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.

S6J3120 Series 32-bit Microcontroller

Traveo™ Family Hardware Manual

Doc. No. 002-04855 Rev. *E

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709
www.cypress.com

Copyrights

© Cypress Semiconductor Corporation, 2014-2018. This document is the property of Cypress Semiconductor Corporation and its subsidiaries, including Spansion LLC (“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 does not assume any liability arising out of any security breach, such as unauthorized access to or use of a Cypress product. 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. Cypress products are not designed, intended, or authorized for use as critical components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or system could cause personal injury, death, or property damage (“Unintended Uses”). A critical component is any component of a device or system whose failure to perform can be reasonably expected to cause the failure of the device or system, 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 or related to all Unintended Uses of Cypress products. You shall indemnify and hold Cypress harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of Cypress products.

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.

Sample programs and development environment

Cypress offers sample programs free of charge for using the peripheral functions. Cypress also makes available descriptions of the development environment required for this series. Feel free to use them to verify the operational specifications and usage of this Cypress microcontroller.

Microcontroller support information:

Note that the sample programs are subject to change without notice. Since they are offered as a way to demonstrate standard operations and usage, evaluate them sufficiently before running them on your system.
Cypress assumes no responsibility for any damage that may occur as a result of using a sample program.

How to Use This Manual

Finding a function

The following methods can be used to search for the explanation of a desired function in this manual:

- Search from the table of the contents

  The table of the contents lists the manual contents in the order of description.

- Search from the register

  The address where each register is located is not described in the text. To verify the address of a register, see "A. I/O Map" in "APPENDIX".

Terminology

This manual uses the following terminology.

Term	Explanation
Double Word	Indicates access in units of 64 bits.
Word	Indicates access in units of 32 bits.
Half word	Indicates access in units of 16 bits.
Byte	Indicates access in units of 8 bits.

Notations

- The notations in bit configuration of the register explanation of this manual are written as follows.
  - bit : bit number
  - Field : bit field name
  - Read/Write Attribute (R/W Attribute) : Attributes for read and write of each bit
    - 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)

(Note)

- *In case factors that influences read value are following, R/W attribute is described with / (slash mark).*

  1. Bit set by bit set register operation

  2. Bit clear by bit clear register operation

  - Initial value : Initial value of the register after reset
    - 0 : Initial value is "0"
    - 1 : Initial value is "1"
    - X : Initial value is undefined

- Protection Attribute Format
  - Attribute :
    - 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

- The multiple bits are written as follows in this manual.
  Example : bit7:0 indicates the bits from bit7 to bit0

- The values such as for addresses are written as follows in this manual.
  - Hexadecimal number: "0x" is attached in the beginning of a value as a prefix (example: 0xFFFF)
  - Binary number: "0b" is attached in the beginning of a value as a prefix (example: 0b1111)
  - Decimal number: Written using numbers only (example: 1000)

Contents

Introduction: Restrictions on Operation Specifications 45
1. About Reset 46
2. About Clock System Specifications 47
3. About Low-Power Consumption Specifications 48
4. About Low-Voltage Detection Specifications 48
5. About the Clock Supervisor 49
6. About External Interrupts 49
7. About the Real Time Clock 49
8. About the BootROM Hardware Interface 49
9. About the Multi-Function Serial Interface 50
10. About the Base Timer 51
11. About the Base Timer I/O Selection Function 51
12. About the 32-Bit Free-Run Timer 51
13. About the 32-Bit Input Capture 52
14. About the 32-Bit Output Compare 52
15. About the External Interrupt Capture 52
16. About TCFLASH 53
17. About the Hardware Watchdog Timer 53
18. About CAN FD Message RAM ECC Function 53
19. About CAN-FD 53
20. About the 32-Bit Reload Timer 54
21. About the Reload Timer Simultaneous Soft Start 54
22. About the Stepper Motor Controller 54
23. About the Trigger Configuration of Stepper Motor Controller 55
24. About the External Bus Interface 55
25. About the Sound Generator 55
26. About the Quad Position & Revolution Counter 55
27. About the DDR High Speed SPI Controller 56
28. About the LCD Controller 56
29. About DMA Complex Subsystem 56
30. About the 12-Bit A/D Converter 56
31. About the Secure Hardware Extension (SHE) 56
32. About the Partial Wakeup Control 56
33. Common Items 57
    33.1. How to Use Reset Factor Register 57
    33.2. Caution in Setting and Clearing Bit 57

CHAPTER 1: Overview 59
1. Overview 60
2. Features 61
    2.1. Cortex-R5 Core 62

2.2. Peripheral Functions 63
3. Product Lineup 67
4. Function Overview 69
5. Block Diagrams 72
6. Memory Map 73
7. Abbreviations 81
8. Pin Assignment 84
9. Package Dimensions 85
10. List of Pin Functions 86
11. I/O Circuit Type 103
CHAPTER 2: CPU 107
1. Overview 108
2. Notes 109
CHAPTER 3: Operation Mode 115
1. Overview 116
2. Configuration 117
3. Explanation of Operation 118
4. Register 119
4.1. Mode Register (MODEC_MODER) 120
CHAPTER 4: Reset 123
1. Overview 124
2. Configuration 125
3. Operational Description 128
3.1. Reset Factor 129
3.1.1. Power-on Reset 130
3.1.2. RAM Retention Low-Voltage Detection RAM Reset 131
3.1.3. RSTX Pin + MD Pin Simultaneous assert Reset 132
3.1.4. Illegal Mode Detection Reset 133
3.1.5. Internal Power Supply Low-Voltage Detection Reset 134
3.1.6. External Power Supply Low-Voltage Detection Reset 135
3.1.7. Extended Internal Power Supply Low-Voltage Reset 136
3.1.8. Extended External Power Supply Low-Voltage Reset 137
3.1.9. Clock Stop Wait Timeout Reset 138
3.1.10. RSTX Pin Input Reset 139
3.1.11. Hardware Watchdog Reset 140
3.1.12. Software Watchdog Reset 141
3.1.13. Main Clock Supervisor Reset 142
3.1.14. Sub Clock Supervisor Reset 143
3.1.15. PLL Clock Supervisor Reset 144
3.1.16. SSCG Clock Supervisor Reset 145
3.1.17. Profile Error Reset 146
3.1.18. Software Trigger Hard Reset 147
3.1.19. Software Reset 148
3.1.20. TRSTX Pin Input Reset 149
3.1.21. Software Debugger Reset 150
3.1.22. PowerDomain Reset 2 151
3.1.23. PowerDomain Reset 3 152
3.1.24. PowerDomain Reset 4 153
3.1.25. PowerDomain Reset 5 154

3.1.26. PowerDomain Reset 6 155
3.2. Internal Reset of Device 156
3.2.1. CPU Reset 157
3.2.2. I/O Reset 158
3.2.3. Reset of 1.2V External Power Supply Control 159
3.3. Reset Sequence 160
3.4. Operations after Reset Release 162
3.4.1. Hard Reset Operation 164
3.4.2. Soft Reset Operation 169
4. Registers 170
4.1. Reset Control Register (SYSC_RSTCNTR) 171
4.2. User Reset Factor Register (SYSC_RSTCAUSEUR) 173
4.3. User Extended CSV Reset Factor Register (SYSC_EXCSVRSTCAUSEUR) 179
4.4. User PowerDomain Reset Factor Register (SYSC_PDRSTCAUSEUR) 182
4.5. BootROM Reset Factor Register (SYSC_RSTCAUSEBT) 185
4.6. BootROM Extended CSV Reset Factor Register (SYSC_EXCSVRSTCAUSEBT) 191
4.7. BootROM PowerDomain Reset Factor Register (SYSC_PDRSTCAUSEBT) 193
4.8. PowerDomain Reset Status Register (SYSC_PDRSTSTATUS) 197
CHAPTER 5: Clock System 201
1. Overview 202
2. Configuration and Block Diagram 203
2.1. Clock System Configuration and Block Diagram 204
2.2. Clock Generator Configuration and Block Diagram 206
2.2.1. PLL Clock Generator Configuration and Block Diagram 207
2.2.2. SSCG PLL Clock Generator Configuration and Block Diagram 208
2.3. Clock Distributer Configuration and Block Diagram 209
3. Operations 212
3.1. Source Clock Generation 213
3.2. PLL/SSCG PLL Clock 214
3.3. Clock Distributer 217
3.4. Clock Gear 221
3.4.1. Gear Up Procedure 222
3.4.2. Gear Down Procedure 223
3.5. Clock Stabilization Time 224
3.6. Interrupt 225
4. Setting Procedure Example 226
4.1. RUN Configuration 227
4.2. PSS Configuration 229
5. Registers 230
5.1. Common Configuration Registers 231
5.1.1. CR Clock Control Register(SYSC_CRCNTR) 231
5.1.2. Main Oscillator Control Register(SYSC_MOSCCNTR) 232
5.1.3. PLL/SSCG Stabilization Time Control Register (SYSC_PLLSSCGSTCNTR) 234
5.1.4. PLL0 Clock Gear Control Register (SYSC_PLL0CGCNTR) 236
5.1.5. SSCG PLL0 Clock Gear Control Register (SYSC_SSCG0CGCNTR) 239
5.2. Clock Output Function Register 242
5.2.1. Clock Output Function Control Register (SYSC_CKOTCNTR) 242
6. Usage Precautions 244

CHAPTER 6: Low Power Consumption 245
1. Overview 246
2. Configuration 247
2.1. Block Diagram 248
3. Explanation of Operation 249
3.1. Low Power Consumption State 250
3.2. Power Supply Domain 252
3.3. Profile Setting Items 255
3.4. Profile 261
3.5. Interrupts 271
3.6. Bus Error Response 272
4. Operation Procedure 273
4.1. RUN Profile Update 274
4.2. Transition from RUN to PSS 276
4.3. Transition from PSS to RUN 278
5. Registers 281
5.1. Protection Register Group (SYSC0) 287
5.1.1. Protection Key Setting Register (SYSC0_PROTKEYR) 288
5.2. RUN Profile Register Group (SYSC0) 289
5.2.1. RUN Power Domain Setting Register (SYSC0_RUNPDCFGR) 290
5.2.2. RUN Clock Source Enable Register (SYSC0_RUNCKSRER) 293
5.2.3. RUN Clock Selection Register (SYSC0_RUNCKSELR) 296
5.2.4. RUN Clock Enable Register (SYSC0_RUNCKER) 297
5.2.5. RUN Clock Divider Register (SYSC0_RUNCKDIVR) 298
5.2.6. RUN PLLx Control Register (SYSC0_RUNPLLxCNTR) 300
5.2.7. RUN SSCGx Control Register 0 (SYSC0_RUNSSCGxCNTR0) 303
5.2.8. RUN SSCGx Control Register 1 (SYSC0_RUNSSCGxCNTR1) 306
5.2.9. RUN Low-Voltage Detection Setting Register (SYSC0_RUNLVDCFGR) 309
5.2.10. RUN Clock Supervisor Setting Register (SYSC0_RUNCSVCFGR) 314
5.2.11. RUN Regulator Setting Register (SYSC0_RUNREGCFGR) 317
5.2.12. RUN Profile Update Trigger Register (SYSC0_TRGRUNCNTR) 318
5.3. PSS Profile Register Group (SYSC0) 319
5.3.1. PSS Power Domain Setting Register (SYSC0_PSSPDCFGR) 320
5.3.2. PSS Clock Source Enable Register (SYSC0_PSSCKSRER) 323
5.3.3. PSS Clock Selection Register (SYSC0_PSSCKSELR) 326
5.3.4. PSS Clock Enable Register (SYSC0_PSSCKER) 327
5.3.5. PSS Clock Divider Register (SYSC0_PSSCKDIVR) 328
5.3.6. PSS PLLx Control Register (SYSC0_PSSPLLxCNTR) 330
5.3.7. PSS SSCGx Control Register 0 (SYSC0_PSSSSCGxCNTR0) 333
5.3.8. PSS SSCGx Control Register 1 (SYSC0_PSSSSCGxCNTR1) 336
5.3.9. PSS Low-Voltage Detection Setting Register (SYSC0_PSSLVDCFGR) 339
5.3.10. PSS Clock Supervisor Setting Register (SYSC0_PSSCSVCFGR) 344
5.3.11. PSS Regulator Setting Register (SYSC0_PSSREGCFGR) 347
5.3.12. PSS Profile Update Enable Register (SYSC0_PSSENR) 348
5.4. APP Profile Register Group (SYSC0) 349
5.4.1. APP Power Domain Setting Register (SYSC0_APPPDCFGR) 350
5.4.2. APP Clock Source Enable Register (SYSC0_APPCKSRER) 353
5.4.3. APP Clock Selection Register (SYSC0_APPCKSELR) 357
5.4.4. APP Clock Enable Register (SYSC0_APPCKER) 358

5.4.5. APP Clock Divider Register (SYSC0_APPCKDIVR) 359
5.4.6. APP PLLx Control Register (SYSC0_APPPLLxCNTR) 361
5.4.7. APP SSCGx Control Register 0 (SYSC0_APPSSCGxCNTR0) 364
5.4.8. APP SSCGx Control Register 1 (SYSC0_APPSSCGxCNTR1) 367
5.4.9. APP Low-Voltage Detection Setting Register (SYSC0_APPLVDCFGR) 370
5.4.10. APP Clock Supervisor Setting Register (SYSC0_APPCSVCFGR) 375
5.4.11. APP Regulator Setting Register (SYSC0_APPREGCFGR) 378
5.5. STS Profile Register Group (SYSC0) 379
5.5.1. STS Power Domain Setting Register (SYSC0_STSPDCFGR) 380
5.5.2. STS Clock Source Enable Register (SYSC0_STSCKSRER) 383
5.5.3. STS Clock Selection Register (SYSC0_STSCKSELR) 389
5.5.4. STS Clock Enable Register (SYSC0_STSCKER) 390
5.5.5. STS Clock Division Register (SYSC0_STSCKDIVR) 391
5.5.6. STS PLLx Control Register (SYSC0_STSPLLxCNTR) 393
5.5.7. STS SSCGx Control Register 0 (SYSC0_STSSSCGxCNTR0) 396
5.5.8. STS SSCGx Control Register 1 (SYSC0_STSSSCGxCNTR1) 399
5.5.9. STS Low-Voltage Detection Setting Register (SYSC0_STSLVDCFGR) 402
5.5.10. STS Clock Supervisor Setting Register (SYSC0_STSCSVCFGR) 407
5.5.11. STS Regulator Setting Register (SYSC0_STSREGCFGR) 410
5.6. System Register Group (SYSC0) 411
5.6.1. System Status Register (SYSC0_SYSSTSR) 412
5.6.2. System Status Interrupt Enable Register (SYSC0_SYSINTER) 414
5.6.3. System Status Flag and Interrupt Clear Register (SYSC0_SYSICLR) 415
5.6.4. System Error Interrupt Factor Register 0 (SYSC0_SYSERRIR0) 416
5.6.5. System Error Interrupt Factor Register 1 (SYSC0_SYSERRIR1) 420
5.6.6. System Error Interrupt Factor Clear Register 0 (SYSC0_SYSERRICLR0) 422
5.6.7. System Error Interrupt Factor Clear Register 1 (SYSC0_SYSERRICLR1) 426
5.6.8. Profile Status Register (SYSC0_SYSPROSTSR) 428
5.6.9. RUN Profile Error Flag Register (SYSC0_SYSRUNPEFR) 430
5.6.10. PSS Profile Error Flag Register (SYSC0_SYSPSSPEFR) 431
5.6.11. System ID Register (SYSC0_SYSIDR) 432
5.6.12 Platform ID Register (SYSC0_SYSPFIDR) 433
5.7. Special Setting Register Group (SYSC0) 434
5.7.1. System Special Setting Register (SYSC0_SPECFGR) 435
5.8. Debug Register Group (SYSC0) 438
5.8.1. JTAG Detection Register (SYSC0_JTAGDETECT) 439
5.8.2. JTAG Setting Register (SYSC0_JTAGCNFG) 440
5.8.3. JTAG Recovery Register (SYSC0_JTAGWAKEUP) 441
5.9. MCU_Config_Group Interrupt Batch Read Register Group (MCG_IRS) 442
5.9.1. MCU_Config_Group Interrupt Request Status Register0 (MCG_IRSR0) 443
5.9.2. MCU_Config_Group Interrupt Request Status Register1 (MCG_IRSR1) 446
5.9.3. MCU_Config_Group Interrupt Request Status Register2 (MCG_IRSR2) 447
5.9.4. MCU_Config_Group Interrupt Request Status Register3 (MCG_IRSR3) 450
5.9.5. MCU_Config_Group Interrupt Request Status Register4 (MCG_IRSR4) 455
5.10. Protection Register Group (SYSC1) 457
5.10.1. Protection Key Setting Register (SYSC1_PROTKEYR) 458
5.11. RUN Profile Register Group (SYSC1) 459
5.11.1. RUN Clock Selection Register 0 (SYSC1_RUNCKSELR0) 460
5.11.2. RUN Clock Selection Register 1 (SYSC1_RUNCKSELR1) 463

5.11.3. RUN Clock Selection Register 2 (SYSC1_RUNCKSELR2) 466
5.11.4. RUN Clock Source Enable Register 0 (SYSC1_RUNCKER0) 468
5.11.5. RUN Clock Source Enable Register 1 (SYSC1_RUNCKER1) 474
5.11.6. RUN Clock Source Enable Register 2 (SYSC1_RUNCKER2) 478
5.11.7. RUN Clock Divider Register 0 (SYSC1_RUNCKDIVR0) 481
5.11.8. RUN Clock Divider Register 1 (SYSC1_RUNCKDIVR1) 484
5.11.9. RUN Clock Divider Register 2 (SYSC1_RUNCKDIVR2) 487
5.11.10. RUN Clock Divider Register 3 (SYSC1_RUNCKDIVR3) 490
5.11.11. RUN Clock Divider Register 4 (SYSC1_RUNCKDIVR4) 492
5.11.12. RUN Clock Divider Register 5 (SYSC1_RUNCKDIVR5) 493
5.11.13. RUN Clock Divider Register 6 (SYSC1_RUNCKDIVR6) 496
5.11.14. RUN Clock Divider Register 7 (SYSC1_RUNCKDIVR7) 499
5.11.15. RUN Clock Divider Register 8 (SYSC1_RUNCKDIVR8) 502
5.11.16. RUN Clock Divider Register 9 (SYSC1_RUNCKDIVR9) 505
5.11.17. RUN Profile Update Enable Register (SYSC1_RUNENR) 508
5.12. PSS Profile Register Group (SYSC1) 509
5.12.1. PSS Clock Selection Register 0 (SYSC1_PSSCKSELR0) 510
5.12.2. PSS Clock Selection Register 1 (SYSC1_PSSCKSELR1) 513
5.12.3. PSS Clock Selection Register 2 (SYSC1_PSSCKSELR2) 516
5.12.4. PSS Clock Source Enable Register 0 (SYSC1_PSSCKER0) 518
5.12.5. PSS Clock Source Enable Register 1 (SYSC1_PSSCKER1) 523
5.12.6. PSS Clock Source Enable Register 2(SYSC1_PSSCKER2) 527
5.12.7. PSS Clock Divider Register 0 (SYSC1_PSSCKDIVR0) 530
5.12.8. PSS Clock Divider Register 1 (SYSC1_PSSCKDIVR1) 533
5.12.9. PSS Clock Divider Register 2 (SYSC1_PSSCKDIVR2) 536
5.12.10. PSS Clock Divider Register 3 (SYSC1_PSSCKDIVR3) 539
5.12.11. PSS Clock Divider Register 4 (SYSC1_PSSCKDIVR4) 542
5.12.12. PSS Clock Divider Register 5 (SYSC1_PSSCKDIVR5) 543
5.12.13. PSS Clock Divider Register 6 (SYSC1_PSSCKDIVR6) 546
5.12.14. PSS Clock Divider Register 7 (SYSC1_PSSCKDIVR7) 549
5.12.15. PSS Clock Divider Register 8 (SYSC1_PSSCKDIVR8) 552
5.12.16. PSS Clock Divider Register 9 (SYSC1_PSSCKDIVR9) 555
5.12.17. PSS Profile Update Enable Register (SYSC1_PSSENR) 558
5.13. APP Profile Register Group (SYSC1) 559
5.13.1. APP Clock Selection Register 0 (SYSC1_APPCKSELR0) 560
5.13.2. APP Clock Selection Register 1 (SYSC1_APPCKSELR1) 563
5.13.3. APP Clock Selection Register 2 (SYSC1_APPCKSELR2) 566
5.13.4. APP Clock Source Enable Register0 (SYSC1_APPCKER0) 568
5.13.5. APP Clock Source Enable Register1 (SYSC1_APPCKER1) 573
5.13.6. APP Clock Source Enable Register2 (SYSC1_APPCKER2) 577
5.13.7. APP Clock Divider Register 0 (SYSC1_APPCKDIVR0) 580
5.13.8. APP Clock Divider Register 1 (SYSC1_APPCKDIVR1) 583
5.13.9. APP Clock Divider Register 2 (SYSC1_APPCKDIVR2) 586
5.13.10. APP Clock Divider Register 3 (SYSC1_APPCKDIVR3) 589
5.13.11. APP Clock Divider Register 4 (SYSC1_APPCKDIVR4) 592
5.13.12. APP Clock Divider Register 5 (SYSC1_APPCKDIVR5) 593
5.13.13. APP Clock Divider Register 6 (SYSC1_APPCKDIVR6) 596
5.13.14. APP Clock Divider Register 7 (SYSC1_APPCKDIVR7) 599
5.13.15. APP Clock Divider Register 8 (SYSC1_APPCKDIVR8) 602

5.13.16. APP Clock Divider Register 9 (SYSC1_APPCKDIVR9) 605
5.14. STS Profile Register Group (SYSC1) 608
5.14.1. STS Clock Selection Register 0 (SYSC1_STSCKSELR0) 609
5.14.2. STS Clock Selection Register 1 (SYSC1_STSCKSELR1) 614
5.14.3. STS Clock Selection Register 2 (SYSC1_STSCKSELR2) 619
5.14.4. STS Clock Source Enable Register 0(SYSC1_STSCKER0) 622
5.14.5. STS Clock Source Enable Register 1(SYSC1_STSCKER1) 627
5.14.6. STS Clock Source Enable Register 2(SYSC1_STSCKER2) 631
5.14.7. STS Clock Divider Register 0 (SYSC1_STSCKDIVR0) 634
5.14.8. STS Clock Divider Register 1 (SYSC1_STSCKDIVR1) 637
5.14.9. STS Clock Divider Register 2 (SYSC1_STSCKDIVR2) 640
5.14.10. STS Clock Divider Register 3 (SYSC1_STSCKDIVR3) 643
5.14.11. STS Clock Divider Register 4 (SYSC1_STSCKDIVR4) 646
5.14.12. STS Clock Divider Register 5 (SYSC1_STSCKDIVR5) 647
5.14.13. STS Clock Divider Register 6 (SYSC1_STSCKDIVR6) 650
5.14.14. STS Clock Divider Register 7 (SYSC1_STSCKDIVR7) 653
5.14.15. STS Clock Divider Register 8 (SYSC1_STSCKDIVR8) 656
5.14.16. STS Clock Divider Register 9 (SYSC1_STSCKDIVR9) 659
6. Other 662
6.1. Restrictions on Transitioning to the PSS 663
6.2. Limitation when Changing the Setting of Regulator 664
6.3. Restriction to use “Clock fixed at L” and changing clock division ratio 665
CHAPTER 7: Low-Voltage Detection 667
1. Overview 668
1.1. Features 668
2. Configuration and Block Diagrams 671
3. Explanation of Operation 673
4. Setting Procedure Examples 675
5. Operation Examples 677
6. Precautions for Using This Device 680
CHAPTER 8: Clock Supervisor 681
1. Overview 682
2. Configuration 683
3. Explanation of Operation 687
4. Setting Procedure Example 689
5. Registers 691
5.1. Main Clock Supervisor Setting Register 0 (SYSC_CSVMOCFGR0) 692
5.2. Main Clock Supervisor Setting Register 1 (SYSC_CSVMOCFGR1) 694
5.3. Sub Clock Supervisor Setting Register 0 (SYSC_CSVSOCFGR0) 696
5.4. Sub Clock Supervisor Setting Register 1 (SYSC_CSVSOCFGR1) 698
5.5. PLL0 Clock Supervisor Setting Register 0 (SYSC_CSVPLL0CFGR0) 700
5.6. PLL0 Clock Supervisor Setting Register 1 (SYSC_CSVPLL0CFGR1) 702
5.7. PLL1 Clock Supervisor Setting Register 0 (SYSC_CSVPLL1CFGR0) 704
5.8. PLL1 Clock Supervisor Setting Register 1 (SYSC_CSVPLL1CFGR1) 706
5.9. PLL2 Clock Supervisor Setting Register 0 (SYSC_CSVPLL2CFGR0) 708
5.10. PLL2 Clock Supervisor Setting Register 1 (SYSC_CSVPLL2CFGR1) 710
5.11. PLL3 Clock Supervisor Setting Register 0 (SYSC_CSVPLL3CFGR0) 712
5.12. PLL3 Clock Supervisor Setting Register 1 (SYSC_CSVPLL3CFGR1) 714
5.13. SSCG PLL0 Clock Supervisor Setting Register 0 (SYSC_CSVSP0CFGR0) 716

5.14. SSCG PLL0 Clock Supervisor Setting Register 1 (SYSC_CSVSP0CFGR1) 718
5.15. SSCG PLL1 Clock Supervisor Setting Register 0 (SYSC_CSVSP1CFGR0) 720
5.16. SSCG PLL1 Clock Supervisor Setting Register 1 (SYSC_CSVSP1CFGR1) 722
5.17. SSCG PLL2 Clock Supervisor Setting Register 0 (SYSC_CSVSP2CFGR0) 724
5.18. SSCG PLL2 Clock Supervisor Setting Register 1 (SYSC_CSVSP2CFGR1) 726
5.19. SSCG PLL3 Clock Supervisor Setting Register 0 (SYSC_CSVSP3CFGR0) 728
5.20. SSCG PLL3 Clock Supervisor Setting Register 1 (SYSC_CSVSP3CFGR1) 730
5.21. Clock Supervisor Output Enable Register (SYSC_CSVOUTER) 732
5.22. Clock Supervisor Test Register (SYSC_CSVTESTR) 733
6. Precautions for Using This Device 736

CHAPTER 9: Source Clock Timer 737

1. Overview 738
2. Configuration 739
3. Explanation of Operation 740
4. Setting Procedure Example 741
5. Registers 742
5.1. High-Speed CR Clock Timer Trigger Register (SYSC_FCRCTTRGR) 743
5.2. High-Speed CR Clock Timer Control Register (SYSC_FCRCTCNTR) 745
5.3. High-Speed CR Clock Timer Compare Prescaler Register (SYSC_FCRCTCPR) 747
5.4. High-Speed CR Clock Timer Status Register (SYSC_FCRCTSTR) 749
5.5. High-Speed CR Clock Timer Interrupt Enable Register (SYSC_FCRCTINTER) 751
5.6. High-Speed CR Clock Timer Interrupt Clear Register (SYSC_FCRCTICLR) 752
5.7. Low-Speed CR Clock Timer Trigger Register (SYSC_SCRCTTRGR) 753
5.8. Low-Speed CR Clock Timer Control Register (SYSC_SCRCTCNTR) 755
5.9. Low-Speed CR Clock Timer Compare Prescaler Register (SYSC_SCRCTCPR) 757
5.10. Low-Speed CR Clock Timer Status Register (SYSC_SCRCTSTR) 759
5.11. Low-Speed CR Clock Timer Interrupt Enable Register (SYSC_SCRCTINTER) 761
5.12. Low-Speed CR Clock Timer Interrupt Clear Register (SYSC_SCRCTICLR) 762
5.13. Main Clock Timer Trigger Register (SYSC_MOCTTRGR) 763
5.14. Main Clock Timer Control Register (SYSC_MOCTCNTR) 765
5.15. Main Clock Timer Compare Prescaler Register (SYSC_MOCTCPR) 767
5.16. Main Clock Timer Status Register (SYSC_MOCTSTR) 769
5.17. Main Clock Timer Interrupt Enable Register (SYSC_MOCTINTER) 771
5.18. Main Clock Timer Interrupt Clear Register (SYSC_MOCTICLR) 772
6. Precautions for Using This Device 773

CHAPTER 10: Real Time Clock 775

1. Overview 776
2. Configuration and Block Diagram 777
3. Operation of the Real Time Clock 780
4. Registers 786
4.1. Timer Control Register (RTC_WTCR) 787
4.2. Timer Status Register (RTC_WTSR) 792
4.3. Interrupt Status Register (RTC_WINS) 794
4.4. Interrupt Enable Register (RTC_WINE) 797
4.5. Interrupt Clear Register (RTC_WINC) 799
4.6. Sub-Second Register (RTC_WTBR) 802
4.7. Real Time Register (RTC_WRT) 804
4.8. Real Time Clock Real Time Register1 (RTC_RTR1) 806
4.9. Calibration Clock Counter Register (RTC_CNTCAL) 807

4.10. Calibration Clock Period Counter Register (RTC_CNTPCAL) 809
4.11. Calibration Duration Register (RTC_DURMW) 811
4.12. Calibration Trigger Register (RTC_CALTRG) 813
4.13. Debug Register (RTC_DEBUG) 814
4.14. Partial Wake Up Trigger Control Register (RTC_PWUTRGCR) 815
4.15. Partial Wake Up Trigger Status Register (RTC_PWUTRGSR) 817
5. Usage Precaution 818
CHAPTER 11: CR Calibration 821
1. Overview 822
2. Configuration 823
3. Explanation of Operation 824
3.1. CR Clock Frequency Calculation 825
3.2. CR Clock Frequency Correction 826
4. Registers 827
4.1. Correction Unit Control Register 1 (CU_CUCR1) (Calibration Unit Control Register 1) ... 828
4.2. CR Clock Timer Data Register 1 (CU_CUTD1) (Calibration Unit Timer Data Register 1) 830
4.3. Main Oscillation Timer Data Register 1 (CU_CUTR1) (Calibration Unit Timer Result Register 1) 831
4.4. Correction Unit Control Clear Register 1 (CU_CUCRC1) (Calibration Unit Control Register Clear Register 1) 832
CHAPTER 12: Backup RAM Interface 833
1. Overview 834
2. Configuration 835
3. Explanation of Operation 838
3.1. RAMECC Function 839
3.2. RAM Diagnosis 843
3.3. RAM Initialization 845
3.4. Required Number of Cycles 846
3.5. Bus Error Response 847
4. Setting Procedure 848
5. Registers 851
5.1. BURAMIF Unlock Register (BURIF_UNLOCK) 852
5.2. BURAMIF Status Register (BURIF_STATUS) 853
5.3. BURAMIF Double-Bit ECC Error Address Register (BURIF_DEEAR) 854
5.4. BURAMIF Single-Bit ECC Error Address Register (BURIF_SEEAR) 855
5.5. BURAMIF ECC Pseudo-Error Generation Address Register (BURIF_EFEAR) 856
5.6. BURAMIF ECC Error Control Register (BURIF_EECSR) 857
5.7. BURAMIF ECC Pseudo-Error Generation Control Register (BURIF_EFECR) 858
5.8. BURAMIF TEST Error Address Register 0 (BURIF_TEAR0) 860
5.9. BURAMIF TEST Error Address Register 1 (BURIF_TEAR1) 862
5.10. BURAMIF TEST Error Address Register 2 (BURIF_TEAR2) 864
5.11. BURAMIF TEST Start Address Register (BURIF_TASAR) 866
5.12. BURAMIF TEST End Address Register (BURIF_TAEAR) 867
5.13. BURAMIF TEST Diagnosis Function Register (BURIF_TTCR) 868
5.14. BURAMIF TEST Initialization Function Register (BURIF_TICR) 870
5.15. BURAMIF TEST Software Reset Generation Control Register (BURIF_TSRCR) 871
5.16. BURAMIF TEST Pseudo-Error Generation Control Register (BURIF_TFECR) 872
5.17. BURAMIF TEST Key Code Control Register (BURIF_TKCCR) 873
5.18. BURAMIF Event Clear Register (BURIF_EVENTCLR) 874

6. Precautions for Using Device 876
CHAPTER 13: External Interrupt 877
1. Overview 878
2. Configuration 879
2.1. Block Diagrams 879
3. Explanation of Operation 880
4. Setting Procedure Example 882
5. Registers 883
5.1. External Interrupt Enable Register (EICxx_ENIR) 884
5.2. External Interrupt Enable Set Register (EICxx_ENISR) 885
5.3. External Interrupt Enable Clear Register (EICxx_ENICR) 886
5.4. External Interrupt Factor Register (EICxx_EIRR) 887
5.5. External Interrupt Factor Clear Register (EICxx_EIRCR) 888
5.6. Noise Filter Enable Register (EICxx_NFER) 889
5.7. Noise Filter Enable Set Register (EICxx_NFESR) 890
5.8. Noise Filter Enable Clear Register (EICxx_NFECR) 891
5.9. External Interrupt Level Registers (EICxx_ELVR0 to 3) 892
5.10. Non-Maskable Interrupt Register (EICxx_NMIR) 894
5.11. DMA Request Enable Register (EICxx_DRER) 895
5.12. DMA Request Enable Set Register (EICxx_DRESR) 896
5.13. DMA Request Enable Clear Register (EICxx_DRECR) 897
5.14. DMA Request Flag Register (EICxx_DRFR) 898
CHAPTER 14: Hardware Watchdog Timer 901
1. Overview 902
2. Configuration and Block Diagram 903
3. Explanation of Operation 904
4. Setting Procedure Example 920
5. Operation Examples 922
6. Register List 926
6.1. Hardware Watchdog Protection Register (HWDG_PROT) 927
6.2. Hardware Watchdog Counter Register (HWDG_CNT) 928
6.3. Hardware Watchdog Reset Factor Register (HWDG_RSTCAUSE) 929
6.4. Hardware Watchdog Trigger 0 Register (HWDG_TRG0) 931
6.5. Hardware Watchdog Trigger 1 Register (HWDG_TRG1) 932
6.6. Hardware Watchdog Interrupt Configuration Register (HWDG_INT) 933
6.7. Hardware Watchdog Interrupt Clear Register (HWDG_INTCLR) 935
6.8. Hardware Watchdog Trigger 0 Configuration Register (HWDG_TRG0CFG) 936
6.9. Hardware Watchdog Trigger 1 Configuration Register (HWDG_TRG1CFG) 937
6.10. Hardware Watchdog Lower Limit RUN Setting Register (HWDG_RUNLLS) 938
6.11. Hardware Watchdog Upper Limit RUN Setting Register (HWDG_RUNULS) 939
6.12. Hardware Watchdog Lower Limit PSS Setting Register (HWDG_PSSLLS) 940
6.13. Hardware Watchdog Upper Limit PSS Setting Register (HWDG_PSSULS) 941
6.14. Hardware Watchdog Reset Delay Counter Register (HWDG_RSTDLY) 942
6.15. Hardware Watchdog Configuration Register (HWDG_CFG) 943
6.16. Hardware Watchdog Lower Limit RUN Current Register (HWDG_RUNLLC) 946
6.17. Hardware Watchdog Upper Limit RUN Current Register (HWDG_RUNULC) 947
6.18. Hardware Watchdog Lower Limit PSS Current Register (HWDG_PSSLLC) 948
6.19. Hardware Watchdog Upper Limit PSS Current Register (HWDG_PSSULC) 949
7. Precautions for Using This Device 950

CHAPTER 15: Software Watchdog Timer 953
1. Overview 954
2. Configuration and Block Diagram 955
3. Explanation of Operation 956
4. Setting Procedure Example 973
5. Operation Examples 974
6. Register List 977
6.1. Software Watchdog Protection Register (SWDG_PROT) 978
6.2. Software Watchdog Counter Register (SWDG_CNT) 979
6.3. Software Watchdog Reset Factor Register (SWDG_RSTCAUSE) 980
6.4. Software Watchdog Trigger 0 Register (SWDG_TRG0) 982
6.5. Software Watchdog Trigger 1 Register (SWDG_TRG1) 983
6.6. Software Watchdog Interrupt Configuration Register (SWDG_INT) 984
6.7. Software Watchdog Interrupt Clear Register (SWDG_INTCLR) 986
6.8. Software Watchdog Trigger 0 Configuration Register (SWDG_TRG0CFG) 987
6.9. Software Watchdog Trigger 1 Configuration Register (SWDG_TRG1CFG) 988
6.10. Software Watchdog Lower Limit RUN Setting Register (SWDG_RUNLLS) 989
6.11. Software Watchdog Upper Limit RUN Setting Register (SWDG_RUNULS) 990
6.12. Software Watchdog Lower Limit PSS Setting Register (SWDG_PSSLLS) 991
6.13. Software Watchdog Upper Limit PSS Setting Register (SWDG_PSSULS) 992
6.14. Software Watchdog Reset Delay Counter Register (SWDG_RSTDLY) 993
6.15. Software Watchdog Configuration Register (SWDG_CFG) 994
6.16. Software Watchdog Lower Limit RUN Current Register (SWDG_RUNLLC) 997
6.17. Software Watchdog Upper Limit RUN Current Register (SWDG_RUNULC) 998
6.18. Software Watchdog Lower Limit PSS Current Register (SWDG_PSSLLC) 999
6.19. Software Watchdog Upper Limit PSS Current Register (SWDG_PSSULC) 1000
7. Precautions for Using This Device 1001
CHAPTER 16: TCRAM Interface 1003
1. Overview 1004
2. Configuration 1005
3. Explanation of Operation 1007
4. Setting Procedure Examples 1011
5. Registers 1013
5.1. TCRAM IF Configuration Register 0 (TRCFGn_TCMCFG0) 1014
5.2. TCRAM IF Configuration Register 1 (TRCFGn_TCMCFG1) 1016
5.3. TCRAM IF Unlock Register (TRCFGn_TCMUNLOCK) 1017
5.4. TCRAM IF TEST Error Address Register 0 (TRCFGn_TEAR0) 1018
5.5. TCRAM IF TEST Error Address Register 1 (TRCFGn_TEAR1) 1020
5.6. TCRAM IF TEST Error Address Register 2 (TRCFGn_TEAR2) 1022
5.7. TCRAM IF TEST End Address Register (TRCFGn_TAEAR) 1024
5.8. TCRAM IF TEST Start Address Register (TRCFGn_TASAR) 1025
5.9. TCRAM IF TEST Pseudo Error Generation Control Register (TRCFGn_TFECR) 1026
5.10. TCRAM IF TEST Initialization Function Register (TRCFGn_TICR) 1027
5.11. TCRAM IF TEST Diagnosis Function Register (TRCFGn_TTCR) 1028
5.12. TCRAM IF TEST Soft Reset Generation Control Register (TRCFGn_TSRCR) 1031
5.13. TCRAM IF TEST Key Code Control Register (TRCFGn_TKCCR) 1032
5.14. TCRAM IF TEST Status Clear Register (TRCFGn_TSCR) 1033
CHAPTER 17: TCFLASH 1035
1. Overview 1036

2. Configuration....1037
2.1. Block Diagrams....1037
2.2. Address/Sector Map of TCFLASH....1039
3. Explanation of Operation....1043
3.1. Operation Mode of TCFLASH....1043
3.2. Programming and Erasing....1044
3.3. TCM Buffer....1045
3.4. Arbitration of TCM and AXI....1046
3.5. Automatic Algorithms....1047
3.5.1. Command Sequence....1048
3.6. Automatic Algorithm Execution Status....1050
3.7. Inserting Wait Cycles....1052
3.8. ECC Generation and Check....1053
3.8.1. ECC Generation....1054
3.8.2. Calculating a Syndrome....1055
3.8.3. Detecting Errors....1056
3.9. Interrupt....1057
3.10. Bus Error Response....1058
4. Setting Procedure Examples....1059
4.1. Setting Wait Cycle Count....1059
4.2. Reading Flash Memory State/Transition to Reset State....1060
4.3. Enable Programming....1061
4.4. Programming Procedure....1063
4.5. Macro Erasure Procedure....1066
4.6. Procedure for Sector Erase....1067
4.6.1. Sector Erase Suspend....1069
4.6.2. Sector Erase Resume....1070
5. Registers....1071
5.1. TCFLASH Configuration Protection Key Register: TCFCFG_FCPROTKEY (TCFLASH Configuration Protection Key Register)....1072
5.2. TCFLASH Configuration Register: TCFCFG_FCFGR (TCFLASH Configuration Register)....1073
5.3. TCFLASH ECC Control Register: TCFCFG_FECCCTRL (TCFLASH ECC Control Register)....1075
5.4. TCFLASH Data Bit Error Injection Register: TCFCFG_FDATEIR (TCFLASH Data Bit Error Injection Register)....1076
5.5. TCFLASH ECC Bit Error Insertion Register: TCFCFG_FECCEIR (TCFLASH ECC Bit Error Injection Register)....1077
5.6. TCFLASH Interrupt Control Register: TCFCFG_FICTRL (TCFLASH Interrupt Control Register)....1079
5.7. TCFLASH Status Register: TCFCFG_FSTAT (TCFLASH Status Register)....1081
5.8. TCFLASH SEC Interrupt Register: TCFCFG_FSECIR (TCFLASH SEC Interrupt Register)....1084
5.9. TCFLASH ECC Error Address Register: TCFCFG_FECCEAR (TCFLASH ECC Error Address Register)....1086
5.10. TCFLASH Uncorrectable Error Detection Interrupt Register: TCFCFG_FUCEDIR (TCFLASH Un-Correctable Error Detection Interrupt Register)....1087
5.11. TCFLASH Uncorrectable Error Address Register: TCFCFG_FUCEAR (TCFLASH Un-Correctable Error Address Register)....1089

5.12. TCFLASH TCM Buffer Region Configuration Register: TCFCFG_BRCFG (TCFLASH Buffer Region Configuration Register) 1090
5.13. TCFLASH TCM Buffer Region Attribute Register: TCFCFG_BRAT (TCFLASH Buffer Region Attribution Register) 1091
5.14. TCFLASH TCM Buffer Register: TCFCFG_TCMBUF0 to 15 (TCFLASH Buffer0 to 15) 1093
6. Others 1095
7. Support for Dual Port Flash Macro 1096
CHAPTER 18: WorkFLASH 1097
1. Overview 1098
2. Configuration 1099
2.1. Block Diagrams 1099
2.2. Address/Sector Maps in User Mode 1100
3. Explanation of Operation 1103
3.1. Read 1103
3.2. Write and Erase 1104
3.3. Transfer of Write Data with DMA 1106
3.4. Inserting a Wait Cycle 1107
3.5. ECC Generation and Check 1108
3.5.1. ECC Generation 1109
3.5.2. Syndrome Calculation 1110
3.5.3. Error Detection 1111
3.6. Interrupts 1112
3.7. Bus Error Response 1113
3.8. WorkFLASH Access Restrictions 1114
4. Setting Procedure 1115
4.1. Setting the Number of Wait Cycles 1115
4.2. Enabling Writing 1116
4.3. Writing Procedure 1118
4.4. Sector Erasing Procedure 1120
5. Registers 1122
5.1. WorkFLASH Configuration Protection Key Register: WFCFG_CPR 1123
5.2. WorkFLASH Configuration Register: WFCFG_CR 1124
5.3. WorkFLASH ECC Control Register: WFCFG_ECR 1126
5.4. WorkFLASH Write Command Sequencer Configuration Register: WFCFG_WCR 1127
5.5. WorkFLASH Write Command Sequencer Status Register: WFCFG_WSR 1128
5.6. WorkFLASH Data Bit Error Injection Register: WFCFG_DBEIR 1129
5.7. WorkFLASH ECC Bit Error Injection Register: WFCFG_EEIR 1130
5.8. WorkFLASH Interrupt Control Register: WFCFG_ICR 1131
5.9. WorkFLASH Status Register: WFCFG_SR 1133
5.10. WorkFLASH SEC Interrupt Register: WFCFG_SECIR 1136
5.11. WorkFLASH ECC Error Address Register: WFCFG_EEAR 1138
5.12. WorkFLASH Extra Mode Enable Register: WFCFG_EMENR 1139
5.13. WorkFLASH Sequencer Command Register: WFCFG_SEQCM 1140
5.14. WorkFLASH Arbitration Error Register: WFCFG_ARBERR 1142
5.15. WorkFLASH Arbitration Error Clear Register: WFCFG_ARBCLR 1143
5.16. WorkFLASH Bus Error Response Status Register: WFCFG_BERR 1144
5.17. WorkFLASH Bus Error Response Status Clear Register: WFCFG_BERRCLR 1147
5.18. WorkFLASH Uncorrectable Error Status Register: WFCFG_UCESR 1150
5.19. WorkFLASH Uncorrectable Error Address Register: WFCFG_UCEAR 1151

5.20. FLASH Write Arbitration Register: FCFG_WARBR 1152
6. Other 1154
7. Secure Hardware Extension (SHE) Support 1155
8. Dual Port FLASH Macro Support 1157
CHAPTER 19: BootROM Hardware Interface 1159
1. Overview 1160
2. Configuration 1161
3. Explanation of Operation 1162
4. Setting Procedure Example 1163
5. Registers 1164
5.1. EXCFG Lock Release Register (EXCFG_UNLOCK) 1165
5.2. EXCFG Setting Register (EXCFG_CNFG) 1166
5.3. EXCFG Inactive Set - Undefined Instruction Vector Register (EXCFG_UNDEFINACT) 1168
5.4. EXCFG Inactive Set - Supervisor Call Vector Register (EXCFG_SVCINACT) 1169
5.5. EXCFG Inactive Set - Prefetch Abort Vector Register (EXCFG_PABORTINACT) 1170
5.6. EXCFG Inactive Set - Data Abort Vector Register (EXCFG_DABORTINACT) 1171
5.7. EXCFG Active Set - Undefined Instruction Vector Register (EXCFG_UNDEFACT) 1172
5.8. EXCFG Active Set - Supervisor Call Vector Register (EXCFG_SVCACT) 1173
5.9. EXCFG Active Set - Prefetch Abort Vector Register (EXCFG_PABORTACT) 1174
5.10. EXCFG Active Set - Data Abort Vector Register (EXCFG_DABORTACT) 1175
6. Others 1176
CHAPTER 20: BootROM Software Interface 1177
1. Overview 1178
2. BootROM Markers 1179
2.1. Overview of BootROM Markers 1180
2.2. Register Lists 1181
2.3. Security Record (SR) 1183
2.4. Debugger Connection Enable Marker (DDR_DSM) 1184
2.5. Debugger Security Key Marker 0 (DDR_DSKM0) 1185
2.6. Debugger Security Key Marker 1 (DDR_DSKM1) 1186
2.7. Debugger Security Key Marker 2 (DDR_DSKM2) 1187
2.8. Debugger Security Key Marker 3 (DDR_DSKM3) 1188
2.9. SHE Secure Boot Mode Marker (BDR_SBMM) 1189
2.10. SHE Secure Boot Size Marker (BDR_SBSM) 1190
2.11. Debugger Connection Wait Enable Marker (BDR_DWEM) 1191
2.12. Alternative Boot Vector Marker (BDR_ABVM) 1192
2.13. Alternative Boot Vector Enable Marker (BDR_ABVEM) 1193
2.14. Hardware Watchdog Interrupt Configuration Marker (WDR_INTM) 1194
2.15. Hardware Watchdog Trigger 0 Configuration Marker (WDR_TRG0CFGM) 1195
2.16. Hardware Watchdog Trigger 1 Configuration Marker (WDR_TRG1CFGM) 1196
2.17. Hardware Watchdog Lower Limit RUN Setting Marker (WDR_RUNLLM) 1197
2.18. Hardware Watchdog Upper Limit RUN Setting Marker (WDR_RUNULM) 1198
2.19. Hardware Watchdog Lower Limit PSS Setting Marker (WDR_PSSLLM) 1199
2.20. Hardware Watchdog Upper Limit PSS Setting Marker (WDR_PSSULM) 1200
2.21. Hardware Watchdog Reset Delay Counter Marker (WDR_RSTDLYM) 1201
2.22. Hardware Watchdog Configuration Marker (WDR_CFGM) 1202
2.23. Hardware Watchdog Configuration Enable Marker (WDR_CEM) 1204
3. BootROM Operation 1205
4. Notes 1215

CHAPTER 21: Exclusive Access Memory (EAM) 1217
1. Overview 1218
2. Configuration 1219
2.1. Block Diagram 1220
3. Explanation of Operation 1221
3.1. General Aspects of Transfer 1222
3.2. Normal Access 1223
3.3. Exclusive Access 1224
3.3.1. Exclusive Load Instruction (LDREXx) 1225
3.3.2. Exclusive Store Instruction (STREXx) 1228
4. Operation Examples 1230
5. Memory Area 1237
5.1. Exclusive Access Memory 1238
CHAPTER 22: Interrupt Controller 1239
1. Overview 1240
2. Configuration 1241
3. Explanation of Operation 1242
4. Setting Procedure Examples 1256
5. Registers 1257
5.1. IRC NMI Vector Address Status Register (IRCn_NMIVAS) 1258
5.2. IRC NMI Status Register (IRCn_NMIST) 1259
5.3. IRC IRQ Vector Address Status Register (IRCn_IRQVAS) 1260
5.4. IRC IRQ Status Register (IRCn_IRQST) 1261
5.5. IRC NMI Vector Address Register (IRCn_NMIVAm) 1263
5.6. IRC IRQ Vector Address Register (IRCn_IRQVAr) 1264
5.7. IRC NMI Priority Level Register (IRCn_NMIPL0) 1265
5.8. IRC NMI Priority Level Register (IRCn_NMIPL1 to IRCn_NMIPL7) 1267
5.9. IRC IRQ Priority Level Register (IRCn_IRQPL0 to IRCn_IRQPL127) 1269
5.10. IRC NMI Software Interrupt Set Register (IRCn_NMIS) 1271
5.11. IRC NMI Software Interrupt Reset Register (IRCn_NMIR) 1272
5.12. IRC NMI Software Interrupt Status Register (IRCn_NMISIS) 1273
5.13. IRC IRQ Software Interrupt Set Register (IRCn_IRQS0 to IRCn_IRQS15) 1274
5.14. IRC IRQ Software Interrupt Reset Register (IRCn_IRQR0 to IRCn_IRQR15) 1275
5.15. IRC IRQ Software Interrupt Status Register (IRCn_IRQSIS0 to IRCn_IRQSIS15) 1277
5.16. IRC IRQ Channel Enable Set Register (IRCn_IRQCES0 to IRCn_IRQCES15) 1279
5.17. IRC IRQ Channel Enable Clear Register (IRCn_IRQCEC0 to IRCn_IRQCEC15) 1280
5.18. IRC IRQ Channel Enable Setting Register (IRCn_IRQCE0 to IRCn_IRQCE15) 1281
5.19. IRC NMI Hold Clear Register (IRCn_NMIHC) 1282
5.20. IRC NMI Hold Status Register (IRCn_NMIHS) 1283
5.21. IRC IRQ Hold Clear Register (IRCn_IRQHC) 1284
5.22. IRC IRQ Hold Status Register (IRCn_IRQHS0 to IRCn_IRQHS15) 1285
5.23. IRC IRQ Priority Level Mask Register (IRCn_IRQPLM) 1287
5.24. IRC Control/Status Register (IRCn_CSR) 1288
5.25. IRC NMI RAW Status Register (IRCn_NMIRS) 1290
5.26. IRC NMI Preprocessed Status Register (IRCn_NMIPS) 1291
5.27. IRC IRQ RAW Status Register (IRCn_IRQRS0 to IRCn_IRQRS15) 1292
5.28. IRC IRQ Preprocessed Status Register (IRCn_IRQPS0 to IRCn_IRQPS15) 1294
5.29. IRC Unlock Register (IRCn_UNLOCK) 1296
5.30. IRC ECC Error Interrupt Register (IRCn_EEI) 1297

5.31. IRC ECC Address Number Register (IRCn_EAN) 1299
5.32. IRC ECC Test Register (IRCn_ET) 1300
5.33. IRC ECC Error Bit Register (IRCn_EEB0 to IRCn_EEB1) 1301
5.34. IRC ECC Error Bit Register (IRCn_EEB2) 1302
5.35. IRC NMI Vector Address Status Register (IRCn_NMIVASBR) 1303
5.36. IRC NMI Vector Address Status Mirror Register (IRC_NMIVASBR) 1304
5.37. IRC ECC Error Vector Address Register (IRCn_IRQEEVA) 1305
6. Others 1306
CHAPTER 23: Time Protection 1307
1. Overview 1308
2. Configuration 1309
3. Explanation of Operation 1310
4. Setting Procedure Examples 1312
5. Registers 1315
5.1. TPU Lock Release Register (TPU0_UNLOCK) 1316
5.2. TPU Lock Status Register (TPU0_LST) 1317
5.3. TPU Configuration Register (TPU0_CFG) 1318
5.4. TPU Timer Interrupt Request Register (TPU0_TIR) 1320
5.5. TPU Timer Status Register (TPU0_TST) 1321
5.6. TPU Timer Interrupt Enable Register (TPU0_TIE) 1322
5.7. TPU Timer m Control Register 0 (TPU0_TCN0m) 1323
5.8. TPU Timer m Control Register 1 (TPU0_TCN1m) 1325
5.9. TPU Timer m Current Count Value Register (TPU0_TCCm) 1327
6. Others 1328
CHAPTER 24: Security 1329
1. Overview 1330
2. Security Scope and Access Restriction by the Security 1331
2.1. Security Scope 1331
2.2. Access Restriction by the Security 1332
3. Using and Configuring the Security 1333
3.1. Set Security Setting 1334
3.2. Erasing Security Setting 1335
3.3. Security Marker Definition 1337
3.3.1. Security Enable Marker (MK_SER) 1339
3.3.2. Security Scope Marker (MK_SSR) 1340
3.3.3. Chip Erase Enable (MK_CEER) 1341
3.3.4. Security Overwrite Enable Marker (MK_SOER) 1342
3.3.5. Sector Write Permission Overwrite Enable Marker (MK_SWPOER) 1343
3.3.6. Work Flash Sector Write Permissions (MK_WSWPR) 1344
3.3.7. Code Flash Sector Write Permissions, Small Sectors (MK_C0SWPR) 1345
3.3.8. Code Flash Sector Write Permissions, Large Sectors (MK_C1SWPR) 1346
3.3.9. Code Flash Sector Write Permissions, Large Sectors (MK_C2SWPR) 1347
3.3.10. Code Flash Sector Write Permissions, Large Sectors (MK_C3SWPR) 1348
3.4. Gain Access in USERMODE 1349
4. Registers 1351
4.1. Security Status Register (TCFCFG0_SECSTAT) 1352
4.2. Security Enable Register (TCFCFG0_SER) 1355
4.3. Security Scope Register (TCFCFG0_SSR) 1357
4.4. Chip Erase Enable Register (TCFCFG0_CEER) 1359

4.5. Security Overwrite Enable Register (TCFCFG0_SOER) 1361
4.6. Sector Write Permission Overwrite Enable Register (TCFCFG0_SWPOER) 1363
4.7. Work Flash Sector Write Permissions Register (TCFCFG0_WSWP) 1365
4.8. Code Flash 0 Sector Write Permissions of the Small Sectors Register (TCFCFG0_C0SWP) 1368
4.9. Code Flash 0 Sector Write Permissions of the Large Sectors Register (TCFCFG0_C1SWP) 1370
4.10. Code Flash 1 Sector Write Permissions of the Large Sectors Register (TCFCFG0_C2SWP) 1376
4.11. Code Flash 2 Sector Write Permissions of the Large Sectors Register (TCFCFG0_C3SWP) 1382

CHAPTER 25: Secure Hardware Extension (SHE) 1389

1. Outline of the Secure Hardware Extension (SHE) Module 1390
2. Secure Hardware Extension Registers 1392
2.1. Configuration Registers for the Command Interface 1393
2.1.1. SHE Command Register (SHE_CMD) 1393
2.1.2. SHE Command Cancel Register (SHE_CMDCANCEL) 1395
2.1.3. SHE Clock Control Register (SHE_CLKCTRL) 1396
2.2. Status Registers for the Command Interface 1398
2.2.1. SHE Status Register (SHE_STATUS) 1398
2.2.2. SHE Error Code Register (SHE_ERC) 1402
2.2.3. SHE Clock Status Register (SHE_CLKSTAT) 1405
2.2.4. SHE Module ID Register (SHE_MID) 1406
2.3. Interrupt Registers for Command Interface 1407
2.3.1. Interrupt Request Register (SHE_IRQ) 1407
2.3.2. Interrupt Request Enable Register (SHE_IRQEN) 1410
2.3.3. Interrupt Request Clear Register (SHE_IRQCLR) 1412
2.4. Configuration Registers for the Data Interface 1414
2.4.1. Input Channel Master Start Address Register (SHE_IMSTADDR) 1414
2.4.2. Output Channel Master Start Address Register (SHE_OMSTADDR) 1416
2.4.3. Input Channel Master Data Transfer Counter (SHE_IMSTCNT) 1418
2.4.4. Output Channel Master Data Transfer Counter (SHE_OMSTCNT) 1420
2.4.5. Input Channel Master Start Trigger (SHE_IMSTSTART) 1422
2.4.6. Output Channel Master Start Trigger (SHE_OMSTSTART) 1423
2.4.7. Input FIFO Configuration Register (SHE_IFIFOCFG) 1424
2.4.8. Output FIFO Configuration Register (SHE_OFIFOCFG) 1426
2.4.9. Input FIFO Compare Value Register (SHE_COMPARE0 to 1) 1428
2.5. Status Registers for Data Interface 1432
2.5.1. Compare Register Access Status Register (SHE_COMPACC) 1432
2.5.2. Data Master Status Register (SHE_MSTSTATUS) 1433
2.5.3. Input Channel Master Error Response Address Register (SHE_IMSTERRADDR) 1436
2.5.4. Output Channel Master Error Response Address Register (SHE_OMSTERRADDR) 1438
2.5.5. FIFO Status Register (SHE_FIFOSTATUS) 1440
2.5.6. FIFO Load Register (SHE_FIFOLOAD) 1442
2.5.7. Input FIFO Data Counter Register (SHE_DATACNT0 to 1) 1444
2.6. Data Transfer Registers 1447
2.6.1. Input FIFO Write Data Register (SHE_IFIFOWRDATA0 to 31) 1447

2.6.2. Output FIFO Read Data Register (SHE_OFIFORDDATA0 to 31) 1449
3. Operation of the Secure Hardware Extension 1451
3.1. Operation of the Command Interface 1451
3.1.1. Power Saving Functionality 1462
3.1.2. Command Interface Access Issues 1464
3.2. Operation of the Data Interface 1465
3.3. Operation of the FIFO Unit 1470
4. Notes on Using SHE 1474
4.1. General Notes on Using SHE 1474
4.1.1. Notes on Using SHE Commands 1475
4.1.2. SHE Used Error Codes/Used Cancel Codes 1477
5. Enhanced Secure Hardware Extension 1482
5.1. Additional Generate Purpose Keys 1482
5.2. Additional Verify Only Flag 1483
5.3. Additional User-accessible Functions 1484
5.3.1. CMD_LOAD_KEY_ESHE 1486
6. References 1487
CHAPTER 26: Memory Protection Unit for AXI of SHE 1489
1. Overview 1490
2. Configuration and Block Diagram 1491
3. Operation of the MPU AXI 1492
4. Registers 1498
4.1. MPU AXI Control Register (MPUXSHE_CTRL0) 1501
4.2. MPU AXI NMI Enable Register (MPUXSHE_NMIEN) 1503
4.3. MPU AXI Write Error Control Register (MPUXSHE_WERRC) 1504
4.4. MPU AXI Write Error Address Register (MPUXSHE_WERRA) 1505
4.5. MPU AXI Read Error Control Register (MPUXSHE_RERRC) 1506
4.6. MPU AXI Read Error Address Register (MPUXSHE_RERRA) 1507
4.7. MPU AXI Region Control Registers (MPUXSHE_CTRL1~8) 1508
4.8. MPU AXI Start Address Registers (MPUXSHE_SADDR1~8) 1509
4.9. MPU AXI End Address Registers (MPUXSHE_EADDR1~8) 1510
4.10. MPU AXI Unlock Register (MPUXSHE_UNLOCK) 1511
4.11. MPU AXI Module ID Register (MPUXSHE_MID) 1512
5. Notes on Using MPU AXI 1513
CHAPTER 27: DMA Complex Subsystem 1515
1. Overview 1516
2. Configuration and Block Diagram 1517
3. Operation of the DMA COMPLEX SUBSYSTEM 1519
3.1. Additional Status Registers 1520
3.2. CMCHIC Reloading 1521
3.3. Reload Timer in DMA Additional Control 1522
3.3.1. Reload Timer Trigger Inputs 1523
3.3.2. Reload Timer Usage Examples 1524
4. Registers 1527
4.1. DMA Additional Control Additional Status Register 0 (DMAAn_ASR0) 1528
4.2. DMA Additional Control Additional Status Register 1 (DMAAn_ASR1) 1529
4.3. DMA Additional Control Additional Status Register 2 (DMAAn_ASR2) 1530
4.4. DMA Additional Control Additional Status Register 3 (DMAAn_ASR3) 1531
4.5. DMA Additional Control Additional Status Register i (DMAAn_ASRi) (i=4 to 19) 1532

4.6. DMA Additional Control CMCHIC Reload Data Bank Register i (DMAAn_CMCHICRDBi) (i=0 to 7) 1533
4.7. DMA Additional Control Reload Timer Trigger Select Register (DMAAn_RTTSR) 1534
4.8. DMA Additional Control Reload Timer Synchronous Software Start Register (DMAAn_RTSSSR) 1536
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
7.1. Internal Clock and External Event Counter Operations of 32-Bit Reload Timer 1540
7.2. Underflow Operation of 32-Bit Reload Timer 1542
7.3. Output Functions of 32-Bit Reload Timer 1544
7.4. Counter Operation State 1545
7.5. DMA Operation 1546
8. Registers in DMA COMPLEX SUBSYSTEM 1547
8.1. DMA Configuration Register (DMAAn_RLTm_DMACFG) 1548
8.2. Timer Control Status Register (DMAAn_RLTm_TMCSR) 1549
8.3. 32-Bit Reload Register (DMAAn_RLTm_TMRLR) 1553
8.4. 32-Bit Timer Register (DMAAn_RLTm_TMR) 1554
CHAPTER 28: DMA Controller 1555
1. Overview 1556
2. Configuration 1557
3. Operational Description 1559
3.1. DMA Channels 1561
3.2. DMA Client Matrix 1572
3.3. DMA Arbiter 1574
3.4. DMA AHB Slave Interface 1577
3.5. Additional Information 1578
4. Registers 1582
4.1. DMA Controller Global Configuration Register (DMAi_R) 1583
4.2. DMA Controller Global Completion Interrupt 1 Register (DMAi_DIRQ1) 1586
4.3. DMA Controller Global Completion Interrupt 2 Register (DMAi_DIRQ2) 1587
4.4. DMA Controller Global Error Interrupt 1 Register (DMAi_EDIRQ1) 1588
4.5. DMA Controller Global Error Interrupt 2 Register (DMAi_EDIRQ2) 1589
4.6. DMA Controller ID Register (DMAi_ID) 1590
4.7. DMA Controller Channel Configuration A Register Channel 'n' (DMAi_An) 1591
4.8. DMA Controller Channel Configuration B Register Channel 'n' (DMAi_Bn) 1594
4.9. DMA Controller Channel Configuration Source Address Register Channel 'n' (DMAi_SAn) 1598
4.10. DMA Controller Channel Configuration Destination Address Register Channel 'n' (DMAi_DAn) 1599
4.11. DMA Controller Channel Configuration C Register Channel 'n' (DMAi_Cn) 1600
4.12. DMA Controller Channel Configuration D Register Channel 'n' (DMAi_Dn) 1601
4.13. DMA Controller Channel Configuration E Register Channel 'n' (DMAi_En) 1604
4.14. DMA Controller Channel Configuration Source Address Shadow Register Channel 'n' (DMAi_SASHDWn) 1605
4.15. DMA Controller Channel Configuration Destination Address Shadow Register Channel 'n' (DMAi_DASHDWn) 1607

4.16. DMA Controller Client Matrix Internal Client Interface Configuration Register 'm' (DMAi_CMICICm) 1609
4.17. DMA Controller Client Matrix Channel Interface Configuration Register 'n' (DMAi_CMCHICn) 1611
5. Additional Information 1613
CHAPTER 29: MPU16 AHB 1615
1. Outline of the MPU16 AHB 1616
2. MPU16 AHB Registers 1618
2.1. MPU16 AHB Control Register (MPUHn_CTRL0) 1620
2.2. MPU16 AHB NMI Enable Register (MPUHn_NMIEN) 1622
2.3. MPU16 AHB Memory Error Control Register (MPUHn_MERRC) 1623
2.4. MPU AHB Memory Error Address Register (MPUHn_MERRA) 1624
2.5. MPU16 AHB Region Control Registers (MPUHn_CTRL1 to 16) 1625
2.6. MPU16 AHB Start Address Registers (MPUHn_SADDR1 to 16) 1627
2.7. MPU16 AHB End Address Registers (MPUHn_EADDR1 to 16) 1628
2.8. MPU16 AHB Unlock Register (MPUHn_UNLOCK) 1629
2.9. MPU16 AHB Module ID Register (MPUHn_MID) 1631
3. Operation of the MPU16 AHB 1632
CHAPTER 30: CAN FD Message RAM ECC Function 1637
1. Overview 1638
2. Configuration 1639
3. Interrupts 1641
4. Explanation of Operation 1642
4.1. Message RAM ECC Generation 1643
4.2. Message RAM ECC Check 1644
4.3. ECC False Error Control Function of the Message RAM 1645
5. Setup Procedure Examples 1646
6. Registers 1647
6.1. CAN FD ECC Error Control Register (FDECR) 1648
6.2. CAN FD ECC Error Status Register (FDESR) 1650
6.3. CAN FD ECC Error Status Clear Register (FDESCR) 1652
6.4. CAN FD ECC Double-Bit Error Address Register (FDDEAR) 1653
6.5. CAN FD ECC Single-Bit Error Address Register (FDSEAR) 1654
6.6. CAN FD ECC False Error Control Register (FDFECR) 1655
7. Precautions 1658
CHAPTER 31: CAN FD Controller 1659
1. Overview 1660
2. Configuration 1661
2.1. Block Diagram 1661
2.2. Dual Clock Sources 1663
2.3. Dual Interrupt Lines 1664
3. Explanation of Operations 1665
3.1. Operating Modes 1665
3.1.1. Software Initialization 1665
3.1.2. Normal Operation 1667
3.1.3. CAN FD Operation 1668
3.1.4. Transmitter Delay Compensation 1670
3.1.5. Restricted Operation Mode 1671
3.1.6. Bus Monitoring Mode 1673

3.1.7. Disabled Automatic Retransmission 1673
3.1.8. Power Down (Sleep Mode) 1675
3.1.9. Test Modes 1675
3.2. Timestamp Generation 1677
3.3. Timeout Counter 1678
3.4. Rx Handling 1679
3.4.1. Acceptance Filtering 1680
3.4.2. Rx FIFOs 1686
3.4.3. Dedicated Rx Buffers 1689
3.4.4. Debug on CAN Support 1690
3.4.5. Protocol Exception Event 1692
3.5. Tx Handling 1693
3.5.1. Transmit Pause 1693
3.5.2. Dedicated Tx Buffers 1694
3.5.3. Tx FIFO 1694
3.5.4. Tx Queue 1695
3.5.5. Mixed Dedicated Tx Buffers/Tx FIFO 1696
3.5.6. Mixed Dedicated Tx Buffers/Tx Queue 1696
3.5.7. Transmit Cancellation 1697
3.5.8. Tx Event Handling 1698
3.6. FIFO Acknowledge Handling 1698
3.7. Configuring the CAN Bit Timing 1699
3.7.1. CAN Bit Timing 1699
3.7.2. CAN Bit Rates 1702
4. Setup Procedure Examples 1704
4.1. Configuration of CAN Bus 1706
4.2. Configuration of Message RAM 1707
4.2.1. Configuration of ID Filter List 1707
4.2.2. Configuration of Rx Buffer and Rx FIFO 1708
4.2.3. Configuration of Tx Buffer and Tx FIFO/Queue 1709
4.2.4. Configuration of ID Filter 1710
4.3. Configuration of Error Monitor 1711
4.4. Configuration of Interrupt 1712
4.5. Control of Communication 1713
4.5.1. Configuration of Transmission Frame 1714
4.6. Interrupt Handling Operation 1715
4.6.1. Bus_Off Status Handling Operation 1716
4.6.2. Message RAM Access Failure Handling Operation 1717
4.6.3. Bit Error Handling Operation 1718
4.6.4. Tx Event FIFO Handling Operation 1719
4.6.5. Dedicated Rx Buffer Handling Operation 1720
4.6.6. High Priority Message Handling Operation 1721
4.6.7. Rx FIFO Handling Operation 1722
5. Registers 1723
5.1. Core Release Register (CREL) 1725
5.2. Endian Register (ENDN) 1727
5.3. Data Bit Timing & Prescaler Register (DBTP) 1728
5.4. Test Register (TEST) 1731
5.5. RAM Watchdog (RWD) 1733

5.6. CC Control Register (CCCR) 1735
5.7. Nominal Bit Timing & Prescaler Register (NBTP) 1739
5.8. Timestamp Counter Configuration (TSCC) 1741
5.9. Timestamp Counter Value (TSCV) 1743
5.10. Timeout Counter Configuration (TOCC) 1745
5.11. Timeout Counter Value (TOCV) 1747
5.12. Error Counter Register (ECR) 1749
5.13. Protocol Status Register (PSR) 1751
5.14. Transmitter Delay Compensation Register (TDCR) 1755
5.15. Interrupt Register (IR) 1757
5.16. Interrupt Enable (IE) 1764
5.17. Interrupt Line Select (ILS) 1766
5.18. Interrupt Line Enable (ILE) 1768
5.19. Global Filter Configuration (GFC) 1770
5.20. Standard ID Filter Configuration (SIDFC) 1772
5.21. Extended ID Filter Configuration (XIDFC) 1774
5.22. Extended ID AND Mask (XIDAM) 1776
5.23. High Priority Message Status (HPMS) 1777
5.24. New Data 1 (NDAT1) 1779
5.25. New Data 2 (NDAT 2) 1780
5.26. Rx FIFO 0 Configuration (RXF0C) 1781
5.27. Rx FIFO 0 Status (RXF0S) 1783
5.28. Rx FIFO 0 Acknowledge (RXF0A) 1785
5.29. Rx Buffer Configuration (RXBC) 1786
5.30. Rx FIFO 1 Configuration (RXF1C) 1788
5.31. Rx FIFO 1 Status (RXF1S) 1790
5.32. Rx FIFO 1 Acknowledge (RXF1A) 1793
5.33. Rx Buffer/FIFO Element Size Configuration (RXESC) 1794
5.34. Tx Buffer Configuration (TXBC) 1796
5.35. Tx FIFO/Queue Status (TXFQS) 1798
5.36. Tx Buffer Element Size Configuration (TXESC) 1800
5.37. Tx Buffer Request Pending (TXBRP) 1802
5.38. Tx Buffer Add Request (TXBAR) 1804
5.39. Tx Buffer Cancellation Request (TXBCR) 1806
5.40. Tx Buffer Transmission Occurred (TXBTO) 1807
5.41. Tx Buffer Cancellation Finished (TXBCF) 1808
5.42. Tx Buffer Transmission Interrupt Enable (TXBTIE) 1809
5.43. Tx Buffer Cancellation Finished Interrupt Enable (TXBCIE) 1810
5.44. Tx Event FIFO Configuration (TXEFC) 1811
5.45. Tx Event FIFO Status (TXEFS) 1813
5.46. Tx Event FIFO Acknowledge (TXEFA) 1815
6. Message RAM 1816
6.1. Message RAM Configuration 1817
6.2. Rx Buffer and FIFO Element 1819
6.3. Tx Buffer Element 1822
6.4. Tx Event FIFO Element 1826
6.5. Standard Message ID Filter Element 1828
6.6. Extended Message ID Filter Element 1830

CHAPTER 32: CAN Prescaler 1833
1. Overview 1834
2. Configuration 1834
3. Explanation of Operation 1835
4. Setting Procedure 1836
5. Registers 1838
   5.1. CAN Prescaler Control Register (CANP_CTR) 1839
   5.2. CAN Prescaler Status Register (CANP_STR) 1841
6. Restriction 1842

CHAPTER 33: Overview of Multi-Function Serial Interface 1843
1. Overview of the UART (Asynchronous Serial Interface) 1844
2. Overview of the CSIO (Clock Synchronous Serial Interface) 1846
3. Overview of the LIN Interface (v2.1) (LIN Communication Control Interface (v2.1)) 1847
   3.1. Manual Mode 1847
   3.2. Assist Mode 1849
4. Overview of the I^{2}C Interface (I^{2}C Communication Control Interface) 1851
5. Block Diagram 1852
6. Registers of the Multi-Function Serial Interface 1855

CHAPTER 34: UART (Asynchronous Serial Interface) 1861
1. Overview of the UART (Asynchronous Serial Interface) 1862
2. Interrupts from the UART 1863
   2.1. Occurrence of Reception Interrupts and Flag Set Timing 1866
   2.2. Occurrence of Interrupts and Flag Set Timing When the Reception FIFO Is Used 1868
   2.3. Occurrence of Transmission Interrupts and Flag Set Timing 1870
   2.4. Occurrence of Interrupts and Flag Set Timing When the Transmission FIFO Is Used 1871
   2.5. Occurrence of Timer Interrupts and Flag Set Timing 1872
3. UART Operation 1873
4. Serial Timer Operation 1881
5. Test Mode 1890
6. Dedicated Baud Rate Generator 1891
   6.1. Setting the Baud Rate 1892
7. Block Transfer 1897
   7.1. ACK Mode 1898
   7.2. RW Access Mode 1902
8. Setup Procedure and Program Flow for Operation Mode 0 (Asynchronous Normal Mode) 1906
9. Setup Procedure and Program Flow for Operation Mode 1 (Asynchronous Multi-processor Mode) 1909
10. Registers of UART (Asynchronous Serial Interface) 1913
    10.1. Serial Control Register (SCR) 1917
    10.2. Serial Mode Register (SMR) 1920
    10.3. Serial Status Register (SSR) 1922
    10.4. Extended Communication Control Register (ESCR) 1926
    10.5. Reception Data Register/Transmission Data Register (RDR/TDR) 1929
    10.6. Serial Auxiliary Control Status Register (SACSR) 1933
    10.7. Serial Timer Register (STMR) 1937
    10.8. Serial Timer Comparison Register (STMCR) 1938
    10.9. Transfer Byte Register (TBYTE0) 1940
    10.10. Baud Rate Generator Register 1, 0 (BGR1, BGR0) 1941
    10.11. FIFO Control Register 1 (FCR1) 1943

