASMP TCNT[1]</td><td>SDATASMP TCNT[0]</td></tr>
<tr><td>ACCESS_TYPE</td><td>R0,W0</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td></tr>
<tr><td>PROT_TYPE</td><td colspan="8">-</td></tr>
<tr><td>INITIAL_VALUE</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
[bit7] Reserved: Reserved

[bit6:0] SDATASMPTCNT[6:0]: SDATA Sample Point Center Control

Correct)
<table>
<tr><td>BIT_OFFSET</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>BIT_NAME</td><td>Reserved</td><td>Reserved</td><td>SDATASMP TCNT[5]</td><td>SDATASMP TCNT[4]</td><td>SDATASMP TCNT[3]</td><td>SDATASMP TCNT[2]</td><td>SDATASMP TCNT[1]</td><td>SDATASMP TCNT[0]</td></tr>
<tr><td>ACCESS_TYPE</td><td>R0,W0</td><td>R0,W0</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td></tr>
<tr><td>PROT_TYPE</td><td colspan="8">-</td></tr>
<tr><td>INITIAL_VALUE</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
[bit7:6] Reserved: Reserved

[bit5:0] SDATASMPTCNT[5:0]: SDATA Sample Point Center Control</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>3195</td><td>CHAPTER 52:DDR High Speed SPI Controller

5. Registers</td><td>Changed the below:
5.26. DDRHSSPI SDATA Left Clock Sample Point Registers (DDRHSSPIn_SDATASAMPLEPTLFT0-7)
Error)
<table>
<tr><td>BIT_OFFSET</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>BIT_NAME</td><td>Reserved</td><td>SDATASMP TLFT[6]</td><td>SDATASMP TLFT[5]</td><td>SDATASMP TLFT[4]</td><td>SDATASMP TLFT[3]</td><td>SDATASMP TLFT[2]</td><td>SDATASMP TLFT[1]</td><td>SDATASMP TLFT[0]</td></tr>
<tr><td>ACCESS_TYPE</td><td>R0,W0</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td></tr>
<tr><td>PROT_TYPE</td><td colspan="8">-</td></tr>
<tr><td>INITIAL_VALUE</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
[bit7] Reserved: Reserved

[bit6:0] SDATASMPTLFT[6:0]: SDATA Left Clock Sample Point Control

Correct)
<table>
<tr><td>BIT_OFFSET</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>BIT_NAME</td><td>Reserved</td><td>Reserved</td><td>SDATASMP TLFT[5]</td><td>SDATASMP TLFT[4]</td><td>SDATASMP TLFT[3]</td><td>SDATASMP TLFT[2]</td><td>SDATASMP TLFT[1]</td><td>SDATASMP TLFT[0]</td></tr>
<tr><td>ACCESS_TYPE</td><td>R0,W0</td><td>R0,W0</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td></tr>
<tr><td>PROT_TYPE</td><td colspan="8">-</td></tr>
<tr><td>INITIAL_VALUE</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
[bit7:6] Reserved: Reserved

[bit5:0] SDATASMPTLFT[5:0]: SDATA Left Clock Sample Point Control</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>3196</td><td>CHAPTER 52:DDR High Speed SPI Controller

5. Registers</td><td>Changed the below:
5.27. DDRHSSPI SDATA Right Clock Sample Point Registers (DDRHSSPIn_SDATASAMPLEPTRGH0-7)

Error)
<table>
<tr><td>BIT_OFFSET</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>BIT_NAME</td><td>Reserved</td><td>SDATASMP TRGH[6]</td><td>SDATASMP TRGH[5]</td><td>SDATASMP TRGH[4]</td><td>SDATASMP TRGH[3]</td><td>SDATASMP TRGH[2]</td><td>SDATASMP TRGH[1]</td><td>SDATASMP TRGH[0]</td></tr>
<tr><td>ACCESS_TYPE</td><td>R0,W0</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td></tr>
<tr><td>PROT_TYPE</td><td colspan="8">-</td></tr>
<tr><td>INITIAL_VALUE</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
[bit7] Reserved: Reserved

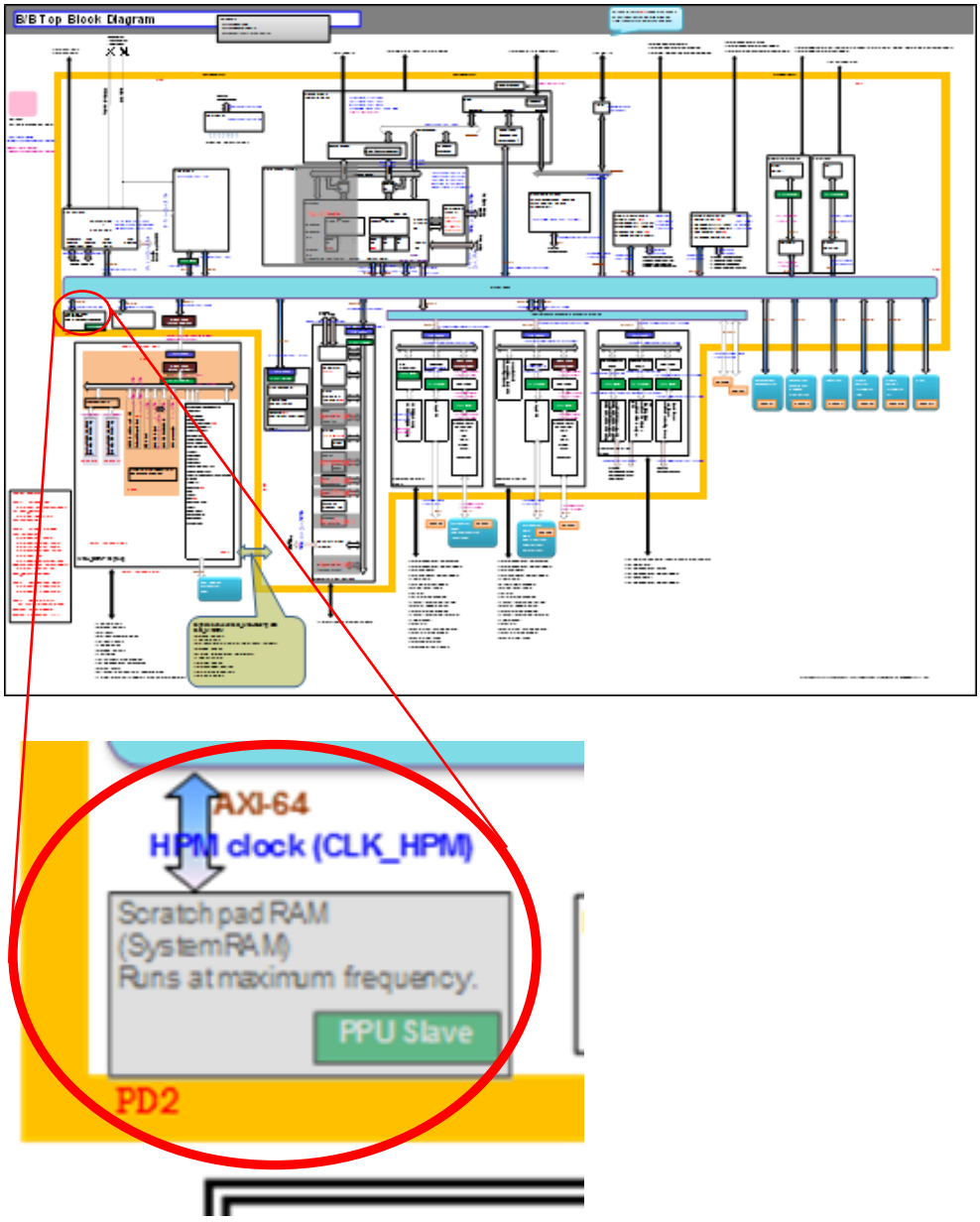
[bit6:0] SDATASMPTRGH[6:0]: SDATA Right Clock Sample Point Control

Correct)
<table>
<tr><td>BIT_OFFSET</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>BIT_NAME</td><td>Reserved</td><td>Reserved</td><td>SDATASMP TRGH[5]</td><td>SDATASMP TRGH[4]</td><td>SDATASMP TRGH[3]</td><td>SDATASMP TRGH[2]</td><td>SDATASMP TRGH[1]</td><td>SDATASMP TRGH[0]</td></tr>
<tr><td>ACCESS_TYPE</td><td>R0,W0</td><td>R0,W0</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td></tr>
<tr><td>PROT_TYPE</td><td colspan="8">-</td></tr>
<tr><td>INITIAL_VALUE</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
[bit7:6] Reserved: Reserved

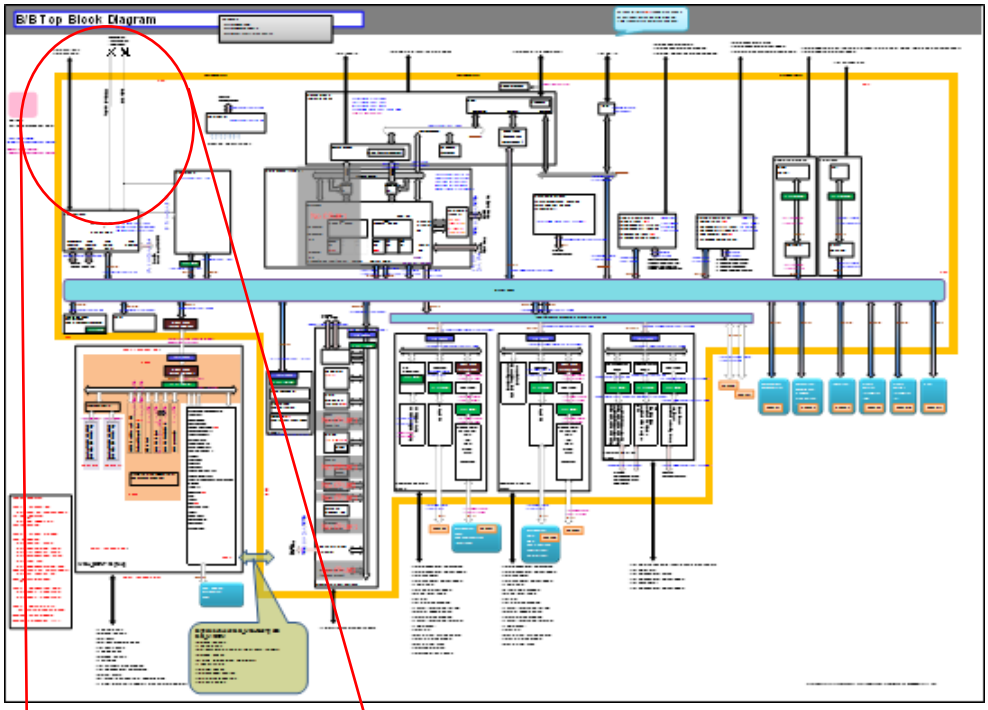
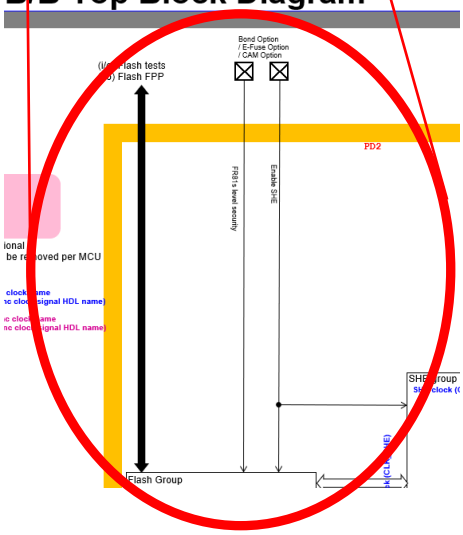
[bit5:0] SDATASMPTRGH[5:0]: SDATA Right Clock Sample Point Control</td></tr>
</table>

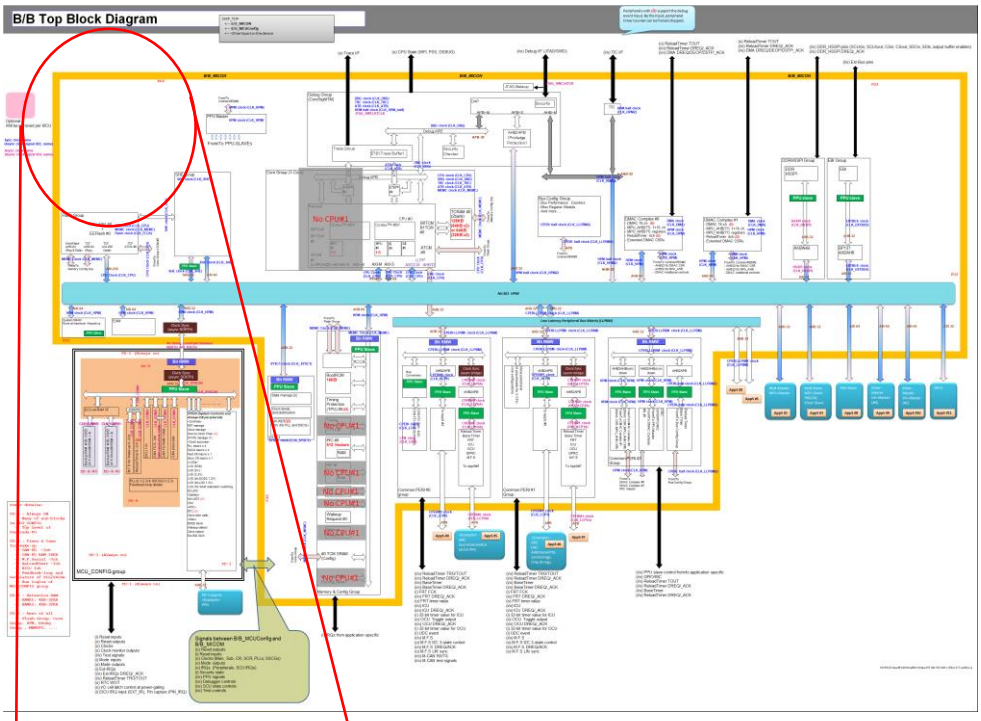
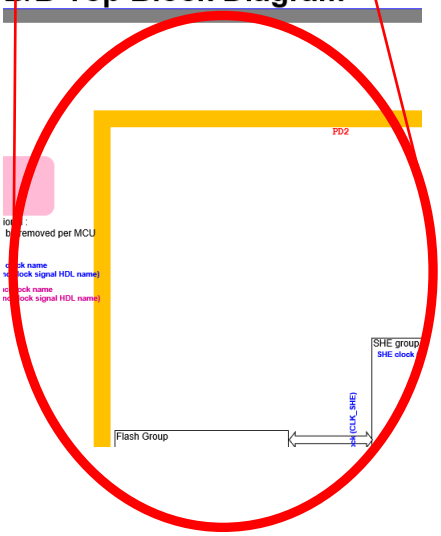
Page	Section	Change Results
002-04854 Rev. *E		
14	CHAPTER 1: Platform Overview	Removed the shading parts as below: Error) B/B is a latest effort of the MCU platform which targets instrument cluster, graphics controller, gateway and body controller application area. Correct) B/B is the MCU platform which targets instrument cluster, graphics controller, gateway and body controller application area.
		Removed the shading parts as below: Error) To minimize design efforts of the target MCU, B/B offers rich and competitive feature sets like power saving features and IP portfolio, easy-to-use parameter option to select them, and verification suite which proves its quality.
		Revised the shading parts as below: ■Computation features Error) Cortex-R5F single core, 16K I-Cache. Targeting >240MHz with 55nm technology. Program Flash I/F which have advanced speculative prefetching algorithm. Corrrect) Cortex-R5F single core, 16K I-Cache. Main Flash I/F which have advanced speculative prefetching algorithm.
15		Removed the shading parts as below: ■Connectivity Error) CAN FD offers latest CAN protocol and highly flexible message buffering. Ext-BUS I/F supporting NAND flash

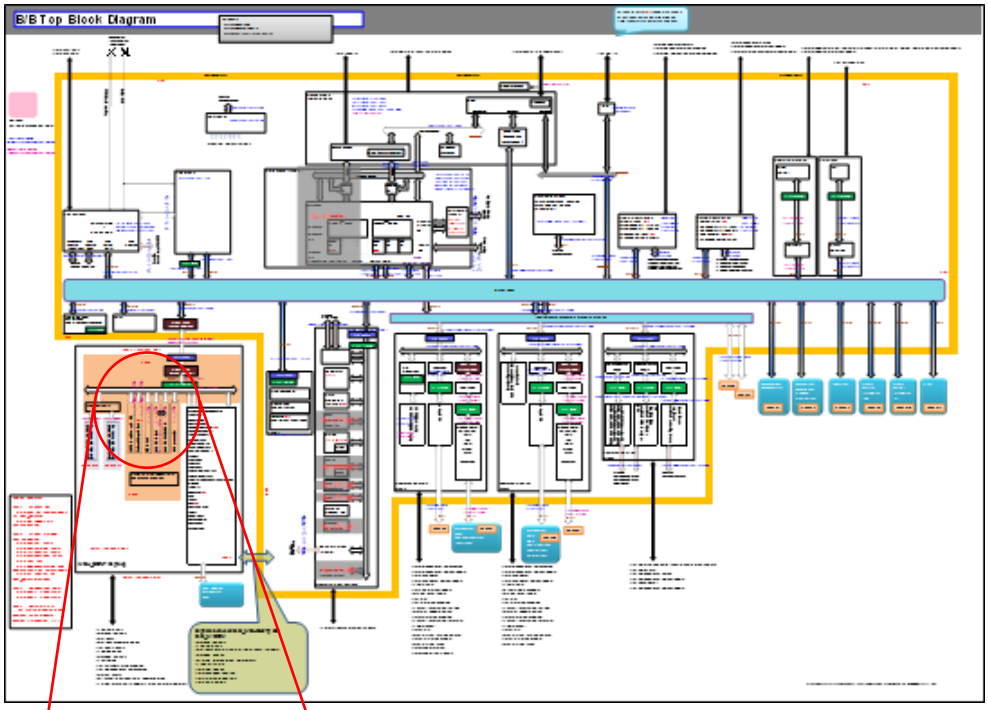
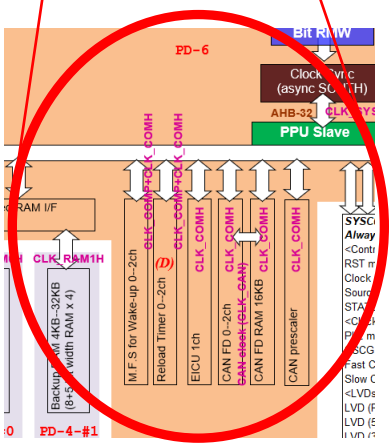
Page	Section	Change Results
16	CHAPTER 1: Platform Overview 1.1. Function Overview	Added the shading parts as below: Correct) CPU : ARM Cortex™-R5F (revision r1p3)
		Removed the shading parts as below: Operation modes : Error) Test modes
		Revised the shading parts as below: Clock controls : Error) Sub clock by 32K RTC resonator Correct) Sub clock by 32kHz RTC resonator
17	CHAPTER 1: Platform Overview 1.1. Function Overview	Added the shading parts as below: Correct) Main Flash memory (TCFlash) : Up to 4MB (product specification) SECDED ECC by 8-byte via AXI Read access via TCM port Read and write access via AXI TCM buffer for effective access via TCM port The ECC movement in TCM port is based on ECC setting inside the CPU

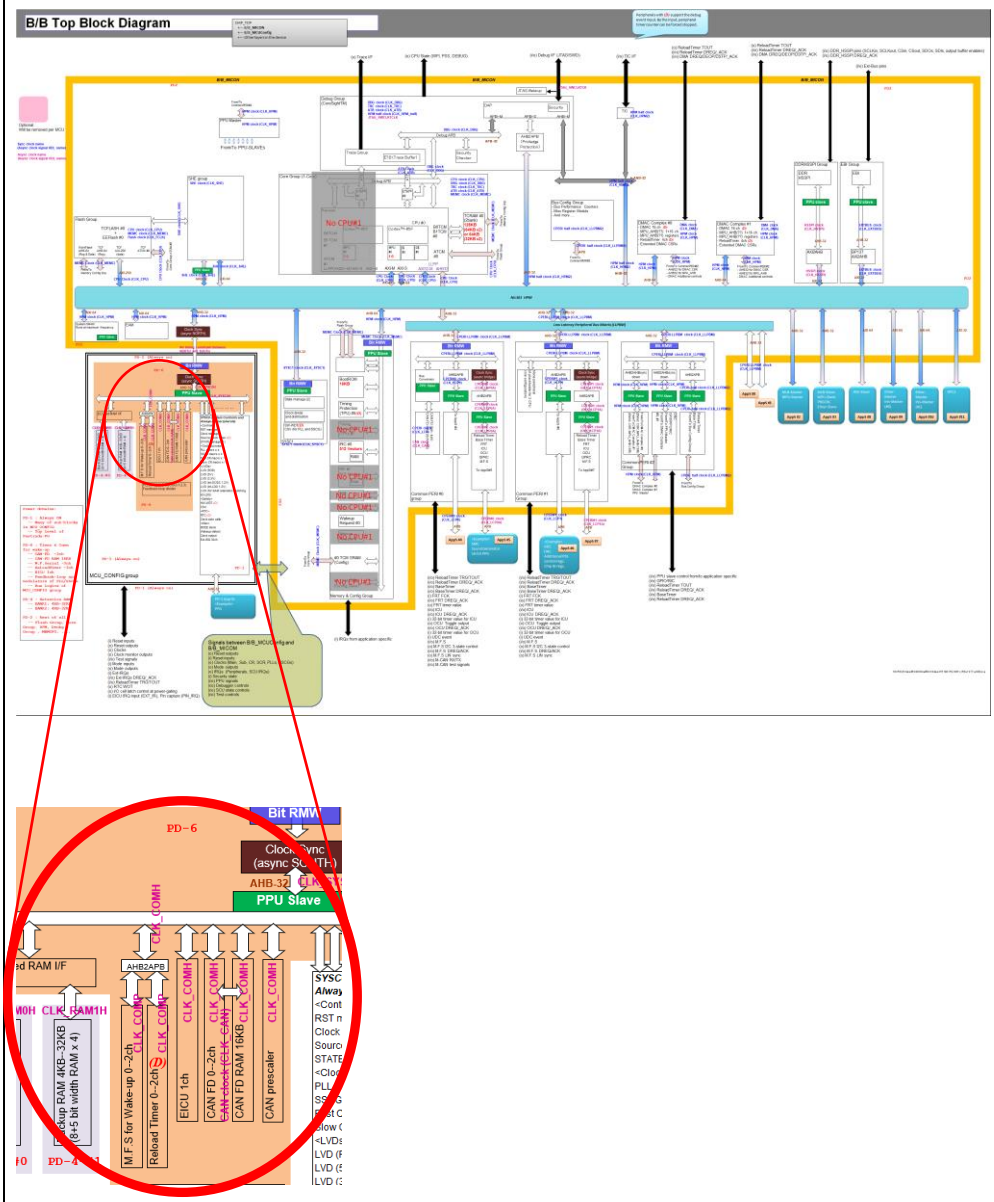
Page	Section	Change Results
20	CHAPTER 1: Platform Overview 3. Configuration	Revised the shading parts as below: Error) Scratch pad RAM (System RAM) 

Page	Section	Change Results
		Correct) System SRAM 

Page	Section	Change Results
		Revised the shading parts as below: Error) (E-Fuse Option)  

Page	Section	Change Results
		Correct)  

Page	Section	Change Results
		Revised the shading parts as below: Error) (Divide CLK_COMP/CLK_COMH Block)  

Page	Section	Change Results
		Correct) 
21	CHAPTER 1: Platform Overview 4. Abbreviations	Removed the shading parts as below: Error) IPCU -- Inter Processor Communication Unit

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>24</td><td>CHAPTER 1: Platform Overview
5. Memory Map</td><td>Revised the shading parts as below:

Error
<table>
<tr><td colspan="2">Adress</td><td colspan="3">size</td><td colspan="2">Module</td></tr>
<tr><td>START</td><td>END</td><td>[Mbyte]</td><td>[Kbyte]</td><td>[Kbyte]</td><td>group</td><td>Part</td></tr>
<tr><td>0200_0000</td><td>027F_FFFF</td><td>256</td><td>32768</td><td>8192</td><td>Shared Flash and memory area</td><td>Scratch Pad RAM 8MB@Max</td></tr>
</table>

Correct
<table>
<tr><td colspan="2">Adress</td><td colspan="3">size</td><td colspan="2">Module</td></tr>
<tr><td>START</td><td>END</td><td>[Mbyte]</td><td>[Kbyte]</td><td>[Kbyte]</td><td>group</td><td>Part</td></tr>
<tr><td>0200_0000</td><td>027F_FFFF</td><td>256</td><td>32768</td><td>8192</td><td>Shared Flash and memory area</td><td>System SRAM 8MB@Max</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>24</td><td>CHAPTER 1: Platform Overview
5. Memory Map</td><td>Revised the shading parts as below:

Error
<table>
<tr><th colspan="2">Adress</th><th colspan="3">size</th><th colspan="2">Module</th></tr>
<tr><th>START</th><th>END</th><th>[Mbyte]</th><th>[Kbyte]</th><th>[Kbyte]</th><th>group</th><th>Part</th></tr>
<tr><td>B000_0000</td><td>B7FF_FFFF</td><td>128</td><td>131072</td><td>131072</td><td>Peri area</td><td>EMEM GROUP (reg)
Scratch Pad RAM (reg)
MCUCFG GROUP
SYSC1
MEMCFG GROUP
DEBUG GROUP
SHE_CONFIG
BUS CONFIG GROUP (via CPERI#2)
CPER#0, #1, #2
AppS#0--1, #4--7
Bit RMW alias</td></tr>
</table>

Correct
<table>
<tr><th colspan="2">Adress</th><th colspan="3">size</th><th colspan="2">Module</th></tr>
<tr><th>START</th><th>END</th><th>[Mbyte]</th><th>[Kbyte]</th><th>[Kbyte]</th><th>group</th><th>Part</th></tr>
<tr><td>B000_0000</td><td>B7FF_FFFF</td><td>128</td><td>131072</td><td>131072</td><td>Peri area</td><td>EMEM GROUP (reg)
System SRAM (reg)
MCUCFG GROUP
SYSC1
MEMCFG GROUP
DEBUG GROUP
SHE_CONFIG
BUS CONFIG GROUP (via CPERI#2)
CPER#0, #1, #2
AppS#0--1, #4--7
Bit RMW alias</td></tr>
</table>
</td></tr>
<tr><td>25</td><td>CHAPTER 1: Platform Overview
5. Memory Map</td><td>Removed the shading parts as below:

Error)
− External Bus Interface (when writing commands to NAND FLASH)</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>26</td><td>CHAPTER 1: Platform Overview
6. I/O Map I/O Map (1)</td><td>Revised the shading parts as below:

Error
<table><tr><td></td><td>START Address</td><td>END Address</td><td>Function</td><td>Power Domain</td><td>Size (DEC)</td><td>Size (HEX)</td></tr><tr><td>Scratch Pad RAM</td><td>B010_8000</td><td>B010_80FF</td><td>Scratch Pad RAM registers</td><td>PD2</td><td>256</td><td>100</td></tr></table>
Correct
<table><tr><td></td><td>START Address</td><td>END Address</td><td>Function</td><td>Power Domain</td><td>Size (DEC)</td><td>Size (HEX)</td></tr><tr><td>System SRAM</td><td>B010_8000</td><td>B010_80FF</td><td>System SRAM registers</td><td>PD2</td><td>256</td><td>100</td></tr></table></td></tr>
<tr><td>27</td><td>CHAPTER 1: Platform Overview
6. I/O Map I/O Map (2)</td><td>Revised the shading parts as below:

Error
<table><tr><td>B475_0000</td><td>B475_7FFF</td><td>PPU</td><td>PD2</td><td>32768</td><td>8000</td><td>77</td></tr></table>
Correct
<table><tr><td>B475_0000</td><td>B475_FFFF</td><td>Reserved</td><td>-</td><td>32768</td><td>8000</td><td>77</td></tr></table></td></tr>
<tr><td>32</td><td>CHAPTER 1: Platform Overview
8.1 How to Use Reset Factor Register</td><td>Revised the shading parts as below:

Error)
Clear all reset registers by lump-sum, It is possible to process early when clearing by the writing in of byte/half Word/Word.

Correct)
Clear all reset registers by lump-sum, It is possible to process early when clearing by the writing in of Word.</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>47</td><td>CHAPTER 3: Operation Mode
4. Register</td><td>Revised the shading parts as below:

Table 4-3 List of Register Mirror Area

Error)
<table>
<tr><th>Address of Register Mirror Area</th><th>Mirrored Peripheral</th><th>Mirror Area Behavior</th><th>PPU</th></tr>
<tr><td>0000_0880 to 0000_08FF
0000_0900 to 0000_097F
0000_0980 to 0000_09FF
0000_0A00 to 0000_0A7F
0000_0A80 to 0000_0AFF
0000_0B00 to 0000_0B7F
0000_0B80 to 0000_0BFF
0000_0C00 to 0000_0C7F
0000_0C80 to 0000_0CFF
0000_0D00 to 0000_0D7F
0000_0D80 to 0000_0DFF
0000_0E00 to 0000_0E7F
0000_0E80 to 0000_0EFF
0000_0F00 to 0000_0F7F
0000_0F80 to 0000_0FFF</td><td>MODEC</td><td>Accessing this region results to the access to MODEC register area with following offset.
Offset: (addr & 0000_007F)</td><td>This area is covered by PPU #55 as well as the mirrored peripheral.</td></tr>
</table>
Correct)
<table>
<tr><th>Address of Register Mirror Area</th><th>Mirrored Peripheral</th><th>Mirror Area Behavior</th><th>PPU</th></tr>
<tr><td>0x0000_0900 to 0x0000_09FF
0x0000_0A00 to 0x0000_0AFF
0x0000_0B00 to 0x0000_0BFF
0x0000_0C00 to 0x0000_0CFF
0x0000_0D00 to 0x0000_0DFF
0x0000_0E00 to 0x0000_0EFF
0x0000_0F00 to 0x0000_0FFF</td><td>MODEC</td><td>Accessing this region results to the access to MODEC register area with following offset.
Offset: (addr & 0000_00FF)</td><td>This area is covered by PPU #55 as well as the mirrored peripheral.</td></tr>
</table></td></tr>
</table>

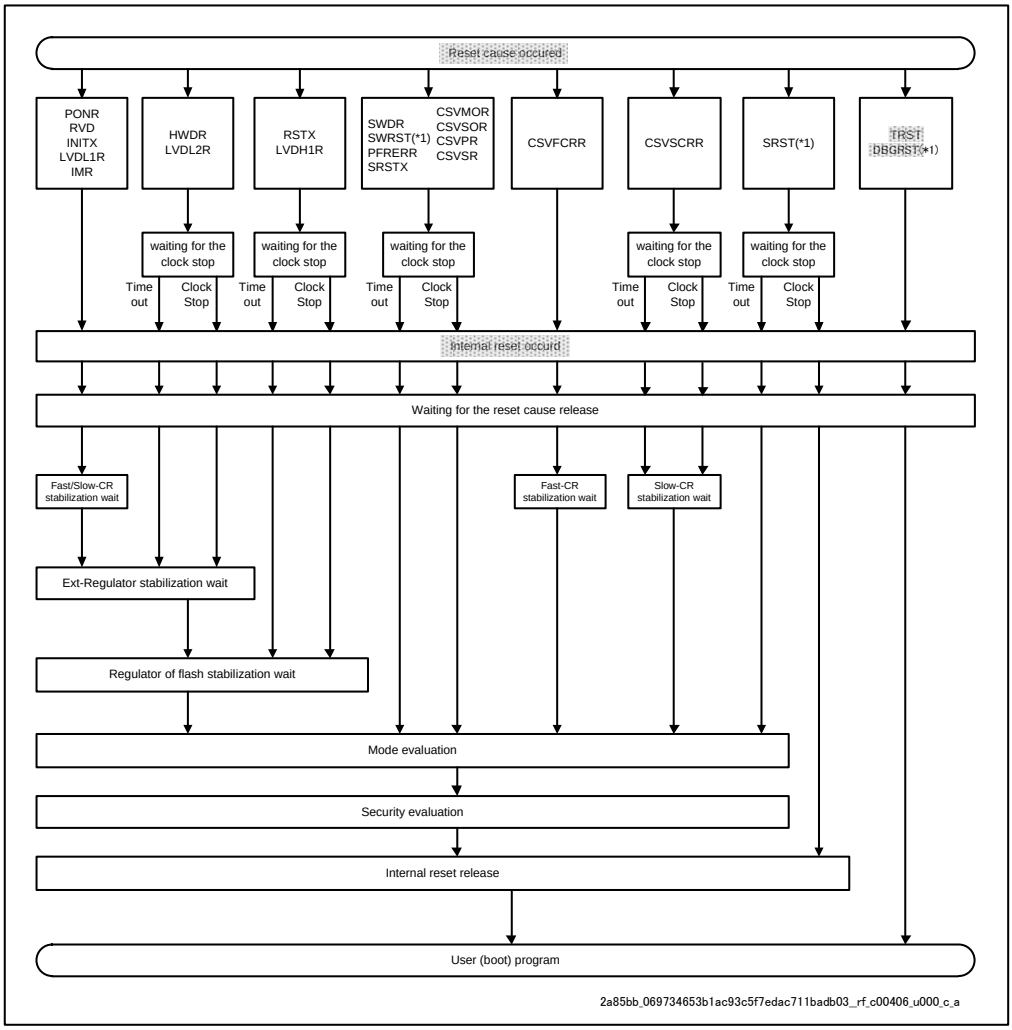
<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>53</td><td>CHAPTER 4
2. Configuration
Figure 2-1 Reset System Block Diagram</td><td>Revised the shading parts in Figure 2-1:

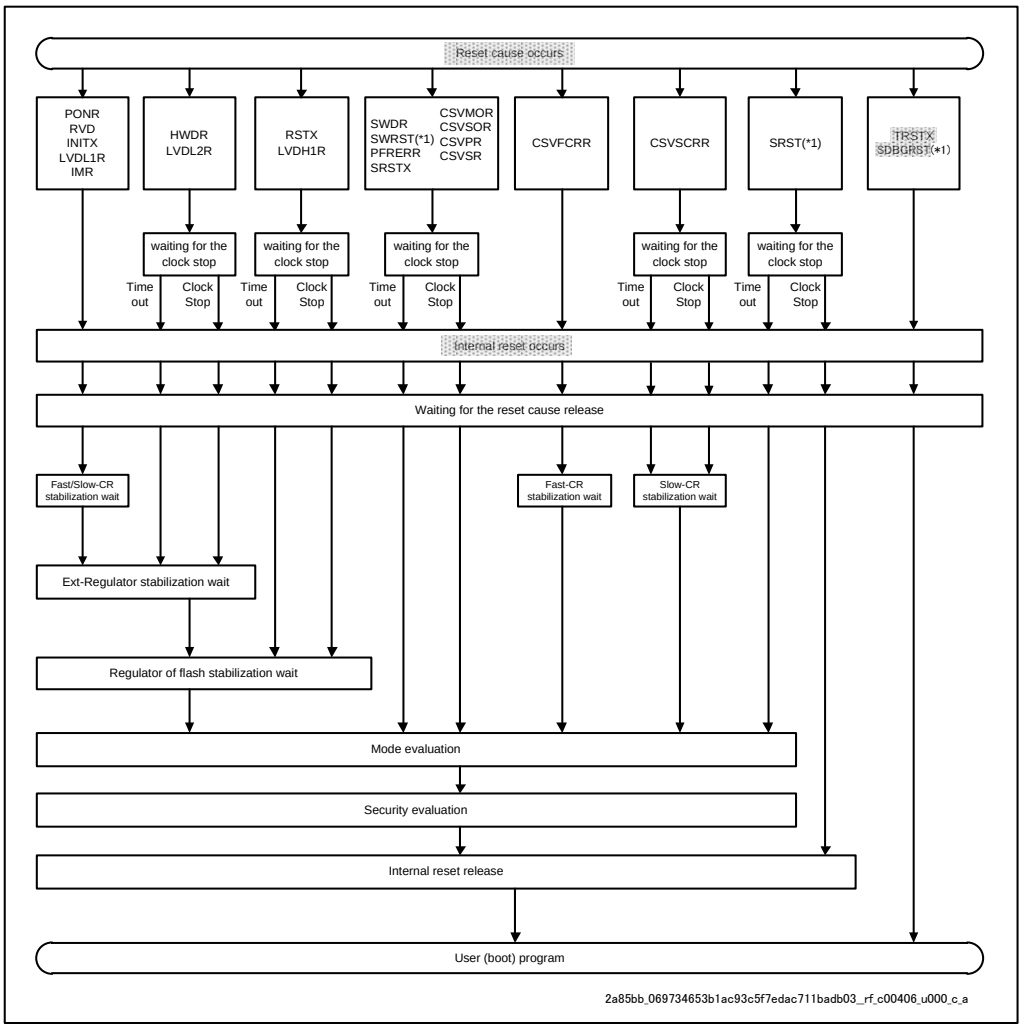
Error)
Retention RAM Power Domain (PD4)
Retention RAM
Retention RAM I/F
Scratch pad RAM
CSVMOR/CSVMOR
CSVPRn/CSVSRN
It is asserted during return from power-off and power-off during.
*3: It consists of PONR/INITX/RVD/LVDL1R/HWDR/CSVMOR /CSVMOR

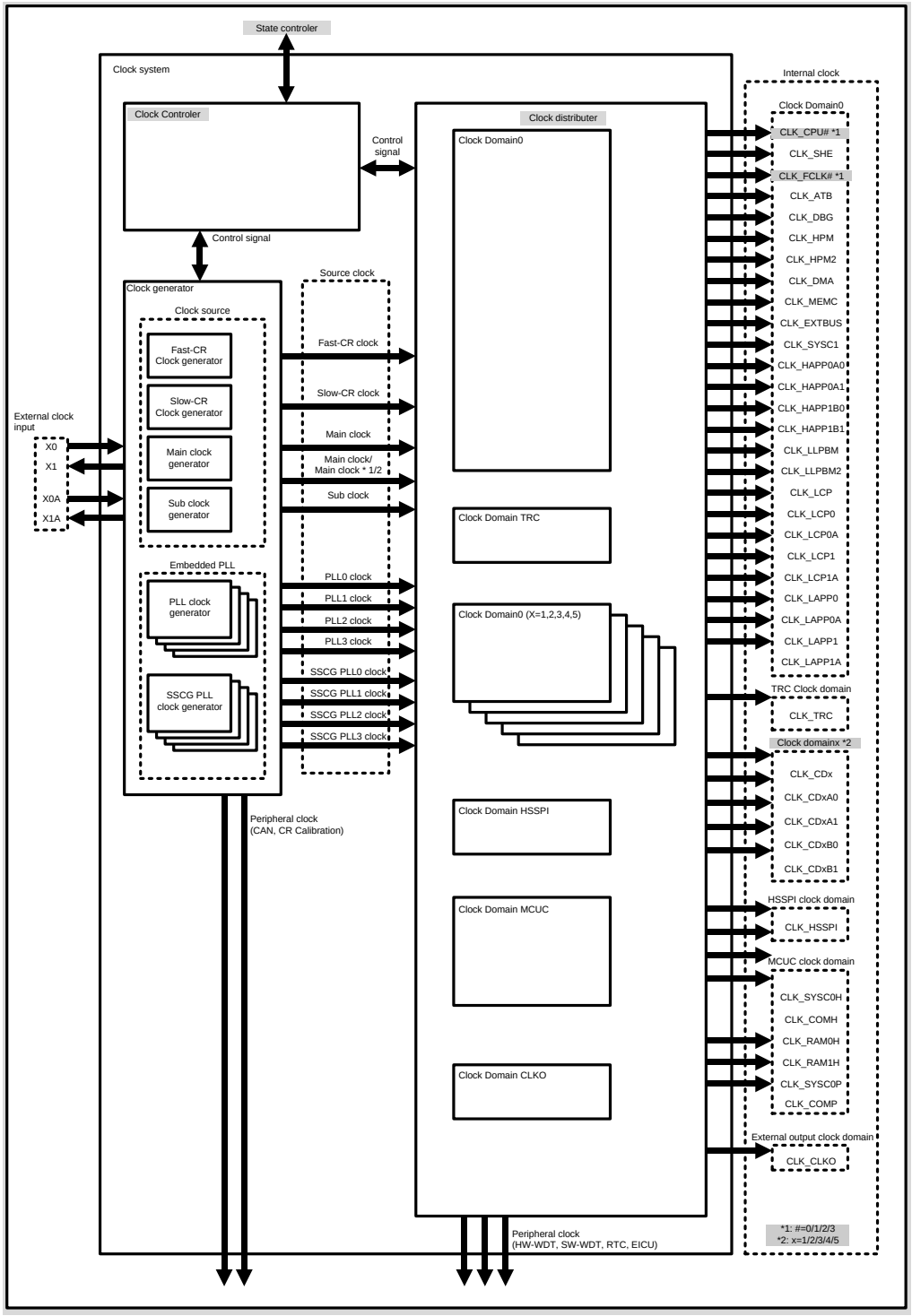
Correct)
Backup RAM Power Domain (PD4)
Backup RAM
Backup RAM I/F
System SRAM
CSVMOR/CSVSOR
CSVPRn/CSVSRn
It is asserted during power-off and during return from power-off.
*3: It consists of PONR/INITX/RVD/LVDL1R/HWDR/CSVMOR/CSVSOR/CSVSCRR.</td></tr>
<tr><td>59</td><td>CHAPTER 4: Reset
3. Operational Description
3.1.3. INITX</td><td>Revised the shading parts as below:

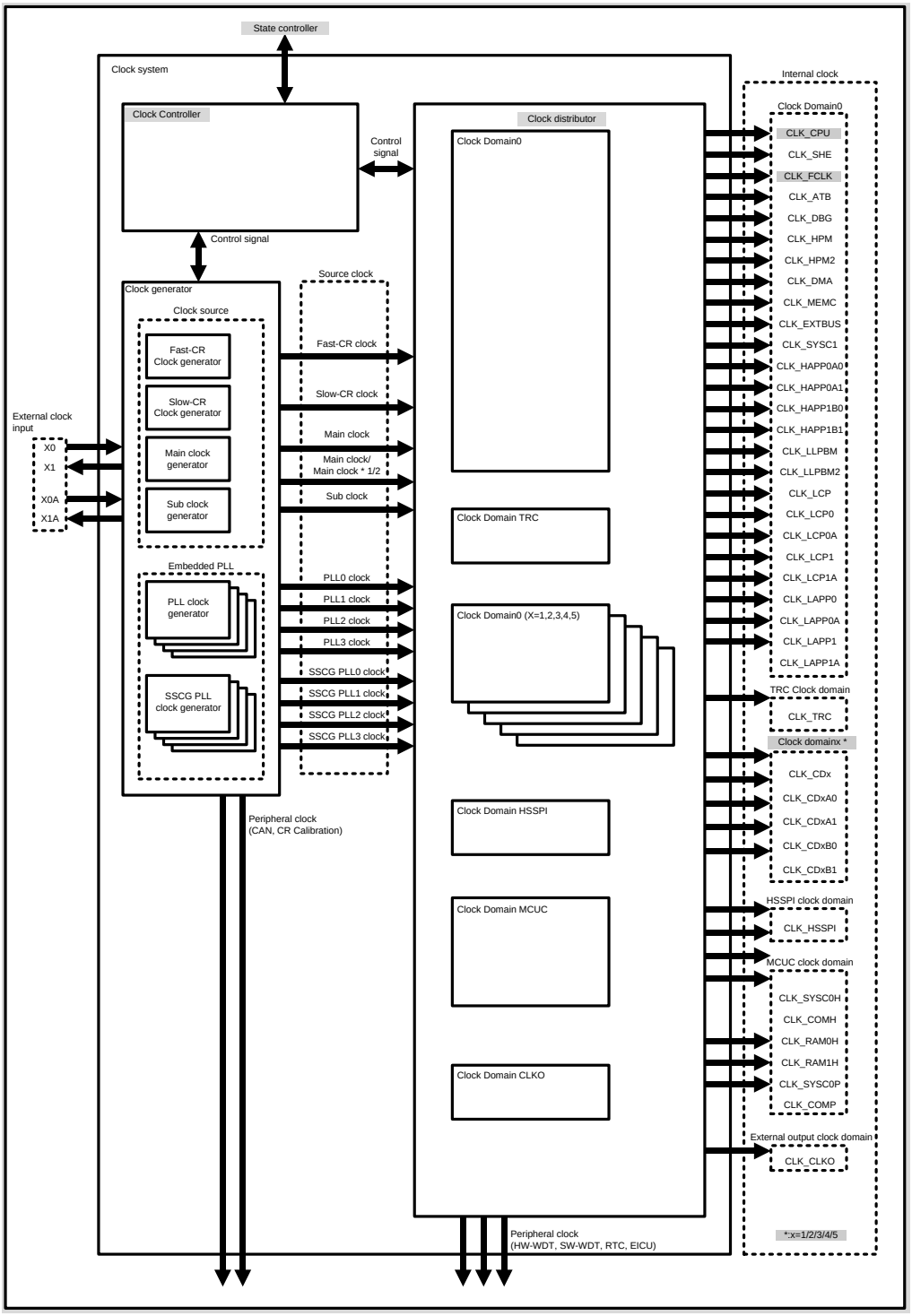
Error)<table><tr><td>Occurrence factor</td><td>It is issued by inputting "L" level to INITX pin.</td></tr><tr><td>Release factor</td><td>It is released by inputting "H" level to INITX pin.</td></tr></table>
Correct)<table><tr><td>Occurrence factor</td><td>Or,it is issued by inputting "L" level to simultaneous assert of RSTX pin and MD pin.</td></tr><tr><td>Release factor</td><td>Or,it is released by inputting other than "L" level to simultaneous assert of RSTX pin and MD pin.</td></tr></table></td></tr>
</table>

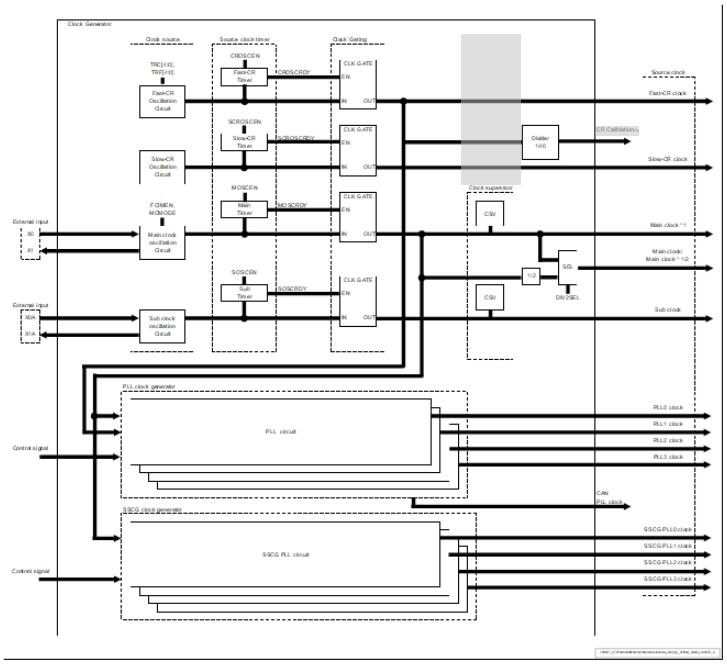
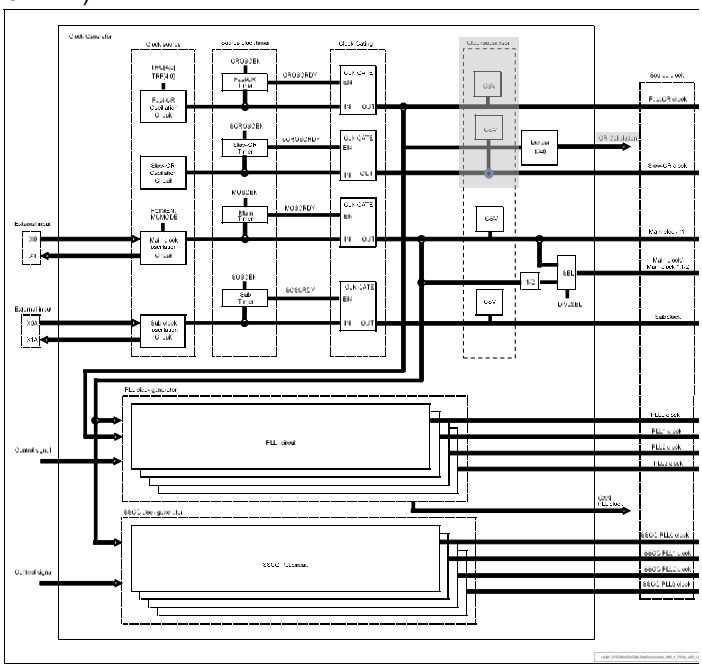
Page	Section	Change Results
82	CHAPTER 4: Reset 3. Operational Description 3.3. Reset Sequence	Revised the shading parts as below: Error)

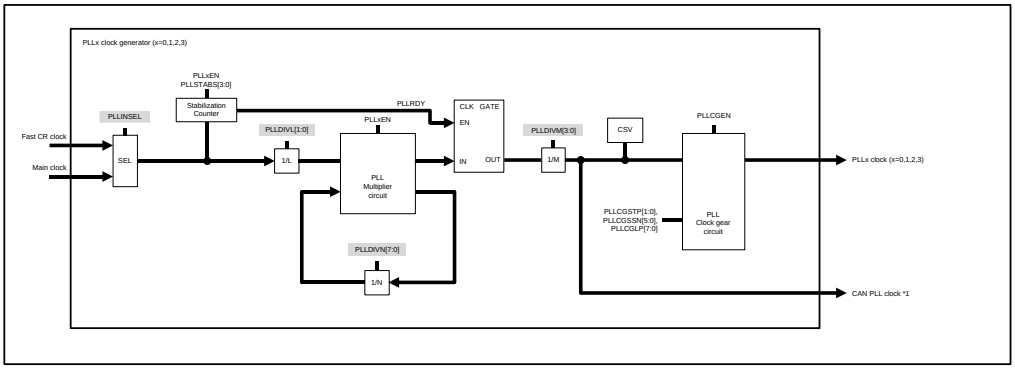
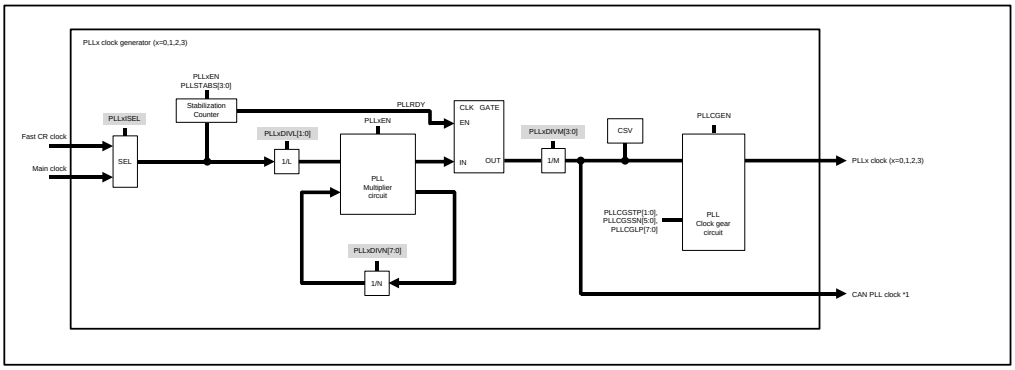


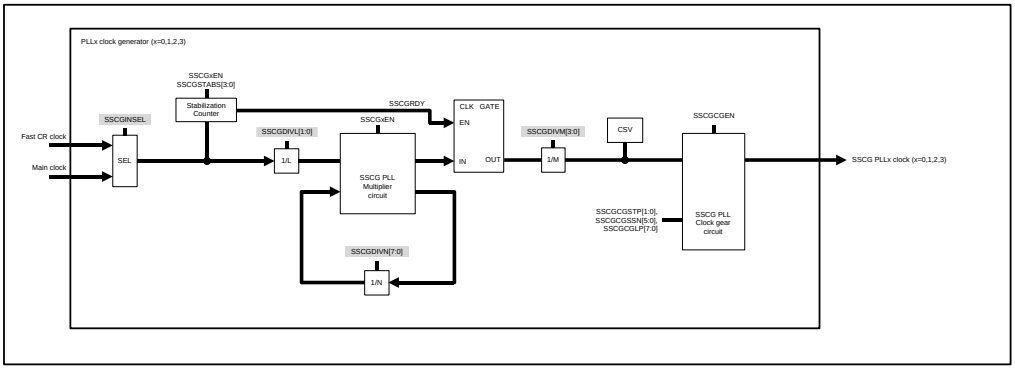
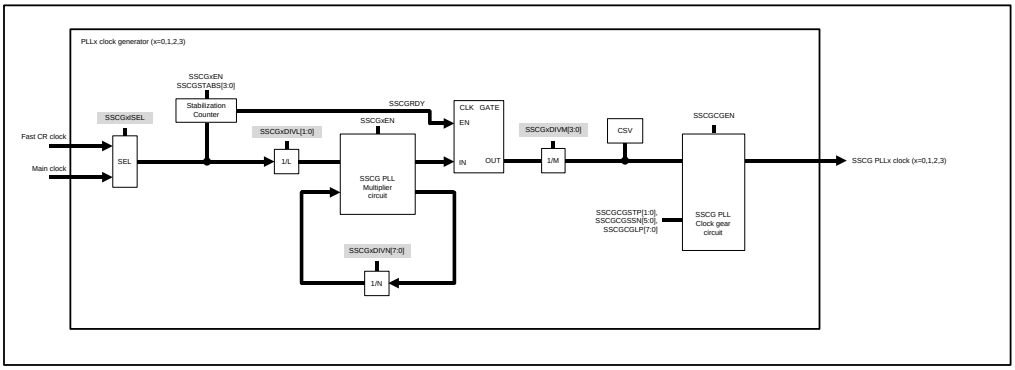
Page	Section	Change Results
		Correct) 
94	CHAPTER 4: Reset 4. Registers	Revised the shading parts as below: Error) - Protection key code input is required to write the registers. - It results bus error response for the violation to the rules. Correct) - The registers are protected by protection key setting register (SYSC0_PROTKEYR). For details on the protection, see the following chapter 6: "Low-power Consumption." - Access to reserved address results bus error response.

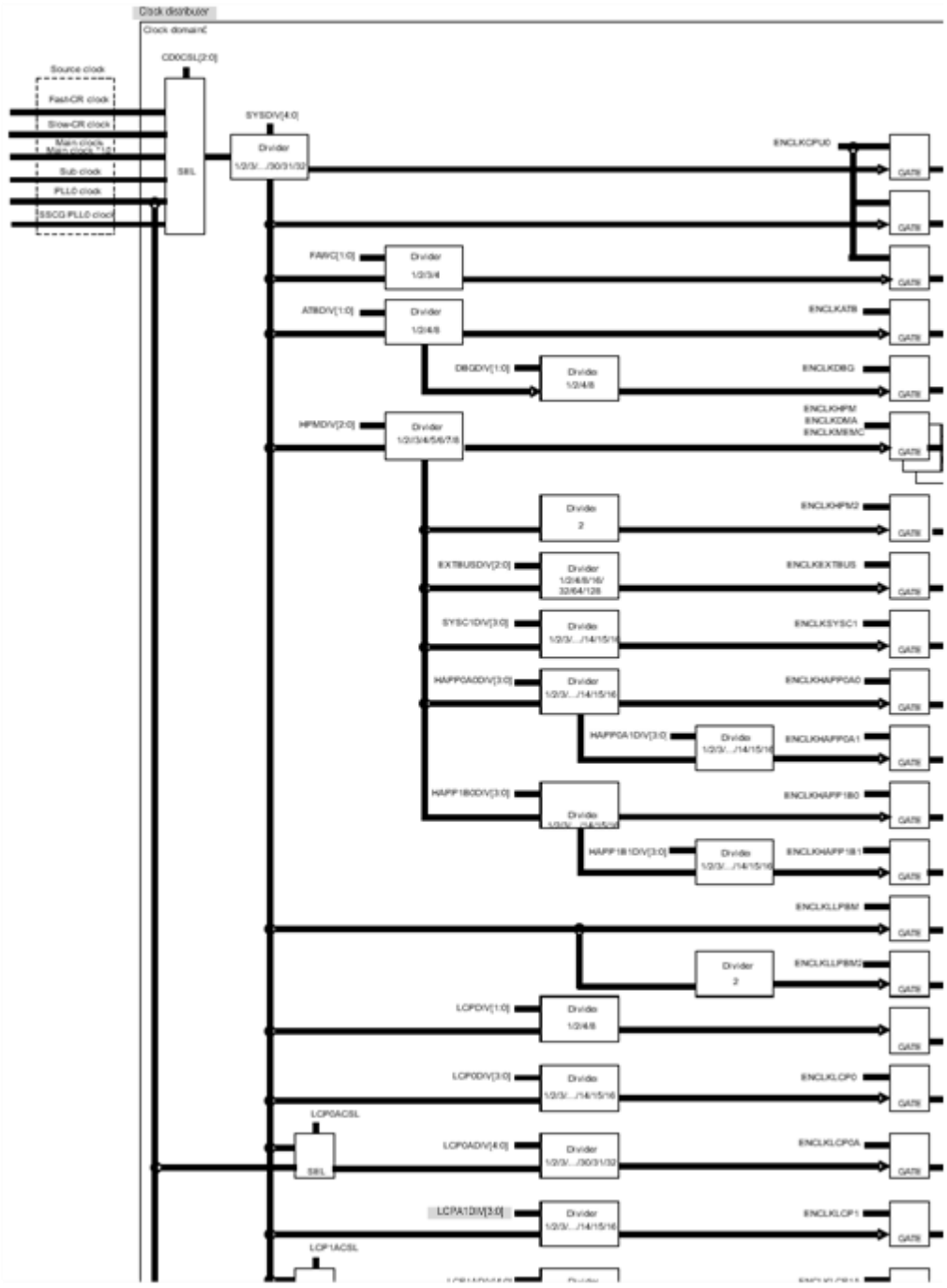
Page	Section	Change Results
127	CHAPTER 5: Clock System 2. Configuration and Block Diagram 2.1. Clock System Configuration and Block Diagram	Revised the shading parts as below: Figure 2-1 Clock System Block Diagram Error) 

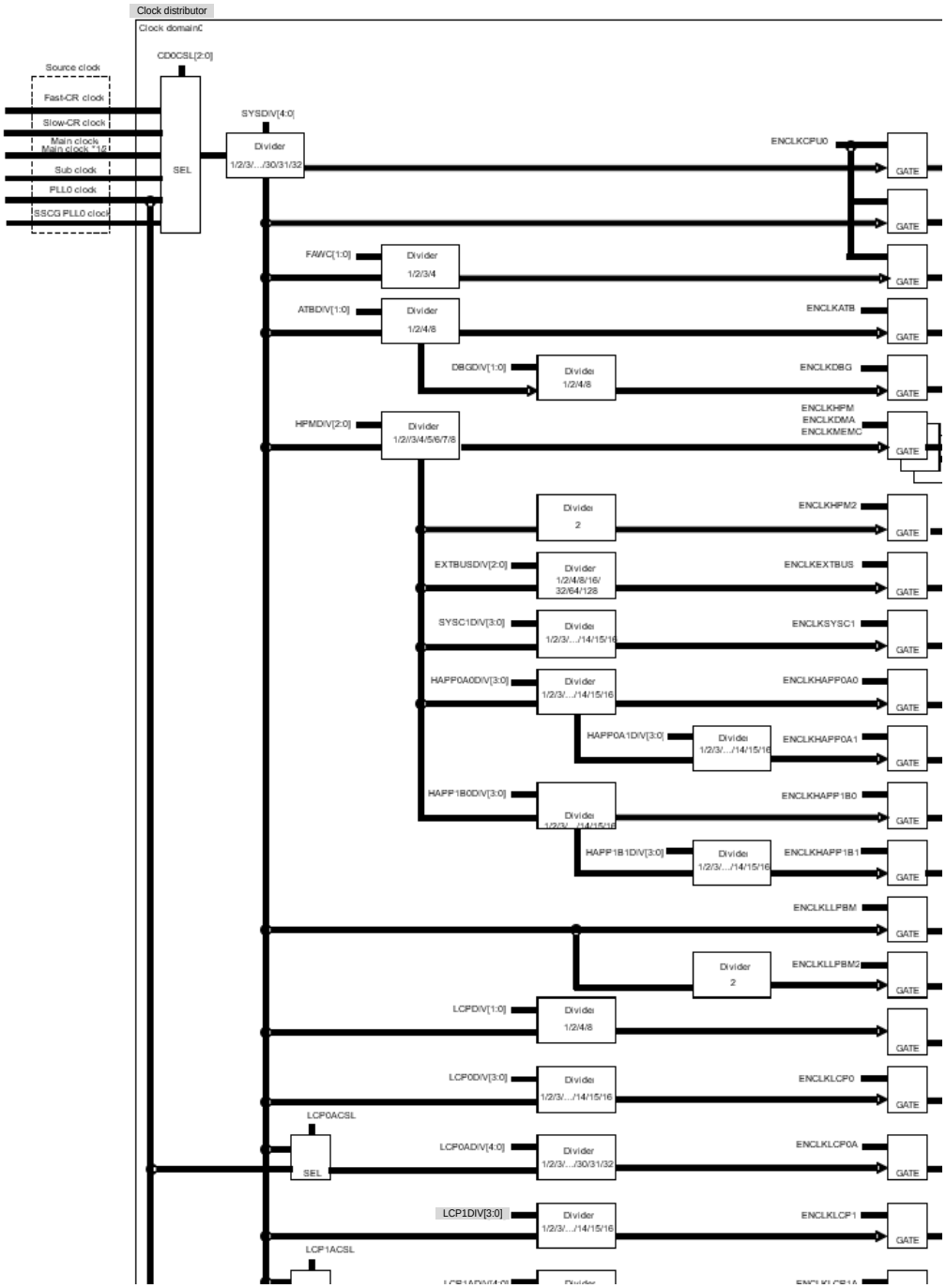
Page	Section	Change Results
		Correct) 