10.12. FIFO Control Register 0 (FCR0) 1947
10.13. FIFO Byte Register (FBYTE) 1951
10.14. Transmission FIFO Interrupt Control Register (FTICR) 1953
10.15. Extended Control Register (ECR) 1955
10.16. Extended Status Register (ESR) 1957
10.17. Transmission Block Size Register (TBSIZE) 1960
10.18. Serial Auxiliary Control Status Clear Register (SACSRC) 1961
10.19. FIFO Control Clear Register 1 (FCR1C) 1963
10.20. FIFO Control Clear Register 0 (FCR0C) 1964
10.21. Extended Status Clear Register (ESRC) 1965
10.22. Serial Auxiliary Control Status Set Register (SACSRS) 1966
10.23. FIFO Control Set Register 1 (FCR1S) 1968
10.24. FIFO Control Set Register 0 (FCR0S) 1969

CHAPTER 35: CSIO (Clock Synchronous Serial Interface) 1971

1. Overview of the CSIO (Clock Synchronous Serial Interface) 1972
2. Interrupts of the CSIO (Clock Synchronous Serial Interface) 1973
   2.1. Reception Interrupt Generation and Flag Set Timing 1976
   2.2. Interrupt Generation and Flag Set Timing When the Reception FIFO Is Used 1978
   2.3. Transmission Interrupt Generation and Flag Set Timing 1980
   2.4. Interrupt Generation and Flag Set Timing When the Transmission FIFO Is Used 1981
   2.5. Timer Interrupt Generation and Flag Set Timing 1982
   2.6. Chip Select Error Occurrence and Flag Set Timing 1983
3. Operation of the CSIO (Clock Synchronous Serial Interface) 1986
   3.1. Normal Transfer (I) 1986
   3.2. Normal Transfer (II) 1995
   3.3. SPI Transfer (I) 2005
   3.4. SPI Transfer (II) 2014
4. Serial Timer Operation 2025
5. Operation of Serial Chip Select 2037
6. Test Mode 2049
7. Dedicated Baud Rate Generator 2050
   7.1. Setting the Baud Rate 2051
   7.2. Setup Procedure and Program Flow for the CSIO (Clock Synchronous Serial Interface) 2054
8. Block Transfer 2057
   8.1. ACK Mode 2058
   8.2. RW Access Mode 2063
9. Registers of the CSIO (Clock Synchronous Serial Interface) 2067
   9.1. Serial Control Register (SCR) 2070
   9.2. Serial Mode Register (SMR) 2074
   9.3. Serial Status Register (SSR) 2077
   9.4. Extended Communication Control Register (ESCR) 2080
   9.5. Reception Data Register/Transmission Data Register (RDR/TDR) 2083
   9.6. Serial Auxiliary Control Status Register (SACSR) 2087
   9.7. Serial Timer Register (STMR) 2093
   9.8. Serial Timer Comparison Register (STMCR) 2094
   9.9. Serial Chip Select Control Status Register (SCSCR) 2096
   9.10. Serial Chip Select Timing Registers (SCSTR3 to SCSTR0) 2100
   9.11. Serial Chip Select Format Registers (SCSFR2 to SCSFR0) 2103

9.12. Transfer Byte Registers (TBYTE3 to TBYTE0) 2113
9.13. Baud Rate Generator Registers 1, 0 (BGR1, BGR0) 2114
9.14. FIFO Control Register 1 (FCR1) 2115
9.15. FIFO Control Register 0 (FCR0) 2118
9.16. FIFO Byte Register (FBYTE) 2122
9.17. Transmission FIFO Interrupt Control Register (FTICR) 2124
9.18. Extended Control Register (ECR) 2126
9.19. Extended Status Register (ESR) 2128
9.20. Transmission Block Size Register (TBSIZE) 2130
9.21. Serial Auxiliary Control Status Clear Register (SACSRC) 2131
9.22. FIFO Control Clear Register 1 (FCR1C) 2133
9.23. FIFO Control Clear Register 0 (FCR0C) 2134
9.24. Extended Status Clear Register (ESRC) 2135
9.25. Serial Auxiliary Control Status Set Register (SACSRS) 2136
9.26. FIFO Control Set Register 1 (FCR1S) 2138
9.27. FIFO Control Set Register 0 (FCR0S) 2139

CHAPTER 36: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 2141

1. LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 2142
   1.1. Manual Mode 2142
   1.2. Assist Mode 2144
2. LIN Interface (v2.1) Interrupts 2146
   2.1. Manual Mode 2146
      2.1.1. Occurrence of Reception Interrupts and Flag Set Timing 2149
      2.1.2. Occurrence of Interrupts and Flag Set Timing When Using Reception FIFO 2150
      2.1.3. Occurrence of Transmission Interrupts and Flag Set Timing 2152
      2.1.4. Occurrence of Interrupts and Flag Set Timing When Using Transmission FIFO 2153
      2.1.5. Occurrence of Timer Interrupts and Flag Set Timing 2154
      2.1.6. Set Timing of LIN Break Field Detection Flag (LBD) 2154
      2.1.7. Occurrence of Sync Field Detection Interrupt and Flag Set Timing 2155
   2.2. Assist Mode 2156
      2.2.1. Occurrence of Reception Interrupts and Flag Set Timing in Assist Mode 2160
      2.2.2. Occurrence of Interrupts and Flag Set Timing When Using Reception FIFO 2168
      2.2.3. Occurrence of Transmission Interrupts and Flag Set Timing 2169
      2.2.4. Occurrence of Interrupts and Flag Set Timing When Using Transmission FIFO 2171
      2.2.5. Occurrence of Timer Interrupts and Flag Set Timing 2171
      2.2.6. Occurrence of Status Interrupt and Flag Set Timing in Assist Mode 2172
3. Serial Timer Operation 2175
4. Test Mode 2180
   4.1. Manual Mode 2180
   4.2. Assist Mode 2181
5. Dedicated Baud Rate Generator 2186
   5.1. Setting Baud Rate 2187
6. Block Transfer 2192
   6.1. ACK Mode 2193
   6.2. RW Access Mode 2198
7. LIN Interface (V2.1) Operation 2202
   7.1. Manual Mode 2202

7.2. Assist Mode 2216
8. Setup Procedure and Program Flow for Operation Mode 3 (LIN Communication Mode) 2237
8.1. Manual Mode 2238
8.2. Assist Mode 2242
9. LIN Interface (V2.1) Registers 2247
9.1. Serial Control Register (SCR) 2250
9.2. Serial Mode Register (SMR) 2254
9.3. Serial Status Register (SSR) 2256
9.4. Extended Communication Control Register (ESCR) 2260
9.5. Reception Data Register/Transmission Data Register (RDR/TDR) 2263
9.6. Serial Auxiliary Control Status Register (SACSR) 2266
9.7. Serial Timer Register (STMR) 2271
9.8. Serial Timer Comparison Register (STMCR) 2272
9.9. Sync Field Upper Limit Register (SFUR) 2273
9.10. Sync Field Lower Limit Register (SFLR) 2274
9.11. Baud Rate Generator Register 1, 0 (BGR1, BGR0) 2275
9.12. LIN Assist Mode Status Register (LAMSR) 2277
9.13. LIN Assist Mode Control Register (LAMCR) 2280
9.14. LIN Assist Mode Interrupt Enable Register (LAMIER) 2283
9.15. LIN Assist Mode Transmission/Reception ID Register (LAMTID/LAMRID) 2285
9.16. LIN Assist Mode Error Status Register (LAMESR) 2287
9.17. LIN Assist Mode Error Test Register (LAMERT) 2290
9.18. FIFO Control Register 1 (FCR1) 2293
9.19. FIFO Control Register 0 (FCR0) 2296
9.20. FIFO Byte Register (FBYTE) 2300
9.21. Transmission FIFO Interrupt Control Register (FTICR) 2302
9.22. Extended Control Register (ECR) 2304
9.23. Extended Status Register (ESR) 2306
9.24. Transmission Block Size Register (TBSIZE) 2309
9.25. Serial Control Clear Register (SCRC) 2310
9.26. Serial Mode Clear Register (SMRC) 2312
9.27. Serial Status Clear Register (SSRC) 2313
9.28. Extended Communication Control Clear Register (ESCRC) 2314
9.29. Serial Auxiliary Control Status Clear Register (SACSRC) 2315
9.30. LIN Assist Mode Status Clear Register (LAMSRC) 2317
9.31. LIN Assist Mode Control Clear Register (LAMCRC) 2318
9.32. LIN Assist Mode Interrupt Enable Clear Register (LAMIERC) 2319
9.33. LIN Assist Mode Error Status Clear Register (LAMESRC) 2321
9.34. FIFO Control Clear Register 1 (FCR1C) 2322
9.35. FIFO Control Clear Register 0 (FCR0C) 2323
9.36. Extended Status Clear Register (ESRC) 2324
9.37. Serial Control Set Register (SCRS) 2325
9.38. Serial Mode Set Register (SMRS) 2327
9.39. Serial Status Set Register (SSRS) 2328
9.40. Extended Communication Control Set Register (ESCRS) 2329
9.41. Serial Auxiliary Control Status Set Register (SACSRS) 2330
9.42. LIN Assist Mode Control Set Register (LAMCRS) 2332
9.43. LIN Assist Mode Interrupt Enable Set Register (LAMIERS) 2333
9.44. FIFO Control Set Register 1 (FCR1S) 2335

9.45. FIFO Control Set Register 0 (FCR0S) 2336
CHAPTER 37: I²C Interface (I²C Communication Control Interface) 2339
1. Overview of the I²C Interface (I²C Communication Control Interface) 2340
2. I²C Interface Interrupts 2341
2.1. Timer Interrupt Generation and Flag Set Timing 2344
3. I²C Interface Operation 2345
3.1. Master Mode 2346
3.2. Slave Mode 2388
3.3. Bus Error 2398
4. Serial Timer Operation 2399
5. Dedicated Baud Rate Generator 2403
6. Block Transfer 2405
6.1. ACK Mode 2406
6.2. RW Access Mode 2410
7. I²C Communication Operation Flowchart Examples 2414
8. I²C Interface Registers 2422
8.1. I²C Bus Control Register (IBCR) 2424
8.2. Serial Mode Register (SMR) 2432
8.3. I²C Bus Status Register (IBSR) 2434
8.4. Serial Status Register (SSR) 2439
8.5. Reception Data Register/Transmission Data Register (RDR/TDR) 2444
8.6. Serial Auxiliary Control Status Register (SACSR) 2446
8.7. Serial Timer Register (STMR) 2450
8.8. Serial Timer Comparison Register (STMCR) 2451
8.9. Noise Filter Control Register (NFCR) 2452
8.10. Extended I²C Bus Control Register (EIBCR) 2454
8.11. 7-Bit Slave Address Mask Register (ISMK) 2457
8.12. 7-Bit Slave Address Register (ISBA) 2458
8.13. Baud Rate Generator Registers 1, 0 (BGR1, BGR0) 2459
8.14. FIFO Control Register 1 (FCR1) 2460
8.15. FIFO Control Register 0 (FCR0) 2464
8.16. FIFO Byte Register (FBYTE) 2468
8.17. Transmission FIFO Interrupt Control Register (FTICR) 2470
8.18. Extended Control Register (ECR) 2472
8.19. Extended Status Register (ESR) 2474
8.20. Transmission Block Size Register (TBSIZE) 2476
8.21. I²C Bus Control Clear Register (IBCRC) 2477
8.22. Serial Mode Clear Register (SMRC) 2479
8.23. Serial Status Clear Register (SSRC) 2480
8.24. I²C Bus Status Clear Register (IBSRC) 2481
8.25. Serial Auxiliary Control Status Clear Register(SACSRC) 2482
8.26. FIFO Control Clear Register 1 (FCR1C) 2484
8.26.1. Bit Configuration of FIFO Control Clear Register 1 (FCR1C) 2484
8.27. FIFO Control Clear Register 0 (FCR0C) 2485
8.28. Extended Status Clear Register (ESRC) 2486
8.29. I²C Bus Control Set Register (IBCRS) 2487
8.30. Serial Mode Set Register (SMRS) 2489
8.31. Serial Status Set Register (SSRS) 2490
8.32. Serial Auxiliary Control Status Set Register (SACSRS) 2491

8.33. FIFO Control Set Register 1 (FCR1S) 2492
8.34. FIFO Control Set Register 0 (FCR0S) 2493
CHAPTER 38: Base Timer 2495
1. Overview of the Base Timer 2496
2. Block Diagrams of the Base Timer 2498
3. Operations of the Base Timer 2502
4. 32-Bit Mode Operation 2504
5. Debug Mode 2506
6. Interrupts from the Base Timer 2507
7. Start of DMA Controller (DMAC) 2508
8. Registers of the Base Timer 2509
9. Notes on Using the Base Timer 2511
10. Base Timer Description by Function Mode 2513
10.1. PWM Timer Function 2514
10.1.1. 16-Bit PWM Timer Operation 2515
10.1.2. One-Shot Operation 2516
10.1.3. Start Delay Function 2517
10.1.4. ADC Trigger Output Timing 2520
10.1.5. External Timer Match Starting 2520
10.1.6. Interrupt Factors and Timing Chart 2521
10.1.7. Output Waveform 2521
10.1.8. PWM Timer Operation Flow 2523
10.1.9. Timer Control Registers (TMCR, TMCR2), Status Control Register (STC), Status Control Clear Register (STCC), and Status Control Set Register (STCS) 2524
10.1.10. PWM Cycle Setting Register (PCSR) 2534
10.1.11. PWM Duty Setting Register (PDUT) 2535
10.1.12. Timer Register (TMR) 2536
10.1.13. Start Delay Value Setting Register (PSDR) 2537
10.1.14. ADC Trigger Value Setting Register (ADTR) 2538
10.1.15. Base Timer Debug Register (BT_DEBUG) 2539
10.1.16. External Timer Compare Data Register (ETCDR) 2540
10.2. PPG Timer Function 2541
10.2.1. 16-Bit PPG Timer Operation 2542
10.2.2. Continuous Operation 2543
10.2.3. One-Shot Operation 2544
10.2.4. External Timer Match Starting 2546
10.2.5. Interrupt Factors and Timing Chart 2547
10.2.6. PPG Timer Operation Flow 2548
10.2.7. Timer Control Registers (TMCR, TMCR2), Status Control Register (STC), Status Control Clear Register (STCC), and Status Control Set Register (STCS) 2549
10.2.8. Low Width Setting Reload Register (PRLL) 2559
10.2.9. High Width Setting Reload Register (PRLH) 2560
10.2.10. Timer Register (TMR) 2561
10.2.11. Base Timer Debug Register (BT_DEBUG) 2562
10.2.12. External Timer Compare Data Register (ETCDR) 2563
10.3. Reload Timer Function 2564
10.3.1. Operations of 16-Bit Reload Timer 2565

10.3.2. External Timer Match Starting 2568
10.3.3. Reload Timer Operation Flow 2569
10.3.4. Timer Control Registers (TMCR, TMCR2), Status Control Register (STC), Status Control Clear Register (STCC), and Status Control Set Register (STCS) 2570
10.3.5. Cycle Setting Register (PCSR) 2580
10.3.6. Timer Register (TMR) 2581
10.3.7. Base Timer Debug Register (BT_DEBUG) 2582
10.3.8. External Timer Compare Data Register (ETCDR) 2583
10.4. PWC Timer Function 2584
10.4.1. Operations of PWC Timer 2585
10.4.2. Timer Control Registers (TMCR, TMCR2), Status Control Register (STC), Status Control Clear Register (STCC), and Status Control Set Register (STCS) 2593
10.4.3. Data Buffer Register (DTBF) 2602
10.4.4. Base Timer Debug Register (BT_DEBUG) 2603
CHAPTER 39: Base Timer I/O Selection Function 2605
1. Overview 2606
2. Configuration 2607
3. Explanation of Operation 2608
4. Registers 2611
4.1. I/O Selection Registers (BT_BTSEL01, BT_BTSEL23, BT_BTSEL45, BT_BTSEL67, BT_BTSEL_89, BT_BTSEL1011) 2612
4.2. I/O Selection Registers (BT_BTSEL1213, BT_BTSEL1415, BT_BTSEL1617, BT_BTSEL1819, BT_BTSEL_2021, BT_BTSEL2223) 2613
4.3. I/O Selection Registers (BT_BTSEL2425, BT_BTSEL2627, BT_BTSEL2829) 2614
4.4. Simultaneous Soft Start Register (BT_BTSSSR0) 2615
4.5. Simultaneous Soft Start Register (BT_BTSSSR12) 2616
4.6. Simultaneous Soft Start Register (BT_BTSSSR24) 2617
CHAPTER 40: Base Timer Simultaneous Operation 2619
1. Overview 2620
2. Configuration 2621
3. Explanation of Simultaneous Soft Start Operation 2622
3.1. Global Time and Base Timers Operation Waveform 2623
3.2. Global Timer Match Starting Channel Buffer Operation by MODE Set 2627
4. Overview of the 16-bit Global Timer 2632
4.1. Configuration Diagram of the 16-bit Global Timer 2633
5. Explanation of the 16-bit Global Timer Operation 2634
5.1. Interrupts of the 16-bit Global Timer 2637
5.2. Setting Procedure Example of the 16-bit Global Timer 2638
5.3. DEBUG MODE 2639
6. Registers of the 16-bit Global Timer 2640
6.1. Global Timer Compare Clear Register (GT_CPCLR) 2642
6.2. Global Timer Data Register (GT_TCDT) 2643
6.3. Global Timer State Control Register (GT_TCCS) 2644
6.4. Global Timer State Control Clear Register (GT_TCCSC) 2648
6.5. Global Timer State Control Set Register (GT_TCCSS) 2650
6.6. Global Timer Debug Register (GT_DEBUG) 2652
6.7. Global Timer Match Starting Control Register (GT_MSCTR) 2653

6.8. Global Timer Match Starting Channel Buffer Register (GT_MSCHB) 2656
6.9. Global Timer Match Starting Channel Register (GT_MSCH) 2658
7. Precautions for Using This Device 2659
CHAPTER 41: 32-Bit Free-Run Timer 2661
1. Overview of the 32-Bit Free-Run Timer 2662
2. Block Diagram of the 32-Bit Free-Run Timer 2663
3. Operation of the 32-Bit Free-Run Timer 2664
3.1. Interrupts of the 32-Bit Free-Run Timer 2671
3.2. Setting Procedure Example of the 32-Bit Free-Run Timer 2672
3.3. Debug Mode 2673
4. Registers of the 32-Bit Free-Run Timer 2674
4.1. Compare Clear Buffer Register (CPCLRB)/Compare Clear Register (CPCLR) 2675
4.2. Timer Data Register (TCDT) 2677
4.3. Timer State Control Register (TCCS) 2679
4.4. Timer Extended State Control Register (TECCS) 2685
4.5. Timer State Control Clear Register (TCCSC) 2688
4.6. Timer State Control Set Register (TCCSS) 2692
4.7. Free-Run Timer Debug Register(FRT_DEBUG) 2695
5. Precautions for Using This Device 2696
CHAPTER 42: 32-Bit Input Capture 2697
1. Overview of the 32-Bit Input Capture 2698
2. Explanation of 32-Bit Input Capture Operation 2700
2.1. Interrupt of the 32-Bit Input Capture 2701
2.2. Setting Procedure Example 2702
3. Registers of the 32-Bit Input Capture 2703
3.1. Input Capture Data Registers 0, 1 (IPCP0, IPCP1) 2705
3.2. Input Capture State Control Register (ICS) 2707
3.3. Input Capture State Control Clear Register (ICSC) 2711
3.4. Input Capture State Control Set Register (ICSS) 2713
4. Precautions for Using This Device 2715
CHAPTER 43: 32-Bit Output Compare 2717
1. Overview 2718
2. Configuration 2719
3. Explanation of Operation 2720
3.1. Interrupt 2720
3.2. 32-Bit Output Compare Operation 2720
4. Setting Procedure Example 2734
5. Registers 2735
5.1. Output Compare Buffer Register (OCCPB0, 1)/Output Compare Register (OCCP0, 1) 2738
5.2. Compare Control Register (OCS) 2741
5.3. Compare Control Clear Register (OCSC) 2748
5.4. Compare Control Set Register (OCSS) 2753
6. Precautions for Using This Device 2758
CHAPTER 44: 32-bit Reload Timer 2759
1. Overview 2760
2. Configuration and Block Diagram 2761
3. Operation of the 32-bit Reload Timer 2762
3.1. Internal Clock and External Event Counter Operations of 32-bit Reload Timer 2763
3.2. Underflow Operation of 32-bit Reload Timer 2765

3.3. Output Functions of 32-bit Reload Timer 2767
3.4. Counter Operation state 2768
3.5. DMA Operation 2769
4. Registers 2770
4.1. DMA Configuration Register(RLTn_DMACFG) 2771
4.2. Timer Control Status Register (RLTn_TMCSR) 2772
4.3. 32-bit Reload Register (RLTn_TMRLR) 2776
4.4. 32-bit Timer Register (RLTn_TMR) 2777
CHAPTER 45: Reload Timer Simultaneous Soft Start 2779
1. Overview 2780
2. Configuration 2781
3. Explanation of Operation 2782
4. Registers 2783
4.1. Reload Timer Trigger Selection Register (RLT0_TSEL, RLT1_TSEL, RLT2_TSEL) 2784
4.2. Reload Timer Simultaneous Soft Start Register (RLT0_SSSR, RLT1_SSSR, RLT2_SSSR) 2785
CHAPTER 46: Resource-linked Operation 2787
1. Overview 2788
2. Configuration with Free-Run Timer 2789
3. Configuration of Free-Run Timer Selector 2790
4. Output Compare Control Configuration 2791
5. Cascade Input of Reload Timer 2792
6. Registers 2793
7. Operation to Select Free-run Timer Output 2796
7.1. Selection Setting with Input Capture 2796
7.2. Selection Setting with Output Compare 2797
8. Output Compare Control Register Settings 2798
9. Cascade Input of Reload Timer Settings 2799
10. Precautions on Operation Specifications 2800
10.1. Precautions on Using the Free-Run Timer Selector 2800
10.2. Precautions on Using OCU 2800
10.3. Precautions on Using RLT 2800
CHAPTER 47: Quad Position & Revolution Counter 2801
1. Overview 2802
2. Configuration 2803
3. Operations 2804
4. Registers 2816
4.1. Quad Position & Revolution Counter Position Count Register (QPCR) 2817
4.2. QPRC Revolution Count Register (QRCR) 2819
4.3. QPRC Position Counter Compare Register (QPCCR) 2820
4.4. QPRC Position and Revolution Counter Compare Register (QPRCR) 2821
4.5. QPRC Control Register (QCR) 2822
4.6. QPRC Extension Control Register (QECR) 2827
4.7. Low-Order Bytes of QPRC Interrupt Control Register (QICRL) 2829
4.8. High-Order Bytes of QPRC Interrupt Control Register (QICRH) 2833
4.9. QPRC Maximum Position Register(QMPR) 2837
CHAPTER 48: External Interrupt Capture 2839
1. Overview 2840
2. Configuration 2841

3. Explanation of Operation 2842
4. Setting Procedure Examples 2845
5. Registers 2848
5.1. Configuration Register (EICUn_CNFGR) 2849
5.2. External Interrupt Pin Enable Register (EICUn_IRENR) 2852
5.3. Sample Registers 0 to 7 (EICUn_SPLR0 to 7) 2853
CHAPTER 49: CRC (Cyclic Redundancy Check) 2855
1. Overview 2856
2. Operations 2857
2.1. Calculation Sequence 2858
2.2. Use Examples 2859
3. Registers 2864
3.1. CRC Control Register (CRC00_CRCCR, CRC01_CRCCR, CRC02_CRCCR, CRC03_CRCCR) 2865
3.2. Initial Value Register (CRC00_CRCINIT, CRC01_CRCINIT, CRC02_CRCINIT, CRC03_CRCINIT) 2868
3.3. Input Data Register (CRC00_CRCIN, CRC01_CRCIN, CRC02_CRCIN, CRC03_CRCIN) 2869
3.4. CRC Register (CRC00_CRCR, CRC01_CRCR, CRC02_CRCR, CRC03_CRCR) 2870
CHAPTER 50: I/O Port 2871
1. Overview 2872
2. Configuration and Block Diagrams 2873
3. Setting Procedure Examples 2875
4. Register List 2885
4.1. Data Direction Register (GPIO_DDRi) (i = 0 to 4) 2886
4.2. Data Direction Set Register (GPIO_DDSRi) (i = 0 to 4) 2887
4.3. Data Direction Clear Register (GPIO_DDCRi) (i = 0 to 4) 2888
4.4. Port Output Data Register (GPIO_PODRi) (i = 0 to 4) 2889
4.5. Port Output Set Register (GPIO_POSRi) (i = 0 to 4) 2890
4.6. Port Output Clear Register (GPIO_POCRi) (i = 0 to 4) 2891
4.7. Port Input Enable Register (GPIO_PORTEN) 2892
4.8. Port Input Data Register (GPIO_PIDRi) (i = 0 to 4) 2893
4.9. GPIO Key Code Register (GPIO_KEYCDR) 2894
4.10. Port Setting Register (PPC_PCFGRijj) (i = 0 to 4, jj = 00 to 31) 2896
4.11. PPC Key Code Register (PPC_KEYCDR) 2900
4.12. Resource Input Setting Register (RIC_RESINn) (n = 0 to 367) 2902
4.13. RIC Key Code Register (RIC_KEYCDR) 2904
5. Precautions for Using This Device 2906
5.1. External Pin setting priority 2907
5.2. Notes on switching the I/O port function 2908
CHAPTER 51: Peripheral Protection Unit 2909
1. Overview 2910
2. Configuration and Block Diagram 2911
3. Operation of the PPU 2912
3.1. Access Type and PPU Operation 2912
3.2. Access Protection to PPU Register 2913
3.3. Access Violation 2916
4. Registers 2917
4.1. PPU Control Register (PPU0_CR) 2918

4.2. PPU Status Register (PPU0_SR) 2920
4.3. PPU Unlock Register (PPU0_UNLOCK) 2923
4.4. PPU Privileged Read Attribute Register (PPU0_PPRi) 2924
4.5. PPU User Read Attribute Register (PPU0_PURi) 2925
4.6. PPU Privileged Write or Access Attribute Register (PPU0_PPWAi) 2926
4.7. PPU User Write or Access Attribute Register (PPU0_PUWAi) 2927
4.8. PPU Lock Register (PPU0_LOCKi) 2928
5. Setup Steps of PPU Register 2929
CHAPTER 52: DDR High Speed SPI Controller 2931
1. Overview 2932
2. Configuration 2934
3. Serial Interface 2936
3.1. Serial Clock Modes and Data Rate Modes 2936
3.2. SPI Clock Frequency 2941
3.3. SPI Input Data Sampling Point 2941
3.4. SPI Data Learning Pattern 2943
3.5. SPI Data Protocol 2943
3.6. Shift Direction 2944
4. Operations of the DDRHSSPI 2952
4.1. Direct Mode 2952
4.1.1. Internal FIFOs 2952
4.1.2. FIFO Size 2953
4.1.3. TX-FIFO Control 2953
4.1.4. Available Command Formats of Serial Flash Memory 2958
4.1.5. FIFO Accesses in Direct Mode 2960
4.1.6. Service Requests 2961
4.1.7. SPI Transfers 2964
4.2. Command Sequencer Mode 2966
4.3. Address Map of DDRHSSPI 2976
4.4. General Use Case Guidelines for DDRHSSPI 2977
4.4.1. Notice On Document Designations 2985
4.4.2. Advance Information 2985
5. Registers 2986
5.1. DDRHSSPI Module Control Register (DDRHSSPIn_MCTRL) 2988
5.2. DDRHSSPI Peripheral Communication Configuration Registers (DDRHSSPIn_PCC0-3) 2990
5.3. DDRHSSPI TX Interrupt Flag Register (DDRHSSPIn_TXF) 2992
5.4. DDRHSSPI TX Interrupt Enable Register (DDRHSSPIn_TXE) 2994
5.5. DDRHSSPI TX Interrupt Clear Register (DDRHSSPIn_TXC) 2996
5.6. DDRHSSPI RX Interrupt Flag Register (DDRHSSPIn_RXF) 2998
5.7. DDRHSSPI RX Interrupt Enable Register (DDRHSSPIn_RXE) 3001
5.8. DDRHSSPI RX Interrupt Clear Register (DDRHSSPIn_RXC) 3003
5.9. DDRHSSPI Fault Status Flag Register (DDRHSSPIn_FAULTF) 3005
5.10. DDRHSSPI Fault Status Clear Register (DDRHSSPIn_FAULTC) 3008
5.11. DDRHSSPI Direct Mode Configuration Register (DDRHSSPIn_DMCFG) 3010
5.12. DDRHSSPI DMA Enable Register (DDRHSSPIn_DMAEN) 3011
5.13. DDRHSSPI Direct Mode Start Register (DDRHSSPIn_DMSTART) 3012
5.14. DDRHSSPI Direct Mode Peripheral Select Register (DDRHSSPIn_DMPSEL) 3013
5.15. DDRHSSPI Direct Mode Transfer Protocol Register (DDRHSSPIn_DMTRP) 3014

5.16. DDRHSSPI Byte Count Control Register (DDRHSSPIn_DMBCC) 3015
5.17. DDRHSSPI Byte Count Status Register (DDRHSSPIn_DMBCS) 3016
5.18. DDRHSSPI Direct Mode FIFO Status Register (DDRHSSPIn_DMFIFOSTATUS) 3017
5.19. DDRHSSPI Direct Mode FIFO Configuration Register (DDRHSSPIn_DMFIFOCFG) 3018
5.20. DDRHSSPI TX-FIFO Registers (DDRHSSPIn_TXFIFO0-23) 3020
5.21. DDRHSSPI RX-FIFO Registers (DDRHSSPIn_RXFIFO0-23) 3022
5.22. DDRHSSPI Read Command Sequencer Data/Control Registers (DDRHSSPIn_RDCSDC0-11) 3024
5.23. DDRHSSPI Module ID Register (DDRHSSPIn_MID) 3026
5.24. DDRHSSPI Command Sequencer Prefetch Address Register (DDRHSSPIn_CSPREFETCHADDR) 3027
5.25. DDRHSSPI SDATA Center Clock Sample Point Registers (DDRHSSPIn_SDATASAMPLEPTCNT0-7) 3028
5.26. DDRHSSPI SDATA Left Clock Sample Point Registers (DDRHSSPIn_SDATASAMPLEPTLFT0-7) 3029
5.27. DDRHSSPI SDATA Right Clock Sample Point Registers (DDRHSSPIn_SDATASAMPLEPTRGH0-7) 3030
5.28. DDRHSSPI Data Learning Pattern Register (DDRHSSPIn_DLP) 3031
5.29. DDRHSSPI Data Learning Pattern Sample Status Register (DDRHSSPIn_DLPSAMPLESTATUS) 3032
5.30. DDRHSSPI Command Sequencer Configuration Register (DDRHSSPIn_CSCFG) 3038
5.31. DDRHSSPI Command Sequencer Idle Time Register (DDRHSSPIn_CSITIME) 3040
5.32. DDRHSSPI Command Sequencer Address Extension Register (DDRHSSPIn_CSAEXT) 3041
5.33. DDRHSSPI Command Sequencer Prefetch Buffer Configuration Register (DDRHSSPIn_CSPBUFFERCFG) 3042
5.34. DDRHSSPI Command Sequencer Prefetch Buffer Status Register (DDRHSSPIn_CSPBUFFERSTATUS) 3043

CHAPTER 53: External Bus Interface 3045

1. Outline of External Bus Interface 3046
   1.1. Pad Interface 3047
   1.2. External Bus Interface and Memory Regions 3048
2. External Bus Interface Configuration Registers 3049
   2.1. Unlock Register (EBI_UNLOCK) 3052
   2.2. Lock Status Register (EBI_LSTSR) 3054
   2.3. SRAM/FLASH Mode Control Register (EBI_SFMR0~7) 3056
   2.4. SRAM/FLASH Access Configuration Register (EBI_SFACCR0~7) 3059
   2.5. SRAM/FLASH Address Control Register (EBI_SFADDCR0~7) 3063
   2.6. Error Register (EBI_ERRR) 3065
3. Operation of External Bus Interface 3067
   3.1. SRAM/FLASH Controller 3067
      3.1.1. Low-Speed Device Interface Function 3068
   3.2. NAND FLASH access 3069
   3.3. Endianess and Enabled byte Lanes 3072
4. External memory Connection to External Bus Interface 3075
5. Access Waveform Examples for External Bus Interface 3078

CHAPTER 54: System SRAM Module (SRAM_IF) 3083

1. Outline of SRAM_IF 3084
2. Operation of SRAM_IF 3085

3. SRAM_IF Register Set 3088
3.1. SRAM_IF Configuration Register 0 (SRCFG_CFG0) 3089
3.2. SRAM_IF Configuration Register 1 (SRCFG_CFG1) 3091
3.3. SRAM_IF Configuration Register 2 (SRCFG_CFG2) 3092
3.4. SRAM_IF Unlock/Lock Key Register (SRCFG_KEY) 3093
3.5. SRAM_IF Error Flag Register (SRCFG_ERRFLG) 3094
3.6. SRAM_IF Interrupt Enable Register (SRCFG_INTE) 3096
3.7. SRAM_IF ECC Enable Register (SRCFG_ECCE) 3097
3.8. SRAM_IF Error Address Register (SRCFG_ERRADR) 3098
3.9. SRAM_IF Module Identification Register (SRCFG_MID) 3100
4. How to Use SRAM_IF 3101
CHAPTER 55: Bit-Band Unit 3103
1. Overview 3104
2. Explanation of Operation 3105
CHAPTER 56: 12-Bit A/D Converter 3109
1. Overview 3110
2. Features 3111
2.1. A/D Activation Compare Function 3112
2.2. A/D Activation Arbitration Functions 3115
2.3. 12-Bit A/D Converter Control Functions 3116
3. Configuration 3117
4. Registers 3120
4.1. Registers of the Analog Input Control 3132
4.1.1. Key Code Register: KEYCDR 3133
4.1.2. Analog Input Control Register: ADER 3135
4.1.3. Failure Diagnosis Function Control Register: ADFDCTR0, ADFDCTR1 3137
4.2. A/D Activation Compare Register 3140
4.2.1. A/D Software Activation Register: ADTSS0, ADTSS1 3141
4.2.2. A/D Software Activation Channel Selection Register: ADTSE0, ADTSE1 3142
4.2.3. Compare Buffer Register: ADCOMPB0 to ADCOMPB63/Compare Register: ADCOMP0 to ADCOMP63 3144
4.2.4. A/D Activation Trigger Control Status Register: ADTCS0 to ADTCS63 3146
4.2.5. A/D Data Register: ADTCD0 to ADTCD63 3151
4.2.6. A/D Activation Trigger Extended Control Register: ADTECS0 to ADTECS633153
4.2.7. Upper Threshold Setting Register: ADRCUT0 to ADRCUT7 3156
4.2.8. Lower Threshold Setting Register: ADRCLT0 to ADRCLT7 3157
4.2.9. Range Comparison Control Status Register: ADRCCS0 to ADRCCS63 3158
4.2.10. Range Comparison Threshold Excess Flag Register: ADRCOT0, ADRCOT13161
4.2.11. Range Comparison Flag Register: ADRCIF0, ADRCIF1 3164
4.2.12. Scan Conversion Control Status Register: ADSCANS0, ADSCANS1 3167
4.2.13. Activation Channel Conversion Count Setting Register: ADNCS0 to ADNCS031 3169
4.2.14. Data Protection Status Flag Register: ADPRTF0, ADPRTF1 3170
4.2.15. Activation Channel Conversion Completion Flag Register: ADEOCF0, ADEOCF1 3172
4.2.16. A/D Activation Trigger Control Status Clear Register: ADTCSC0 to ADTCSC63 3176
4.2.17. Range Comparison Status Register: ADRCS0 to ADRCS63 3177
4.2.18. Range Comparison Flag Clear Register: ADRCIFC0, ADRCIFC1 3179

4.2.19. Scan Conversion Control Status Clear Register: ADSCANSC0, ADSCANSC1 3181
4.3. Registers of the 12-Bit A/D Converter Control 3182
4.3.1. A/D Control Status Register: ADCS0, ADCS1 3183
4.3.2. A/D Channel Status Register: ADCH 3184
4.3.3. A/D Mode Setting Register: ADMD 3185
4.3.4. Sampling Time Setting per A/D Channel Register: ADSTPCS 3187
5. Explanation of Operation 3189
5.1. Interrupts From A/D Activation Compare 3190
5.1.1. A/D Conversion End Interrupt 3191
5.1.2. Conversion Count-Specified Scan Conversion Completion Interrupt 3192
5.1.3. Range Comparison Interrupt 3193
5.2. Operation of the A/D Activation Compare 3194
5.2.1. A/D Activation 3195
5.2.2. Enabling A/D Activation 3196
5.2.3. Free-Run Timer Input 3197
5.2.4. Analog Channel Selection 3201
5.2.5. Software Activation 3202
5.2.6. Reload Timer Activation 3203
5.2.7. Compare Match Activation 3204
5.2.8. PWM Timer Activation 3212
5.2.9. Activation Request Modes 3235
5.2.10. A/D Conversion Data 3236
5.2.11. Protection Function 3237
5.2.12. About the Scan Conversion Mode 3238
5.2.13. About High-Priority Activation Request Operation for Another Activation Channel during Scan Conversion 3246
5.2.14. Forcible Termination of an Activation Request 3248
5.2.15. Range Comparison Function 3249
5.3. Operation of the A/D Activation Arbitration 3255
5.3.1. Arbitration for an A/D Activation Trigger 3256
5.3.2. Analog Channel Selection 3257
5.3.3. A/D Conversion Cancellation Function 3257
5.4. Operation of the 12-Bit A/D Converter 3258
5.4.1. Operation Timing 3259
5.4.2. About Activation Factors 3260
5.4.3. About A/D Conversion 3261
5.4.4. About Reactivation 3262
5.4.5. About A/D Conversion Cancellation 3263
5.4.6. Analog Channel Selection Control 3264
5.4.7. About A/D Conversion Time 3265
5.4.8. A/D Conversion End and A/D Data Capture 3266
5.4.9. Power Down 3267
5.5. Failure Diagnosis Function of the 12-Bit A/D Converter 3268
5.5.1. AVRH and AVRL Voltage Conversion Diagnosis 3269
6. Notes 3272
CHAPTER 57: LCD Controller 3275
1. Overview 3276
2. Features 3277

3. Configuration 3278
4. Operation 3279
4.1. Operation of LCD Controller/Driver (LCDC) 3280
4.2. 1/2 Duty Output Waveform 3282
4.3. 1/3 Duty Output Waveform 3285
4.4. 1/4 Duty Output Waveform 3288
4.5. Static Drive Output Waveform 3291
5. Setting 3294
6. Registers 3296
6.1. LCD Control Register 0: LCR0 3297
6.2. Data Memory for Display: VRAM 3299
6.3. LCDC Control Register 1: LCR1 3301
6.4. Common Pin Switching Register: LCDCMR 3302
6.5. LCDC Static Control Register: LCRS 3303
6.6. Static LCD Display Data Register: LDR 3305
6.7. Key Code Register: LCD_KEYCDR 3307
6.8. Segment Output Register: SEGER 3310
6.9. Common Output V Pin Control Register: COMVER 3311
7. Q&A 3313
7.1. How Can I Set Pins to COM Output Pins or SEG Output Pins? 3314
7.2. How to Set VRAM? 3316
7.3. How Can I Setting the Frame Cycle? 3317
7.4. How Can I Set the Bias? 3318
7.5. How Can I Set the Duty? 3319
7.6. How Can I Control the LCD Operation Start/Stop? 3320
7.7. How Can I Execute/Cancel the Display? 3321
7.8. How Can I Display during the PSS Timer Mode (Main Oscillation Operation/ Sub Oscillation Operation)? 3322
7.9. How Can I Select Either Internal or External for the Division Resistor? 3323
7.10. How Can I Select Pin of V3 Voltage? 3324
7.11. How Can I Select Either Internal or External for the Division Resistor? 3325
7.12. How Can I Adjust the Brightness When the Internal Division Resistor is Used? 3327
7.13. How Can I Block the Current with the External Division Resistor When the LCD Stops? 3328
7.14. How Can I Display/Non-Display the LCD with Static Drive (ST0 to ST8)? 3329
8. Sample Program 3330
9. Notes 3337
CHAPTER 58: Sound Generator 3339
1. Overview 3340
2. Configuration 3342
3. Operations 3343
3.1. Relation between the Amplitude Data Register (SGAR) and PWM Pulse 3345
3.2. Relation between the Frequency Data Register (SGFR) and Tone Pulse Signal 3346
3.3. Relation between the PWM Cycle Data Register (SGPCR) and PWM cycle 3347
3.4. Relation between the DMA Transfer Update Enable Register (SGDER) and DMA Settings 3348
3.4.1. The Number of DMA Transfers 3348
3.4.2. DMA Transfer size 3348
3.4.3. Transfer Byte Position in the DMA Transfer Intermediate Register (SGDMAR) 3348

3.4.4. DMA Transfer Image 3351
3.5. Sound Generator Operation 3353
3.5.1. Single operation of Sound Generator by MCU 3355
3.6. Continuous operation of Sound Generator by MCU 3356
3.7. Sound Generator operation with DMA 3359
3.7.1. In Case of DMA Transfer with 4-Byte size x 2 is Made N Times 3360
3.7.2. In Case of DMA Transfer with 2-Byte size x 2 is Made N Times 3363
3.7.3. In Case of DMA Transfer with 1-Byte size x 1 is Made N Times 3366
3.7.4. In Case of DMA Transfer with 4-Byte size x 2 is Made N Times and DMA Transfer with 2-Byte size x 1 is Made M Times (The Case the Transfer Byte size is Changed While Sound Outputting) 3369
3.7.5. In Case of DMA Transfer with 4-Byte size x 2 is made N Times and DMA Transfer with 4-Byte size x 2 is Made M Times (The Case the Transfer Byte size and the Increase/Decrease setting is Changed While Sound Outputting) 3373
3.8. Interrupt of Sound Generator 3377
4. Registers 3378
4.1. DMA Transfer Update Enable Register (SGDER) 3379
4.2. Sound Control Register (SGCR) 3381
4.3. Amplitude Data Register (SGAR) 3386
4.4. Frequency Data Register (SGFR) 3387
4.5. Tone Output Number Register (SGNR) 3388
4.6. Time Cycle Register (SGTCR) 3389
4.7. Increase and Decrease Data Register (SGIDR) 3390
4.8. PWM Cycle Data Register (SGPCR) 3391
4.9. DMA Transfer Intermediate Register (SGDMAR) 3392
4.10. Interrupt Clear Register (SGCCR) 3394
CHAPTER 59: Stepper Motor Controller 3395
1. Overview 3396
2. Configuration 3397
3. Operation of the Stepper Motor Controller 3399
3.1. Operation of PWM Pulse Generator 3400
3.2. Selection of Motor Drive Signals 3404
3.3. Synchronization System of Stepper Motor Controller 3405
3.4. Operation of the SMC Trigger Delay Function 3406
4. Registers 3409
4.1. PWM Control Register (PWC) 3410
4.2. PWM1 and PWM2 Compare Registers (PWC1, PWC2) 3413
4.2.1. PWM1 Compare Register(PWC1) 3413
4.2.2. PWM2 Compare Register(PWC2) 3414
4.3. PWM Selection Register (PWS) 3415
4.4. PWM Selection Set Register (PWSS) 3418
4.5. SMC Trigger Delay Register (PTRGDL) 3419
CHAPTER 60: Trigger Configuration of Stepper Motor Controller 3421
1. Overview 3422
2. Configuration and Block Diagram 3423
3. Operation 3424
4. Registers 3425
4.1. SMC Trigger Selection Register (SMCTGg_PTRGS) 3426

4.2. SMC Trigger Register (SMCTGg_PTRG) 3429
CHAPTER 61: Partial Wakeup Control 3431
1. Overview 3432
2. Configuration 3433
3. Explanation of Operation 3437
3.1. State Transition Diagram 3437
3.2. Flowchart 3438
3.3. Timing charts 3451
4. Registers 3454
4.1. PWU Control Register (PWUC) 3455
5. Precautions for Using This Device 3456
CHAPTER 62: APPENDIX: Various Settings of I/O Ports 3459
1. Input Level Setting 3460
2. Output Drive Capacity Setting 3461
3. Output Resource Selection 3462
4. Resource Input Selection 3468
CHAPTER 63: APPENDIX: State Transition 3503
1. Overview 3504
2. Diagram of State Transition 3505
3. Fetching the Operation Mode 3507
4. Changes to PSS and RUN 3510
CHAPTER 64: APPENDIX: I/O Map 3513
1. I/O MAP 3514
2. I/O Map (S6J3120 Series) 3515
3. APPS#5 3639
CHAPTER 65: APPENDIX: DMA Channel Activation Factors 3663
1. DMA Channel Activation Factors (S6J312xHzC*) 3664
CHAPTER 66: APPENDIX: List of Interrupt Vector and NMI Map 3667
1. List of Interrupt Vector (S6J312xHzC*) 3668
2. NMI Map (S6J312xHzC*) 3675
CHAPTER 67: APPENDIX: Major Changes 3677

Introduction: Restrictions on Operation Specifications

This introduction explains restrictions on the specifications of this model that are described in each chapter.

1. About Reset
2. About Clock System Specifications
3. About Low-Power Consumption Specifications
4. About Low-Voltage Detection Specifications
5. About the Clock Supervisor
6. About External Interrupts
7. About the Real Time Clock
8. About the BootROM Hardware Interface
9. About the Multi-Function Serial Interface
10. About the Base Timer
11. About the Base Timer I/O Selection Function
12. About the 32-Bit Free-Run Timer
13. About the 32-Bit Input Capture
14. About the 32-Bit Output Compare
15. About the External Interrupt Capture
16. About TCFLASH
17. About the Hardware Watchdog Timer
18. About CAN FD Message RAM ECC Function
19. About CAN-FD
20. About the 32-Bit Reload Timer
21. About the Reload Timer Simultaneous Soft Start
22. About the Stepper Motor Controller
23. About the Trigger Configuration of Stepper Motor Controller
24. About the External Bus Interface
25. About the Sound Generator
26. About the Quad Position & Revolution Counter
27. About the DDR High Speed SPI Controller
28. About the LCD Controller
29. About DMA Complex Subsystem
30. About the 12-Bit A/D Converter
31. About the Secure Hardware Extension (SHE)
32. About the Partial Wakeup Control
33. Common items

1. About Reset

This section explains restrictions on the operation specifications for reset.

- The following reset factors do not exist for this model.
  - Extended internal power supply low-voltage detection reset (LVDL2R)
  - Extended external power supply low-voltage detection reset (LVDH2R)
  - Sub clock supervisor reset (CSVSOR)nSRST pin input reset (SRSTX)
  - nSRST pin input reset (SRSTX)

  Register functions related to the items above are also not supported.

- This model does not have any INITX input pins implemented. A reset factor for INITX pin input can be generated by inputting an "L" level signal to the MD and RSTX pins simultaneously.

- This model does not support the following functions.
  - SSCG PLL1 to 3
    - PLL1 to 3
    - 3 V I/O (RSTX_IO_3V)
    - 3/5V I/O (RSTX_IO_35V)
    - External 1.2 V feed circuit

    Register functions related to the items above are also not supported.

- Implemented power domains are PD2, PD3, PD4_0, PD4_1, and PD6_0. PD5_0 to PD5_3 and PD6_1 are not implemented.

- Be sure to configure the same register settings for PD2 and PD3.

2. About Clock System Specifications

This section explains restrictions on the operation specifications for the clock system.

- This model does not support the following clocks.
  - Sub clock
  - PLL1 clock
  - PLL2 clock
  - PLL3 clock
  - SSCG PLL1 clock
  - SSCG PLL2 clock
  - SSCG PLL3 clock
  - External clock output

- For the following clock domains, to reduce current consumption, disable the maximum division setting and the clock oscillation of each supply clock in the Fast-CR oscillation operation state after power-on. For details on register settings, see the explanation on clock divider register n and clock source enable register n of the profile register group in "CHAPTER: LOW-POWER CONSUMPTION." These profile registers require control in RUN and PSS modes.
  - Clock domain 1 to 5
  - CLKO clock domain

- About PLL0 and SSCG PLL0 settings at the time of main clock 4 MHz input.
  The input frequency supplied to PLL must be divided by 1 (4 MHz).
  When CPUs operate at 128 MHz, usage by multiplying by 128 (PLLout=512 MHz) is recommended.
  When CPUs operate at 80 MHz, usage by multiplying by 120 (PLLout=480 MHz) is recommended.

  When SSCG PLL is used, use PLLout within a range between 400 MHz and 512 MHz
  The stabilization wait time of SSCG PLL must be set to 2^{10} cycles or greater when the main clock is 4 MHz

- 5.1.2 Main Clock Oscillation Control Register
  [bit31] MCMODE: Main clock amplifier oscillation mode bit. The initial value of this bit specifies High drive mode.
  To reduce noise, set the mode to low drive mode after transition to the user program.

- 5.1.3 PLL/SSCG stabilization time control register
  Setting the value 1000 to the PLLSTABS and SSCGSTABS bits is prohibited.

3. About Low-Power Consumption Specifications

This section explains restrictions on the operation specifications for low-power consumption.

- This model does not support the following functions.
  - Sub CLK
  - LVDL2 (extended internal low-voltage detection)
  - LVDH2 (extended external low-voltage detection)
  - SSCGPLL1 to 3 (SSCGPLLx=0 is used.)
  - PLL1 to 3 (PLLx=0 is used.)
  - 3 V I/O
  - External Power Supply Stabilization Time Setting(EXSTBCNT): Because there is no 1.2V external power supply

  Register functions related to the items above are also not supported.

- Implemented power domains are PD2, PD3, PD4_0, PD4_1, and PD6_0. PD5_0 to PD5_3 and PD6_1 are not implemented.
- Be sure to configure the same register settings for PD2 and PD3.
- The reload timer, CAN-FD, and MFS in MCU_CONFIG_GROUP are not implemented.
- This model does not support IRQ_PW bit in MCG_IRSR0 register.
  This bit is always read to 0.

4. About Low-Voltage Detection Specifications

This section explains restrictions on the operation specifications for low-voltage detection.

- Although 2 channels (ch.0 and ch.1) are described for low-voltage detection in a 1.2 V system power supply, this model has only 1 channel (ch.0). LVDL1 registers control ch.0.
- Low-voltage detection in a 3.3 V system power supply is not available.
- Although 2 channels (ch.0 and ch.1) are described for low-voltage detection in a 5V system power supply, this model has only 1 channel (ch.0). LVDH1 registers control ch.0.

5. About the Clock Supervisor

This section explains restrictions on the operation specifications for the clock supervisor.

- This model does not have the following functions.
  - Main clock supervisor 1
  - Sub clock supervisor 0
  - Sub clock supervisor 1
  - PLL1 clock supervisor
  - PLL2 clock supervisor
  - PLL3 clock supervisor
  - SSCG PLL1 clock supervisor
  - SSCG PLL2 clock supervisor
  - SSCG PLL3 clock supervisor
  - Function to output the abnormality detection bit to a port

  Register functions related to the items above are also not supported.

- The PLL Implementation channels are PLLm (m=0) and SSCG PLLn (n=0).

- This model does not have any INITX input pins implemented. A reset factor for INITX pin input can be generated by inputting an "L" level signal to the MD and RSTX pins simultaneously.

6. About External Interrupts

This section explains restrictions on the operation specifications for external interrupts.

- This model has 16 external interrupt input channels.
- DMA activation due to an external interrupt can be supported only for 1 channel (ch.0).

7. About the Real Time Clock

This section explains restrictions on the operation specifications for the real time clock.

- This model does not support sub oscillation (Sub CLK).

8. About the BootROM Hardware Interface

This section explains restrictions on the operation specifications for the BootROM hardware interface.

- Reading from a read prohibition area of BootROM (FFFF_00C0$_H$ to FFFF_037F$_H$) causes a bus error.

9. About the Multi-Function Serial Interface

This section explains restrictions on the operation specifications for the multi-function serial interface.

- This model supports the I^2C communication function.
  This model does not support high-speed mode. (Max. 400kbps)
  This model does not support Min. IOL = 3mA with VOL = 0.4V.
- At Multi-function serial ch.1, ch.2, and ch.12, I2C-UART function is not supported.
- The synchronous transmission function is not supported for transmission operation synchronized with an external trigger.
- The maximum frequency of a peripheral clock is 32 MHz in UART (asynchronous serial interface) operation mode.
- Hardware flow control is not supported in UART (asynchronous serial interface) operation mode.
- The maximum frequency of a peripheral clock is 32 MHz in CSIO (clock synchronous serial interface) operation mode.
- The maximum frequency of a peripheral clock is 32 MHz in LIN interface (v2.1) (LIN communication control interface (v2.1)) operation mode.
- This model does not support Interrupt request to ICU.(LIN Sync field detection;LSYN)
- This model does not support external trigger to Timer function.
- This model does not support Wake Up Function.
- When reset is issued, the following ports become "L" output as the initial state.
  - SCS0_0 /SIN4_1 (assigned to P023)
  - SIN1_0 (assigned to P028)
  - SOT1_0 (assigned to P029)
  - SCS1_0 (assigned to P031)
  - SCK1_0 (assigned to P100)
  - SIN2_0 (assigned to P408)
  - SOT2_0 (assigned to P409)
  - SOT4_1 (assigned to P024)
  - SCS42_1 (assigned to P027)
  - SCS43_1 (assigned to P030)
  - SOT12_0 (assigned to P406)
  - SCK12_0 /SCK10_1 (assigned to P407)

  Take care of the above port while used as UART/CSIO/LIN.
  When reset is issued, the communication may be disturbed by "L" output.
  Regarding the initial state for each ports, refer to the data sheet "11.PIN STATUSES IN CPU STATUS".

10. About the Base Timer

This section explains restrictions on the operation specifications for the base timer.

- For the base timer, configuration diagrams for only 2 channels (ch.0 and ch.1) of 16 bits are shown. This model consists of a 2 channel x 15 configuration, and thus has 30 channels implemented.
- Each register is implemented for each channel number.

11. About the Base Timer I/O Selection Function

This section explains restrictions on the operation specifications for the base timer I/O selection function.

- Simultaneous software activation is supported for each of the following groups consisting of 12 channels each.
  - Simultaneous software activation exceeding a group is not supported.
  - Group consisting of base timer ch.0 to ch.11
  - Group consisting of base timer ch.12 to ch.23
  - Group consisting of base timer ch.24 to ch.29

12. About the 32-Bit Free-Run Timer

This section explains restrictions on the operation specifications for the 32-bit free-run timer.

- For the 32-bit free-run timer, a configuration diagram for only 1 channel is shown. This model has 6 channels implemented.
- Each register is implemented for each channel number.

(Example) Compare clear buffer register CPCLRB

FRT00_CPCLRB	FRT ch.0
FRT01_CPCLRB	FRT ch.1
FRT02_CPCLRB	FRT ch.2
FRT03_CPCLRB	FRT ch.3
FRT04_CPCLRB	FRT ch.4
FRT05_CPCLRB	FRT ch.5

13. About the 32-Bit Input Capture

This section explains restrictions on the operation specifications for the 32-bit input capture.

- For the 32-bit input capture, configuration diagrams for only 2 channels are shown.
  This model consists of a 2 channel x 6 configuration, and thus has 12 channels implemented.
- Each register is implemented for each channel number.

(Example) Input capture data register IPCP0

ICU00_IPCP0	ICU ch.0, ch.1
ICU01_IPCP0	ICU ch.2, ch.3
ICU02_IPCP0	ICU ch.4, ch.5
ICU03_IPCP0	ICU ch.6, ch.7
ICU04_IPCP0	ICU ch.8, ch.9
ICU05_IPCP0	ICU ch.10, ch.11

14. About the 32-Bit Output Compare

This section explains restrictions on the operation specifications for the 32-bit output compare.

- For the 32-bit output compare, configuration diagrams for only 2 channels are shown.
  This model consists of a 2 channel x 6 configuration, and thus has 12 channels implemented.
- Each register is implemented for each channel number.
- For the 32-bit output compare, the MODn signal that controls the OUT pin output level is described as an external pin. However, this control is performed by an internal register.

(Example) Output compare buffer register OCCPB0

OCU00_OCCPB0	OCU ch.0, ch.1
OCU01_OCCPB0	OCU ch.2, ch.3
OCU02_OCCPB0	OCU ch.4, ch.5
OCU03_OCCPB0	OCU ch.6, ch.7
OCU04_OCCPB0	OCU ch.8, ch.9
OCU05_OCCPB0	OCU ch.10, ch.11

15. About the External Interrupt Capture

This section explains restrictions on the operation specifications for the external interrupt capture.

- This model has 16 external interrupt input channels.

16. About TCFLASH

This section explains restrictions on the operation specifications for TCFLASH.

- This model supports TCFLASH memory sizes are as follows.

Part number	TCFlash size (Byte)
S6J312AHzC*	1024K + 64K
S6J3129HzC*	768K + 64K

* z: A / B

17. About the Hardware Watchdog Timer

This section explains restrictions on the operation specifications for the hardware watchdog timer.

- When this model is in PSS mode, the count operation of the hardware watchdog timer is stopped.
- This model does not have a function for outputting the watchdog counter monitor bit.

18. About CAN FD Message RAM ECC Function

This section explains restrictions on the operation specifications for About CAN FD Message RAM ECC Function.

- This model is implemented as setting with TM_CANFERR=L.

19. About CAN-FD

This section explains restrictions on the operation specifications for CAN-FD.

- The following function is not mounted.
  Time stamp function, external counter operation
- The "canfd_bclk" is used as bus clock for CAD-FD macro, and it is supplied by CLK_LCP.
- The "canfd_cclk" is the clock supplied by the CAN prescaler (it is described as "CAN Clock" in "CHAPTER: CAN PRESCALER").
- That the number of words that can be used will be limited by the size of the actual Message RAM. Message RAM size of this model is 4096words.
- CAN-FD in MCU_CONFIG_GROUP is not implemented.

20. About the 32-Bit Reload Timer

This section explains restrictions on the operation specifications for the reload timer.

- For the reload timer, a configuration diagram for only 1 channel is shown. This model has 10 channels implemented.
- Each register is implemented for each channel number.

Common Peri #0	RLT ch.0, ch.1, ch.2, ch.3
Common Peri #1	RLT ch.16, ch.17, ch.18, ch.19
Common Peri #2	RLT ch.32, ch.33

21. About the Reload Timer Simultaneous Soft Start

This section explains restrictions on the operation specifications for the reload timer simultaneous soft start.

- Simultaneous software activation is supported for each of the following groups consisting of channel each. Simultaneous software activation exceeding a group is not supported.

Group consisting of base timer ch.0 to ch.3
Group consisting of base timer ch.16 to ch.19
Group consisting of base timer ch.32 to ch.33

22. About the Stepper Motor Controller

This section explains restrictions on the operation specifications for the stepper motor controller.

- For the stepper motor controller, a configuration diagram for only 1 channel is shown. This model has 4 channels implemented.
- Each register is implemented for each channel number.

SMC ch.0, ch.1, ch.2, ch.3

23. About the Trigger Configuration of Stepper Motor Controller

This section explains restrictions on the operation specifications for the trigger configuration of stepper motor controller.

- For the trigger configuration of stepper motor controller, a block diagram of the SMC module for 6 channels as SMC is shown.
  This model has 4 channels implemented as SMC.

24. About the External Bus Interface

This section explains restrictions on the operation specifications for the external bus interface.

- For the external bus interface, the following function is not supported.
  NAND-FLASH
- By function is not supported, the following terminals will be not support.

  MAD [24]
  MRDatb [31:16]
  MWDatb [31:16]
  MDQM [3:2]
  MCSX [7:4]
  MNALE
  MNCLE
  MNWEX
  MNREX

25. About the Sound Generator

This section explains restrictions on the operation specifications for the sound generator.

- For the sound generator, a configuration diagram for only 1 channel is shown.
  This model has 3 channels implemented.
- Each register is implemented for each channel number.

  SG ch.0, ch.1, ch.2

26. About the Quad Position & Revolution Counter

This section explains restrictions on the operation specifications for the quad position & revolution counter.

- For the Quad Position & Revolution Counter, a configuration diagram for only 1 channel is shown.
  This model has 2 channels implemented.
- Each register is implemented for each channel number.

  QPRC ch.8, ch.9

27. About the DDR High Speed SPI Controller

This section explains restrictions on the operation specifications for the DDR HS-SPI.

- This model is implemented as setting with DDRHSSPIn_MID.MID=0x00000001.

28. About the LCD Controller

This section explains restrictions on the operation specifications for the LCD Controller.

- For the LCD Controller, the following function is not supported.
  - Selection of sub-clock by the clock select bits
- When reset is issued, the following ports become "L" output as the initial state.
  - P019, P023, P024, P027, P028, P029, P030, P031, P100, P406, P407, P408, P409

  Therefore, don't add the Pull-up registers outside of this product to the above ports.
  In case that the above ports are used as Pull-up, use the Pull-up function implemented in this product.

29. About DMA Complex Subsystem

This section explains restrictions on the operation specifications for DMA Complex Subsystem.

- Although the output signals "TOT0 to 3" from the Reload Timer are described, this model has not the signals.
- Although the input signal "TIN" to the Reload Timer is described, the connection from the external pin to the "TIN" is not available.

30. About the 12-Bit A/D Converter

This section explains restrictions on the operation specifications for the 12-Bit A/D Converter.

- The following numbers are not supported as activation channels in this model.

  Channel No. 0, 1, 2, 4, 7, 8, 11, 16, 25, 26, 44, 45, 54, 63

31. About the Secure Hardware Extension (SHE)

This section explains restrictions on the operation specifications for the Secure Hardware Extension (SHE)

- Regarding S6J312xHBC*, The models are not supported SHE function.

* x: A/9

32. About the Partial Wakeup Control

This section explains restrictions on the operation specifications for Partial Wakeup Control

- This model does not support the following analog channels for partial wakeup.
  AN44(PWU_AN4), AN45(PWU_AN5)

33. Common Items

33.1. 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 byte/half Word/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

33.2. 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. 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 bit1 and you want to clear bit2 in SomeRegister.

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

[ex2] SomeRegister has "0 write clear bit" in bit1 and you want to set bit2 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 1: Overview

This chapter provides an overview.

1. **Overview**
2. **Features**
3. **Product Lineup**
4. **Function Overview**
5. **Block Diagrams**
6. **Memory Map**
7. **Abbreviations**
8. **Pin Assignment**
9. **Package Dimensions**
10. **List of Pin Functions**
11. **I/O Circuit Type**

1. Overview

This section provides an overview of the S6J3120 series.

The S6J3120 series is a set of 32-bit microcontrollers designed for in-vehicle use. It uses the ARM® Cortex®-R5 CPU as a CPU.

2. Features

This section explains the features of the S6J3120 series.

2.1. Cortex-R5
2.1. Peripheral Functions

2.1. Cortex-R5 Core

This section explains the Cortex-R5 CPU core.

- ARM Cortex-R5
- 32-bit ARM architecture
  - 2-instruction issuance super scalar
  - 8-stage pipeline
- ARMv7/Thumb®-2 instruction set
- MPU (memory protection) equipped
  - 16-area support
- ECC support for the TCM ports for RAM
  1-bit error correction and 2-bit error detection (SEC-DED)
- TCM ports
  2 TCM ports
  - ATCM port
  - BTCM port (B0TCM, B1TCM)
- Caches
  - Instruction cache 16 KB
  - Data cache 16 KB
- VIC port
  Low latency interrupt
- AXI master interface
  64-bit AXI interface (instruction/data access)
  32-bit AXI interface (I/O access)
- AXI slave interface
  64-bit AXI interface (TCM port access)
- ETM-R5 trace

2.2. Peripheral Functions

This section explains peripheral functions.

- Clock generation
  - Main clock oscillation (4 MHz)
  - No sub clock oscillation
  - CR oscillation (100 kHz)
  - CR oscillation (4 MHz)
- Built-in Flash memory size
  - Program: 1024 K + 64 KB (S6J312AHzC*)/768 K + 64 KB (S6J3129HzC*)
    *z: A/B
  - Work: 112 KB (S6J312AHzC*)/ 112 KB (S6J3129HzC*)
    *z: A/B
- Built-in RAM size
  - TCRAM 64 KB(S6J312AHzC*)/ 48 KB(S6J3129HzC*)
  - System SRAM 16 KB (S6J312AHzC*)/ 16 KB (S6J3129HzC*)
  - Backup RAM 8 KB (S6J312AHzC*)/ 8 KB (S6J3129HzC*)
    *z: A/B
- General-purpose ports: 112 channels (S6J312AHzC*)/ 112 channels (S6J3129HzC*)
  *z: A/B
- External bus interface
  - 24-bit address, 16bit data
- DMA controller
  - Up to 16 channels can be activated simultaneously.
- A/D converter (successive approximation type)
  - 12-bit resolution, 2 units mounted: Max 50 channels (22 channels + 28 channels)(S6J312AHzC*)/ Max 50 channels (22 channels + 28 channels)(S6J3129HzC*)
    *z: A/B
- External interrupt input: 16 channels
  - Level ("H"/"L") and edge (rising/falling) can be detected.
- Multi-function serial (transmission and reception FIFOs mounted) :Max 10 channels(S6J312AHzC*)/ Max 10 channels(S6J3129HzC*)
  *z: A/B

  <I^2C>
  - Full duplex, double buffering system; 64-byte transmission FIFO, 64-byte reception FIFO
  - Standard mode (Max. 100kbps) is supported only.
  - DMA transfer is supported.

  <UART (asynchronous serial interface) >
  - Full duplex, double buffering system; 64-byte transmission FIFO, 64-byte reception FIFO
  - Parity check can be enabled/disabled.
  - Built-in dedicated baud rate generator
  - An external clock can be used as a transfer clock.
  - Parity, frame, overrun error detection functions are available.
  - DMA transfer is supported.

  <CSIO (synchronous serial interface) >

- Full duplex, double buffering system; 64-byte transmission FIFO, 64-byte reception FIFO
- Support for SPI. Both master and slave roles are supported. Data length in bits can be set to a value from 5 to 16 or one of the values of 20, 24, and 32.
- Built-in dedicated baud rate generator (master operation)
- External clock input is enabled (slave operation).
- Overrun error detection function is available.
- DMA transfer is supported.
- Serial chip select SPI function

<LIN-UART (asynchronous serial interface for LIN) >

- Full duplex, double buffering system; 64-byte transmission FIFO, 64-byte reception FIFO
- Support for LIN protocol revision 2.1
- Both master and slave roles are supported.
- Framing error and overrun error detection
- LIN Synch break generation and detection, LIN Synch Delimiter generation
- Built-in dedicated baud rate generator
- The external clock can be adjusted by the reload counter.
- DMA transfer is supported.

■ CAN controller: CAN-FD Max 3 channel
- CAN-FD (V3.2.0)
- CAN transfer speed :5Mbps
- CAN Clock :Max 40MHz
- 192 message buffers/channel (reception message buffer size)
- 32 message buffer/channel (transmission message buffer size)

■ Base timer: MAX 30 channels
- 16bit Timer.
- It is selectable by 4 functions of the PWM/PPG/PWC/Reload Timer.
- 2-channel cascade connection enables operation as a 32-bit timer.(PWC and Reload Timer)

■ Reload timer: Max 10 channels
- 32bit Timer.

■ Free-run timer: Max 6 channels
- 32bit Timer.
- Main clock oscillation and CR oscillation are available.
- Free-run timer output can work in combination with an input capture and an output compare.

■ Input capture: Max 12 channels
- 32bit Timer.

■ Output compare: Max 12 channels
- 32bit Timer.

■ Sound generator : Max 3 channels
- Frequency and amplitude sequencers provided.

■ Stepping motor controller : Max 4 channels
- 8/10-bit PWM
- High current output supported (4 lines × 4 channels)
- Can refer back electromotive force using pin-shared A/D converter

■ LCD controller
- Common output : 4 , Segment output : 32
- Duty drive (SEG0 to SEG31) and static drive (ST0 to ST8) can be switched.

  - Each of COM0 to COM3, SEG0 to SEG31, V0, V1, V2, and V3 pins for duty drive can be switched to the general-purpose port. (The SEG23 to SEG31 pins can be switched to static driving.)
  - V0, V1, V2 and V3 pin can be used as the general-purpose port. But V3 pin cannot be used as an output pin.
  - Each of ST0 to ST8 pins for static drive can be switched to the general-purpose port, or it can be switched to the segment output of duty drive.
- Quad position & revolution counter(QPRC) : Max 2 channels
- Real time clock (RTC) (day/hour/minute/second)
  - Main clock oscillation or CR oscillation (100 kHz) can be selected as an operation clock.
- Calibration: Real time clock (RTC) driven by the CR clock
  - Correction can be done by configuring the prescaler of the real time clock based on the ratio between the main clock and the CR clock.
- Clock supervisor
  - Abnormality (such as damaged crystal) of the main clock oscillation (4 MHz) can be monitored.
  - The clock can switch to the CR clock when an abnormality is detected.
  - PLL abnormality can be detected.
- CRC generation
  - Fixed-length CRC
    - CCITT CRC16 generator polynomial: 0x1021
    - IEEE-802.3 CRC32 generator polynomial: 0x04C11DB7
- DDR HS-SPI
  - E^2PROM and the flash device of the Single/Dual/Quad-SPI protocol can be connected.
- Watchdog timer
  - Hardware watchdog
  - Software watchdog
- NMI
- I/O relocation
  - Peripheral function pin locations can be changed.
- Low-power consumption control
  - Standby function
  - Power-off function
  - Partial wakeup function
- Power-on reset
- Low-voltage detection reset
- Security
  - Flash security
  - Interface security (JTAG + test port)
  - SHE
  - Unique device ID
- Package: LEU144 (S6J312xHzC*)
  *x:A/9, z: A/B
- CMOS 55 nm technology
- Power supply
  - 5 V single power supply
  - The voltage step-down circuit generates internal 1.2 V from 5 V.

- 5 V power supply is used for I/O.

3. Product Lineup

This section explains the product lineup in the S6J3120 series.

Table 3-1 Memory Size

		S6J312AHzC*	S6J3129HzC*
Flash	Program	1024K bytes + Small sector (8 KB x 8)	768K bytes + Small sector (8 KB x 8)
	Work	112K bytes	112K bytes
RAM	TCRAM	64K bytes	48K bytes
	System SRAM	16K bytes	16K bytes
	Backup RAM	8K bytes	8K bytes

*z: A/B

Table 3-2 SHE Option

	S6J312xHAC*	S6J312xHBC*
Security (SHE)	ON	OFF

* x: A/9

Table 3-3: Product Lineup

	S6J312xHzC*[1]
CPU core	Coretex-R5
CMOS 55 nm technology	55 nm
Package	LEU144
Main clock	4 MHz
Built-in CR oscillator	100 kHz
	4 MHz
Maximum CPU operating frequency	128 MHz
Watchdog timer	1 channel (hardware) 1 channel (software)
Clock supervisor	YES
External power supply, low-voltage detection reset	YES
Internal power supply, low-voltage detection reset	YES
NMI request	YES
External interrupt	16 channels
DMA controller	16 channels
CAN-FD	3 channels (192 msg buffers/ch)
Multi-function serial	10 channels*[1]

	S6J312xHzC*1
A/D converter	12-bit (2 units) Unit 0 x 22 channels Unit 1 x 28 channels
Free-run timer	6 channels
Input capture	12 channels
Output compare	12 channels
Base timer (16-bit)	30 channels
Real time clock (RTC)	1 channel
CR clock calibration	YES
CRC generation	YES
Low-power consumption mode	Standby function Power-off function Partial wakeup function
SHE	YES
External BUS I/F	Address : 24-bit Data :16-bit
Reload Timer(32bit)	10 channels
Quad Position & Revolution Counter	2 channels
DDR HS-SPI	YES
LCD Controller	32seg x 4com(Static drive 8seg x 1com)
Sound Generator	3 channels
Stepping Motor Controller	4 channels
General-purpose port GPIO	112 channels
Power supply	5 V ± 10%
Operation assurance temperature (Ta)	-40 °C to +105 °C
On-chip debugger (JTAG)	YES

*1: x: A/9, z: A/B

*2: I^2C-UART function is not supported at Multi-function serial ch.1, ch.2, and ch.12.

4. Function Overview

This section provides a functional overview of this series.

Table 4-1: Functional Overview

Function	Features
CPU	ARM Cortex-R5 32-bit ARM architecture
Clock	Main clock oscillation 4 MHz PLL, SSCG PLL CR oscillator 100 kHz, 4 MHz
I/O Port	Programmable in units of bits as input/output or peripheral signal
External BUS I/F	Address : 24-bit Data :8-bit or 16-bit
Flash interface	1 wait (target) at operating frequency of 128 MHz * Time taken until data is acquired from the Flash memory is calculated by adding the access time for each connection bus to the above-described time.
DMA controller	16 channels
Free-run timer	32-bit up counter Free-run timer: 32-bit
Input capture	32-bit capture register that detects a rising or falling edge or both edges By the edge detection of the terminal input, the free-run timer takes in a counter level and produces IRQ
Output compare	Outputs an interrupt signal at the time of comparison with the 32-bit free-run timer
Reload timer	32bit Timer.
Base timer	16bit Timer. It is selectable by 4 functions of the PWM/PPG/PWC/Reload Timer. 2-channel cascade connection enables operation as a 32-bit timer.(PWC and Reload Timer)
Real time clock	Day/hour/minute/second register Main clock oscillation or built-in 100 kHz CR oscillation can be selected as an operation clock. Correction function for built-in 100 kHz CR oscillation clock - Monitors cycle errors of the built-in 100 kHz CR oscillation clock using the main clock. - Reflects the obtained error to the second counter setting value. An interrupt can be generated in units of 0.5 seconds, seconds, minutes, hours, or days.
External interrupt	16 channels, independent An interrupt factor can be selected from rising edge, falling edge, "L" level, or "H" level. Edge input detection at the time of recovery from standby is supported.
A/D converter	Built-in A/D converter with 12-bit resolution 2 units

Function	Features
Multi-function serial	4 functions are available: UART, CSIO, LIN-UART and I²C. 64-byte transmission FIFO and 64-byte reception FIFO are mounted. Reception interrupt factor (3 types) - Reception error detection (parity, overrun, or frame error) - It is detected that data received by a FIFO has reached the set threshold. - An idle period longer than 8 baud rate clocks is detected when reception data has not reached the threshold set for the FIFO. Transmission interrupt factor (2 types) - No transmission - Transmission FIFO is empty (including during transmission). Support for SPI (serial peripheral interface) Support for LIN Protocol Ver2.1 Support for 100kbps I²C only
DDR HS-SPI	E²PROM and the flash device of the Single/Dual/Quad-SPI protocol can be connected.
LCD controller/driver	This model does not support sub clock. <Duty drive> Segment driver and common driver that can directly drive LCD panel (liquid crystal display) : 32SEG x 4COM Each of COM0 to COM3, SEG0 to SEG31, V0, V1, V2, and V3 pins can be switched to the general-purpose port by the software. (The SEG23 to SEG31 pins can be switched to static driving.) <Static drive> Segment driver and common driver that can directly drive LCD panel (liquid crystal display) : 8SEG/1COM Each of ST0 to ST8 pins can be switched to the general-purpose port or to the segment output of duty drive by software.
Interrupt controller	Detects an interrupt factor. Sets the interrupt level.
CAN-FD Interface	Complies with the CAN specification, version 2.0, parts A and B. Supported reception message buffer size: Up to 8 byte x 192 buffers Supported transmission message buffer size: Up to 8 byte x 32 buffers A communication rate supports 5 Mbps CAN with flexible data rate This model has CAN prescaler implemented by CAN clock Can wakeup function The CAN clock source can change the main clock and PLL0 clock.
Stepping motor controller	High current output × 4 lines - The PWM cycle can be set to 15.625 kHz (when the peripheral clock operates in 16MHz). - The back electromotive force from the motor can be detected by the pin sharing ADC.
Quad position & revolution counter	16-bit position counter, 16-bit revolution counter
Sound generator	In addition to the frequency data and amplitude data setting, the followings can be set : - Decrement or increment data, and execution cycle - Tone output pulse count (output interval)
Calibration	Corrects the real time clock driven by the CR clock in comparison with the main clock.
Software watchdog	Whether counter is enabled or not in RUN state can be set Whether counter is enabled or not in PSS state can be set

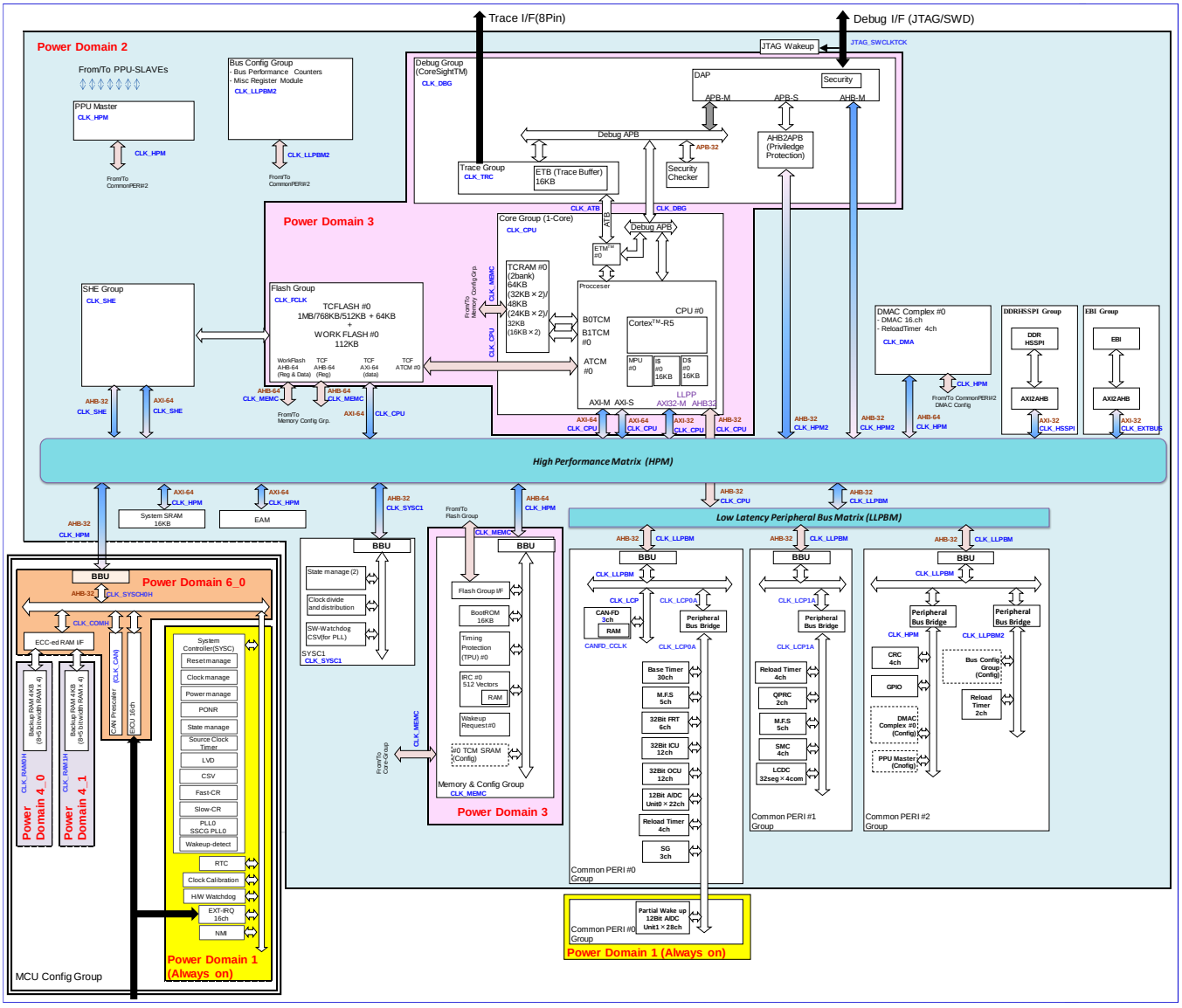
Function	Features
Hardware watchdog	CR-based CPU operation detection counter Measure against program runaways
CRC generation	Displays a CRC code in a result register by successively writing to an input register.
SHE	Extended hardware security
Low-voltage detection	Low-voltage detection for each of the voltages is mounted. (5 V and 1.2 V)
Low-power consumption control	Power-off function is mounted. Standby function is mounted. Partial wakeup function is mounted.
I/O relocation	Peripheral functions to be relocated and number of branches - CAN-FD:MAX 3 branches - External interrupt input: Max 2 Branches - Multi-function serial : Max 2 Branches - Base Timer output :Max 2 Branches - Output compare : Max 2 Branches - Input capture : Max 3 Branches - Free-run Timer : Max 2 Branches - Quad position & revolution counter : Max 2 Branches - Sound generator : Max 2 Branches
NMI request	Non-maskable interrupt signal input from the NMIX pin
Debug interface	ARM CoreSight technology Trace function: Embedded Trace Macrocell (ETM) Debug interface JTAG

5. Block Diagrams

This section provides block diagrams of the S6J3120 series.

Figure 5-1 S6J312xHzC* Block Diagram

***x: A/9, z: A/B**



6. Memory Map

This section explains the memory map.

Figure 6-1 Memory Map

ADDRESS START	ADDRESS END	group	S6J312AHzC* part	S6J3129HzC* part
0x0000_0000	0x0000_7FFF	Internal area for CR5 Complex	TCRAM (Main 64KByte)	TCRAM (Main 48KByte)
0x0000_8000	0x0000_BFFF			
0x0000_C000	0x0000_FFFF			Reserved
0x0001_0000	0x007F_FFFF		Reserved	Reserved
0x0080_0000	0x009E_FFFF		Reserved	Reserved
0x009F_0000	0x009F_FFFF		TCM_FLASH (Small Sector 8KByte×8)	TCM_FLASH (Small Sector 8KByte×8)
0x00A0_0000	0x00A7_FFFF		TCM_FLASH (Code 1MByte)	TCM_FLASH (Code 768KByte)
0x00A8_0000	0x00AB_FFFF			
0x00AC_0000	0x00AF_FFFF			Reserved
0x00B0_0000	0x00DF_FFFF		Reserved	Reserved
0x00E0_0000	0x00FF_FFFF		Reserved	Reserved
0x0100_0000	0x019E_FFFF		Reserved	Reserved
0x019F_0000	0x019F_FFFF		AXI_FLASH_MEMORY (Small Sector 8KByte×8 *Mirror)	AXI_FLASH_MEMORY (Small Sector 8KByte×8 *Mirror)
0x01A0_0000	0x01A7_FFFF		AXI_FLASH_MEMORY (Code 1MByte *Mirror)	AXI_FLASH_MEMORY (Code 768KByte *Mirror)
0x01A8_0000	0x01AB_FFFF			
0x01AC_0000	0x01AF_FFFF			Reserved
0x01B0_0000	0x01DF_FFFF		Reserved	Reserved
0x01E0_0000	0x01FF_FFFF		Reserved	Reserved
0x0200_0000	0x0200_3FFF	Shared Flash and memory area	SYSTEM SRAM (16KByte)	SYSTEM SRAM (16KByte)
0x0200_4000	0x0203_FFFF		Reserved	Reserved
0x0204_0000	0x027F_FFFF		Reserved	Reserved
0x0280_0000	0x0280_0FFF		Exclusive Access Memory	Exclusive Access Memory
0x0280_1000	0x03FF_FFFF		Reserved	Reserved
0x0400_0000	0x05FF_FFFF		AXI_SLAVE_CORE0	AXI_SLAVE_CORE0
0x0600_0000	0x0DFF_FFFF		Reserved	Reserved
0x0E00_0000	0x0E01_BFFF		WORK_FLASH (112KByte mirror area 1)	WORK_FLASH (112KByte mirror area 1)
0x0E01_C000	0x0E0F_FFFF		Reserved	Reserved
0x0E10_0000	0x0E1F_FFFF		Reserved	Reserved
0x0E20_0000	0x0E21_BFFF		WORK_FLASH (112KByte mirror area 3)	WORK_FLASH (112KByte mirror area 3)
0x0E21_C000	0x0E2F_FFFF		Reserved	Reserved
0x0E30_0000	0x0E31_BFFF		WORK_FLASH (112KByte mirror area 4)	WORK_FLASH (112KByte mirror area 4)
0x0E31_C000	0x0E3F_FFFF		Reserved	Reserved
0x0E40_0000	0x0E7F_FFFF		Reserved	Reserved
0x0E80_0000	0x0E80_1FFF		Backup RAM 8KByte	Backup RAM 8KByte
0x0E80_2000	0x0E80_FFFF		Reserved	Reserved
0x0E81_0000	0x0E87_FFFF		Reserved	Reserved
0x0E88_0000	0x0FFF_FFFF		Reserved	Reserved
0x1000_0000	0x1FFF_FFFF	External bus area	EBI_MEMORY(SRAM/FLASH)	EBI_MEMORY(SRAM/FLASH)
0x2000_0000	0x7FFF_FFFF	Reserved	Reserved	Reserved
0x8000_0000	0x8FFF_FFFF	HSSPI memory area	HSSPI0_MEMORY	HSSPI0_MEMORY
0x9000_0000	0xAFFF_FFFF	Reserved	Reserved	Reserved
0xB000_0000	0xB483_FFFF	Peri area	Peri_area	Peri_area
0xB484_0000	0xB484_FFFF		APPS#5	APPS#5
0xB485_0000	0xB48B_FFFF		Peri_area	Peri_area
0xB48C_0000	0xB48C_FFFF		APPS#7	APPS#7
0xB48D_0000	0xB7FF_FFFF		Peri_area	Peri_area
0xB800_0000	0xBFFF_FFFF			
0xC000_0000	0xFFFE_DFFF	Reserved	Reserved	Reserved
0xFFFE_E000	0xFFFE_FFFF	Error Config	ERRCFG	ERRCFG
0xFFFF_0000	0xFFFF_3FFF	BootROM	BootRom	BootRom
0xFFFF_4000	0xFFFF_FFFF		Reserved	Reserved

*z: A/B

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.

In each of the following memory area combinations, the areas are physically the same memory area.

1. TCM FLASH (0x00A0_0000 -) and AXI FLASH MEMORY (0x01A0_0000 -)
2. TCM FLASH Small Sector (0x009F_0000 -) and AXI FLASH MEMORY Small Sector (0x019F_0000 -)
3. WORKFLASH (0x0E00_0000 -), WORKFLASH (0x0E20_0000 -), and WORKFLASH (0x0E30_0000 -)

The ECC movement in TCM port is based on ECC setting inside the CPU.

- The differences between the TCM FLASH and AXI FLASH include the following.

Function	TCM FLASH	AXI FLASH
High-speed Access Using Dedicated Bus	Applicable	Not applicable
Write and Erase	Not applicable (Read-only)	Applicable
Read	Applicable	Applicable

- The differences between WORKFLASH areas include the following.

Area	Function
WORKFLASH Area 1	Used in write operation (with ECC)
WORKFLASH Area 3	Used in write operation (without ECC)
WORKFLASH Area 4	Used in read operation

- Terms are as follows.

Term	Description
TCM RAM	Main RAM
TCM FLASH	Program FLASH (TCM area)
AXI FLASH	Program FLASH (AXI area) This is physically the same as the TCM FLASH.
SYSTEM RAM	System RAM
AXI SLAVE CORE	AXI CPU control area
WORKFLASH	FLASH for work
BACKUP RAM	Backup RAM
EBI MEMORY	Memory for External bus interface
HSSPI0 MEMORY	Memory for DDR HS-SPI
Peri area	Entire area for peripheral functions
APPS#5	Part of area for peripheral functions
APPS#7	Part of area for peripheral functions

Term	Description
ERRCFG	Error configuration area
BootROM	ROM for reset boot

Figure 6-2 S6J312xHzC* Peripheral Map *x: A/9 z: A/B

START Address	END Address	Group	Function	PPU No
B000_0000	B010_7FFF		Reserved	-
B010_0000	B010_03FF		EBI registers	0
B010_0400	B010_0FFF		Reserved	-
B010_1000	B010_13FF		DDR_HSSPI	1
B010_0400	B010_7FFF		Reserved	-
B010_8000	B010_80FF	SystemSRAM	SystemSRAM registers	-
B010_8100	B02F_FFFF		Reserved	-
B030_0000	B030_7FFF	SYSC1	System Controller #1	-
B030_8000	B03F_FFFF	SYSC1	SWDT	-
B040_0000	B040_7FFF	MEMORY_CONFIG_GROUP	IRC0	21
B040_8000	B040_FFFF	MEMORY_CONFIG_GROUP	TPU0	19
B041_0000	B041_0FFF	MEMORY_CONFIG_GROUP	TCRAM Control Status Register	16
B041_1000	B041_1FFF	MEMORY_CONFIG_GROUP	TCFlash Control Status Register	17
B041_2000	B041_20FF	MEMORY_CONFIG_GROUP	WFlash Control Status Register	18
B041_2100	B04F_FFFF		Reserved	-
B050_0000	B05F_FFFF		Reserved	-
B060_0000	B060_007F	MCU_CONFIG_GROUP	Protection register area	-
B060_0080	B060_00FF	MCU_CONFIG_GROUP	RUN profile register area	-
B060_0100	B060_017F	MCU_CONFIG_GROUP	PSS profile register area	-
B060_0180	B060_01FF	MCU_CONFIG_GROUP	APP profile register area	-
B060_0200	B060_027F	MCU_CONFIG_GROUP	STS profile register area	-
B060_0280	B060_02FF	MCU_CONFIG_GROUP	System register area	-
B060_0300	B060_037F	MCU_CONFIG_GROUP	CSV	-
B060_0380	B060_03FF	MCU_CONFIG_GROUP	RESET	-
B060_0400	B060_047F	MCU_CONFIG_GROUP	SCT(Fast CR)	34
B060_0480	B060_04FF	MCU_CONFIG_GROUP	SCT(Slow CR)	33
B060_0500	B060_05FF	MCU_CONFIG_GROUP	SCT(Main clock)	35
B060_0600	B060_067F	MCU_CONFIG_GROUP	Clock System	-
B060_0680	B060_06FF	MCU_CONFIG_GROUP	Special register area	-
B060_0700	B060_07FF	MCU_CONFIG_GROUP	Debug register area	-
B060_0800	B060_BFFF	MCU_CONFIG_GROUP	Mode	-
B060_C000	B060_FFFF	MCU_CONFIG_GROUP	HWDT	-
B061_0000	B061_7FFF		Reserved	-
B061_8000	B061_FFFF	MCU_CONFIG_GROUP	RTC	32
B062_0000	B063_FFFF	MCU_CONFIG_GROUP	EIC	-
B064_0000	B065_FFFF		Reserved	-
B066_0000	B067_FFFF		Reserved	-
B068_0000	B068_7FFF	MCU_CONFIG_GROUP	BURAMIF	-
B068_8000	B068_83FF	MCU_CONFIG_GROUP	EICU	37
B068_8400	B068_87FF	MCU_CONFIG_GROUP	CR_Calibration	38
B068_8800	B068_8BFF	MCU_CONFIG_GROUP	IRQ ALL	42
B068_8C00	B068_FFFF	MCU_CONFIG_GROUP	CAN Prescaler	43
B069_0000	B06F_FFFF		Reserved	-
B070_0000	B07F_FFFF		Reserved	-
B080_0000	B0FF_FFFF	Bit RMW alias	BBU for MCU Config (Covers B060_0000 -- B06F_FFFF)	-
B100_0000	B10F_FFFF	Bit RMW alias	BBU for SYSC1 (Covers B030_0000 -- B031_FFFF)	-
B110_0000	B11F_FFFF	Bit RMW alias	BBU for MEMC (Covers B040_0000 -- B041_FFFF)	-
B120_0000	B1FF_FFFF		Reserved	-
B200_0000	B20F_FFFF	SHE	SHE configuration registers	63
B210_0000	B46F_FFFF		Reserved	-

START Address	END Address	Group	Function	PPU No
B470_0000	B470_3FFF	CommonPERI #2	DMAC #0 registers	64
B470_4000	B470_FFFF		Reserved	-
B471_0000	B471_0FFF	CommonPERI #2	MPU for DMAC#0	66
B471_1000	B471_3FFF		Reserved	-
B471_4000	B471_4FFF	CommonPERI #2	DMA Complex #0 registers (Additional registers, RLTs)	68
B471_5000	B471_7FFF		Reserved	-
B471_8000	B471_83FF	CommonPERI #2	CRC#0	70
B471_8400	B471_87FF	CommonPERI #2	CRC#1	71
B471_8800	B471_8BFF	CommonPERI #2	CRC#2	72
B471_8C00	B471_8FFF	CommonPERI #2	CRC#3	73
B471_9000	B473_7FFF		Reserved	-
B473_8000	B473_FFFF	CommonPERI #2	GPIO	74
B474_0000	B474_7FFF	CommonPERI #2	PPC	75
B474_8000	B474_FFFF	CommonPERI #2	RIC	76
B475_0000	B475_7FFF	CommonPERI #2	PPU	-
B475_8000	B478_7FFF		Reserved	-
B478_8000	B478_83FF	CommonPERI #2	Reload Timer ch.32	160
B478_8400	B478_87FF	CommonPERI #2	Reload Timer ch.33	161
B478_8800	B478_FBFF		Reserved	-
B478_FC00	B478_FFFF	CommonPERI #2	Misc registers	82
B479_0000	B47F_FFFF		Reserved	-
B480_0000	B480_03FF	CommonPERI #0	M.F.Serial ch.0	176
B480_0400	B480_07FF	CommonPERI #0	M.F.Serial ch.1	177
B480_0800	B480_0BFF	CommonPERI #0	M.F.Serial ch.2	178
B480_0C00	B480_0FFF	CommonPERI #0	M.F.Serial ch.3	179
B480_1000	B480_13FF	CommonPERI #0	M.F.Serial ch.4	180
B480_1400	B480_7FFF		Reserved	-
B480_8000	B480_83FF	CommonPERI #0	BaseTimer ch.0	88
B480_8400	B480_87FF	CommonPERI #0	BaseTimer ch.1	89
B480_8800	B480_8BFF	CommonPERI #0	BaseTimer ch.2	90
B480_8C00	B480_8FFF	CommonPERI #0	BaseTimer ch.3	91
B480_9000	B480_93FF	CommonPERI #0	BaseTimer ch.4	92
B480_9400	B480_97FF	CommonPERI #0	BaseTimer ch.5	93
B480_9800	B480_9BFF	CommonPERI #0	BaseTimer ch.6	94
B480_9C00	B480_9FFF	CommonPERI #0	BaseTimer ch.7	95
B480_A000	B480_A3FF	CommonPERI #0	BaseTimer ch.8	96
B480_A400	B480_A7FF	CommonPERI #0	BaseTimer ch.9	97
B480_A800	B480_ABFF	CommonPERI #0	BaseTimer ch.10	98
B480_AC00	B480_AFFF	CommonPERI #0	BaseTimer ch.11	99
B480_B000	B480_FFFF		Reserved	-
B481_0000	B481_03FF	CommonPERI #0	Reload Timer ch.0	128
B481_0400	B481_07FF	CommonPERI #0	Reload Timer ch.1	129
B481_0800	B481_0BFF	CommonPERI #0	Reload Timer ch.2	130
B481_0C00	B481_0FFF	CommonPERI #0	Reload Timer ch.3	131
B481_1000	B481_FFFF		Reserved	-
B482_0000	B482_03FF	CommonPERI #0	FRT ch.0	208
B482_0400	B482_07FF	CommonPERI #0	FRT ch.1	209
B482_0800	B482_0BFF	CommonPERI #0	FRT ch.2	210
B482_0C00	B482_0FFF	CommonPERI #0	FRT ch.3	211
B482_1000	B482_13FF	CommonPERI #0	FRT ch.4	212
B482_1400	B482_17FF	CommonPERI #0	FRT ch.5	213
B482_1800	B482_7FFF		Reserved	-
B482_8000	B482_83FF	CommonPERI #0	ICU ch.0 / ch.1	224
B482_8400	B482_87FF	CommonPERI #0	ICU ch.2 / ch.3	225
B482_8800	B482_8BFF	CommonPERI #0	ICU ch.4 / ch.5	226
B482_8C00	B482_8FFF	CommonPERI #0	ICU ch.6 / ch.7	227
B482_9000	B482_93FF	CommonPERI #0	ICU ch.8 / ch.9	228
B482_9400	B482_97FF	CommonPERI #0	ICU ch.10 / ch.11	229
B482_9800	B482_FFFF		Reserved	-

START Address	END Address	Group	Function	PPU No
B483_0000	B483_03FF	CommonPERI #0	OCU ch.0 / ch.1	240
B483_0400	B483_07FF	CommonPERI #0	OCU ch.2 / ch.3	241
B483_0800	B483_0BFF	CommonPERI #0	OCU ch.4 / ch.5	242
B483_0C00	B483_0FFF	CommonPERI #0	OCU ch.6 / ch.7	243
B483_1000	B483_13FF	CommonPERI #0	OCU ch.8 / ch.9	244
B483_1400	B483_17FF	CommonPERI #0	OCU ch.10 / ch.11	245
B483_1800	B483_FBFF		Reserved	-
B483_FC00	B483_FFFF	Common PERI #0	Misc registers	80
B484_0000	B484_FFFF	APPS #5	APPS#5 area	-
B485_0000	B487_FFFF		Reserved	-
B488_0000	B488_03FF	CommonPERI #1	M.F.Serial ch.8	184
B488_0400	B488_07FF	CommonPERI #1	M.F.Serial ch.9	185
B488_0800	B488_0BFF	CommonPERI #1	M.F.Serial ch.10	186
B488_0C00	B488_0FFF	CommonPERI #1	M.F.Serial ch.11	187
B488_1000	B488_13FF	CommonPERI #1	M.F.Serial ch.12	188
B488_1400	B488_FFFF		Reserved	-
B489_0000	B489_03FF	CommonPERI #1	Reload Timer ch.16	144
B489_0400	B489_07FF	CommonPERI #1	Reload Timer ch.17	145
B489_0800	B489_0BFF	CommonPERI #1	Reload Timer ch.18	146
B489_0C00	B489_0FFF	CommonPERI #1	Reload Timer ch.19	147
B489_1000	B489_7FFF		Reserved	-
B489_8000	B489_83FF	CommonPERI #1	QPRC ch.8	200
B489_8400	B489_87FF	CommonPERI #1	QPRC ch.9	201
B489_8800	B48B_0FFF		Reserved	-
B48B_1000	B48B_FBFF		Reserved	-
B48B_FC00	B48B_FFFF	CommonPERI #1	Misc registers	81
B48C_0000	B48C_FFFF	APPS #7	APPS#7 area	-
B48D_0000	B48F_FFFF		Reserved	-
B490_0000	B490_FFFF	CommonPERI #0	CAN_FD ch.0	256
B491_0000	B491_FFFF	CommonPERI #0	CAN_FD ch.1	257
B492_0000	B492_FFFF	CommonPERI #0	CAN_FD ch.2	258
B493_0000	B4BF_FFFF		Reserved	-
B4C0_0000	B4FF_FFFF	Bit RMW alias	BBU for CommonPERI#0 (Covers B490_0000 -- B497_FFFF)	-
B500_0000	B5FF_FFFF		Reserved	-
B600_0000	B6FF_FFFF		Reserved	-
B700_0000	B77F_FFFF	Bit RMW alias	BBU alias for CommonPERI#2 (Covers B470_0000 -- B47F_FFFF)	-
B780_0000	B7BF_FFFF	Bit RMW alias	BBU alias for CommonPERI#0 (Covers B480_0000 -- B487_FFFF)	-
B7C0_0000	B7FF_FFFF	Bit RMW alias	BBU alias for CommonPERI#1 (Covers B488_0000 -- B48F_FFFF)	-
B800_0000	FFFE_DFFF		Reserved	-
FFFE_E000	FFFE_FBFC	Error Config	IRC	-
FFFE_FC00	FFFE_FFFF	Error Config	BootROM I/F	20

- APPS#5 area

START Address	END Address		Function	PPU No
B484_0000	B484_03FF	APPS #5	Sound Generator ch.0	264
B484_0400	B484_07FF	APPS #5	Sound Generator ch.1	265
B484_0800	B484_0BFF	APPS #5	Sound Generator ch.2	266
B484_0C00	B484_37FF		Reserved	-
B484_3800	B484_3BFF	APPS #5	BaseTimer ch.12	278
B484_3C00	B484_3FFF	APPS #5	BaseTimer ch.13	279
B484_4000	B484_43FF	APPS #5	BaseTimer ch.14	280
B484_4400	B484_47FF	APPS #5	BaseTimer ch.15	281
B484_4800	B484_4BFF	APPS #5	BaseTimer ch.16	282
B484_4C00	B484_4FFF	APPS #5	BaseTimer ch.17	283
B484_5000	B484_53FF	APPS #5	BaseTimer ch.18	284
B484_5400	B484_57FF	APPS #5	BaseTimer ch.19	285
B484_5800	B484_5BFF	APPS #5	BaseTimer ch.20	286
B484_5C00	B484_5FFF	APPS #5	BaseTimer ch.21	287
B484_6000	B484_63FF	APPS #5	BaseTimer ch.22	288
B484_6400	B484_67FF	APPS #5	BaseTimer ch.23	289
B484_6800	B484_6BFF	APPS #5	BaseTimer ch.24	290
B484_6C00	B484_6FFF	APPS #5	BaseTimer ch.25	291
B484_7000	B484_73FF	APPS #5	BaseTimer ch.26	292
B484_7400	B484_77FF	APPS #5	BaseTimer ch.27	293
B484_7800	B484_7BFF	APPS #5	BaseTimer ch.28	294
B484_7C00	B484_7FFF	APPS #5	BaseTimer ch.29	295
B484_8000	B484_83FF	APPS #5	A/D unit0	296
B484_8400	B484_87FF	APPS #5	A/D unit1 , Partial Wake Up	297
B484_8800	B484_8BFF	APPS #5	A/D analog input control	298
B484_8C00	B484_8FFF		Reserved	-
B484_9000	B484_93FF	APPS #5	Global Timer	300
B484_9400	B484_FFFF		Reserved	-

- APPS#7 area

START Address	END Address		Function	PPU No
B48C_0000	B48C_3FFF		Reserved	-
B48C_4000	B48C_43FF	APPS #7	Stepper Motor Control ch.0	317
B48C_4400	B48C_47FF	APPS #7	Stepper Motor Control ch.1	318
B48C_4800	B48C_4BFF	APPS #7	Stepper Motor Control ch.2	319
B48C_4C00	B48C_4FFF	APPS #7	Stepper Motor Control ch.3	320
B48C_5000	B48C_57FF		Reserved	-
B48C_5800	B48C_5BFF	APPS #7	SMC Trigger Generator	323
B48C_5C00	B48C_5FFF	APPS #7	Liquid Crystal Display Controller	324
B48C_6000	B48C_63FF	APPS #7	Liquid Crystal Display input/output control	325
B48C_6400	B48C_FFFF		Reserved	-

When MPU attribute of Cortex®-R5 is configured as "Normal", store buffer inside Cortex®-R5 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]*

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

- *FLASH Memory (when writing commands)*

SHE OFF product is prohibited to access SHE area (B200_0000 to B20F_FFFF)

7. Abbreviations

This section provides a list of the abbreviations used in S6J3120 series specifications.

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

BBU -- BIT Banding Unit

BDR -- Boot Description Record

BPC – Bus Performance Counter

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

DDRHSSPI -- DDR High Speed SPI Interface

DED -- Double Error Detection

DMA -- Direct Memory Access

DMAC -- DMA Controller

EAM -- Exclusive Access Memory

EBI -- External Bus Interface

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

IPCU -- Inter-Processor Communication Unit

IRC -- Interrupt Controller

IRQ -- Interrupt Request

ISR -- Interrupt Service Routine

JTAG -- Joint Test Action Group

LCDC -- Liquid Crystal Display Controller

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

QPRC -- Quad Position & Revolution Counter

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

SG -- Sound Generator

SMC -- Stepper Motor Controller

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 -- Timing Protection Unit

VIC -- Vectored Interrupt Controller

WDR -- Watchdog Description Record

WDT -- Watchdog Timer

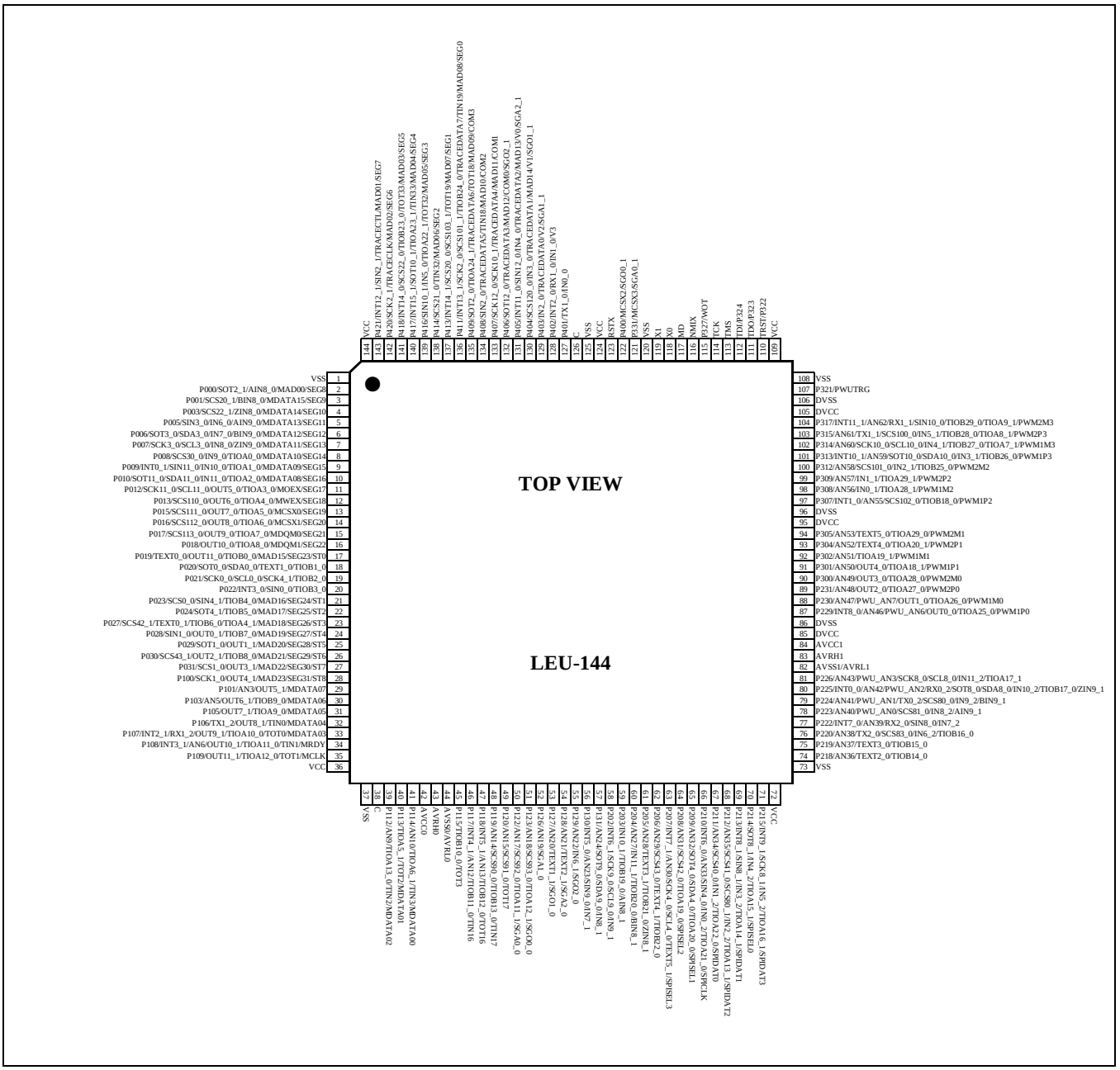
WorkFLASH -- Work Flash Memory

8. Pin Assignment

This section provides the pin assignment of the S6J3120 series.

Figure 8-1 Pin Assignment for S6J312xHzC*

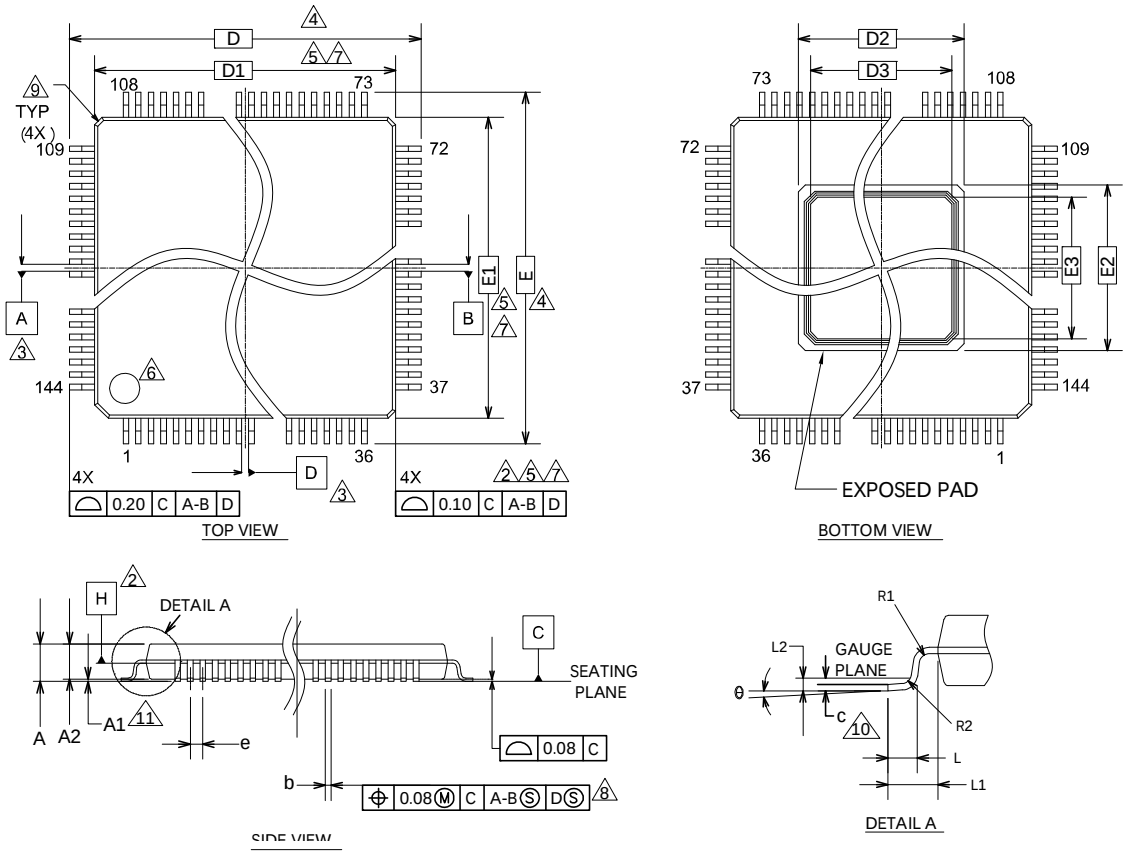
*** x: A/9, z: A/B**



9. Package Dimensions

This section provides the package dimensions of the S6J3120 series.

Figure 9-1 LEU144 Package Dimensions



SYMBOL	DIMENSIONS		
	MIN.	NOM.	MAX.
A	——	——	1.70
A1	0.00	——	0.20
A2	1.35	1.40	1.45
D	22.00 BSC		
D1	20.00 BSC		
D2	7.05 BSC		
D3	5.85 BSC		
E	22.00 BSC		
E1	20.00 BSC		
E2	7.05 BSC		
E3	5.85 BSC		

SYMBOL	DIMENSIONS		
	MIN.	NOM.	MAX.
b	0.17	0.22	0.27
e	0.50 BSC.		
R 1	0.08	——	——
R 2	0.08	——	0.20
θ	0°	4°	8°
c	0.12	——	0.20
L	0.45	0.60	0.75
L 1	1.00 REF		
L 2	0.25		

NOTES :

1. ALL DIMENSIONS ARE IN MILLIMETERS.
2. DATUM PLANE H IS LOCATED AT THE BOTTOM OF THE MOLD PARTING LINE COINCIDENT WITH WHERE THE LEAD EXITS THE BODY.
3. DATUMS A-B AND D TO BE DETERMINED AT DATUM PLANE H.
4. TO BE DETERMINED AT SEATING PLANE C.
5. DIMENSIONS D1 AND E1 DO NOT INCLUDE MOLD PROTRUSION. ALLOWABLE PROTRUSION IS 0.25mm PRE SIDE. DIMENSIONS D1 AND E1 INCLUDE MOLD MISMATCH AND ARE DETERMINED AT DATUM PLANE H.
6. DETAILS OF PIN 1 IDENTIFIER ARE OPTIONAL BUT MUST BE LOCATED WITHIN THE ZONE INDICATED.
7. REGARDLESS OF THE RELATIVE SIZE OF THE UPPER AND LOWER BODY SECTIONS. DIMENSIONS D1 AND E1 ARE DETERMINED AT THE LARGEST FEATURE OF THE BODY EXCLUSIVE OF MOLD FLASH AND GATE BURRS. BUT INCLUDING ANY MISMATCH BETWEEN THE UPPER AND LOWER SECTIONS OF THE MOLDER BODY.
8. DIMENSION b DOES NOT INCLUDE DAMBER PROTRUSION. THE DAMBAR PROTRUSION(S) SHALL NOT CAUSE THE LEAD WIDTH TO EXCEED b MAXIMUM BY MORE THAN 0.08mm. DAMBAR CANNOT BE LOCATED ON THE LOWER RADIUS OR THE LEAD FOOT.
9. EXACT SHAPE AND SIZE OF THIS FEATURE IS OPTIONAL.
10. THESE DIMENSIONS APPLY TO THE FLAT SECTION OF THE LEAD BETWEEN 0.10mm AN D 0.25mm FROM THE LEAD TIP.
11. A1 IS DEFINED AS THE DISTANCE FROM THE SEATING PLANE TO THE LOWEST POINT OF THE PACKAGE BODY.

002-10858 Rev. **

PACKAGE OUTLINE, 144 LEAD TEQFP
20.0X20.0X1.7 MM LEU144 REV**

10. List of Pin Functions

This section provides a list of the pin functions of the S6J3120 series.

Table 10-1 S6J312xHzC* Pin Functions

*** x: A/9, z: A/B**

Pin No.	Pin Name	Polarity	I/O Circuit Type	Function
2	P000	-	K	General-purpose I/O port
	SOT2_1	-		Multi-function serial ch.2 serial data output pin (1)
	AIN8_0	-		QPRC ch.8 AIN input pin (0)
	MAD00	-		External bus interface address bit0 output pin
	SEG8	-		LCDC segment 8 (Duty) output pin
3	P001	-	K	General-purpose I/O port
	SCS20_1	-		Multi-function serial ch.2 serial chip select 0 I/O pin (1)
	BIN8_0	-		QPRC ch.8 BIN input pin (0)
	MDATA15	-		External bus interface data bus bit15 I/O pin
	SEG9	-		LCDC segment 9 (Duty) output pin
4	P003	-	K	General-purpose I/O port
	SCS22_1	-		Multi-function serial ch.2 serial chip select 2 output pin (1)
	ZIN8_0	-		QPRC ch.8 ZIN input pin (0)
	MDATA14	-		External bus interface data bus bit14 I/O pin
	SEG10	-		LCDC segment 10 (Duty) output pin
5	P005	-	K	General-purpose I/O port
	SIN3_0	-		Multi-function serial ch.3 serial data input pin (0)
	IN6_0	-		Input capture ch.6 input pin (0)
	AIN9_0	-		QPRC ch.9 AIN input pin (0)
	MDATA13	-		External bus interface data bus bit13 I/O pin
	SEG11	-		LCDC segment 11 (Duty) output pin
6	P006	-	K	General-purpose I/O port
	SOT3_0	-		Multi-function serial ch.3 serial data output pin (0)
	SDA3_0	-		I^2C bus ch.3 serial data I/O pin
	IN7_0	-		Input capture ch.7 input pin (0)
	BIN9_0	-		QPRC ch.9 BIN input pin (0)
	MDATA12	-		External bus interface data bus bit12 I/O pin
	SEG12	-		LCDC segment 12 (Duty) output pin
7	P007	-	K	General-purpose I/O port
	SCK3_0	-		Multi-function serial ch.3 clock I/O pin (0)
	SCL3_0	-		I^2C bus ch.3 serial clock I/O pin
	IN8_0	-		Input capture ch.8 input pin (0)
	ZIN9_0	-		QPRC ch.9 ZIN input pin (0)
	MDATA11	-		External bus interface data bus bit11 I/O pin
	SEG13	-		LCDC segment 13 (Duty) output pin

Pin No.	Pin Name	Polarity	I/O Circuit Type	Function
8	P008	-	K	General-purpose I/O port
	SCS30_0	-		Multi-function serial ch.3 serial chip select 0 I/O pin (0)
	IN9_0	-		Input capture ch.9 input pin (0)
	TIOA0_0	-		Base timer ch.0 TIOA output pin (0)
	MDATA10	-		External bus interface data bus bit10 I/O pin
	SEG14	-		LCDC segment 14 (Duty) output pin
9	P009	-	K	General-purpose I/O port
	INT0_1	-		INT0 external interrupt input pin (1)
	SIN11_0	-		Multi-function serial ch.11 serial data input pin (0)
	IN10_0	-		Input capture ch.10 input pin (0)
	TIOA1_0	-		Base timer ch.1 TIOA I/O pin (0)
	MDATA09	-		External bus interface data bus bit9 I/O pin
	SEG15	-		LCDC segment 15 (Duty) output pin
10	P010	-	K	General-purpose I/O port
	SOT11_0	-		Multi-function serial ch.11 serial data output pin (0)
	SDA11_0	-		I^2C bus ch.11 serial data I/O pin
	IN11_0	-		Input capture ch.11 input pin (0)
	TIOA2_0	-		Base timer ch.2 TIOA output pin (0)
	MDATA08	-		External bus interface data bus bit8 I/O pin
	SEG16	-		LCDC segment 16 (Duty) output pin
11	P012	-	K	General-purpose I/O port
	SCK11_0	-		Multi-function serial ch.11 clock I/O pin (0)
	SCL11_0	-		I^2C bus ch.11 serial clock I/O pin
	OUT5_0	-		Output compare ch.5 output pin (0)
	TIOA3_0	-		Base timer ch.3 TIOA I/O pin (0)
	MOEX	-		External bus interface read enable output pin
	SEG17	-		LCDC segment 17 (Duty) output pin
12	P013	-	K	General-purpose I/O port
	SCS110_0	-		Multi-function serial ch.11 serial chip select 0 I/O pin (0)
	OUT6_0	-		Output compare ch.6 output pin (0)
	TIOA4_0	-		Base timer ch.4 TIOA output pin (0)
	MWEX	-		External bus interface write enable output pin
	SEG18	-		LCDC segment 18 (Duty) output pin
13	P015	-	K	General-purpose I/O port
	SCS111_0	-		Multi-function serial ch.11 serial chip select 1 output pin (0)
	OUT7_0	-		Output compare ch.7 output pin (0)
	TIOA5_0	-		Base timer ch.5 TIOA I/O pin (0)
	MCSX0	-		External bus interface chip select 0 output pin
	SEG19	-		LCDC segment 19 (Duty) output pin

Pin No.	Pin Name	Polarity	I/O Circuit Type	Function
14	P016	-	K	General-purpose I/O port
	SCS112_0	-		Multi-function serial ch.11 serial chip select 2 output pin (0)
	OUT8_0	-		Output compare ch.8 output pin (0)
	TIOA6_0	-		Base timer ch.6 TIOA output pin (0)
	MCSX1	-		External bus interface chip select 1 output pin
	SEG20	-		LCDC segment 20 (Duty) output pin
15	P017	-	K	General-purpose I/O port
	SCS113_0	-		Multi-function serial ch.11 serial chip select 3 output pin (0)
	OUT9_0	-		Output compare ch.9 output pin (0)
	TIOA7_0	-		Base timer ch.7 TIOA I/O pin (0)
	MDQM0	-		External bus interface byte mask 0 output pin
	SEG21	-		LCDC segment 21 (Duty) output pin
16	P018	-	K	General-purpose I/O port
	OUT10_0	-		Output compare ch.10 output pin (0)
	TIOA8_0	-		Base timer ch.8 TIOA output pin (0)
	MDQM1	-		External bus interface byte mask 1 output pin
	SEG22	-		LCDC segment 22 (Duty) output pin
17	P019	-	K	General-purpose I/O port
	TEXT0_0	-		Free-run timer 0 clock input pin (0)
	OUT11_0	-		Output compare ch.11 output pin (0)
	TIOB0_0	-		Base timer ch.0 TIOB input pin (0)
	MAD15	-		External bus interface address bit15 output pin
	SEG23/ST0	-		LCDC segment 23 (Duty) / segment 0 (Static) output pin
18	P020	-	Q	General-purpose I/O port
	SOT0_0	-		Multi-function serial ch.0 serial data output pin (0)
	SDA0_0	-		I^2C bus ch.0 serial data I/O pin
	TEXT1_0	-		Free-run timer 1 clock input pin (0)
	TIOB1_0	-		Base timer ch.1 TIOB input pin (0)
19	P021	-	Q	General-purpose I/O port
	SCK0_0	-		Multi-function serial ch.0 clock I/O pin (0)
	SCL0_0	-		I^2C bus ch.0 serial clock I/O pin
	SCK4_1	-		Multi-function serial ch.4 clock I/O pin (1)
	TIOB2_0	-		Base timer ch.2 TIOB input pin (0)
20	P022	-	Q	General-purpose I/O port
	INT3_0	-		INT3 external interrupt input pin (0)
	SIN0_0	-		Multi-function serial ch.0 serial data input pin (0)
	TIOB3_0	-		Base timer ch.3 TIOB input pin (0)
21	P023	-	K	General-purpose I/O port
	SCS0_0	-		Multi-function serial ch.0 serial chip select 0 I/O pin (0)
	SIN4_1	-		Multi-function serial ch.4 serial data input pin (1)
	TIOB4_0	-		Base timer ch.4 TIOB input pin (0)
	MAD16	-		External bus interface address bit16 output pin
	SEG24/ST1	-		LCDC segment 24 (Duty) / segment 1 (Static) output pin

Pin No.	Pin Name	Polarity	I/O Circuit Type	Function
22	P024	-	K	General-purpose I/O port
	SOT4_1	-		Multi-function serial ch.4 serial data output pin (1)
	TIOB5_0	-		Base timer ch.5 TIOB input pin (0)
	MAD17	-		External bus interface address bit17 output pin
	SEG25/ST2	-		LCDC segment 25 (Duty) / segment 2 (Static) output pin
23	P027	-	K	General-purpose I/O port
	SCS42_1	-		Multi-function serial ch.4 serial chip select 2 output pin (1)
	TEXT0_1	-		Free-run timer 0 clock input pin (1)
	TIOB6_0	-		Base timer ch.6 TIOB input pin (0)
	TIOA4_1	-		Base timer ch.4 TIOA output pin (1)
	MAD18	-		External bus interface address bit18 output pin
	SEG26/ST3	-		LCDC segment 26 (Duty) / segment 3 (Static) output pin
24	P028	-	K	General-purpose I/O port
	SIN1_0	-		Multi-function serial ch.1 serial data input pin (0)
	OUT0_1	-		Output compare ch.0 output pin (1)
	TIOB7_0	-		Base timer ch.7 TIOB input pin (0)
	MAD19	-		External bus interface address bit19 output pin
	SEG27/ST4	-		LCDC segment 27 (Duty) / segment 4 (Static) output pin
25	P029	-	K	General-purpose I/O port
	SOT1_0	-		Multi-function serial ch.1 serial data output pin (0)
	OUT1_1	-		Output compare ch.1 output pin (1)
	MAD20	-		External bus interface address bit20 output pin
	SEG28/ST5	-		LCDC segment 28 (Duty) / segment 5 (Static) output pin
26	P030	-	K	General-purpose I/O port
	SCS43_1	-		Multi-function serial ch.4 serial chip select 3 output pin (1)
	OUT2_1	-		Output compare ch.2 output pin (1)
	TIOB8_0	-		Base timer ch.8 TIOB input pin (0)
	MAD21	-		External bus interface address bit21 output pin
	SEG29/ST6	-		LCDC segment 29 (Duty) / segment 6 (Static) output pin
27	P031	-	K	General-purpose I/O port
	SCS1_0	-		Multi-function serial ch.1 serial chip select 0 I/O pin (0)
	OUT3_1	-		Output compare ch.3 output pin (1)
	MAD22	-		External bus interface address bit22 output pin
	SEG30/ST7	-		LCDC segment 30 (Duty) / segment 7 (Static) output pin
28	P100	-	K	General-purpose I/O port
	SCK1_0	-		Multi-function serial ch.1 clock I/O pin (0)
	OUT4_1	-		Output compare ch.4 output pin (1)
	MAD23	-		External bus interface address bit23 output pin
	SEG31/ST8	-		LCDC segment 31 (Duty) / segment 8 (Static) output pin

Pin No.	Pin Name	Polarity	I/O Circuit Type	Function
29	P101	-	B	General-purpose I/O port
	AN3	-		ADC analog 3 input pin
	OUT5_1	-		Output compare ch.5 output pin (1)
	MDATA07	-		External bus interface data bit7 I/O pin
30	P103	-	B	General-purpose I/O port
	AN5	-		ADC analog 5 input pin
	OUT6_1	-		Output compare ch.6 output pin (1)
	TIOB9_0	-		Base timer ch.9 TIOB input pin (0)
	MDATA06	-		External bus interface data bit6 I/O pin
31	P105	-	Q	General-purpose I/O port
	OUT7_1	-		Output compare ch.7 output pin (1)
	TIOA9_0	-		Base timer ch.9 TIOA I/O pin (0)
	MDATA05	-		External bus interface data bit5 I/O pin
32	P106	-	Q	General-purpose I/O port
	TX1_2	-		CAN transmission data 1 output pin (2)
	OUT8_1	-		Output compare ch.8 output pin (1)
	TIN0	-		Reload timer ch.0 event input pin (0)
	MDATA04	-		External bus interface data bit4 I/O pin
33	P107	-	Q	General-purpose I/O port
	INT2_1	-		INT2 external interrupt input pin (1)
	RX1_2	-		CAN reception data 1 input pin (2)
	OUT9_1	-		Output compare ch.9 output pin (1)
	TIOA10_0	-		Base timer ch.10 TIOA output pin (0)
	TOT0	-		Reload timer ch.0 output pin (0)
	MDATA03	-		External bus interface data bit3 I/O pin
34	P108	-	B	General-purpose I/O port
	INT3_1	-		INT3 external interrupt input pin (1)
	AN6	-		ADC analog 6 input pin
	OUT10_1	-		Output compare ch.10 output pin (1)
	TIOA11_0	-		Base timer ch.11 TIOA I/O pin (0)
	TIN1	-		Reload timer ch.1 event input pin (0)
	MRDY	-		External bus interface ready input pin
35	P109	-	Q	General-purpose I/O port
	OUT11_1	-		Output compare ch.11 output pin (1)
	TIOA12_0	-		Base timer ch.12 TIOA output pin (0)
	TOT1	-		Reload timer ch.1 output pin (0)
	MCLK	-		External bus interface system clock output pin

Pin No.	Pin Name	Polarity	I/O Circuit Type	Function
39	P112	-	B	General-purpose I/O port
	AN9	-		ADC analog 9 input pin
	TIOA13_0	-		Base timer ch.13 TIOA I/O pin (0)
	TIN2	-		Reload timer ch.2 event input pin (0)
	MDATA02	-		External bus interface data bit2 I/O pin
40	P113	-	Q	General-purpose I/O port
	TIOA5_1	-		Base timer ch.5 TIOA I/O pin (1)
	TOT2	-		Reload timer ch.2 output pin (0)
	MDATA01	-		External bus interface data bit1 I/O pin
41	P114	-	B	General-purpose I/O port
	AN10	-		ADC analog 10 input pin
	TIOA6_1	-		Base timer ch.6 TIOA output pin (1)
	TIN3	-		Reload timer ch.3 event input pin (0)
	MDATA00	-		External bus interface data bit0 I/O pin
45	P115	-	Q	General-purpose I/O port
	TIOB10_0	-		Base timer ch.10 TIOB input pin (0)
	TOT3	-		Reload timer ch.3 output pin (0)
46	P117	-	B	General-purpose I/O port
	INT4_1	-		INT4 external interrupt input pin (1)
	AN12	-		ADC analog 12 input pin
	TIOB11_0	-		Base timer ch.11 TIOB input pin (0)
	TIN16	-		Reload timer ch.16 event input pin (0)
47	P118	-	B	General-purpose I/O port
	INT5_1	-		INT5 external interrupt input pin (1)
	AN13	-		ADC analog 13 input pin
	TIOB12_0	-		Base timer ch.12 TIOB input pin (0)
	TOT16	-		Reload timer ch.16 output pin (0)
48	P119	-	B	General-purpose I/O port
	AN14	-		ADC analog 14 input pin
	SCS90_0	-		Multi-function serial ch.9 serial chip select 0 I/O pin (0)
	TIOB13_0	-		Base timer ch.13 TIOB input pin (0)
	TIN17	-		Reload timer ch.17 event input pin (0)
49	P120	-	B	General-purpose I/O port
	AN15	-		ADC analog 15 input pin
	SCS91_0	-		Multi-function serial ch.9 serial chip select 1 output pin (0)
	TOT17	-		Reload timer ch.17 output pin (0)
50	P122	-	B	General-purpose I/O port
	AN17	-		ADC analog 17 input pin
	SCS92_0	-		Multi-function serial ch.9 serial chip select 2 output pin (0)
	TIOA11_1	-		Base timer ch.11 TIOA I/O pin (1)
	SGA0_0	-		Sound generator ch.0 SGA output pin (0)

Pin No.	Pin Name	Polarity	I/O Circuit Type	Function
51	P123	-	B	General-purpose I/O port
	AN18	-		ADC analog 18 input pin
	SCS93_0	-		Multi-function serial ch.9 serial chip select 3 output pin (0)
	TIOA12_1	-		Base timer ch.12 TIOA output pin (1)
	SGO0_0	-		Sound generator ch.0 SGO output pin (0)
52	P126	-	B	General-purpose I/O port
	AN19	-		ADC analog 19 input pin
	SGA1_0	-		Sound generator ch.1 SGA output pin (0)
53	P127	-	B	General-purpose I/O port
	AN20	-		ADC analog 20 input pin
	TEXT1_1	-		Free-run timer 1 clock input pin (1)
	SGO1_0	-		Sound generator ch.1 SGO output pin (0)
54	P128	-	B	General-purpose I/O port
	AN21	-		ADC analog 21 input pin
	TEXT2_1	-		Free-run timer 2 clock input pin (1)
	SGA2_0	-		Sound generator ch.2 SGA output pin (0)
55	P129	-	B	General-purpose I/O port
	AN22	-		ADC analog 22 input pin
	IN6_1	-		Input capture ch.6 input pin (1)
	SGO2_0	-		Sound generator ch.2 SGO output pin (0)
56	P130	-	B	General-purpose I/O port
	INT5_0	-		INT5 external interrupt input pin (0)
	AN23	-		ADC analog 23 input pin
	SIN9_0	-		Multi-function serial ch.9 serial data input pin (0)
	IN7_1	-		Input capture ch.7 input pin (1)
57	P131	-	B	General-purpose I/O port
	AN24	-		ADC analog 24 input pin
	SOT9_0	-		Multi-function serial ch.9 serial data output pin (0)
	SDA9_0	-		I^2C bus ch.9 serial data I/O pin
	IN8_1	-		Input capture ch.8 input pin (1)
58	P202	-	Q	General-purpose I/O port
	INT6_1	-		INT6 external interrupt input pin (1)
	SCK9_0	-		Multi-function serial ch.9 clock I/O pin (0)
	SCL9_0	-		I^2C bus ch.9 serial clock I/O pin
	IN9_1	-		Input capture ch.9 input pin (1)
59	P203	-	Q	General-purpose I/O port
	IN10_1	-		Input capture ch.10 input pin (1)
	TIOB19_0	-		Base timer ch.19 TIOB input pin (0)
	AIN8_1	-		QPRC ch.8 AIN input pin (1)

Pin No.	Pin Name	Polarity	I/O Circuit Type	Function
60	P204	-	B	General-purpose I/O port
	AN27	-		ADC analog 27 input pin
	IN11_1	-		Input capture ch.11 input pin (1)
	TIOB20_0	-		Base timer ch.20 TIOB input pin (0)
	BIN8_1	-		QPRC ch.8 BIN input pin (1)
61	P205	-	B	General-purpose I/O port
	AN28	-		ADC analog 28 input pin
	TEXT3_1	-		Free-run timer 3 clock input pin (1)
	TIOB21_0	-		Base timer ch.21 TIOB input pin (0)
	ZIN8_1	-		QPRC ch.8 ZIN input pin (1)
62	P206	-	B	General-purpose I/O port
	AN29	-		ADC analog 29 input pin
	SCS43_0	-		Multi-function serial ch.4 serial chip select 3 output pin (0)
	TEXT4_1	-		Free-run timer 4 clock input pin (1)
	TIOB22_0	-		Base timer ch.22 TIOB input pin (0)
63	P207	-	B	General-purpose I/O port
	INT7_1	-		INT7 external interrupt input pin (1)
	AN30	-		ADC analog 30 input pin
	SCK4_0	-		Multi-function serial ch.4 clock I/O pin (0)
	SCL4_0	-		I^2C bus ch.4 serial clock I/O pin
	TEXT5_1	-		Free-run timer 5 clock input pin (1)
	SPISEL3	-		HS-SPI slave select 3 output pin
64	P208	-	B	General-purpose I/O port
	AN31	-		ADC analog 31 input pin
	SCS42_0	-		Multi-function serial ch.4 serial chip select 2 output pin (0)
	TIOA19_0	-		Base timer ch.19 TIOA I/O pin (0)
	SPISEL2	-		HS-SPI slave select 2 output pin
65	P209	-	B	General-purpose I/O port
	AN32	-		ADC analog 32 input pin
	SOT4_0	-		Multi-function serial ch.4 serial data output pin (0)
	SDA4_0	-		I^2C bus ch.4 serial data I/O pin
	TIOA20_0	-		Base timer ch.20 TIOA output pin (0)
	SPISEL1	-		HS-SPI slave select 1 output pin
66	P210	-	B	General-purpose I/O port
	INT6_0	-		INT6 external interrupt input pin (0)
	AN33	-		ADC analog 33 input pin
	SIN4_0	-		Multi-function serial ch.4 serial data input pin (0)
	IN0_2	-		Input capture ch.0 input pin (2)
	TIOA21_0	-		Base timer ch.21 TIOA I/O pin (0)
	SPICLK	-		HS-SPI clock output pin

Pin No.	Pin Name	Polarity	I/O Circuit Type	Function
67	P211	-	B	General-purpose I/O port
	AN34	-		ADC analog 34 input pin
	SCS40_0	-		Multi-function serial ch.4 serial chip select 0 I/O pin (0)
	IN1_2	-		Input capture ch.1 input pin (2)
	TIOA22_0	-		Base timer ch.22 TIOA output pin (0)
	SPIDAT0	-		HS-SPI data 0 I/O pin
68	P212	-	B	General-purpose I/O port
	AN35	-		ADC analog 35 input pin
	SCS41_0	-		Multi-function serial ch.4 serial chip select 1 output pin (0)
	SCS80_1	-		Multi-function serial ch.8 serial chip select 0 I/O pin (1)
	IN2_2	-		Input capture ch.2 input pin (2)
	TIOA13_1	-		Base timer ch.13 TIOA I/O pin (1)
	SPIDAT2	-		HS-SPI data 2 I/O pin
69	P213	-	Q	General-purpose I/O port
	INT8_1	-		INT8 external interrupt input pin (1)
	SIN8_1	-		Multi-function serial ch.8 serial data input pin (1)
	IN3_2	-		Input capture ch.3 input pin (2)
	TIOA14_1	-		Base timer ch.14 TIOA output pin (1)
	SPIDAT1	-		HS-SPI data 1 I/O pin
70	P214	-	Q	General-purpose I/O port
	SOT8_1	-		Multi-function serial ch.8 serial data output pin (1)
	IN4_2	-		Input capture ch.4 input pin (2)
	TIOA15_1	-		Base timer ch.15 TIOA I/O pin (1)
	SPISEL0	-		HS-SPI slave select 0 output pin
71	P215	-	Q	General-purpose I/O port
	INT9_1	-		INT9 external interrupt input pin (1)
	SCK8_1	-		Multi-function serial ch.8 clock I/O pin (1)
	IN5_2	-		Input capture ch.5 input pin (2)
	TIOA16_1	-		Base timer ch.16 TIOA output pin (1)
	SPIDAT3	-		HS-SPI data 3 I/O pin
74	P218	-	B	General-purpose I/O port
	AN36	-		ADC analog 36 input pin
	TEXT2_0	-		Free-run timer 2 clock input pin (0)
	TIOB14_0	-		Base timer ch.14 TIOB input pin (0)
75	P219	-	B	General-purpose I/O port
	AN37	-		ADC analog 37 input pin
	TEXT3_0	-		Free-run timer 3 clock input pin (0)
	TIOB15_0	-		Base timer ch.15 TIOB input pin (0)

Pin No.	Pin Name	Polarity	I/O Circuit Type	Function
76	P220	-	B	General-purpose I/O port
	AN38	-		ADC analog 38 input pin
	TX2_0	-		CAN transmission data 2 output pin (0)
	SCS83_0	-		Multi-function serial ch.8 serial chip select 3 output pin (0)
	IN6_2	-		Input capture ch.6 input pin (2)
	TIOB16_0	-		Base timer ch.16 TIOB input pin (0)
77	P222	-	B	General-purpose I/O port
	INT7_0	-		INT7 external interrupt input pin (0)
	AN39	-		ADC analog 39 input pin
	RX2_0	-		CAN reception data 2 input pin (0)
	SIN8_0	-		Multi-function serial ch.8 serial data input pin (0)
	IN7_2	-		Input capture ch.7 input pin (2)
78	P223	-	B	General-purpose I/O port
	AN40	-		ADC analog 40 input pin
	PWU_AN0	-		Partial wakeup ADC analog 0 input pin
	SCS81_0	-		Multi-function serial ch.8 serial chip select 1 output pin (0)
	IN8_2	-		Input capture ch.8 input pin (2)
	AIN9_1	-		QPRC ch.9 AIN input pin (1)
79	P224	-	B	General-purpose I/O port
	AN41	-		ADC analog 41 input pin
	PWU_AN1	-		Partial wakeup ADC analog 1 input pin
	TX0_2	-		CAN transmission data 0 output pin (2)
	SCS80_0	-		Multi-function serial ch.8 serial chip select 0 I/O pin (0)
	IN9_2	-		Input capture ch.9 input pin (2)
	BIN9_1	-		QPRC ch.9 BIN input pin (1)
80	P225	-	B	General-purpose I/O port
	INT0_0	-		INT0 external interrupt input pin (0)
	PWU_AN2	-		Partial wakeup ADC analog 2 input pin
	AN42	-		ADC analog 42 input pin
	RX0_2	-		CAN reception data 0 input pin (2)
	SOT8_0	-		Multi-function serial ch.8 serial data output pin (0)
	SDA8_0	-		I^2C bus ch.8 serial data I/O pin
	IN10_2	-		Input capture ch.10 input pin (2)
	TIOB17_0	-		Base timer ch.17 TIOB input pin (0)
	ZIN9_1	-		QPRC ch.9 ZIN input pin (1)
81	P226	-	B	General-purpose I/O port
	AN43	-		ADC analog 43 input pin
	PWU_AN3	-		Partial wakeup ADC analog 3 input pin
	SCK8_0	-		Multi-function serial ch.8 clock I/O pin (0)
	SCL8_0	-		I^2C bus ch.8 serial clock I/O pin
	IN11_2	-		Input capture ch.11 input pin (2)
	TIOA17_1	-		Base timer ch.17 TIOA I/O pin (1)

Pin No.	Pin Name	Polarity	I/O Circuit Type	Function
87	P229	-	M	General-purpose I/O port
	INT8_0	-		INT8 external interrupt input pin (0)
	AN46	-		ADC analog 46 input pin
	PWU_AN6	-		Partial wakeup ADC analog 6 input pin
	OUT0_0	-		Output compare ch.0 output pin (0)
	TIOA25_0	-		Base timer ch.25 TIOA I/O pin (0)
	PWM1P0	-		SMC ch.0 (P1) output pin
88	P230	-	M	General-purpose I/O port
	AN47	-		ADC analog 47 input pin
	PWU_AN7	-		Partial wakeup ADC analog 7 input pin
	OUT1_0	-		Output compare ch.1 output pin (0)
	TIOA26_0	-		Base timer ch.26 TIOA output pin (0)
	PWM1M0	-		SMC ch.0 (M1) output pin
89	P231	-	M	General-purpose I/O port
	AN48	-		ADC analog 48 input pin
	OUT2_0	-		Output compare ch.2 output pin (0)
	TIOA27_0	-		Base timer ch.27 TIOA I/O pin (0)
	PWM2P0	-		SMC ch.0 (P2) output pin
90	P300	-	M	General-purpose I/O port
	AN49	-		ADC analog 49 input pin
	OUT3_0	-		Output compare ch.3 output pin (0)
	TIOA28_0	-		Base timer ch.28 TIOA output pin (0)
	PWM2M0	-		SMC ch.0 (M2) output pin
91	P301	-	M	General-purpose I/O port
	AN50	-		ADC analog 50 input pin
	OUT4_0	-		Output compare ch.4 output pin (0)
	TIOA18_1	-		Base timer ch.18 TIOA output pin (1)
	PWM1P1	-		SMC ch.1 (P1) output pin
92	P302	-	M	General-purpose I/O port
	AN51	-		ADC analog 51 input pin
	TIOA19_1	-		Base timer ch.19 TIOA I/O pin (1)
	PWM1M1	-		SMC ch.1 (M1) output pin
93	P304	-	M	General-purpose I/O port
	AN52	-		ADC analog 52 input pin
	TEXT4_0	-		Free-run timer 4 clock input pin (0)
	TIOA20_1	-		Base timer ch.20 TIOA output pin (1)
	PWM2P1	-		SMC ch.1 (P2) output pin
94	P305	-	M	General-purpose I/O port
	AN53	-		ADC analog 53 input pin
	TEXT5_0	-		Free-run timer 5 clock input pin (0)
	TIOA29_0	-		Base timer ch.29 TIOA I/O pin (0)
	PWM2M1	-		SMC ch.1 (M2) output pin

Pin No.	Pin Name	Polarity	I/O Circuit Type	Function
97	P307	-	M	General-purpose I/O port
	INT1_0	-		INT1 external interrupt input pin (0)
	AN55	-		ADC analog 55 input pin
	SCS102_0	-		Multi-function serial ch.10 serial chip select 2 output pin (0)
	TIOB18_0	-		Base timer ch.18 TIOB input pin (0)
	PWM1P2	-		SMC ch.2 (P1) output pin
98	P308	-	M	General-purpose I/O port
	AN56	-		ADC analog 56 input pin
	IN0_1	-		Input capture ch.0 input pin (1)
	TIOA28_1	-		Base timer ch.28 TIOA output pin (1)
	PWM1M2	-		SMC ch.2 (M1) output pin
99	P309	-	M	General-purpose I/O port
	AN57	-		ADC analog 57 input pin
	IN1_1	-		Input capture ch.1 input pin (1)
	TIOA29_1	-		Base timer ch.29 TIOA I/O pin (1)
	PWM2P2	-		SMC ch.2 (P2) output pin
100	P312	-	M	General-purpose I/O port
	AN58	-		ADC analog 58 input pin
	SCS101_0	-		Multi-function serial ch.10 serial chip select 1 output pin (0)
	IN2_1	-		Input capture ch.2 input pin (1)
	TIOB25_0	-		Base timer ch.25 TIOB input pin (0)
	PWM2M2	-		SMC ch.2 (M2) output pin
101	P313	-	M	General-purpose I/O port
	INT10_1	-		INT10 external interrupt input pin (1)
	AN59	-		ADC analog 59 input pin
	SOT10_0	-		Multi-function serial ch.10 serial data output pin (0)
	SDA10_0	-		I^2C bus ch.10 serial data I/O pin
	IN3_1	-		Input capture ch.3 input pin (1)
	TIOB26_0	-		Base timer ch.26 TIOB input pin (0)
	PWM1P3	-		SMC ch.3 (P1) output pin
102	P314	-	M	General-purpose I/O port
	AN60	-		ADC analog 60 input pin
	SCK10_0	-		Multi-function serial ch.10 clock I/O pin (0)
	SCL10_0	-		I^2C bus ch.10 serial clock I/O pin
	IN4_1	-		Input capture ch.4 input pin (1)
	TIOB27_0	-		Base timer ch.27 TIOB input pin (0)
	TIOA7_1	-		Base timer ch.7 TIOA I/O pin (1)
	PWM1M3	-		SMC ch.3 (M1) output pin

Pin No.	Pin Name	Polarity	I/O Circuit Type	Function
103	P315	-	M	General-purpose I/O port
	AN61	-		ADC analog 61 input pin
	TX1_1	-		CAN transmission data 1 output pin (1)
	SCS100_0	-		Multi-function serial ch.10 serial chip select 0 I/O pin (0)
	IN5_1	-		Input capture ch.5 input pin (1)
	TIOB28_0	-		Base timer ch.28 TIOB input pin (0)
	TIOA8_1	-		Base timer ch.8 TIOA output pin (1)
	PWM2P3	-		SMC ch.3 (P2) output pin
104	P317	-	M	General-purpose I/O port
	INT11_1	-		INT11 external interrupt input pin (1)
	AN62	-		ADC analog 62 input pin
	RX1_1	-		CAN reception data 1 input pin (1)
	SIN10_0	-		Multi-function serial ch.10 serial data input pin (0)
	TIOB29_0	-		Base timer ch.29 TIOB input pin (0)
	TIOA9_1	-		Base timer ch.9 TIOA I/O pin (1)
	PWM2M3	-		SMC ch.3 (M2) output pin
107	P321	-	R	General-purpose output port
	PWUTRG	-		Partial wakeup trigger output pin
110	TRST	N	J	JTAG test reset input pin
	P322	-		General-purpose output port
111	TDO	-	I	JTAG test data output pin
	P323	-		General-purpose output port
112	TDI	-	D	JTAG test data input pin
	P324	-		General-purpose output port
113	TMS	-	E	JTAG test mode state input pin
114	TCK	-	E	JTAG test clock input pin
115	P327	-	Q	General-purpose I/O port
	WOT	-		RTC output pin
116	NMIX	-	F	Non-maskable interrupt input pin
117	MD	-	C	Mode pin
118	X0	-	G	Main clock oscillation input pin
119	X1	-	G	Main clock oscillation output pin
121	P331	-	Q	General-purpose I/O port
	MCSX3	-		External bus interface chip select 3 output pin
	SGA0_1	-		Sound generator ch.0 SGA output pin (1)
122	P400	-	Q	General-purpose I/O port
	MCSX2	-		External bus interface chip select 2 output pin
	SGO0_1	-		Sound generator ch.0 SGO output pin (1)
123	RSTX	N	F	External reset input pin

Pin No.	Pin Name	Polarity	I/O Circuit Type	Function
127	P401	-	Q	General-purpose I/O port
	TX1_0	-		CAN transmission data 1 output pin (0)
	IN0_0	-		Input capture ch.0 input pin (0)
128	P402	-	L	General-purpose I/O port (Input only. No output.)
	INT2_0	-		INT2 external interrupt input pin (0)
	RX1_0	-		CAN reception data 1 input pin (0)
	IN1_0	-		Input capture ch.1 input pin (0)
	V3	-		LCDC reference voltage V3 input pin
129	P403	-	B	General-purpose I/O port
	IN2_0	-		Input capture ch.2 input pin (0)
	TRACEDATA0	-		Trace data 0 output pin
	V2	-		LCDC reference voltage V2 input pin
	SGA1_1	-		Sound generator ch.1 SGA output pin (1)
130	P404	-	B	General-purpose I/O port
	SCS120_0	-		Multi-function serial ch.12 serial chip select 0 I/O pin (0)
	IN3_0	-		Input capture ch.3 input pin (0)
	TRACEDATA1	-		Trace data 1 output pin
	MAD14	-		External bus interface address bit14 output pin
	V1	-		LCDC reference voltage V1 input pin
	SGO1_1	-		Sound generator ch.1 SGO output pin (1)
131	P405	-	B	General-purpose I/O port
	INT11_0	-		INT11 external interrupt input pin (0)
	SIN12_0	-		Multi-function serial ch.12 serial data input pin (0)
	IN4_0	-		Input capture ch.4 input pin (0)
	TRACEDATA2	-		Trace data 2 output pin
	MAD13	-		External bus interface address bit13 output pin
	V0	-		LCDC reference voltage V0 input pin
	SGA2_1	-		Sound generator ch.2 SGA output pin (1)
132	P406	-	K	General-purpose I/O port
	SOT12_0	-		Multi-function serial ch.12 serial data output pin (0)
	TRACEDATA3	-		Trace data 3 output pin
	MAD12	-		External bus interface address bit12 output pin
	COM0	-		LCDC segment(duty) common 0 output pin
	SGO2_1	-		Sound generator ch.2 SGO output pin (1)

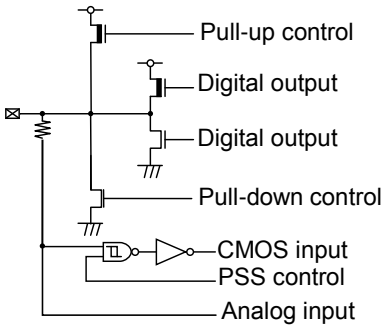
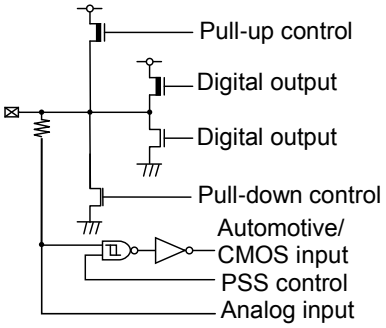
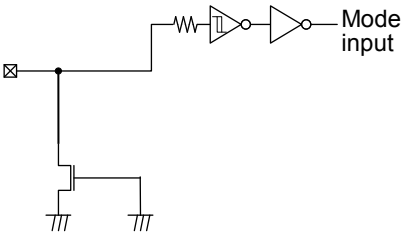
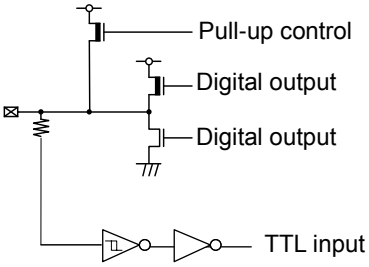
Pin No.	Pin Name	Polarity	I/O Circuit Type	Function
133	P407	-	K	General-purpose I/O port
	SCK12_0	-		Multi-function serial ch.12 clock I/O pin (0)
	SCK10_1	-		Multi-function serial ch.10 clock I/O pin (1)
	TRACEDATA4	-		Trace data 4 output pin
	MAD11	-		External bus interface address bit11 output pin
	COM1	-		LCDC segment(duty) common 1 output pin
134	P408	-	K	General-purpose I/O port
	SIN2_0	-		Multi-function serial ch.2 serial data input pin (0)
	TRACEDATA5	-		Trace data 5 output pin
	TIN18	-		Reload timer ch.18 event input pin (0)
	MAD10	-		External bus interface address bit10 output pin
	COM2	-		LCDC segment(duty) common 2 output pin
135	P409	-	K	General-purpose I/O port
	SOT2_0	-		Multi-function serial ch.2 serial data output pin (0)
	TIOA24_1	-		Base timer ch.24 TIOA output pin (1)
	TRACEDATA6	-		Trace data 6 output pin
	TOT18	-		Reload timer ch.18 output pin (0)
	MAD09	-		External bus interface address bit9 output pin
	COM3	-		LCDC segment(duty) common 3 output pin
136	P411	-	K	General-purpose I/O port
	INT13_1	-		INT13 external interrupt input pin (1)
	SCK2_0	-		Multi-function serial ch.2 clock I/O pin (0)
	SCS101_1	-		Multi-function serial ch.10 serial chip select 1 output pin (1)
	TIOB24_0	-		Base timer ch.24 TIOB input pin (0)
	TRACEDATA7	-		Trace data 7 output pin
	TIN19	-		Reload timer ch.19 event input pin (0)
	MAD08	-		External bus interface address bit8 output pin
	SEG0	-		LCDC segment 0 (Duty) output pin
137	P413	-	K	General-purpose I/O port
	INT14_1	-		INT14 external interrupt input pin (1)
	SCS20_0	-		Multi-function serial ch.2 serial chip select 0 I/O pin (0)
	SCS103_1	-		Multi-function serial ch.10 serial chip select 3 output pin (1)
	TOT19	-		Reload timer ch.19 output pin (0)
	MAD07	-		External bus interface address bit7 output pin
	SEG1	-		LCDC segment 1 (Duty) output pin
138	P414	-	K	General-purpose I/O port
	SCS21_0	-		Multi-function serial ch.2 serial chip select 1 output pin (0)
	TIN32	-		Reload timer ch.32 event input pin (0)
	MAD06	-		External bus interface address bit6 output pin
	SEG2	-		LCDC segment 2 (Duty) output pin

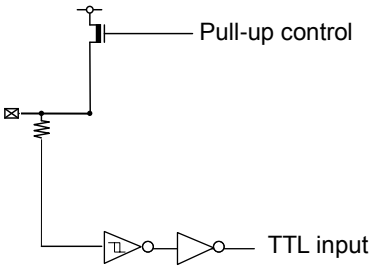
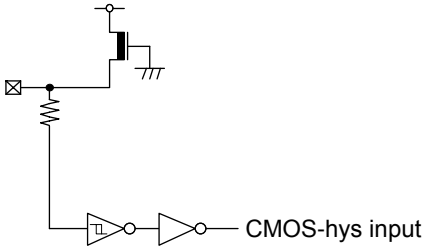
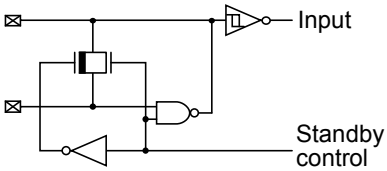
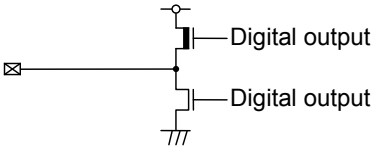
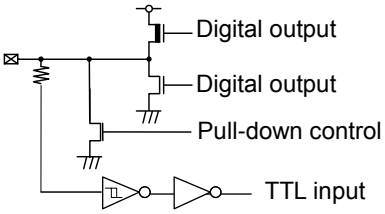
Pin No.	Pin Name	Polarity	I/O Circuit Type	Function
139	P416 SIN10_1 IN5_0 TIOA22_1 TOT32 MAD05 SEG3	- - - - - - -	K	General-purpose I/O port Multi-function serial ch.10 serial data input pin (1) Input capture ch.5 input pin (0) Base timer ch.22 TIOA output pin (1) Reload timer ch.32 output pin (0) External bus interface address bit5 output pin LCDC segment 3 (Duty) output pin
140	P417 INT15_1 SOT10_1 TIOA23_1 TIN33 MAD04 SEG4	- - - - - - -	K	General-purpose I/O port INT15 external interrupt input pin (1) Multi-function serial ch.10 serial data output pin (1) Base timer ch.23 TIOA I/O pin (1) Reload timer ch.33 event input pin (0) External bus interface address bit4 output pin LCDC segment 4 (Duty) output pin
141	P418 INT14_0 SCS22_0 TIOB23_0 TOT33 MAD03 SEG5	- - - - - - -	K	General-purpose I/O port INT14 external interrupt input pin (0) Multi-function serial ch.2 serial chip select 2 output pin (0) Base timer ch.23 TIOB input pin (0) Reload timer ch.33 output pin (0) External bus interface address bit3 output pin LCDC segment 5 (Duty) output pin
142	P420 SCK2_1 TRACECLK MAD02 SEG6	- - - - -	K	General-purpose I/O port Multi-function serial ch.2 clock I/O pin (1) Trace clock External bus interface address bit2 output pin LCDC segment 6 (Duty) output pin
143	P421 INT12_1 SIN2_1 TRACECTL MAD01 SEG7	- - - - - -	K	General-purpose I/O port INT12 external interrupt input pin (1) Multi-function serial ch.2 serial data input pin (1) Trace control External bus interface address bit1 output pin LCDC segment 7 (Duty) output pin
42	AVCC0	-	-	Analog power supply pin for AD converter unit 0
84	AVCC1	-	-	Analog power supply pin for AD converter unit 1
43	AVRH0	-	-	Upper-limit reference voltage pin for AD converter unit 0
83	AVRH1	-	-	Upper-limit reference voltage pin for AD converter unit 1
44	AVSS0 AVRL0	-	-	GND pin for AD converter unit 0 Lower-limit reference voltage pin for AD converter unit 0
82	AVSS1 AVRL1	-	-	GND pin for AD converter unit 1 Lower-limit reference voltage pin for AD converter unit 1

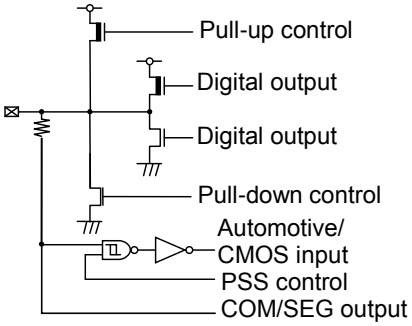
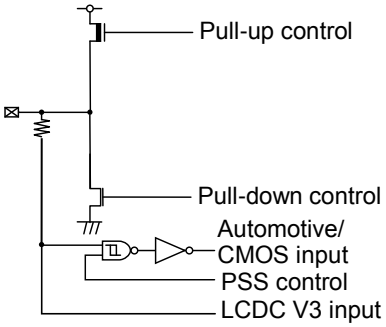
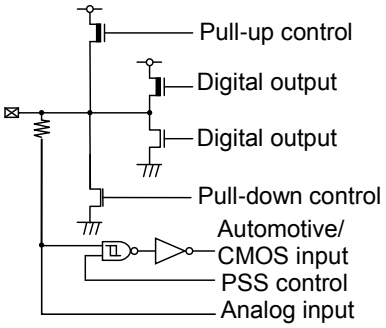
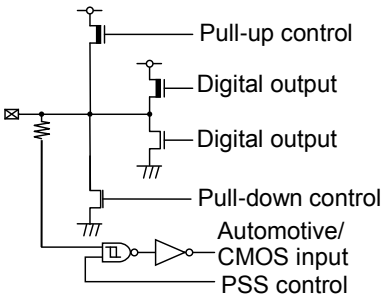
Pin No.	Pin Name	Polarity	I/O Circuit Type	Function
38	C	-	-	External capacity connection output pin
126				
36	VCC	-	-	Power supply pin
72				
109				
124				
144				
1	VSS	-	-	GND
37				
73				
108				
120				
125				
85	DVCC	-	-	Power Supply pin for SMC high current
95				
105				
86	DVSS	-	-	GND pin for SMC high current
96				
106				

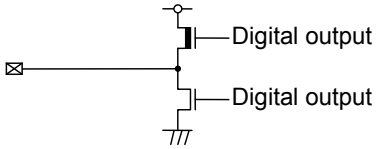
11. I/O Circuit Type

This section explains I/O circuit types.