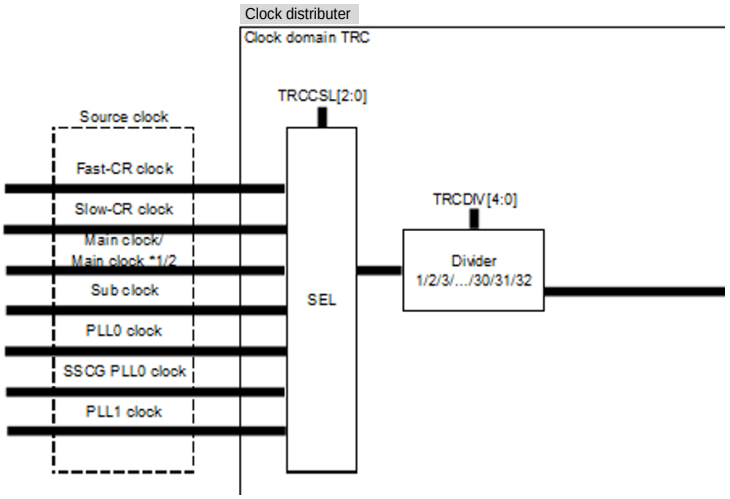
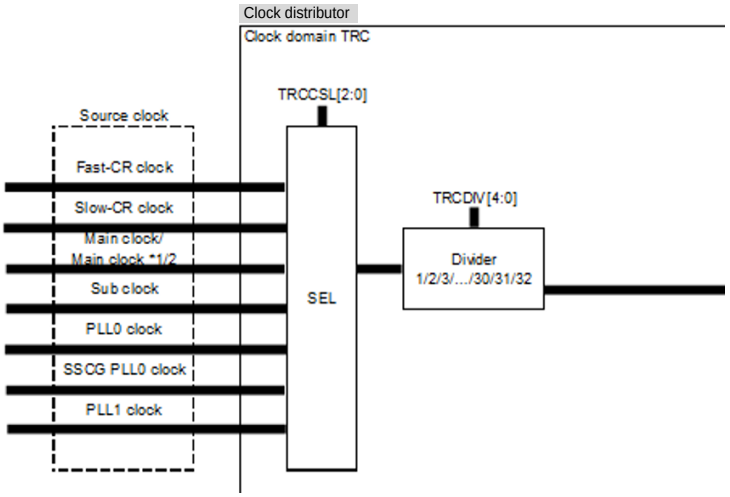
Page	Section	Change Results
129	5. Clock System 2.2. Clock Generator Configuration and Block Diagram	Changed the below: Figure 2-2 Clock Generator Block Diagram Error)  Correct) 

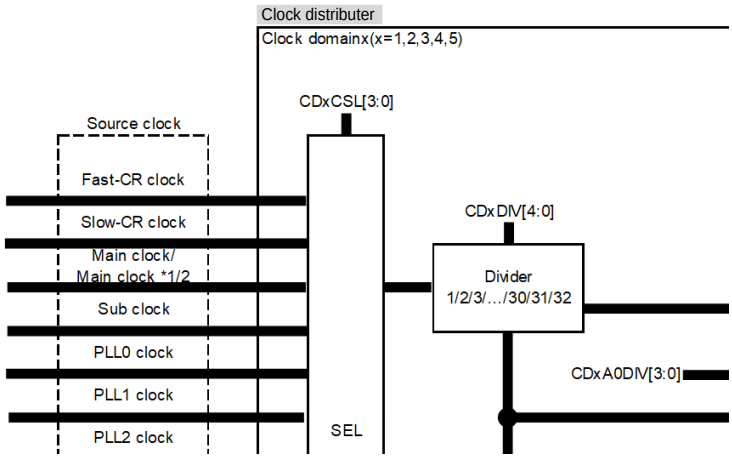
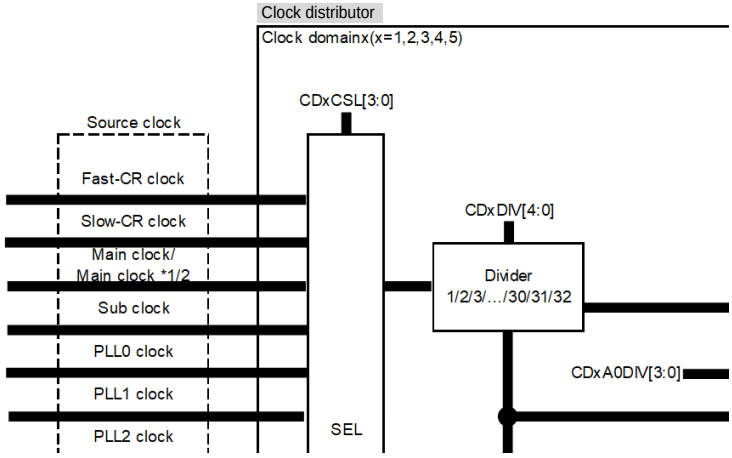
Page	Section	Change Results
130	CHAPTER 5: Clock System 2. Configuration and Block Diagram 2.2.1. PLL Clock Generator Configuration and Block Diagram	Revised the shading parts as below: Figure 2-3 PLLx (x=0,1,2,3)Clock Generator Block Diagram Error)  Correct) 

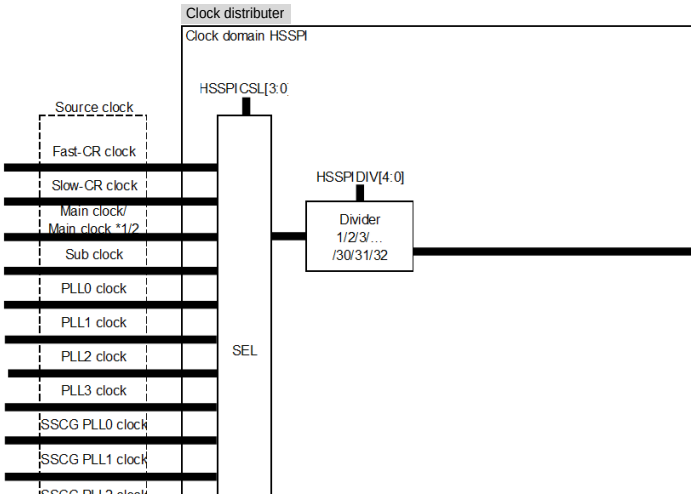
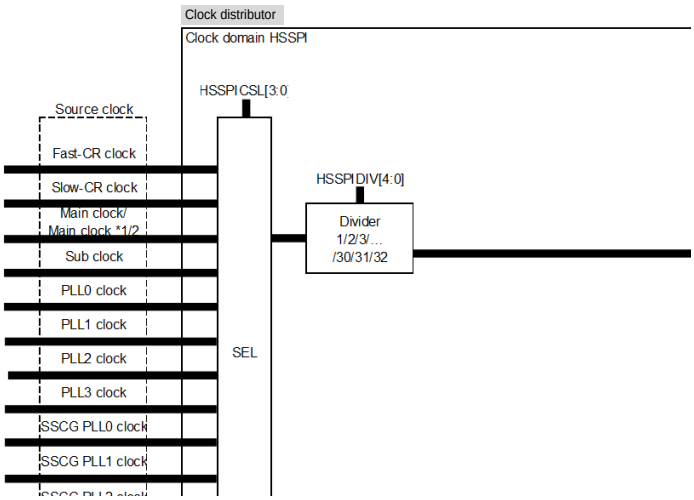
Page	Section	Change Results
131	CHAPTER 5: Clock System 2. Configuration and Block Diagram 2.2.2. SSCG PLL Clock Generator Configuration and Block Diagram	Revised the shading parts as below: Figure 2-4 SSCG PLLx (x=0,1,2,3)Clock Generator Block Diagram Error)  Correct) 

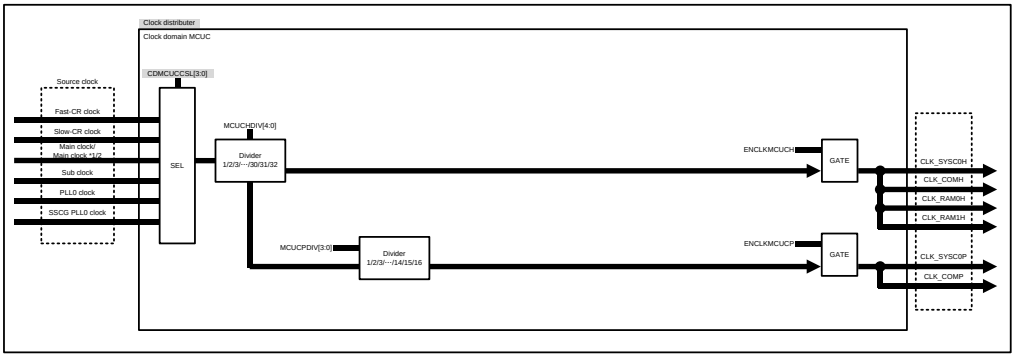
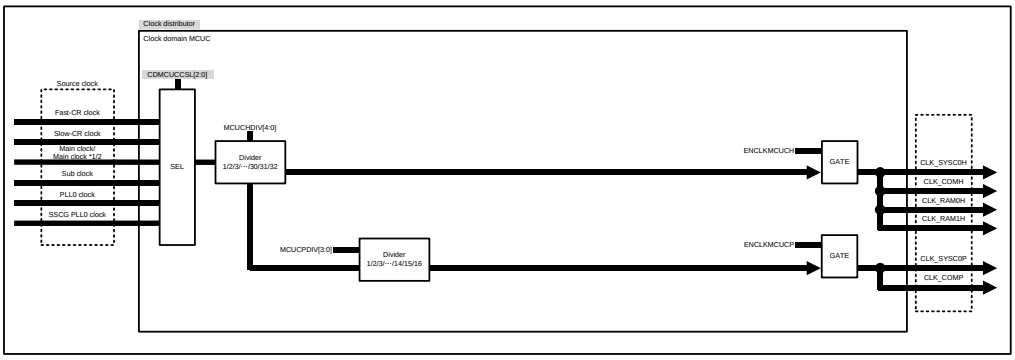
Page	Section	Change Results
132	CHAPTER 5: Clock System 2.3. Clock Distributor Configuration and Block Diagram	Revised the shading parts as below: Figure 2-5 Clock domain0 Block Diagram Error) 

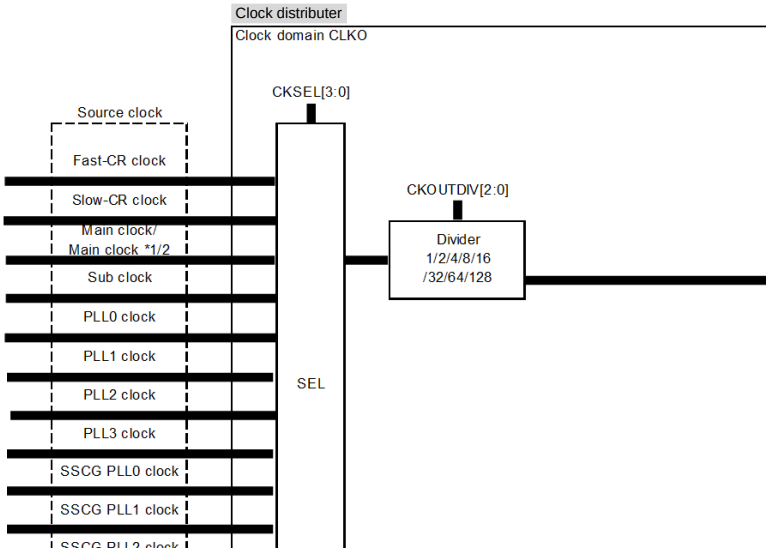
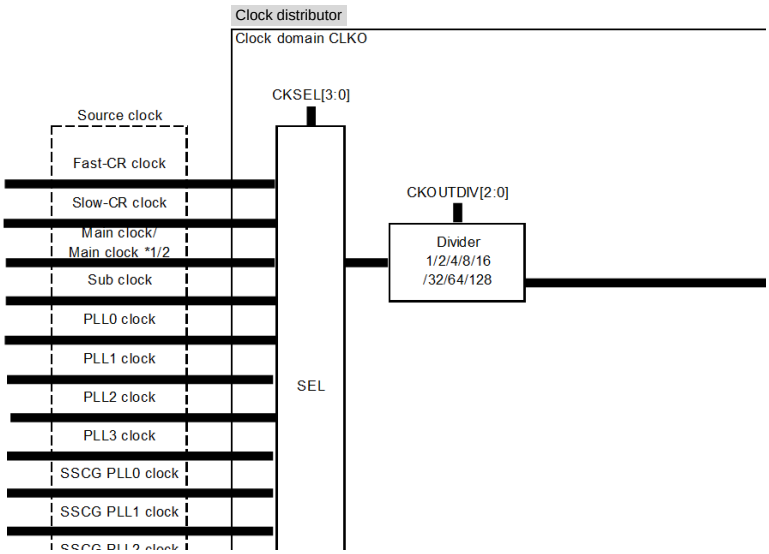
Page	Section	Change Results
		Correct) 

Page	Section	Change Results
133	CHAPTER 5: Clock System 2.3. Clock Distributor Configuration and Block Diagram	Revised the shading parts as below: Figure 2-6 Clock Domain TRC Block Diagram Error)  Correct) 

Page	Section	Change Results
133	CHAPTER 5: Clock System 2.3. Clock Distributor Configuration and Block Diagram	Revised the shading parts as below: Figure 2-7 Clock domain(x=1,2,3,4,5) Block Diagram Error)  Correct) 

Page	Section	Change Results
134	CHAPTER 5: Clock System 2.3. Clock Distributor Configuration and Block Diagram	Revised the shading parts as below: Figure 2-8 Clock Domain HSSPI Block Diagram Error)  Correct) 

Page	Section	Change Results
134	CHAPTER 5: Clock System 2.3. Clock Distributor Configuration and Block Diagram	Revised the shading parts as below: Figure 2-9 Clock Domain MCUC Block Diagram Error)  Correct) 

Page	Section	Change Results
135	CHAPTER 5: Clock System 2.3. Clock Distributor Configuration and Block Diagram	Revised the shading parts as below: Figure 2-10 Clock Domain CLKO Block Diagram Error)  Correct) 

Page	Section	Change Results
140, 141	CHAPTER 5: Clock System 3. Operations 3.3. Clock Distributor	Replaced the shading parts as below: Table 3-5 Clock Divider Configuration and Maximum Frequency Error)

Clock Domain	Internal Clock	Resources	Initial State	RUN/PSS	Maximum Frequency
CD0	CLK_CPU	CPU	1	Programmable	240MHz
	CLK_SHE	SHE	-	Not control	Same as CLK_CPU
	CLK_FCLK	TCFLASH WFLASH	1	Programmable	100MHz
	CLK_ATB	ATB	1	Programmable	120MHz
	CLK_DBG	DBG	1	Programmable	120MHz
	CLK_HPM	High Performance Matrix, Common Peripheral2	1	Programmable	240MHz
	CLK_HPM2		2	2	120MHz
	CLK_DMA	DMA	1	Programmable	240MHz
	CLK_MEMC	Memory & Config Group	1	Programmable	240MHz
	CLK_EXTBUS	Ext Bus Group	1	Programmable	120MHz
	CLK_SYSC1	MCU Config Group1	1	Programmable	120MHz
	CLK_HAPP0A0	Application Specific0 (HPM)	1	Programmable	120MHz
	CLK_HAPP0A1		1	Programmable	60MHz
	CLK_HAPP0B0	Application Specific1 (HPM)	1	Programmable	120MHz
	CLK_HAPP0B1		1	Programmable	60MHz
	CLK_LLPBM	Common Peripheral (LLPBM)	1	Programmable	240MHz
	CLK_LLPBM2	Common Peripheral0/1/2	2	2	120MHz
	CLK_LCP	Common Peripheral0/1	1	Programmable	120MHz
	CLK_LCP0	Common Peripheral0	1	Programmable	120MHz
	CLK_LCP0A		1	Programmable	120MHz
	CLK_LCP1	Common Peripheral1	1	Programmable	120MHz
	CLK_LCP1A		1	Programmable	120MHz
	CLK_LAPP0	Application Specific0 (LLPPBM)	1	Programmable	120MHz
	CLK_LAPP0A		1	Programmable	120MHz
	CLK_LAPP1	Application Specific1 (LLPPBM)	1	Programmable	120MHz
	CLK_LAPP1A		1	Programmable	120MHz
CDx (x=1/2/3/4/5)	CLK_CDx	(product specification)	1	Programmable	240MHz
	CLK_CDxA0		1	Programmable	120MHz
	CLK_CDxA1		1	Programmable	60MHz
	CLK_CDxB0		1	Programmable	120MHz
	CLK_CDxB1		1	Programmable	60MHz
CD_TRC	CLK_TRC	Debug Group	1	Programmable	120MHz
CD_HSSPI	CLK_HSSPI	HSSPI	1	Programmable	200MHz
CD_HSSPI	CLK_SYSC0H	MCU Config Group (AHB)	1	Programmable	120MHz
	CLK_COMH	Communication Peripheral (AHB)	1	Programmable	120MHz
	CLK_RAM0H	Back-up RAM0	1	Programmable	120MHz
	CLK_RAM1H	Back-up RAM1	1	Programmable	120MHz
	CLK_SYSC0P	MCU Config Group0 (APB)	1	Programmable	60MHz
	CLK_COMP	Communication Peripheral (APB)	1	Programmable	60MHz
CD_CLKO	CLK_CLKO	Clock output function	1	Programmable	240MHz

Page	Section	Change Results

Correct)

Clock Domain	Internal Clock	Resources	Parent Clock	Initial State	RUN/PSS	Maximum Frequency
CD0	CLK_CPU	CPU	Source clock selected by CD0CSL[2:0]	1	Programmable	240MHz
	CLK_SHE	SHE	CLK_CPU	1	No div of CLK_CPU	Same as CLK_CPU
	CLK_FCLK	TCFLASH WFLASH	CLK_CPU	4	Programmable	100MHz
	CLK_ATB	ATB		1	Programmable	120MHz
	CLK_DBG	DBG	CLK_CPU	1	Programmable	120MHz
	CLK_HPM	High Performance Matrix, Common Peripheral2	CLK_ATB	1	Programmable	240MHz
	CLK_HPM2		CLK_CPU	2	2	120MHz
	CLK_DMA	DMA	CLK_HPM	1	Programmable	240MHz
	CLK_MEMC	Memory & Config Group	CLK_HPM	1	Programmable	240MHz
	CLK_EXTBUS	Ext Bus Group	CLK_HPM	1	Programmable	120MHz
	CLK_SYSC1	MCU Config Group1	CLK_HPM	1	Programmable	120MHz
	CLK_HAPP0A0	Application Specific0 (HPM)	CLK_HPM	1	Programmable	120MHz
	CLK_HAPP0A1		CLK_HAPP0A0	1	Programmable	60MHz
	CLK_HAPP0B0	Application Specific1	CLK_HPM	1	Programmable	120MHz
	CLK_HAPP0B1	(HPM)	CLK_HAPP0B0	1	Programmable	60MHz
	CLK_LLPBM	Common Peripheral (LLPBM)	CLK_CPU	1	Programmable	240MHz
	CLK_LLPBM2	Common Peripheral0/1/2	CLK_LLPBM2	2	2	120MHz
	CLK_LCP	Common Peripheral0/1	CLK_CPU	1	Programmable	120MHz
	CLK_LCP0	Common Peripheral0	CLK_CPU	1	Programmable	120MHz
	CLK_LCP0A		Source clock selected by LCP0ACSEL	1	Programmable	120MHz
	CLK_LCP1	Common Peripheral1	CLK_CPU	1	Programmable	120MHz
	CLK_LCP1A		Source clock selected by LCP1ACSEL	1	Programmable	120MHz
	CLK_LAPP0	Application Specific0 (LLPPBM)	CLK_CPU	1	Programmable	120MHz
	CLK_LAPP0A		Source clock selected by LAPP0ACSEL	1	Programmable	120MHz
	CLK_LAPP1	Application Specific1 (LLPPBM)	CLK_CPU	1	Programmable	120MHz
	CLK_LAPP1A		Source clock selected by LAPPAACSEL	1	Programmable	120MHz
CDx (x=1/2/3/4/5)	CLK_CDx	(product specification)	Source clock selected by CDxCSL	1	Programmable	240MHz
	CLK_CDxA0		CLK_CDx	1	Programmable	120MHz
	CLK_CDxA1		CLK_CDxA0	1	Programmable	60MHz
	CLK_CDxB0		CLK_CDx	1	Programmable	120MHz
	CLK_CDxB1		CLK_CDxB0	1	Programmable	60MHz
CD_TRC	CLK_TRC	Debug Group	Source clock selected by TRCCSL	1	Programmable	120MHz
CD_HSSPI	CLK_HSSPI	HSSPI	Source clock selected by HSSPICSL	1	Programmable	200MHz
CD_MCUC	CLK_SYSC0H	MCU Config Group (AHB)	Source clock selected by CDMCUCCSL	1	Programmable	120MHz
	CLK_COMH	Communication Peripheral (AHB)	Source clock selected by CDMCUCCSL	1	Programmable	120MHz
	CLK_RAM0H	Back-up RAM0	Source clock selected by CDMCUCCSL	1	Programmable	120MHz
	CLK_RAM1H	Back-up RAM1	Source clock selected by CDMCUCCSL	1	Programmable	120MHz
	CLK_SYSC0P	MCU Config Group0 (APB)	CLK_SYSC0H	1	Programmable	60MHz
	CLK_COMP	Communication Peripheral (APB)	CLK_SYSC0H	1	Programmable	60MHz
CD_CLKO	CLK_CLKO	Clock output function	Source clock selected by CKSEL	1	Programmable	240MHz

Page	Section	Change Results
142, 143	CHAPTER 5: Clock System 3. Operations 3.4. Clock Gear	Revised the shading parts as below: Error) ■Start step configuration ■Step loop configuration ■Step width configuration Correct) ■Start step configuration − Setting to select the range ("START STEP" to STEP 63) during Gear operation that defined by Start Step configuration (xxxCGSSN). − It is selectable from STEP 0 to 63. − At Gear Up, the Gear operation starts from the selected STEP 'n' and stops at STEP 63. (Refer to Step Width Configuration for the increment width of 'n' after 1 step operation.) − At Gear Down, the Gear operation starts from STEP 63 and stops at STEP 'n' the selected. (Refer to Step Width Configuration for the decrement width of 'n' after 1 step operation.) ■Step loop configuration − STEP: − Loop that is repeated a defined number of times when the Loop configuration (xxxCGLP). − It is configurable from 1 to 256 loops. − 1 step is xxxCGLP <loop> = xxxCGLP x 64 <cycle>. − LOOP: − The minimum operation unit that configurates gear operation. − 1 Loop fixed at 64 <cycle>. ■Step width configuration − The width defined by Step Width configration (xxxCGSTP) which "n" of STEP "n" is increment / decrement (when Gear Up / Gear Down) after 1 STEP execution. − Step width can be set to 1 to 4. − At Gear Up, the width to increment 'n' after 1 step of STEP 'n'. (In the case of Width = 2, STEP 0, STEP 2, STEP 4 ...). − At Gear Down, the width to decrement 'n' after 1 step of STEP 'n'. (In the case of Width 4, STEP 63, STEP 59, STEP 55 ...) ***Note:*** *STEP 0 to 63:* *STEP 0 to 63: Clock output patterns of Gear operation.* *As the number of step increases, the clock output increases.* *STEP 0 to 1: clock/loop* *STEP 1 to 2: clock/loop* *.* *.* *.* *STEP 63 to 64 clock/loop (Clock output for all cycles in loop)* *# Refer to Figure 3-1 for details pattern.* *The length of STEP 0 to 63 (the number of Loop repetitions) is determined by the setting of STEP (xxxCGLP x 64 <cycle>).*

Page	Section	Change Results
160, 163, 166, 169	5. Clock System 5.1.5 PLL0 Clock Gear Control Register (SYSC_PLL0CGCNTR) 5.1.6 PLL1 Clock Gear Control Register (SYSC_PLL1CGCNTR) 5.1.7 PLL2 Clock Gear Control Register (SYSC_PLL2CGCNTR) 5.1.8 PLL3 Clock Gear Control Register (SYSC_PLL3CGCNTR)	Revised the shading parts as below: Error) Note: - This bit is cleared after starting clock gear operation. Correct) Note: - This bit is cleared after completion of clock gear operation.
162, 165, 168	5. Clock System 5.1.6 PLL1 Clock Gear Control Register (SYSC_PLL1CGCNTR) 5.1.7 PLL2 Clock Gear Control Register (SYSC_PLL2CGCNTR) 5.1.8 PLL3 Clock Gear Control Register (SYSC_PLL3CGCNTR)	Revised the shading parts as below: Error) Notes: - These bits have to be set before PLL clock enable setting. - Changing these bits are prohibited after PLL clock enable setting. Correct) Notes: - In the PLL clock stop or PLL clock stabilization wait, return a "00" status. - if clock gear not used, the status is "X" (undefined).

Page	Section	Change Results
171, 174, 177, 180	5. Clock System 5.1.9 SSCG PLL0 Clock Gear Control Register (SYSC_SSCG0CGCNTR) 5.1.10 SSCG PLL1 Clock Gear Control Register (SYSC_SSCG1CGCNTR) 5.1.11 SSCG PLL2 Clock Gear Control Register (SYSC_SSCG2CGCNTR) 5.1.12 SSCG PLL0 Clock Gear Control Register (SYSC_SSCG3CGCNTR)	Revised the shading parts as below: Error) Notes: - These bits have to be set before PLL clock enable setting. - Changing these bits are prohibited after PLL clock enable setting. Correct) Notes: - In the SSCG PLL clock stop or SSCG PLL clock stabilization wait, return a "00" status. - if clock gear not used, the status is "X" (undefined).
172, 175, 178, 181	5. Clock System 5.1.9 SSCG PLL0 Clock Gear Control Register (SYSC_SSCG0CGCNTR) 5.1.10 SSCG PLL1 Clock Gear Control Register (SYSC_SSCG1CGCNTR) 5.1.11 SSCG PLL2 Clock Gear Control Register (SYSC_SSCG2CGCNTR) 5.1.12 SSCG PLL0 Clock Gear Control Register (SYSC_SSCG3CGCNTR)	Revised the shading parts as below: Error) Note: - This bit is cleared after starting clock gear operation. Correct) Note: - This bit is cleared after completion of clock gear operation.

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>184</td><td>5. Clock System
6. Usage Precautions</td><td>Added the shading parts as below:

Correct)
The implementation of PLL1-3 and SSCG0-3 is dependent of product variation.
Please refer to the TRM of the each product.
If you select a clock that does not exist, Fast CR clock is selected.
The following registers for non-existent clocks no longer exist and become the reserve area.

SYSC_PLLxCGCNTR
SYSC_SSCGxCGCNTR</td></tr>
<tr><td>199</td><td>CHAPTER 6:Low-power Consumption

3.4. Profile</td><td>Changed the below:
Error)<table><tr><td>Clock Power</td><td>The set source clock of clock domain 0 was any clock other than the high-speed CR clock, with the RUN profile that is set at the return time from the PSS that powered OFF PD3 (when a Security reevaluation is necessary).</td><td>PSS</td><td>SYSC0_SYSPSSPEFR:PEF9</td></tr></table>
Correct)<table><tr><td>Clock Power</td><td>The set source clock of clock domain 0 was any clock other than the high-speed CR clock, with the RUN profile that is set at the return time from the PSS that powered OFF PD2 (when a Security reevaluation is necessary).</td><td>PSS</td><td>SYSC0_SYSPSSPEFR:PEF9</td></tr></table></td></tr>
<tr><td>213</td><td>CHAPTER 6: 4.1.RUN Profile Update</td><td>Revised the shading parts as below:

Error)
2. The circuit copies the RUN profile contents to the APPLIED profile.

Correct)
2. The circuit copies the RUN profile contents to the APP profile.</td></tr>
<tr><td>215</td><td>CHAPTER 6: 4.2. Transition from RUN to PSS</td><td>Revised the shading parts as below:

Error)
2. The circuit copies the PSS profile contents to the APPLIED profile.

Correct)
2. The circuit copies the PSS profile contents to the APP profile.</td></tr>
</table>

Page	Section	Change Results
		Revised the shading parts as below: Error) - When returning from PSS mode to Run mode, Main_OSC must be valid. Therefore, be sure to enable Main_OSC setting of the RUN mode profile before entering the PSS mode. Correct) - When returning to Run mode, Main_OSC must be valid. Therefore, be sure to enable Main_OSC setting of the RUN mode profile before entering the PSS mode.
218	CHAPTER 6: 4.3. Transition from PSS to RUN	Revised the shading parts as below: Error) ■ Peripheral interrupt (PD1): External interrupt, RTC, NMI, low-voltage detection *1, HW-WDT *1, CSV *1 Correct) ■ Peripheral interrupt (PD1): External interrupt, RTC, NMI, low-voltage detection *1, HW-WDT *1
219	CHAPTER 6: 4.3. Transition from PSS to RUN	Revised the shading parts as below: Error) 2. The circuit copies the RUN profile contents to the APPLIED profile. Correct) 2. The circuit copies the RUN profile contents to the APP profile.

Page	Section	Change Results
221, 222	CHAPTER 6: 5. Registers	Revised the shading parts as below: Error) Notes: − Protection is applied to the SYSC register. To write to the register, the protection key must first be released. Invalid written data is discarded. The following shows an example of the procedure for accessing the SYSC register. 1. The accessing master releases the protection key. 2. The master that released the protection key writes to the SYSC register. 3. The hardware enables the protection key. Repeat steps 1 to 3 until completing the required settings. − The following shows protection effective conditions. 1. The protection target register was write-accessed from the master that released the protection key. 2. The protection target register was write-accessed from anything other than the master that released the protection key. Correct) Notes: − Protection is applied to the SYSC register. To write to the register, the protection key must first be unlocked. Invalid written data is discarded. The following shows an example of the procedure for accessing the SYSC register. 1. The accessing master unlocks the protection key. 2. The master that unlocked the protection key writes to the SYSC register. 3. The hardware locks the protection key. Repeat steps 1 to 3 until completing the required settings. − The following shows protection effective conditions. 1. The protection target register was write-accessed from the master that unlocked the protection key. 2. The protection target register was write-accessed from anything other than the master that unlocked the protection key.

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>238</td><td>CHAPTER 6:
5.1. Protection Register Group (SYSC0)
5.1.1. Protection Key Setting Register (SYSC0_PROTKEYR)</td><td>Revised the shading parts as below:

Error)
The SYSC0_PROTKEYR register has settings for releasing the protection of SYSC0 registers.

[bit31:0] PROTKEY: Protection Release Setting Bits
These bits compose the register for releasing protection keys.

Lock release

Correct)
The SYSC0_PROTKEYR register has settings for unlocking the protection of SYSC0 registers.

[bit31:0] PROTKEY: Protection Unlock Setting Bits
These bits compose the register for unlocking protection keys.

Unlock</td></tr>
<tr><td>293</td><td>CHAPTER 6:
5. Registers
5.3.11. PSS Regulator Setting Register (SYSC0_PSSREGCFGR)</td><td>Added the shading parts as below:

Error)

[bit7] RMSEL: Regulator Mode Setting Bit
This bit set the regulator mode setting.
<table><tr><th>Bit</th><th>Description</th></tr><tr><td>0</td><td>Main mode</td></tr><tr><td>1</td><td>Standby mode</td></tr></table>
Correct)

[bit7] RMSEL: Regulator Mode Setting Bit
This bit set the regulator mode setting.
<table><tr><th>Bit</th><th>Description</th></tr><tr><td>0</td><td>Main mode</td></tr><tr><td>1</td><td>Standby mode</td></tr></table>
Main mode is normal operation in regulator.
Standby mode is low power operation in regulator.
In `Standby mode`, current consumption in PSS mode can be reduced than Main mode by setting this bit.