Type	Circuit	Overview
A		– General-purpose I/O port with analog input – Output of 1 mA or 2 mA selectable – 50 kΩ with pull-up resistor control – 50 kΩ with pull-down resistor control – CMOS hysteresis input
B		– General-purpose I/O port with analog input – Output of 1 mA or 2 mA selectable – 50 kΩ with pull-up resistor control – 50 kΩ with pull-down resistor control – Automotive/CMOS hysteresis input selectable
C		– Mode input – CMOS hysteresis input
D		– JTAG – General-purpose output port – Output of 2 mA – 50 kΩ with pull-up resistor control – TTL input

Type	Circuit	Overview
E		– JTAG – 50 kΩ with pull-up resistor control – TTL input
F		– CMOS hysteresis input – 50 kΩ with pull-up resistor
G		– Main oscillation I/O
I		– JTAG – Output of 2 mA
J		– JTAG – General-purpose output port – Output of 2 mA – 50 kΩ with pull-down resistor control – TTL input

Type	Circuit	Overview
K		– General-purpose I/O port with COM/SEG output – Output of 1 mA or 2 mA selectable – 50 kΩ with pull-up resistor control – 50 kΩ with pull-down resistor control – Automotive/CMOS hysteresis input selectable
L		– General-purpose I/O port with LCDC V3 input – 50 kΩ with pull-up resistor control – 50 kΩ with pull-down resistor control – Automotive/CMOS hysteresis input selectable
M		– General-purpose I/O port with analog input – Output of 1 mA or 2 mA or 30mA selectable – 50 kΩ with pull-up resistor control – 50 kΩ with pull-down resistor control – Automotive/CMOS hysteresis input selectable
Q		– General-purpose I/O port – Output of 1 mA or 2 mA selectable – 50 kΩ with pull-up resistor control – 50 kΩ with pull-down resistor control – Automotive/CMOS hysteresis input selectable

Type	Circuit	Overview
R		− Output of 2 mA

CHAPTER 2: CPU

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

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

CPU-T00P00R01L02-E1-GBB00

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 not equipped with a 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)

Feature	Options	Sub-options	Configuration
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
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	2x32KB
		Interleaved on 64-bit granularity in memory	Interleaved on 64-bit granularity in memory
		Adjacent in memory	

Feature	Options	Sub-options	Configuration
Instruction endianness	Little-endian	-	Little-endian
	Pin-configured	Little-endian	
		Big-endian	
Floating point (VFP)	No FPU	-	No FPU
	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
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

Feature	Options	Sub-options	Configuration
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

Feature	Options	Sub-options	Configuration
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	-	-
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**

MODE-T00P00R01L01-E1-GBB00

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-modes.

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

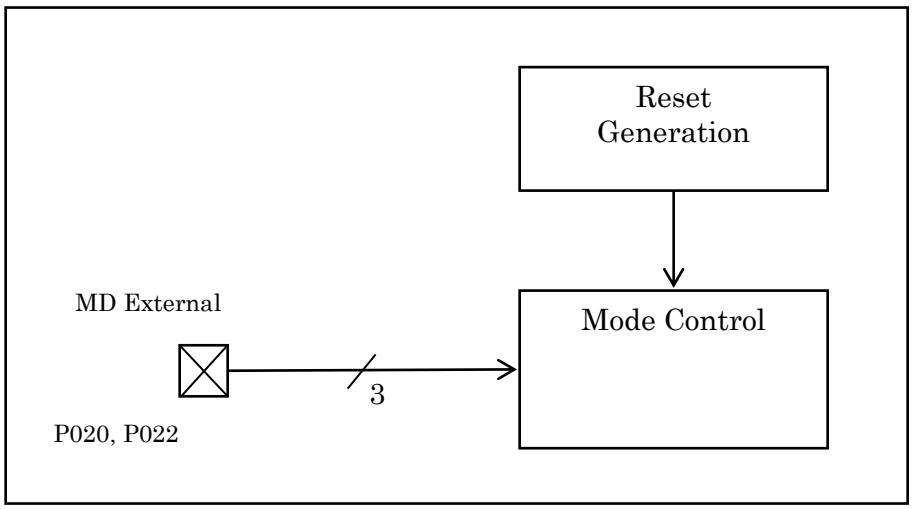


Table 2-2 Operation Mode

Operation Mode	MD Pin	P020	P022
User Mode	1	-	-
Serial Write Mode (Sync)	0	1	0
Serial Write Mode(Async)	0	1	1
JTAG Boundary Scan Mode	0	0	0

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 List of Register Mirror Area

Address of Register Mirror Area	Mirrored Peripheral	Mirror Area Behavior	PPU
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	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-T00P00R01L03-E1-GBB02

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) RSTX pin + MD pin simultaneous assert reset (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) 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 reset4 (PD4Rn) PowerDomain reset 5 (PD5Rn) PowerDomain reset6 (PD6Rn)

Notes:

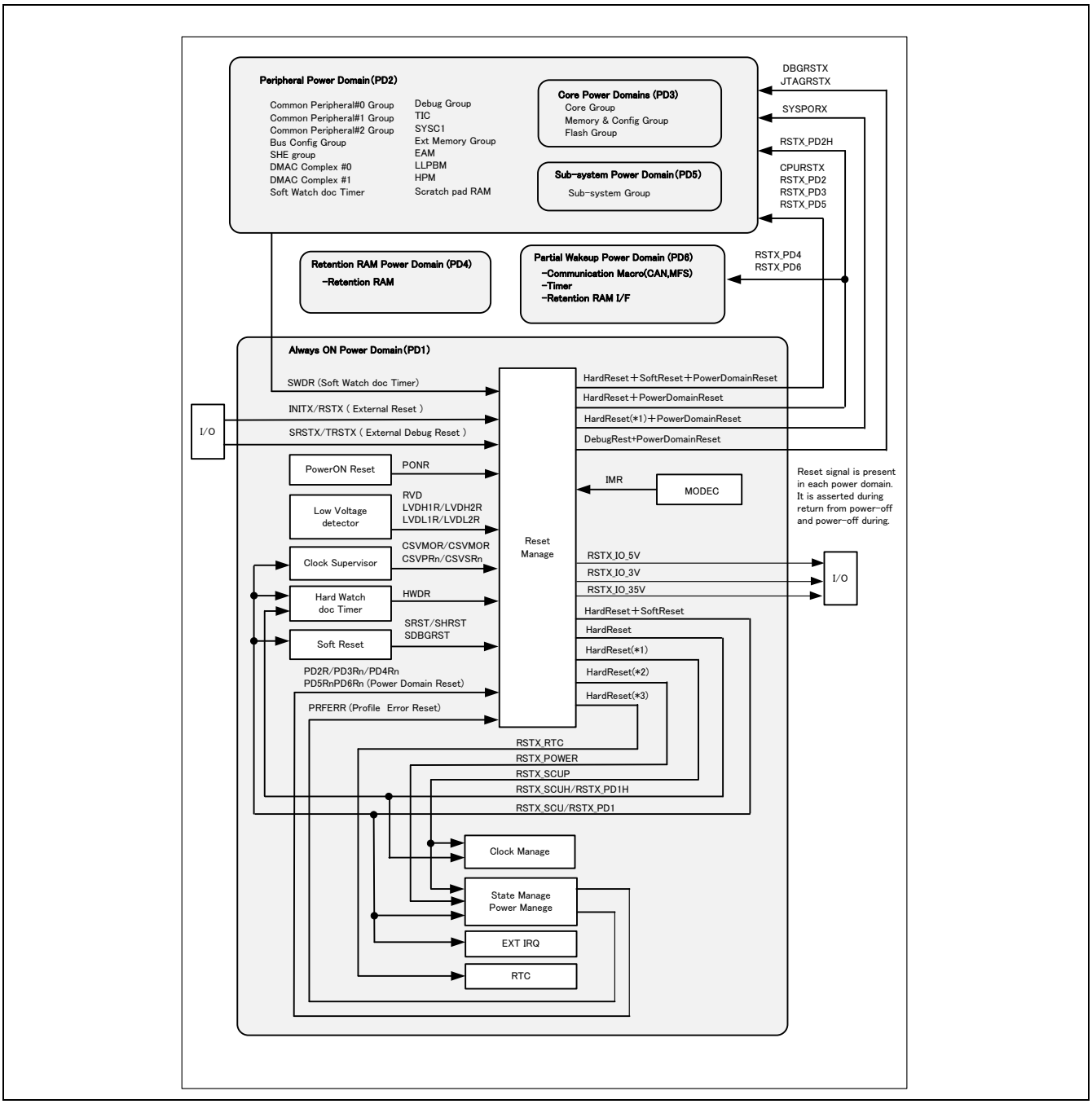
- *nSRST pin input reset cannot be used. Therefore, please ignore the description about nSRST in RESET chapter.*
- *Following reset factors are not supported in this product: Extended internal power supply low-voltage detection reset (LVDL2R), Extended external power supply low-voltage detection reset (LVDH2R), Sub clock supervisor reset (CSVSOR), nSRST pin input reset (SRSTX), Power domain reset 3 (PD3Rn) and Power domain reset 5 (PD5Rn)*

2. Configuration

This section shows block diagrams of reset system.

Following block diagram show connections between the control logics and PowerDomains.

Figure 2-1 Reset System Block Diagram



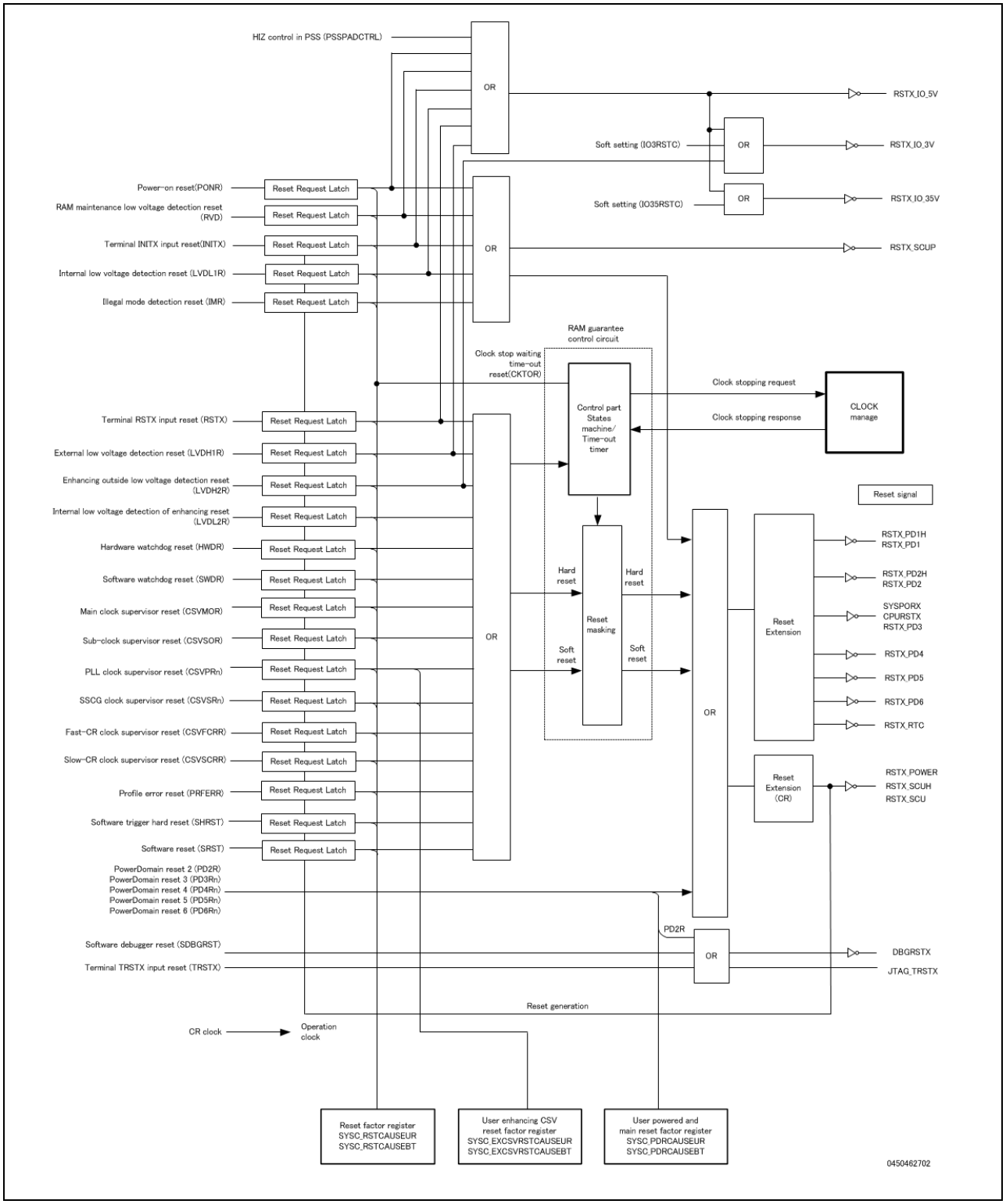
(*1) It consists of PONR/INITX/RVD/LVDL1R.

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

(*3) It consists of PONR/INITX/RVD/LVDL1R/HWDR/CSVMOR/CSVMOR.

Following block diagram shows the control logic of the reset.

Figure 2-2 RST manage Block Diagram



Notes:

- *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 "CHAPTER:INTRODUCTION".*
- *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 "CHAPTER:INTRODUCTION".*
- *Fast-CR clock supervisor reset (CSVFCRR) cannot be used. Therefore, please ignore the description about Fast-CR clock supervisor reset (CSVFCRR) in RESET chapter.*
- *Slow-CR clock supervisor reset (CSVSCRR) cannot be used. Therefore, please ignore the description about Slow-CR clock supervisor reset (CSVSCRR) in RESET chapter.*

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 RSTX pin + MD pin simultaneous assert reset (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 detection reset (LVDL2R)
3.1.8 Extended external Power supply low-voltage detection 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 Profile error reset (PRFERR)
3.1.18 Software trigger hard reset (SHRST)
3.1.19 Software reset (SRST)
3.1.20 TRSTX pin input reset (TRSTX)
3.1.21 Software debugger reset (SDBGRST)
3.1.22 PowerDomain reset2 (PD2R)
3.1.23 PowerDomain reset3 (PD3Rn)
3.1.24 PowerDomain reset4 (PD4Rn)
3.1.25 PowerDomain reset5 (PD5Rn)
3.1.26 PowerDomain reset6 (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 RAM Reset

It explains RAM Retention Low-voltage Detection RAM reset.

Table 3-2 RAM Retention Low-Voltage Detection RAM 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. RSTX Pin + MD Pin Simultaneous assert Reset

It explains RSTX pin + MD pin simultaneous assert reset.

Table 3-3 RSTX Pin + MD Pin Simultaneous assert Reset

Occurrence factor	It is issued by simultaneously inputting “L” level to RSTX pin and inputting “L” level to MD pin.
Release factor	It releases it by inputting “H” level to the terminal RSTX
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.*

Note:

- *This product does not support this reset factor.*

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.*

Note:

- *This product does not support PLL1, PLL2 and PLL3.*

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.*

Note:

- *This product does not support SSCG1, SSCG2 andSSCG3.*

3.1.17. Profile Error Reset

It explains profile error reset.

Table 3-17 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.*

3.1.18. Software Trigger Hard Reset

It explains software trigger hard reset.

Table 3-18 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.*

3.1.19. Software Reset

It explains software reset.

Table 3-19 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.*
- *Initialization target registers are described outline only. Please refer to the register description of the appropriate chapter about the details.*

3.1.20. TRSTX Pin Input Reset

It explains TRSTX Pin Input

Table 3-20 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.*

3.1.21. Software Debugger Reset

It explains software debugger reset.

Table 3-21 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.*

3.1.22. PowerDomain Reset 2

It explains PowerDomain reset 2.

Table 3-22 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.*

3.1.23. PowerDomain Reset 3

It explains PowerDomain reset 3.

Table 3-23 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.*

Note:

- *This product does not support this reset factor.*

3.1.24. PowerDomain Reset 4

It explains PowerDomain reset 4.

Table 3-24 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.*

3.1.25. PowerDomain Reset 5

It explains PowerDomain reset 5.

Table 3-25 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.*

Note:

- *This product does not support this reset factor.*

3.1.26. PowerDomain Reset 6

It explains PowerDomain reset 6.

Table 3-26 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
- Peripheral 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 3-27 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 RSTX pin + MD pin simultaneous assert reset 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 RSTX pin + MD pin simultaneous assert reset 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 RSTX pin + MD pin simultaneous assert reset 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 3-28 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 RSTX pin + MD pin simultaneous assert reset
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 RSTX pin + MD pin simultaneous assert reset Extended external power supply low-voltage detection reset Soft setting (*1)
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 RSTX pin + MD pin simultaneous assert reset Soft setting (*2)

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 IO3RSTCNT in system special setting register (SYSC0_SPECFGR).*
- *(*2) It is controlled by IO35RSTCNT in system special setting register (SYSC0_SPECFGR)*

3.2.3. Reset of 1.2V External Power Supply Control

This section explains the reset of the 1.2V external power supply control.

Table 3-29 Reset of 1.2V 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 Extended internal power supply low-voltage detection reset RSTX pin + MD pin simultaneous assert reset Illegal mode detection reset Hardware watchdog reset
		Standby transition reset	PowerDomain reset 2

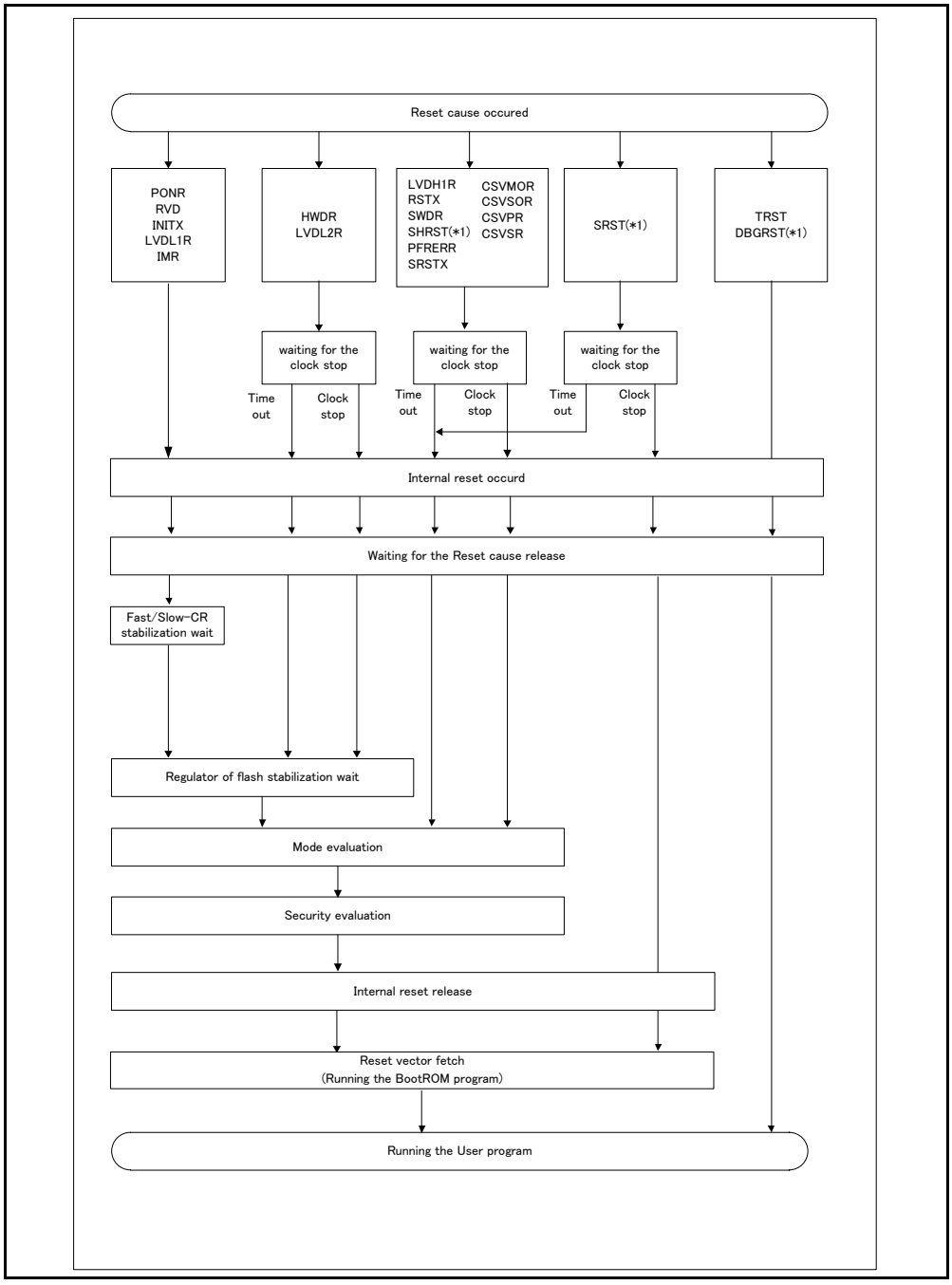
Note:

- *This product does not support this feature.*

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 to be secured. 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. Regulator of flash stabilization wait

   Wait for Flash macro's power supply to stabilize.

7. Mode evaluation

   Operation mode is set and announced to each hardware.

8. Security evaluation

   Security is judged and announced to each hardware.

9. Internal reset release

   Release of internal reset is done.

10. Reset vector fetch (Running the Boot ROM program)

    CPU starts to fetch the reset vectors. CPU executes the program Boot ROM.

11. 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 3-30 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 RSTX pin + MD pin simultaneous assert reset Illegal mode detection reset	Fast CR Slow CR FLASH 1.2V external power supply control	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 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)

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 guarantees 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) 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.*

Note:

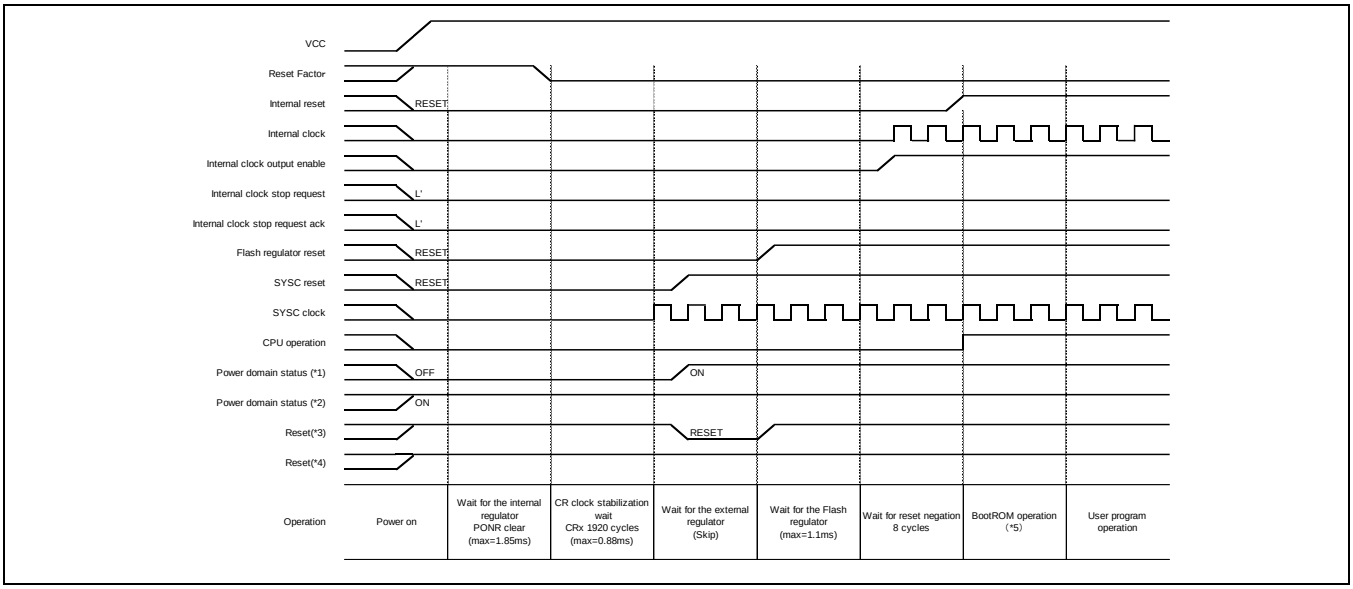
- *Following reset factors are not supported in this product: Extended internal power supply low-voltage detection reset (LVDL2R), Extended external power supply low-voltage detection reset (LVDH2R), Sub clock supervisor reset (CSVSOR), nSRST pin input reset (SRSTX), Power domain reset 3 (PD3Rn) and Power domain reset 5 (PD5Rn)*

Note:

- *This product does not use external 1.2V power supply.*

3.4.1. Hard Reset Operation

Figure 3-2 Operation at Power-on Reset



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

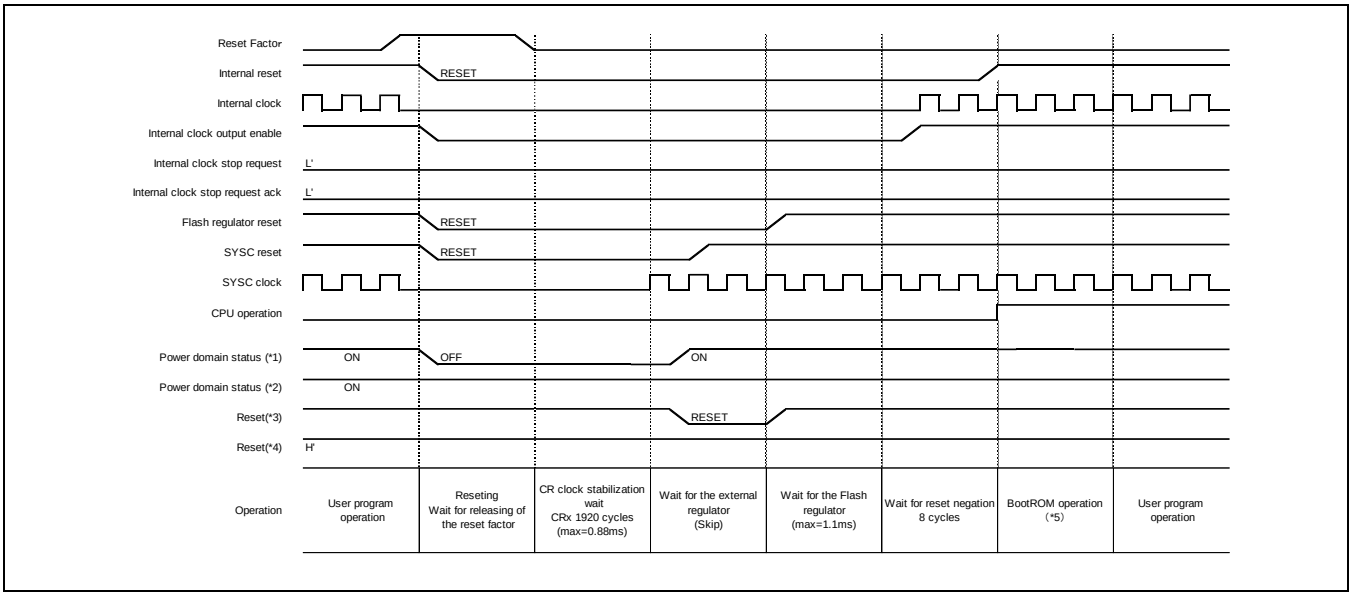
(*2) Internal power supply PowerDomain area status (PD4,PD6)

(*3) External power supply PowerDomain reset

(*4) Internal power supply PowerDomain reset

(*5) Processing time: (Example) 2.41ms at SHE enable and BDR_SBMM=0

Figure 3-3 Operation at Internal Power Supply Low-Voltage Detection Reset or RSTX Pin + MD Pin Simultaneous Assert Reset or RAM Retention Low-Voltage Detection RAM Reset or Illegal Mode Detection Reset



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

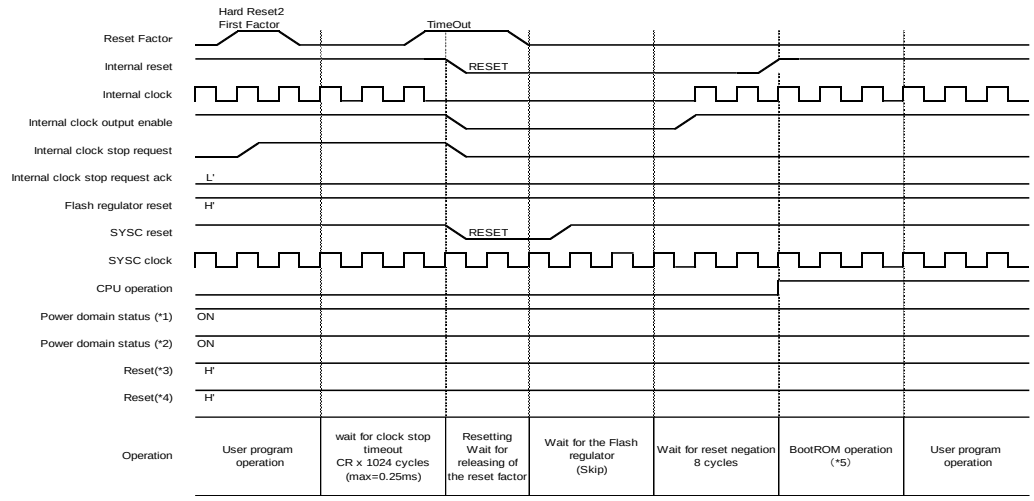
(*2) Internal power supply PowerDomain area status (PD4,PD6)

(*3) External power supply PowerDomain reset

(*4) Internal power supply PowerDomain reset

(*5) Processing time: (Example) 2.41ms at SHE enable and BDR_SBMM=0

Figure 3-4 Operation at Clock Stop Wait Timeout Reset



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

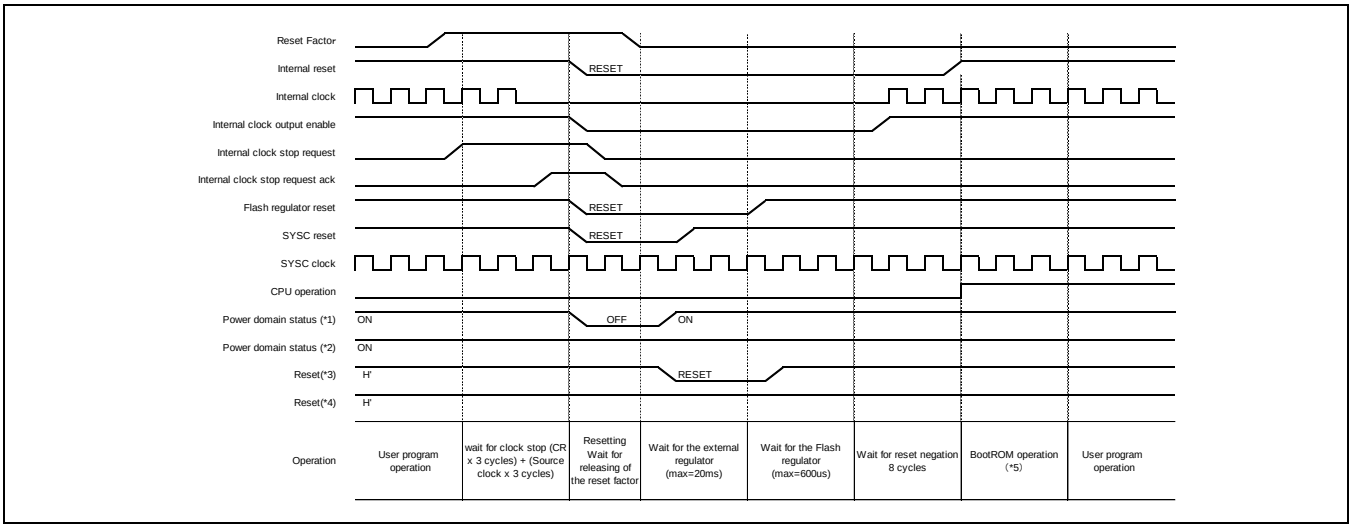
(*2) Internal power supply PowerDomain area status (PD4,PD6)

(*3) External power supply PowerDomain reset

(*4) Internal power supply PowerDomain reset

(*5) Processing time: (Example) 2.41ms at SHE enable and BDR_SBMM=0

Figure 3-5 Operation at Hardware Watchdog Reset or Extended Internal Power Supply Low-Voltage Reset



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

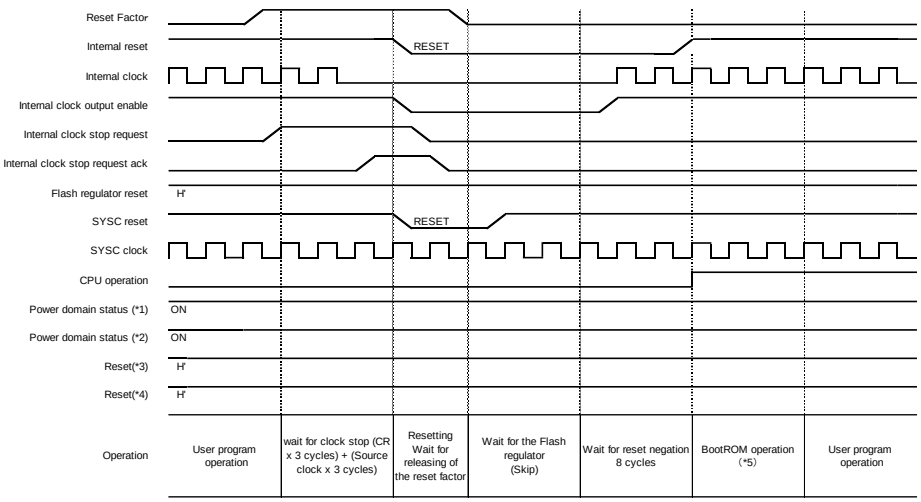
(*2) Internal power supply PowerDomain area status (PD4,PD6)

(*3) External power supply PowerDomain reset

(*4) Internal power supply PowerDomain reset

(*5) Processing time: (Example) 2.41ms at SHE enable and BDR_SBMM=0

Figure 3-6 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

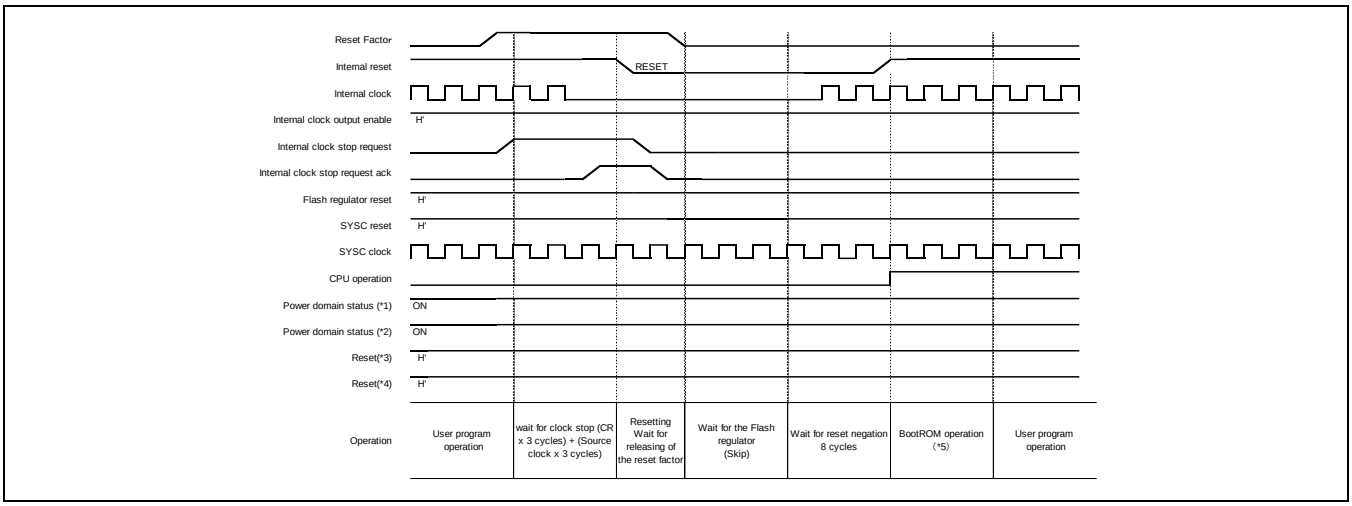
(*5) Processing time: (Example) 2.41ms at SHE enable and BDR_SBMM=0

Notes:

- *The clocks in figure do not indicate actual numbers of clock cycle of the operations.*

3.4.2. Soft Reset Operation

Figure 3-7 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) 2.41ms at SHE enable and BDR_SBMM=0

Note:

- *The clocks in figure do not indicate actual numbers of clock cycle of the operations.*

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:

- *Base_Address is "0xB060_0380".Reserved bits are always read "0"; Write "0" when writing.*
- *Protection key code input is required to write the registers.*
- *It results bus error response for the violation to the rules.*
- *"*": 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

Bit	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

Bit	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

Bit	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 RSTX pin + MD pin simultaneous assert reset(INITX). Protection key code input is required for writing.

Register Structure

Bit	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit 24
Field	LVDL2R	LVDL1R	Reserved		CSVSR0	CSVPR0	CSVSOR	CSVMOR
R/W attribute	R,W	R,W	R0,W0		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.

Bit29:28 Reserved

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 a bit that confirms the software reset detection. Please clear by writing "0" before 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.

When any of the following reset (RVD, INITX, IMR, LVDL1R, HWDR) happens with PSS mode shutdown (PD2:OFF), This bit takes the following value.

SWDR value

Before the reset	After the reset
0	X
1	1

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: RSTX Pin + MD Pin Simultaneous assert Reset Detection Bit

This bit is used to confirm detection of the RSTX pin + MD pin simultaneous assert 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 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:

- *Following reset factors are not supported in this product: Extended internal power supply low-voltage detection reset (LVDL2R), Extended external power supply low-voltage detection reset (LVDH2R), Sub clock supervisor reset (CSVSOR), nSRST pin input reset (SRSTX), Power domain reset 3 (PD3Rn) and Power domain reset 5 (PD5Rn)*
- *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 CSVSR 3: 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 CSVSR 2: 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 CSVSR 1: 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:

- *Following reset factors are not supported in this product: Extended internal power supply low-voltage detection reset (LVDL2R), Extended external power supply low-voltage detection reset (LVDH2R), Sub clock supervisor reset (CSVSOR), nSRST pin input reset (SRSTX), Power domain reset 3 (PD3Rn) and Power domain reset 5 (PD5Rn)*
- *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:

- *Following reset factors are not supported in this product: Extended internal power supply low-voltage detection reset (LVDL2R), Extended external power supply low-voltage detection reset (LVDH2R), Sub clock supervisor reset (CSVSOR), nSRST pin input reset (SRSTX), Power domain reset 3 (PD3Rn) and Power domain reset 5 (PD5Rn)*
- *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 RSTX pin + MD pin simultaneous assert reset(INITX). Protection key code input is required for writing.

Register Structure

Bit	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit 24
Field	LVDL2R	LVDL1R	Reserved		CSVSR0	CSVPR0	CSVSOR	CSVMOR
R/W attribute	R,W	R,W	R0,W0		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.

Bit29:28 Reserved

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.

When any of the following reset (RVD, INITX, IMR, LVDL1R, HWDR) happens with PSS mode shutdown (PD2:OFF), This bit takes the following value.

SWDR value

Before the reset	After the reset
0	X
1	1

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: RSTX Pin + MD Pin Simultaneous assert Reset Detection Bit

This bit is used to confirm detection of the RSTX pin + MD pin simultaneous assert 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 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:

- *Following reset factors are not supported in this product: Extended internal power supply low-voltage detection reset (LVDL2R), Extended external power supply low-voltage detection reset (LVDH2R), Sub clock supervisor reset (CSVSOR), nSRST pin input reset (SRSTX), Power domain reset 3 (PD3Rn) and Power domain reset 5 (PD5Rn)*
- *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:

- *Following reset factors are not supported in this product: Extended internal power supply low-voltage detection reset (LVDL2R), Extended external power supply low-voltage detection reset (LVDH2R), Sub clock supervisor reset (CSVSOR), nSRST pin input reset (SRSTX), Power domain reset 3 (PD3Rn) and Power domain reset 5 (PD5Rn)*
- *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 PD5R1: 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:

- *Following reset factors are not supported in this product: Extended internal power supply low-voltage detection reset (LVDL2R), Extended external power supply low-voltage detection reset (LVDH2R), Sub clock supervisor reset (CSVSOR), nSRST pin input reset (SRSTX), Power domain reset 3 (PD3Rn) and Power domain reset 5 (PD5Rn)*
- *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 PD5RS1: 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:

- *Following reset factors are not supported in this product: Extended internal power supply low-voltage detection reset (LVDL2R), Extended external power supply low-voltage detection reset (LVDH2R), Sub clock supervisor reset (CSVSOR), nSRST pin input reset (SRSTX), Power domain reset 3 (PD3Rn) and Power domain reset 5 (PD5Rn)*
- *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-T00P00R01L04-E-GBB03

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 cock
- Main clock/Main clock * 1/2
- PLL0 clock
- SSCG PLL0 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
- TRC 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)

- 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) 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

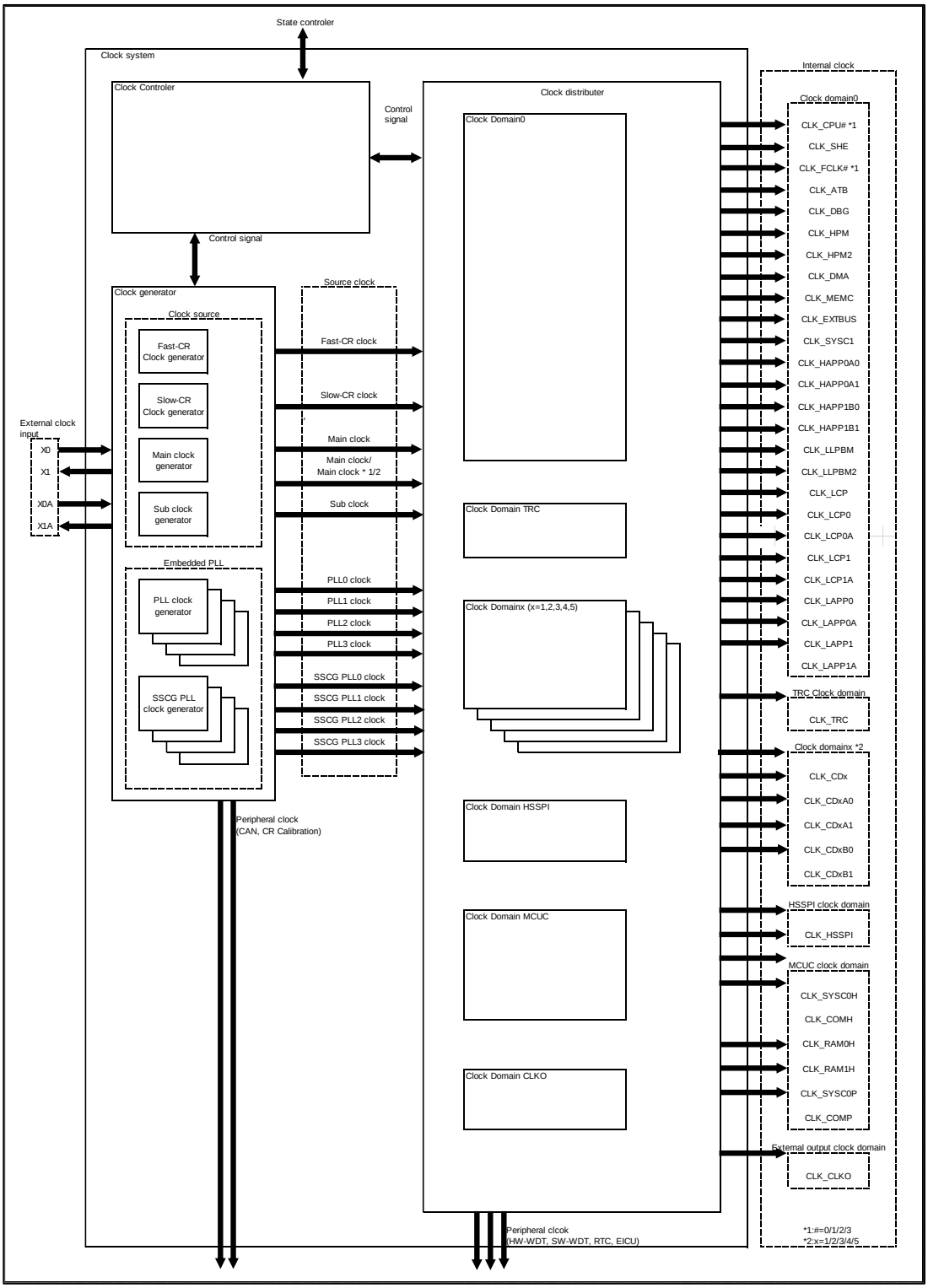
2. Configuration and Block Diagram

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
- PLLx clock (x=0)
- SSCG PLLx clock (x=0)

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

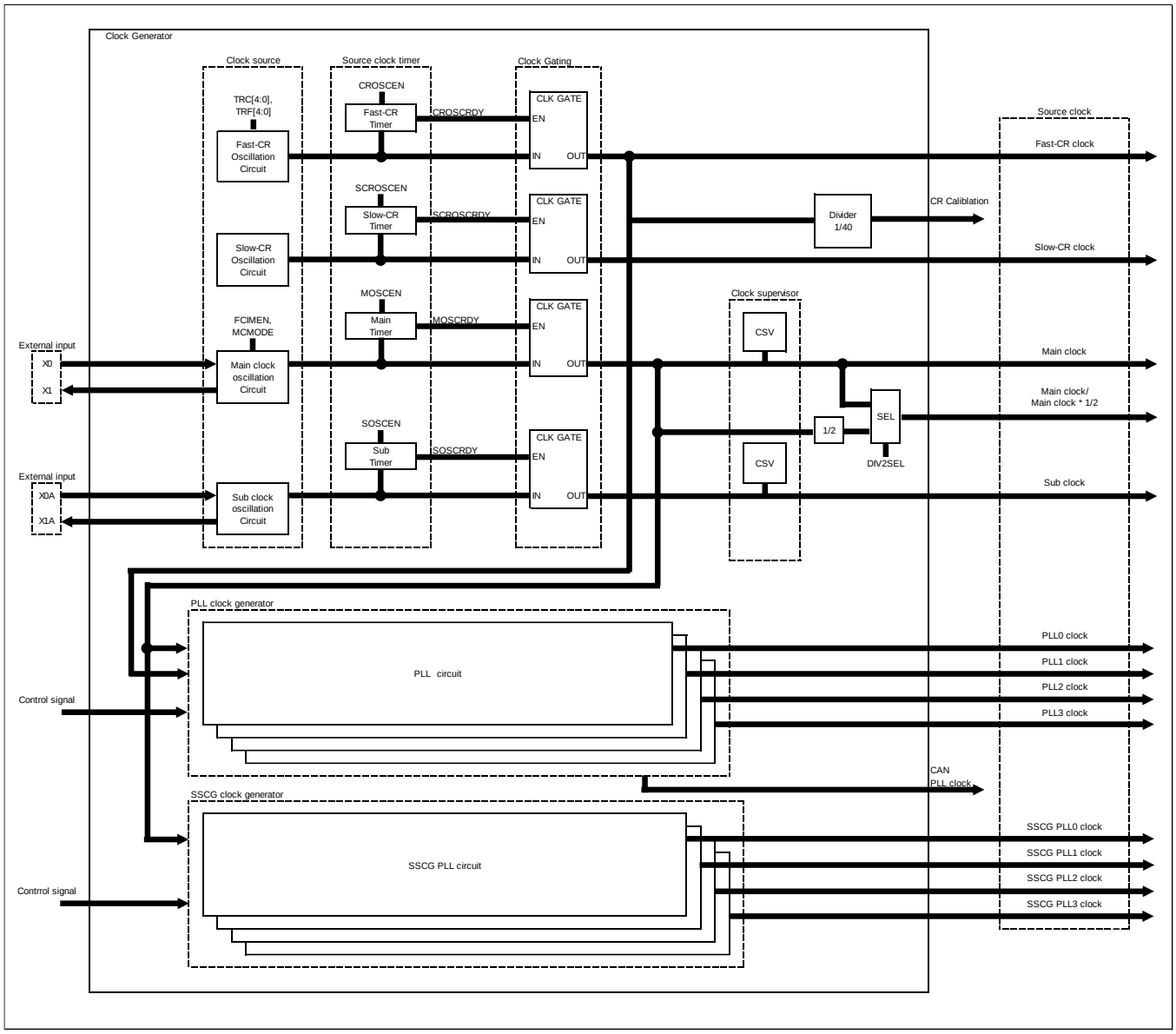
Clock Distributer
Internal clocks generate from dividing into domain clock.

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.

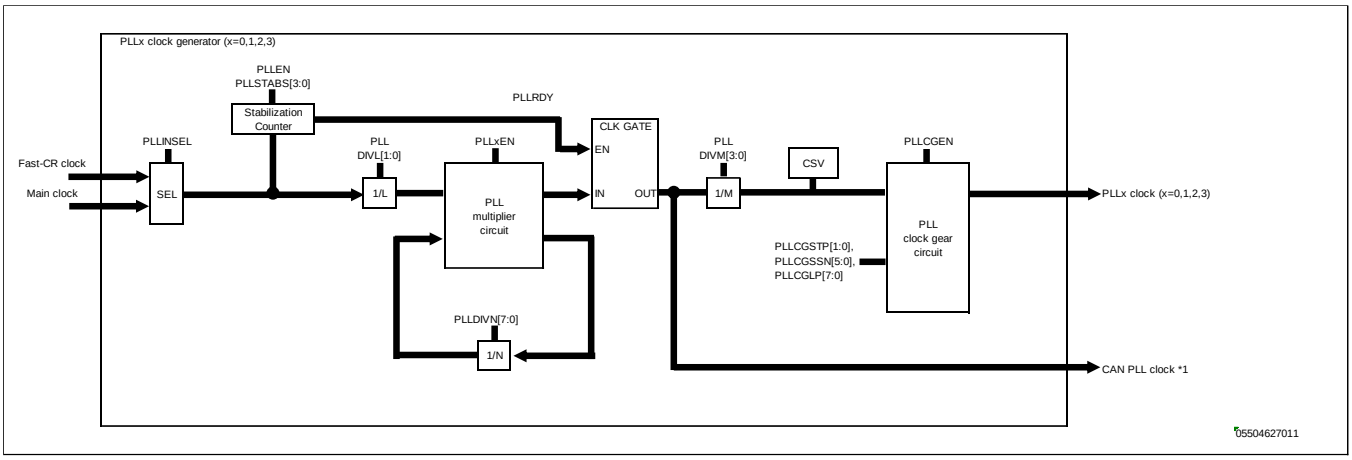
Figure 2-2 Clock Generator Block Diagram



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

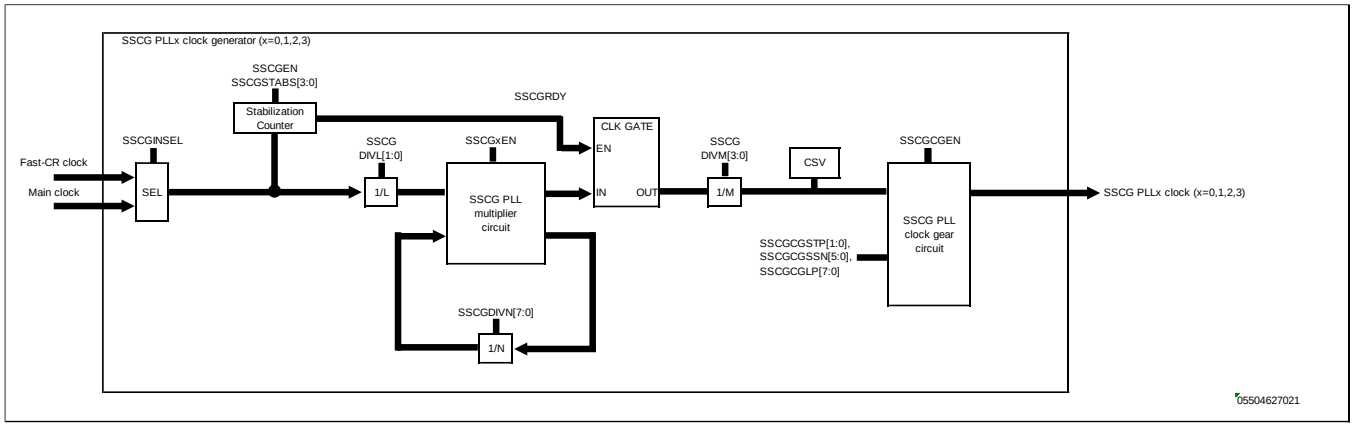


*1 : Only "PLL0" supports the CAN PLL clock.

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 Distributer Configuration and Block Diagram

This section shows the block diagram of the clock distributer.

Figure 2-5 Clock Domain0 Block Diagram

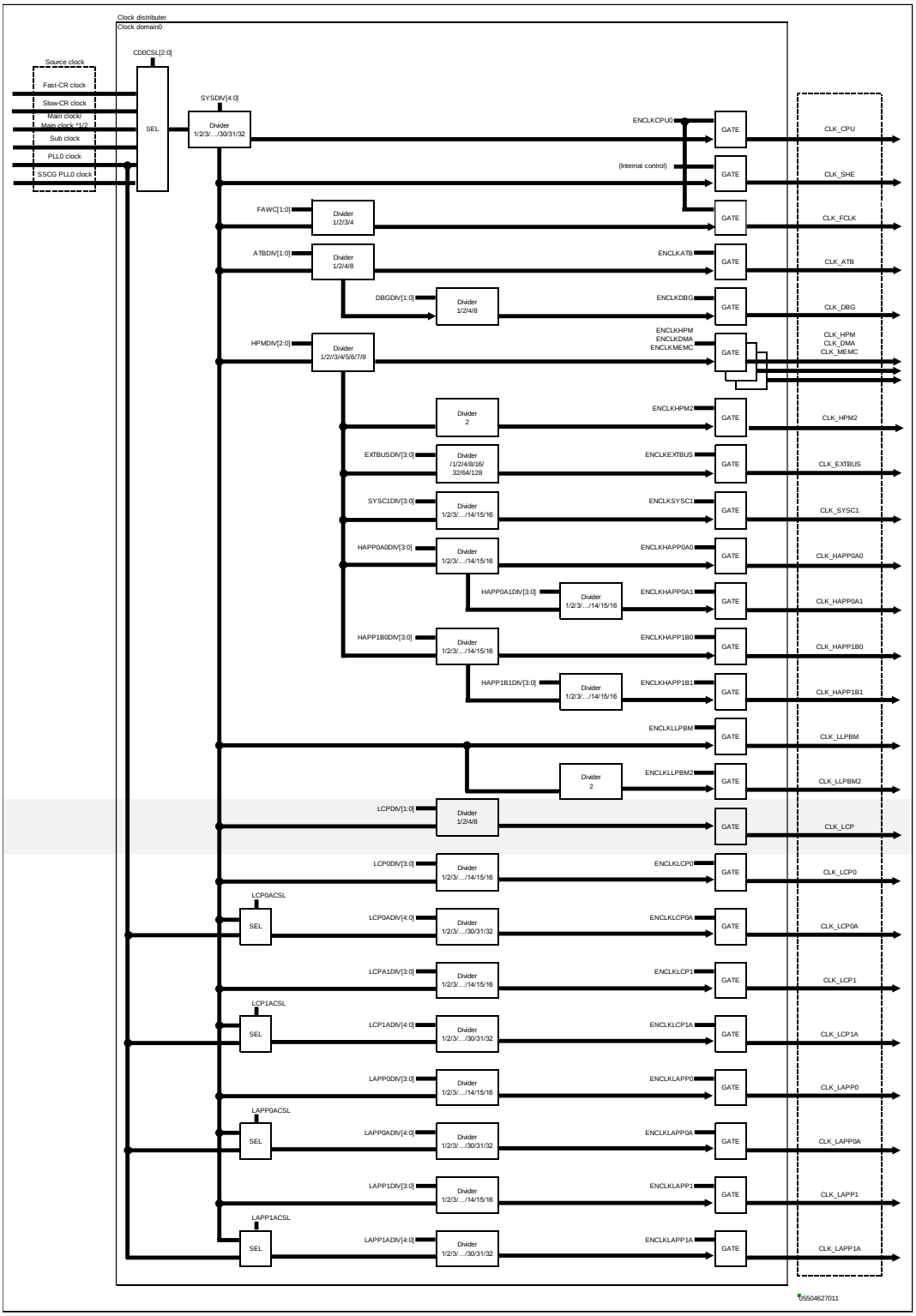


Figure 2-6 Clock Domain TRC Block Diagram

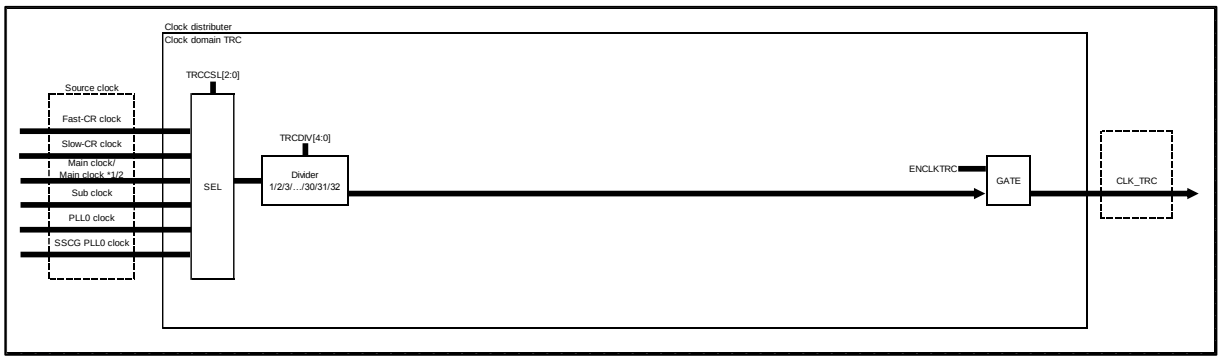


Figure 2-7 Clock Domainx(X=1,2,3,4,5) Block Diagram

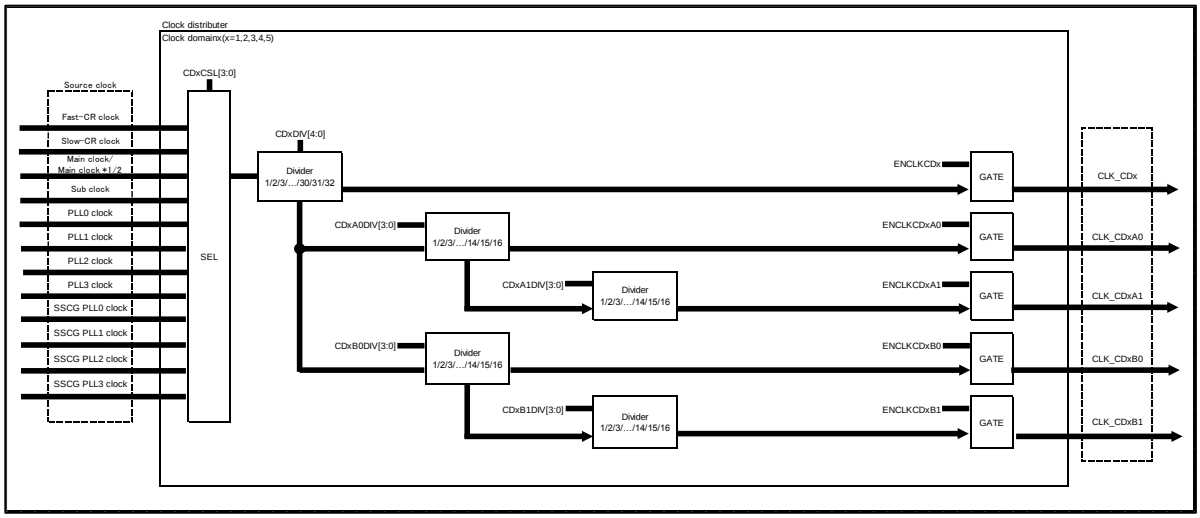


Figure 2-8 Clock Domain HSSPI Block Diagram

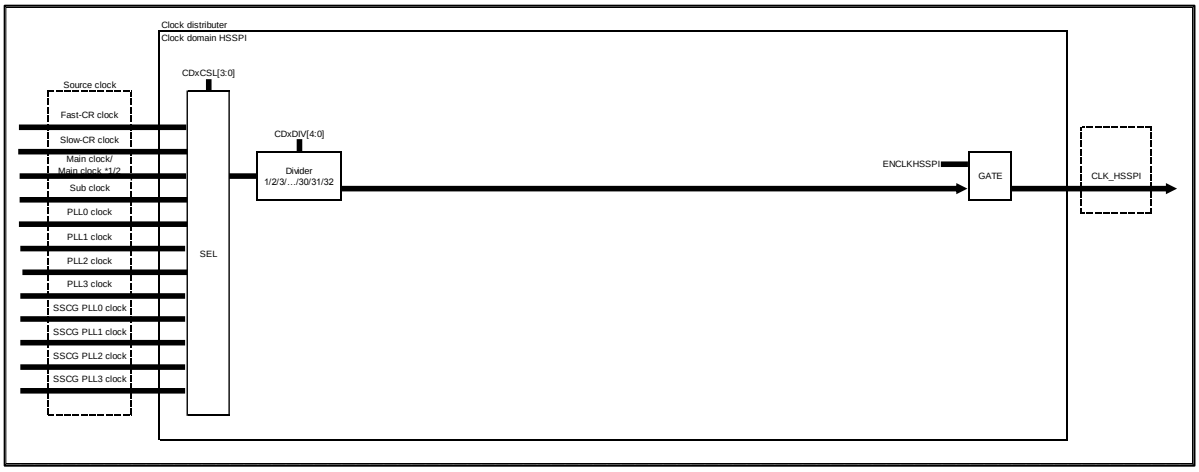


Figure 2-9 Clock Domain MCUC Block Diagram

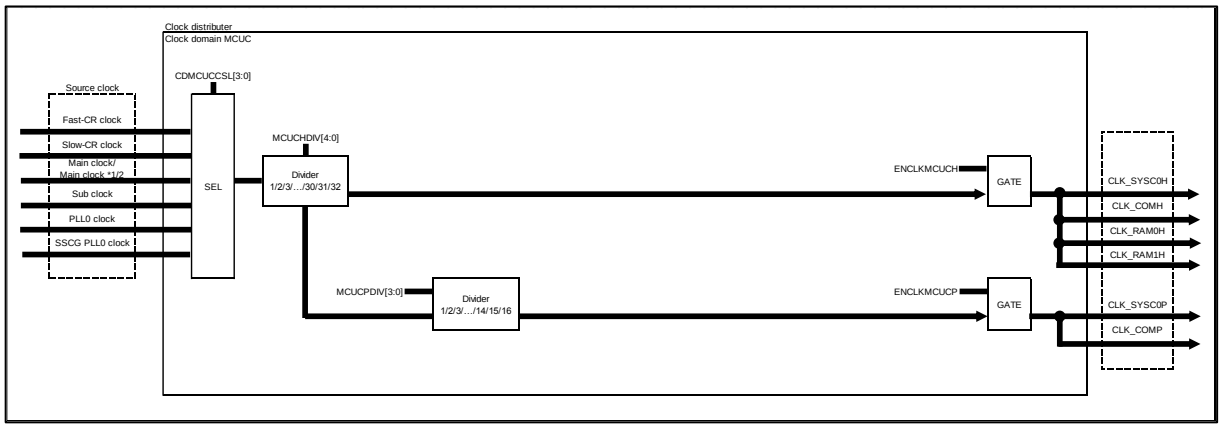
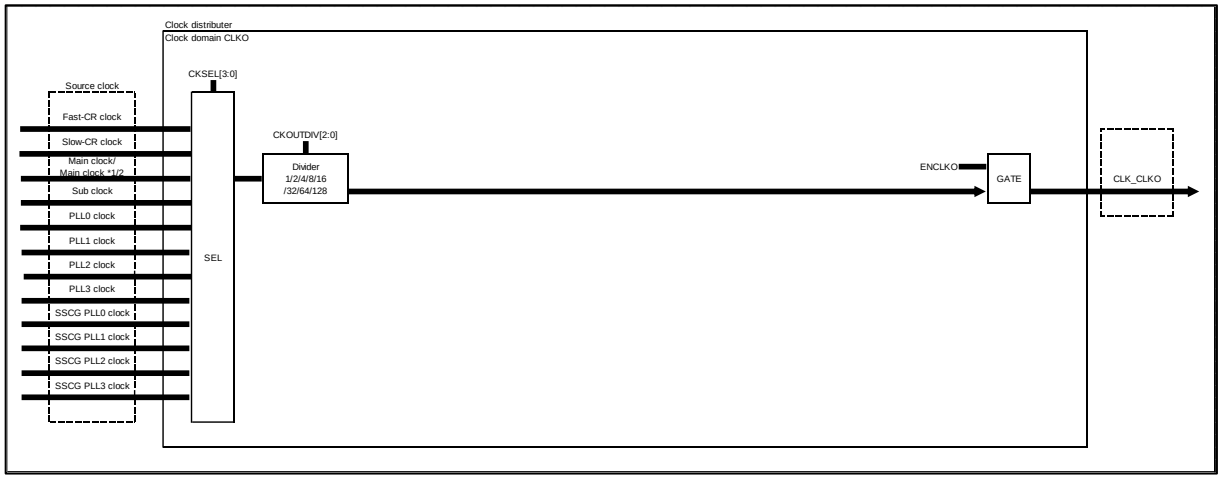


Figure 2-10 Clock Domain CLKO Block Diagram



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
PLL0 clock	Disable	Programmable	Programmable
SSCG PLL0 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.(SYSC_RUNPLLxCNTR/SYSC_PSSPLLxCNTR)
3. Set the RUN/PSS profile register for PLLxEN (SYSC_RUNCKSRER/SYSC_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
6. 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.(SYSC_RUNSSCGxCNTR0 and SYSC_RUNSSCGxCNTR1/SYSC_PSSSSCGxCNTR and SYSC_PSSSSCGxCNTR)
3. Set the RUN/PSS profile register for SSCG PLLxEN (SYSC_RUNCKSRER/SYSC_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
6. 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 '5.1.3 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	128	512MHz	4	128MHz
4MHz	1	4MHz	120	480MHz	6	80MHz

Note:

- *Make the input clock frequency division setting so that the PLL input clock (PLLin) is at 4 MHz. Make the multiplication setting so that the PLL output clock (PLLout) is at 400 MHz to 512 MHz.*
- *The maximum PLL clock frequency must be 128MHz.*
  Output divider configuration can be set by the following.
  - PLLDIVM bit in SYSC0_RUNPLL0CNTR register
  - PLLDIVM bit in SYSC0_PSSPLL0CNTR register

  (e.g. If PLLout is 512MHz, these settings must be configured as "multiply by 4" and over multiplication setting.)

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	128	512MHz	4	128MHz
4MHz	1	4MHz	120	480MHz	6	80MHz

Notes:

- *Make the input clock frequency division setting so that the SSCG PLL input clock (SSCG PLLin) is at 4 MHz. Make the multiplication setting so that the SSCG PLL output clock (SSCG PLLout) is at 400 MHz to 512 MHz.*
- *The maximum SSCG PLL clock frequency must be 128MHz.*
  Output divider configuration can be set by the following.
  - SSCGDIVM bit in SYSC0_RUNSSCG0CNTR0 register
  - SSCGDIVM bit in SYSC0_PSSSSCG0CNTR0 register

  (e.g. If SSCG PLLout is 512MHz, these settings must be configured as "multiply by 4" and over multiplication setting.)

3.3. Clock Distributer

This section explains clock distributer

Clock distributer 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
CD_TRC	TRC_CLK	Debug 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	Initial State	RUN/PSS	S6J312xHzC* Frequency
CD0	CLK_CPU	CPU	1	Programmable	128MHz
	CLK_SHE	SHE	-	Not control	Same as CLK_CPU
	CLK_FCLK	TCFLASH WFLASH	1	Programmable	64MHz
	CLK_ATB	ATB	1	Programmable	64MHz
	CLK_DBG	DBG	1	Programmable	64MHz
	CLK_HPM	High Performance Matrix, Common Peripheral2	1	Programmable	32MHZ
	CLK_HPM2		2	2	16MHz
	CLK_DMA	DMA	1	Programmable	32MHz
	CLK_MEMC	Memory & Config Group	1	Programmable	32MHz
	CLK_EXTBUS	Ext Bus Group	1	Programmable	25MHz
	CLK_SYSC1	MCU config Group1	1	Programmable	32MHz

* x: A/9, z: A/B

Note:

- *The appendix reference for CLK_EXTBUS.*

Clock Domain	Internal Clock	Resources	Initial State	RUN/PSS	S6J312xHzC* Frequency
CD0	CLK_HAPP0A0	Application Specific0 (HPM)	1	Programmable	32MHz
	CLK_HAPP0A1		1	Programmable	32MHz
	CLK_HAPP1B0	Application Specific1 (HPM)	1	Programmable	32MHz
	CLK_HAPP1B1		1	Programmable	32MHz
	CLK_LLPBM	Common Peripheral (LLPBM)	1	Programmable	128MHz
	CLK_LLPBM2	Common Peripheral0/1/2	2	2	64MHz
	CLK_LCP	Common Peripheral0/1	1	Programmable	64MHz
	CLK_LCP0	Common Peripheral0	1	Programmable	32MHz
	CLK_LCP0A		1	Programmable	32MHz
	CLK_LCP1	Common Peripheral1	1	Programmable	32MHz
	CLK_LCP1A		1	Programmable	32MHz
	CLK_LAPP0	Application Specific0 (LLPPBM)	1	Programmable	32MHz
	CLK_LAPP0A		1	Programmable	32MHz
	CLK_LAPP1	Application Specific1 (LLPPBM)	1	Programmable	32MHz
	CLK_LAPP1A		1	Programmable	32MHz

* x: A/9, z: A/B

Clock Domain	Internal Clock	Resources	Initial State	RUN/PSS	S6J312xHzC* Frequency
CDx (x=1/2/3/4/5)	CLK_CDx	CDx_RESOURCES	1	Programmable	-
	CLK_CDxA0		1	Programmable	-
	CLK_CdxA1		1	Programmable	-
	CLK_CDxB0		1	Programmable	-
	CLK_CDxB1		1	Programmable	-
CD_TRC	CLK_TRC	Debug Group	1	Programmable	64MHz
CD_HSSPI	CLK_HSSPI	HSSPI	1	Programmable	32MHz
CD_MCUC	CLK_SYSC0H	MCU Config Group (AHB)	1	Programmable	32MHz
	CLK_COMH	Communication Peripheral (AHB)	1	Programmable	32MHz
	CLK_RAM0H	Back-up RAM0	1	Programmable	32MHz
	CLK_RAM1H	Back-up RAM1	1	Programmable	32MHz
	CLK_SYSC0P	MCU Config Group0 (APB)	1	Programmable	32MHz
	CLK_COMP	Communication Peripheral (APB)	1	Programmable	32MHz
CD_CLKO	CLK_CLKO	Clock output function	1	Programmable	128MHz

* x: A/9, z: A/B

Notes:

- *Make the frequency division setting so that no clocks are supplied that exceed the maximum operating frequency of any internal operating clock.*
- *The appendix reference for CDx and CD_HSSPI.*

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.

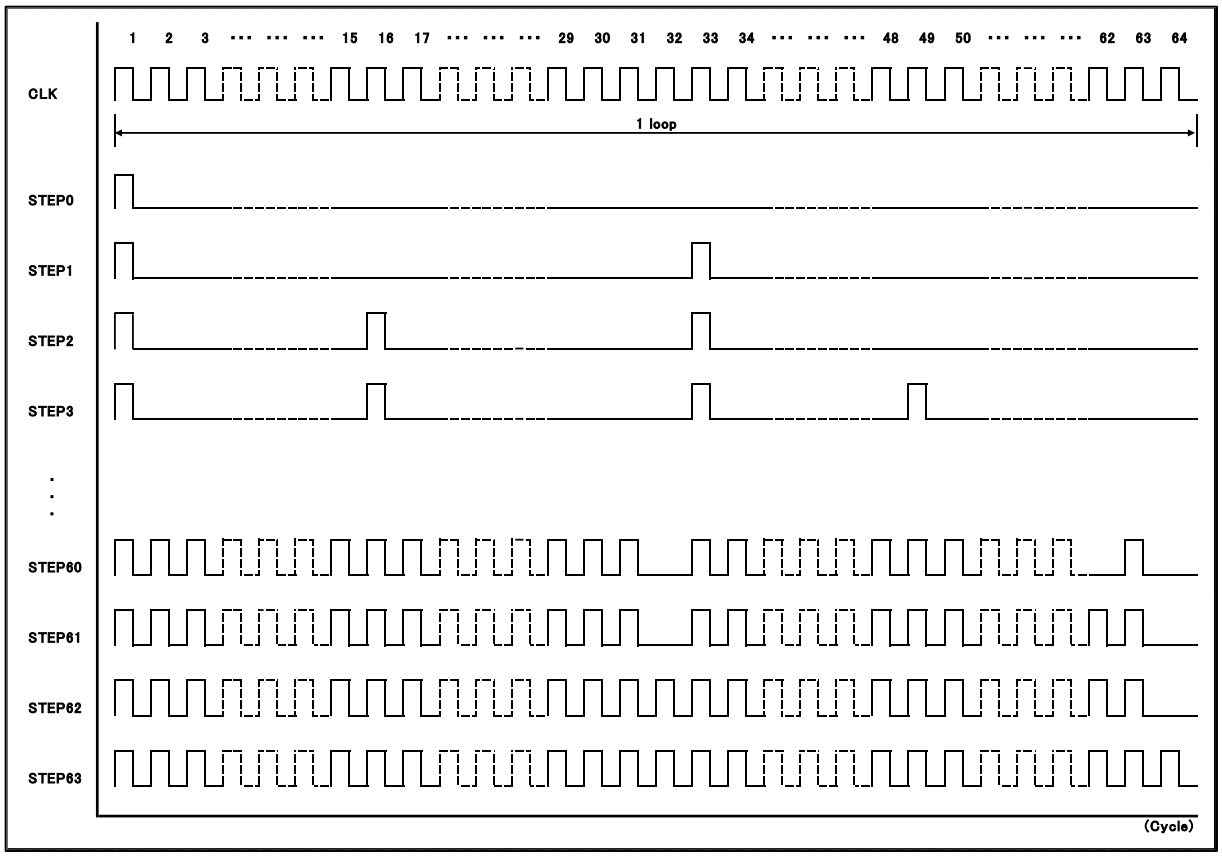
Clock Gear Control

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
- Step loop configuration
- Step width configuration

Figure 3-1 Gear Clock Output



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'.

4. Setting Procedure Example

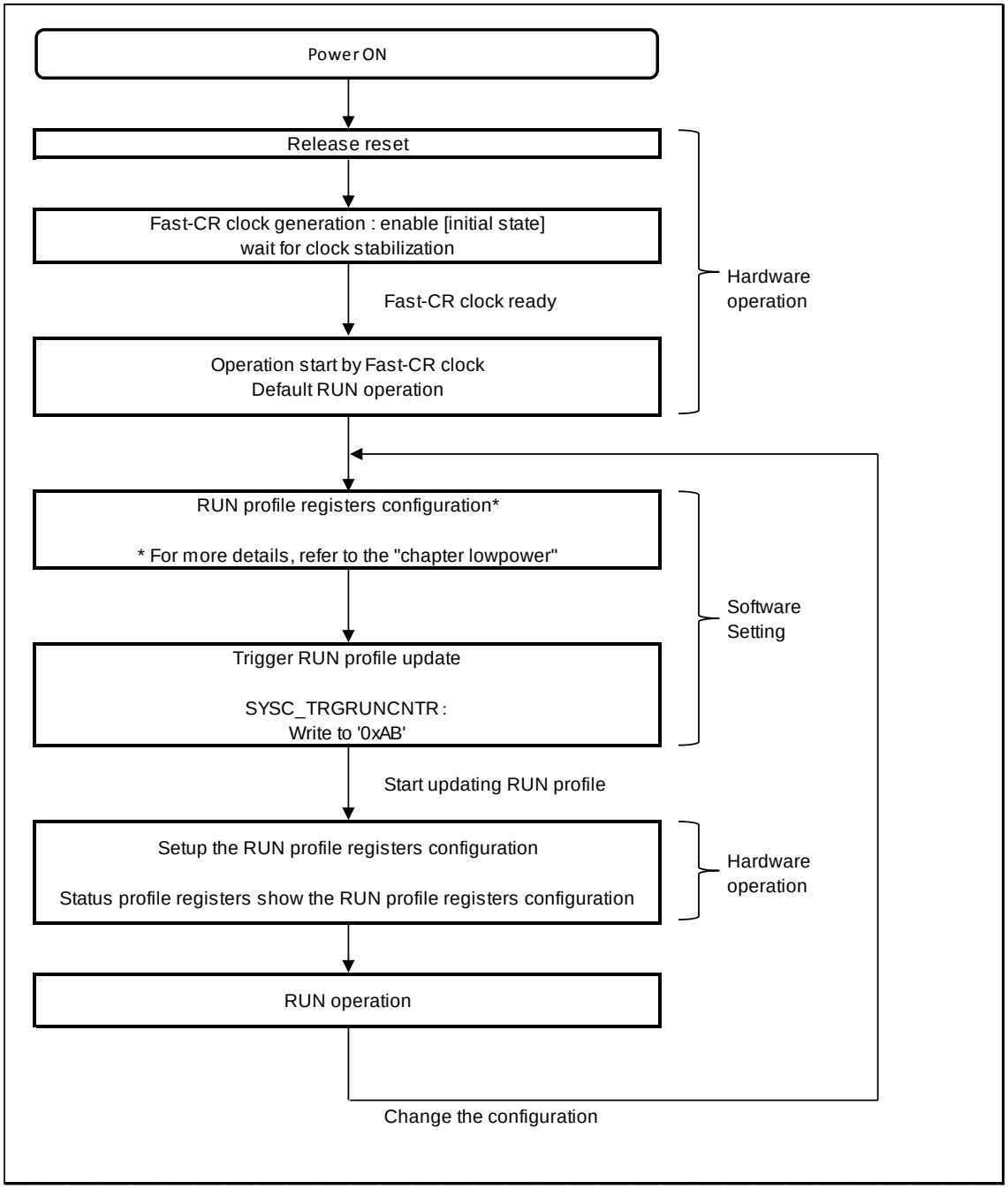
This section explains a setting procedure example of the clock system.

4.1. RUN Configuration

This section explains RUN configuration of clock system.

Setting Procedure Example for RUN Configuration

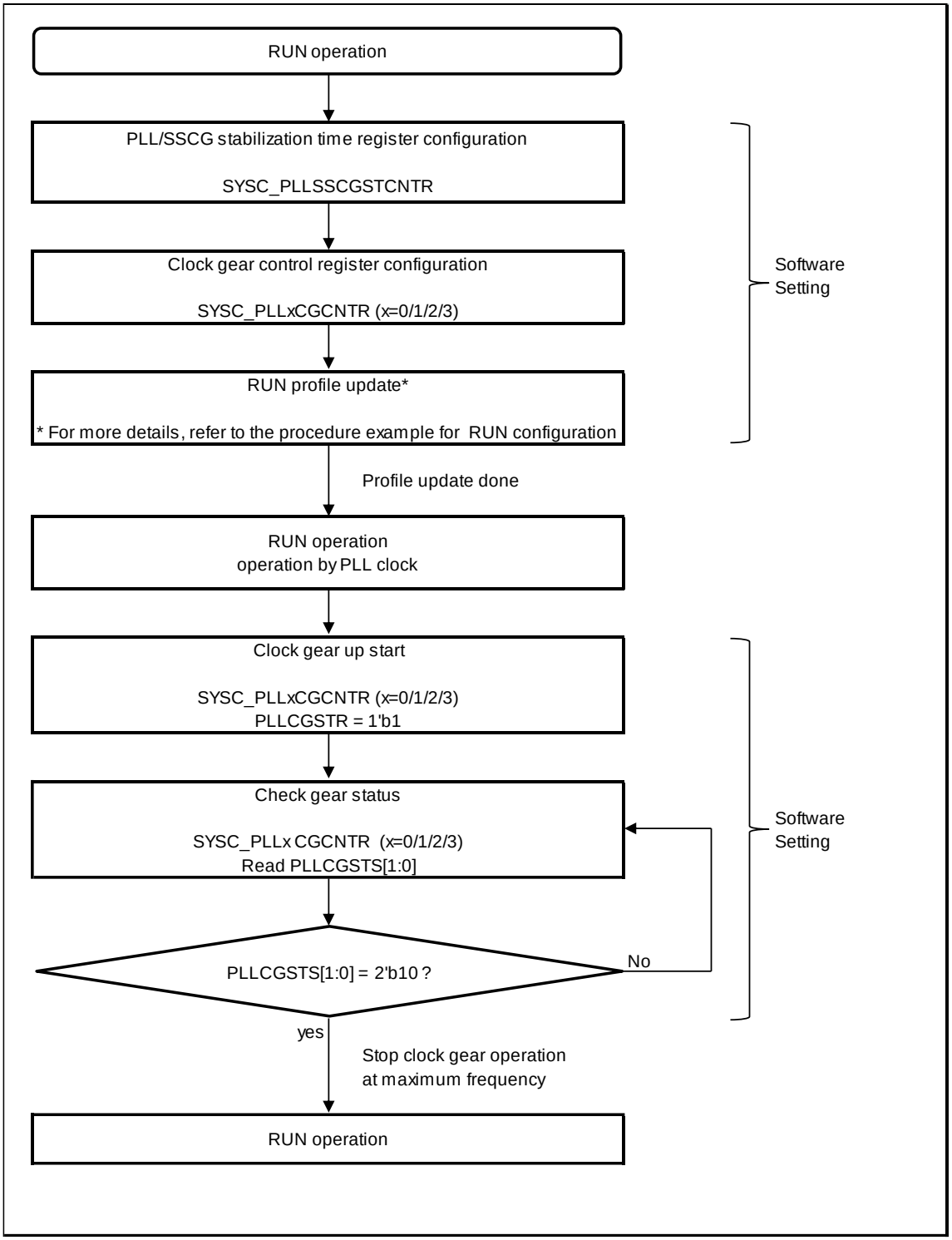
Figure 4-1 Setting Procedure Example for RUN Configuration



Note:

- *To use the main clock and the CLK_PLL_SSCG 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:

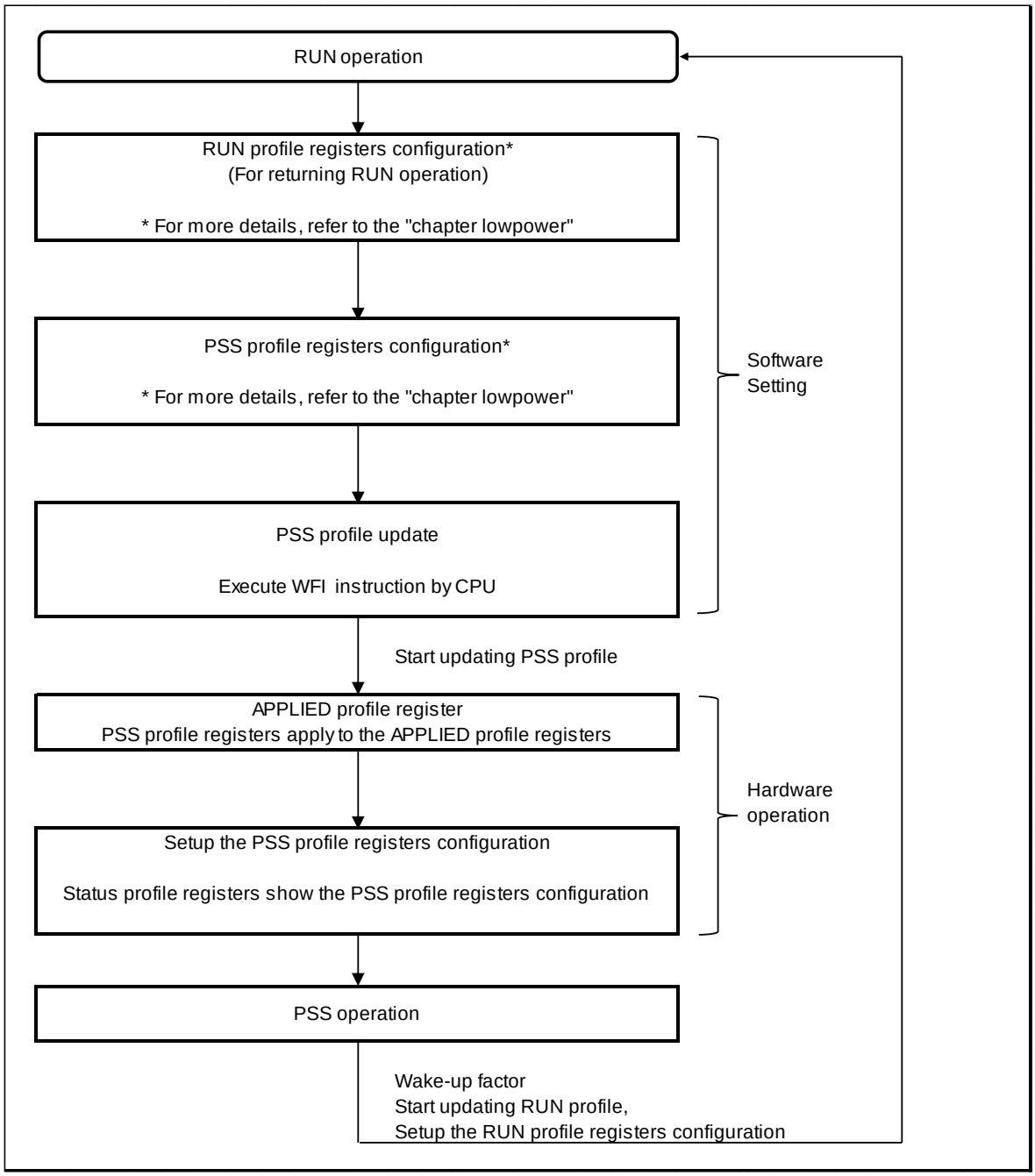
To use the pll clock or sscg pll clock, clock gear must be used. Make sure to set the start gear step does not exceed the maximum frequency of every internal clock in the clock domain.

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 CLK_PLL_SSCG 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:lowpower".

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_PLLSSCGSTCNTR	PLL/SSCG stabilization time control register	5.1.3
SYSC_PLL0CGCNTR	PLL0 clock gear control register	5.1.4
SYSC_SSCG0CGCNTR	SSCG PLL0 clock gear control register	5.1.5

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

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			01111				

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			10010				

[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	0	000			00		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							R/W
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

[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

[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

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

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

PLL/SSCG stabilization time control register(SYSC_STCNTR) 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 0000000							
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 : Main clock period[s] * 2^9 [cycle]
1001	Stabilization time : Main clock period[s] * 2^10 [cycle]
1010	Stabilization time : Main clock period[s] * 2^11 [cycle]
1011	Stabilization time : Main clock period[s] * 2^12 [cycle]
1100	Stabilization time : Main clock period[s] * 2^13 [cycle]
1101	Stabilization time : Main clock period[s] * 2^14 [cycle]
1110	Stabilization time : Main clock period[s] * 2^15 [cycle]
1111	Stabilization time : Main clock period[s] * 2^16 [cycle]

Reading SSCGSTABS[3] bit always returns '1'

Note:

- *These bits have to be set before PLL clock enable setting.*
- *Changing these bits are prohibited after 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 : Main clock period[s] * 2^9 [cycle]
1001	Stabilization time : Main clock period[s] * 2^10 [cycle]
1010	Stabilization time : Main clock period[s] * 2^11 [cycle]
1011	Stabilization time : Main clock period[s] * 2^12 [cycle]
1100	Stabilization time : Main clock period[s] * 2^13 [cycle]
1101	Stabilization time : Main clock period[s] * 2^14 [cycle]
1110	Stabilization time : Main clock period[s] * 2^15 [cycle]
1111	Stabilization time : Main clock period[s] * 2^16 [cycle]

Reading PLLSTABS[3] bit always returns '1'

Note:

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

5.1.4. 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	0000000							

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

[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

Note:

- *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

Note:

- *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

Note:

- *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	Stop clock gear operation at minimum frequency
01	Gear up operation
10	Stop clock gear operation at maximum frequency
11	Gear down operation

Note:

- *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 starting 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

Note:

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

5.1.5. 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	0000000							

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

[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

Note:

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

Note:

- *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

Note:

- *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	Stop clock gear operation at minimum frequency
01	Gear up operation
10	Stop clock gear operation at maximum frequency
11	Gear down operation

Note:

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

[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 starting 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

Note:

- *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	00000				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 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 Lower/Upper 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

  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-T00P00R01L03-E1-GBB01

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.

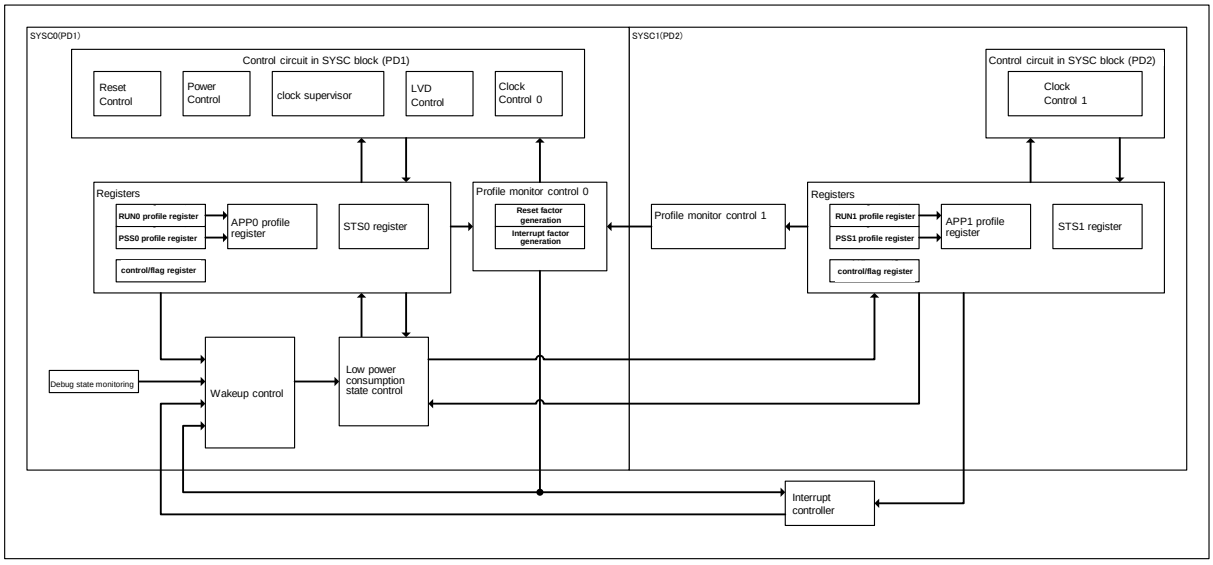
2. Configuration

This section explains the configuration for low power consumption.

2.1. Block Diagram

This section shows the configuration for low power consumption control.

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.

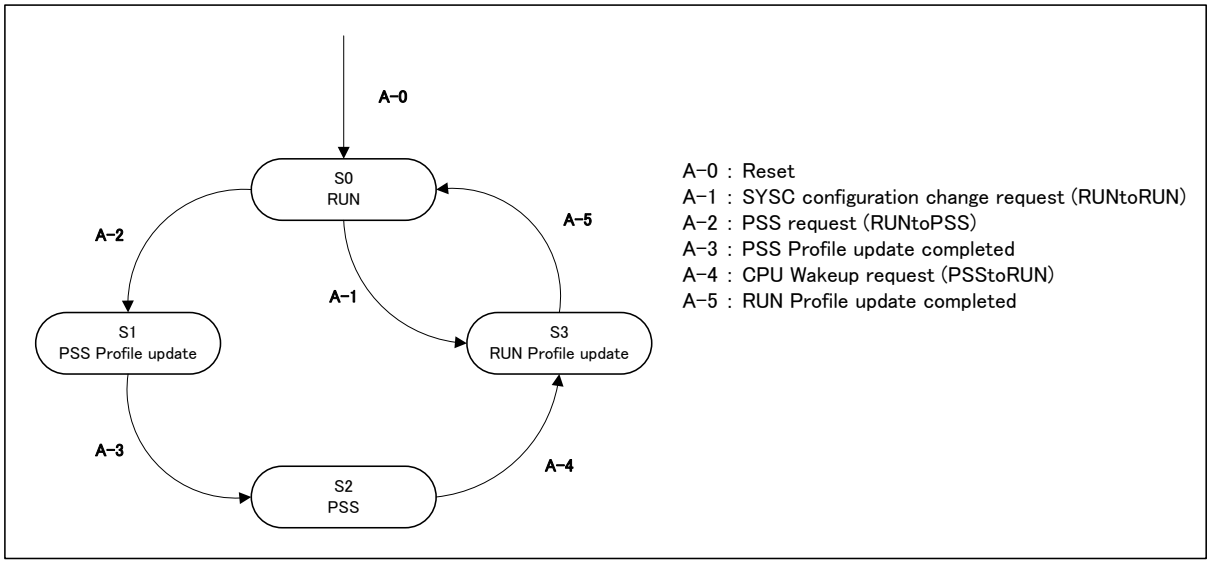
PSS

This is the state in which the CPU has stopped programs and can set low power consumption.

Access becomes impossible since the Flash macro causes the Deep Sleep 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)

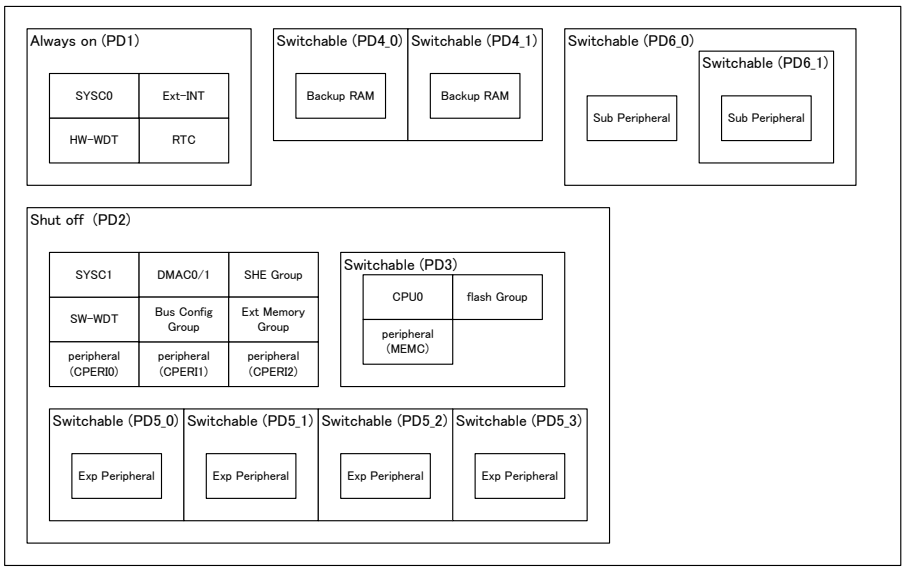
Power is always supplied to the Always ON power supply domain (PD1). Other domains can select their supply of power. Some power supply domains have a parent-child relationship. PD2 and PD3 can have only settings that are the same. To turn OFF the power supply of the Core power supply domain or Peripheral 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, PD2, PD5_0, PD5_1, PD5_2, PD5_3

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

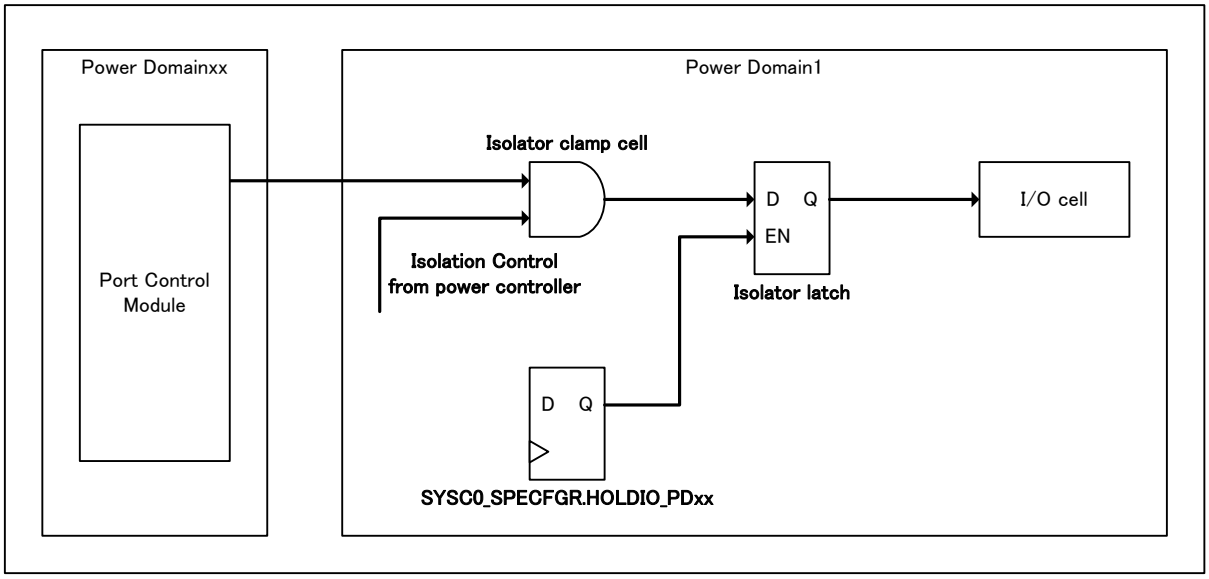
To shut down the power supply, turn OFF all PSWs simultaneously.

I/O Control when Power Supply is OFF

Turning OFF the PD2 power supply also turns OFF the power supply of the circuit controlling I/O. 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:

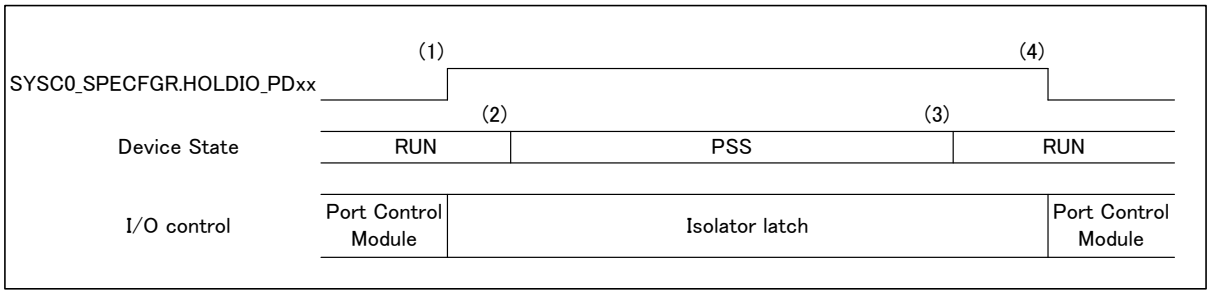
- *PDxx = PD2/PD5_0/PD5_1/PD5_2/PD5_3/PD6_0/PD6_1*

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
	LCPP0 clock oscillation enabled	Enable	Enable	Enable control	SYSC1
	LCPP1 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
	LCPP0A clock oscillation enable	Enable	Enable	Enable control	SYSC1
	LCPP1A 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	0b 0	Enable Control	Enable Control	SYSC0
	Detection voltage	0b 10	Enable Control	Enable Control	SYSC0
	Operation enable	0b 1	Enable Control	Enable Control	SYSC0
LVDL2	Operation selection	0b 0	Fixed	Fixed	SYSC0
	Detection voltage	0b 00	Fixed	Fixed	SYSC0
	Operation enable	0b 0	Fixed	Fixed	SYSC0
LVDH1	Operation selection	0b 0	Enable Control	Enable Control	SYSC0

Category	Setting Item	RUN Setting (Initial Value)	RUN Setting	PSS Setting	Control Block
	Detection voltage	0b 0110	Enable Control	Enable Control	SYSC0
	Operation enable	0b 1	Enable Control	Enable Control	SYSC0
LVDH2	Operation selection	0b 0	Fixed	Fixed	SYSC0
LVDH2	Detection voltage	0b 0000	Fixed	Fixed	SYSC0
	Operation enable	0b 0	Fixed	Fixed	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

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.

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 PD3 (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.*
- *X : Don't care*
- *0/1 : active value*
- *C : Object be to compare*

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

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

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		
CDMCUCDCSL[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

PSS

There is no error condition.

Notes:

- *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.*

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_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	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.

CSV Operation was Enabled while Monitoring/Reference Clock Oscillation was Disabled.

RUN

SYSC0_RUNCKSRER						SYSC0_RUNCSVCFGR			
SSCG0 EN	PLL0 EN	SOSC EN	MOSC EN	SCROSC EN	CROSC EN	SSCG0 CSVE	PLL0 CSVE	SO CSVE	MO CSVE
1	X	X	0	X	X	1	X	X	X
0	X	X	1	X	X	1	X	X	X
X	1	X	0	X	X	X	1	X	X
X	0	X	1	X	X	X	1	X	X
X	X	0	X	X	X	X	X	1	X
X	X	X	0	X	X	X	X	X	1

PSS

SYSC0_PSSCKSRER						SYSC0_PSSCSVCFGR			
SSCG0 EN	PLL0 EN	SOSC EN	MOSC EN	SCROSC EN	CROSC EN	SSCG0 CSVE	PLL0 CSVE	SO CSVE	MO CSVE
1	X	X	0	X	X	1	X	X	X
0	X	X	1	X	X	1	X	X	X
X	1	X	0	X	X	X	1	X	X
X	0	X	1	X	X	X	1	X	X
X	X	0	X	X	X	X	X	1	X
X	X	X	X	0	X	X	X	1	X
X	X	X	0	X	X	X	X	X	1
X	X	X	X	0	X	X	X	X	1

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

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

The Clock Used by the Software Watchdog has stopped.

RUN

SYSC0_RUNCKSRER				SWDG0_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				SWDG0_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

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

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

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

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 update enable 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.

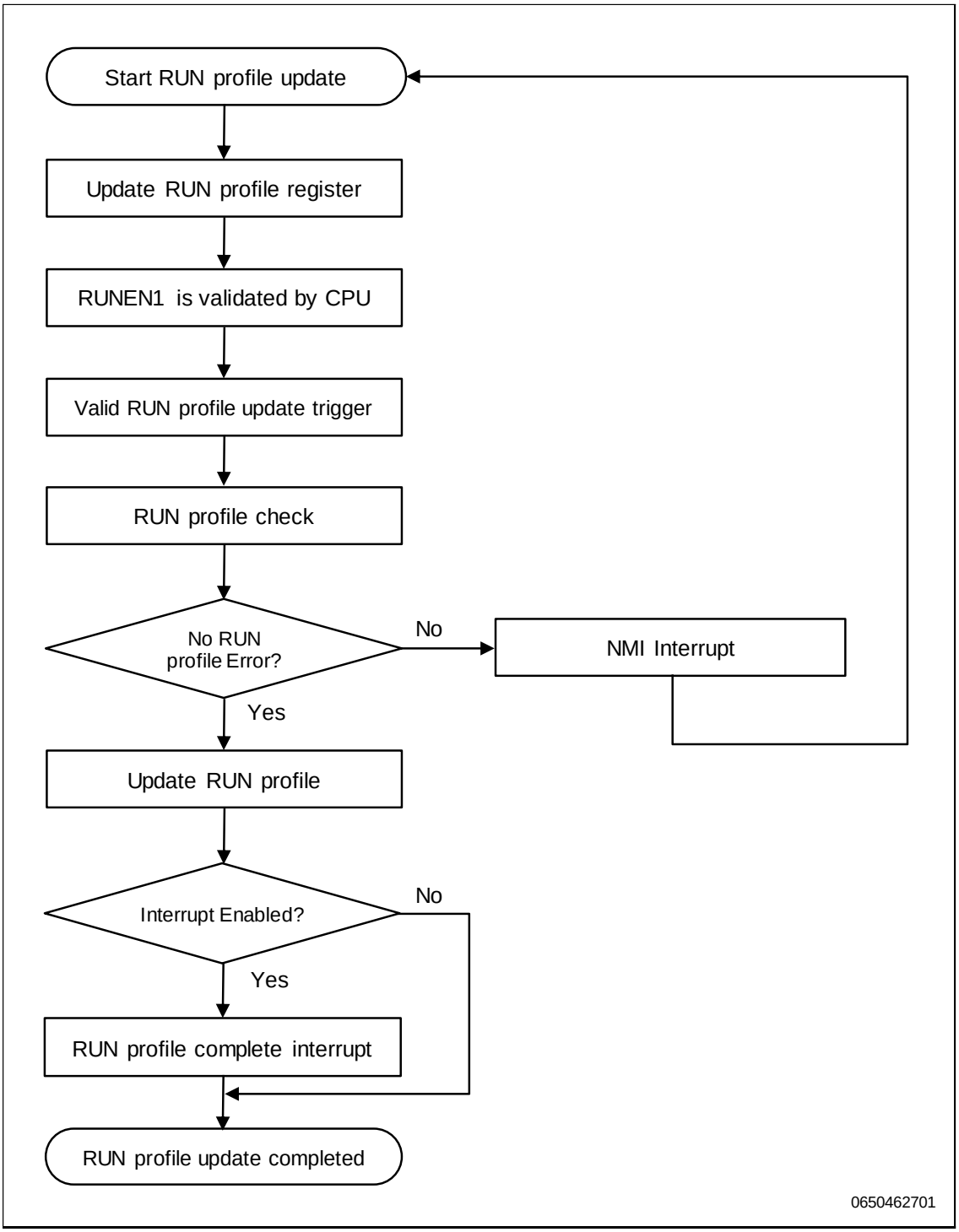
The following example explains an operation procedure for RUN profile update.

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 APPLIED 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 of each CPU to PSS.

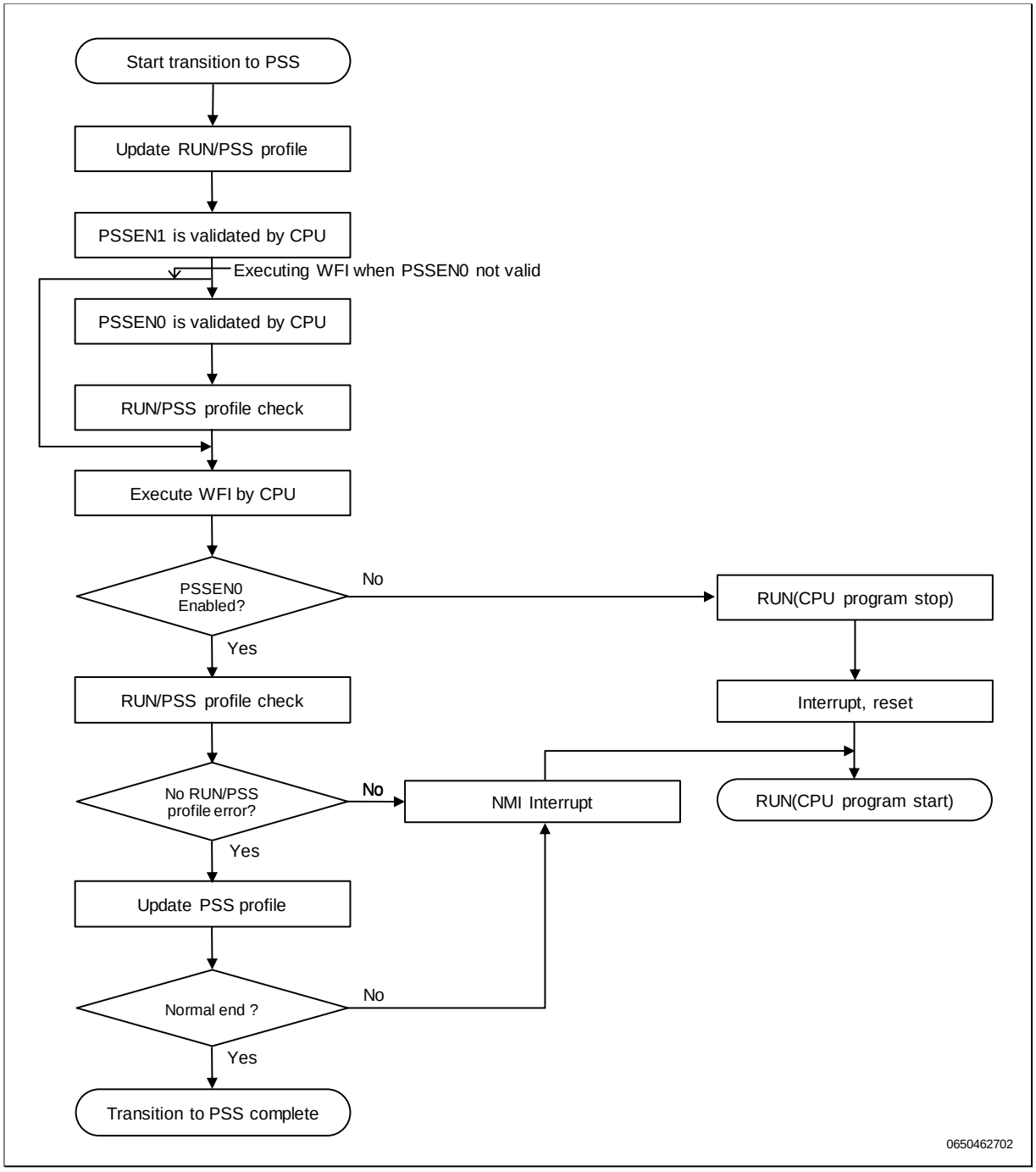
1. The user prepares a new PSS profile and RUN profile (register settings). For the RUN profile, make settings for returning from PSS.
2. Write 0xBA to the PSS profile update enable register (SYSC1_PSSENR:PSSEN1).
3. Write 0xBA to the PSS profile update enable register (SYSC0_PSSENR:PSSEN0).
4. Have 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 APPLIED 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 of each CPU (SYSC0_SYSSTSR:PSSSTS0) to "0" and the PSS profile completion flag of each CPU (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 WFI 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 disconnect the power of PD2, enable the HOLD data latch setting bit. If you do not set it, I/O state can not be retained.*
- *When transferring from a RUN mode to PSS mode, please be sure to make Main_OSC enable.*

The following figure is a flowchart of RUN to PSS transition operations.

Figure 4-2 RUN to PSS Operation Flowchart

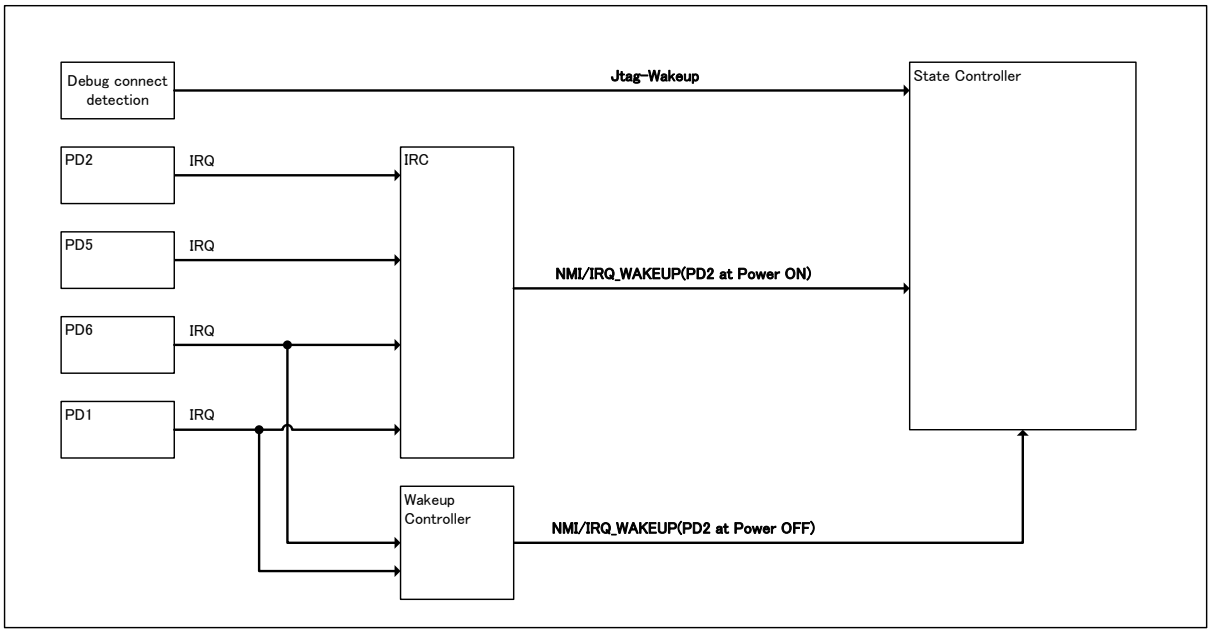


4.3. Transition from PSS to RUN

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, CSV *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 all CPUs from PSS to RUN.

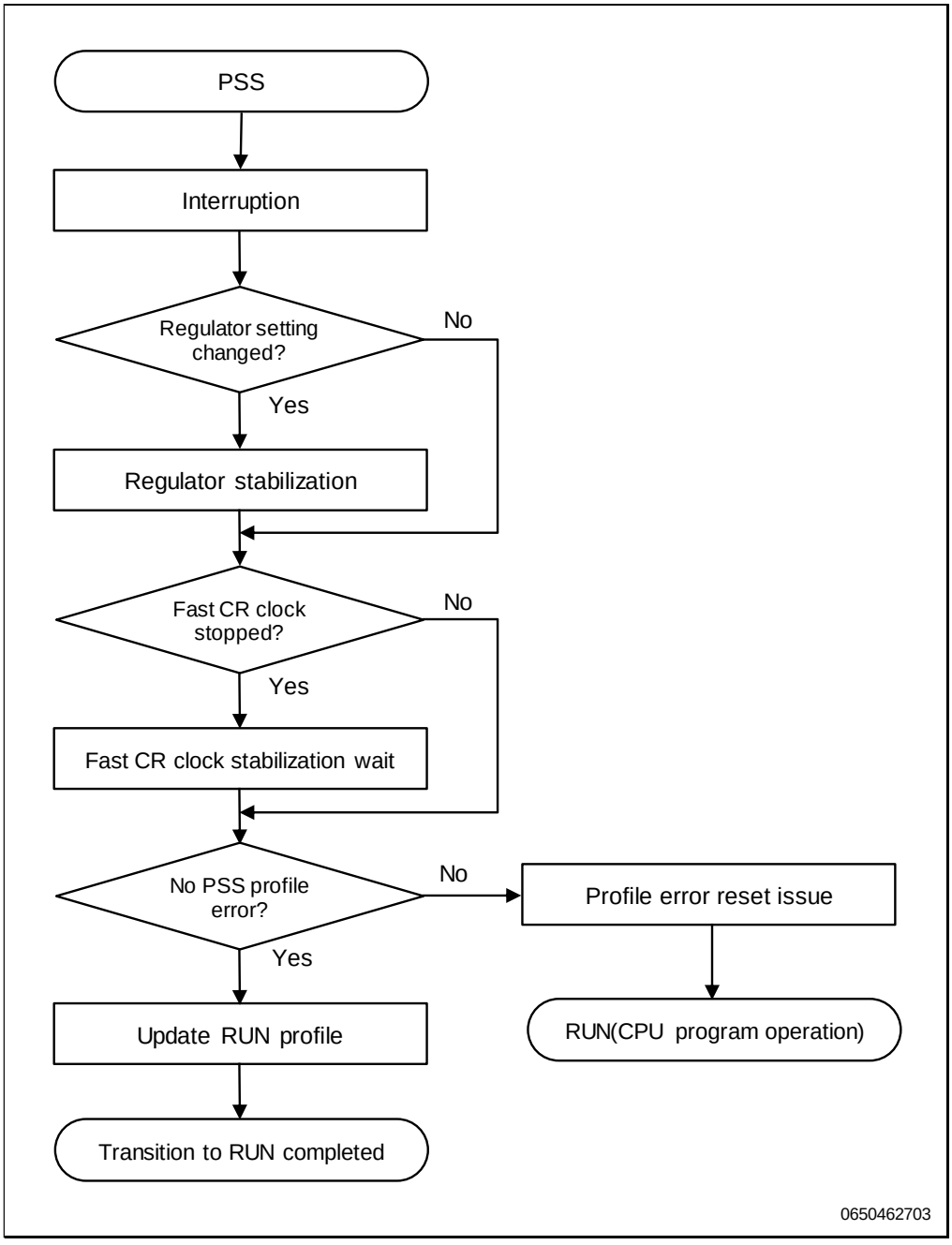
1. Generate a return factor. After the control circuit recognizes the return factor, the CPUs start to return.
2. If the high-speed CR clock has stopped, the CPUs wait for oscillation of the high-speed CR clock to stabilize.
3. If the contents of the RUN profile settings have a problem (bit error, etc.), 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:RUNSTS0) to "1".
    2. The circuit copies the RUN profile contents to the APPLIED 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:RUNSTS0) of each CPU to "0".
5. CPUs wait for Flash macro to stabilize from DeepStandby release. Stabilize wait time will occur for regulator if voltage/mode change occurs.
6. There is a transition to RUN, and the CPUs start 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 that PLL/SSCG oscillation allowed. After the power domain 6 is turn ON in RUN mode, it please allowed 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 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.

- *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_SYSSTSR	System status register	5.6.1
SYSC0_SYSINTER	System status interrupt enable register	5.6.2
SYSC0_SYSICLR	System status flag interrupt clear register	5.6.3
SYSC0_SYSERRIR0	System error interrupt factor register 0	5.6.4
SYSC0_SYSERRIR1	System error interrupt factor register 1	5.6.5
SYSC0_SYSERRICLR0	System error interrupt factor clear register 0	5.6.6
SYSC0_SYSERRICLR1	System error interrupt factor clear register 1	5.6.7
SYSC0_SYSPROSTSR	Profile status register	5.6.8
SYSC0_SYSRUNPEFR	RUN profile error flag register	5.6.9
SYSC0_SYSPSSPEFR	PSS profile error flag register	5.6.10
SYSC0_SYSIDR	System ID register	5.6.11
SYSC0_SYSPFIDR	Platform ID 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	0
SYSC1_RUNCKDIVR2	RUN clock divider register 2	5.11.9
SYSC1_RUNCKDIVR3	RUN clock divider register 3	0
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

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 releasing 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 Release Setting Bits

These bits compose the register for releasing protection keys.

Write

bit31:0	Description
0x5CAC_CE55	Lock release
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
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

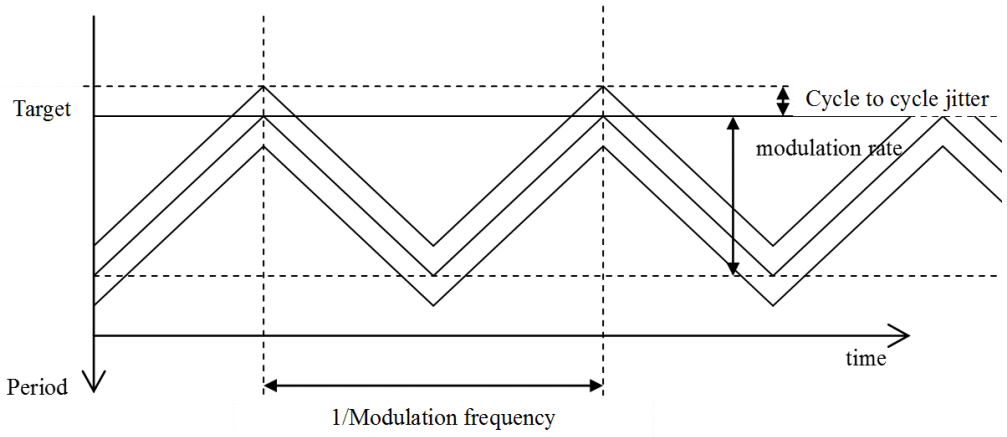
* Fin: SSCG reference clock frequency, Fmod: Modulation frequency

[bit16] SSCGxMODE: SSCG PLLx modulation Mode Setting Bit

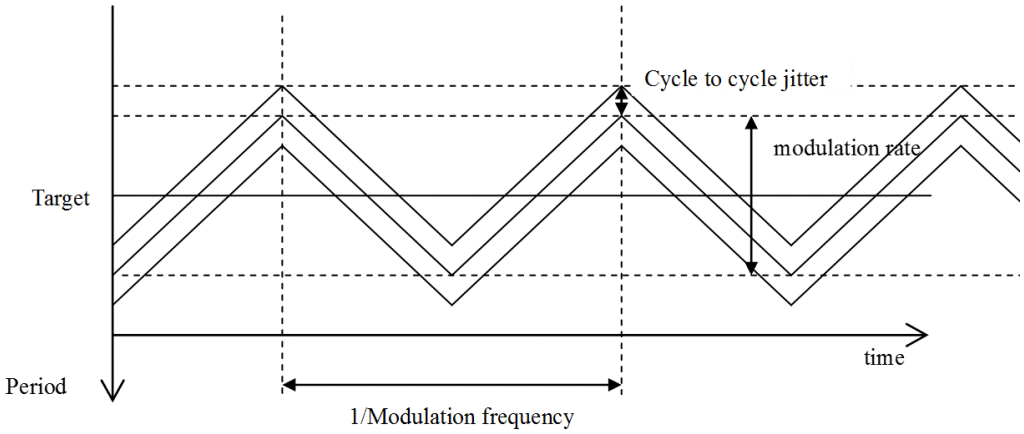
This bit sets the modulation mode of the SSCG PLLx clock.

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	R/W	R0,WX			R/W		R/W
Protection Attribute	WPS							
Initial Value	0	0	000			10		1

bit	23	22	21	20	19	18	17	16
Field	Reserved	LVDL2S	Reserved			LVDL2V		LVDL2E
R/W Attribute	R0,WX	R0,WX	R0,WX			R0,WX		R0,WX
Protection Attribute	WPS							
Initial Value	0	0	000			00		0

bit	15	14	13	12	11	10	9	8
Field	Reserved	LVDH1S	Reserved	LVDH1V				LVDH1E
R/W Attribute	R0,WX	R/W	R0,WX	R/W				R/W
Protection Attribute	WPS							
Initial Value	0	0	0	0110				1

bit	7	6	5	4	3	2	1	0
Field	Reserved	LVDH2S	Reserved	LVDH2V				LVDH2E
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX				R0,WX
Protection Attribute	WPS							
Initial Value	0	0	0	0000				0

[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.

Bit	Description	Guaranteed MCU operation range
00	Reserved	No
01	Reserved	
10*	0.97±0.05V	
11*	1.07±0.05V	

Note:

- *After the detection voltage is changed, low-voltage detection is unavailable until the low-voltage detection stability time elapses.*

*: This LVD cannot be used to reliably generate a reset before voltage dips below minimum guaranteed MCU operation voltage,as these detection levels are below the minimum guaranteed MCU operation voltage.

[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	Reserved
01	Reserved
10	0.97±0.05V
11	1.07±0.05V

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	Reserved
0001	Reserved
0010	Reserved
0011	Reserved
0100	3.8±0.2V
0101	4.0±0.2V
0110	4.2±0.2V
0111	Reserved
1xxx	Reserved

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	Reserved
0001	Reserved
0010	Reserved
0011	Reserved
0100	3.8±0.2V
0101	4.0±0.2V
0110	4.2±0.2V
0111	Reserved
1xxx	Reserved

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						SO CSVE	MO CSVE
R/W Attribute	R0,WX						R/W	R/W
Protection Attribute	WPS							
Initial Value	000000						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.

[bit7:2] Reserved: Reserved Bits

[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/W0
Protection Attribute	WPS							
Initial Value	0	000000						0

[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	Setting Prohibited

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
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															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
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
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 PLL 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

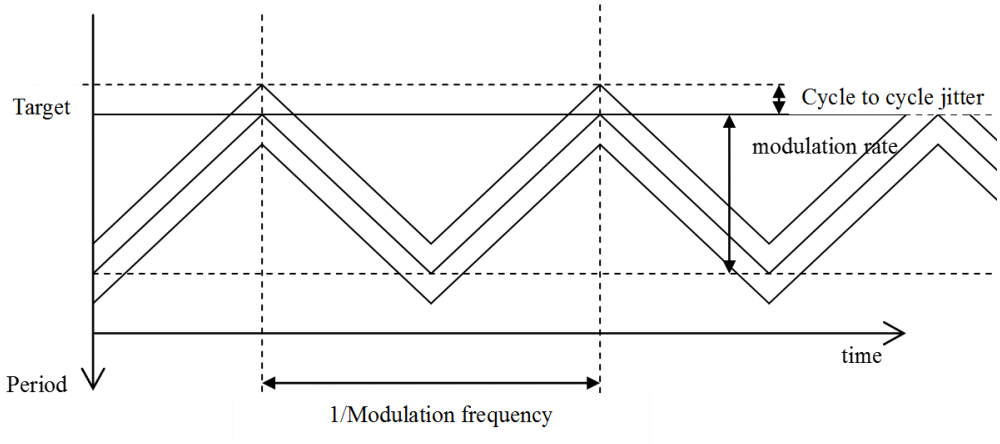
* Fin: SSCG reference clock frequency, Fmod: Modulation frequency

[bit16] SSCGxMODE: SSCG PLL modulation Mode Setting Bit

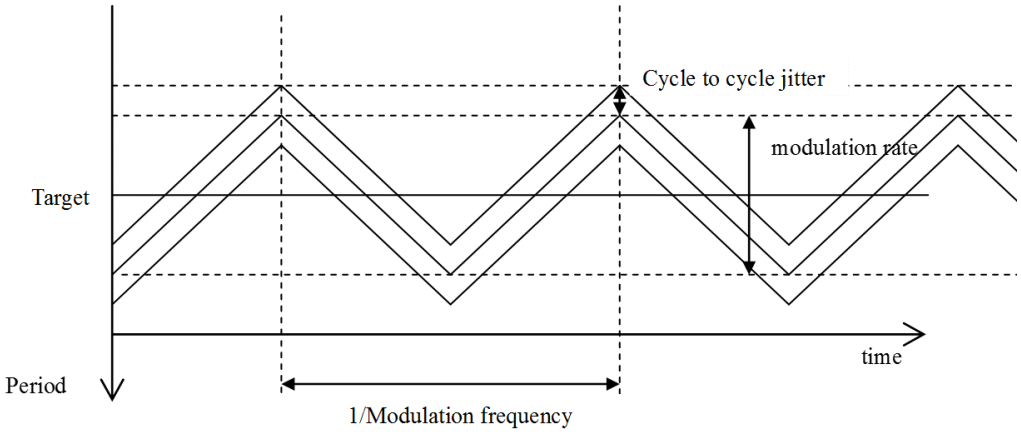
This bit sets the modulation mode of the SSCG PLL clock.

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 PLL 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	R/W	R0,WX			R/W		R/W
Protection Attribute	WPS							
Initial Value	0	0	000			10		1

bit	23	22	21	20	19	18	17	16
Field	Reserved	LVDL2S	Reserved			LVDL2V		LVDL2E
R/W Attribute	R0,WX	R0,WX	R0,WX			R0,WX		R0,WX
Protection Attribute	WPS							
Initial Value	0	0	000			00		0

bit	15	14	13	12	11	10	9	8
Field	Reserved	LVDH1S	Reserved	LVDH1V				LVDH1E
R/W Attribute	R0,WX	R/W	R0,WX	R/W				R/W
Protection Attribute	WPS							
Initial Value	0	0	0	0110				1

bit	7	6	5	4	3	2	1	0
Field	Reserved	LVDH2S	Reserved	LVDH2V				LVDH2E
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX				R0,WX
Protection Attribute	WPS							
Initial Value	0	0	0	0000				0

[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	Guaranteed MCU operation range
00	Reserved	No
01	Reserved	
10*	0.97±0.05V	
11*	1.07±0.05V	

Note:

- *After the detection voltage is changed, low-voltage detection is unavailable until the low-voltage detection stability time elapses.*

*: This LVD cannot be used to reliably generate a reset before voltage dips below minimum guaranteed MCU operation voltage, as these detection levels are below the minimum guaranteed MCU operation voltage.

[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	Reserved
01	Reserved
10	0.97±0.05V
11	1.07±0.05V

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	Reserved
0001	Reserved
0010	Reserved
0011	Reserved
0100	3.8±0.2V
0101	4.0±0.2V
0110	4.2±0.2V
0111	Reserved
1xxx	Reserved

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	Reserved
0001	Reserved
0010	Reserved
0011	Reserved
0100	3.8±0.2V
0101	4.0±0.2V
0110	4.2±0.2V
0111	Reserved
1xxx	Reserved

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						SO CSVE	MO CSVE
R/W Attribute	R0,WX						R/W	R/W
Protection Attribute	WPS							
Initial Value	000000						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.

[bit7:2] Reserved: Reserved Bits

[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						0

[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

[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	-

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 sets 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_APPCKSRER0 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				SSCG3 EN	SSCG2 EN	SSCG1 EN	SSCG0 EN
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	MOSC EN	SCROSC EN	CROSC EN
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_APPCKSELR0 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
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						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: 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							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 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,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 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,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

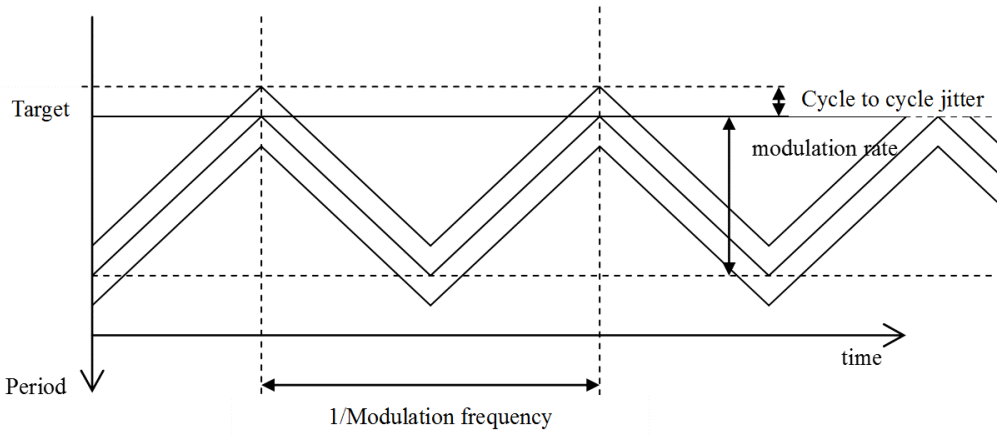
* Fin: SSCG reference clock frequency, Fmod: Modulation frequency

[bit16] SSCGxMODE: SSCG PLLx modulation Mode Setting Bit

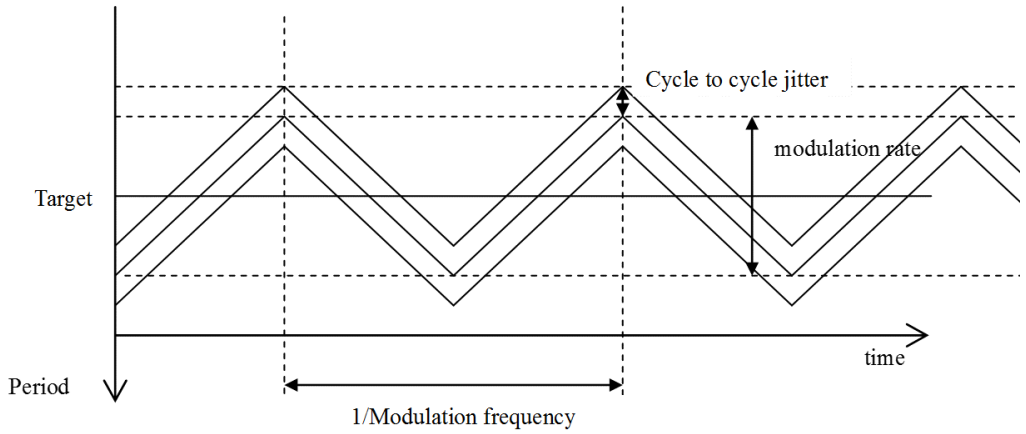
This bit sets the modulation mode of the SSCG PLLx clock.

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 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	R,WX	R0,WX			R,WX		R,WX
Protection Attribute	WPS							
Initial Value	0	0	000			10		1

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	0	000			00		0

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	0	0	0110				1

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	0	0	0000				0

[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	Guaranteed MCU operation range
00	Reserved	No
01	Reserved	
10*	0.97±0.05V	
11*	1.07±0.05V	

Note:

- *After the detection voltage is changed, low-voltage detection is unavailable until the low-voltage detection stability time elapses.*

*: This LVD cannot be used to reliably generate a reset before voltage dips below minimum guaranteed MCU operation voltage, as these detection levels are below the minimum guaranteed MCU operation voltage.

[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	Reserved
01	Reserved
10	0.97±0.05V
11	1.07±0.05V

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	Reserved
0001	Reserved
0010	Reserved
0011	Reserved
0100	3.8±0.2V
0101	4.0±0.2V
0110	4.2±0.2V
0111	Reserved
1xxx	Reserved

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	Reserved
0001	Reserved
0010	Reserved
0011	Reserved
0100	3.8±0.2V
0101	4.0±0.2V
0110	4.2±0.2V
0111	Reserved
1xxx	Reserved

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 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,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						SO CSVE	MO CSVE
R/W Attribute	R0,WX						R,WX	R,WX
Protection Attribute	WPS							
Initial Value	000000						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.

[bit7:2] Reserved: Reserved Bits

[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 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,WX	R0,WX						R,WX
Protection Attribute	WPS							
Initial Value	0	000000						0

[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	-

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
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
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															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

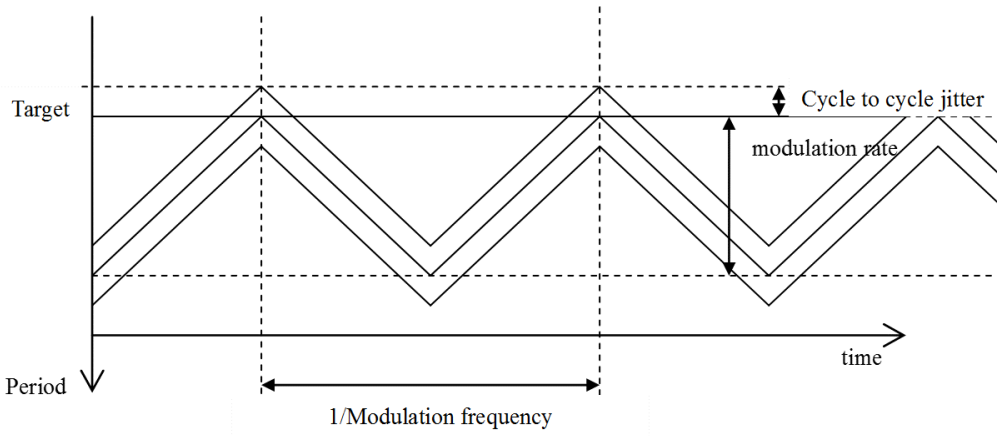
* Fin: SSCG reference clock frequency, Fmod: Modulation frequency

[bit16] SSCGxMODE: SSCG PLLx modulation Mode Setting Bit

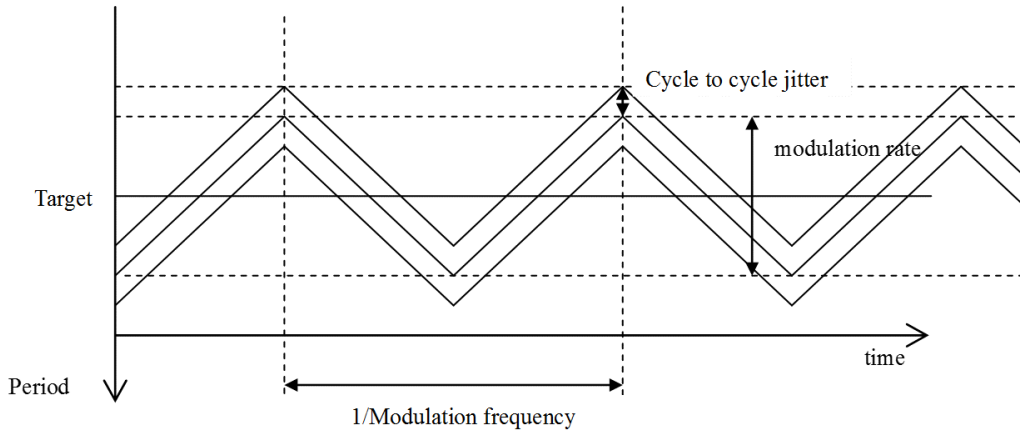
This bit indicates the set value for the modulation mode of the SSCG PLLx clock.

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	1	0	000			10		1

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	0	0	000			00		0

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	1	0	0	0110				1

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	0	0	0	0000				0

[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	Guaranteed MCU operation range
00	Reserved	No
01	Reserved	
10*	0.97±0.05V	
11*	1.07±0.05V	

Note:

- *After the detection voltage is changed, low-voltage detection is unavailable until the low-voltage detection stability time elapses.*

*: This LVD cannot be used to reliably generate a reset before voltage dips below minimum guaranteed MCU operation voltage, as these detection levels are below the minimum guaranteed MCU operation voltage.

[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	Reserved
01	Reserved
10	0.97±0.05V
11	1.07±0.05V

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	Reserved
0001	Reserved
0010	Reserved
0011	Reserved
0100	3.8±0.2V
0101	4.0±0.2V
0110	4.2±0.2V
0111	Reserved
1xxx	Reserved

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	Reserved
0001	Reserved
0010	Reserved
0011	Reserved
0100	3.8±0.2V
0101	4.0±0.2V
0110	4.2±0.2V
0111	Reserved
1xxx	Reserved

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						SO CSVE	MO CSVE
R/W Attribute	R0,WX						R,WX	R,WX
Protection Attribute	WPS							
Initial Value	000000						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.

[bit7:2] Reserved: Reserved Bits

[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						0

[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	-

5.6. System Register Group (SYSC0)

The registers in this group are system control-related registers.

5.6.1. 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

5.6.2. 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

5.6.3. 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		PSSDFCLR0	RUNDFCLR0	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

5.6.4. 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.

5.6.5. 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.

5.6.6. 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".

5.6.7. 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".

5.6.8. 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.*

5.6.9. 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.

5.6.10. 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.

5.6.11. System ID Register (SYSC0_SYSIDR)

The SYSC0_SYSIDR register displays model information.

bit	31	0
Field	SYSID	
R/W Attribute	R,WX	
Protection Attribute	WPS	
Initial Value	00010000_00000000_01000101_11001111: S6J312AHzC*, S6J3129HzC* * z: A/B	

[bit31:0] CHIPID: Chip ID Display Bits

This bit is a specific fixed value. Write access has no effect.

5.6.12 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	00000000_00010001_00000011_00000000	

[bit31:0] PFID: Platform ID Display Bits

This bit is a specific fixed value. The value has no meaning.

Write access has no effect.