Note: Specific value of current consumption depend on each product.
Please refer to Datasheet for any device-specific information.</td></tr>
</table>

Page	Section	Change Results
295	CHAPTER 6: 5. Registers 5.4.1. APP Power Domain Setting Register (SYSC0_APPPDCFGR)	Revised the shading parts as below: Error) The SYSC0_APPPDCFGR register sets the ON/OFF of the power domain which is going to be updated. Correct) The SYSC0_APPPDCFGR register indicates the ON/OFF of the power domain which is going to be updated.
308	CHAPTER 6: 5. Registers 5.4.7. APP SSCGx Control Register 0 (SYSC0_APPSSCGxCNTR0)	Revised the shading parts as below: Error) The SYSC0_APPSSCGxCNTR0 register sets the division, multiplication rate, etc. for SSCG PLLx. Correct) The SYSC0_APPSSCGxCNTR0 register indicates the set values of the division, multiplication rate, etc. for SSCG PLLx to be updated.
311	CHAPTER 6: 5. Registers 5.4.8. APP SSCGx Control Register 1 (SYSC0_APPSSCGxCNTR1)	Revised the shading parts as below: Error) The SYSC0_APPSSCGxCNTR1 register sets modulation enable, the modulation mode, etc. for SSCG PLLx. Correct) The SYSC0_APPSSCGxCNTR1 register indicates the set values for modulation enable, the modulation mode, etc. for SSCG PLLx to be updated.
314	CHAPTER 6: 5. Registers 5.4.9. APP Low-voltage Detection Setting Register (SYSC0_APPLVDCFGR)	Revised the shading parts as below: Error) The SYSC0_APPLVDCFGR register sets each internal low-voltage detection. Correct) The SYSC0_APPLVDCFGR register indicates the set values for each internal low-voltage detection to be updated.

Page	Section	Change Results
319	CHAPTER 6: 5. Registers 5.4.10. APP Clock Supervisor Setting Register (SYSC0_APPCSVCFGR)	Revised the shading parts as below: Error) The SYSC0_APPCSVCFGR register sets whether to enable or disable operation of each clock supervisor. Correct) The SYSC0_APPCSVCFGR register indicates the value for setting whether to enable or disable operation of each clock supervisor to be updated.
322	CHAPTER 6: 5. Registers 5.4.11. APP Regulator Setting Register (SYSC0_APPREGCFGR)	Revised the shading parts as below: Error) The SYSC0_APPREGCFGR register sets the regulator mode and output voltage. Correct) The SYSC0_APPREGCFGR register indicates the set values for the regulator mode and output voltage to be updated.
352	CHAPTER 6: 5. Registers 5.5.11. STS Regulator Setting Register (SYSC0_STSREGCFGR)	Deleted the shading parts as below: Delete) Note: - The initial value and setting value of these bits can be set with the external pin.

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>376</td><td>CHAPTER 6: 5.7.1. System Special Setting Register (SYSC0_SPECFGR)</td><td>
Revised the shading parts as below:

Error)

[bit22] IO3RSTC：I/O3V Reset configuring Bit
This bit sets 3V I/O reset control.
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>I/O reset released</td></tr>
<tr><td>1</td><td>I/O reset enabled</td></tr>
</table>

[bit21] IO35RSTC：I/O5/3V Reset configuring Bit
This bit sets 5/3V I/O reset control.
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>I/O reset released</td></tr>
<tr><td>1</td><td>I/O reset enabled</td></tr>
</table>

Correct)

[bit22] IO3RSTC：I/O3V Reset configuring Bit
This bit sets 3V I/O reset control.
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>I/O reset released</td></tr>
<tr><td>1</td><td>I/O reset asserted</td></tr>
</table>

[bit21] IO35RSTC：I/O5/3V Reset configuring Bit
This bit sets 5/3V I/O reset control.
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>I/O reset released</td></tr>
<tr><td>1</td><td>I/O reset asserted</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>396</td><td>CHAPTER 6:
5.10. Protection Register Group (SYSC0)
5.10.1. Protection Key Setting Register (SYSC1_PROTKEYR)</td><td>Revised the shading parts as below:

Error)
The SYSC1_PROTKEYR register has settings for releasing the protection of SYSC1 registers.

[bit31:0] PROTKEY: Protection Release Setting Bits
These bits compose the register for releasing protection keys.

Write
<table>
<tr><th>bit31:0</th><th>Description</th></tr>
<tr><td>0x5CAC_CE55</td><td>Lock release</td></tr>
<tr><td>Other than above</td><td>Invalid</td></tr>
</table>
Correct)
The SYSC1_PROTKEYR register has settings for unlocking the protection of SYSC1 registers.

[bit31:0] PROTKEY: Protection Unlock Setting Bits
These bits compose the register for unlocking protection keys.

Write
<table>
<tr><th>bit31:0</th><th>Description</th></tr>
<tr><td>0x5CAC_CE55</td><td>Unlock</td></tr>
<tr><td>Other than above</td><td>Invalid</td></tr>
</table></td></tr>
<tr><td>616</td><td>CHAPTER 8:
2. Configuration
2.2. Detector for Stopped Clock Only</td><td>Revised the shading parts as below:

Clock stop Detection Judgment

Error)
The clock stop detection judgment judges the presence of rising edge of the reference clock within the period of the clock that is divided by the reference clock.

Correct)
The clock stop detection judgment judges the presence of rising edge of the monitored clock within the period of the clock that is divided by the reference clock.</td></tr>
</table>

Page	Section	Change Results
618	CHAPTER 8: 3. Explanation of Operation	Revised the shading parts as below: PLL Clock Supervisor Error) ■PLLm clock supervisor setting register 0 (SYSC_CSVPLLmCFGR0) and PLLm clock supervisor setting register 1 (SYSC_CSVPLLmCFGR1) are used for settings. ■Reset generation or interrupt generation can be selected with the judgment selection bit (JDGSEL). ■Upon detecting an abnormality of the PLL clock and generating a reset, the PLL clock supervisor can confirm it with the PLL clock supervisor reset detection bit (CSVPR) in the user reset factor register (SYSC_RSTCAUSEUR). Correct) ■PLLm clock supervisor setting register 0 (SYSC_CSVPLLmCFGR0) and PLLm clock supervisor setting register 1 (SYSC_CSVPLLmCFGR1) are used for settings. ■In case of PLL0, If PLL0 clock is selected for clock domain 0 or MCUC clock domain when abnormal state is detected by CSV, CSV generates a reset. CSV generates an interrupt under other conditions. ■In case of PLL1, PLL2 or PSS3, reset generation or interrupt generation can be selected with the judgment selection bit (JDGSEL). ■Upon detecting an abnormality of the PLL clock and generating a reset, the PLL clock supervisor can confirm it with the PLL clock supervisor reset detection bit (CSVPR) in the user reset factor register (SYSC_RSTCAUSEUR). SSCG PLL Clock Supervisor Error) ■SSCG PLLn clock supervisor setting register 0 (SYSC_CSVSPnCFGR0) and SSCG PLLn clock supervisor setting register 1 (SYSC_CSVSPnCFGR1) are used for settings. ■Upon detecting an abnormality of the SSCG PLL clock and generating a reset, the SSCG PLL clock supervisor can confirm it with the SSCG PLL clock supervisor reset detection bit (CSVSR) in the user reset factor register (SYSC_RSTCAUSEUR). Correct) ■SSCG PLLn clock supervisor setting register 0 (SYSC_CSVSPnCFGR0) and SSCG PLLn clock supervisor setting register 1 (SYSC_CSVSPnCFGR1) are used for settings. ■In case of PLL0, If SSCG clock is selected for clock domain 0 or MCUC clock domain when abnormal state is detected by CSV, CSV generates a reset. CSV generates an interrupt under other conditions. ■In case of SSCG1, SSCG or SSCG3, reset generation or interrupt generation can be selected with the judgment selection bit (JDGSEL). ■Upon detecting an abnormality of the SSCG PLL clock and generating a reset, the SSCG PLL clock supervisor can confirm it with the SSCG PLL clock supervisor reset detection bit (CSVSR) in the user reset factor register (SYSC_RSTCAUSEUR).

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>621</td><td>CHAPTER 8:
5. Registers</td><td>Revised the shading parts as below:

Error)
This section lists the clock supervisor registers.
The clock supervisor registers are as follows:

Correct)
This section lists the clock supervisor registers.
The clock supervisor registers are as follows:
These registers are subject to the SYSC0 protection register.
The response to any writing done to reserved bit is a bus error.</td></tr>
<tr><td>627</td><td>CHAPTER 8:
5. Registers
5.2. Main Clock Supervisor Setting Register 1 (SYSC_CSVMOCFGR1)</td><td>Revised the shading parts as below:

[bit16] JDGSEL: Judgment selection bit

Error)

This bit selects an additional condition under which a detected abnormal state (by CSV) leads to reset generation. Without the signalling of an abnormal state by the CSV, neither reset nor interrupt are generated. If the signal of an abnormal state occurs, the reaction depends on the condition which is selected by JDGSEL. If the condition as defined by the bit value description is true, the abnormal state leads to a reset, if it is false, an interrupt will be generated.

<table><tr><th>Bit</th><th>Description</th></tr><tr><td>0</td><td>Main clock is selected for clock domain 0 or for MCUC clock domain or for software watchdog timer.</td></tr><tr><td>1</td><td>Main clock is selected for clock domain 0 or for MCUC clock domain.</td></tr></table>- Example: When JDGSEL = 1 and main clock is neither selected for clock domain 0 nor for MCUC, the selected condition is false and an abnormal state leads to interrupt generation.
-

Correct)

This bit selects a reset condition when abnormal state is detected by CSV.

<table><tr><th>Bit</th><th>Description</th></tr><tr><td>0</td><td>Main clock is selected for clock domain 0 , MCUC clock domain or software watchdog timer.</td></tr><tr><td>1</td><td>Main clock is selected for clock domain 0 or MCUC clock domain.</td></tr></table>- If it is not the above condition when an abnormal state is detected by CSV, CSV generates interrupt.</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>630</td><td>CHAPTER 8:
5. Registers
5.4. Sub Clock Supervisor Setting Register 1 (SYSC_CSVSOCFGR1)</td><td>Revised the shading parts as below:

[bit16] JDGSEL: Judgment selection bit

Error)

This bit selects an additional condition under which a detected abnormal state (by CSV) leads to reset generation. Without the signalling of an abnormal state by the CSV, neither reset nor interrupt are generated. If the signal of an abnormal state occurs, the reaction depends on the condition which is selected by JDGSEL. If the condition as defined by the bit value description is true, the abnormal state leads to a reset, if it is false, an interrupt will be generated.

<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Main clock is selected for clock domain 0 or for MCUC clock domain or for software watchdog timer.</td></tr>
<tr><td>1</td><td>Main clock is selected for clock domain 0 or for MCUC clock domain.</td></tr>
</table>
- Example: When JDGSEL = 1 and main clock is neither selected for clock domain 0 nor for MCUC, the selected condition is false and an abnormal state leads to interrupt generation.
-

Correct)

This bit selects a reset condition when abnormal state is detected by CSV.

<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Main clock is selected for clock domain 0 , MCUC clock domain or software watchdog timer.</td></tr>
<tr><td>1</td><td>Main clock is selected for clock domain 0 or MCUC clock domain.</td></tr>
</table>
- If it is not the above condition when an abnormal state is detected by CSV, CSV generates interrupt.</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>634</td><td>CHAPTER 8:
5. Registers
5.6. PLLm Clock Supervisor Setting Register 1 (SYSC_CSVPLLmCFGR1)</td><td>Revised the shading parts as below:

[bit16] JDGSEL: Judgment selection bit

Error)

This bit selects an additional condition under which a detected abnormal state (by CSV) leads to reset generation. Without the signalling of an abnormal state by the CSV, neither reset nor interrupt are generated. If the signal of an abnormal state occurs, the reaction depends on the condition which is selected by JDGSEL. If the condition as defined by the bit value description is true, the abnormal state leads to a reset, if it is false, an interrupt will be generated.

<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Main clock is selected for clock domain 0 or for MCUC clock domain or for software watchdog timer.</td></tr>
<tr><td>1</td><td>Main clock is selected for clock domain 0 or for MCUC clock domain.</td></tr>
</table>
- Example: When JDGSEL = 1 and main clock is neither selected for clock domain 0 nor for MCUC, the selected condition is false and an abnormal state leads to interrupt generation.
-

Correct)

This bit selects a reset condition when abnormal state is detected by CSV.

<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Main clock is selected for clock domain 0 , MCUC clock domain or software watchdog timer.</td></tr>
<tr><td>1</td><td>Main clock is selected for clock domain 0 or MCUC clock domain.</td></tr>
</table>
- If it is not the above condition when an abnormal state is detected by CSV, CSV generates interrupt.</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>638</td><td>CHAPTER 8:
5. Registers
5.8. SSCG PLLn Clock Supervisor Setting Register 1 (SYSC_CSVSPnCFGR1)</td><td>Revised the shading parts as below:

[bit16] JDGSEL: Judgment selection bit

Error)

This bit selects an additional condition under which a detected abnormal state (by CSV) leads to reset generation. Without the signalling of an abnormal state by the CSV, neither reset nor interrupt are generated. If the signal of an abnormal state occurs, the reaction depends on the condition which is selected by JDGSEL. If the condition as defined by the bit value description is true, the abnormal state leads to a reset, if it is false, an interrupt will be generated.

<table><tr><th>Bit</th><th>Description</th></tr><tr><td>0</td><td>Main clock is selected for clock domain 0 or for MCUC clock domain or for software watchdog timer.</td></tr><tr><td>1</td><td>Main clock is selected for clock domain 0 or for MCUC clock domain.</td></tr></table>- Example: When JDGSEL = 1 and main clock is neither selected for clock domain 0 nor for MCUC, the selected condition is false and an abnormal state leads to interrupt generation.
-

Correct)

This bit selects a reset condition when abnormal state is detected by CSV.

<table><tr><th>Bit</th><th>Description</th></tr><tr><td>0</td><td>Main clock is selected for clock domain 0 , MCUC clock domain or software watchdog timer.</td></tr><tr><td>1</td><td>Main clock is selected for clock domain 0 or MCUC clock domain.</td></tr></table>- If it is not the above condition when an abnormal state is detected by CSV, CSV generates interrupt.</td></tr>
<tr><td>646</td><td>CHAPTER 8:
5. Registers
5.12. Clock Supervisor Test Register (SYSC_CSVTESTR)</td><td>Added the shading parts as below:

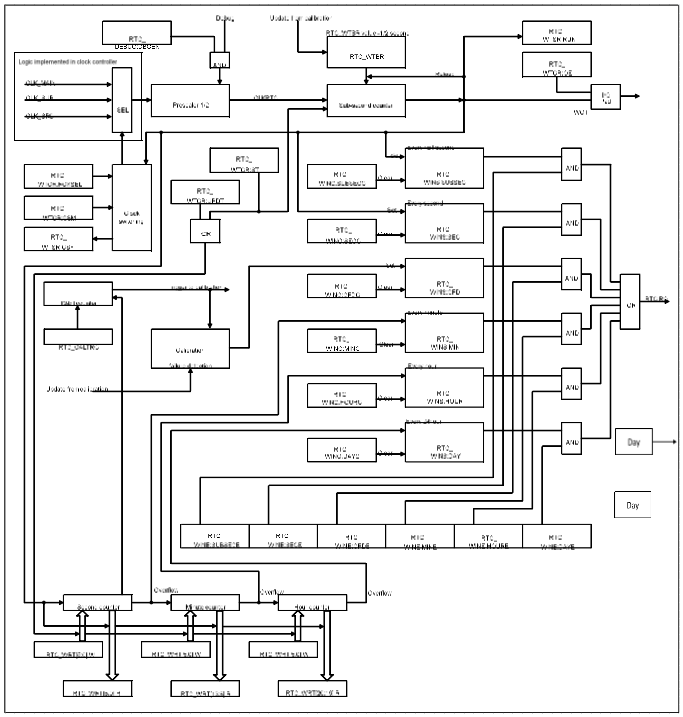
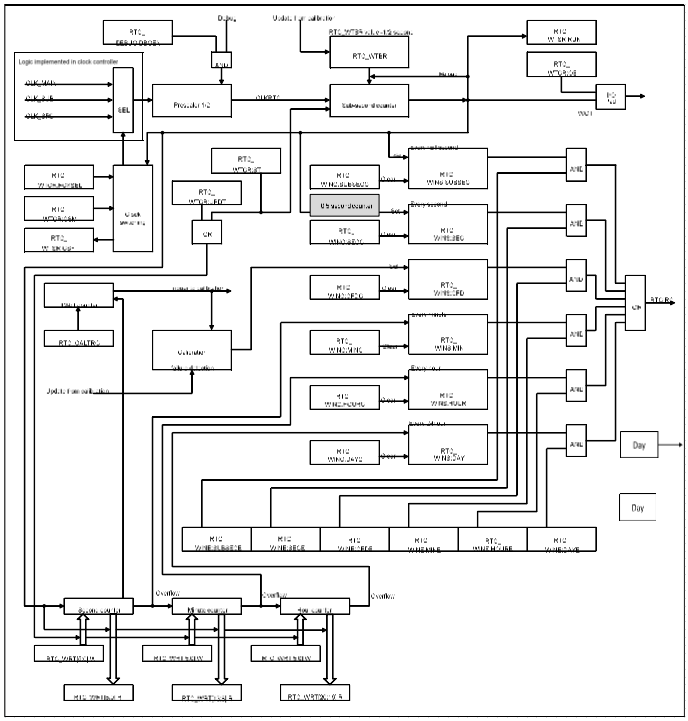
Correct)
[bit15:12] Reserved: Reserved bits</td></tr>
</table>

Page	Section	Change Results
652	CHAPTER 9: Source Clock Timer 1. Overview	Removed the shading parts as below: Error) High-Speed CR Clock Timer The high-speed CR clock timer has an oscillation stabilization wait time of 0.24 to 0.48 ms. Low-Speed CR Clock Timer The low-speed CR clock timer has an oscillation stabilization wait time of 0.43 to 1.28 ms. Main Clock Timer The main clock timer has an initial value of 8.19 ms (at 4 MHz) for the oscillation stabilization wait time. Sub Clock Timer The sub clock timer has an initial value of 8.0 ms or 16.0 ms (at 32 kHz) for the oscillation stabilization wait time. Correct) High-Speed CR Clock Timer The high-speed CR clock timer has an oscillation stabilization wait time. Low-Speed CR Clock Timer The low-speed CR clock timer has an oscillation stabilization wait time. Main Clock Timer The main clock timer has an initial value for the oscillation stabilization wait time. Sub Clock Timer The sub clock timer has an initial value for the oscillation stabilization wait time.
655	CHAPTER 9: Source Clock Timer 3. Explanation of Operation	Revised the shading parts as below: Error) ■If data is written to a trigger register, timer control register, timer compare prescaler register, ■interrupt enable register, or interrupt clear register during the oscillation stabilization wait operation, release the sequence protection in privileged mode. Correct) ■If data is written to a trigger register, timer compare prescaler register, ■interrupt enable register, or interrupt clear register during the oscillation stabilization wait operation, release the sequence protection in privileged mode.

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>659</td><td>CHAPTER 9: Source Clock Timer
5. Register</td><td>Deleted the shading parts as below:

Error)
<table><tr><th>Main Oscillation Stabilization Wait Time (at 4 MHz)</th><th>Initial Value of SYSC_MOCTCPR</th></tr><tr><td>8.19 [ms]</td><td>00000000_00000110_00000010_00000000</td></tr><tr><td>16.38 [ms]</td><td>00000000_00000110_00000100_00000000</td></tr></table>
Correct)
<table><tr><th>Main Oscillation Stabilization Wait Time (at 4 MHz)</th><th>Initial Value of SYSC_MOCTCPR</th></tr><tr><td>8.19 [ms]</td><td>00000000_00000110_00000010_00000000</td></tr></table></td></tr>
<tr><td>696</td><td>CHAPTER 9: Source Clock Timer
5. Registers
5.22. Sub Clock Timer Status Register (SYSC_SOCTSTR)</td><td>Added the shading bit description list.

[bit1] TST: Timer Status Bit
This bit indicates the operation status of the sub clock timer.
<table><tr><th>Bit</th><th>Description</th></tr><tr><td>0</td><td>Timer stopped</td></tr><tr><td>1</td><td>Timer operating</td></tr></table></td></tr>
</table>

Page	Section	Change Results
704	CHAPTER 10:Real Time Clock 2. Configuration and Block Diagram	Changed the Below: Figure 2-2 RTC Timer Module Diagram Error)  Correct) 

Page	Section	Change Results
713	CHAPTER 10:Real Time Clock 4. Registers	Revised the shading parts as below: Note: Error) All RTC registers and counters are initialized only by the power-on reset, low-voltage reset, external reset and Clock Supervisor (CSV) reset. Correct) All RTC registers and counters are initialized only by the PONR, INITX, IMR, LVDL1R, RVD, CSVMOR, CSVSOR and CSVSCRR.
762	CHAPTER 12:Backup RAM Interface 1. Overview	Revised the shading parts as below: Error) Target RAMs ● Backup RAM0: 32 KB ● Backup RAM1: 32 KB Correct) Target RAMs ● Backup RAM0: size is product specific ● Backup RAM1: size is product specific
		Revised the shading parts as below: Error) ■Unique (as unique data, {address[3:0],{6{address[7:0]}} is written in 52 bits containing the ECC bit) Correct) ■Unique (as unique data, address information (below) is written in 52 bits containing the ECC bit.) − Address information ((address & 0x0000000F) ＜＜ 48) + ((address & 0x000000FF) ＜＜ 40) + ((address & 0x000000FF) ＜＜ 32) + ((address & 0x000000FF) ＜＜ 24) + ((address & 0x000000FF) ＜＜ 16) + ((address & 0x000000FF) ＜＜ 8) + (address & 0x000000FF)

Page	Section	Change Results
771	CHAPTER 12:Backup RAM Interface 3.2. RAM Diagnosis	Revised the shading parts as below: Error) 1. Unique data (write {address[3:0],{6{address[7:0]}} consisting of 52 bits containing the ECC bit) Correct) 1. Unique (as unique data, address information (below) is written in 52 bits containing the ECC bit.) − Address information ((address & 0x0000000F) ＜＜ 48) + ((address & 0x000000FF) ＜＜ 40) + ((address & 0x000000FF) ＜＜ 32) + ((address & 0x000000FF) ＜＜ 24) + ((address & 0x000000FF) ＜＜ 16) + ((address & 0x000000FF) ＜＜ 8) + (address & 0x000000FF)
786	CHAPTER 12: Backup RAM Interface 5. Registers 5.3. BURAMIF Double-Bit ECC Error Address Register (BURIF_DEEAR)	Revised the shading parts as below: Error) Notes: - ERR_ADDR is an offset in word units (RAM address). - Obtain the absolute address by adding {ERR_ADDR, 2'b00} to the base address. - The effective address of ERR_ADDR changes according to RAM size. - 64KB Effective address: ERR_ADDR[13:0] Correct) Notes: - ERR_ADDR is an offset in word units (RAM address). - Obtain the absolute address by adding {ERR_ADDR << 2} to the base address. - The effective address of ERR_ADDR changes according to RAM size. - 64KB Effective address: ERR_ADDR[13:0]

Page	Section	Change Results
787	CHAPTER 12: Backup RAM Interface 5. Registers 5.4. BURAMIF Single-Bit ECC Error Address Register (BURIF_SEEAR)	Revised the shading parts as below: Error) Notes: - ERR_ADDR is an offset in word units (RAM address). - Obtain the absolute address by adding {ERR_ADDR, 2'b00} to the base address. - The effective address of ERR_ADDR changes according to RAM size. - 64KB Effective address: ERR_ADDR[13:0] Correct) Notes: - ERR_ADDR is an offset in word units (RAM address). - Obtain the absolute address by adding {ERR_ADDR << 2} to the base address. - The effective address of ERR_ADDR changes according to RAM size. - 64KB Effective address: ERR_ADDR[13:0]
788	CHAPTER 12: Backup RAM Interface 5. Registers 5.5. BURAMIF ECC Pseudo-Error Generation Address Register (BURIF_EFEAR)	Revised the shading parts as below: Error) Notes: - ERR_ADDR is an offset in word units (RAM address). - Obtain the absolute address by adding {ERR_ADDR, 2'b00} to the base address. - The effective address of ERR_ADDR changes according to RAM size. - 64KB Effective address: ERR_ADDR[13:0] Correct) Notes: - ERR_ADDR is an offset in word units (RAM address). - Obtain the absolute address by adding {ERR_ADDR << 2} to the base address. - The effective address of ERR_ADDR changes according to RAM size. - 64KB Effective address: ERR_ADDR[13:0]

Page	Section	Change Results
796	CHAPTER 12: Backup RAM Interface 5. Registers 5.10. BURAMIF TEST Error Address Register 0 (BURIF_TEAR0)	Revised the shading parts as below: Error) Notes: - When any of the TER bits becomes "1", even if an error occurs while another diagnosis is being performed, the corresponding error source bit does not become "1". - ERR_ADDR is an offset address in units of words (RAM address). - To calculate the absolute address, add {ERR_ADDR, 2'b00} to the base address. - The effective address of ERR_ADDR changes according to RAM size. - 64KB Effective address: ERR_ADDR[13:0] Correct) Notes: - When any of the TER bits becomes "1", even if an error occurs while another diagnosis is being performed, the corresponding error source bit does not become "1". - ERR_ADDR is an offset address in units of words (RAM address). - To calculate the absolute address, add {ERR_ADDR << 2} to the base address. - The effective address of ERR_ADDR changes according to RAM size. - 64KB Effective address: ERR_ADDR[13:0]
798	CHAPTER 12: Backup RAM Interface 5. Registers 5.11. BURAMIF TEST Error Address Register 1 (BURIF_TEAR1)	Revised the shading parts as below: Error) Notes: - When any of the TER bits becomes "1", even if an error occurs while another diagnosis is being performed, the corresponding error source bit does not become "1". - ERR_ADDR is an offset address in units of words (RAM address). - To calculate the absolute address, add {ERR_ADDR, 2'b00} to the base address. - The effective address of ERR_ADDR changes according to RAM size. - 64KB Effective address: ERR_ADDR[13:0] Correct) Notes: - When any of the TER bits becomes "1", even if an error occurs while another diagnosis is being performed, the corresponding error source bit does not become "1". - ERR_ADDR is an offset address in units of words (RAM address). - To calculate the absolute address, add {ERR_ADDR << 2} to the base address. - The effective address of ERR_ADDR changes according to RAM size. - 64KB Effective address: ERR_ADDR[13:0]

Page	Section	Change Results
800	CHAPTER 12: Backup RAM Interface 5. Registers 5.12. BURAMIF TEST Error Address Register 2 (BURIF_TEAR2)	Revised the shading parts as below: Error) Notes: - When any of the TER bits becomes "1", even if an error occurs while another diagnosis is being performed, the corresponding error source bit does not become "1". - ERR_ADDR is an offset address in units of words (RAM address). - To calculate the absolute address, add {ERR_ADDR, 2'b00} to the base address. - The effective address of ERR_ADDR changes according to RAM size. - 64KB Effective address: ERR_ADDR[13:0] Correct) Notes: - When any of the TER bits becomes "1", even if an error occurs while another diagnosis is being performed, the corresponding error source bit does not become "1". - ERR_ADDR is an offset address in units of words (RAM address). - To calculate the absolute address, add {ERR_ADDR << 2} to the base address. - The effective address of ERR_ADDR changes according to RAM size. - 64KB Effective address: ERR_ADDR[13:0]
801	CHAPTER 12: Backup RAM Interface 5. Registers 5.13. BURAMIF TEST Start Address Register (BURIF_TASAR)	Revised the shading parts as below: Error) Notes: - SADDR is an offset address in units of words (RAM address). - To calculate the absolute address, add {SADDR, 2'b00} to the base address. - A value outside the Backup RAM area and a value such that BURIF_TASAR.SADDR > BURIF_TAEAR.EADDR cannot be set. - The effective address of SADDR changes according to RAM size. - 64KB Effective address: SADDR[13:0] Correct) Notes: - SADDR is an offset address in units of words (RAM address). - To calculate the absolute address, add {SADDR << 2} to the base address. - A value outside the Backup RAM area and a value such that BURIF_TASAR.SADDR > BURIF_TAEAR.EADDR cannot be set. - The effective address of SADDR changes according to RAM size. - 64KB Effective address: SADDR[13:0]

Page	Section	Change Results
802	CHAPTER 12: Backup RAM Interface 5. Registers 5.14. BURAMIF TEST End Address Register (BURIF_TAEAR)	Revised the shading parts as below: Error) Notes: - EADDR is an offset in word units (RAM address). - Obtain the absolute address by adding {EADDR, 2'b00} to the base address. - A value outside the Backup RAM area and a value such that BURIF_TASAR.SADDR > BURIF_TAEAR.EADDR cannot be set. - The effective address of EADDR changes according to RAM size. - The "*" display of the initial value means that the initial value of EADDR changes according to RAM size. - 64KB Effective address: EADDR[13:0] - EADDR initial Value changes according to RAM size. - 00111111_11111111 64KB Correct) Notes: - EADDR is an offset in word units (RAM address). - Obtain the absolute address by adding {EADDR << 2} to the base address. - A value outside the Backup RAM area and a value such that BURIF_TASAR.SADDR > BURIF_TAEAR.EADDR cannot be set. - The effective address of EADDR changes according to RAM size. - The "*" display of the initial value means that the initial value of EADDR changes according to RAM size. - 64KB Effective address: EADDR[13:0] - EADDR initial Value changes according to RAM size. - 00111111_11111111 64KB
838	CHAPTER 14: Hardware Watchdog Timer 1. Overview	Deleted the shading parts as below: Hardware Watchdog Timer Features ■ A protection function of the Hardware watchdog trigger 0/1 register using PPU setting is implemented.(Function not supported by B/B)

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td></td><td></td><td>Added the shading parts as below:

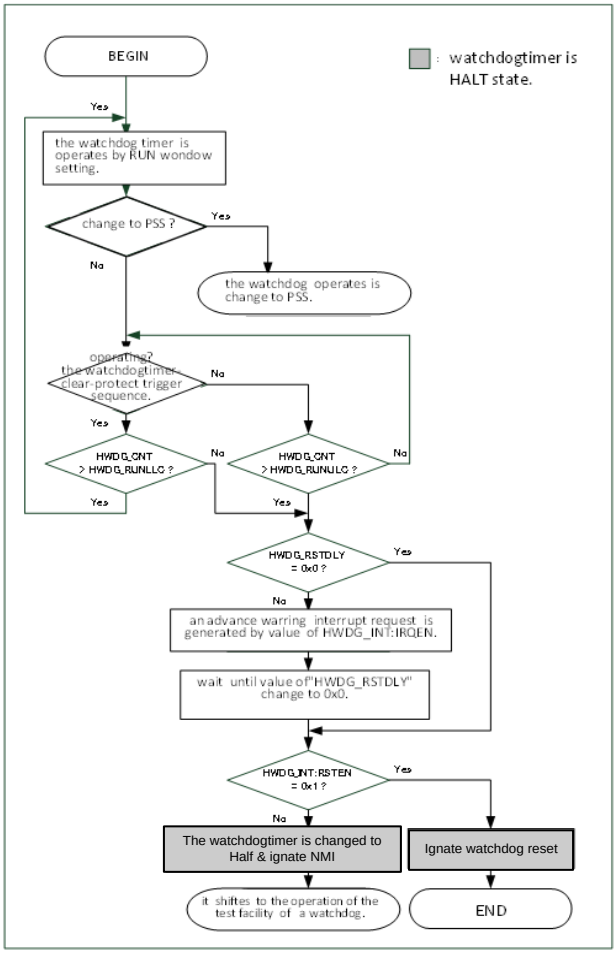
Correct)
Notes:
− The use of the watchdog interrupt request (NMI) is prohibited except when used as a test function. (This is prohibited because if such use is allowed, ***security*** will be compromised during the execution of an application.)
− Figure 3-6 shows the operation of the test function of the Hardware watchdog.</td></tr>
<tr><td>845</td><td>CHAPTER 14: Hardware Watchdog Timer 3. Explanation of Operation</td><td>Deleted the shading parts as below:

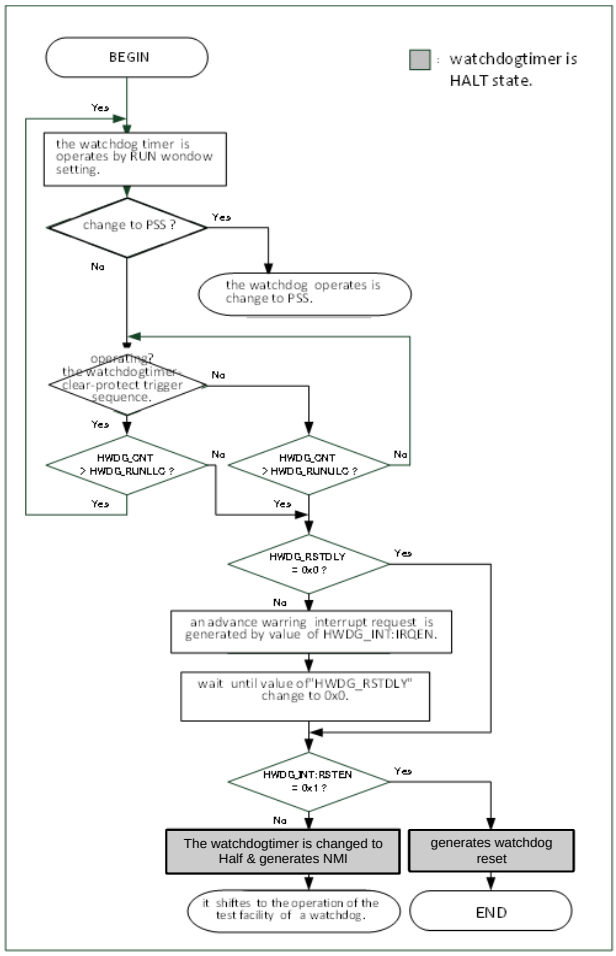
Protection of Hardware Watchdog Trigger 0/1 Register Using PPU Setting
The Hardware watchdog timer uses the PPU setting for the protection of the Hardware watchdog trigger 0/1 register (HWDG_TRG0 and HWDG_TRG1). Specifically, the write operation in non-privileged mode (user access) is enabled or disabled. (Function not supported by B/B)</td></tr>
<tr><td>846</td><td>CHAPTER 14: Hardware Watchdog Timer 3. Explanation of Operation</td><td>Revised the shading parts as below:

Note:

Error)
− The use of the watchdog interrupt request (NMI) is prohibited except when used as a test function. (This is prohibited because if such use is allowed, security will be compromised during the execution of an application.) Figure 3 6 shows the operation of the test function of the Hardware watchdog.
Correct)
− The use of the watchdog interrupt request (NMI) is prohibited except when used as a test function. (This is prohibited because if such use is allowed, safety will be compromised during the execution of an application.) Figure 3 6 shows the operation of the test function of the Hardware watchdog.</td></tr>
<tr><td>847</td><td>CHAPTER 14: Hardware Watchdog Timer 3. Explanation of Operation</td><td>Deleted the shading parts as below

Table 3-4 Differences between Software Watchdog and Hardware Watchdog
<table><tr><th>Item</th><th>Software Watchdog</th><th>Hardware Watchdog</th></tr><tr><td>Protection of Watchdog Trigger 0/1 Register Using PPU Setting</td><td>Supported
(Function not supported by B/B)</td><td>Same as on left
(Function not supported by B/B)</td></tr></table></td></tr>
</table>

Page	Section	Change Results
852	CHAPTER 14: Hardware Watchdog Timer 3. Explanation of Operation	Revised the shading parts as below Error) 

Page	Section	Change Results
		Correct) 
861, 862	CHAPTER 14: Hardware Watchdog Timer 5. Operation Examples	Revised the shading parts as below: Error) - Set the same value following two upper limit registers. - Hardware watchdog lower limit PSS setting register (HWDG_PSSLLS) - Hardware watchdog lower limit RUN setting register (HWDG_RUNLLS) Correct) - Set the same value following two lower limit registers. - Hardware watchdog lower limit PSS setting register (HWDG_PSSLLS) - Hardware watchdog lower limit RUN setting register (HWDG_RUNLLS)

Page	Section	Change Results
869	CHAPTER 14: Hardware Watchdog Timer 6. Register List 6.4. Hardware Watchdog Trigger 0 Register (HWDG_TRG0)	Deleted the shading parts as below When performing write access to this register, you do not need to follow the watchdog register write protection sequence. This register is used to execute the watchdog counter clear protection trigger sequence. Write the value defined in the Hardware watchdog trigger 0 configuration register (HWDG_TRG0CFG) to this register. When a value other than the one in the Hardware watchdog trigger 0 configuration register (HWDG_TRG0CFG) is written, the watchdog reset request or watchdog interrupt request (NMI) is generated. PPU setting enables or disables write access to this register in non-privileged mode. (Function not supported by B/B)
870	CHAPTER 14: Hardware Watchdog Timer 6. Register List 6.5. Hardware Watchdog Trigger 1 Register (HWDG_TRG1)	Deleted the shading parts as below When performing write access to this register, you do not need to follow the watchdog register write protection sequence. This register is used to execute the watchdog counter clear protection trigger sequence. Write the value defined in the Hardware watchdog trigger 1 configuration register (HWDG_TRG1CFG) to this register. When a value other than the one in the Hardware watchdog trigger 1 configuration register (HWDG_TRG1CFG) is written, the watchdog reset request or watchdog interrupt request (NMI) is generated. PPU setting enables or disables write access to this register in non-privileged mode. (Function not supported by B/B)
871	CHAPTER 14: Hardware Watchdog Timer 6. Register List	Revised the shading parts as below: 6.6. Hardware Watchdog Interrupt Configuration Register (HWDG_INT) [bit17] RSTEN: Reset/NMI Enable Bit Error) When "1" is written: The watchdog interrupt request is generated. Correct) When "1" is written: The watchdog reset request is generated.
877		Revised the shading parts as below: 6.11. Hardware Watchdog Upper Limit RUN Setting Register (HWDG_RUNULS) Note Error) The window lower limit value actually used is the value in the Hardware watchdog lower limit RUN current register (HWDG_RUNLLC). Correct) The window upper limit value actually used is the value in the Hardware watchdog upper limit RUN current register (HWDG_RUNULC).

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>889</td><td>CHAPTER 14: Hardware Watchdog Timer 7. Precautions for Using This Device</td><td>Deleted the shading parts as below:

Factors for Bus Error Responses (Data Abort)

■Write access in non-privileged mode to the Hardware watchdog trigger 0/1 register (HWDG_TRG0 and HWDG_TRG1) when such access is prohibited by PPU setting (Function not supported by B/B)</td></tr>
<tr><td>892</td><td>CHAPTER 15: Software Watchdog Timer 1. Overview</td><td>Deleted the shading parts as below:

Software Watchdog Timer Features
■A protection function of the Software watchdog trigger 0/1 register using PPU setting is implemented. (Function not supported by B/B)</td></tr>
<tr><td>899</td><td>CHAPTER 15: Software Watchdog Timer 3. Explanation of Operation</td><td>Deleted the shading parts as below:

Protection of Software Watchdog Trigger 0/1 Register Using PPU Setting
The Software watchdog timer uses the PPU setting for the protection of the Software watchdog trigger 0/1 register (SWDG_TRG0 and SWDG_TRG1). Specifically, the write operation in non-privileged mode (user access) is enabled or disabled. (Function not supported by B/B)</td></tr>
<tr><td>902</td><td>CHAPTER 15: Software Watchdog Timer 3. Explanation of Operation</td><td>Deleted the shading parts as below:

Deleted)

Table 3-4 Differences between Software Watchdog and Hardware Watchdog
<table>
<tr><th>Item</th><th>Software Watchdog</th><th>Hardware Watchdog</th></tr>
<tr><td>Protection of Watchdog Trigger 0/1 Register Using PPU Setting</td><td>Supported
(Function not supported by B/B)</td><td>Same as on left
(Function not supported by B/B)</td></tr>
</table></td></tr>
<tr><td>915</td><td>CHAPTER 15: Software Watchdog Timer 5. Operation Examples</td><td>Revised the shading parts as below:

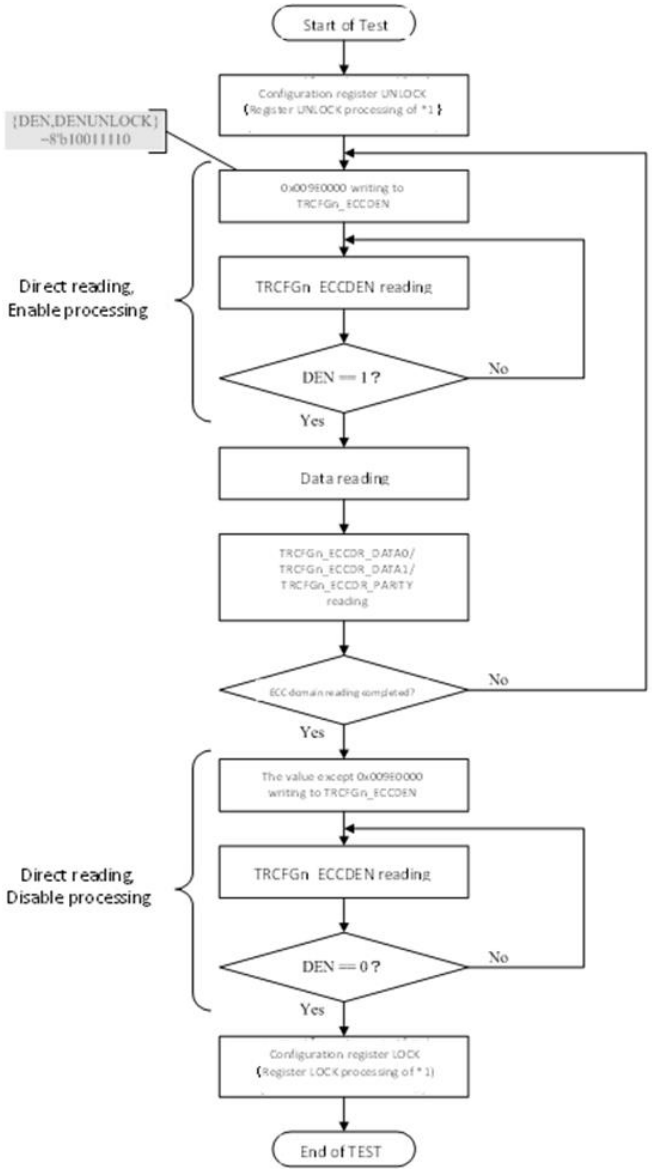
Error)
- Set the same value following two upper limit registers.
- Software watchdog lower limit PSS setting register (SWDG_PSSLLS)
- Software watchdog lower limit RUN setting register (SWDG_RUNLLS)

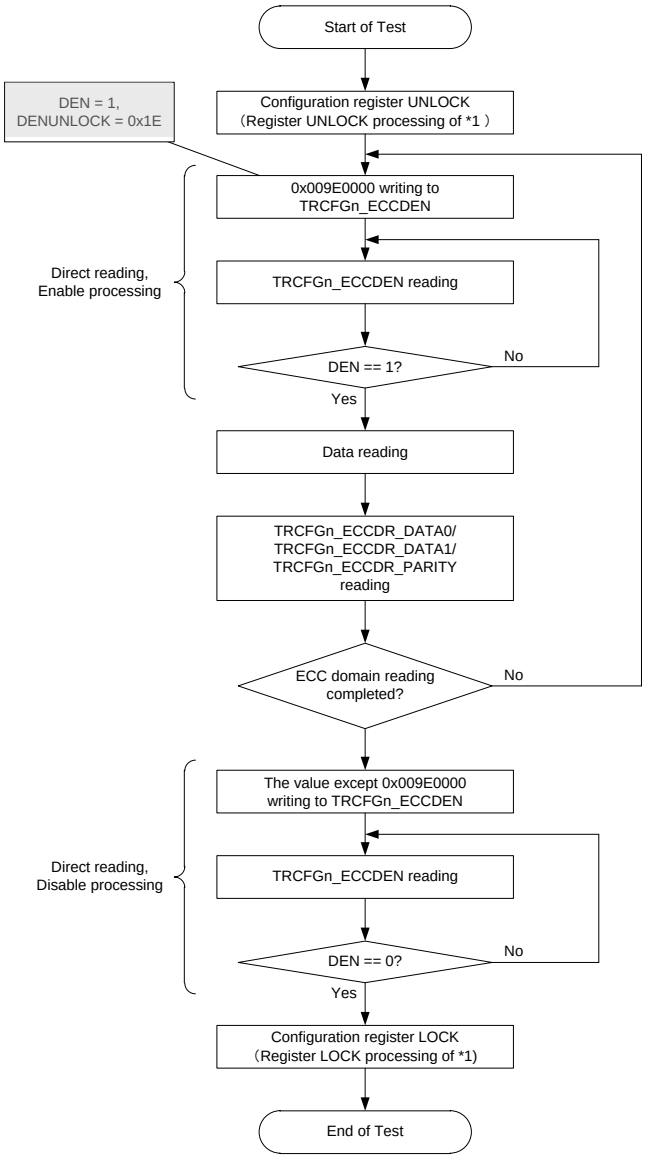
Correct)
- Set the same value following two lower limit registers.
- Software watchdog lower limit PSS setting register (SWDG_PSSLLS)
- Software watchdog lower limit RUN setting register (SWDG_RUNLLS)</td></tr>
</table>

Page	Section	Change Results
922	CHAPTER 15: Software Watchdog Timer 6. Register List 6.4. Software Watchdog Trigger 0 Register (SWDG_TRG0)	Deleted the shading parts as below: When performing write access to this register, you do not need to follow the watchdog register write protection sequence. This register is used to execute the watchdog counter clear protection trigger sequence. Write the value defined in the Software watchdog trigger 0 configuration register (SWDG_TRG0CFG) to this register. When a value other than the one in the Software watchdog trigger 0 configuration register (SWDG_TRG0CFG) is written, the watchdog reset request or watchdog interrupt request (NMI) is generated. PPU setting enables or disables write access to this register in non-privileged mode. (Function not supported by B/B)
923	CHAPTER 15: Software Watchdog Timer 6. Register List 6.5. Software Watchdog Trigger 1 Register (SWDG_TRG1)	Deleted the shading parts as below: When performing write access to this register, you do not need to follow the watchdog register write protection sequence. This register is used to execute the watchdog counter clear protection trigger sequence. Write the value defined in the Software watchdog trigger 1 configuration register (SWDG_TRG1CFG) to this register. When a value other than the one in the Software watchdog trigger 1 configuration register (SWDG_TRG1CFG) is written, the watchdog reset request or watchdog interrupt request (NMI) is generated. PPU setting enables or disables write access to this register in non-privileged mode. (Function not supported by B/B)
924	CHAPTER 15: Software Watchdog Timer 6. Register List	Revised the shading parts as below: 6.6. Software Watchdog Interrupt Configuration Register (SWDG_INT) Error) This register is used to make settings related to the watchdog interrupt request (NMI) and prior warning interrupt request. Correct) This register is used to make settings related to the watchdog reset request (NMI) and prior warning interrupt request. [bit17] RSTEN: Reset/NMI Enable Bit Error) When "1" is written: The watchdog interrupt request is generated. Correct) When "1" is written: The watchdog reset request is generated.

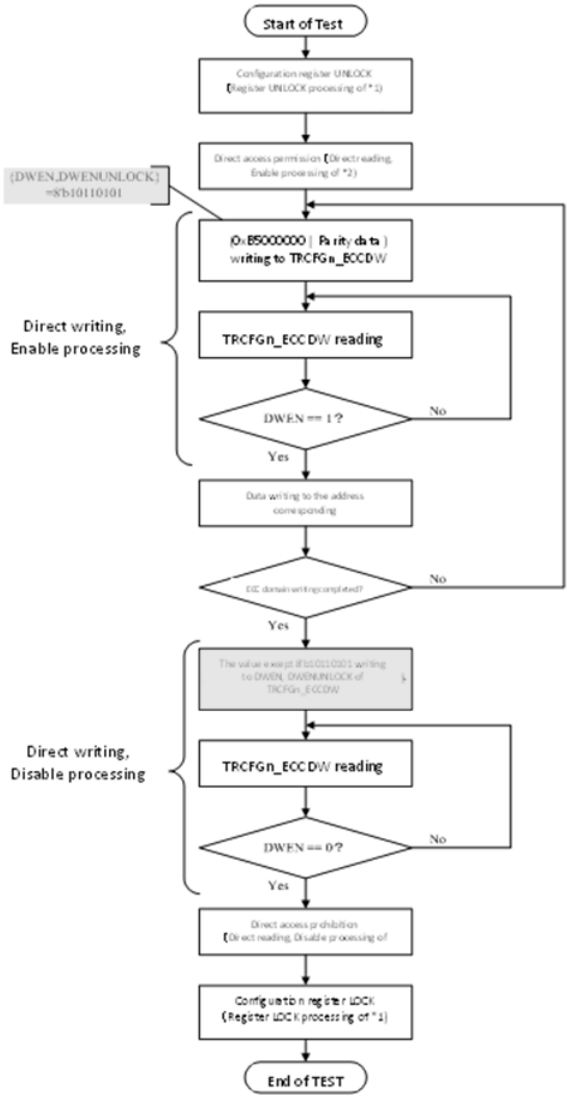
Page	Section	Change Results
930		Revised the shading parts as below: 6.11. Software Watchdog Upper Limit RUN Setting Register (SWDG_RUNULS) Note Error) The window lower limit value actually used is the value in the Software watchdog lower limit RUN current register (SWDG_RUNLLC). Correct) The window upper limit value actually used is the value in the Software watchdog upper limit RUN current register (SWDG_RUNULC).
942	CHAPTER 15: Software Watchdog Timer 7. Precautions for Using This Device	Deleted the shading parts as below: **Factors for Bus Error Responses (Data Abort)** ■ Write access in non-privileged mode to the Software watchdog trigger 0/1 register (SWDG_TRG0 and SWDG_TRG1) when such access is prohibited by PPU setting (Function not supported by B/B)
944	CHAPTER 16: TCRAM Interface 1. Overview	Revised the shading parts as below: ■RAM diagnosis and initialization function Error) − Unique (unique data is {address [6:0], 4 {address [7:0]}}. Upper 7 bits are ECC data.) Correct) − Unique (as unique data, address information (below), Upper 7 bits are ECC data.) Address Information ((address & 0x0000007F) << 32) + ((address & 0x000000FF) << 24) + ((address & 0x000000FF) << 16) + ((address & 0x000000FF) << 8) + (address & 0x000000FF)

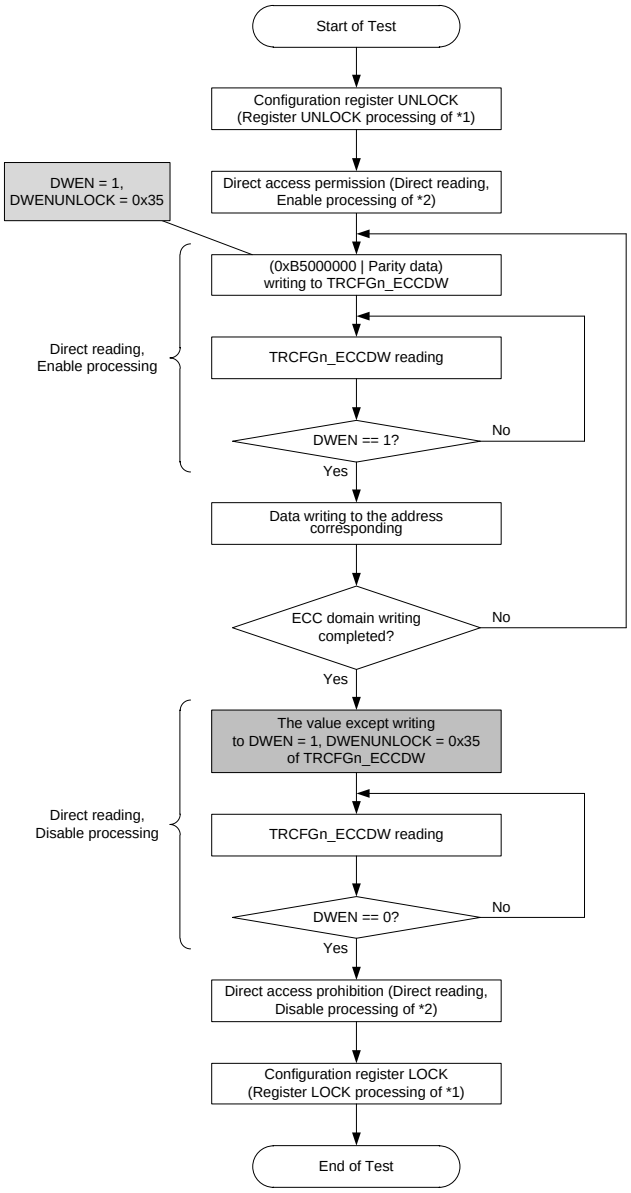
Page	Section	Change Results
949	CHAPTER 16: TCRAM Interface 3. Explanation of Operation	Revised the shading parts as below: RAM Diagnosis Error) ■Unique (unique data is {address [6:0], 4 {address [7:0]}}. Upper 7 bits are ECC data.) Correct) ■Unique (as unique data, address information (below), Upper 7 bits are ECC data.) Address Information ((address & 0x0000007F) << 32) + ((address & 0x000000FF) << 24) + ((address & 0x000000FF) << 16) + ((address & 0x000000FF) << 8) + (address & 0x000000FF)

Page	Section	Change Results
953	CHAPTER 16: TCRAM Interface 4. Setting Procedure Examples	Changed the shading parts as below: **Figure 4-2 Direct Read Access Flow** Error) 

Page	Section	Change Results
		Correct) 

Page	Section	Change Results
954	CHAPTER 16: TCRAM Interface 4. Setting Procedure Examples	Changed the shading parts as below: **Figure 4-3 Direct Write Access Flow** Error)

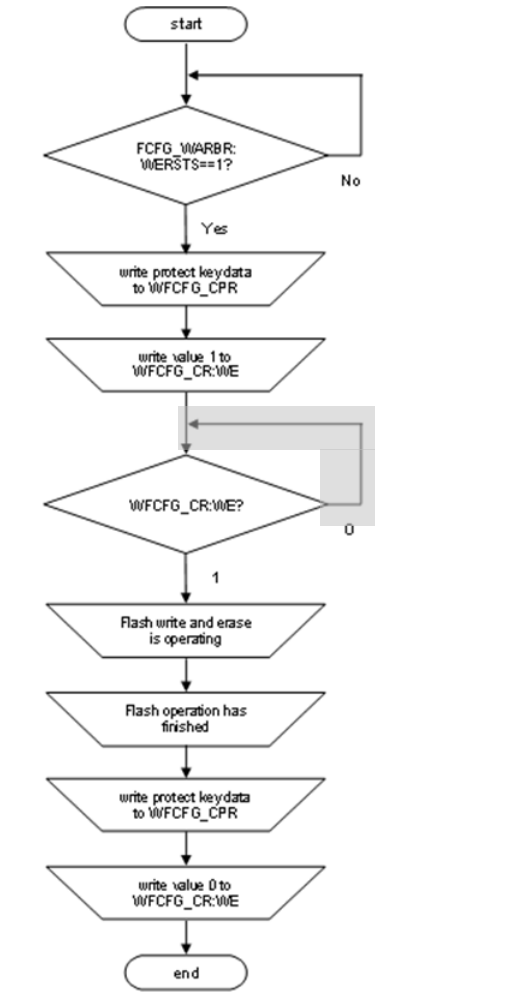


Page	Section	Change Results
		Correct) 

Page	Section	Change Results
971	CHAPTER 16: TCRAM Interface 5. Registers 5.9. TCRAM IF TEST Error Address Register 0 (TRCFGn_TEAR0)	Revised the shading parts as below: Notes: Error) - To calculate the absolute address, add {ERR_ADDR, 2'b00} to the base address. Correct) - To calculate the absolute address, add {ERR_ADDR << 2} to the base address.
973	CHAPTER 16: TCRAM Interface 5. Registers 5.10. TCRAM IF TEST Error Address Register 1 (TRCFGn_TEAR1)	Revised the shading parts as below: Notes: Error) - To calculate the absolute address, add {ERR_ADDR, 2'b00} to the base address. Correct) - To calculate the absolute address, add {ERR_ADDR << 2} to the base address.
975	CHAPTER 16: TCRAM Interface 5. Registers 5.11. TCRAM IF TEST Error Address Register 2 (TRCFGn_TEAR2)	Revised the shading parts as below: Notes: Error) - To calculate the absolute address, add {ERR_ADDR, 2'b00} to the base address. Correct) - To calculate the absolute address, add {ERR_ADDR << 2} to the base address.
976	CHAPTER 16: TCRAM Interface 5. Registers 5.12. TCRAM IF TEST End Address Register (TRCFGn_TAEAR)	Revised the shading parts as below: Notes: Error) - Obtain the absolute address by adding {EADDR, 2'b00} to the base address. Correct) - Obtain the absolute address by adding {EADDR << 2} to the base address.

Page	Section	Change Results
977	CHAPTER 16: TCRAM Interface 5. Registers 5.13. TCRAM IF TEST Start Address Register (TRCFGn_TASAR)	Revised the shading parts as below: Notes: Error) - To calculate the absolute address, add {SADDR, 2'b00} to the base address. Correct) - To calculate the absolute address, add {SADDR << 2} to the base address.
996	CHAPTER 17: TCFLASH 3. Explanation of Operation 3.1. Operation Mode of TCFLASH	Changed the shading parts as below: Error) If this LSI is in parallel programmer mode, the pins of the flash memory are directly assigned to the pins of this LSI. Therefore, programming or erasing by using the flash programmer is possible. In parallel programmer mode, ECC is generated and tested for programming and reading. Correct) In the serial writer mode or the parallel writer mode, the Flash can be read/programmed/erased according to security setting.
1000	CHAPTER 17: TCFLASH 3. Explanation of Operation 3.5.1. Command Sequence	Revised the shading parts as below: Error) ■ The address SA given upon the 6th program cycle of the sector erase command indicates the address of the sector to be erased. SA is specified in the same format as PA. Correct) ■ The address SA given upon the 6th program cycle of the sector erase command and 1st cycle of sector erase suspend/resume commnad, and the address SA0, SA1 given upon 6th, 7th cycle of multi sector erase command indicates the address of the sector to be erased. SA is specified in the same format as PA.
1026	CHAPTER 17: TCFLASH 5. Registers 5.2. TCFLASH Configuration Register : TCFCFG_FCFGR (TCFLASH Configuration Register)	Added the shading parts as below: Added) Note: - Avoid TCFlash access via AXI bus while changing Flash wait configuration (FAWC[1:0]) 0 -> X (X=1,2,3).

Page	Section	Change Results
1047	CHAPTER 17: TCFLASH 6. Others	Revised the shading parts as below: Error) **Reset during Execution of Program or Erase Operation** If this LSI is reset during execution of the program operation, the contents of the target address are shown as undefined. If it is reset during execution of the sector erase operation, the contents of the target sector are shown as undefined. In these cases, retry the suspended program or erase operation after reset completes. Correct) **Reset during Execution of Program or Erase Operation** If this LSI is reset during execution of the program operation, the contents of the target address are shown as undefined. If it is reset during execution of the sector erase operation, the contents of the target sector are shown as undefined. In these cases, following steps should be done. 1. erase the sector. 2. re-program the sector
1047	CHAPTER 17: TCFLASH 6. Others	Added the shading parts as below: Note: TCFLASH Status Register Bits Since some status information of flash memory is common to Code side (TCFLASH) and Work side (WorkFLASH) in Dual Port FLASH macro, TCFLASH status register bits listed below are affected by WorkFLASH operation in addition to TCFLASH operation: - TCFCFG_FSTAT.ESPS - TCFCFG_FSTAT.ERSEC - TCFCFG_FSTAT.RDY - TCFCFG_FSTAT.HANG - TCFCFG_FSTAT.RDYINT - TCFCFG_FSTAT.HANGINT Use these bits only when program/erase is performed in TCFLASH.
1052	CHAPTER 18: WorkFLASH 1. Overview	Revised the shading parts as below: Error) ■ The Flash security function has a function to protect against writing and erasing in units of sectors. Correct) ■ The Flash security function has a function to protect against writing and erasing in units of sectors. CPU can execute instructions on WorkFLASH in writing or erasing TCFLASH.