5.7. 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	PSSPAD CTRL	IO3RSTC	Reserved					
R/W Attribute	R/W	R/W	R0,WX					
Protection Attribute	WPS							
Initial Value	0	1	000000					

bit	15	14	13	12	11	10	9	8
Field	Reserved							BRAM SC
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				EXSTBCNT			
R/W Attribute	R0,WX				R/W			
Protection Attribute	WPS							
Initial Value	0000				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 enabled

[bit21:9] Reserved

[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] Reserved: Reserved Bits

[bit3:0] EXSTBCNT: External Power Supply Stabilization Time Setting Bit

This bit sets the stabilization time for 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

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							DBGDONE
R/W Attribute	R0,WX							R/W
Protection Attribute	-							
Initial Value	0000000							1

[bit31:1] Reserved: Reserved Bits

[bit0] DBGDONE: Debugger Status Bit

This bit indicates the debugger setting status.

Bit	Description
0	The debugger is connected.
1	The debugger is not connected, or debugger setting has completed.

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 releasing 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 Release Setting Bits

These bits compose the register for releasing protection keys.

Write

bit31:0	Description
0x5CAC_CE55	Lock release
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
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
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
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
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
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
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															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
100	PLL0 clock
101	SSCG0 clock
110	Setting prohibited
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
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/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

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	Disable clock oscillation.
1	Enable clock oscillation.

[bit27] ENCLKCD3B0: CD3B0 Clock Oscillation Enable Bit

This bit sets the CD3B0 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit26] ENCLKCD3A1: CD3A1 Clock Oscillation Enable Bit

This bit sets the CD3A1 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit25] ENCLKCD3A0: CD3A0 Clock Oscillation Enable Bit

This bit sets the CD3A0 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit24] ENCLKCD3: CD3 Clock Oscillation Enable Bit

This bit sets 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 sets the CD2B1 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit19] ENCLKCD2B0: CD2B0 Clock Oscillation Enable Bit

This bit sets the CD2B0 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit18] ENCLKCD2A1: CD2A1 Clock Oscillation Enable Bit

This bit sets the CD2A1 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit17] ENCLKCD2A0: CD2A0 Clock Oscillation Enable Bit

This bit sets the CD2A0 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit16] ENCLKCD2: CD2 Clock Oscillation Enable Bit

This bit sets 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 sets the CD1B1 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit11] ENCLKCD1B0: CD1B0 Clock Oscillation Enable Bit

This bit sets the CD1B0 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit10] ENCLKCD1A1: CD1A1 Clock Oscillation Enable Bit

This bit sets the CD1A1 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit9] ENCLKCD1A0: CD1A0 Clock Oscillation Enable Bit

This bit sets the CD1A0 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit8] ENCLKCD1: CD1 Clock Oscillation Enable Bit

This bit sets 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 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			ENCLK CD5B1	ENCLK CD5B0	ENCLK CD5A1	ENCLK CD5A0	ENCLK CD5
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	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/W	R/W	R/W	R/W	R/W
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	Disable clock oscillation.
1	Enable clock oscillation.

[bit11] ENCLKCD5B0: CD5B0 Clock Oscillation Enable Bit

This bit sets the CD5B0 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit10] ENCLKCD5A1: CD5A1 Clock Oscillation Enable Bit

This bit sets the CD5A1 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit9] ENCLKCD5A0: CD5A0 Clock Oscillation Enable Bit

This bit sets the CD5A0 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit8] ENCLKCD5: CD5 Clock Oscillation Enable Bit

This bit sets 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 sets the CD4B1 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit3] ENCLKCD4B0: CD4B0 Clock Oscillation Enable Bit

This bit sets the CD4B0 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit2] ENCLKCD4A1 : CD4A1 Clock Oscillation Enable Bit

This bit sets the CD4A1 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit1] ENCLKCD4A0: CD4A0 Clock Oscillation Enable Bit

This bit sets the CD4A0 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit0] ENCLKCD4: CD4 Clock Oscillation Enable Bit

This bit sets the CD4 clock.

Bit	Description
0	Disable clock oscillation.
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
0_0000	No division
0_0001	Divided by 2
0_0010	Divided by 3
0_0011	Divided by 4
. . .	. . .
1_1101	Divided by 30
1_1110	Divided by 31
1_1111	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
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.

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
100	PLL0 clock
101	SSCG0 clock
110	Setting prohibited
111	Clock fixed at "L"

Note:

- *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
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
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
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
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															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
100	PLL0 clock
101	SSCG0 clock
110	Setting prohibited
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
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/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

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	Disable clock oscillation.
1	Enable clock oscillation.

[bit27] ENCLKCD3B0: CD3B0 Clock Oscillation Enable Bit

This bit sets the CD3B0 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit26] ENCLKCD3A1: CD3A1 Clock Oscillation Enable Bit

This bit sets the CD3A1 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit25] ENCLKCD3A0: CD3A0 Clock Oscillation Enable Bit

This bit sets the CD3A0 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit24] ENCLKCD3: CD3 Clock Oscillation Enable Bit

This bit sets 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 sets the CD2B1 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit19] ENCLKCD2B0: CD2B0 Clock Oscillation Enable Bit

This bit sets the CD2B0 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit18] ENCLKCD2A1: CD2A1 Clock Oscillation Enable Bit

This bit sets the CD2A1 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit17] ENCLKCD2A0: CD2A0 Clock Oscillation Enable Bit

This bit sets the CD2A0 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit16] ENCLKCD2: CD2 Clock Oscillation Enable Bit

This bit sets 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 sets the CD1B1 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit11] ENCLKCD1B0: CD1B0 Clock Oscillation Enable Bit

This bit sets the CD1B0 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit10] ENCLKCD1A1: CD1A1 Clock Oscillation Enable Bit

This bit sets the CD1A1 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit9] ENCLKCD1A0: CD1A0 Clock Oscillation Enable Bit

This bit sets the CD1A0 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit8] ENCLKCD1: CD1 Clock Oscillation Enable Bit

This bit sets 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 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			ENCLK CD5B1	ENCLK CD5B0	ENCLK CD5A1	ENCLK CD5A0	ENCLK CD5
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	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/W	R/W	R/W	R/W	R/W
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	Disable clock oscillation.
1	Enable clock oscillation.

[bit11] ENCLKCD5B0: CD5B0 Clock Oscillation Enable Bit

This bit sets the CD5B0 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit10] ENCLKCD5A1: CD5A1 Clock Oscillation Enable Bit

This bit sets the CD5A1 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit9] ENCLKCD5A0: CD5A0 Clock Oscillation Enable Bit

This bit sets the CD5A0 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit8] ENCLKCD5: CD5 Clock Oscillation Enable Bit

This bit sets 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 sets the CD4B1 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit3] ENCLKCD4B0: CD4B0 Clock Oscillation Enable Bit

This bit sets the CD4B0 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit2] ENCLKCD4A1: CD4A1 Clock Oscillation Enable Bit

This bit sets the CD4A1 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit1] ENCLKCD4A0: CD4A0 Clock Oscillation Enable Bit

This bit sets the CD4A0 clock.

Bit	Description
0	Disable clock oscillation.
1	Enable clock oscillation.

[bit0] ENCLKCD4: CD4 Clock Oscillation Enable Bit

This bit sets the CD4 clock.

Bit	Description
0	Disable clock oscillation.
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

[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
0_0000	No division
0_0001	Divided by 2
0_0010	Divided by 3
0_0011	Divided by 4
. . .	. . .
1_1101	Divided by 30
1_1110	Divided by 31
1_1111	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

[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

[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

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	0000			0000				

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			LAPP1A CSL	Reserved			LAPP0A CSL
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			LCP1A CSL	Reserved			LCP0A CSL
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
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
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	00000				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
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 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
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
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 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
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															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
100	PLL0 clock
101	SSCG0 clock
110	Setting prohibited
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
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_APPCKSRER0 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
0_0000	No division
0_0001	Divided by 2
0_0010	Divided by 3
0_0011	Divided by 4
. . .	. . .
1_1101	Divided by 30
1_1110	Divided by 31
1_1111	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
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"

[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
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
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
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
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"

[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
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
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
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
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
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
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
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
100	PLL0 clock
101	SSCG0 clock
110	Setting prohibited
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
100	PLL0 clock
101	SSCG0 clock
110	Setting prohibited
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
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
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
0_0000	No division
0_0001	Divided by 2
0_0010	Divided by 3
0_0011	Divided by 4
. . .	. . .
1_1101	Divided by 30
1_1110	Divided by 31
1_1111	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, the Memory Configuration clock, the SCU clock, and the MCUCH clock. 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-T00P00R01L03-E1-GBB02

1. Overview

This section provides an overview of the low-voltage detection.

1.1. Features

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: 0.95 V±0.05 V
- 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.

- 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:0.97V±0.05V
  Detection voltage:1.07V±0.05V
- Operation: Always active.
  The RAM contents after a 1.2 V power supply low-voltage detection reset cannot be guaranteed.

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:3.8V±0.2V
  Detection voltage:4.0V±0.2V
  Detection voltage:4.2V±0.2V

- 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 1-1Low-Voltage Detection Specifications Table

Kind	Detection Voltage	Detection Error:	Operation	Note
Internal Low-voltage Detection	0.95V	±0,05V	Reset is Always active	Values in RAM can not be guaranteed
1.2V Power Supply Low-voltage Detection	0.97V 1.07V	±0.05 V	Interrupt or Reset Switchable to active/inactive state through user settings	Values in RAM can not be guaranteed
3.3/5.0V Power Supply Low-voltage Detection	3.8V 4.0V 4.2V	±0.2 V	Interrupt or Reset Switchable to active/inactive state through user settings	

The following registers are defined below as "LVD registers."

Table 1-2 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 1-3 LVD Operation Select Bits

LVDL1S	1.2 V power supply low-voltage detection operation select bit ch1
LVDL2S	1.2 V 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 1-4 LVD Ready Bits

LVDL1RDY	1.2 V power supply low-voltage detection ready bit ch1
LVDL2RDY	1.2 V power supply low-voltage detection ready bit ch2
LVDH1RDY	3.3/5.0 V power supply low-voltage detection ready bit ch1
LVDH2RDY	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 1-5 LVD Reset Source Bits

LVDL1E	1.2 V power supply low-voltage detection operation enable bit ch1
LVDL2E	1.2 V power supply low-voltage detection operation enable bit ch2
LVDH1E	3.3/5.0 V power supply low-voltage detection operation enable bit ch1
LVDL2E	3.3/5.0 V power supply low-voltage detection operation enable bit ch2

Notes:

- *In the S6J3120 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
- *In the S6J3120 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.*
- *Ch.2 is not implemented*

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

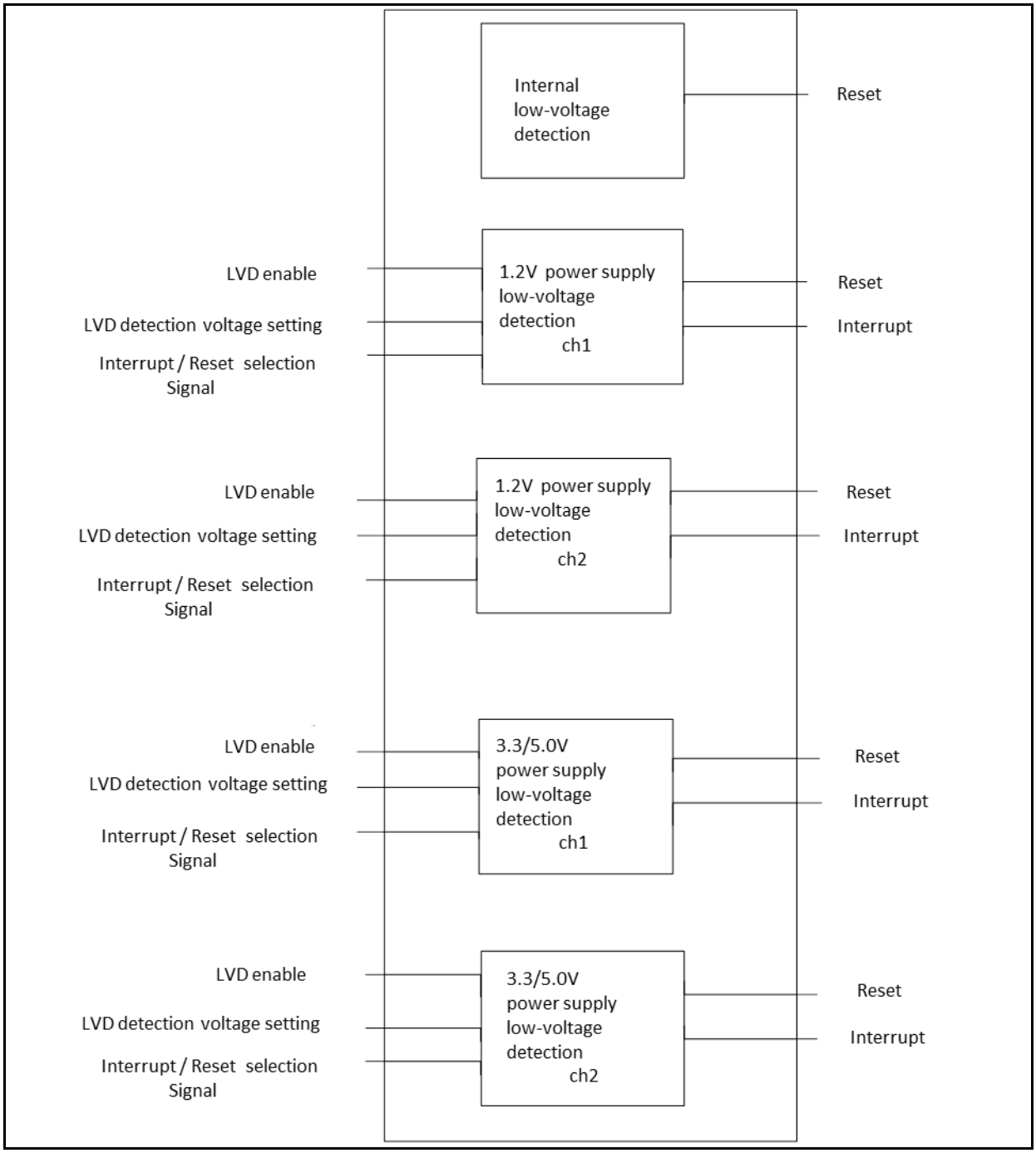


Figure 2-2 Internal Low-Voltage Detection

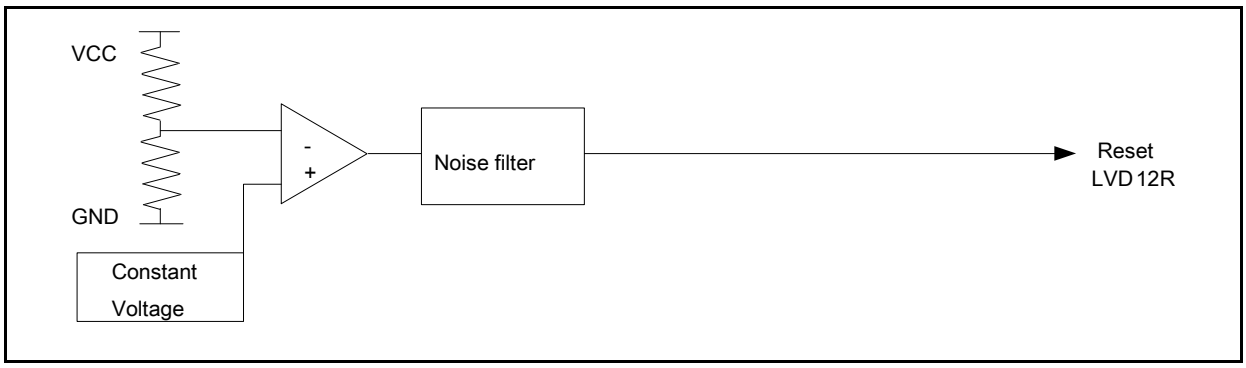
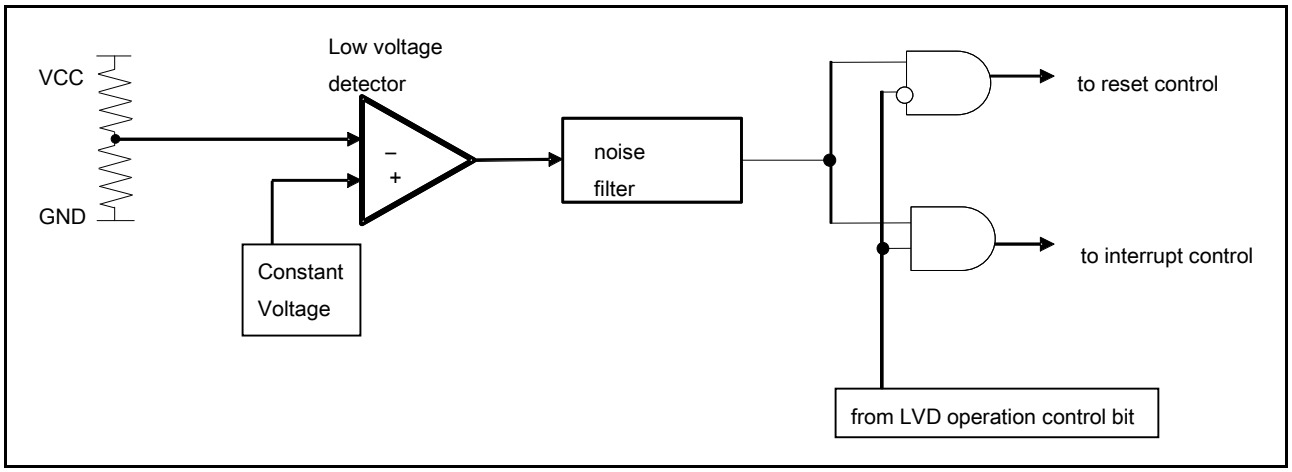


Figure 2-3 1.2 V Power Supply Low-Voltage Detection, 3.3/5.0 V Power Supply Low-Voltage Detection



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."

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.

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 µs). 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."

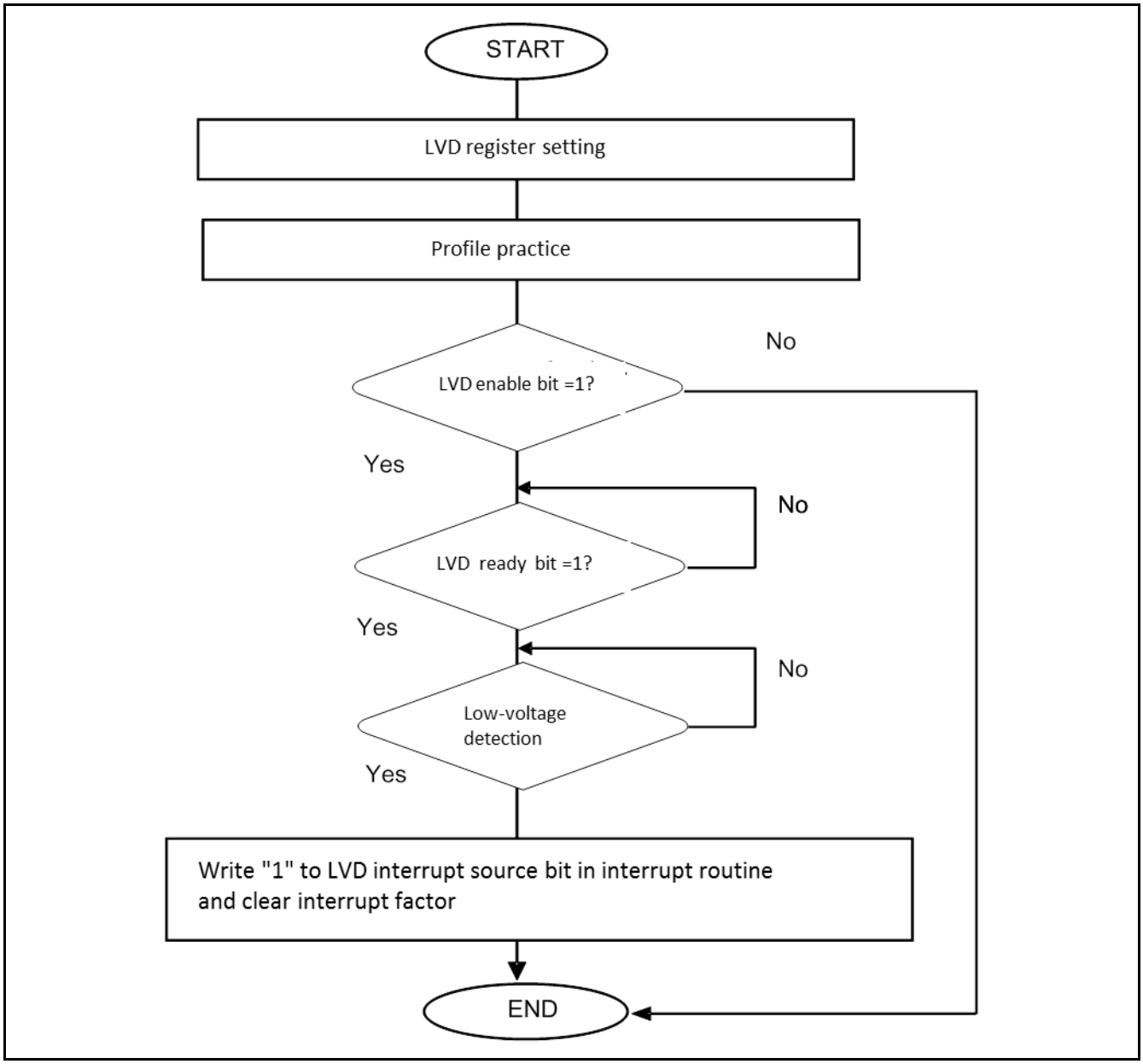
Reset will occur when LVD operation select bit is changed from "interrupt generation" to "reset generation" while interrupt is generated by low voltage detection circuit due to external supply voltage being lower than detection voltage.

4. Setting Procedure Examples

This section explains setting procedure examples of the low-voltage detection.

Setting the Low-Voltage Detection for Interrupts

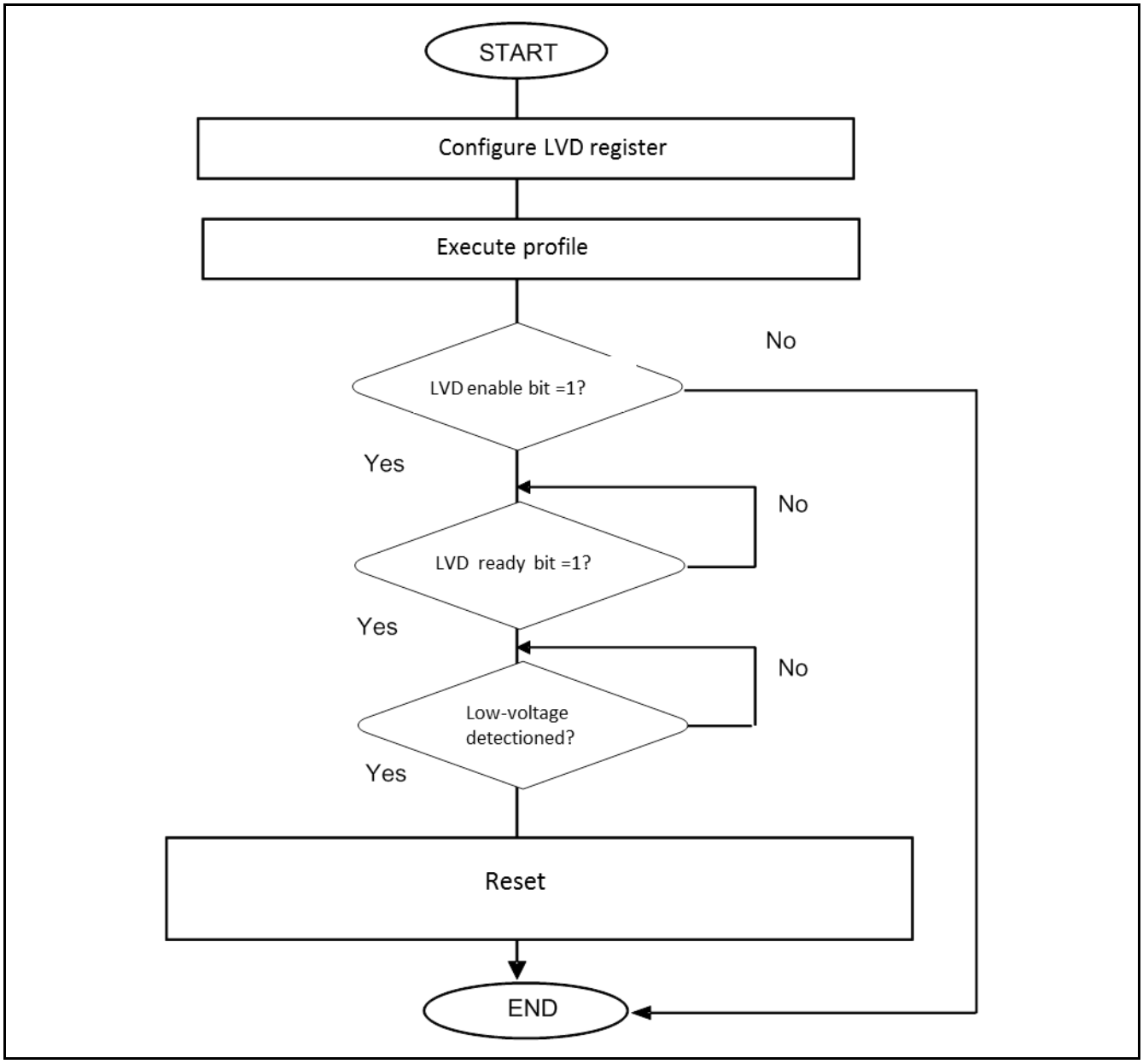
Figure 4-1 Setting the Low-Voltage Detection for Interrupts



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



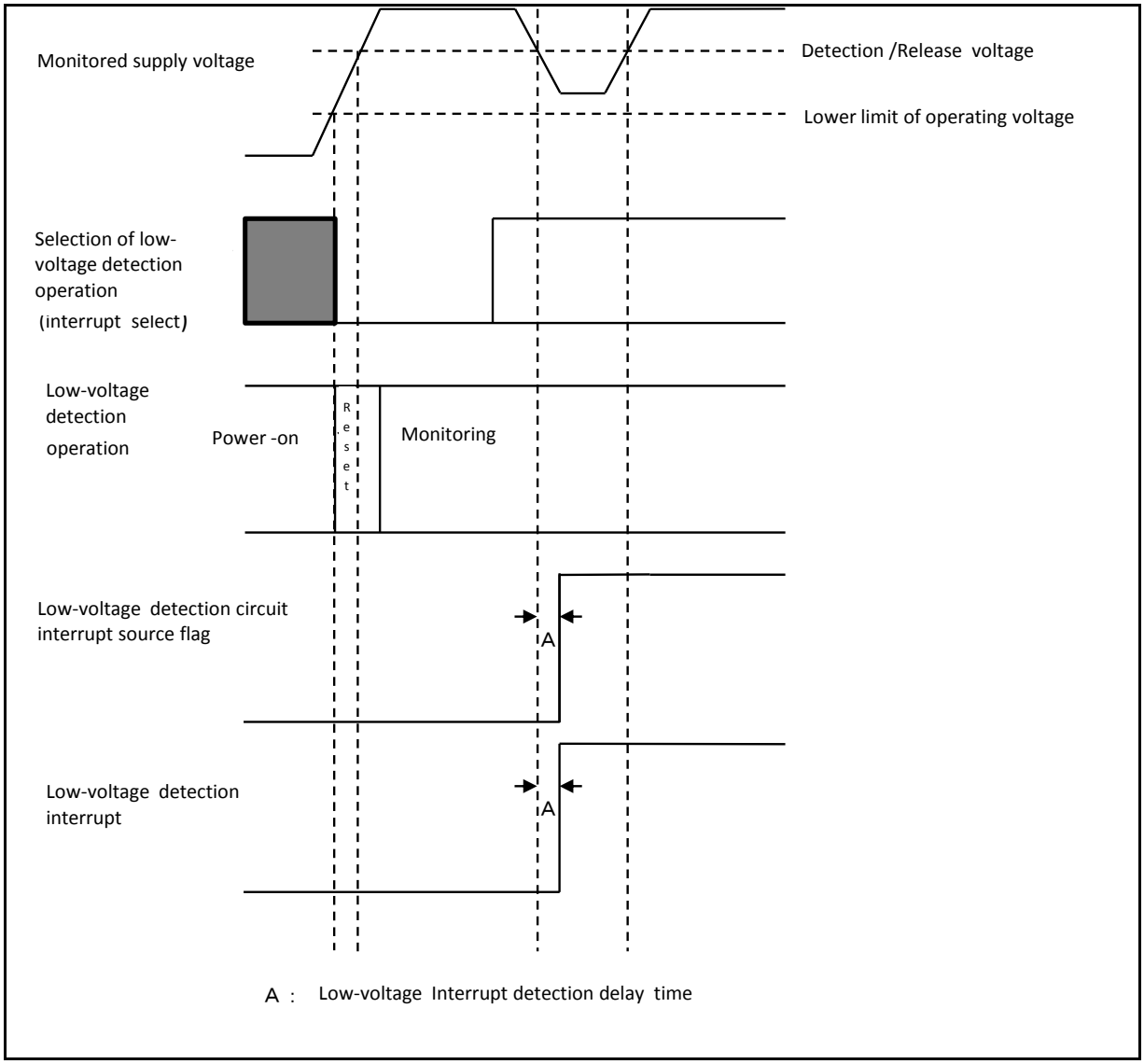
The execution of a profile applies the low-voltage detection settings made for the profile register. Low-voltage monitoring begins when the LVD ready bit becomes "1". Then, a reset is generated upon the detection of a low voltage.

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.

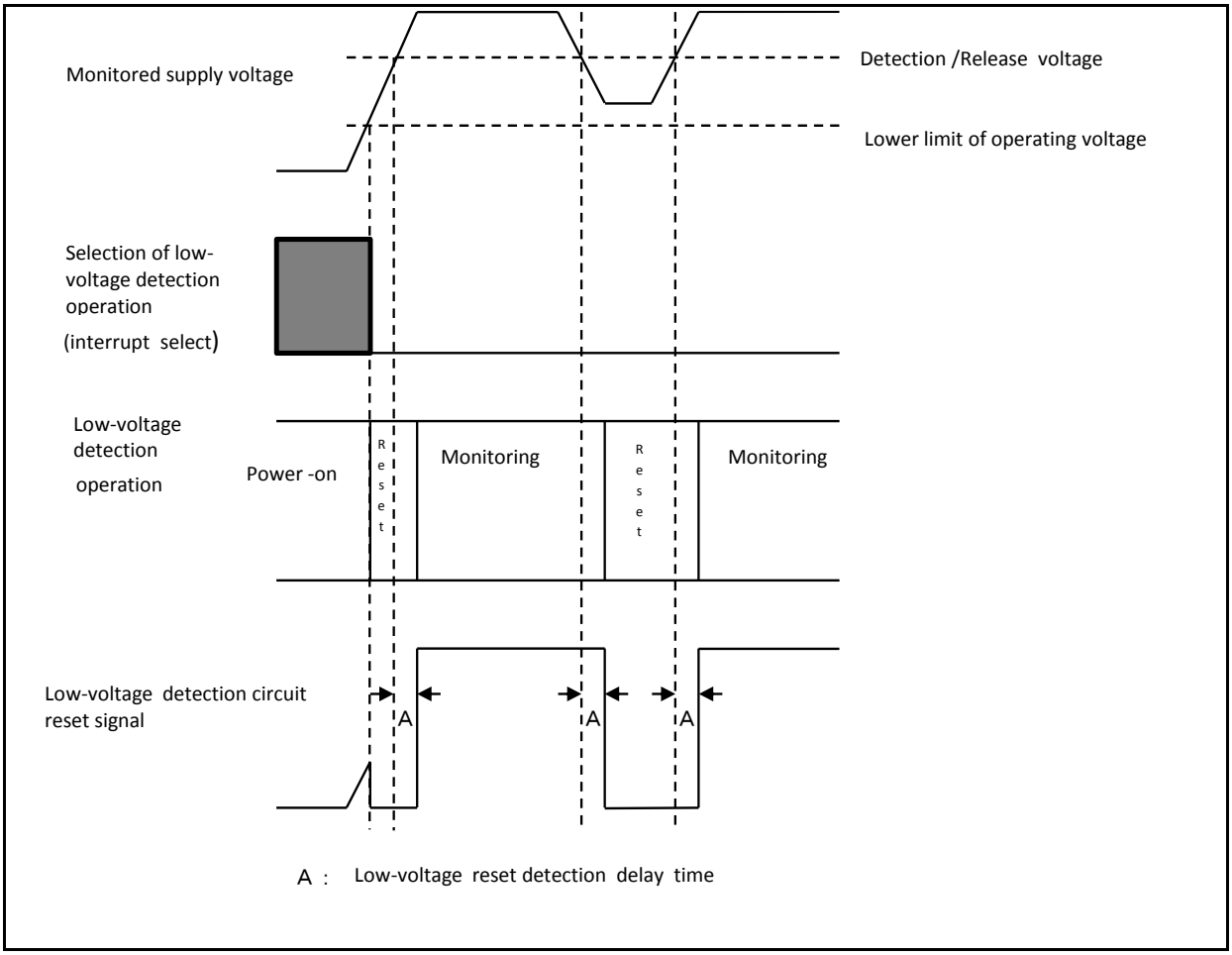
Figure 5-1 Timing Chart When Low-Voltage Detection Interrupts Are Enabled



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

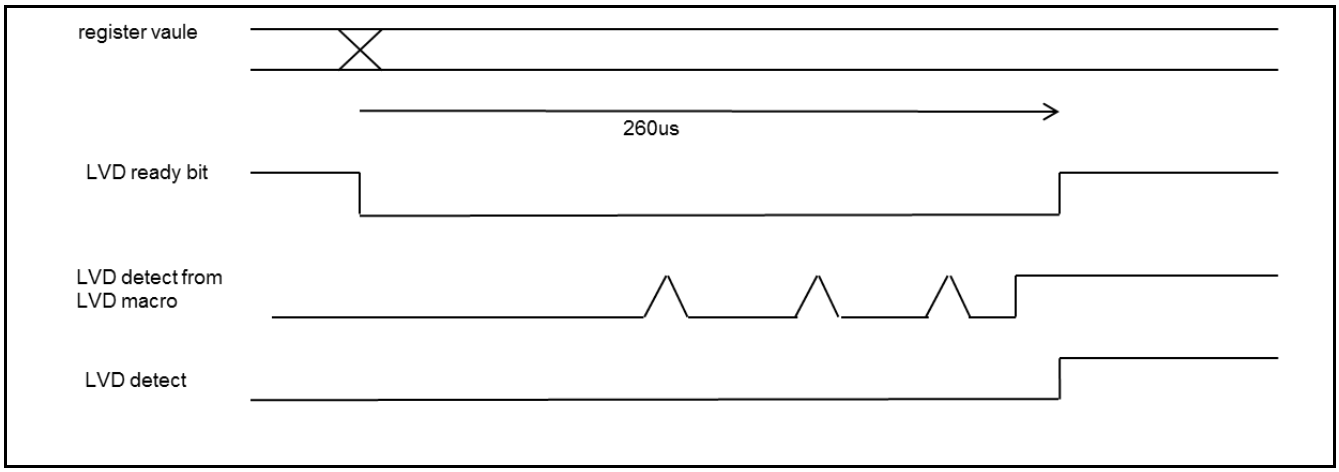


When the Register Setting Value is Changed

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 :	low voltage detect enable for 1.2V power supply, ch1
LVDL2E :	low voltage detect enable for 1.2V power supply, ch2
LVDH1E :	low voltage detect enable for 3.3V or 5.5V power supply, ch1
LVDH2E :	low voltage detect enable for 3.3V or 5.5V power supply, 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 µs.

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**

CSV-T00P00R01L02-E1-GBB00

1. Overview

The clock supervisor is used to detect clock operation abnormalities. This section provides an overview of the clock supervisor.

1. Detection 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.

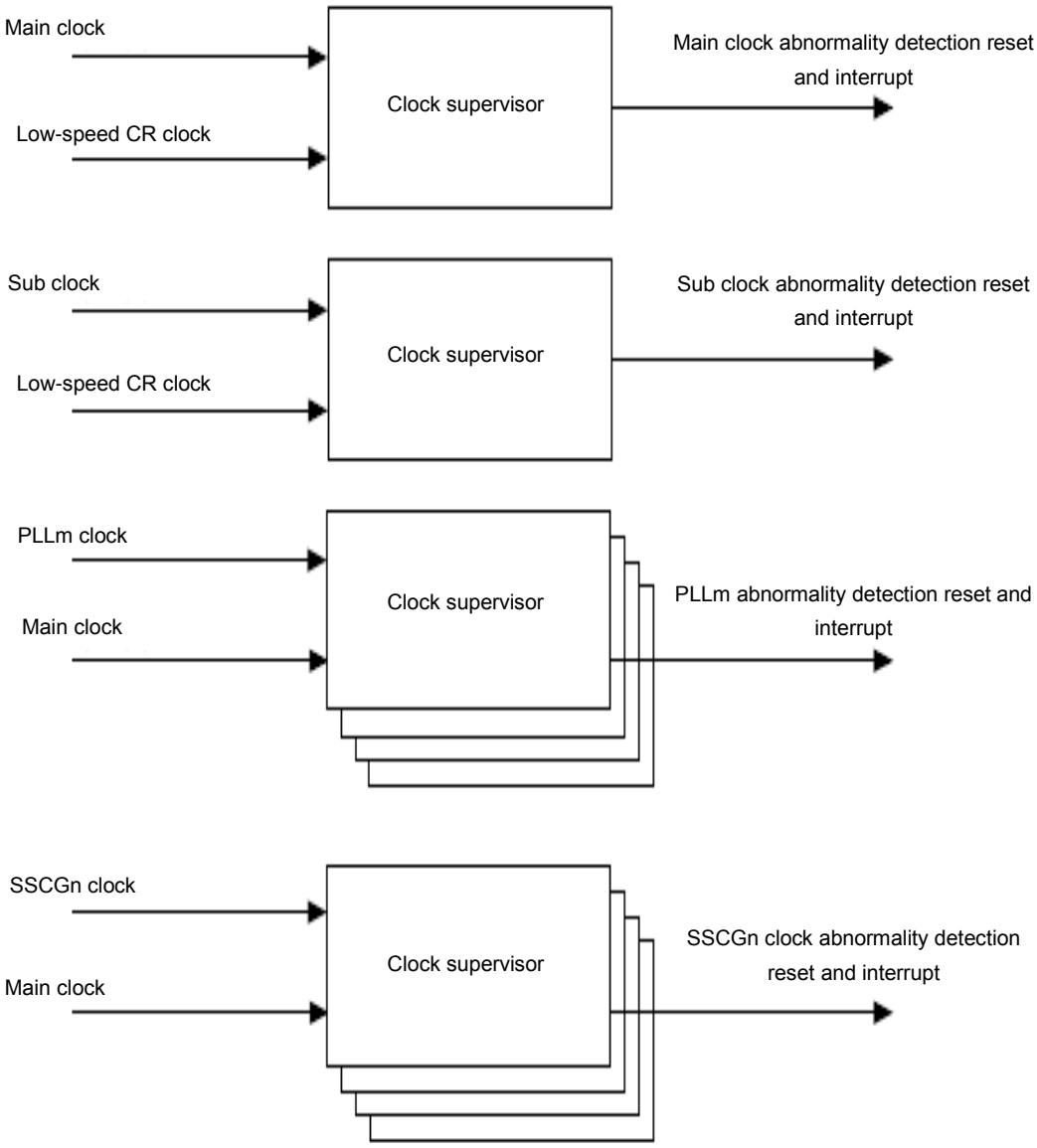
Table 1-1 Overview of Each Clock Supervisor Instances

Monitoring Clock	Reference Clock	Clock Supervisor Feature
Main clock	Low-speed CR clock	Detection of stop or frequency range abnormality of the clock.
Sub clock	Low-speed 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	
SSCG2clock	Main clock	
SSCG3 clock	Main clock	

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 0 monitors the main clock. The reference clock is the low-speed CR clock. The monitoring clock is the main clock. Upon detecting an abnormality of the main clock, the supervisor generates a reset or interrupt.

Sub Clock Supervisor

- Sub clock supervisor 0 monitors the sub clock. The reference clock is the low-speed CR clock.
- Upon detecting an abnormality of the sub clock, the supervisor generates a reset or interrupt.

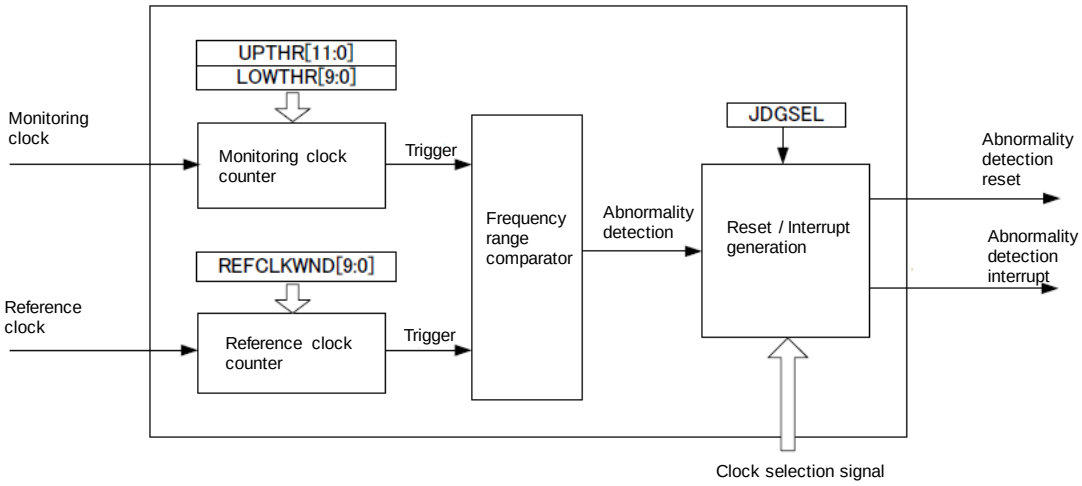
PLLm Clock Supervisor

- The PLLm clock supervisor monitors 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 PLLn clock supervisor monitors the SSCG PLLn clock. The reference clock is the main clock.
- Upon detecting an abnormality of the SSCG PLLn clock, the supervisor generates a reset or interrupt.

Figure 2-2 Block Diagram of Clock Supervisor



Monitoring Clock Counter

There are 2 down counters for counting with the monitoring clock. The counters load the values that are set in the upper-limit threshold value bits (UPTHR[9:0]) 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 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. 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 monitoring clock.

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_CSVPLLCFGR0) and PLLm clock supervisor setting register 1 (SYSC_CSVPLLCFGR1) 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).
- 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

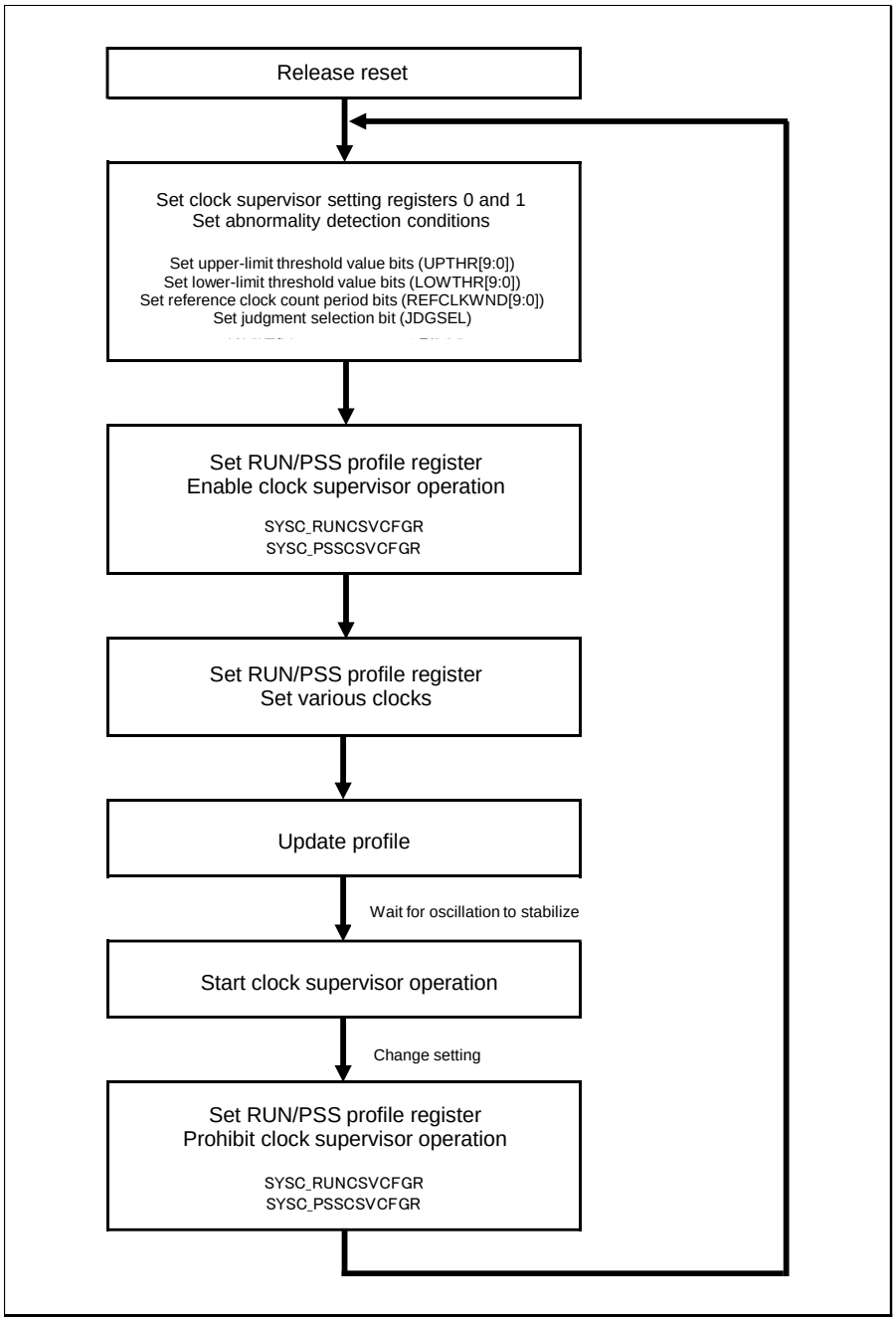
- SSCG PLLn clock supervisor setting register 0 (SYSC_CSVSPCFGR0) and SSCG PLLn clock supervisor setting register 1 (SYSC_CSVSPCFGR1) are used for settings.
- Upon detecting an abnormality of the SSCG PLL clock and generating a reset, the SSCG PLL clock supervisor can confirm it with the SSCG PLL clock supervisor reset detection bit (CSVSPR) 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).

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 monitoring clock and reference clock oscillation has stabilized.
- The clock supervisor can enter the clock abnormality detection state by gating the monitoring clock for self-testing.

5. Registers

This section lists the clock supervisor registers.

The clock supervisor registers are as follows:

- RUN profile registers
- PSS profile registers
- APPLIED profile registers
- Status profile registers
- CSV configuration register group

For details on the RUN profile registers, PSS profile registers, APPLIED profile registers, and status 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	5.7
SYSC_CSVPLL1CFGR1	PLL1 clock supervisor setting register 1	5.8
SYSC_CSVPLL2CFGR0	PLL2 clock supervisor setting register 0	5.9
SYSC_CSVPLL2CFGR1	PLL2 clock supervisor setting register 1	5.10
SYSC_CSVPLL3CFGR0	PLL3 clock supervisor setting register 0	5.11
SYSC_CSVPLL3CFGR1	PLL3 clock supervisor setting register 1	5.12
SYSC_CSVSP0CFGR0	SSCG PLL0 clock supervisor setting register 0	5.13
SYSC_CSVSP0CFGR1	SSCG PLL0 clock supervisor setting register 1	5.14
SYSC_CSVSP1CFGR0	SSCG PLL1 clock supervisor setting register 0	5.15
SYSC_CSVSP1CFGR1	SSCG PLL1 clock supervisor setting register 1	5.16
SYSC_CSVSP2CFGR0	SSCG PLL2 clock supervisor setting register 0	5.17
SYSC_CSVSP2CFGR1	SSCG PLL2 clock supervisor setting register 1	5.18
SYSC_CSVSP3CFGR0	SSCG PLL3 clock supervisor setting register 0	5.19
SYSC_CSVSP3CFGR1	SSCG PLL3 clock supervisor setting register 1	5.20
SYSC_CSVOUTER	Clock supervisor output enable register	5.21
SYSC_CSVTESTR	Clock supervisor test register	5.22

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 monitoring clock count value. If the monitoring 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 monitoring clock count value. If the monitoring 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 monitoring 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							
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

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.

Bit	Description
0	Monitoring clock is selected as the clock for clock domain 0/MCUC clock domain /software watchdog timer.
1	Monitoring clock is selected as the clock for clock domain 0/ MCUC clock domain.

[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 monitoring clock count value. If the monitoring 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 monitoring clock count value. If the monitoring 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 monitoring 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

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.

Bit	Description
0	Monitoring clock is selected as the clock for clock domain 0/MCUC clock domain /software watchdog timer.
1	Monitoring clock is selected as the clock for clock domain 0/ MCUC clock domain.

[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. 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.

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 monitoring clock count value. If the monitoring 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 monitoring clock count value. If the monitoring 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 monitoring 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. PLL0 Clock Supervisor Setting Register 1 (SYSC_CSVPLL0CFGR1)

PLL0 clock supervisor setting register 1 (SYSC_CSVPLL0CFGR1) 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

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.

Bit	Description
0	Monitoring clock is selected as the clock for clock domain 0/MCUC clock domain
1	- (same condition)

[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. 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.

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 monitoring clock count value. If the monitoring 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 monitoring clock count value. If the monitoring 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 monitoring 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. PLL1 Clock Supervisor Setting Register 1 (SYSC_CSVPLL1CFGR1)

PLL clock supervisor setting register 1 (SYSC_CSVPLLCFGR1) 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

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.

Bit	Description
0	Generate a reset.
1	No conditions (Generate an 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.*

5.9. PLL2 Clock Supervisor Setting Register 0 (SYSC_CSVPLL2CFGR0)

PLL2 clock supervisor setting register 0 (SYSC_CSVPLL2CFGR0) 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 monitoring clock count value. If the monitoring 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 monitoring clock count value. If the monitoring 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 monitoring clock cycle.*
- *Set the upper-limit threshold value bit (UPTHR) value to be greater than the lower-limit threshold value bit (LOWTHR) value.*

5.10. PLL2 Clock Supervisor Setting Register 1 (SYSC_CSVPLL2CFGR1)

PLL2 clock supervisor setting register 1 (SYSC_CSVPLL2CFGR1) 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

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.

Bit	Description
0	Generate a reset.
1	No conditions (Generate an 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.*

5.11. PLL3 Clock Supervisor Setting Register 0 (SYSC_CSVPLL3CFGR0)

PLL3 clock supervisor setting register 0 (SYSC_CSVPLL3CFGR0) 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 monitoring clock count value. If the monitoring 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 monitoring clock count value. If the monitoring 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 monitoring clock cycle.*
- *Set the upper-limit threshold value bit (UPTHR) value to be greater than the lower-limit threshold value bit (LOWTHR) value.*

5.12. PLL3 Clock Supervisor Setting Register 1 (SYSC_CSVPLL3CFGR1)

PLL3 clock supervisor setting register 1 (SYSC_CSVPLL3CFGR1) 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

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.

Bit	Description
0	Generate a reset.
1	No conditions (Generate an 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.*

5.13. SSCG PLL0 Clock Supervisor Setting Register 0 (SYSC_CSVSP0CFGR0)

SSCG PLL0 clock supervisor setting register 0 (SYSC_CSVSP0CFGR0) 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 monitoring clock count value. If the monitoring 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 monitoring clock count value. If the monitoring 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 monitoring clock cycle.*
- *Set the upper-limit threshold value bit (UPTHR) value to be greater than the lower-limit threshold value bit (LOWTHR) value.*

5.14. SSCG PLL0 Clock Supervisor Setting Register 1 (SYSC_CSVSP0CFGR1)

SSCG PLL0 clock supervisor setting register 1 (SYSC_CSVSP0CFGR1) 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

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.

Bit	Description
0	Monitoring clock is selected as the clock for clock domain 0/MCUC clock domain
1	- (same condition)

[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.*

5.15. SSCG PLL1 Clock Supervisor Setting Register 0 (SYSC_CSVSP1CFGR0)

SSCG PLL1 clock supervisor setting register 0 (SYSC_CSVSP1CFGR0) 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 monitoring clock count value. If the monitoring 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 monitoring clock count value. If the monitoring 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 monitoring clock cycle.*
- *Set the upper-limit threshold value bit (UPTHR) value to be greater than the lower-limit threshold value bit (LOWTHR) value.*

5.16. SSCG PLL1 Clock Supervisor Setting Register 1 (SYSC_CSVSP1CFGR1)

SSCG PLL1 clock supervisor setting register 1 (SYSC_CSVSP1CFGR1) 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

Reset generation or interrupt generation for when an abnormality is detected can be selected with this bit.

An interrupt can be generated when reset generation condition is not correspondence.

Bit	Description
0	Generate a reset.
1	No conditions (Generate an 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.*

5.17. SSCG PLL2 Clock Supervisor Setting Register 0 (SYSC_CSVSP2CFGR0)

SSCG PLL2 clock supervisor setting register 0 (SYSC_CSVSP2CFGR0) 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 monitoring clock count value. If the monitoring 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 monitoring clock count value. If the monitoring 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 monitoring clock cycle.*
- *Set the upper-limit threshold value bit (UPTHR) value to be greater than the lower-limit threshold value bit (LOWTHR) value.*

5.18. SSCG PLL2 Clock Supervisor Setting Register 1 (SYSC_CSVSP2CFGR1)

SSCG PLL2 clock supervisor setting register 1 (SYSC_CSVSP2CFGR1) 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

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.

Bit	Description
0	Generate a reset.
1	No conditions (Generate an 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.*

5.19. SSCG PLL3 Clock Supervisor Setting Register 0 (SYSC_CSVSP3CFGR0)

SSCG PLL3 clock supervisor setting register 0 (SYSC_CSVSP3CFGR0) 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 monitoring clock count value. If the monitoring 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 monitoring clock count value. If the monitoring 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 monitoring clock cycle.*
- *Set the upper-limit threshold value bit (UPTHR) value to be greater than the lower-limit threshold value bit (LOWTHR) value.*

5.20. SSCG PLL3 Clock Supervisor Setting Register 1 (SYSC_CSVSP3CFGR1)

SSCG PLL3 clock supervisor setting register 1 (SYSC_CSVSP3CFGR1) 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

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.

Bit	Description
0	Generate a reset.
1	No conditions (Generate an 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.*

5.21. 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 simultaneous input of "L" to a power-on reset, an internal power supply low-voltage detection reset, and the NMIX pin + RSTX pin. It is not affected by other types of reset.*

5.22. Clock Supervisor Test Register (SYSC_CSVTESTR)

The clock supervisor test register (SYSC_CSVTESTR) can perform function tests using input clock gating.

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				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

[bit31:20] 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)

[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)

[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 monitoring clock and reference clock.
- To stop the clock supervisor, simultaneously stop oscillation of the monitoring clock.
- After a reset is generated by the detection of a clock abnormality, MCU will operate with FastCR clock. At this time, CSV cause clock must not be selected again
- 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-T00P00R01L02-E1-GBB01

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 3 source clocks are the high-speed CR clock, low-speed CR clock, and main clock, and they are prepared 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 of 0.32 to 0.8 ms.

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 of 0.43 to 1.28 ms.

Main Clock Timer

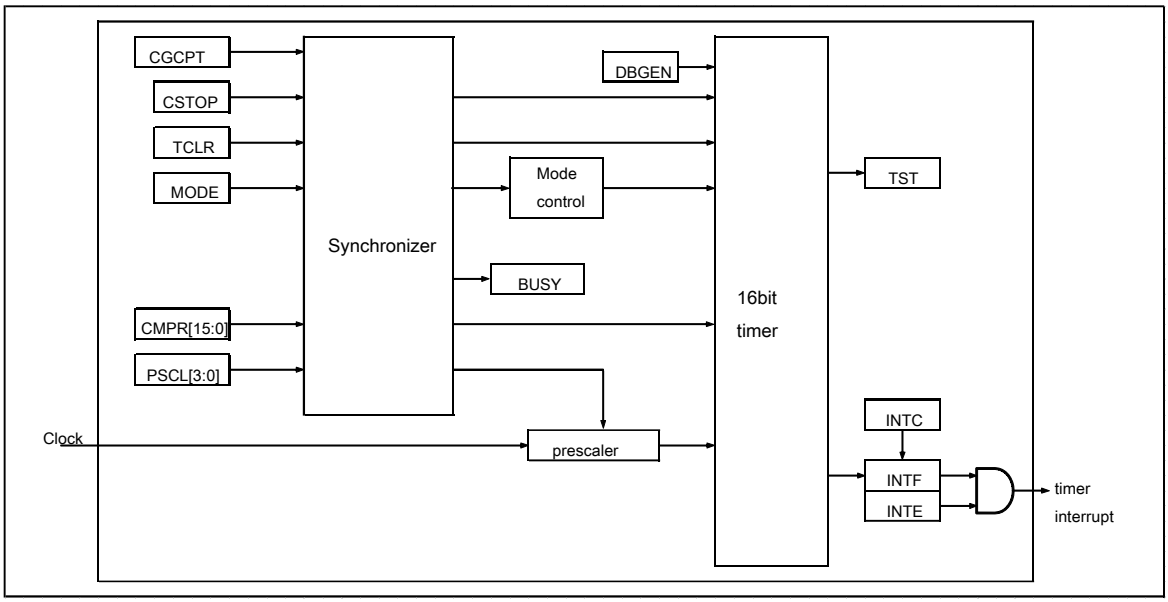
The main clock timer counts the main clock. This timer is used to count time to wait for the main clock oscillation to stabilize.

The main clock timer has an initial value of 8.19 ms (at 4 MHz) 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

- 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.

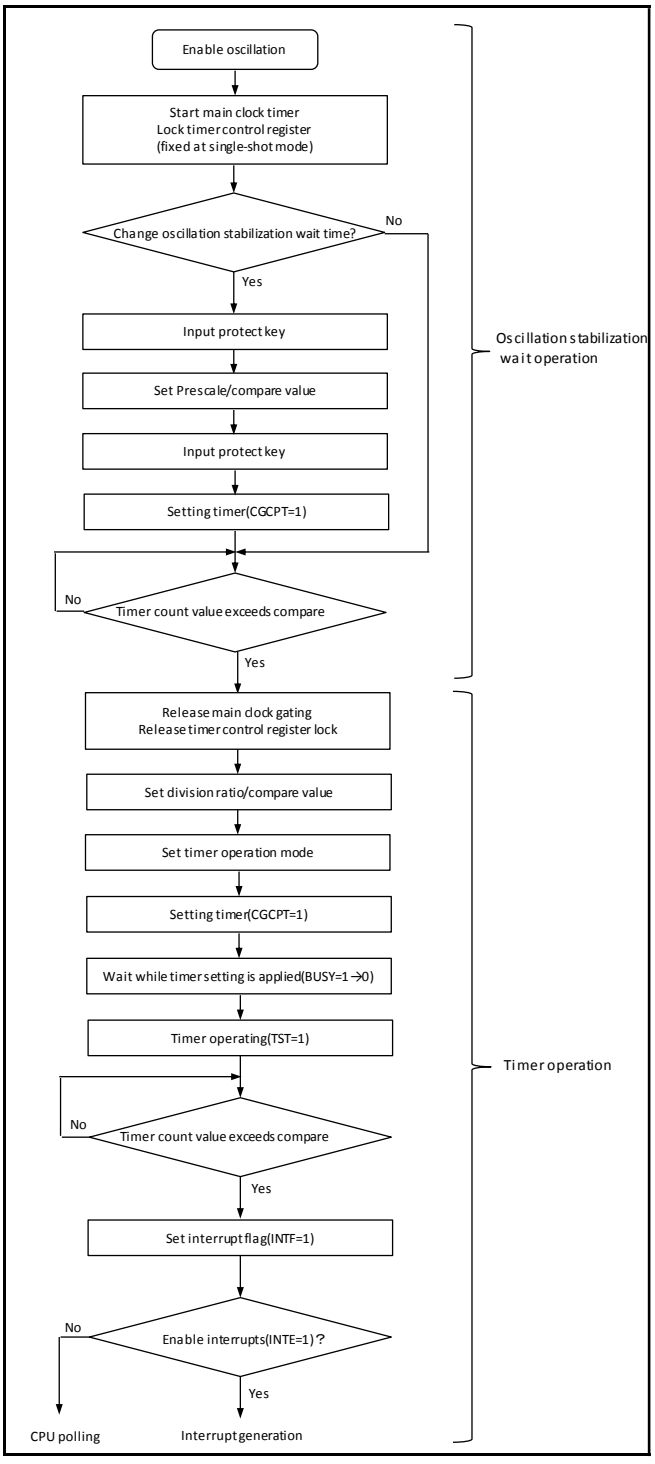
Timer Operation

- 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

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

Note:

- *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

Bits				Description
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)	0.32 to 0.8 [ms]

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

Note:

- *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)	0.43 to 1.28 [ms]

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".

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 DS 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**

RTC-T00P00R01L05-E1-GBB00

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 RC 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

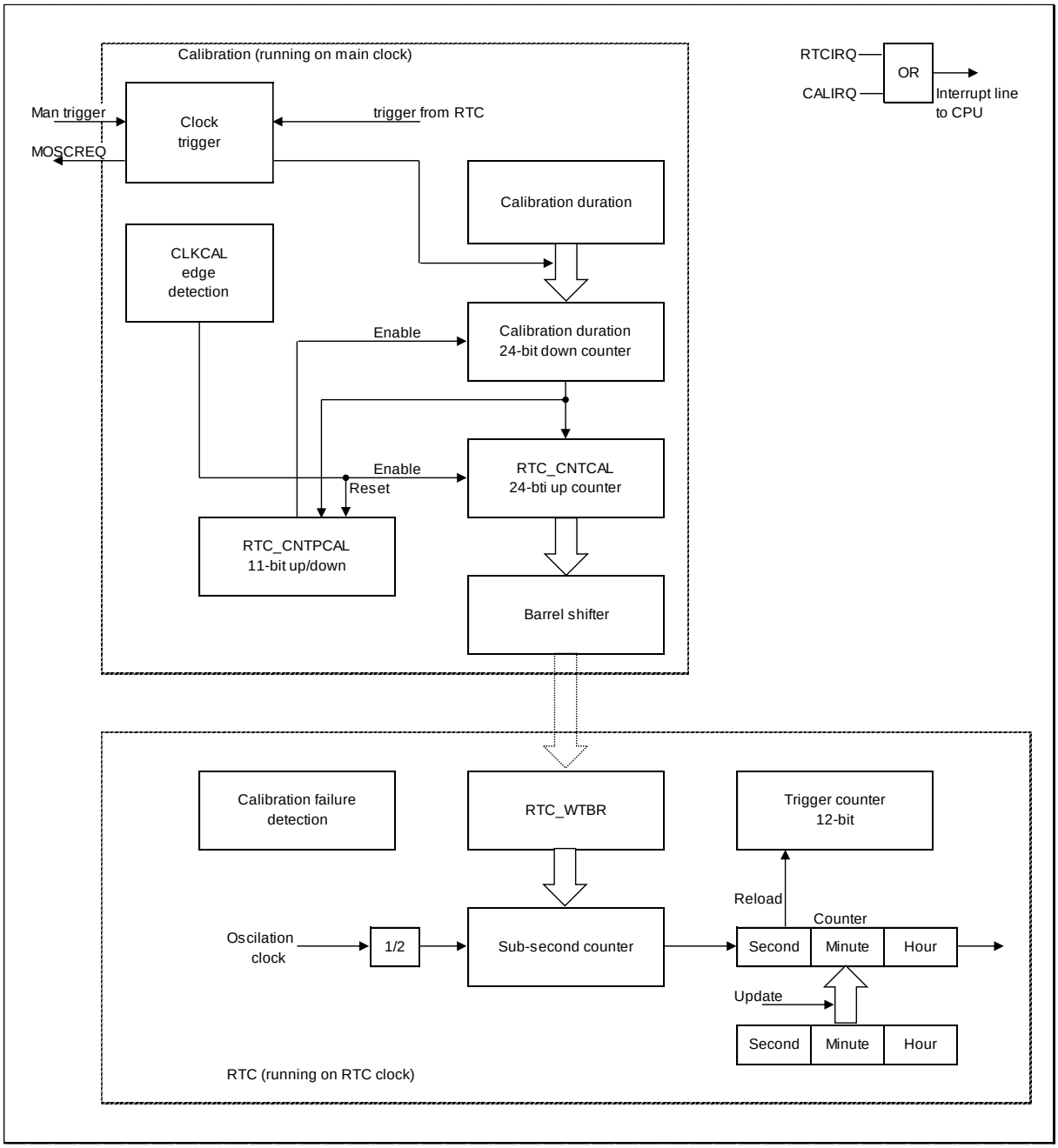
- The automatic cyclic calibration/measurement of Sub clock (CLK_SUB) or Slow RC clock (CLK_SRC) against Main clock (CLK_MAIN)
- The automatic correction of the RTC Sub-second counter value to keep accurate time
- Low-power modes due to Automatic wake-up/sleep of Main oscillation before/after Calibration 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

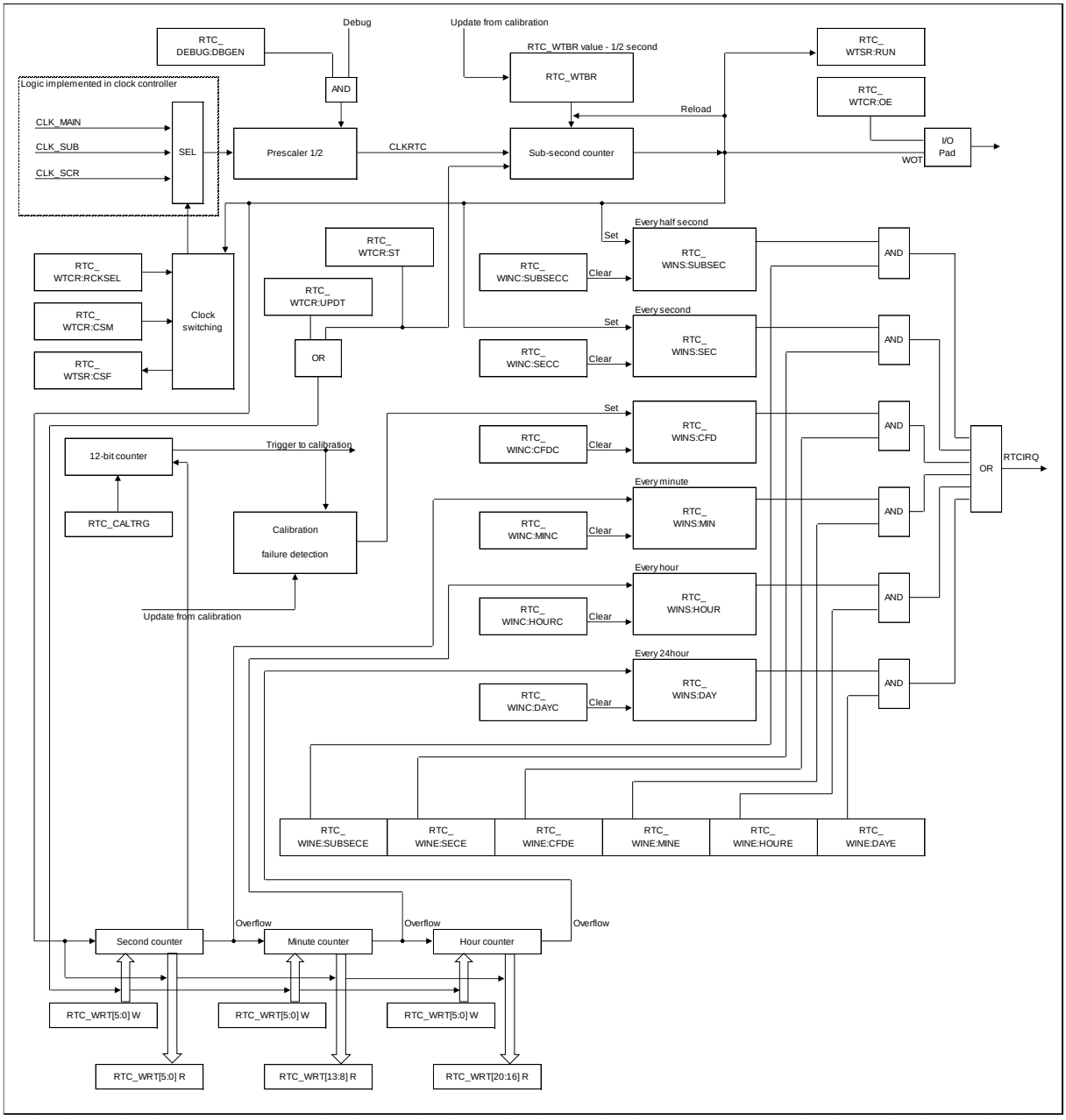
Block Diagram of Real Time Clock

Figure 2-1 Block Diagram of RTC



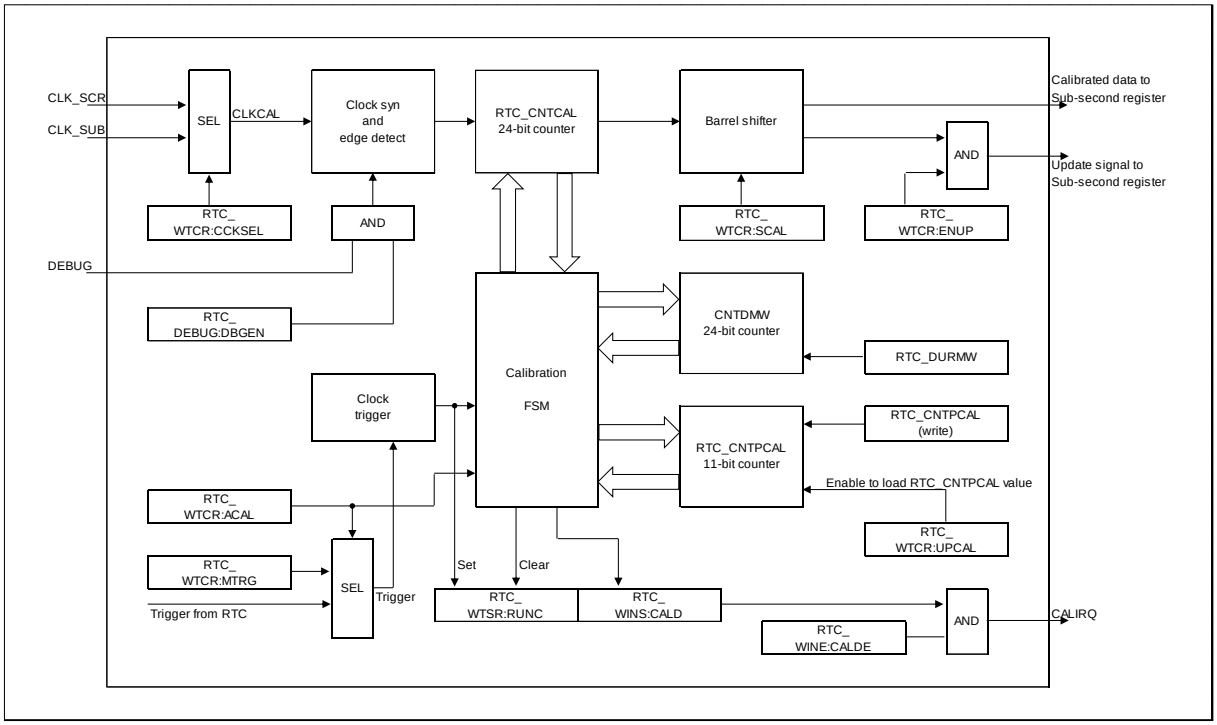
Block Diagram of RTC Configuration

Figure 2-2 RTC Timer Module Diagram



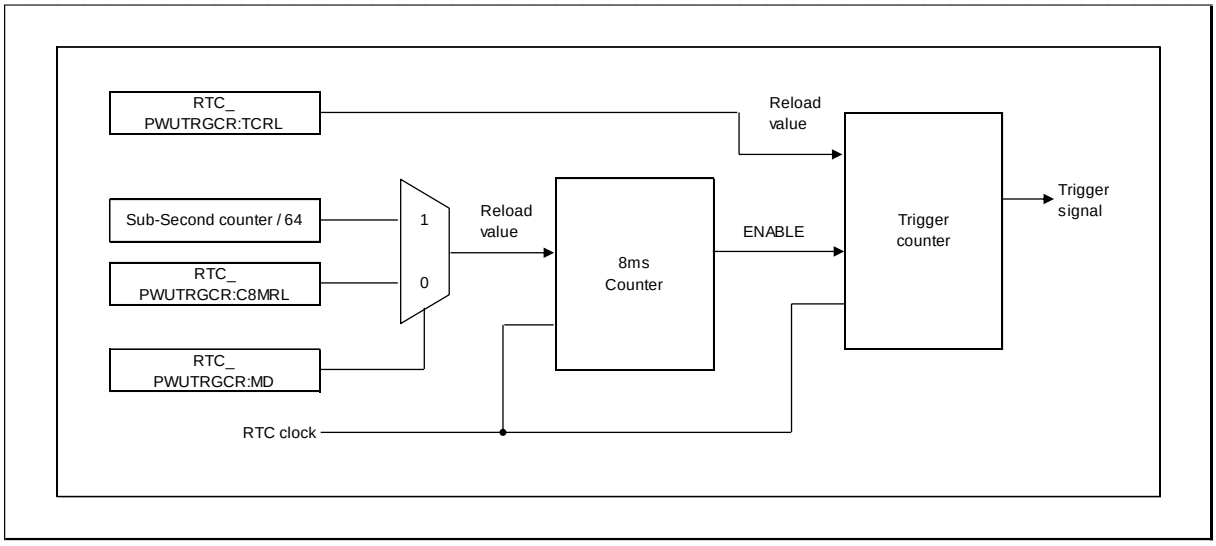
Calibration Configuration Diagram

Figure 2-3 Calibration Module Diagram



Block Diagram of Partial Wake Up Trigger

Figure 2-4 Partial Wake Up Trigger Module Diagram

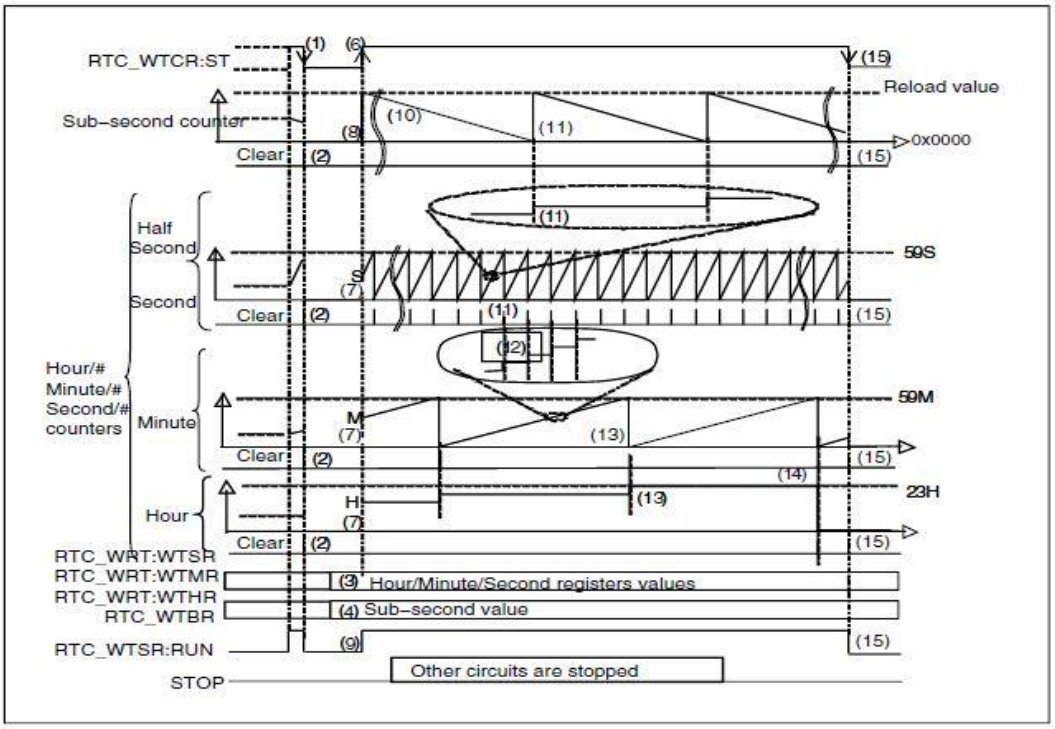


3. Operation of the Real Time Clock

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:

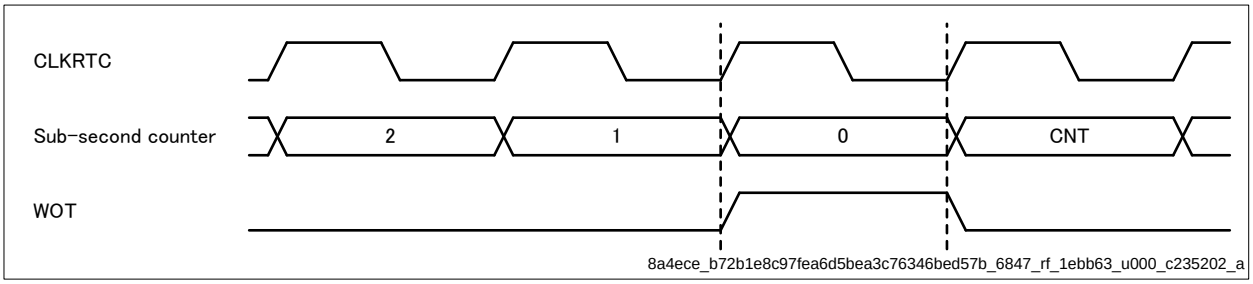
1. The software writes the start bit RTC_WTCR:ST to '1' and then '0'. (register initialization operation)
2. This resets the Sub-second counter to '0'. While RTC_WTCR:ST is '0' all counters for Subsecond, Second, Minute, and Hour are stopped.
3. The software writes Hour, Minute, and Second values to 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 RC 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.

Notes:

- *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:

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.

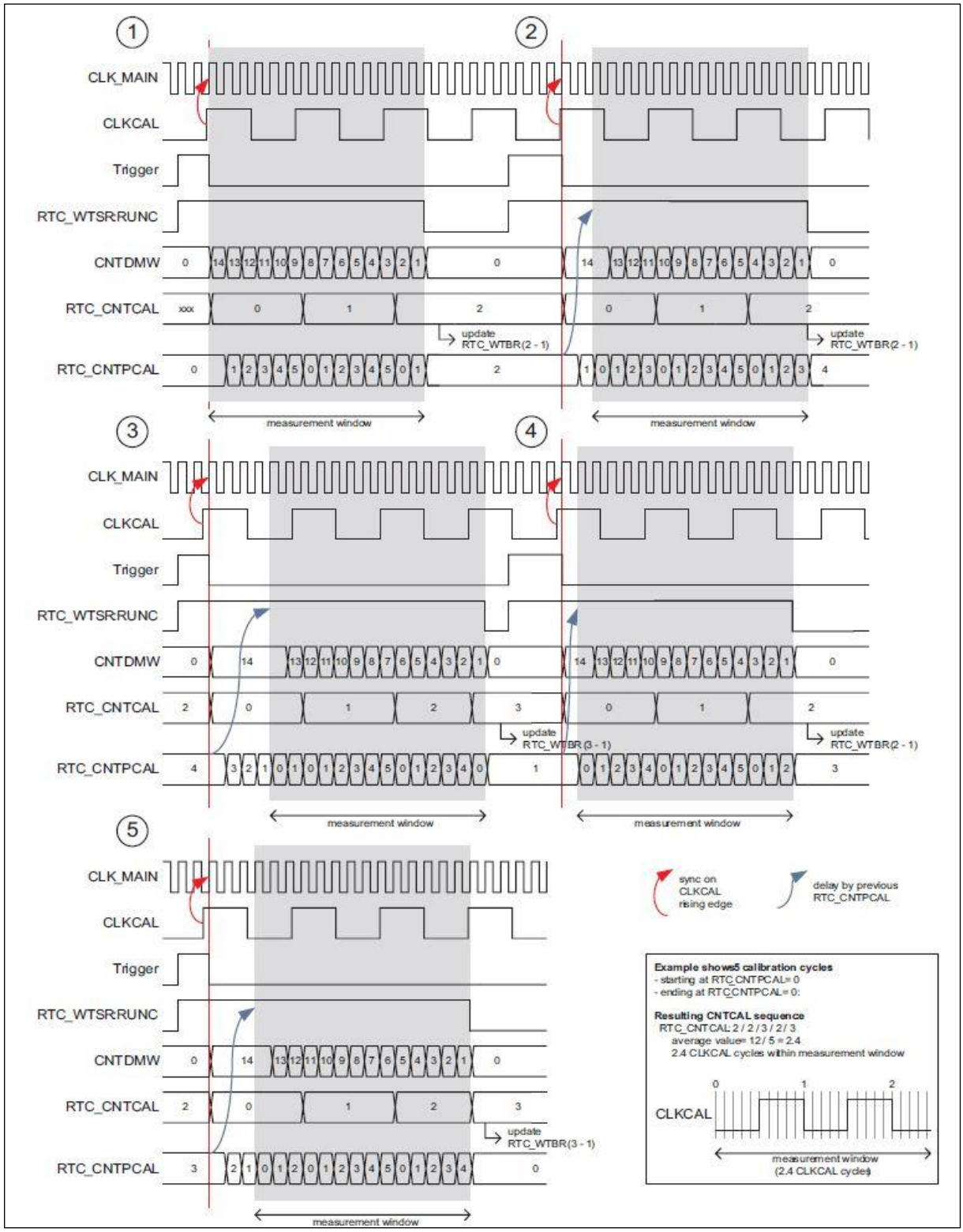
Manual Calibration:

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.

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.*

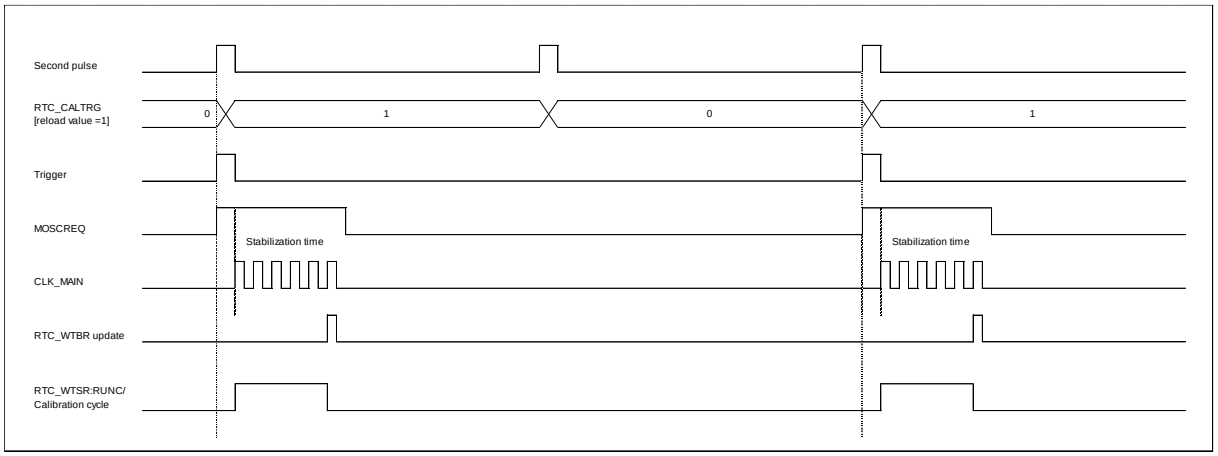
Figure 3-3 Calibration Operation



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_CALTRIG) 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

Notes:

- *All RTC registers and counters are initialized only by the power-on reset, low-voltage reset,*
- *external reset and Clock Supervisor (CSV) reset.*

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	R0, 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 by4
011	Multiply by8
100	Multiply by16
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`

[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 RC clock (CLKSRC) 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.

[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

Note:

- *For a detailed description of the Automatic RTC Calibration, refer to Section 3 'Operation of the Real Time Clock' and Section 5 'Cautions'.*

[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 RC 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 'Cautions'.

[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	0	00000					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 4-2 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 Subsecond counter requirements for 0.5 s.

Table 4-3 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 - 1

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'

- 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

- 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 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)

- 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-T00P00R01L03-E1-GBB00

1. Overview

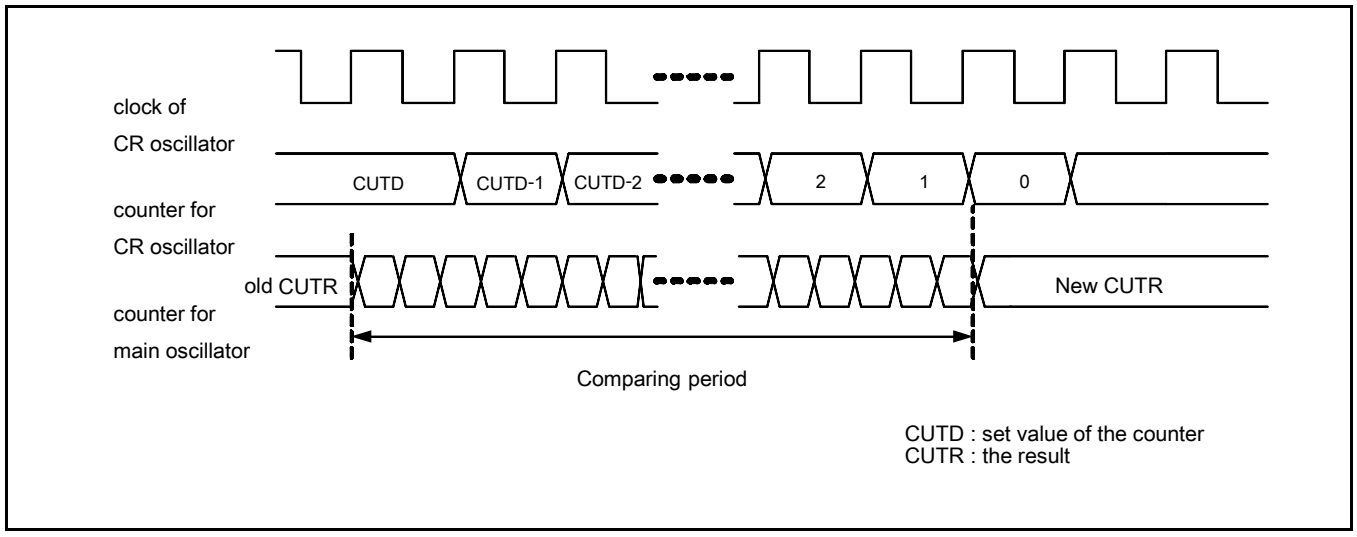
This section provides an overview of CR calibration.

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).

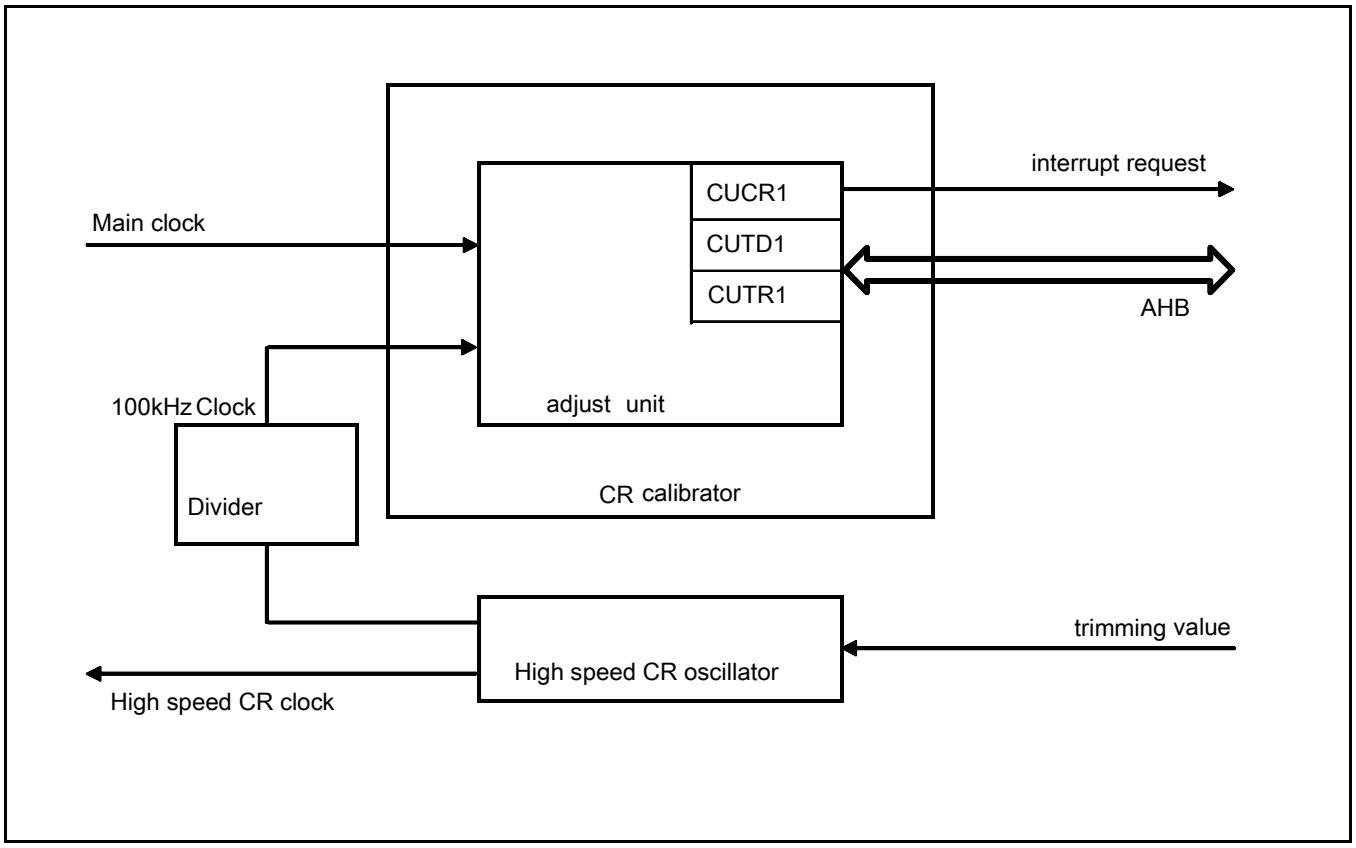
Figure 1-1 Comparison of Counters Driven by Each Clock



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 5'b11111 in the TRF bits.
2. Set 5'b00000 in the TRC bits to measure the frequency. Let the measured frequency be FC0.
3. Set 5'b11111 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)

6. Set the TRC code calculated in Step 1.
7. Set 5'b00000 in the TRF bits to measure the frequency. Let the measured frequency be FF0.
8. Set 5'b11111 in the TRF bits to measure the frequency. Let the measured frequency be FF31.
9. 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)
10. 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.
11. 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.
12. 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

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	0		0	0

[bit15:8] Reserved: Reserved Bits

[bit7] Reserved: Reserved Bit

This bit disables writing of "1". 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-T00P00R01L03-E1-GBB01

1. Overview

This section explains the features of the Backup RAM interface.

Target RAMs

- Backup RAM0: 4 KB
- Backup RAM1: 4 KB

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

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[3:0],{6{address[7:0]}} is written in 52 bits containing the ECC bit.)
- 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

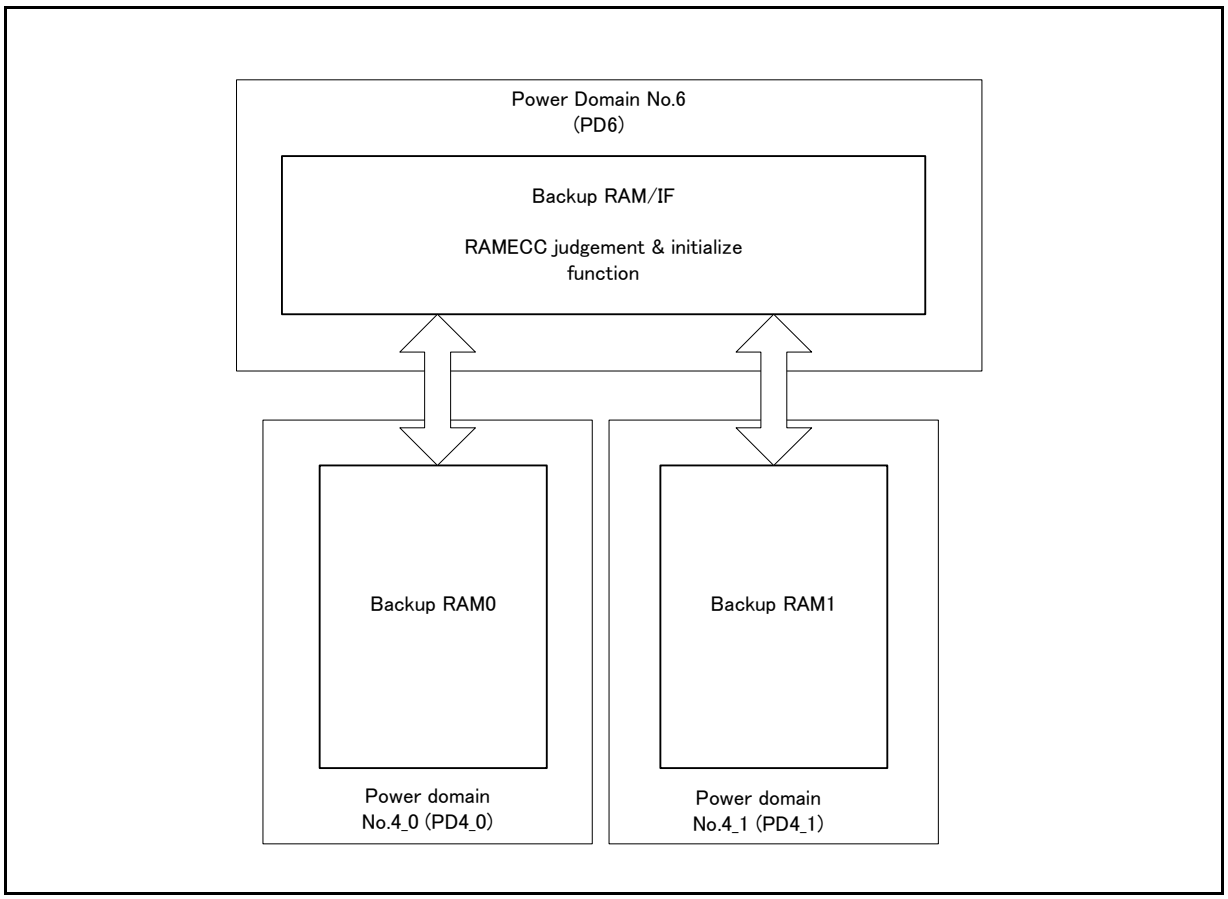


Figure 2-2 RAMECC Function Block Diagram

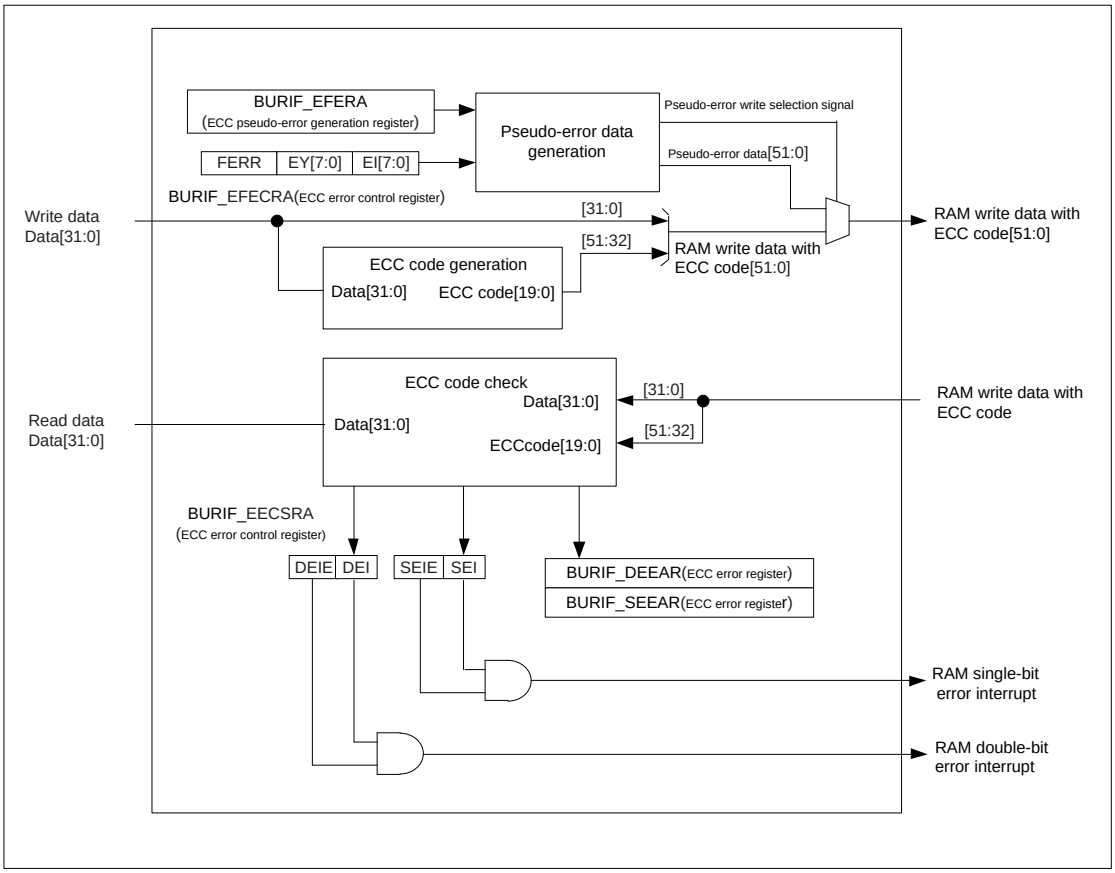
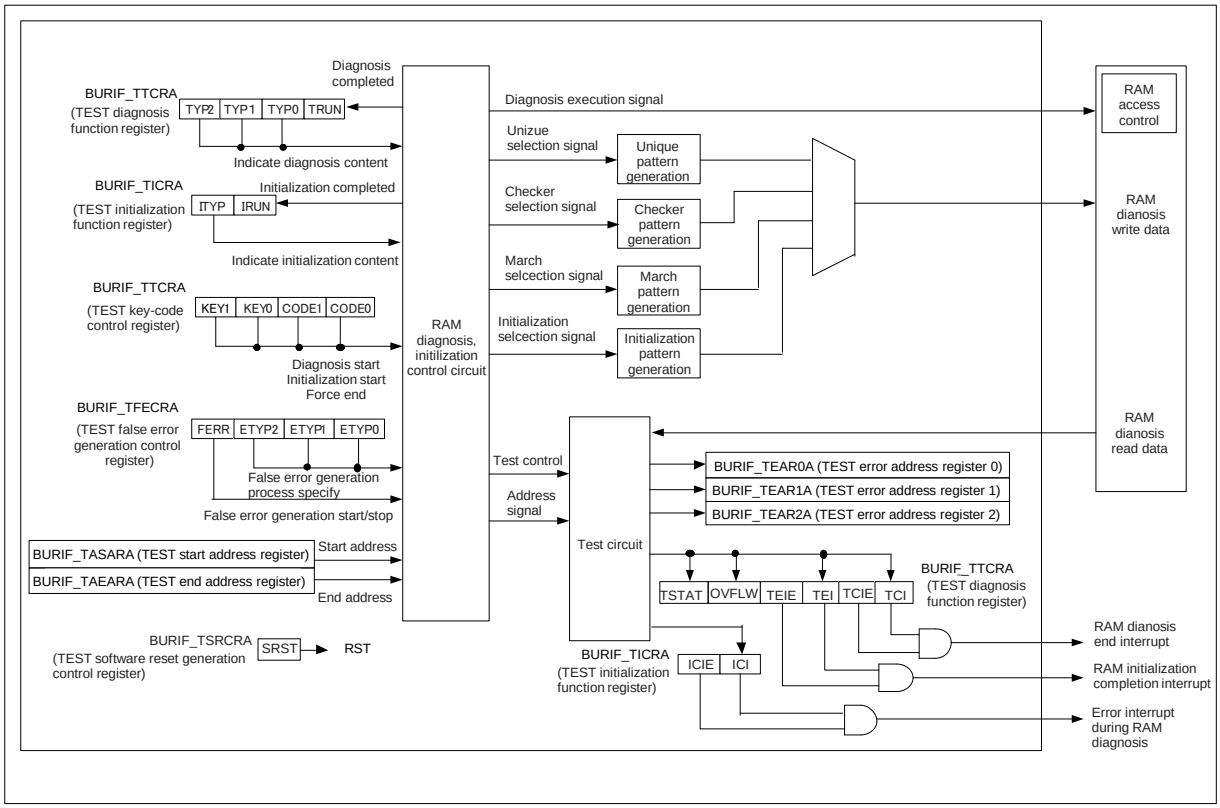


Figure 2-3 RAMECC Diagnosis Function Block Diagram



3. Explanation of Operation

This section explains the operation of the Backup RAM interface.

No Table of Contents Entries Found.

3.1. RAMECC Function

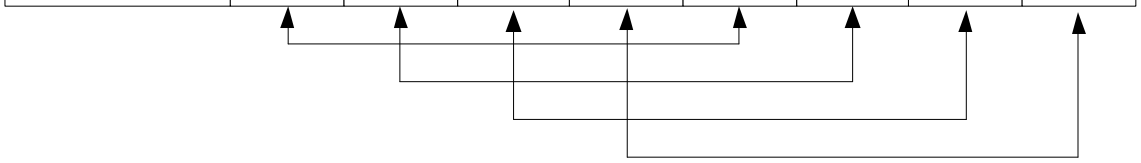
This section explains the RAMECC function.

Generation of ECC

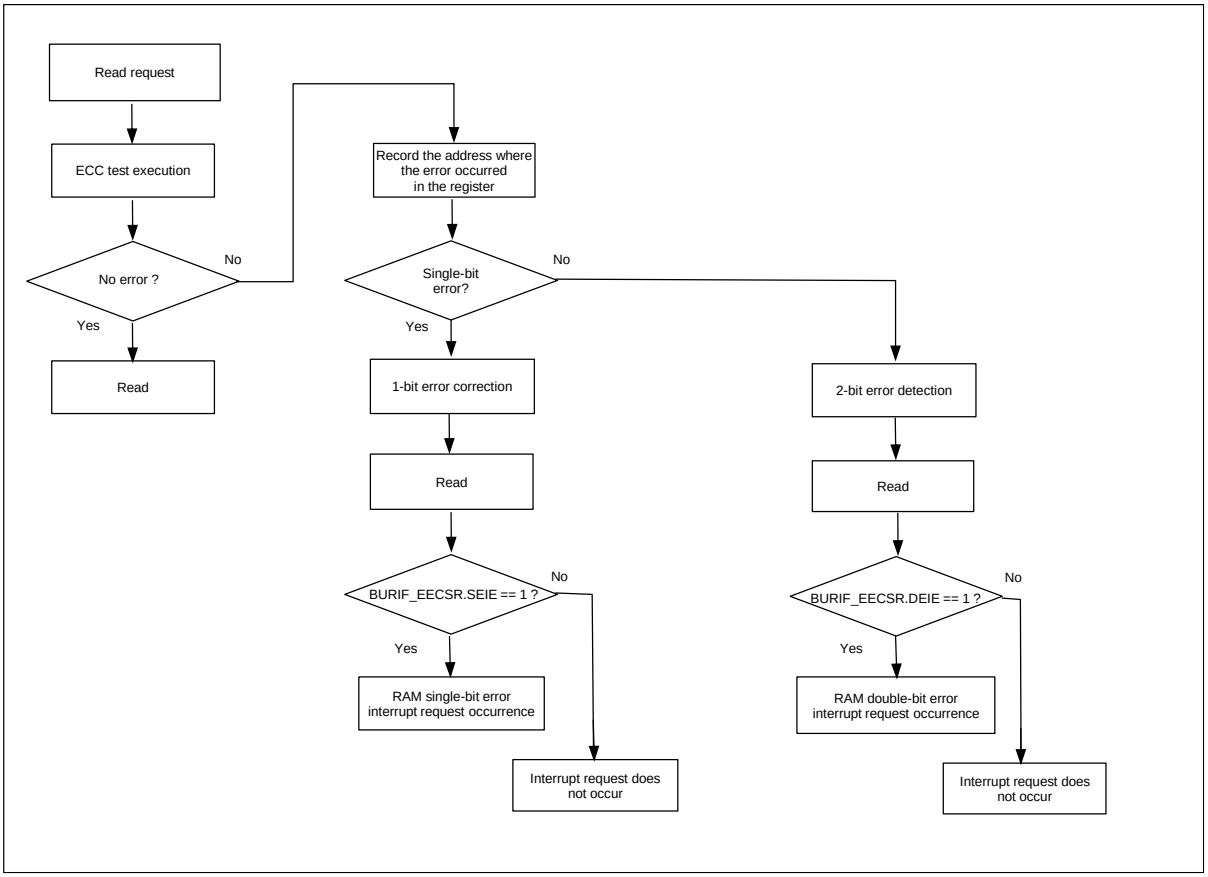
The RAMECC function is constantly running. (It does stop, however, when RAM diagnosis is being performed.) Upon the detection of a 1-bit error, the error occurrence address is held in the BURIF_SEEAR register. Upon the detection of a 2-bit error, the error occurrence address is held in the BURIF_DEEAR 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.


Read request
ECC test execution
No error ?
No
Yes
Read
Record the address where the error occurred in the register
Single-bit error?
No
Yes
1-bit error correction
Read
BURIF_EECSR.SEIE == 1 ?
No
Yes
RAM single-bit error interrupt request occurrence
Interrupt request does not occur
2-bit error detection
Read
BURIF_EECSR.DEIE == 1 ?
No
Yes
RAM double-bit error interrupt request occurrence
Interrupt request does not occur

Interrupt-Related Registers

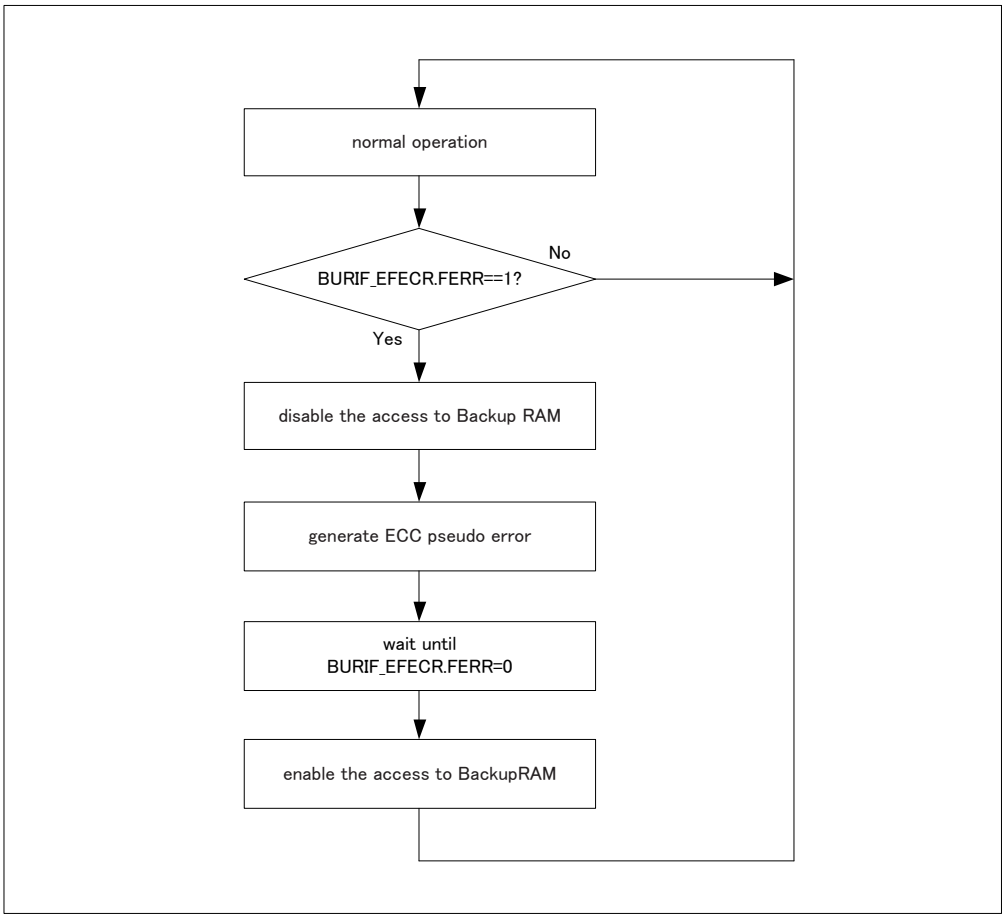
To generate an interrupt, write "1" to the interrupt generation enable bit (DEIE or SEIE) in the ECC error control register (BURIF_EECSR) in accordance with the purpose, and set the RAMECC interrupt vector. The interrupt request flag is not cleared automatically. For this reason, clear it by software 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.*

3.2. RAM Diagnosis

This section explains the RAM diagnosis.

RAM Diagnosis Procedure

RAM diagnosis consists of the following steps.

1. Unique data (write {address [3:0],{6{address[7:0]}} consisting of 52 bits containing the ECC bit.)
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 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.

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.

RAM Diagnosis (Uniqueness)

For a 1-word address:

- Write (1 cycle)
- Read 1 (1 cycle)
- Read 2 (1 cycle)

RAM Diagnosis (Checker)

For a 1-word address:

- Write 1 (1 cycle)
- Read 1 (1 cycle)
- 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.

RAM Diagnosis (March)

Each word address requires 3 writes and 2 reads.

- Write (1 x 3 cycles)
- Read (2 x 2 cycles)

RAM Initialization

For a 1-word address:

- Write (1 cycle)

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

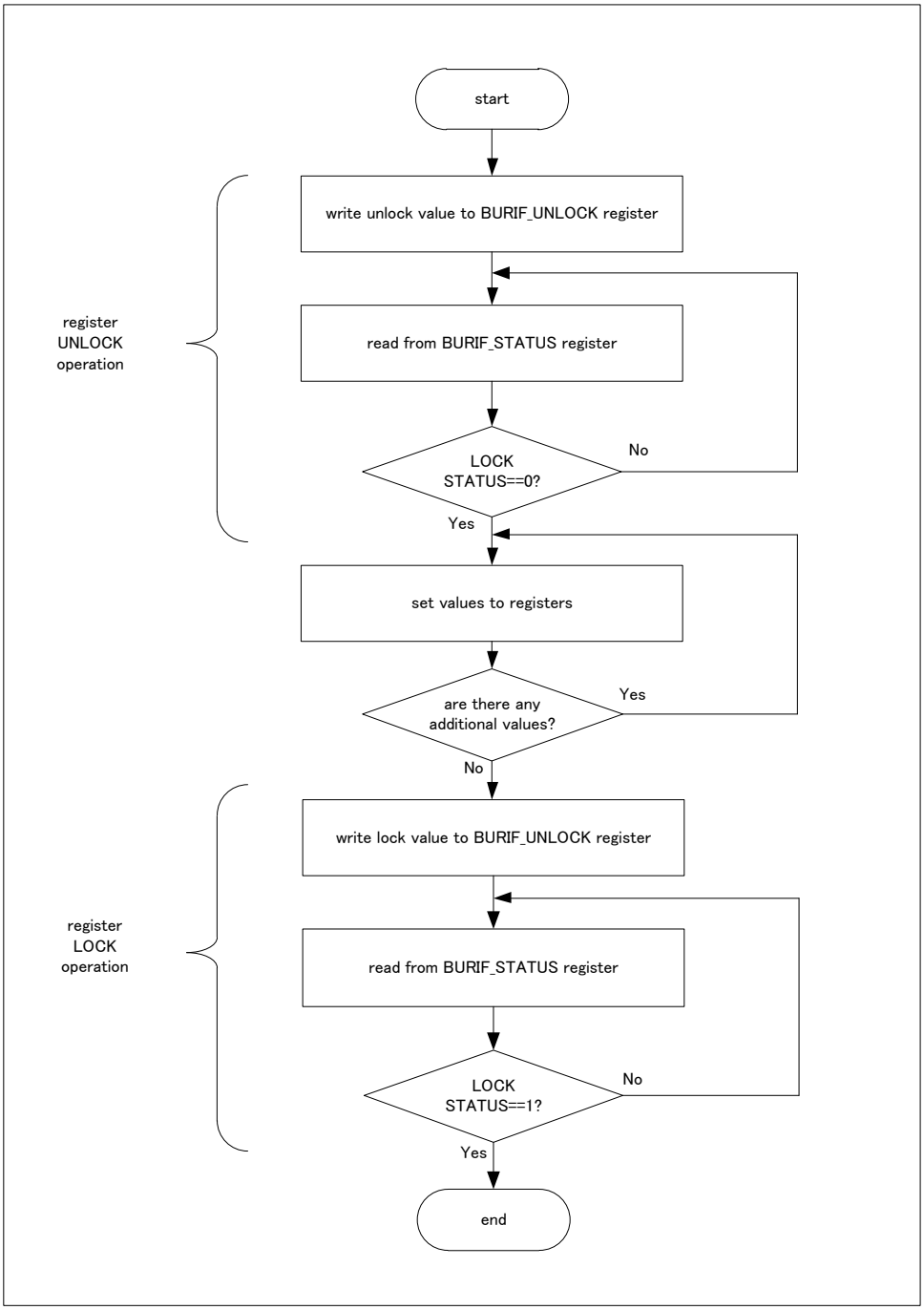


Figure 4-2 Setting Flow of RAM Self Diagnosis

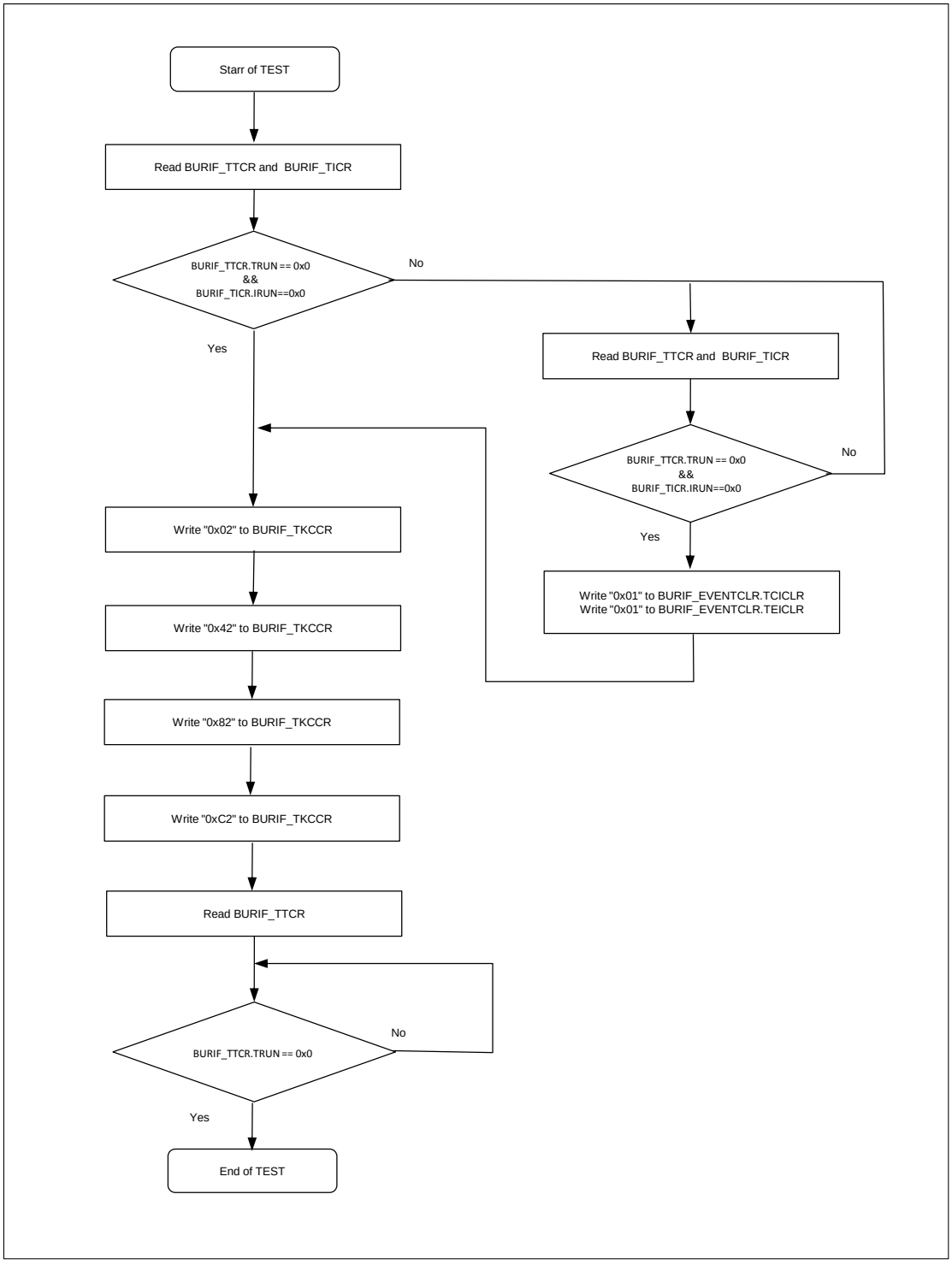
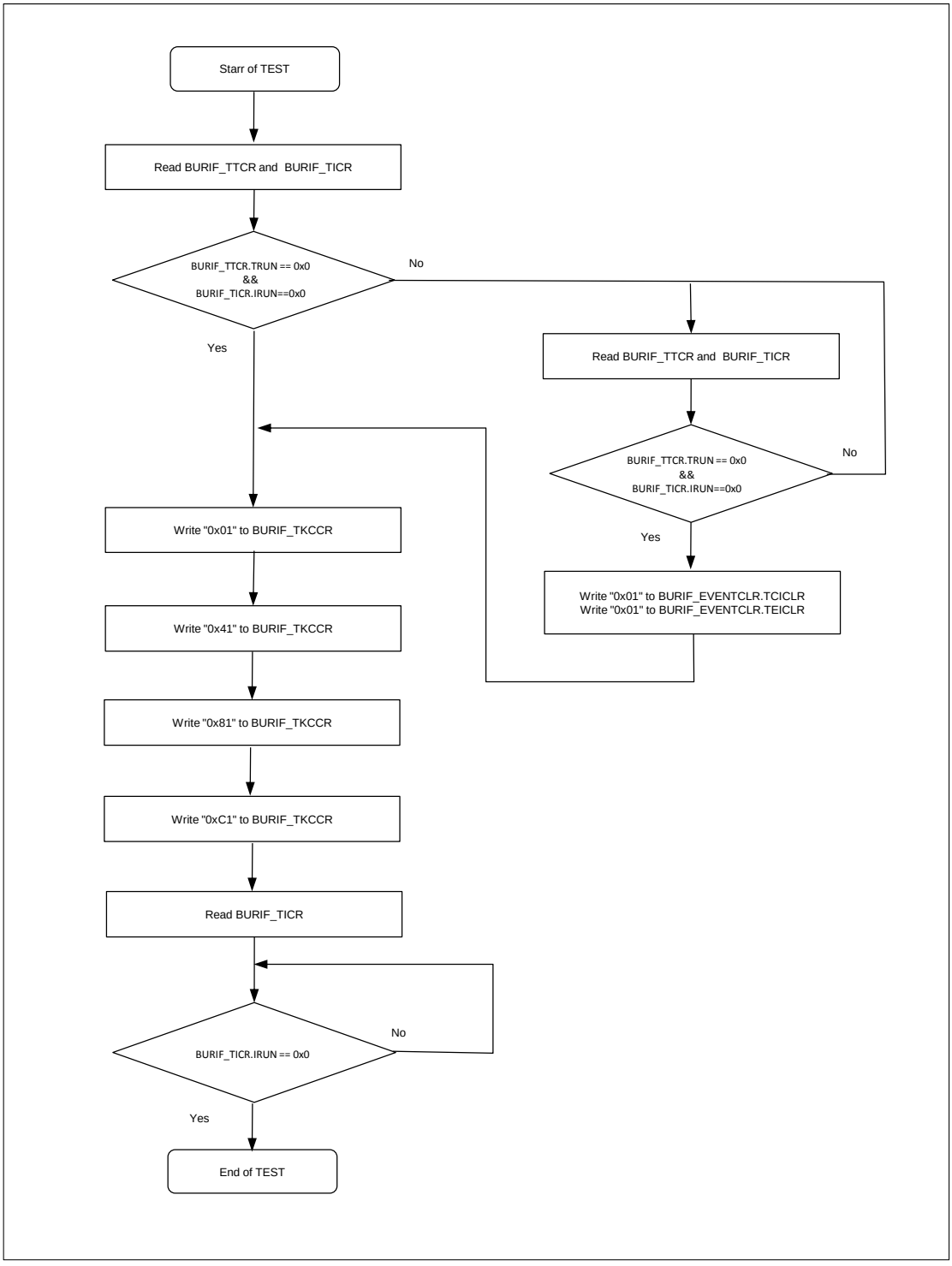


Figure 4-3 Setting Flow of RAM Initialization



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_TEAR0	BURAMIF TEST error address register 0	5.8
BURIF_TEAR1	BURAMIF TEST error address register 1	5.9
BURIF_TEAR2	BURAMIF TEST error address register 2	5.10
BURIF_TASAR	BURAMIF TEST start address register	5.11
BURIF_TAEAR	BURAMIF TEST end address register	5.12
BURIF_TTCR	BURAMIF TEST diagnosis function register	5.13
BURIF_TICR	BURAMIF TEST initialization function register	5.14
BURIF_TFECR	BURAMIF TEST pseudo-error generation control register	5.16
BURIF_TKCCR	BURAMIF TEST key code control register	5.17
BURIF_TSRCR	BURAMIF TEST software reset generation control register	5.15
BURIF_EVENTCLR	BURAMIF event clear register	5.18

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.

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							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial Value	00000000							

[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)

[bit7:0] Reserved: Reserved Bits

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'b00} to the base address.*
- *The effective address of ERR_ADDR changes according to RAM size.*
  - 8KB Effective address: ERR_ADDR[10:0]

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'b00} to the base address.*
- *The effective address of ERR_ADDR changes according to RAM size.*
  - 8KB Effective address: ERR_ADDR[10:0]

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'b00} to the base address.*
- *The effective address of ERR_ADDR changes according to RAM size.*
  - 8KB Effective address: ERR_ADDR[10:0]

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 (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".

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 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'b00} to the base address.*
- *The effective address of ERR_ADDR changes according to RAM size.*
  - 8KB Effective address: ERR_ADDR[10:0]

5.9. 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 "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'b00} to the base address.*
- *The effective address of ERR_ADDR changes according to RAM size.*
  - 8KB Effective address: ERR_ADDR[10:0]

5.10. 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'b00} to the base address.*
- *The effective address of ERR_ADDR changes according to RAM size.*
  - 8KB Effective address: ERR_ADDR[10:0]

5.11. 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'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.*
- *The effective address of SADDR changes according to RAM size.*
  - 8KB Effective address: SADDR[10:0]

5.12. 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'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.*
- *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.*
  - 8KB Effective address: EADDR[10:0]
- *EADDR initial Value changes according to RAM size.*
  - 00000111_11111111 8KB

5.13. 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.*

5.14. 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.

5.15. 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

5.16. 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.

5.17. 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 4-2 Setting Flow of RAM Self Diagnosis" and "Figure 4-3 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.*

5.18. 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-T00P00R01L03-E1-GBB00

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 127 pin
- Noise filter bypass possible
- Non-maskable interrupt supported
- DMA supports to 127 channels

2. Configuration

This section explains the configuration for external interrupts.

2.1. Block Diagrams

This section shows block diagrams of external interrupt circuits.

Figure 2-1 External Interrupt Block Diagram

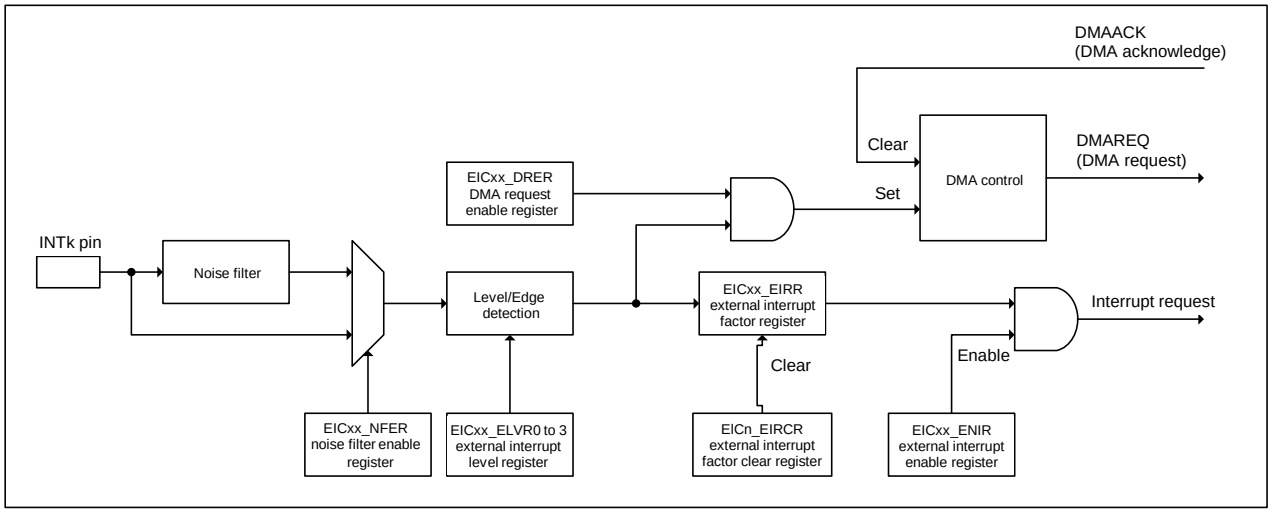
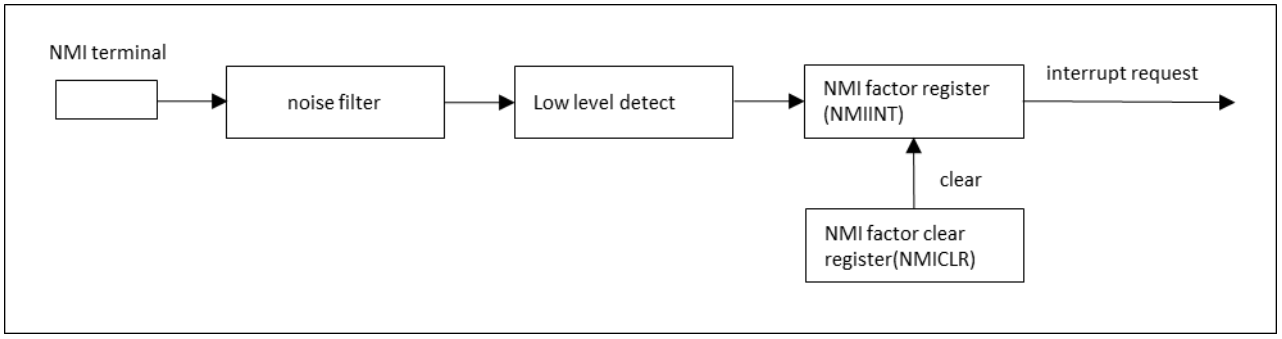


Figure 2-2 NMI (Non-Maskable Interrupt) Block Diagram



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 S6J3120 design 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

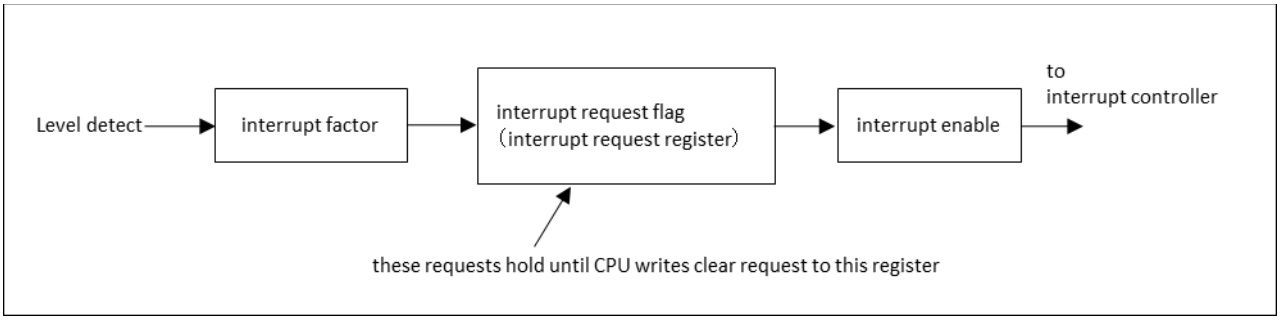


Figure 3-2 Interrupt Factor and Interrupt Request to Interrupt Controller while Interrupts Are Enabled

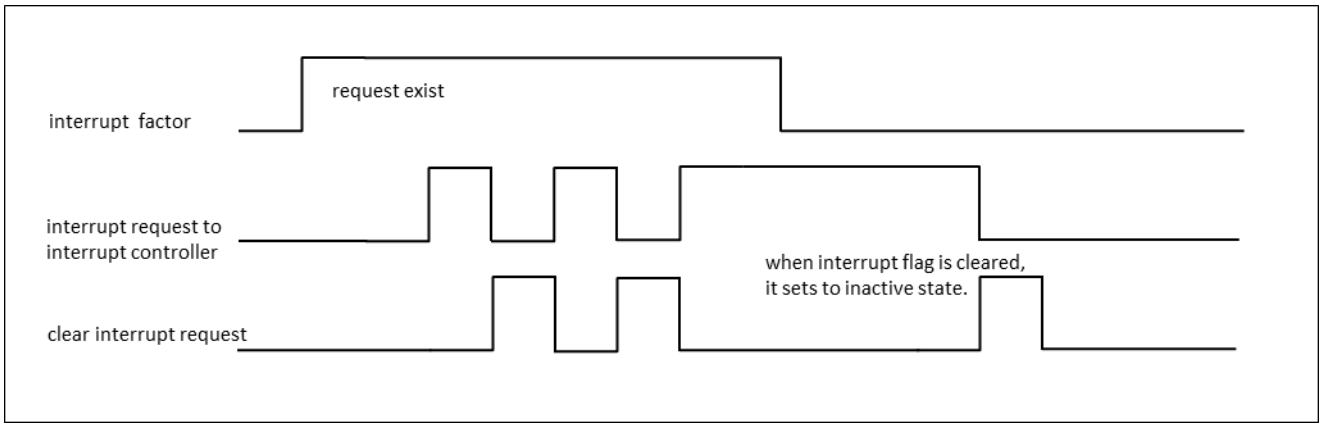
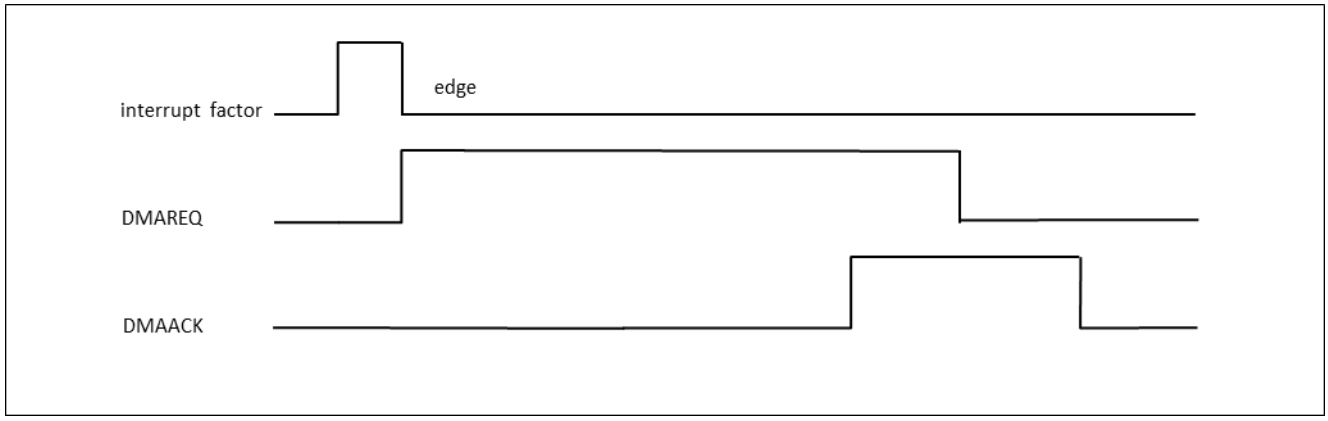


Figure 3-3 Interrupt Factor and DMAREQ to DMA while DMA Is Enabled



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

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
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-T00P00R01L03-E1-GBB00

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 "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.")
- The watchdog counter can be stopped in the debugging state of the processor.

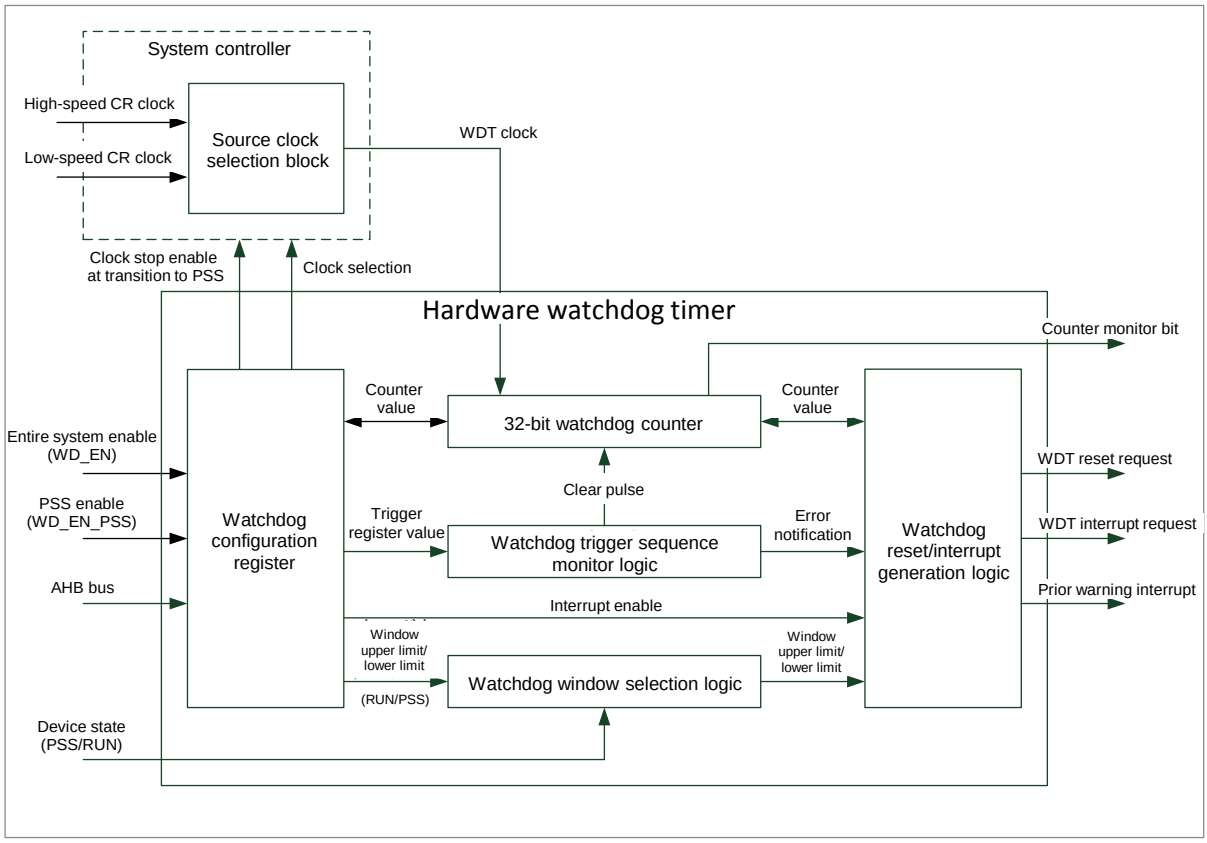
Notes:

- 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).

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 .")

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, 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.)*

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).

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	3 types - High-speed CR clock - Low-speed CR 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)

Item	Software Watchdog	Hardware Watchdog
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	Unsupported	Unsupported
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	Stop only in PSS (Fixed hardware)

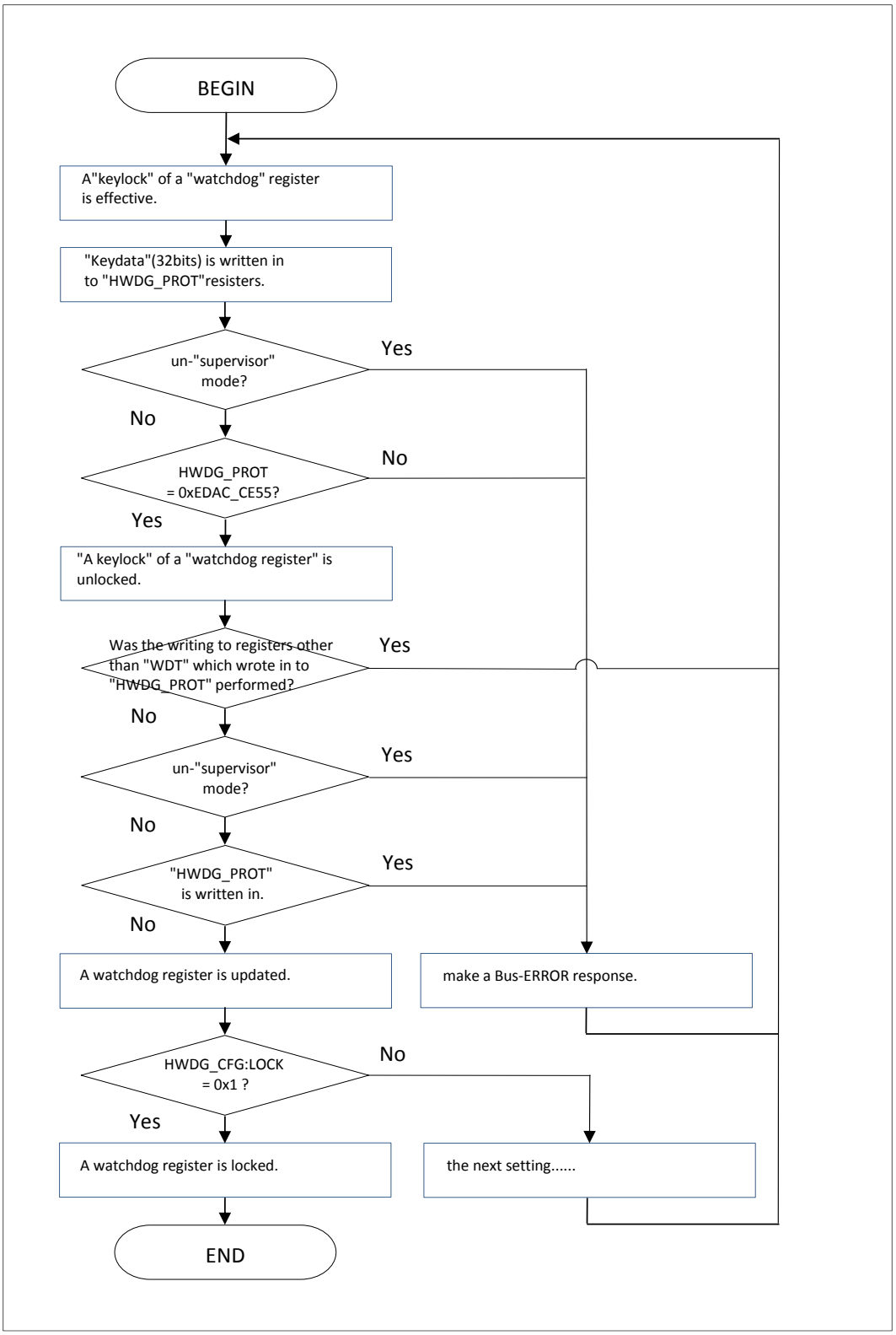
Notes:

- *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).*
  - HWDT_INT
  - HWDT_TRG0CFG
  - HWDT_TRG1CFG
  - HWDT_RUNLLS
  - HWDT_RUNULS
  - HWDT_PSSLLS
  - HWDT_PSSULS
  - HWDT_RSTDLY
  - HWDT_CFG

For details on the registers, see "6.Register List"

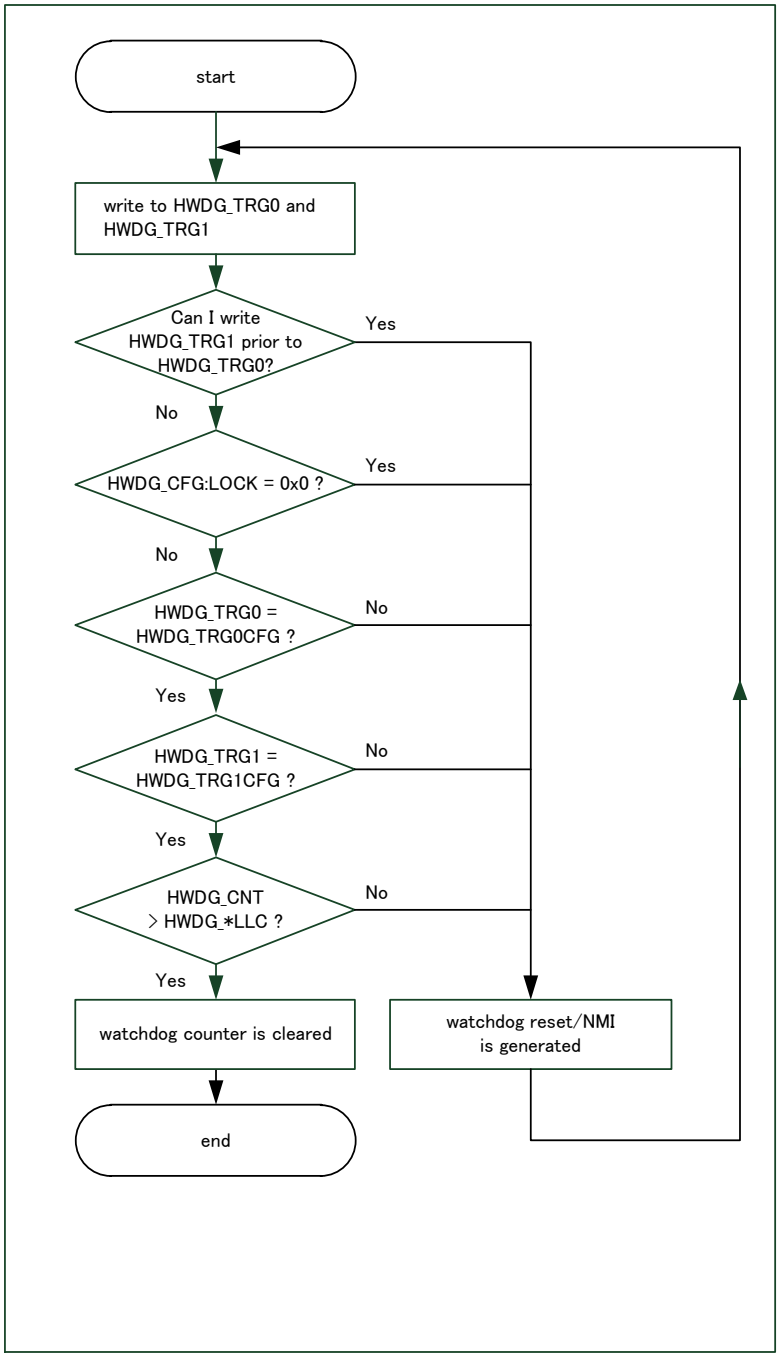
The following figure shows the watchdog register write protection sequence setting procedure.

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

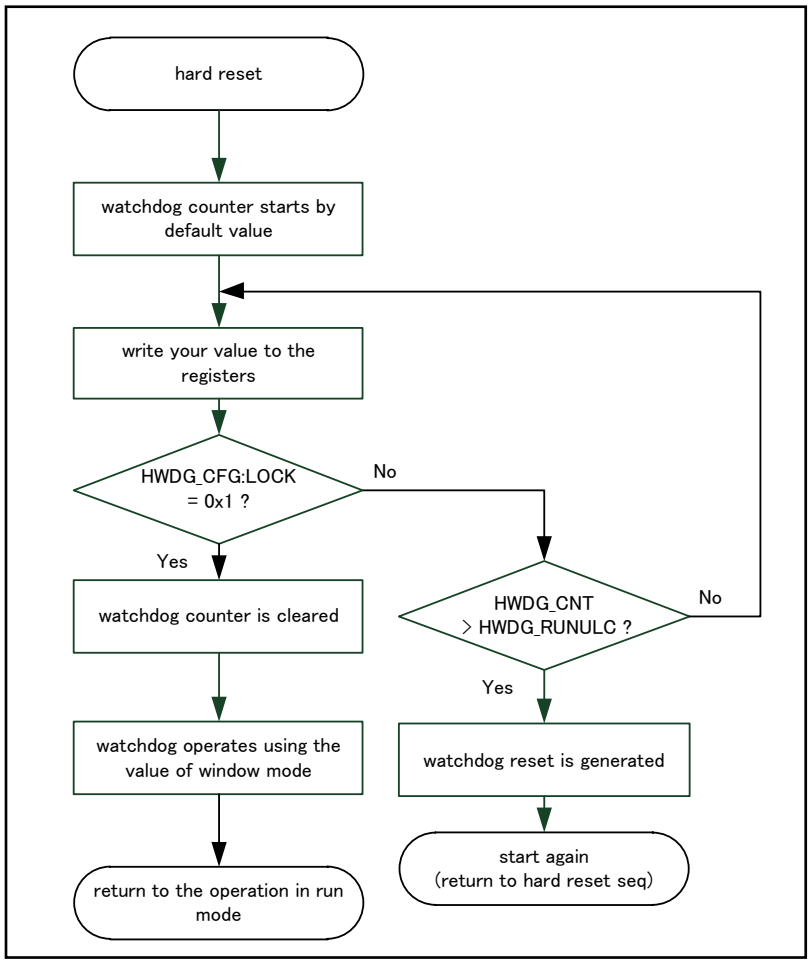


Note:

- *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



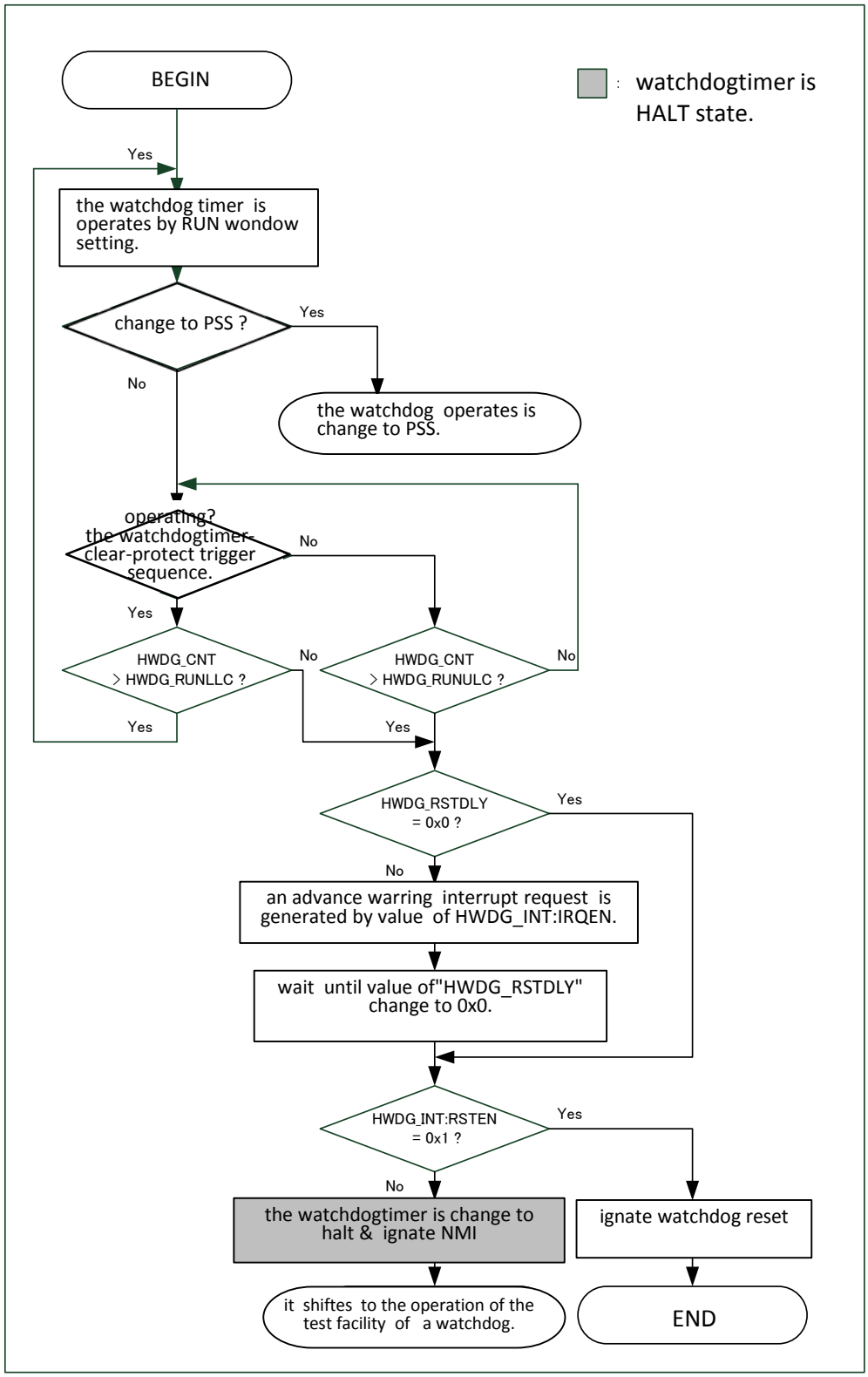
1. The hard reset initializes the hardware watchdog timer. Then, immediately after the hard reset is cleared, the operating state is indicated.
2. The default window setting indicates the window lower limit value (= 0x0000_0000) and window upper limit value (= 0x0100_0000).
3. You can change the register 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.