Page	Section	Change Results
1067	CHAPTER 18: WorkFLASH 4. Setting Procedure 4.2. Enabling Writing	Changed the below: **Figure 4-1 Procedure for Setting Flash Write Enable** Error)  1eb0ca_d6771446b6aad0a6136b80806f907693_5802_rf_5762c3_u000_c235498_a

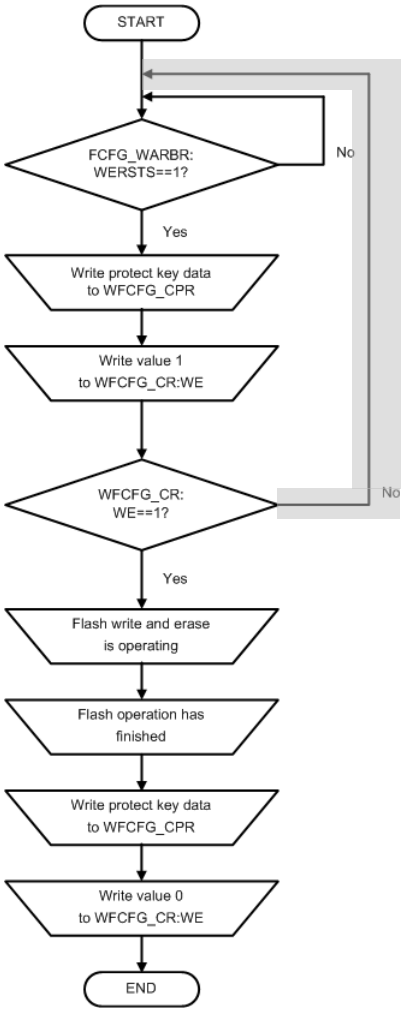
<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td></td><td></td><td>Correct)

</td></tr>
<tr><td>1074</td><td>CHAPTER 18: WorkFLASH
5. Registers</td><td>Changed the shading parts as below:

Table 5-2 Register Memory Map

Error)
<table>
<tr><th>Offset</th><th>Abbreviated Register Name</th><th>Description</th></tr>
<tr><td>0x008</td><td>WFCFG_CR
00000000_00000000_00000000_00000010</td><td>WorkFLASH configuration protection key register</td></tr>
</table>
Correct)
<table>
<tr><th>Offset</th><th>Abbreviated Register Name</th><th>Description</th></tr>
<tr><td>0x008</td><td>WFCFG_CR
00000000_00000000_00000000_00001001</td><td>WorkFLASH configuration protection key register</td></tr>
</table></td></tr>
</table>

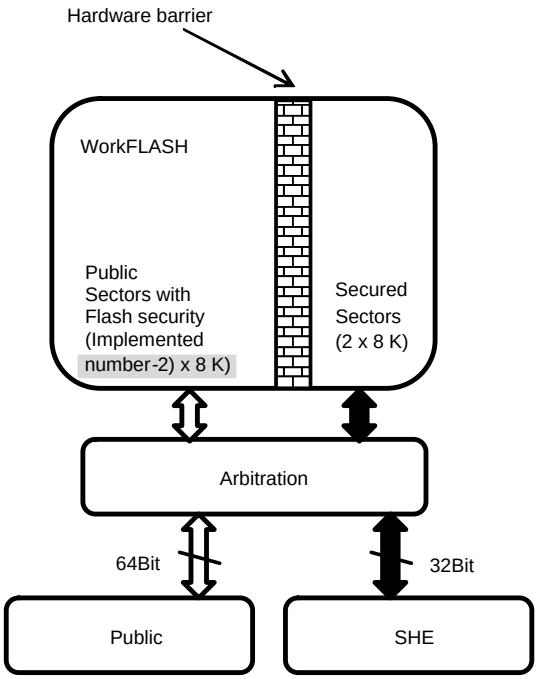
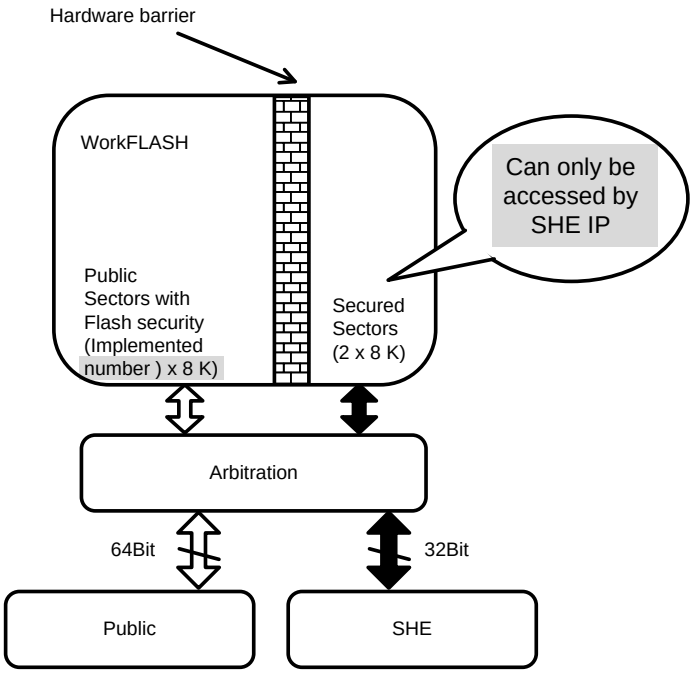
<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1075</td><td>CHAPTER 18: WorkFLASH
5. Registers</td><td>Changed the shading parts as below:

Table 5-2 Register Memory Map

Error)
<table>
<tr><th>Offset</th><th>Abbreviated Register Name</th><th>Description</th></tr>
<tr><td>0x080 to 0x0FC</td><td>-</td><td>Reserved area</td></tr>
</table>
Correct)
<table>
<tr><th>Offset</th><th>Abbreviated Register Name</th><th>Description</th></tr>
<tr><td>0x080 to 0x3FC</td><td>-</td><td>Reserved area</td></tr>
</table></td></tr>
<tr><td>1107</td><td>CHAPTER 18: WorkFLASH
7. Secure Hardware Extension (SHE) Support</td><td>Revised the shading parts as below:

Error)
Overview
SHE IP requires non-volatile memory to store encryption keys. For this reason, WorkFLASH is shared by SHE IP. The last two of the implemented sectors are assigned exclusively to SHE. (They can not be used as normal WorkFLASH areas.)

Correct)
S6J3200 series containing 48 KB/112 KB/240 KB WorkFLASH have in total 6/14/30 sectors allocated for general use cases and two additional sectors which are not part of memory map but internally used only by SHE IP.</td></tr>
</table>

Page	Section	Change Results
1108	CHAPTER 18: WorkFLASH 7. Secure Hardware Extension (SHE) Support	Revised the shading parts as below: Error)  Correct) 

Page	Section	Change Results
1170	CHAPTER 21: Exclusive Access Memory (EAM) 3. Explanation of Operation	Revised the shading parts as below: Error) The letter "x" in LDRx, STRx, LDREXx, STREXx, and so on in the commands described below represents the size of the command: "D" = Doubleword, "W" = Word, "H" = Halfword, and "B" = Byte. (There is not suffix of "x" in the Word case.) Correct) The letter "x" in LDRx, STRx, LDREXx, STREXx, and so on in the commands described below represents the size of the command: "D" = Doubleword, "" = Word, "H" = Halfword, and "B" = Byte. (There is not suffix of "x" in the Word case.)
1332	CHAPTER 25: Memory 1. Overview	Removed the shading parts as below: ■Features of the MPU AXI - Supports MPU stop feature that allows blocking of all the accesses to memory space
1333	CHAPTER 25: Memory 2. Configuration and Block Diagram	Removed the shading parts as below: Figure 2-2-1 MPU stop request
1339	CHAPTER 25: Memory 3. Operation of the MPU AXI	Removed the shading parts as below: ■MPU Stop Feature Optionally MPU AXI supports MPU stop feature. When this mode is enabled all accesses to memory space are blocked and MPU AXI does the following actions: ·Burst type signal is driven to FIXED type burst ·Burst address is driven to predefined FIXED address

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1342</td><td>CHAPTER 25: Memory 4. Registers</td><td>Revised the shading parts as below:

■Memory Layout of MPU AXI Registers

Error)
<table><tr><td>Offset</td><td>+3</td><td>+2</td><td>+1</td><td>+0</td></tr><tr><td>0x0000007C</td><td colspan="4">MPUXSHE_MID
00000000 00000000 00000000 00000000</td></tr></table>
Correct)
<table><tr><td>Offset</td><td>+3</td><td>+2</td><td>+1</td><td>+0</td></tr><tr><td>0x0000007C</td><td colspan="4">MPUXSHE_MID
00000000 00001101 00000010 00000000</td></tr></table></td></tr>
<tr><td>1343</td><td>CHAPTER 25: Memory 4. Registers
4.1. MPU AXI Control Register (MPUXSHE_CTRL0)</td><td>Revised the shading parts as below:

Error)
It also provides controls for enabling or disabling of privileged mode overwrite feature and MPU stop feature.

Correct)
It also provides controls for enabling or disabling of privileged mode overwrite feature.</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td></td><td></td><td>
Revised the shading parts as below:

■MPUXSHE_CTRL0

Error
<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>PROT</td><td>POEN</td><td>MPUSTOPEN</td><td>MPUSTOP</td><td>LST</td></tr>
<tr><td>R0</td><td>R0</td><td>R0</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R</td><td>R</td></tr>
<tr><td>-</td><td>-</td><td>-</td><td>WPS</td><td>WPS</td><td>WPS</td><td>-</td><td>-</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr>
</table>
Correct)
<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>PROT</td><td>POEN</td><td>Reserved</td><td>Reserved</td><td>LST</td></tr>
<tr><td>R0</td><td>R0</td><td>R0</td><td>R/W</td><td>R/W</td><td>R0</td><td>R0</td><td>R</td></tr>
<tr><td>-</td><td>-</td><td>-</td><td>WPS</td><td>WPS</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr>
</table>
</td></tr>
<tr><td>1344</td><td></td><td>
Revised the shading parts as below:
■MPUXSHE_CTRL0
Error)
<table>
<tr><td>[10]</td><td>MPUSTOPEN</td><td>Enable for MPU STOP Feature
"0": MPU stop feature is disabled "1": MPU stop feature is enabled
This bit along with MPU stop input controls the STOP status of MPU AXI.</td></tr>
<tr><td>[9]</td><td>MPUSTOP</td><td>MPU Stop Status
"0": MPU AXI is not stopped
"1": MPU AXI is stopped (i.e.MPUXSHE_CTRL0:MPUSTOPEN = "1" and MPU stop input is asserted). All accesses on AXI master memory write or read address channel are converted to FIXED type burst with predefined address
Note: For availability of MPU stop feature, refer to device specific datasheet.</td></tr>
</table>
Correct)
<table>
<tr><td>[10:9]</td><td>Reserved</td><td>-</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1358</td><td>CHAPTER 25: Memory Protection Unit for AXI of SHE
4.11 MPU AXI Module ID Register</td><td>Revised the shading parts as below:

■MPUXSHE_MID

Error)
<table>
<tr><th>Bit Position</th><th>Bit Field Name</th><th>Bit Description</th></tr>
<tr><td>[31:0]</td><td>MID</td><td>Module ID
The MPU AXI module implemented in the device may vary from device to device. This register identifies the particular version of the hardware used in</td></tr>
</table>
Correct)
<table>
<tr><th>Bit Position</th><th>Bit Field Name</th><th>Bit Description</th></tr>
<tr><td>[31:0]</td><td>MID</td><td>Module ID
Always return a meaningless constant.</td></tr>
</table></td></tr>
<tr><td>1363</td><td>CHAPTER 26: Secure Hardware Extension (SHE)
1. Overview</td><td>Added the shading parts as below:

Note:
Bus-error occurs when register is accessed at SHE-OFF parts.</td></tr>
<tr><td>1380</td><td>CHAPTER 26: Secure Hardware Extension (SHE)
3.2. Status Registers for the Command Interface
3.2.1. SHE Status Register (SHE_STATUS)</td><td>Revised the shading parts as below:

Error)
− In the case of the other values except above ones, it isnot in the normal staus.

Correct)
− In the case of the other values except above ones, it is not in the normal status.</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1410</td><td>CHAPTER 26: Secure Hardware Extension (SHE) 3.5.2. Data Master Status Register (SHE_MSTSTATUS)</td><td>Revised the shading parts as below:
Table 3-23 Data Master Status Register (SHE_MSTSTATUS) Bits

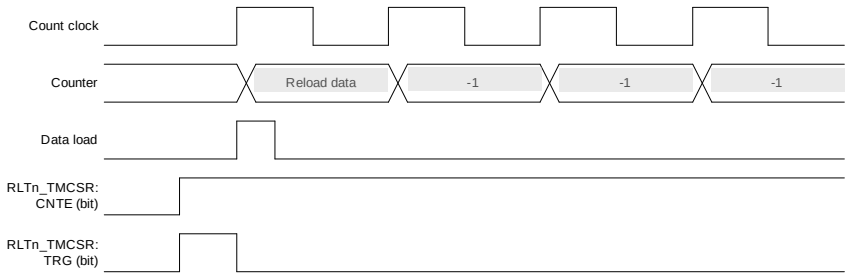
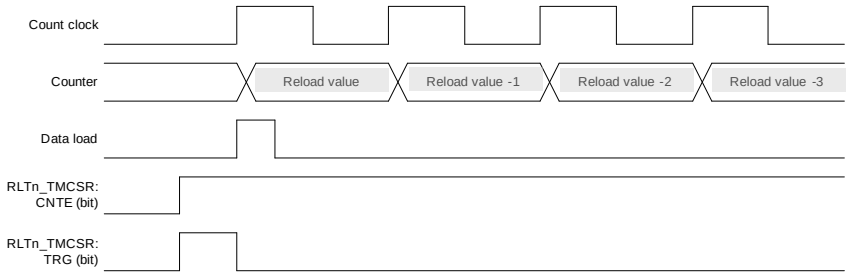
Error)
<table>
<tr><th>Bit Position</th><th>Bit Field Name</th><th>Bit Description</th></tr>
<tr><td>[26:25]</td><td>OMSTERRRESP</td><td>Output Data Channel Master Error Response Code
These bits contain the error response code of the AXI protocol.
'00': OKAY (Normal access okay)
A normal access has been successful. This code can also indicate an exclusive access failure
'01': EXOKAY (Exclusive access okay)
Either the read or write portion of an exclusive access has been successful
'10': SLVERR (Slave error)
Access has reached the slave successfully but the slave wishes to return an error condition to the originating master</td></tr>
</table>
Correct)
<table>
<tr><th>Bit Position</th><th>Bit Field Name</th><th>Bit Description</th></tr>
<tr><td>[26:25]</td><td>OMSTERRRESP</td><td>Output Data Channel Master Error Response Code
These bits contain the error response code of the AXI protocol.
'00': OKAY (Normal access okay)
A normal access has been successful. This code can also indicate an exclusive access failure
'01': EXOKAY (Exclusive access okay)
Either the read or write portion of an exclusive access has been successful
'10': SLVERR (Slave error)
Access has reached the slave successfully but the slave returns an error condition to the originating master</td></tr>
</table></td></tr>
</table>

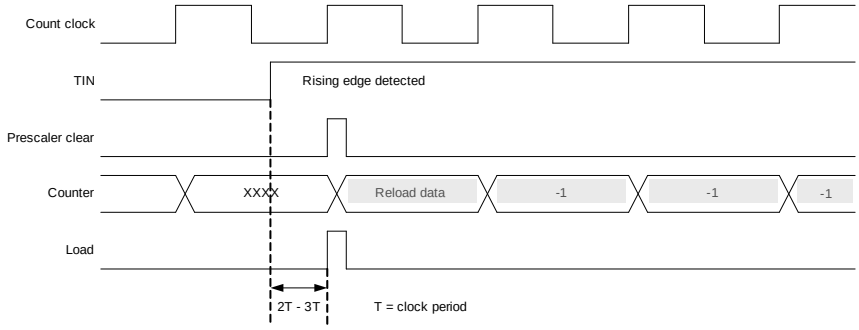
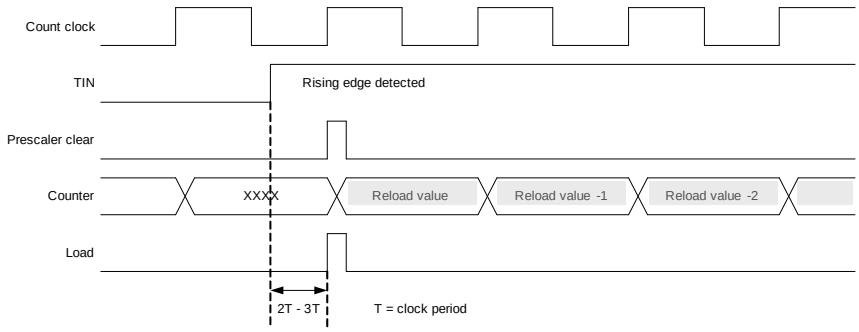
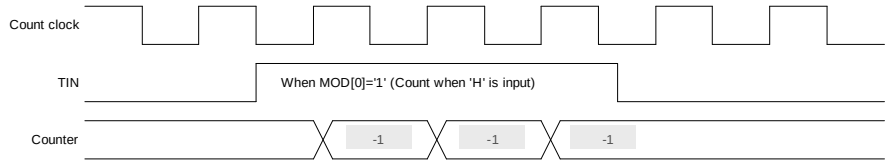
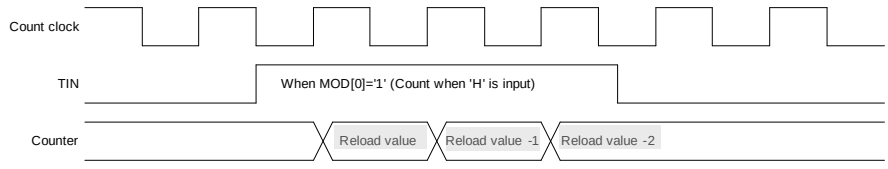
<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1411</td><td>CHAPTER 26: Secure Hardware Extension (SHE) 3.5.2. Data Master Status Register (SHE_MSTSTATUS)</td><td>Revised the shading parts as below:
Table 3-23 Data Master Status Register (SHE_MSTSTATUS) Bits

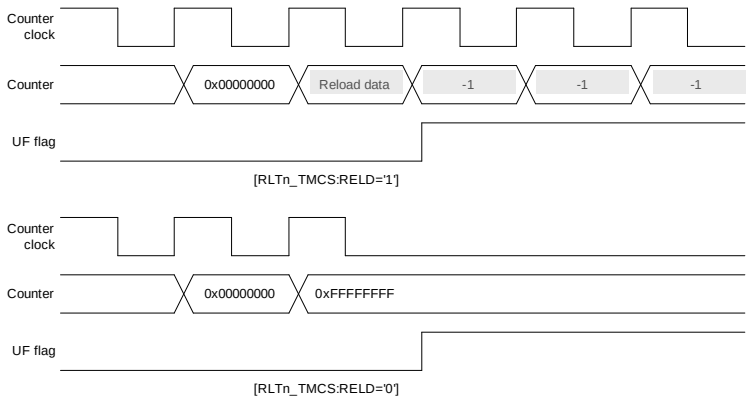
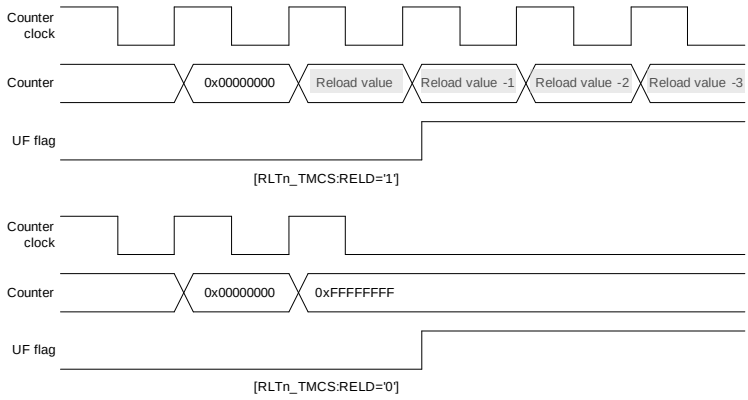
Error)
<table>
<tr><th>Bit Position</th><th>Bit Field Name</th><th>Bit Description</th></tr>
<tr><td>[10:9]</td><td>IMSTERRRESP</td><td>Input Data Channel Master Error Response Code
These bits contain the error response code of the AXI protocol.
'00': OKAY (Normal access okay)
Normal access has been successful. This value can also indicate an exclusive access failure
'01': EXOKAY (Exclusive access okay)
Either the read or write portion of an exclusive access has been successful
'10': SLVERR (Slave error)
Slave error is used when the access has successfully reached the slave but the slave wishes to return an error condition to the originating master</td></tr>
</table>
Correct)
<table>
<tr><th>Bit Position</th><th>Bit Field Name</th><th>Bit Description</th></tr>
<tr><td>[10:9]</td><td>IMSTERRRESP</td><td>Input Data Channel Master Error Response Code
These bits contain the error response code of the AXI protocol.
'00': OKAY (Normal access okay)
Normal access has been successful. This value can also indicate an exclusive access failure
'01': EXOKAY (Exclusive access okay)
Either the read or write portion of an exclusive access has been successful
'10': SLVERR (Slave error)
Slave error is used when the access has successfully reached the slave but the slave returns an error condition to the originating master</td></tr>
</table></td></tr>
</table>

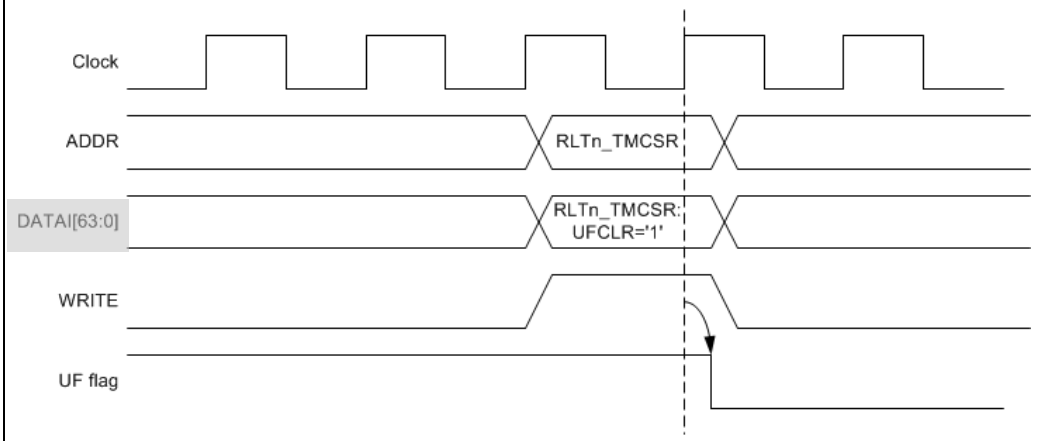
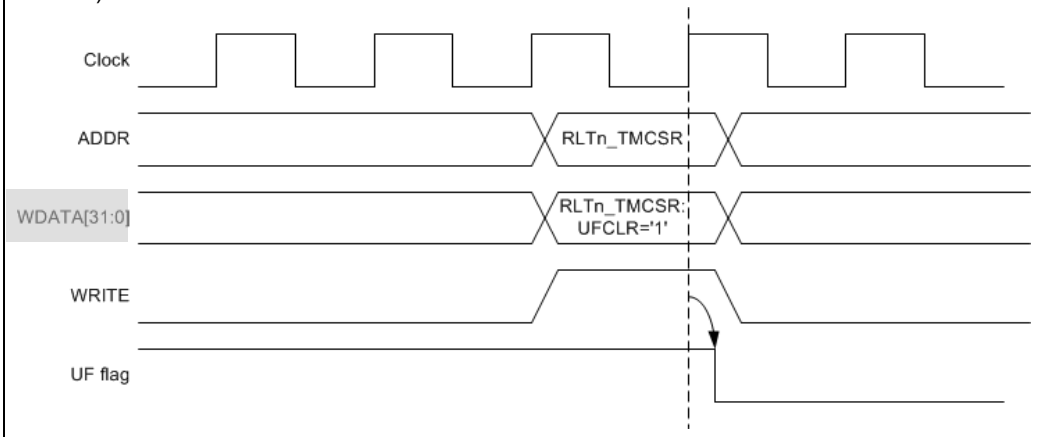
<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1450</td><td>CHAPTER 26: Secure Hardware Extension (SHE) 5.1.2. SHE Used Error Codes/Used Cancel Codes</td><td>Revised the shading parts as below:
Error)
<table>
<tr><th>Cancel Error</th><th>Command Error</th><th>Main Error</th><th>Description</th></tr>
<tr><td>0x0010</td><td>0x030D</td><td>ERC_GENERAL_ERROR</td><td>Command forbidden due to invalid device state or the NVM state.</td></tr>
</table>
Correct)
<table>
<tr><th>Cancel Error</th><th>Command Error</th><th>Main Error</th><th>Description</th></tr>
<tr><td>0x0010</td><td>0x030D</td><td>ERC_GENERAL_ERROR</td><td>Command disabled or command code unknown.</td></tr>
</table>
Deleted the shading parts as below:
Correct)
<table>
<tr><th>Cancel Error</th><th>Command Error</th><th>Main Error</th><th>Description</th></tr>
<tr><td>0x0010</td><td>0x040D</td><td>ERC_GENERAL_ERROR</td><td>Unknown command request code.</td></tr>
</table></td></tr>
<tr><td>1451</td><td></td><td>Inserted the shading parts below:
Error)
<table>
<tr><th>Cancel Error</th><th>Command Error</th><th>Main Error</th><th>Description</th></tr>
<tr><td>0x0010</td><td>0x230C</td><td>ERC_MEMORY_FAILURE</td><td>The HANGINT error occurred when erasing a sector of the NVM.</td></tr>
<tr><td>0x0010</td><td>0x080C</td><td>ERC_MEMORY_FAILURE</td><td>The NVM Master was either not idle before reading the NVM or does not get ready after sequencer operation (sector erase, blank check)</td></tr>
</table>
Correct)
<table>
<tr><th>Cancel Error</th><th>Command Error</th><th>Main Error</th><th>Description</th></tr>
<tr><td>0x0010</td><td>0x230C</td><td>ERC_MEMORY_FAILURE</td><td>The HANGINT error occurred when erasing a sector of the NVM.</td></tr>
<tr><td>0x0010</td><td>0x540C</td><td>ERC_MEMORY_FAILURE</td><td>SHE could not lock Flash by setting WE bit.</td></tr>
<tr><td>0x0010</td><td>0x5501</td><td>ERC_MEMORY_FAILURE</td><td>SHE may not have properly released Flash IF by clearing WE bit.</td></tr>
<tr><td>0x0010</td><td>0x080C</td><td>ERC_MEMORY_FAILURE</td><td>The NVM Master was either not idle before reading the NVM or does not get ready after sequencer operation (sector erase, blank check)</td></tr>
</table></td></tr>
</table>

Page	Section	Change Results
1457	CHAPTER 26: Secure Hardware Extension (SHE) 6.3.1. CMD_LOAD_KEY_ESHE	Revised the shading parts as below: Error) However the ID and AuthID are evaluated according to Table 4-6. Correct) However the ID and AuthID are evaluated according to Table 6-5.
1500	CHAPTER 27:DMA Controller 4. Register	Added the shading parts as below: Correct) 4.8. DMA Controller Channel Configuration B Register Channel 'n' (DMAi_Bn) [bit15:12] SP[3:0] : Source protection ***Note:*** – *Configuration of SP[3] and SP[2] has no effect.* [bit11:8] DP[3:0] : Destination protection ***Note:*** – *Configuration of DP[3] has an effect.* – *DP[3]=1: Posted write buffer in HPM is used. Due to the posted write behavior, DMA cannot trap the error returned from the destination even if there is an error in the write access (e.g.: The destination register/memory requires Privilege level for writing, but DP[1] is "0").* – *DP[3]=0: Posted write buffer in HPM is unused. DMA always traps the error when there is an error in the write access. (DMAi_Bn:EQ)* – *Configuration of DP[2] has no effect.*
1515	CHAPTER 27: DMA Controller 5. Additional Information	Added the shading parts as below: Notes: When the channel is used with Multi-function Serial Interface to write the transmission data or read the reception data. - When corresponding TXBLKEN (at transmission) or RXBLKEN (at reception) is 0: Configure DMAi_An:BC[3:0] to "0000". - When corresponding TXBLKEN (at transmission) or RXBLKEN (at reception) is 1: Configure DMAi_An:BC[3:0] to a suitable value regarding TBSIZE (at transmission) or FBYTE (at reception).

Page	Section	Change Results
1543	CHAPTER 28: DMA COMPLEX SUBSYSTEM 7. Operation of the 32-Bit Reload Timer in DMA COMPLEX SUBSYSTEM 7.1 Internal Clock and External Event Counter Operations of 32-Bit Reload Timer	Revised the shading parts as below: Figure 7-1 Error)  Correct) 

Page	Section	Change Results
1544	CHAPTER 28: DMA COMPLEX SUBSYSTEM 7. Operation of the 32-Bit Reload Timer in DMA COMPLEX SUBSYSTEM 7.1 Internal Clock and External Event Counter Operations of 32-Bit Reload Timer	Revised the shading parts as below: Figure 7-2 Error)  Correct) 
		Revised the shading parts as below: Figure 7-3 Error)  Correct) 

Page	Section	Change Results
1545	CHAPTER 28: DMA COMPLEX SUBSYSTEM 7. Operation of the 32-Bit Reload Timer in DMA COMPLEX SUBSYSTEM 7.2 Underflow Operation of 32-Bit Reload Timer	Revised the shading parts as below: Figure 7-4 Error)  Correct) 

Page	Section	Change Results
1546	CHAPTER 28: DMA COMPLEX SUBSYSTEM 7.2. Underflow Operation of 32-Bit Reload Timer	Revised the shading parts as below: **Figure 7-5 Clearing of Underflow Bit** Error)  Correct) 
1554	CHAPTER 28: DMA COMPLEX SUBSYSTEM 8. Registers in DMA COMPLEX SUBSYSTEM 8.2. Timer Control Status Register (DMAAn_RLTm_TMCSR)	Deleted the shading parts as below: Deleted) [bit7] DBGE: Debug mode enable - This bit is enable at - from ch. 0 to ch. 3 of CPERI#2 - from ch.0 to ch. 1 of MCU_CONFIG_GROUP

Page	Section	Change Results
1810	CHAPTER 31: CAN FD Controller(MCAN 3.2) 5. Registers	Add below:

Table 5-4 Register Mirror Area CAN FD CONTROLLER

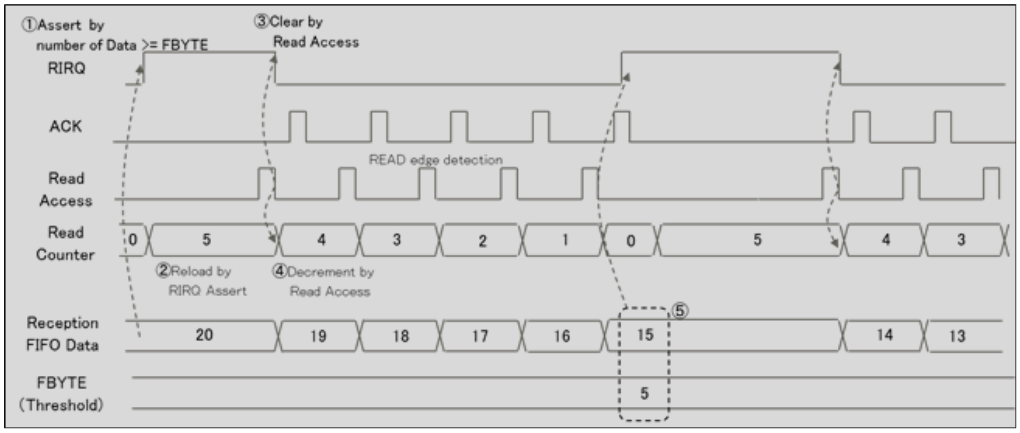
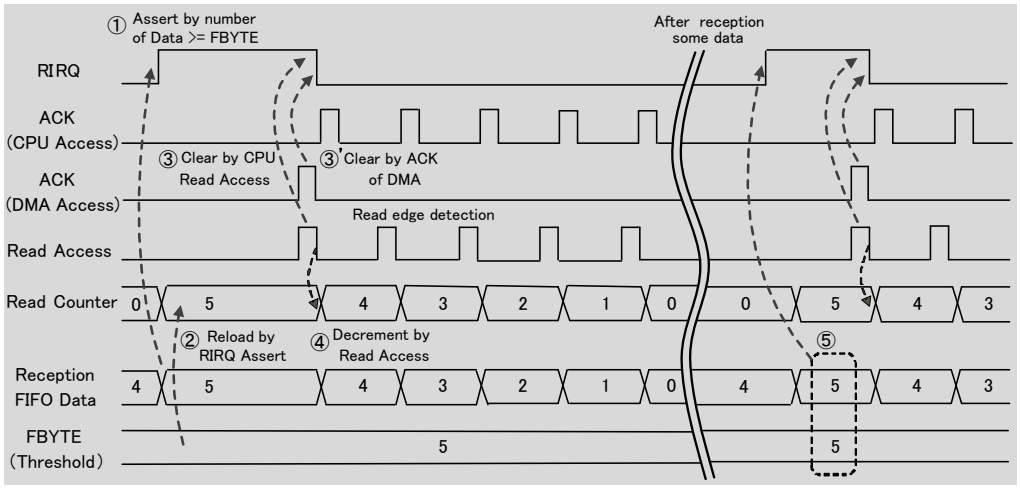
Address of Register Mirror Area	Mirrored Peripheral	Mirror Area Behavior	PPU
0000_C000 to 0000_FFFF	MCU_CONFIG CAN FD ch.0	Accessing this region results to the access to MCU_CONFIG CAN FD ch.0 Message RAM area (0000_0000 to 0000_BFFF)	This area is covered by PPU #47 as well as the mirrored peripheral.
0000_C000 to 0000_FFFF	MCU_CONFIG CAN FD ch.1	Accessing this region results to the access to MCU_CONFIG CAN FD ch.1 Message RAM area (0000_0000 to 0000_BFFF)	This area is covered by PPU #48 as well as the mirrored peripheral.
0000_C000 to 0000_FFFF	MCU_CONFIG CAN FD ch.2	Accessing this region results to the access to MCU_CONFIG CAN FD ch.2 Message RAM area (0000_0000 to 0000_BFFF)	This area is covered by PPU #49 as well as the mirrored peripheral.
0000_C000 to 0000_FFFF	CommonPERI#0 CAN FD ch.0	Accessing this region results to the access to CommonPERI#0 CAN FD ch.0 Message RAM area (0000_0000 to 0000_BFFF)	This area is covered by PPU #256 as well as the mirrored peripheral.
0000_C000 to 0000_FFFF	CommonPERI#0 CAN FD ch.1	Accessing this region results to the access to CommonPERI#0 CAN FD ch.1 Message RAM area (0000_0000 to 0000_BFFF)	This area is covered by PPU #257 as well as the mirrored peripheral.
0000_C000 to 0000_FFFF	CommonPERI#0 CAN FD ch.2	Accessing this region results to the access to CommonPERI#0 CAN FD ch.2 Message RAM area (0000_0000 to 0000_BFFF)	This area is covered by PPU #258 as well as the mirrored peripheral.
0000_C000 to 0000_FFFF	CommonPERI#0 CAN FD ch.3	Accessing this region results to the access to CommonPERI#0 CAN FD ch.3 Message RAM area (0000_0000 to 0000_BFFF)	This area is covered by PPU #259 as well as the mirrored peripheral.
0000_C000 to 0000_FFFF	CommonPERI#0 CAN FD ch.4	Accessing this region results to the access to CommonPERI#0 CAN FD ch.4 Message RAM area (0000_0000 to 0000_BFFF)	This area is covered by PPU #260 as well as the mirrored peripheral.

Page	Section	Change Results
1846	CHAPTER 31: CAN FD Controller(MCAN 3.2) 5.15. Interrupt Register (MCG_CANFDx_IR, CPG_CANFDx_IR)	Revised the shading parts as below: [bit17] MRAF: Message RAM Access Failure Error) In both cases the Rx FIFO put index is not updated resp. the New Data flag for a dedicated Rx Buffer is not set, a partly stored message is overwritten when the next message is stored to this location. Correct) The data of a received frame may be incompletely stored to the Message RAM after the flag is set by the Rx Handler operation. In addition, Rx FIFO 0 Status (RXF0S), Rx FIFO 1 Status (RXF1S), or New Data 1/2 (NDAT1/2) is also updated according as the storage location (Rx FIFO0, Rx FIFO1, or an Rx Buffer). Therefore, in this case discard the received frame data.
		Removed the shading parts as below: Note: - The data of a received frame may be incompletely stored to the Message RAM after the flag is set by the Rx Handler operation. In addition, Rx FIFO 0 Status (RXF0S), Rx FIFO 1 Status (RXF1S), or New Data 1/2 (NDAT1/2) is also updated according as the storage location (Rx FIFO0, Rx FIFO1, or an Rx Buffer). Therefore, in this case discard the received frame data.
1975	CHAPTER 35: Overview of Multi-function Serial Interface 6. Registers of the Multi-Function Serial Interface	Revised the shading parts as below: **Table 6-1 Register Map of UART/CSIO/LIN/I2C in Each Modes** **UART** Error) MCU Config Group (Channel No.: 16, 17, and 18) Notes: - xx is the channel number. (16 to 18) Correct) MCU Config Group (Channel No.: 0, 1, and 2) Notes: - xx is the channel number. (0 to 2)

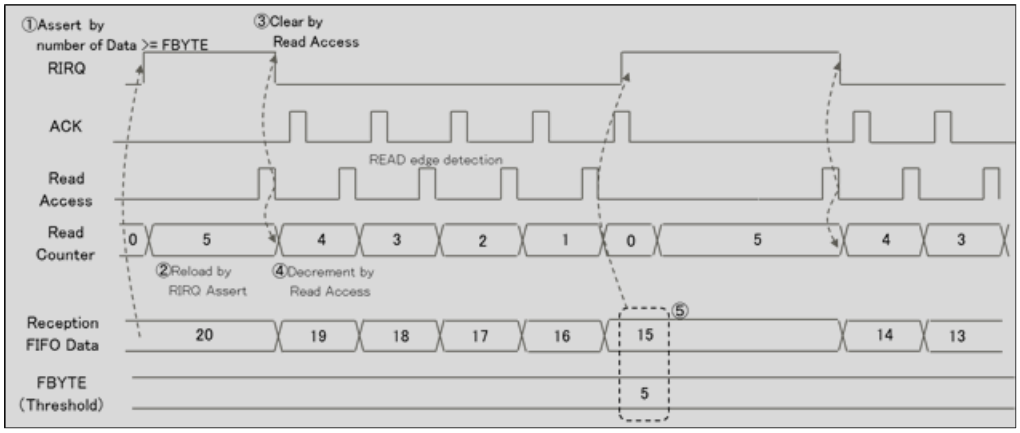
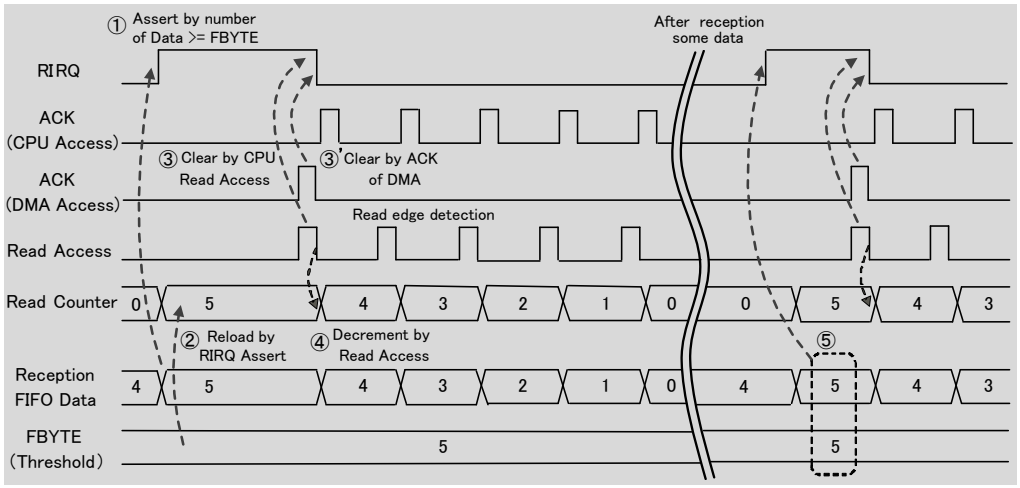
Page	Section	Change Results
1977	CHAPTER 35: Overview of Multi-function Serial Interface 6. Registers of the Multi-Function Serial Interface	Revised the shading parts as below: **Table 6-1 Register Map of UART/CSIO/LIN/I2C in Each Modes** **CSIO** Error) MCU Config Group (Channel No.: 16, 17, and 18) Notes: - xx is the channel number. (16 to 18) Correct) MCU Config Group (Channel No.: 0, 1, and 2) Notes: - xx is the channel number. (0 to 2)
1979	CHAPTER 35: Overview of Multi-function Serial Interface 6. Registers of the Multi-Function Serial Interface	Revised the shading parts as below: **Table 6-1 Register Map of UART/CSIO/LIN/I2C in Each Modes** **LIN** Error) MCU Config Group (Channel No.: 16, 17, and 18) Notes: - xx is the channel number. (16 to 18) Correct) MCU Config Group (Channel No.: 0, 1, and 2) Notes: - xx is the channel number (0 to 2)

Page	Section	Change Results
1981	CHAPTER 35: Overview of Multi-function Serial Interface 6. Registers of the Multi-Function Serial Interface	Revised the shading parts as below: **Table 6-1 Register Map of UART/CSIO/LIN/I2C in Each Modes** **I2C** Error) MCU Config Group (Channel No.: 16, 17, and 18) Notes: - xx is the channel number. (16 to 18) Correct) MCU Config Group (Channel No.: 0, 1, and 2) Notes: - xx is the channel number (0 to 2)
1987	CHAPTER 36: UART (Asynchronous Serial Interface)	Removed the shading parts as below: Error) Note: - ACK Mode of Block Transfer is not supported for this device.
2025	CHAPTER 36: UART (Asynchronous Serial Interface) 7. Block Transfer	Revised the shading parts as below: Error) Block transfer can be performed in that mode in which it is operated with the acknowledgment standard (ACK mode) and in that mode in which it is operated with the READ/WRITE access standard (RW access mode). Correct) Block transfer can be performed in that mode in which it is operated with the READ/WRITE access standard (RW access mode).
		Removed the shading parts as below: Error) Note: - ACK Mode of Block Transfer is not supported for this device.

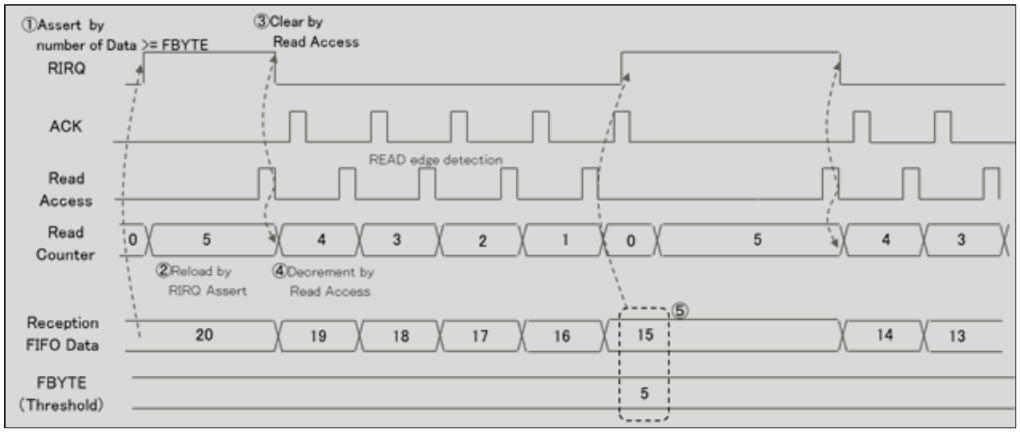
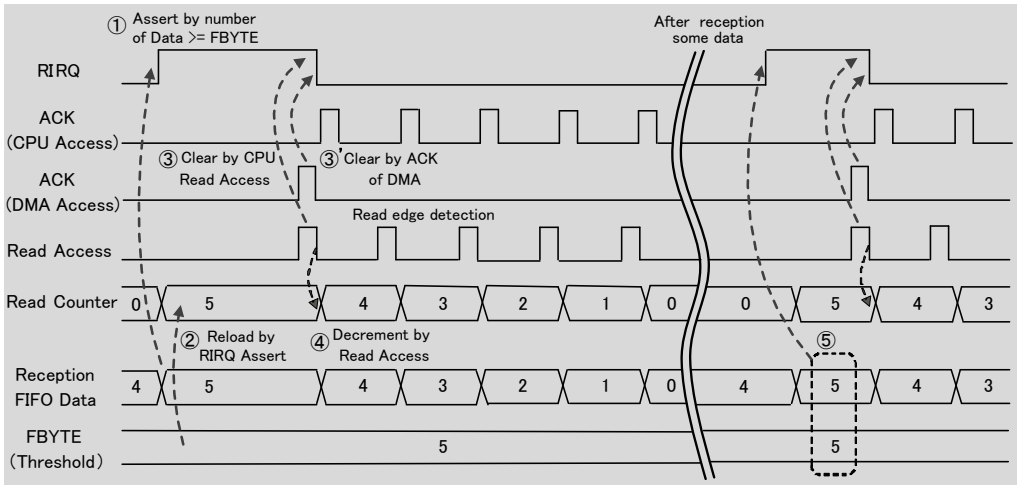
Page	Section	Change Results
2026	CHAPTER 36: UART (Asynchronous Serial Interface) 7.1. ACK Mode	Removed the shading section as below: **7.1. ACK Mode**
2026	CHAPTER 36: UART (Asynchronous Serial Interface) 7.1. RW Access Mode	Revised the shading parts as below: Error) 7.2. RW Access mode Correct) 7.1. RW Access mode
2026	CHAPTER 36: UART (Asynchronous Serial Interface) 7.1. RW Access Mode	Added the shading parts as below: Correct) During application operation, if the next block transfer request occurs before the completion of the previous block transfer, this state might be reported to the system by a block transfer error. This state does not occur if you configure FBYTE=1. Completion of block transfer by DMA depends on occupation of bus transaction by application. You need to fully evaluate the application integrally, design the system, and choose the optimal number of block transfers.

Page	Section	Change Results
2026	CHAPTER 36: UART (Asynchronous Serial Interface) 7.1. RW Access Mode	Revised the shading parts as below: Figure 7.1 Reception Block Transfer in RW Access Mode Error)  Correct) 
2073	CHAPTER 36: UART (Asynchronous Serial Interface) 10.13. FIFO Byte Register (FBYTE)	Removed the shading parts as below: Table 10-4 Data Count Stored in FIFO Error) Note: - Set reception FIFO FBYTE to “2” or a larger value when block transfer. At other conditions, set "1" or a larger value in the FBYTE for the reception FIFO.
2091	CHAPTER 37: CSIO (Clock Synchronous Serial Interface)	Removed the shading parts as below: Error) Note: - ACK Mode of Block Transfer is not supported for this device.

Page	Section	Change Results
2171	CHAPTER 37: CSIO (Clock Synchronous Serial Interface) 8. Block Transfer	Revised the shading parts as below: Error) Block transfer can be performed in that mode in which it is operated with the acknowledgment standard (ACK mode) and in that mode in which it is operated with the READ/WRITE access standard (RW access mode). Correct) Block transfer can be performed in that mode in which it is operated with the READ/WRITE access standard (RW access mode).
		Removed the shading parts as below: Error) Note: - ACK Mode of Block Transfer is not supported for this device.
2172	CHAPTER 37: CSIO (Clock Synchronous Serial Interface) 8.1. ACK Mode	Removed the shading section as below: Error) **8.1. ACK Mode**
2172	CHAPTER 37: CSIO (Clock Synchronous Serial Interface) 8.1. RW Access Mode	Added the shading parts as below: Added) During application operation, if the next block transfer request occurs before the completion of the previous block transfer, this state might be reported to the system by a block transfer error. This state does not occur if you configure FBYTE=1. Completion of block transfer by DMA depends on occupation of bus transaction by application. You need to fully evaluate the application integrally, design the system, and choose the optimal number of block transfers.

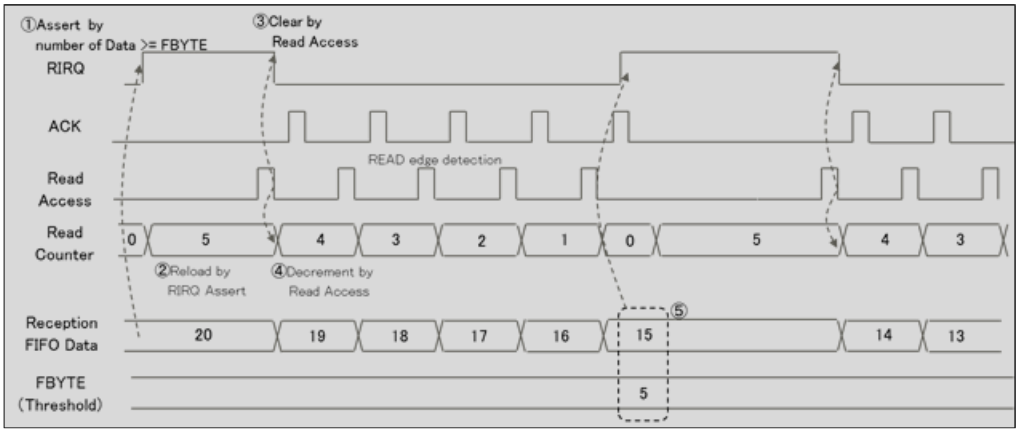
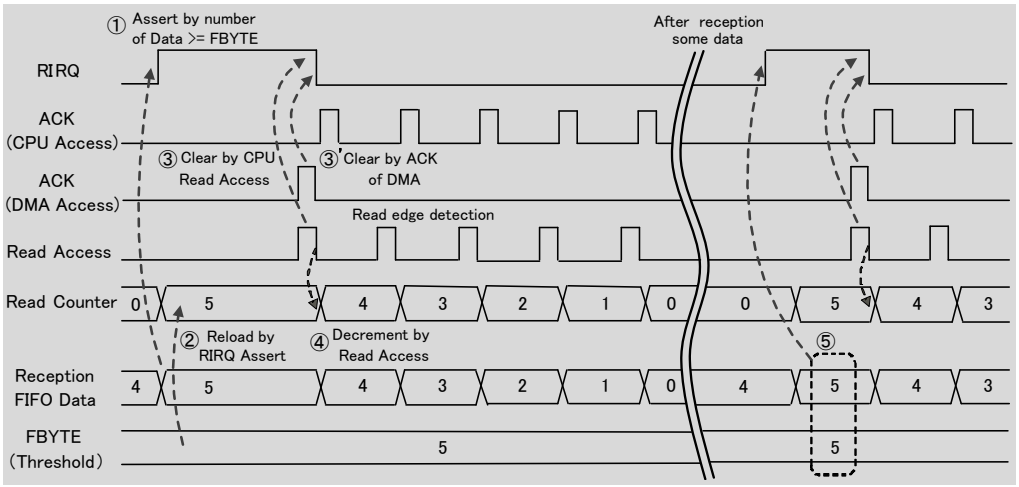
Page	Section	Change Results
2172	CHAPTER 37: CSIO (Clock Synchronous Serial Interface) 8.1. RW Access Mode	Revised the shading parts as below: Figure 8-1 Reception Block Transfer in RW Access Mode Error)  Correct) 
2172	CHAPTER 37: CSIO (Clock Synchronous Serial Interface) 8.1. RW Access Mode	Revised the shading parts as below: Error) 8.2. RW Access mode Correct) 8.1. RW Access mode

Page	Section	Change Results
2229	CHAPTER 37: CSIO (Clock Synchronous Serial Interface) 9.16. FIFO Byte Register (FBYTE)	Removed the shading parts as below: Table 9-4 Data Count Stored in FIFO Error) Note: - Set reception FIFO FBYTE to “2” or a larger value when block transfer. At other conditions, set "1" or a larger value in the FBYTE for the reception FIFO.
2312	CHAPTER 38: I^2C Interface (I^2C Communication Control Interface) 6.Block Transfer	Removed the shading parts as below: 6. Block Transfer Error) Block transfer has two modes: ACK mode and RW access mode. ACK mode operates based on acknowledgment. RW access mode operates based on READ/WRITE access.
2313	CHAPTER 38: I^2C Interface (I^2C Communication Control Interface) 6.1. ACK Mode	Removed the shading section as below: Error) **6.1. ACK Mode**
2313	CHAPTER 38: I^2C Interface (I^2C Communication Control Interface) 6.1. RW Access Mode	Added the shading parts as below: Correct) During application operation, if the next block transfer request occurs before the completion of the previous block transfer, this state might be reported to the system by a block transfer error. This state does not occur if you configure FBYTE=1. Completion of block transfer by DMA depends on occupation of bus transaction by application. You need to fully evaluate the application integrally, design the system, and choose the optimal number of block transfers.

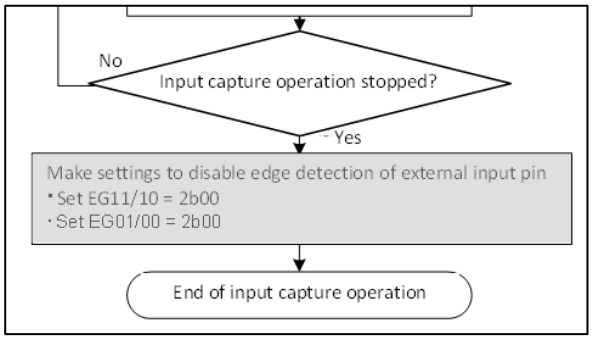
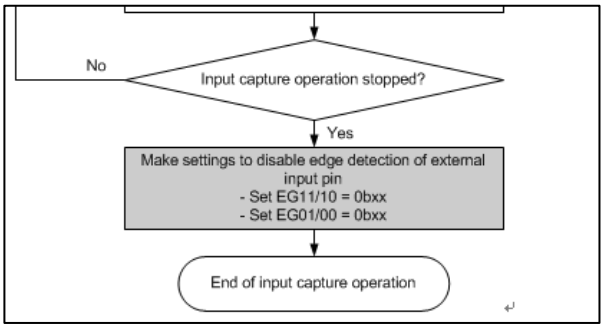
Page	Section	Change Results
2313	CHAPTER 38: I²C Interface (I²C Communication Control Interface) 6.1. RW Access Mode	Revised the shading parts as below: Figure 6-1 Reception Block Transfer in RW Access Mode Error)  Correct) 
2313	CHAPTER 38: I²C Interface (I²C Communication Control Interface) 6.1. RW Access Mode	Revised the shading parts as below: Error) 6.2. RW Access mode Correct) 6.1. RW Access mode

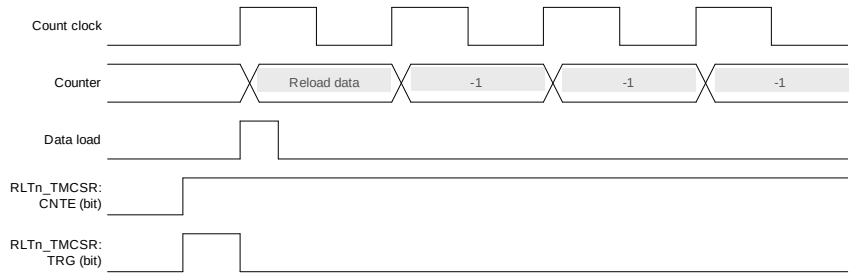
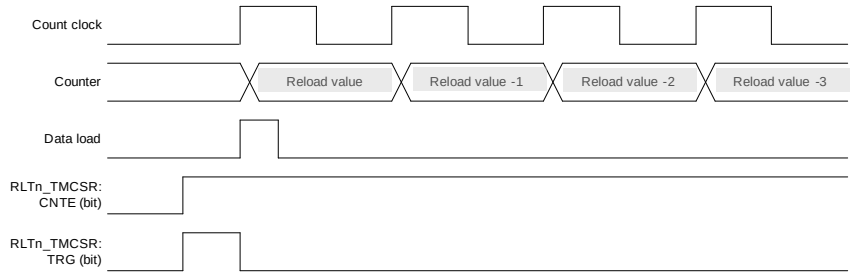
Page	Section	Change Results
2335	CHAPTER 38: I²C Interface (I²C Communication Control Interface) 8.2. Serial Mode Register (SMR)	Added the shading parts as below: **[bit4] WUCR: WAKE UP Control Bit** Note: − The WAKE UP function is not supported.
2369	CHAPTER 38: I²C Interface (I²C Communication Control Interface) 8.16. FIFO Byte Register (FBYTE)	Removed the shading parts as below: Table 8-4 Data Count Stored in FIFO Error) Note: - Set reception FIFO FBYTE to “2” or a larger value when block transfer. At other conditions, set "1" or a larger value for FBYTE of the reception FIFO.
2397	CHAPTER 39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1))	Removed the shading parts as below: Error) Note: - ACK Mode of Block Transfer is not supported for this device.
2397	CHAPTER 39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1))	Deleted the shading parts as below: Error) This chapter describes the LIN communication function supported by operation mode 3 of the multi-function serial interface functions. The LIN communication function has the following 2 modes: Manual mode, which allows the transmission/reception of the LIN header section using interrupt functions, and the assist mode, which allows automatic transmission/reception of the LIN header.
2416	CHAPTER 39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 2.2.1. Occurrence of Reception Interrupts and Flag Set Timing in Assist Mode	Revised the shading parts as below: Error) Occurrence of Framing Error Iterrupt and Flag Set Timing Correct) Occurrence of Framing Error Interrupt and Flag Set Timing

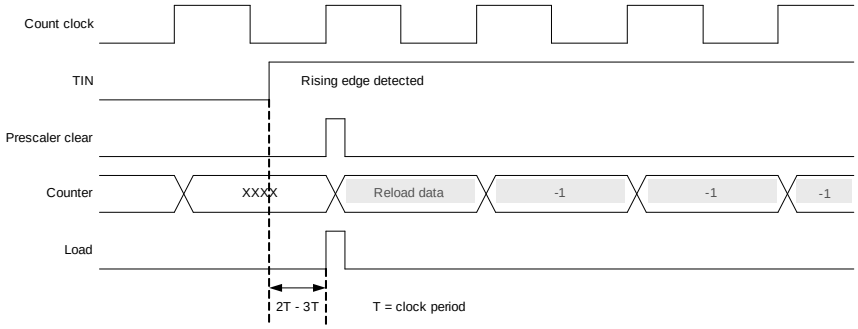
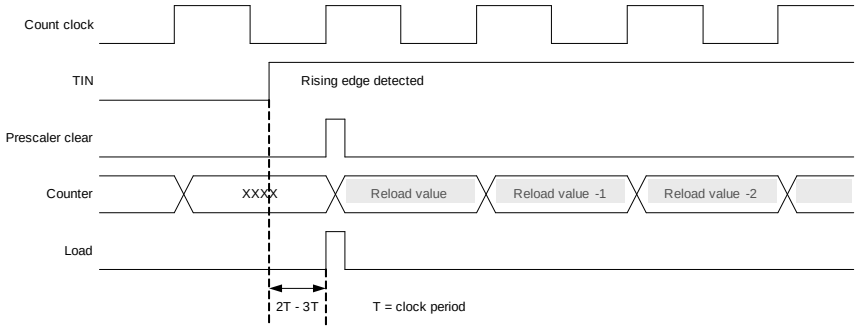
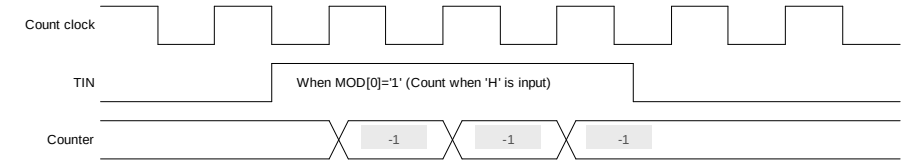
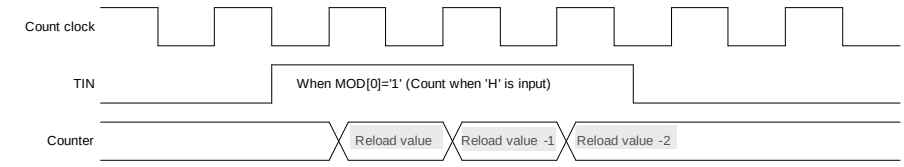
Page	Section	Change Results
2447	CHAPTER 39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 6. Block Transfer	Revised the shading parts as below: Error) Block transfer can be performed in that mode in which it is operated with the acknowledgment standard (ACK mode) and in that mode in which it is operated with the READ/WRITE access standard (RW access mode). Correct) Block transfer can be performed in that mode in which it is operated with the READ/WRITE access standard (RW access mode).
		Removed the shading parts as below: Error) Note: - ACK Mode of Block Transfer is not supported for this device.
2448	CHAPTER 39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 6.1. ACK Mode	Removed the shading section as below: Error) **6.1. ACK Mode**
2448	CHAPTER 39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 6.1. RW Access Mode	Added the shading parts as below: Correct) During application operation, if the next block transfer request occurs before the completion of the previous block transfer, this state might be reported to the system by a block transfer error. This state does not occur if you configure FBYTE=1. Completion of block transfer by DMA depends on occupation of bus transaction by application. You need to fully evaluate the application integrally, design the system, and choose the optimal number of block transfers.

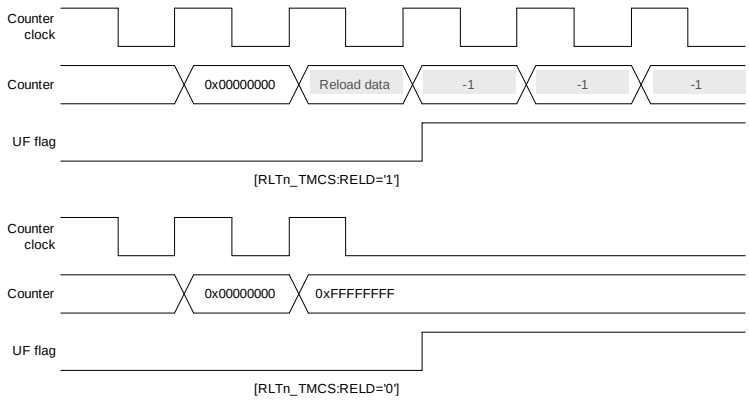
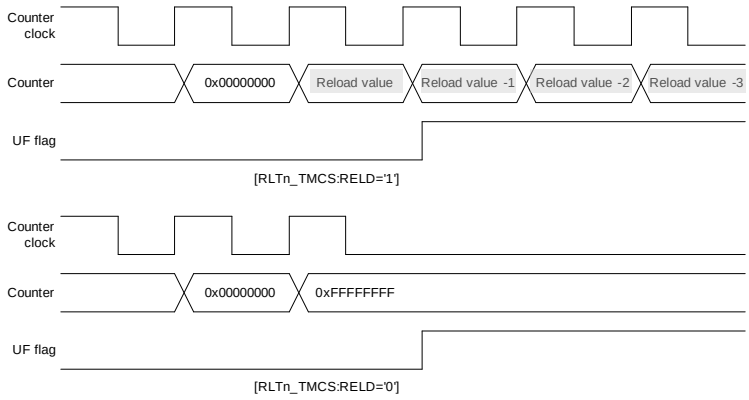
Page	Section	Change Results
2448	CHAPTER 39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 6.1. RW Access Mode	Revised the shading parts as below: Figure 6-1 Reception Block Transfer in RW Access Mode Error)  Correct) 
2448	CHAPTER 39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 6.1. RW Access Mode	Revised the shading parts as below: Error) 6.2. RW Access mode Correct) 6.1. RW Access mode

Page	Section	Change Results
2502	CHAPTER 39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.1. Serial Control Registor (SCR)	Revised the shading parts as below: Error) - a LINN Break Field is generated. Correct) - a LIN Break Field is generated.
2532	CHAPTER 39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.14 LIN Assist Mode Interrupt Enable Register (LAMIER)	Revised the shading parts as below: [bit14] LCSERIE: LIN Checksum Error Interrupt Enable Bit Error) LAMIERS:LCERIES Correct) LAMIERS:LCSERIES
2549	CHAPTER 39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.20. FIFO Byte Register (FBYTE)	Removed the shading parts as below: Table 9-3 Data Count Stored in FIFO Error) Note: - During block transfer, set "2" or a larger value in FBYTE for the reception FIFO. For the others, set "1" or a larger value in FBYTE for the reception FIFO.
2733	CHAPTER 42: 32-bit Free-run Timer 5. Precautions for Using This Device	Revised the shading parts as below: Error) This register supports writing from the bit-band alias area. Correct) This register doesn't support writing from the bit-band alias area.

Page	Section	Change Results
2740	CHAPTER 43: 32-bit Input Capture 3. Explanation of 32-Bit Input Capture Operation 3.2. Setting Procedure Example	Revised the shading parts as below: Error)  Correct) 

Page	Section	Change Results
2802	CHAPTER 45: 32-bit Reload Timer 3. Operation of the 32-Bit Reload Timer 3.1. Internal Clock and External Event Counter Operations of 32-Bit Reload Timer	Revised the shading parts as below: Figure 3-1 Error)  Correct) 

Page	Section	Change Results
2803	CHAPTER 45: 32-bit Reload Timer 3. Operation of the 32-Bit Reload Timer 3.1. Internal Clock and External Event Counter Operations of 32-Bit Reload Timer	Revised the shading parts as below: Figure 3-2 Error)  Correct) 
		Revised the shading parts as below: Figure 3-3 Error)  Correct) 

Page	Section	Change Results
2804	CHAPTER 45: 32-bit Reload Timer 3. Operation of the 32-Bit Reload Timer 3.2. Underflow Operation of 32-Bit Reload Timer	Revised the shading parts as below: Figure 3-4 Error)  Correct) 
2936	CHAPTER 51: Peripheral Protection Unit 1. Overview	Revised the shading parts as below: Error) Refer PLATFORM OVERVIEW chapter to confirm which bus master is affected by the PPU. Correct) Only CPU is the bus master which is affected by PPU.

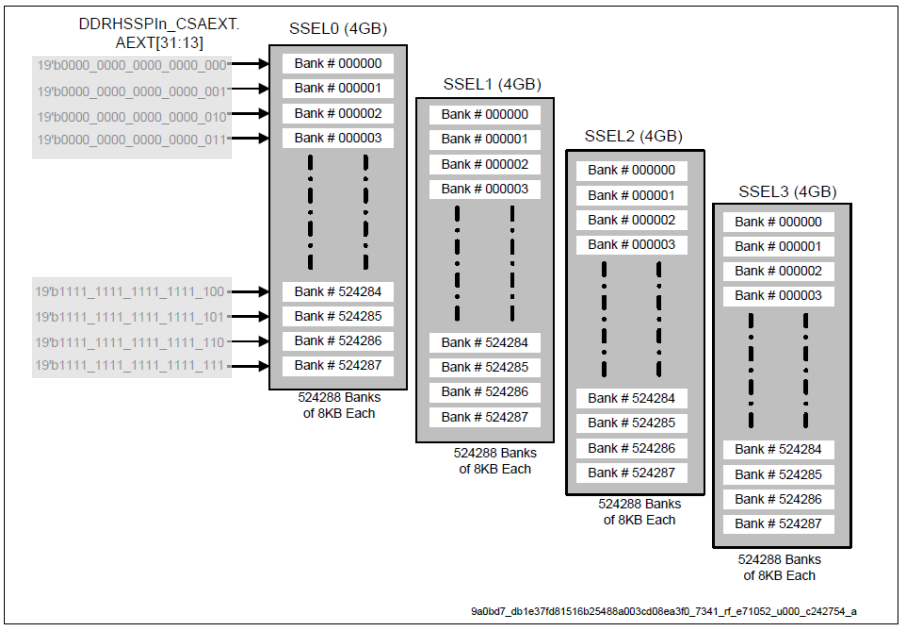
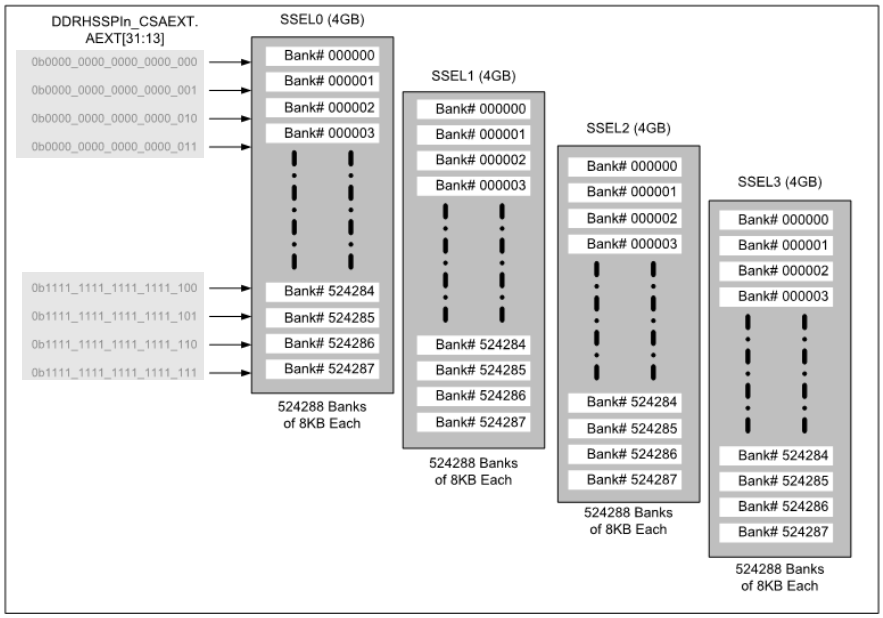
<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>2951</td><td>CHAPTER 51: Peripheral Protection Unit
4.2. PPU Status Register (PPU0_SR)</td><td>Revised the shading parts as below:

[bit20:16] VL[4:0] : Violation Location

Error)
<table>
<tr><th>BIts</th><th>Description</th></tr>
<tr><td>00011</td><td>Violated access was done to a register in Memory Config Group or Scratch Pad RAM.</td></tr>
</table>
Correct)
<table>
<tr><th>BIts</th><th>Description</th></tr>
<tr><td>00011</td><td>Violated access was done to a register in Memory Config Group or System SRAM.</td></tr>
</table>
</td></tr>
<tr><td>2952</td><td>CHAPTER 51: Peripheral Protection Unit
4.2. PPU Status Register (PPU0_SR)</td><td>Revised the shading parts as below:

Note:
− Priority of settings to PPU0_SR in the case where the access violation occurs simultaneously In several groups is as follows.

Error)
<table>
<tr><th>Group Name</th><th>Priorities</th></tr>
<tr><td>Scratch Pad RAM</td><td></td></tr>
</table>
Correct)
<table>
<tr><th>Group Name</th><th>Priorities</th></tr>
<tr><td>System SRAM</td><td></td></tr>
</table>
</td></tr>
</table>

Page	Section	Change Results
3008	CHAPTER 52: DDR High Speed SPI Controller 4. Operations of the DDRHSSPI	Changed the below: **4.2. Command Sequencer Mode** **Figure 4-6 Addressing 4GB Serial Flash Memories on Each Slave Select, Through Different Bank (DDRHSSPIn_CSCFG.MSEL= 0b0000)** Error)  Correct) 

Page	Section	Change Results
3023	CHAPTER 52: DDR High Speed SPI Controller 4. Operations of the DDRHSSPI	Revised the shading parts as below: **4.4.4. Using the DDRHSSPI in Command Sequencer Mode of Operation** Error) 2.The next step is to configure the transfer protocol (i.e. whether the DDRHSSPI serial transfers use the Quad or Octal Protocol in the DDRHSSPIn_CSCFG.MBM). The DDRHSSPIn_CSCFG. DDRMODE bit shall be set same as DDRHSSPIn_DMFIFOCFG.DDRM bit. Correct) 2.The next step is to configure the transfer protocol (i.e. whether the DDRHSSPI serial transfers use the Quad or Octal Protocol in the DDRHSSPIn_CSCFG.MBM). The DDRHSSPIn_CSCFG. DDRMODE bit shall be set same as DDRHSSPIn_DMTRP.DDRM bit.
3030	CHAPTER 52: DDR High Speed SPI Controller 4. Operations of the DDRHSSPI	Deleted the below: **4.4. General Use Case Guidelines for DDRHSSPI** 4.4.5. Notice On Document Designations Cypress issues documents and data sheets with Advance Information or Preliminary designations to advise readers of product information or intended specifications throughout the product life cycle,including development, qualification, initial production, and full production. In all cases, however, readers are encouraged to verify that they have the latest information before finalizing their design. The following descriptions of Cypress data sheet designations are presented here to highlight their presence and definitions 4.4.6. Advance Information The Advance Information designation indicates that Cypress Semiconductor Corp. is developing one or more specific products, but has not committed any design to production. Information presented in a document with this designation is likely to change, and in some cases, development on the product may discontinue. Cypress Semiconductor Corp. therefore places the following conditions upon Advance Information content: This document contains information on one or more products under development at Cypress Semiconductor Corp. The information is intended to help you evaluate this product. Do not design in this product without contacting the Cypress Semiconductor Corp. Cypress Semiconductor Corp. reserves the right to change or discontinue work on this proposed product without notice.

Page	Section	Change Results
3119, 3120	CHAPTER 53: System SRAM Module 4.4. SRAM_IF Unlock/Lock Key Register (SRCFG_KEY)	Added the shading parts as below: Correct) [bit31:0] UNLOCK[31:0]: SRAM_IF Unlock or Lock Key Locked registers are as follows. - SRCFG_CFG0 - SRCFG_CFG1 - SRCFG_CFG - SRCFG_ERRFLG - SRCFG_INTE - SRCFG_ECCE - SRCFG_ERRADR
3133 to 3152	Major Changes	Added the contents of Errata Document (002-04822) as Major changes from MN708-00006-1v0-E to MN708-00006-2v0-E
002-04854 Rev. *G		
153	CHAPTER 5: Clock System 5. Registers 5.1. Common Configuration Registers 5.1.2. Main Oscillator Control Register(SYSC_MOSCCNTR)	Added the shading parts as below: **[bit31] MCMODE : Main Clock Amplifier Oscillation Mode Bit** Correct) ***Note:*** *The initial value (value "0") of this bit specifies High drive mode.* *To reduce noise, set the mode to low drive mode (value "1") after transition to the user program.*
154		Added the shading parts as below: **[bit27:26] MCMODE : MCGAIN[1:0] : Main Clock GAIN Bit** Correct) ***Note:*** *These frequency values are just reference value.* *Set optimum value according to oscillator matching evaluation.* *Generally, MCMODE should be "1" (Stop mode) to optimize by MCGAIN setting.*

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td rowspan="2">618</td><td rowspan="2">CHAPTER 8: Clock Supervisor
3. Explanation of Operation</td><td>Revised the shading parts as below:

PLL Clock Supervisor

Error)

■In case of PLL1, PLL2 or PSS3, reset generation or interrupt generation can be selected with the judgment selection bit (JDGSEL).

Correct)
■In case of PLL1, PLL2 or PLL3, reset generation or interrupt generation can be selected with the judgment selection bit (JDGSEL).</td></tr>
<tr><td>Revised the shading parts as below:

SSCG PLL Clock Supervisor

Error)
■In case of SSCG1, SSCG or SSCG3, reset generation or interrupt generation can be selected with the judgment selection bit (JDGSEL).

Correct)
■In case of SSCG1, SSCG2 or SSCG3, reset generation or interrupt generation can be selected with the judgment selection bit (JDGSEL).</td></tr>
<tr><td>634</td><td>CHAPTER 8: Clock Supervisor
5. Registers
5.6. PLLm Clock Supervisor Setting Register 1 (SYSC_CSVPLLmCFGR1)</td><td>Revised the shading parts as below:

[bit16] JDGSEL: Judgment selection bit

Error)
<table><tr><th>Bit</th><th>Description</th></tr><tr><td>0</td><td>PLLm clock is selected for clock domain 0 or for MCUC clock domain.</td></tr><tr><td>1</td><td>- (same condition)</td></tr></table>
Correct)
<table><tr><th>Bit</th><th>Description</th></tr><tr><td>0</td><td>Main clock is selected for clock domain 0 , MCUC clock domain or software watchdog timer.</td></tr><tr><td>1</td><td>Main clock is selected for clock domain 0 or MCUC clock domain</td></tr></table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>638</td><td>CHAPTER 8: Clock Supervisor 5. Registers 5.8. SSCG PLLn Clock Supervisor Setting Register 1 (SYSC_CSVSPnCFGR1)</td><td>Revised the shading parts as below:

[bit16] JDGSEL: Judgment selection bit

Error)
<table><tr><th>Bit</th><th>Description</th></tr><tr><td>0</td><td>SSCG PLLn clock is selected for clock domain 0 or for MCUC clock domain.</td></tr><tr><td>1</td><td>- (same condition)</td></tr></table>
Correct)
<table><tr><th>Bit</th><th>Description</th></tr><tr><td>0</td><td>Main clock is selected for clock domain 0 , MCUC clock domain or software watchdog timer.</td></tr><tr><td>1</td><td>Main clock is selected for clock domain 0 or MCUC clock domain</td></tr></table></td></tr>
<tr><td>865</td><td>CHAPTER 14: Hardware Watchdog Timer 6. Register List 6.1. Hardware Watchdog Protection Register (HWDG_PROT)</td><td>Added the shading parts as below:

Error)
– A protect key will be locked again by writing to the address where is in the same group area (MCU Config Group), however protect key will not be locked by writing to no protect target register.

Correct)
– A protect key will be locked again by writing to the address where is in the same group area (MCU Config Group), however protect key will not be locked by writing to no protect target register (HWDG_TRG0/1) in Hardware Watchdog.</td></tr>
<tr><td>918</td><td>CHAPTER 15: Hardware Watchdog Timer 6. Register List 6.1. Software Watchdog Protection Register (SWDG_PROT)</td><td>Added the shading parts as below:

Error)
– A protect key will be locked again by writing to the address where is in the same group area(SYSC1 Group), however protect key will not be locked by writing to no protect target register.

Correct)
– A protect key will be locked again by writing to the address where is in the same group area(SYSC1 Group), however protect key will not be locked by writing to no protect target register (SWDG_TRG0/1) in Software Watchdog.</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td rowspan="2">1106</td><td rowspan="2">CHAPTER 18: WorkFLASH
6. Other</td><td>Removed the shading parts as below:

Correct)
Handling the Values Read from Reserved Bits
"0" is read from the reserved bits in the registers in WorkFLASH. From the viewpoint of ensuring compatibility with future products and software, make sure that in programming, no significance is attached to any value that is read from a reserved bit.</td></tr>
<tr><td>Added the shading parts as below:

Correct)
WorkFLASH Status Register Bits
Since some status information of flash memory is common to Code side (TCFLASH) and Work side
(WorkFLASH) in Dual Port FLASH macro,
WorkFLASH status register bits listed below are affected by TCFLASH operation in addition to WorkFLASH operation:
- WFCFG_SR.ESPS
Use this bit only when program/erase is performed in WorkFLASH.</td></tr>
<tr><td>1485</td><td>CHAPTER 27: DMA Controller
4. Registers</td><td>Removed the shading parts as below:

Table 4-1 List of Register Area

Error)
<table><tr><td>0x0000_2800(DMAC#0)
0x0000_6800(DMAC#1)
The reserve area is changed by the following arithmetic expression.
+ (n*0x04)</td><td>DMAi_CMCHICn
00000000_00000000_00000000_0000*1</td></tr></table>
Correct)
<table><tr><td>0x0000_2800(DMAC#0)
0x0000_6800(DMAC#1)
+ (n*0x04)</td><td>DMAi_CMCHICn
00000000_00000000_00000000_0000*1</td></tr></table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1841</td><td>CHAPTER 31: CAN FD Controller(MCAN 3.2)
5. Registers
5.14. Transmitter Delay Compensation Register (MCG_CANFDx_TDCR, CPG_CANFDx_TDCR)</td><td>Revised the shading parts as below:

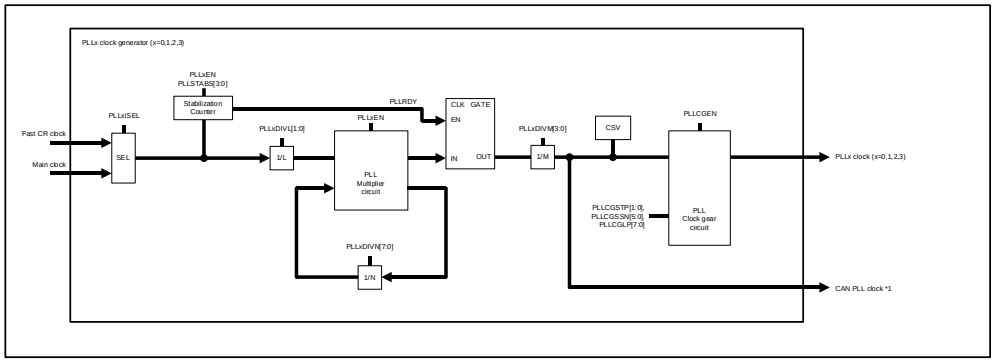
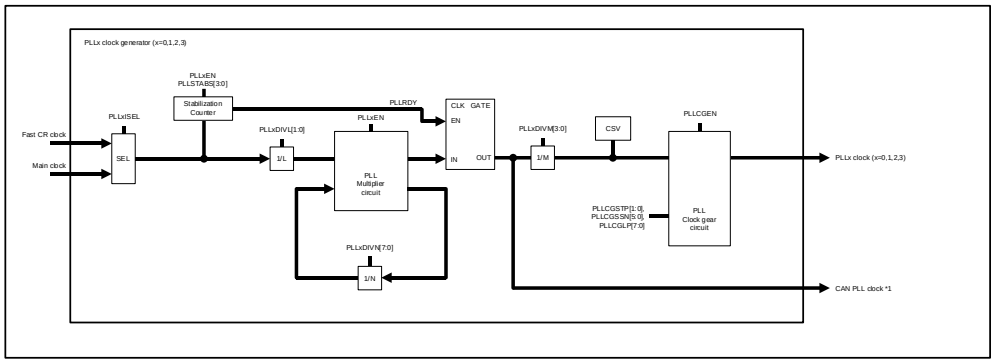
Error)
<table>
<tr><td>bit</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td>Field</td><td colspan="8">Reserved</td></tr>
<tr><td>R/W Attribute</td><td colspan="8">RX,W0</td></tr>
<tr><td>Protection Attribute</td><td colspan="8">-</td></tr>
<tr><td>Initial value</td><td colspan="8">0x00</td></tr>
</table>
<table>
<tr><td>bit</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td>Field</td><td colspan="8">Reserved</td></tr>
<tr><td>R/W Attribute</td><td colspan="8">RX,W0</td></tr>
<tr><td>Protection Attribute</td><td colspan="8">-</td></tr>
<tr><td>Initial value</td><td colspan="8">0x00</td></tr>
</table>
<table>
<tr><td>bit</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td>Field</td><td>Reserved</td><td colspan="7">TDCO[6:0]</td></tr>
<tr><td>R/W Attribute</td><td colspan="8">R/W</td></tr>
<tr><td>Protection Attribute</td><td colspan="8">-</td></tr>
<tr><td>Initial value</td><td colspan="8">0x00</td></tr>
</table>
<table>
<tr><td>bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Field</td><td>Reserved</td><td colspan="7">TDCF[6:0]</td></tr>
<tr><td>R/W Attribute</td><td colspan="8">R/W</td></tr>
<tr><td>Protection Attribute</td><td colspan="8">-</td></tr>
<tr><td>Initial value</td><td colspan="8">0x00</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td></td><td></td><td>Correct)
<table>
<tr><td>bit</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td>Field</td><td colspan="8">Reserved</td></tr>
<tr><td>R/W Attribute</td><td colspan="8">R0,W0</td></tr>
<tr><td>Protection Attribute</td><td colspan="8">-</td></tr>
<tr><td>Initial value</td><td colspan="8">0x00</td></tr>
</table>
<table>
<tr><td>bit</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td>Field</td><td colspan="8">Reserved</td></tr>
<tr><td>R/W Attribute</td><td colspan="8">R0,W0</td></tr>
<tr><td>Protection Attribute</td><td colspan="8">-</td></tr>
<tr><td>Initial value</td><td colspan="8">0x00</td></tr>
</table>
<table>
<tr><td>bit</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td>Field</td><td>Reserved</td><td colspan="7">TDCO[6:0]</td></tr>
<tr><td>R/W Attribute</td><td>R0, W0</td><td colspan="7">R/W</td></tr>
<tr><td>Protection Attribute</td><td>-</td><td colspan="7">-</td></tr>
<tr><td>Initial value</td><td>0</td><td colspan="7">0x00</td></tr>
</table>
<table>
<tr><td>bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Field</td><td>Reserved</td><td colspan="7">TDCF[6:0]</td></tr>
<tr><td>R/W Attribute</td><td>R0, W0</td><td colspan="7">R/W</td></tr>
<tr><td>Protection Attribute</td><td>-</td><td colspan="7">-</td></tr>
<tr><td>Initial value</td><td>0</td><td colspan="7">0x00</td></tr>
</table></td></tr>
<tr><td>2181</td><td>CHAPTER 37: CSIO (Clock Synchronous Serial Interface) 9. Registers of the CSIO (Clock Synchronous Serial Interface) 9.2. Serial Mode Register (SMR)</td><td>Added the shading parts as below:

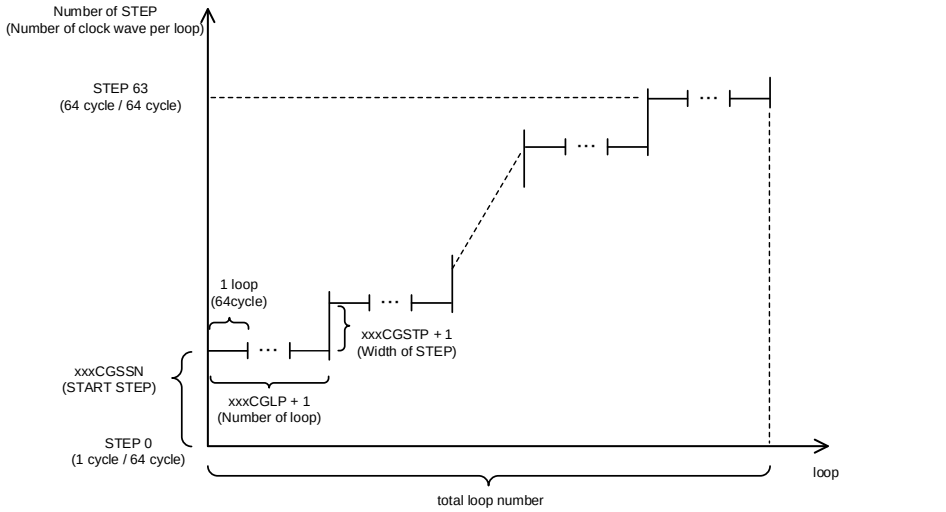
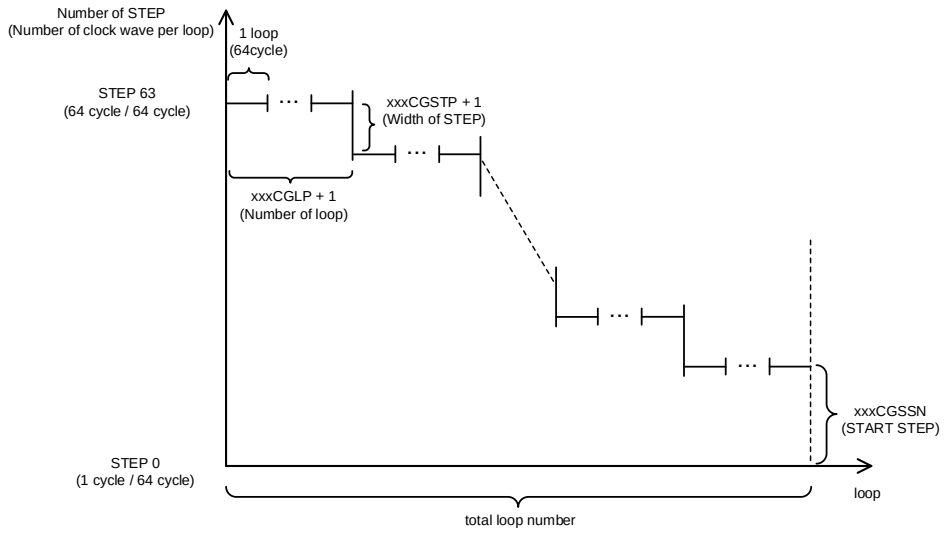
[bit4] WUCR: WAKE UP Control Bit

Correct)
Note:
– The WAKE UP function is not supported.</td></tr>
</table>

Page	Section	Change Results
2505	CHAPTER 39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9. Interface (V2.1) Registers 9.2. Serial Mode Register (SMR)	Added the shading parts as below: **[bit4] WUCR: WAKE UP Control Bit** Correct) ***Note:*** – *The WAKE UP function is not supported.*

Page	Section	Change Results
002-04854 Rev. *H		
126	CHAPTER 5: 2. Configuration and Block Diagram 2.2. Clock Generator Configuration and Block Diagram 2.2.1. PLL Clock Generator Configuration and Bloack Diagram	Revised the below figure: **Figure 2-3 PLLx (x=0,1,2,3)Clock Generator Block Diagram** Error)  Correct) 

Page	Section	Change Results
138	CHAPTER 5: Clock System 3.4. Clock Gear	Revised the shading parts as below: ***Note:*** Error) *STEP 0 to 63:* *STEP 0 to 63: Clock output patterns of Gear operation.* *As the number of step increases, the clock output increases.* *STEP 0 to 1: clock/loop* *STEP 1 to 2: clock/loop* *:* *:* *:* *STEP 63 to 64 clock/loop (Clock output for all cycles in loop)* Correct) *STEP 0 to 63: Clock output patterns of Gear operation.* *As the number of step increases, the clock output increases.* *STEP 0 : 1 clock/loop* *STEP 1 : 2 clock/loop* *:* *:* *:* *STEP 63 : 64 clock/loop (Clock output for all cycles in loop)*

Page	Section	Change Results
140	CHAPTER 5: Clock System 3.4. Clock Gear	Added Figure 3-2 and Figure 3-3. Correct) Figure.3-2(Gear Up) and Figure.3-3(Gear Down) show the relationship between the register configuration and the Clock Gear STEP. **Figure 0-1 Clock Gear STEP by register configuration (Gear Up)**  Transition time of clock gear = total loop number * 64cycle * (PLL or SSCG period) = (xxxCGLP + 1) * (64 − xxxCGSSN) / (xxxCGSTP + 1) * 64cycle * (PLL or SSCG period) Note: The number is rounded up after the decimal point in "(64 − xxxCGSSN) / (xxxCGSTP + 1)". **Figure 0-2 Clock Gear STEP by register configuration (Gear Down)**  Transition time of clock gear = Same as gear up (Figure.3-2)

Page	Section	Change Results
994	CHAPTER 17: TCFLASH 3. Explanation of Operation 3.2. Programming and Erasing	Added the shading parts as below: Correct) ■After erase operation completed, all the values of the flash memory cells in the target sectors are "1".
1055	CHAPTER 18: WorkFLASH 3. Explanation of Operation 3.2. Write and Erase	Revised the shading parts as below: Error) Overwriting of the same data bit is prohibited. (Even overwriting of 1 -> 0, 0 -> 0, or 1 -> 1 is also prohibited.) The Flash security function does not allow writing and erasing of protected sectors. For details, see the security specifications. Correct) Overwriting of the same data bit is prohibited. (Even overwriting of 1 -> 0, 0 -> 0, or 1 -> 1 is also prohibited.) After erase operation completed, all the values of the flash memory cells in the target sectors are "1". The Flash security function does not allow writing and erasing of protected sectors. For details, see the security specifications.
1605	CHAPTER 30: CAN FD Controller(MCAN 3.2) 3.5: Tx Handling 3.5.2. Dedicated Tx Buffers	Added the shading parts as below: Correct) Dedicated Tx Buffers are intended for message transmission under complete control of the CPU. Each Dedicated Tx Buffer is configured with a specific Message ID. In case that multiple Tx Buffers are configured with the same Message ID, the Tx Buffer with the lowest buffer number is transmitted first. These Tx buffers shall be requested in ascending order with lowest buffer number first. Alternatively all Tx buffers configured with the same Message ID can be requested simultaneously by a single write access to TXBAR.

Page	Section	Change Results
1606	CHAPTER 30: CAN FD Controller(MCAN 3.2) 3.5: Tx Handling 3.5.4. Tx Queue	Revised the shading parts as below: Error) Tx Queue operation is configured by programming the Tx FIFO/Queue Mode bit TXBC.TFQM to "1". Messages stored in the Tx Queue are transmitted starting with the message with the lowest Message ID (highest priority). In case that multiple Queue Buffers are configured with the same Message ID, the Queue Buffer with the lowest buffer number is transmitted first. Correct) Tx Queue operation is configured by programming the Tx FIFO/Queue Mode bit TXBC.TFQM to "1". Messages stored in the Tx Queue are transmitted starting with the message with the lowest Message ID (highest priority). If multiple Tx Queue buffers are configured with the same Message ID, the transmission order depends on numbers of the buffers where the messages were stored for transmission. As these buffer numbers depend on the then current states of the PUT index, a prediction of the transmission order is not possible.
1606	CHAPTER 30: CAN FD Controller(MCAN 3.2) 3.5: Tx Handling 3.5.4. Tx Queue	Revised the shading parts as below: Error) **Adding Messages and Requesting Transmissions** New messages have to be written to the Tx Buffer referenced by the Tx FIFO/Queue Put Index TXFQS.TFQPI[4:0]. An Add Request cyclically increments the Put Index to the next free Tx Buffer. Correct) **Adding Messages and Requesting Transmissions** New messages have to be written to the Tx Buffer referenced by the Tx FIFO/Queue Put Index TXFQS.TFQPI[4:0]. The PUT Index always points to that free buffer of the Tx Queue with the lowest buffer number.
2041	CHAPTER 36: UART (Asynchronous Serial Interface) 10.Registers of UART (Asynchronous Serial Interface) 10.2. Serial Mode Register (SMR)	Added the shading parts as below: Correct) [bit4] WUCR: WAKE UP Control Bit WUCR is not supported This bit selects a pin used for external interrupts.

Page	Section	Change Results
2179	CHAPTER 37: CSIO (Clock Synchronous Serial Interface) 9. Registers of the CSIO (Clock Synchronous Serial Interface) 9.2. Serial Mode Register (SMR)	Added the shading parts as below: Correct) [bit4] WUCR: WAKE UP Control Bit WUCR is not supported This bit selects a pin used for external interrupts.
2332	CHAPTER 38: I2C Interface (I2C Communication Control Interface) 8. I^2C Interface Registers 8.2. Serial Mode Register (SMR)	Added the shading parts as below: Correct) [bit4] WUCR: WAKE UP Control Bit WUCR is not supported This bit selects a pin used for external interrupts.
2503	CHAPTER 39: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9. Interface (V2.2) Registers 9.2. Serial Mode Register (SMR)	Added the shading parts as below: Correct) [bit4] WUCR: WAKE UP Control Bit WUCR is not supported This bit selects a pin used for external interrupts.

Revision History

Document Revision History

Document Title: 32-Bit Microcontroller TRAVEO™ T1G Family S6J32XX Series Hardware Manual Platform Part Document Number: 002-04854			
Revision	**ECN No.**	**Date**	**Description of Change**
**	-	08/07/2015	Initial release
*A	5371852	07/26/2016	Migrated to Cypress format
*B	5454251	11/28/2016	Added bookmarks
*C	5697668	04/17/2017	Fixed some clerical errors. For details, please see the chapter 55 of Appendix: Major Changes.
*D	6085216	03/02/2018	Fixed some clerical errors. For details, please see the chapter 55 of Appendix: Major Changes.
*E	6658068	08/20/2019	Fixed some clerical errors. For details, please see the chapter 55 of Appendix: Major Changes.
*F	6663935	08/29/2019	Updated the resolution of Figure 3-1 Block Diagram on Chapter 1: Platform Overview.
*G	6883245	05/26/2020	Fixed some clerical errors. For details, please see the chapter 55 of Appendix: Major Changes.
*H	7721133	03/08/2022	Fixed some clerical errors. For details, please see the chapter 55 of Appendix: Major Changes.