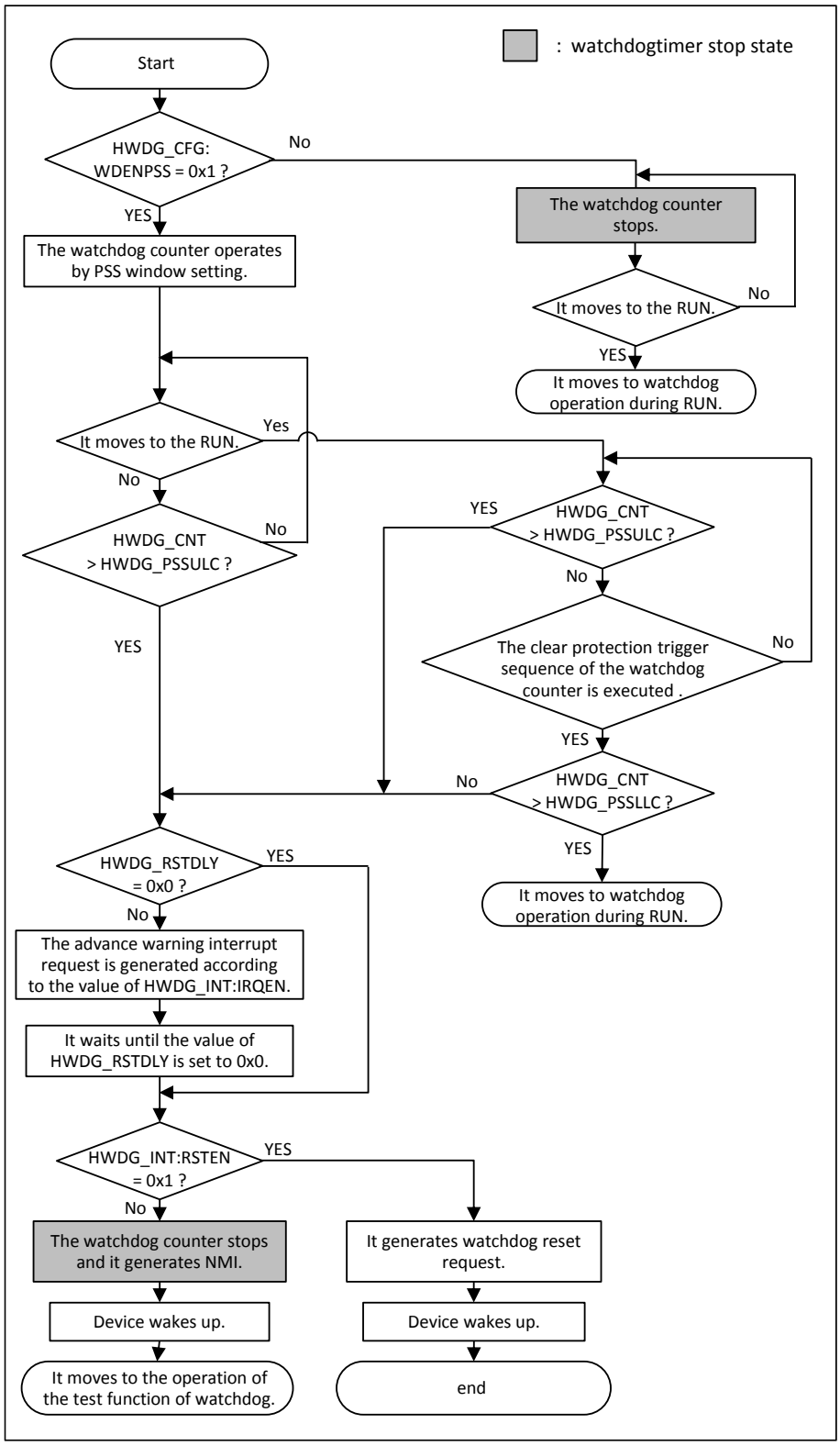
Figure 3-4 Hardware Watchdog Operation in RUN



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, seeThe following figure shows the Hardware watchdog operation in PSS.
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 3-5.

The following figure shows the Hardware watchdog operation in PSS.

Figure 3-5 Hardware Watchdog Operation in PSS



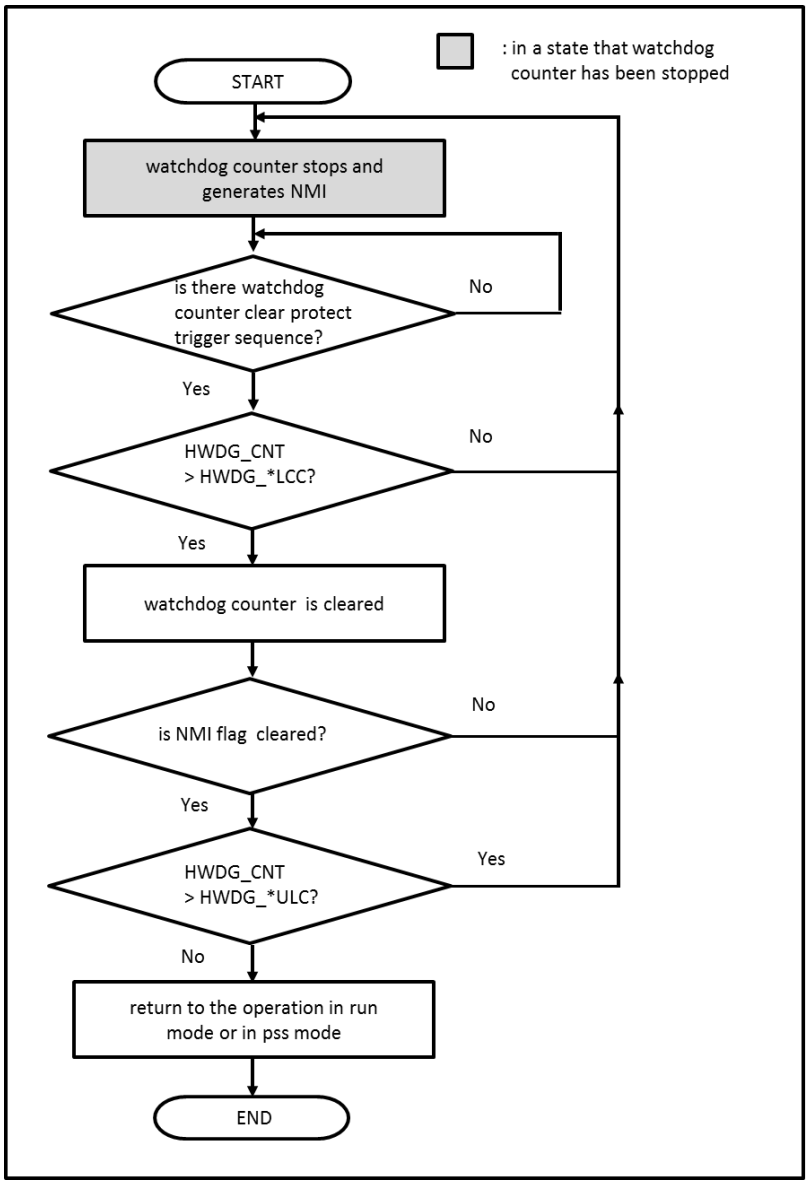
1. The watchdog counter in PSS operates or stops according to the value of the WDENPSS bit in the Hardware watchdog configuration register (HWDG_CFG).
2. When the device state transitions from PSS to RUN, the watchdog counter operation is started immediately. However, the PSS window setting continues 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.

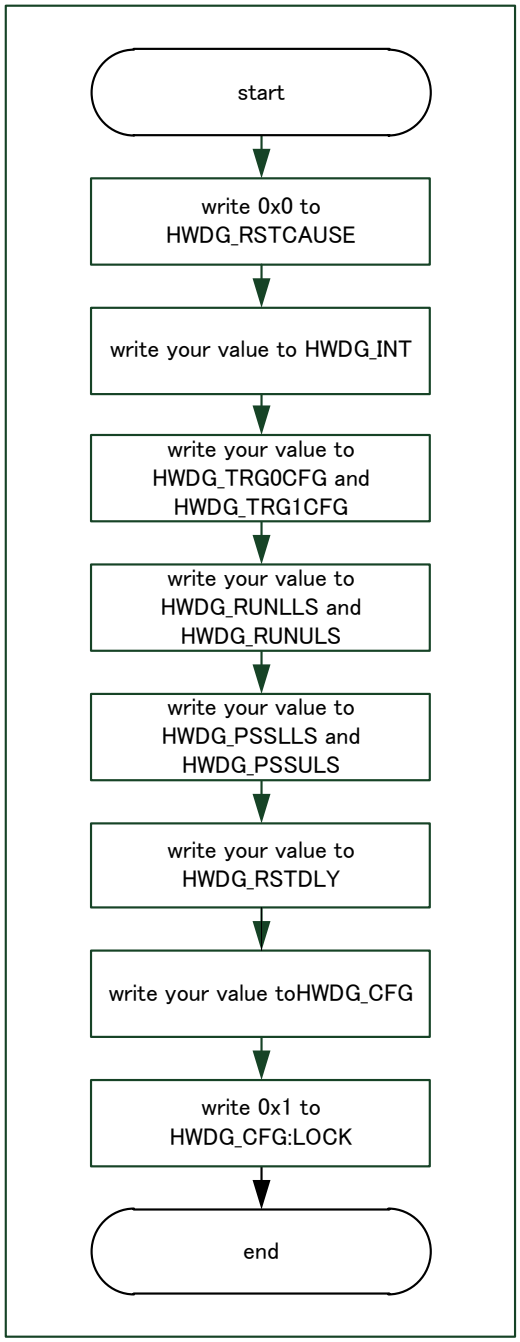
Note:

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



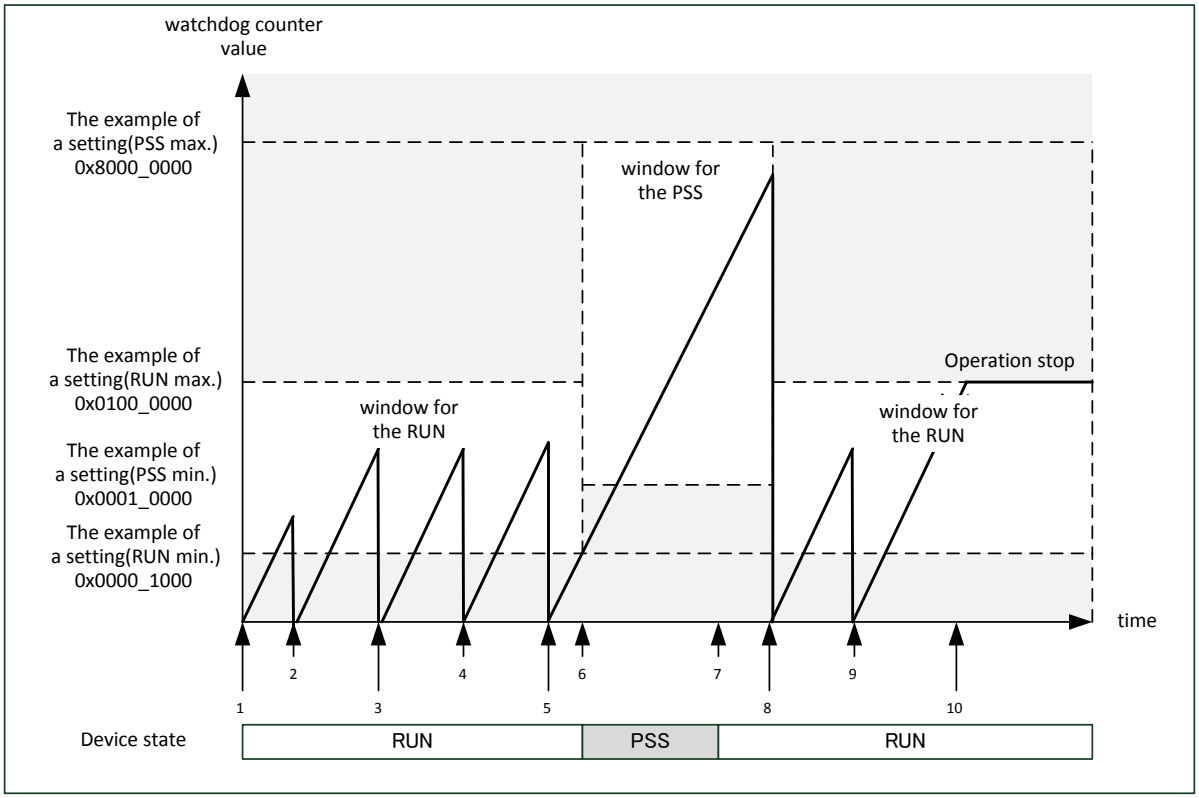
Notes:

- *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)

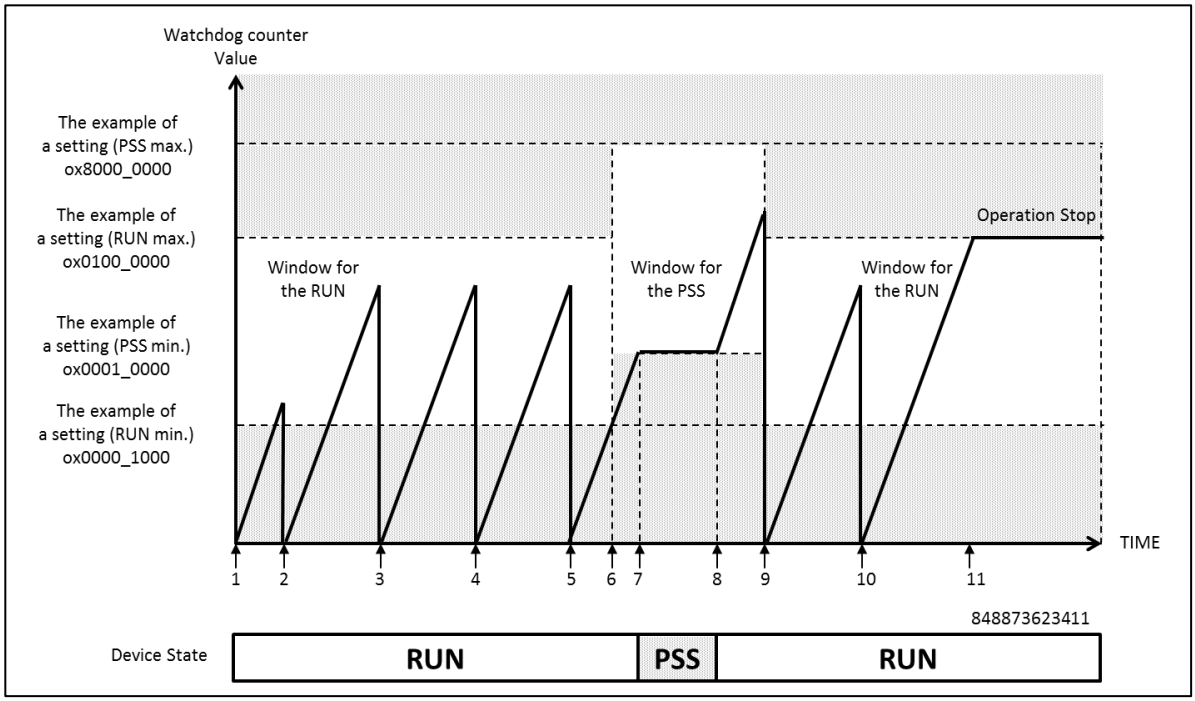


1. After power-on, the hardware watchdog timer starts operation. (The initial value of the window upper limit is 0x01000000.)
2. After writing setting values in the registers, the LOCK bit in the hardware watchdog configuration register (HWDG_CFG) is set to"1". At this time, the watchdog counter is automatically 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 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)
- *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

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.

Register Configurations

Bit	31	0
Field	KEY[31:0]	
R/W Attribute	R, W	
Protection Attribute	WP	
Initial Value	00000000_00000000_00000000_00000000	

Register Functions

[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.*

6.2. Hardware Watchdog Counter Register (HWDG_CNT)

This register indicates the current up count value of the watchdog counter.

Register Configurations

Bit	31	0
Field	WDGCNT[31:0]	
R/W Attribute	R, WX	
Protection Attribute	-	
Initial Value	00000000_00000000_00000000_00000000	

Register Functions

[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. To check the factor of the watchdog reset request, read this register.

Register Configurations

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		RSTCAUSE4	RSTCAUSE3	RSTCAUSE2	RSTCAUSE1	RSTCAUSE0
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

Register Functions

[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.

Register Configurations

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							

Register Functions

[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.

Register Configurations

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							

Register Functions

[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 interrupt request (NMI) and prior warning interrupt request. This register also includes interrupt status flags.

Register Configurations

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

Register Functions

[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 interrupt 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 ."*

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.

Register Configurations

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

Register Functions

[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.

Register Configurations

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							

Register Functions

[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.

Register Configurations

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							

Register Functions

[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.

Register Configurations

Bit	31	0
Field	WDGRUNLLS[31:0]	
R/W Attribute	R/W	
Protection Attribute	WPS	
Initial Value	00000000_00000000_00000000_00000000	

Register Functions

[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.

Register Configurations

Bit	31	0
Field	WDGRUNULS[31:0]	
R/W Attribute	R/W	
Protection Attribute	WPS	
Initial Value	00000001_00000000_00000000_00000000	

Register Functions

[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 lower limit value actually used is the value in the Hardware watchdog lower limit RUN current register (HWDG_RUNLLC).*

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.

Register Configurations

Bit	31	0
Field	WDGPSSLLS[31:0]	
R/W Attribute	R/W	
Protection Attribute	WPS	
Initial Value	00000000_00000000_00000000_00000000	

Register Functions

[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.

Register Configurations

Bit	31	0
Field	WDGPSSULS[31:0]	
R/W Attribute	R/W	
Protection Attribute	WPS	
Initial Value	10000000_00000000_00000000_00000000	

Register Functions

[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).

Register Configurations

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	

Register Functions

[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.

Register Configurations

Bit	bit31	bit25	bit24
Field	Reserved		LOCK
R/W Attribute	R0, WX		R, W
Protection Attribute	WPS		
Initial Value	0000000		1

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

Register Functions

[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.

LOCK	Description
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 "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.")

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, "1" 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.

Register Configurations

Bit	bit31	bit0
Field	WDGRUNLLC[31:0]	
R/W Attribute	R, WX	
Protection Attribute	-	
Initial Value	00000000_00000000_00000000_00000000	

Register Functions

[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.

Register Configurations

Bit	bit31	bit0
Field	WDGRUNULC[31:0]	
R/W Attribute	R, WX	
Protection Attribute	-	
Initial Value	00000001_00000000_00000000_00000000	

Register Functions

[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.

Register Configurations

Bit	bit31	bit0
Field	WDGPSSLLC[31:0]	
R/W Attribute	R, WX	
Protection Attribute	-	
Initial Value	00000000_00000000_00000000_00000000	

Register Functions

[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.

Register Configurations

Bit	bit31	bit0
Field	WDGPSSULC[31:0]	
R/W Attribute	R, WX	
Protection Attribute	-	
Initial Value	10000000_00000000_00000000_00000000	

Register Functions

[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 (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).

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 products stop the Hardware watchdog timer operation in PSS by the fixed hardware.

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-T00P00R01L03-E1-GBB01

1. Overview

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.

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 3 types. (*a)
  (High-speed CR clock (default), low-speed CR 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).
- The watchdog counter can be stopped in the debugging state of the processor.

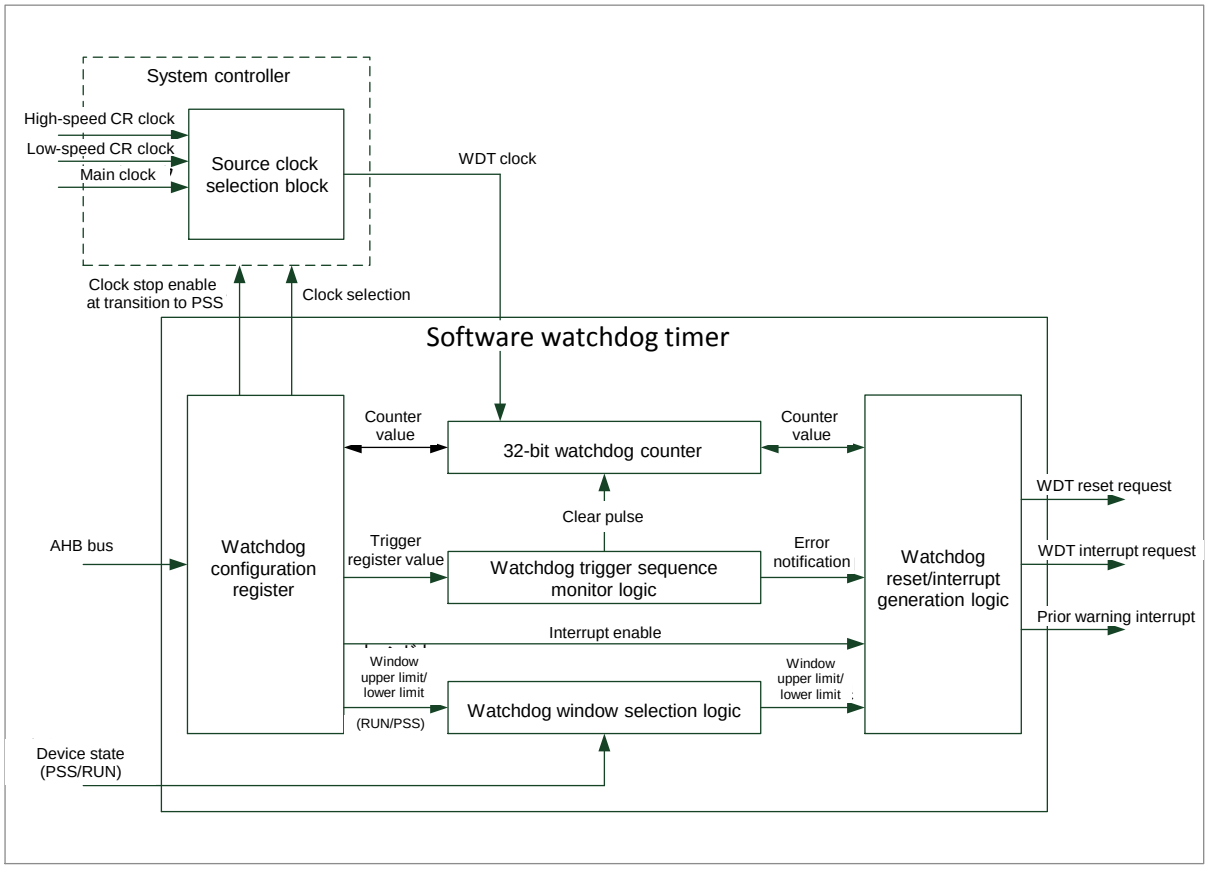
Notes:

- 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.

- 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 (SWDGn_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 (3 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 3 types of source clocks. Specifically, select a source clock from the high-speed CR clock, low-speed CR 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 .")

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 (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)

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 Hardware 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

For this product, output of the watchdog counter monitor bit of MCU output pin is unsupported by the Software watchdog timer.

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	3 types - High-speed CR clock - Low-speed CR 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	Unsupported	Unsupported
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

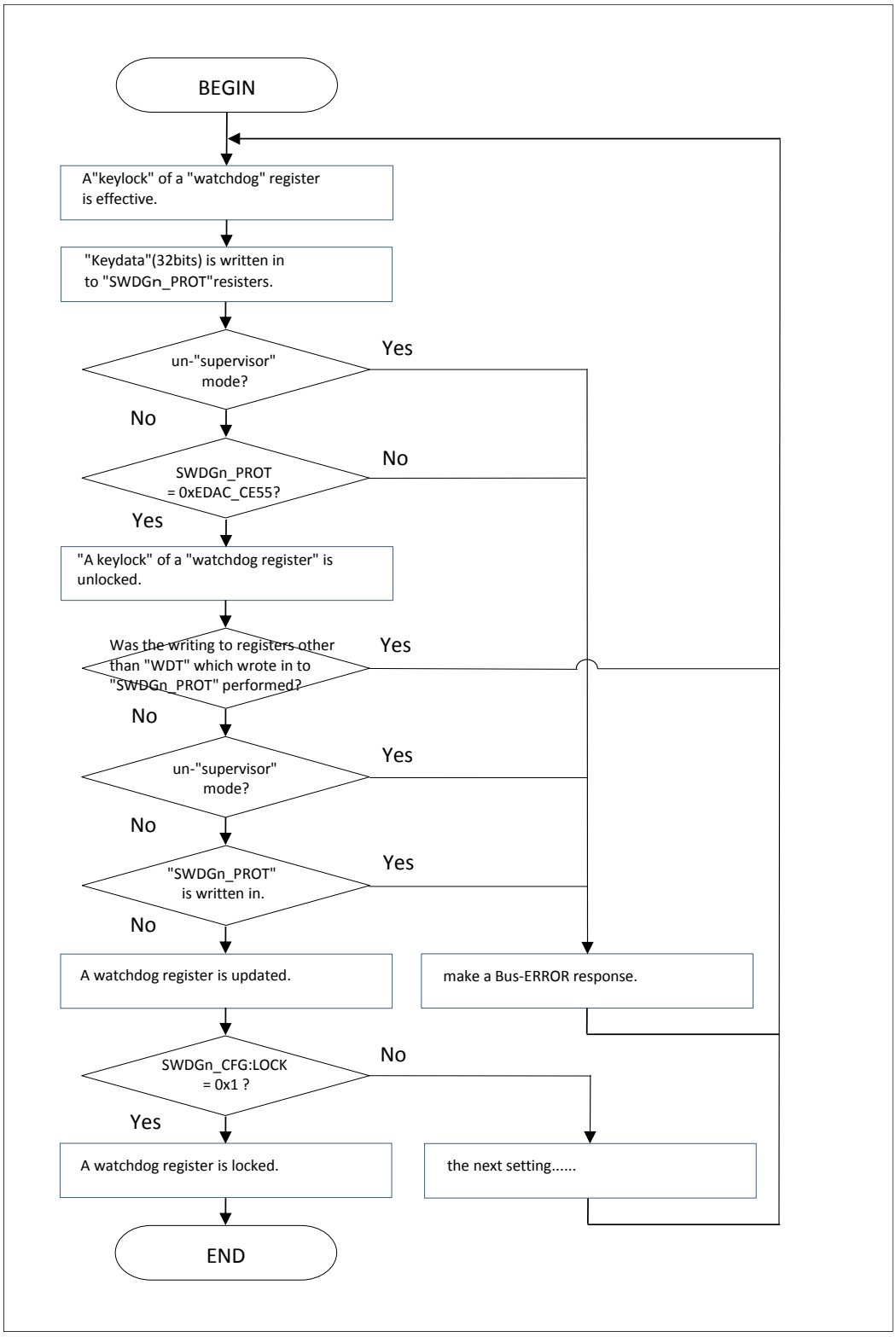
Item	Software Watchdog	Hardware Watchdog
Watchdog Counter Enable Bit Control	Register write by a user program	Stop only in PSS (Fixed hardware)

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.)*

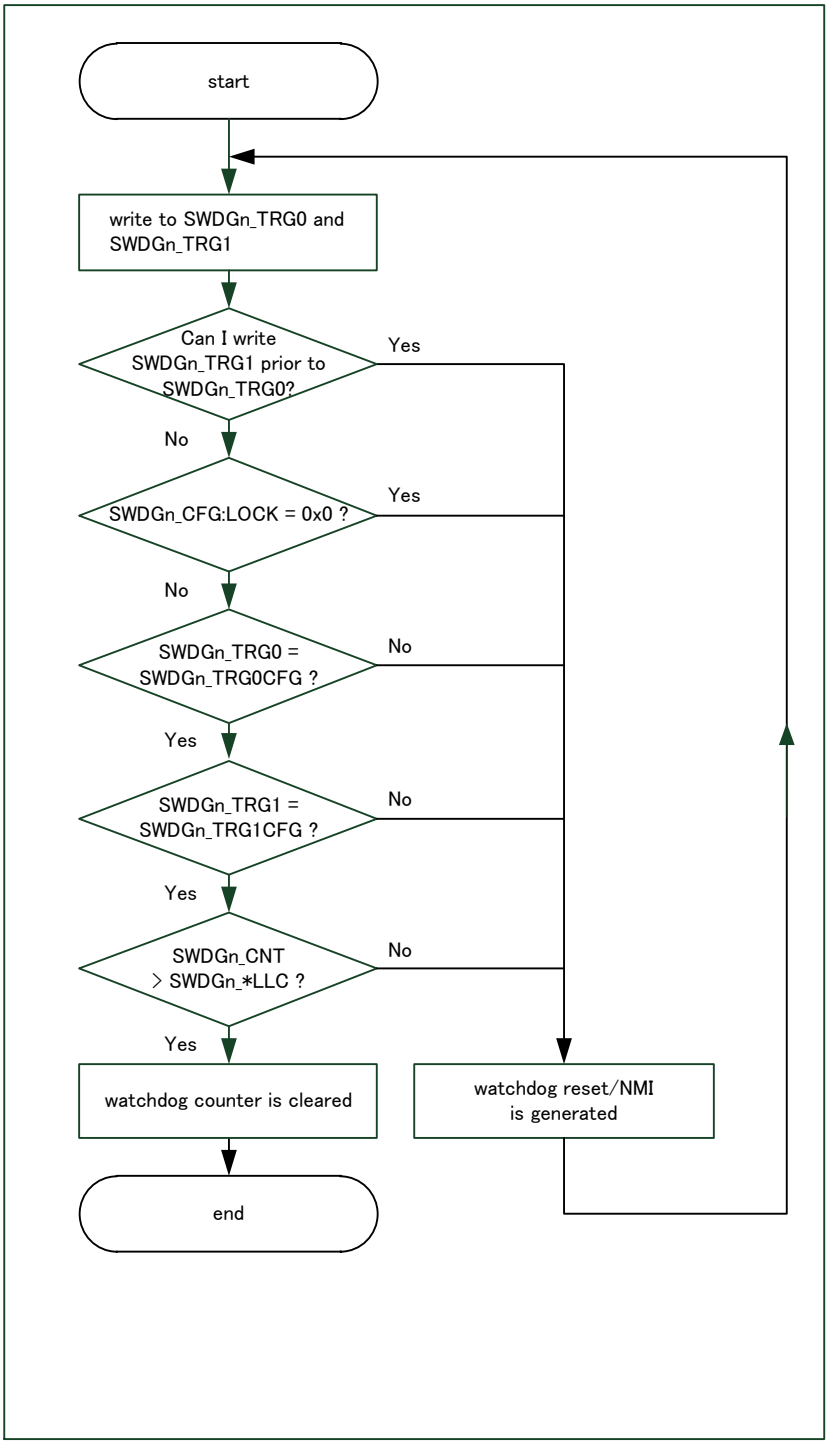
The following figure shows the watchdog register write protection sequence setting procedure.

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

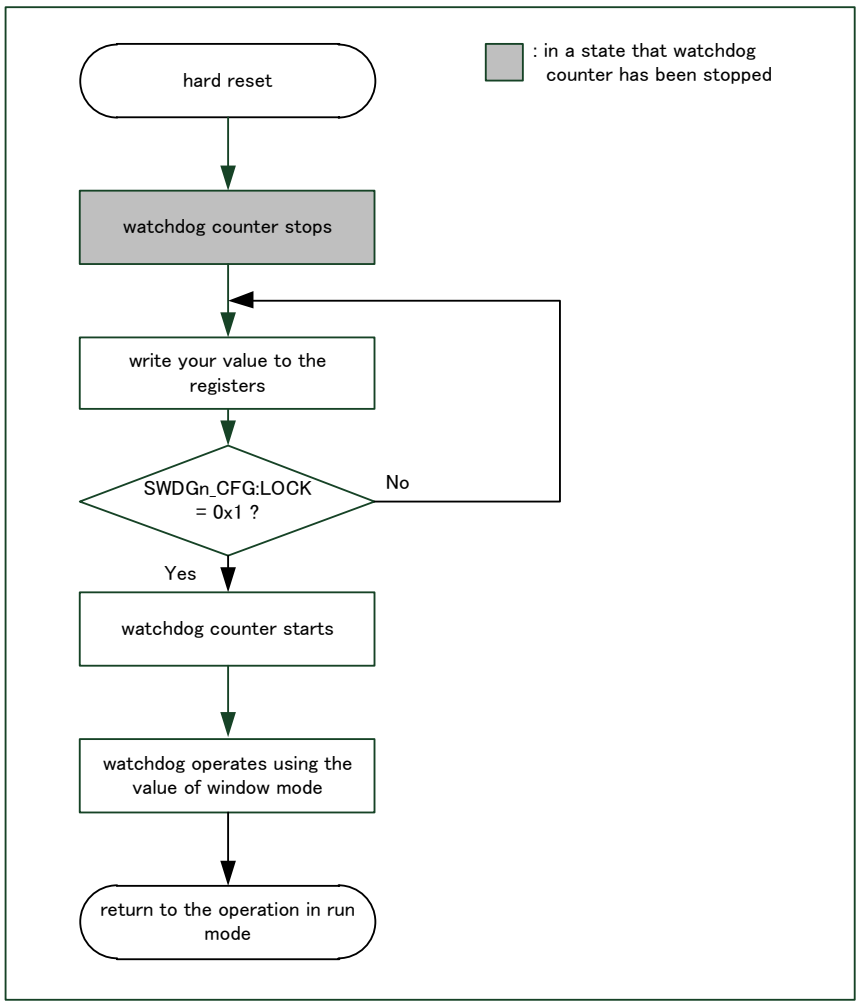


Note:

- *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.

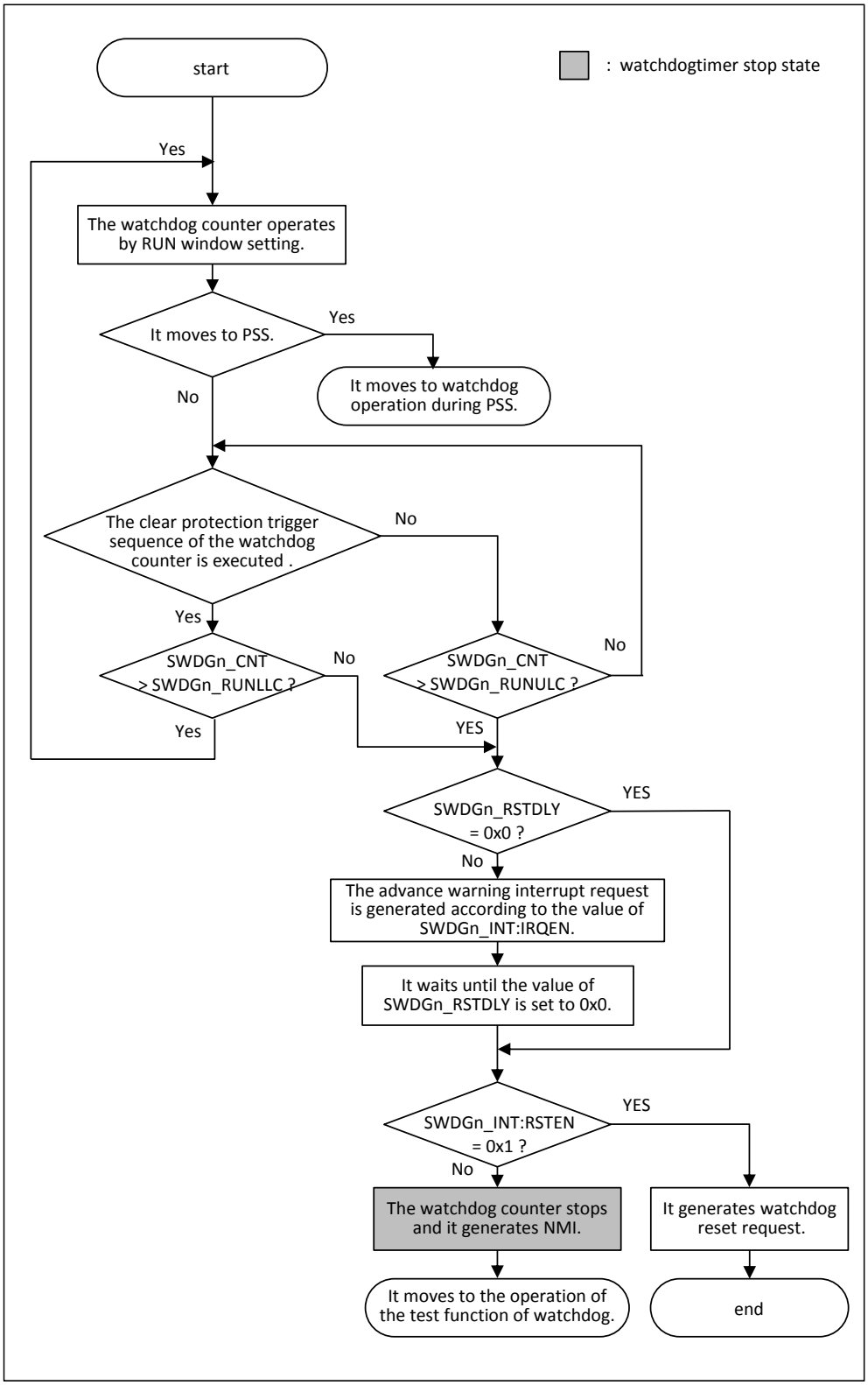
Figure 3-3 Startup of Software Watchdog Timer



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 (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.

The following figure shows the Software watchdog operation in RUN.

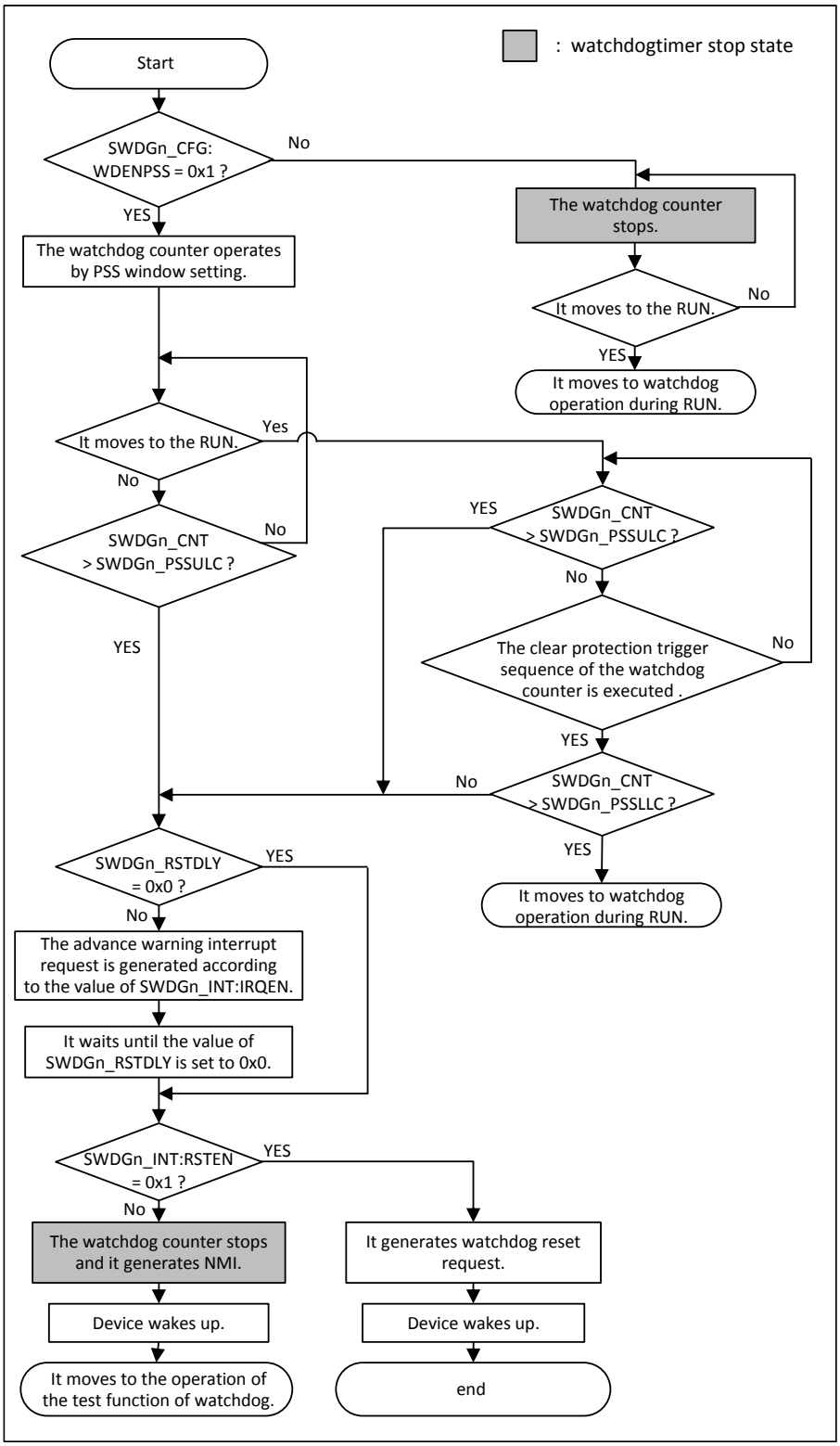
Figure 3-4 Software Watchdog Operation in RUN



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, seeThe following figure shows the Software watchdog operation in PSS.
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 3-5.

The following figure shows the Software watchdog operation in PSS.

Figure 3-5 Software Watchdog Operation in PSS



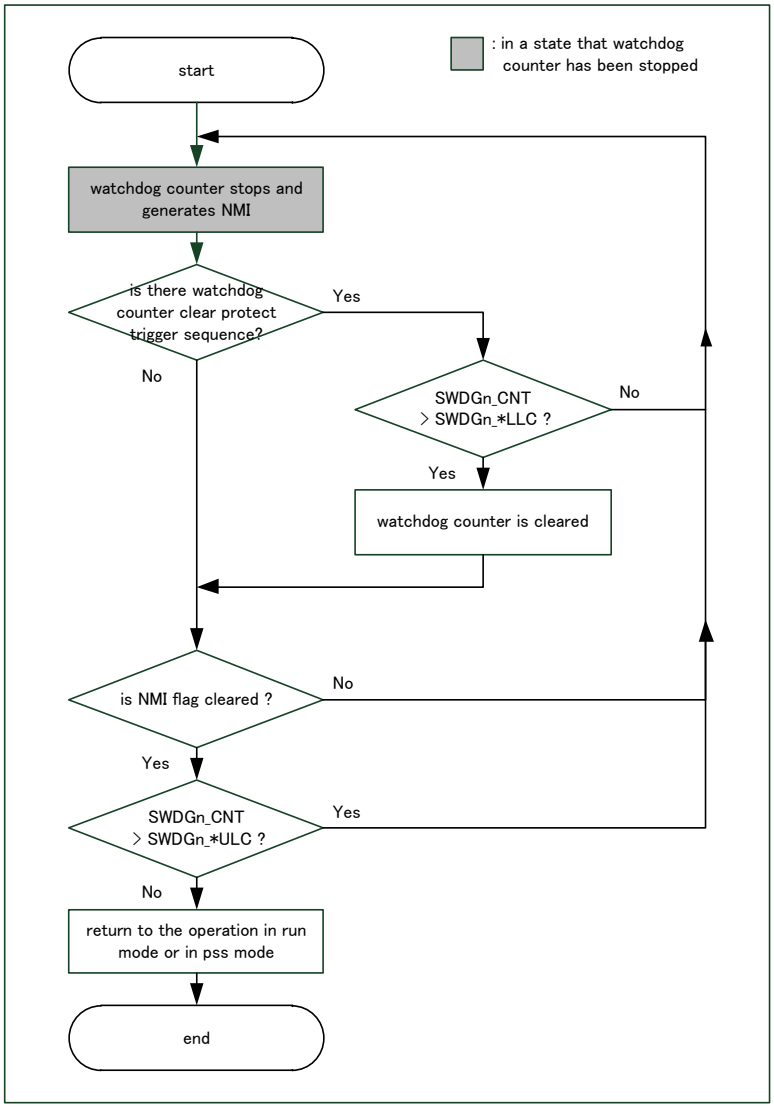
1. The watchdog counter in PSS operates or stops according to the value of the WDENPSS bit in the Software watchdog configuration register (SWDG_CFG).
2. When the device state transitions from PSS to RUN, the watchdog counter operation is started immediately. However, the PSS window setting continues 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.

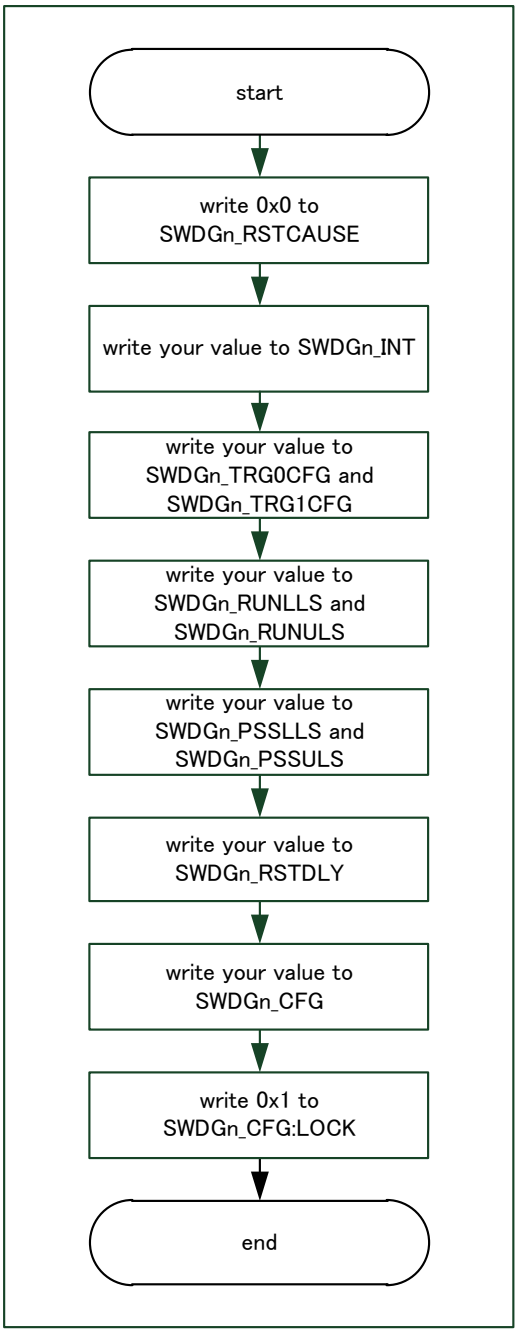
Note:

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



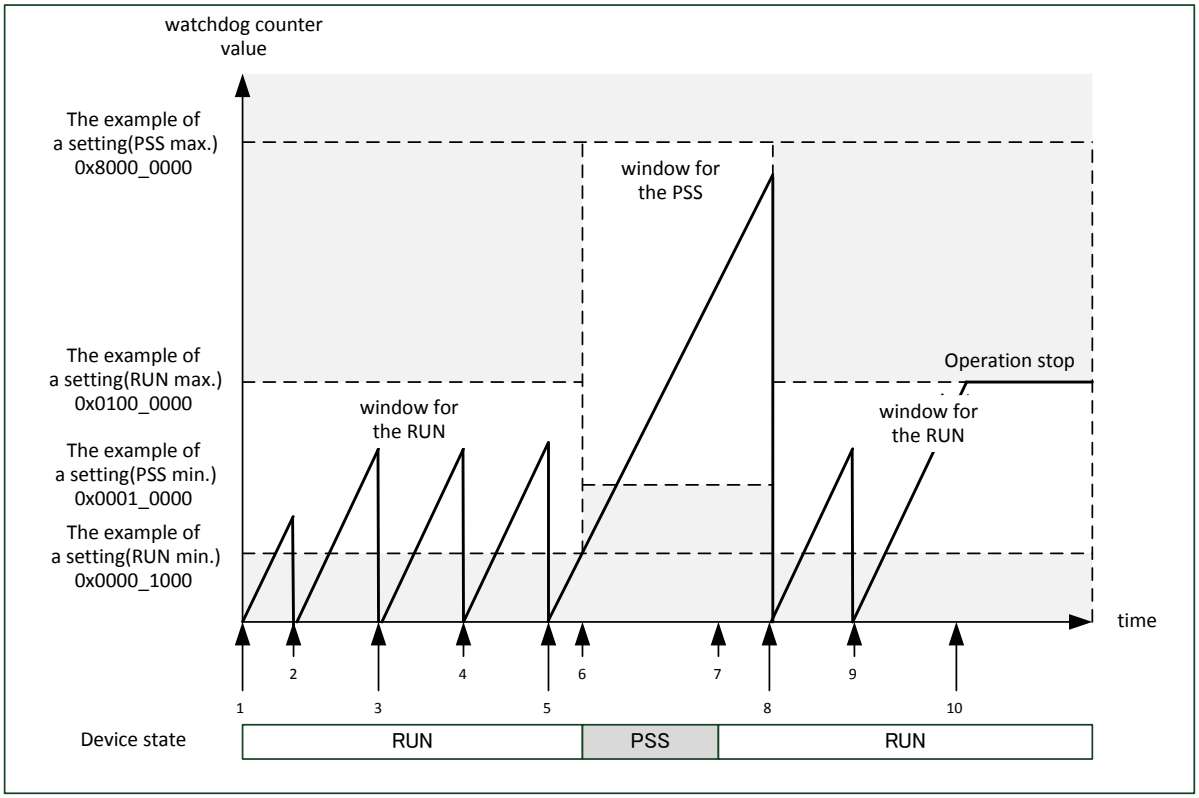
Note:

- *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)

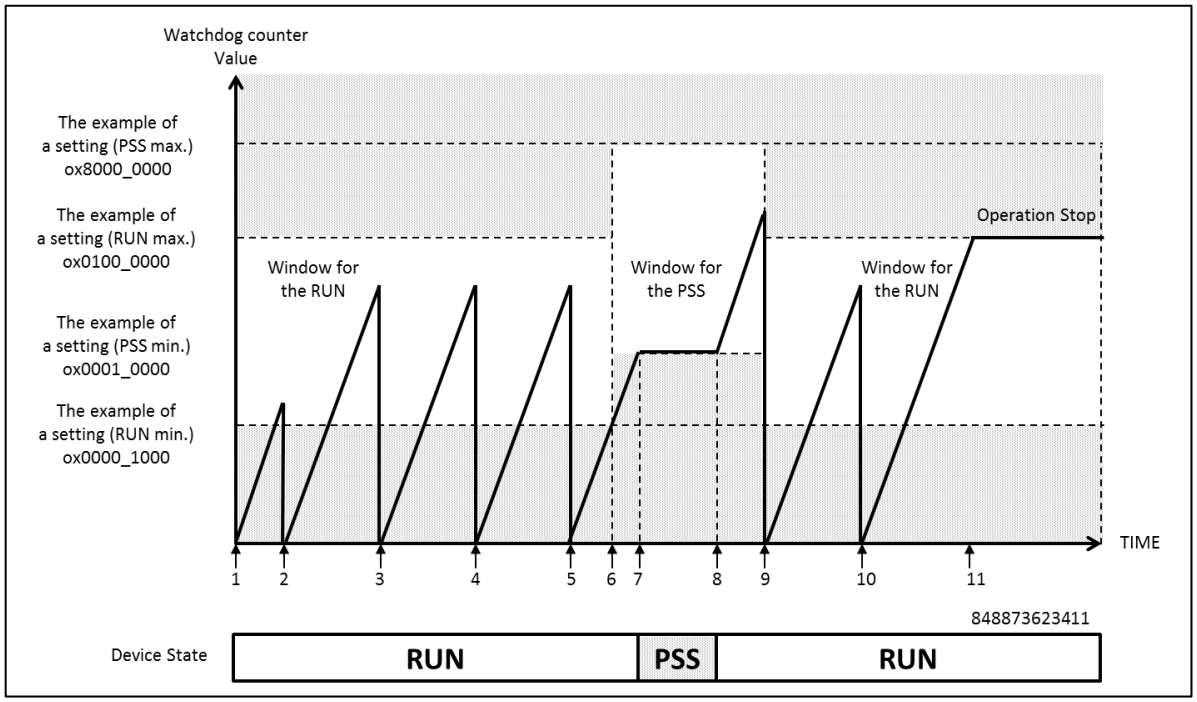


1. Before starting the software watchdog timer, the setting values of the registers are written.
2. The LOCK bit in the software watchdog configuration register (SWDGn_CFG) is set to "1" to start the software watchdog timer. At this time, it starts up-counting from 0x00000000.
3. The watchdog counter clear protection trigger sequence is executed, and the watchdog counter is cleared.
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 (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.
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 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 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.*

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.

Register Configurations

Bit	31	0
Field	KEY[31:0]	
R/W Attribute	R, W	
Protection Attribute	WP	
Initial Value	00000000_00000000_00000000_00000000	

Register Functions

[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.*

6.2. Software Watchdog Counter Register (SWDG_CNT)

This register indicates the current up count value of the watchdog counter.

Register Configurations

Bit	31	0
Field	WDGCNT[31:0]	
R/W Attribute	R, WX	
Protection Attribute	-	
Initial Value	00000000_00000000_00000000_00000000	

Register Functions

[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. To check the factor of the watchdog reset request, read this register.

Register Configurations

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

Register Functions

[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.

Register Configurations

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							

Register Functions

[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.

Register Configurations

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							

Register Functions

[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 interrupt request (NMI) and prior warning interrupt request. This register also includes interrupt status flags.

Register Configurations

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

Register Functions

[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 interrupt 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 ."*

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.

Register Configurations

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

Register Functions

[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.

Register Configurations

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							

Register Functions

[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.

Register Configurations

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							

Register Functions

[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.

Register Configurations

Bit	31	0
Field	WDGRUNLLS[31:0]	
R/W Attribute	R/W	
Protection Attribute	WPS	
Initial Value	00000000_00000000_00000000_00000000	

Register Functions

[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.

Register Configurations

Bit	31	0
Field	WDGRUNULS[31:0]	
R/W Attribute	R/W	
Protection Attribute	WPS	
Initial Value	00000001_00000000_00000000_00000000	

Register Functions

[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 lower limit value actually used is the value in the Software watchdog lower limit RUN current register (SWDG_RUNLLC).*

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.

Register Configurations

Bit	31	0
Field	WDGPSSLLS[31:0]	
R/W Attribute	R/W	
Protection Attribute	WPS	
Initial Value	00000000_00000000_00000000_00000000	

Register Functions

[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.

Register Configurations

Bit	31	0
Field	WDGPSSULS[31:0]	
R/W Attribute	R/W	
Protection Attribute	WPS	
Initial Value	10000000_00000000_00000000_00000000	

Register Functions

[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).

Register Configurations

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	

Register Functions

[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.

Register Configurations

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						
R/W Attribute	R0, WX		R0/W0				
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	0000			0	1	1

Register Functions

[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.

[bit23:10] 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: prohibit to setting. 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 (SWDGn_RUNLLS) when the watchdog counter clear protection sequence is executed.

Register Configurations

Bit	bit31	bit0
Field	WDGRUNLLC[31:0]	
R/W Attribute	R, WX	
Protection Attribute	-	
Initial Value	00000000_00000000_00000000_00000000	

Register Functions

[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 (SWDGn_RUNULS) when the watchdog counter clear protection trigger sequence is executed.

Register Configurations

Bit	bit31	bit0
Field	WDGRUNULC[31:0]	
R/W Attribute	R, WX	
Protection Attribute	-	
Initial Value	00000001_00000000_00000000_00000000	

Register Functions

[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 (SWDGn_PSSLLS) when the watchdog counter clear protection trigger sequence is executed.

Register Configurations

Bit	bit31	bit0
Field	WDGPSSLLC[31:0]	
R/W Attribute	R, WX	
Protection Attribute	-	
Initial Value	00000000_00000000_00000000_00000000	

Register Functions

[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 (SWDGn_PSSULS) when the watchdog counter clear protection trigger sequence is executed.

Register Configurations

Bit	bit31	bit0
Field	WDGPSSULC[31:0]	
R/W Attribute	R, WX	
Protection Attribute	-	
Initial Value	10000000_00000000_00000000_00000000	

Register Functions

[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 (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.
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 (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).

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 (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.

CHAPTER 16: TCRAM Interface

This chapter provides an overview of the interface between the Cortex-R5 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

1. Overview

This section describes the features of the TCRAM interface.

Features

The TCRAM interface has the following features.

- Connection between the Cortex-R5 BTCM port and SRAM (32 KB x 2 modules)

- Wait control function

  The Cortex-R5 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-R5 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-R5.

- 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 (unique data is {address [6:0], 4 {address [7:0]}}. Upper 7 bits are ECC data.)
  - Checker
  - March (performed in the order of all "0's" to all "1's")

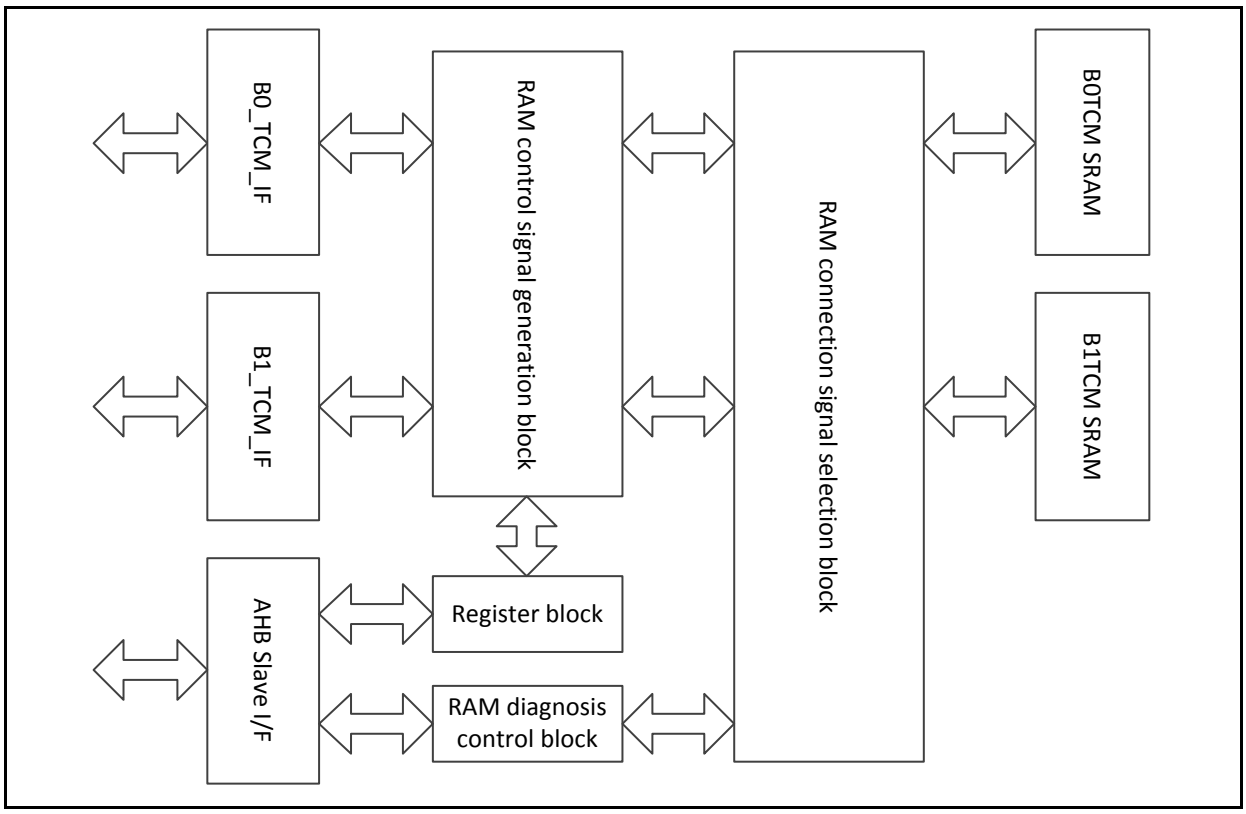
  One of the following can be selected as initialization.

  - 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-R5 B0TCM port

B1_TCM_IF

Interface for the Cortex-R5 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-R5 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-R5 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 TCRM 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-R5 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-R5 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".*

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 (unique data is {address [6:0], 4 {address [7:0]}}. Upper 7 bits are ECC data.)
- 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, and Figure 4-2 shows a procedure for starting RAM diagnosis and initialization.

Figure 4-1 Register Setting Flow

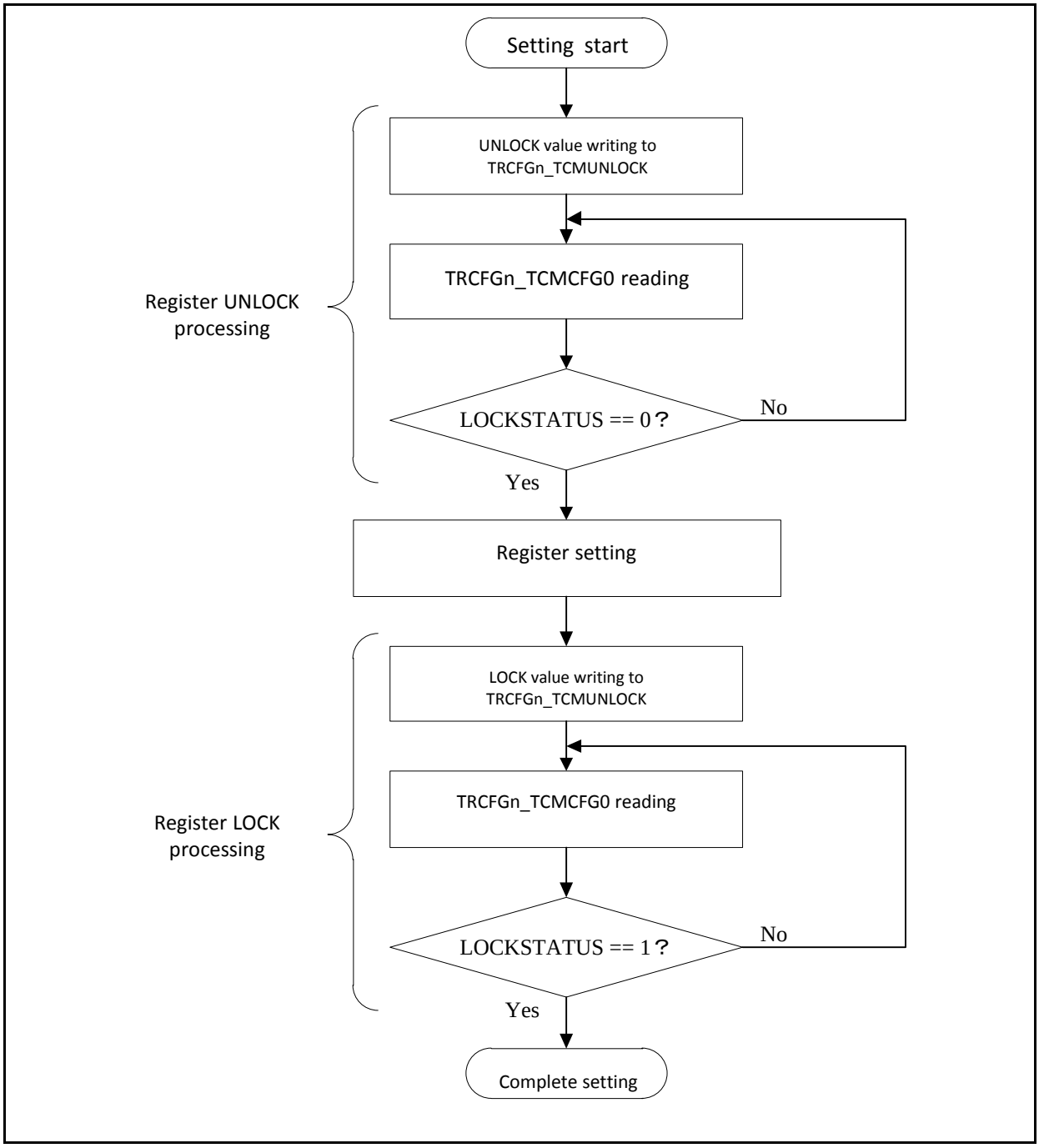
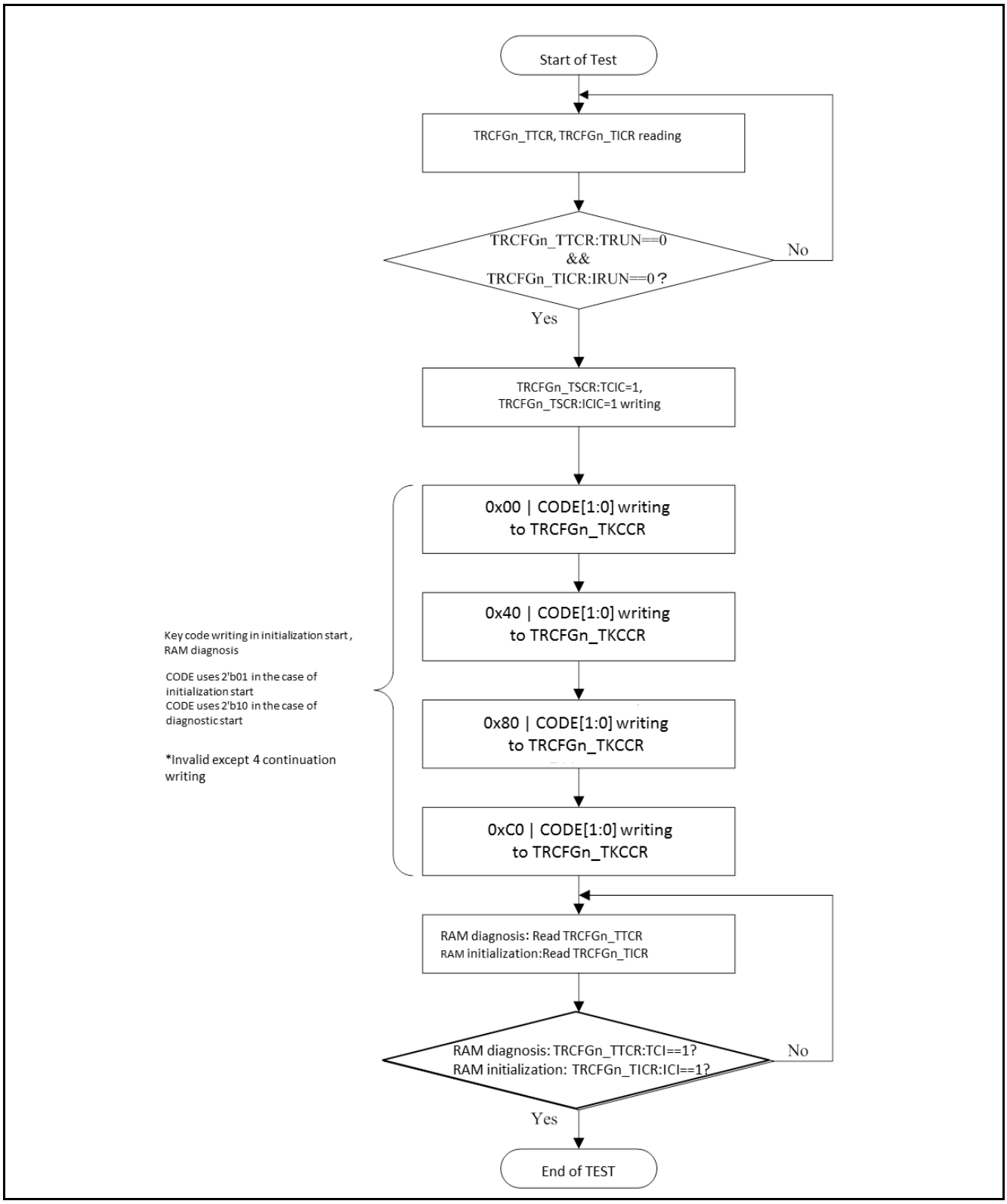


Figure 4-2 RAM Diagnosis and Initialization Start Flow



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_TEAR0	TEST error address register 0	5.4
TRCFGn_TEAR1	TEST error address register 1	5.5
TRCFGn_TEAR2	TEST error address register 2	5.6
TRCFGn_TAEAR	TEST end address register	5.7
TRCFGn_TASAR	TEST start address register	5.8
TRCFGn_TFECR	TEST pseudo error generation control register	5.9
TRCFGn_TICR	TEST initialization function register	5.10
TRCFGn_TTCR	TEST diagnosis function register	5.11
TRCFGn_TSRCR	TEST soft reset generation control register	5.12
TRCFGn_TSCR	TEST status clear register	5.14
TRCFGn_TKCCR	TEST key code control register	5.13

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

LOCK STATUS	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-R5 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-R5 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-R5 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	WPS	
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 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[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	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.

[bit28:14] Reserved: Reserved Bits

[bit13:0] 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 0b000000_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'b00} to the base address.*

5.5. 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[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	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".

[bit28:14] Reserved: Reserved Bits

[bit13:0] 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 0b000000_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'b00} to the base address.*

5.6. 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[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	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".

[bit28:14] Reserved: Reserved Bits

[bit13:0] 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 0b0b000000_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'b00} to the base address.*

5.7. 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[13:8]					
R/W Attribute	R0,WX		R/W					
Protection Attribute	WPS							
Initial Value	00		111111					

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							

[bit15:14] Reserved: Reserved Bit

[bit13:0] 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'b00} to the base address.*

5.8. 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[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	SADDR[7:0]							
R/W Attribute	R/W							
Protection Attribute	WPS							
Initial Value	00000000							

[bit15:14] Reserved: Reserved Bit

[bit13:0] 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'b00} to the base address.*

5.9. 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.

5.10. 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.

5.11. 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 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 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.*

5.12. 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

5.13. 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 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 4-2.
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*

5.14. 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 2 times as wide as 64 bits. If there are consecutive addresses, 64-bit data can be used with a frequency 2 times as high as the flash clock frequency.
- The TCM has 16 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)-R5 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-R5 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.
- Bus error response is sent when access is to the reserved area.
- 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.

2. Configuration

This section explains the configuration within TCFLASH.

2.1. Block Diagrams

- There is 1 TCFLASH per 1 Cortex-R5 core. (TCFLASH has 3 flash memories: 1 memory of 1 MB/2 MB and 2 memories of 4 MB.)
- When Cortex-R5 core accesses via the TCM region, TCFLASH is handled as a read-only L1 memory. On the other hand, when Cortex-R5 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

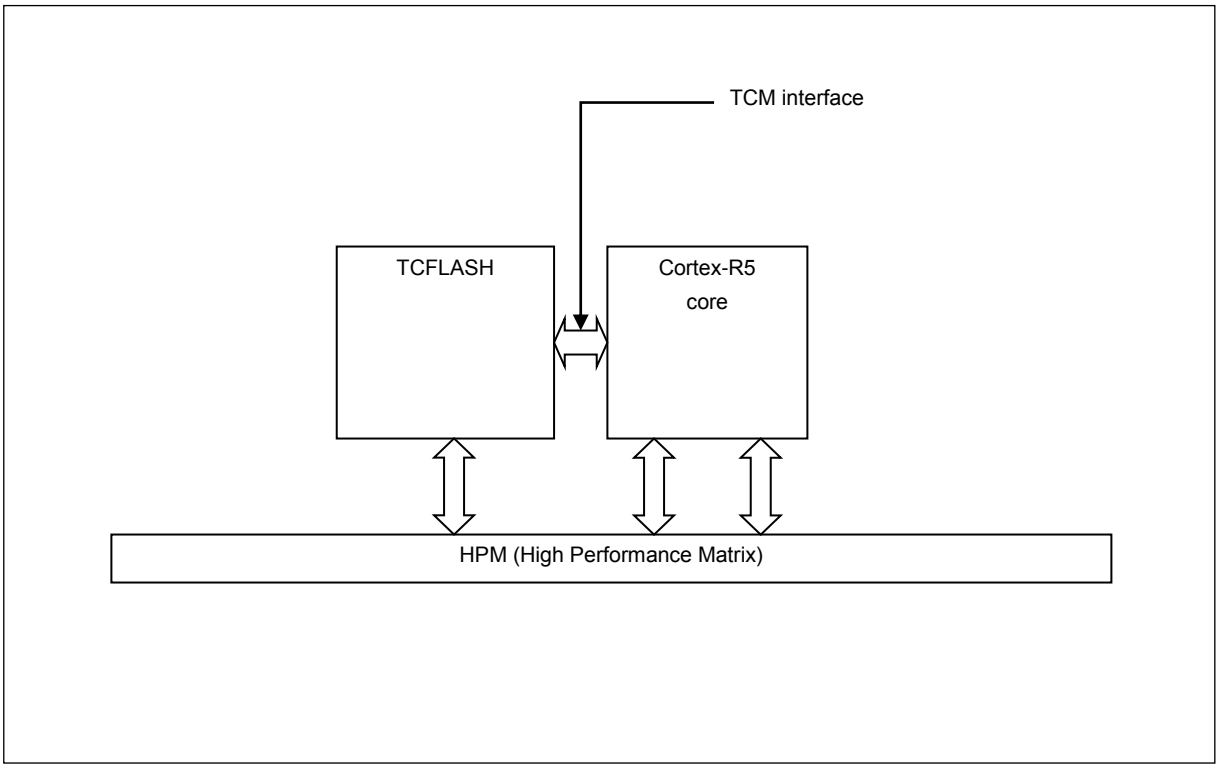
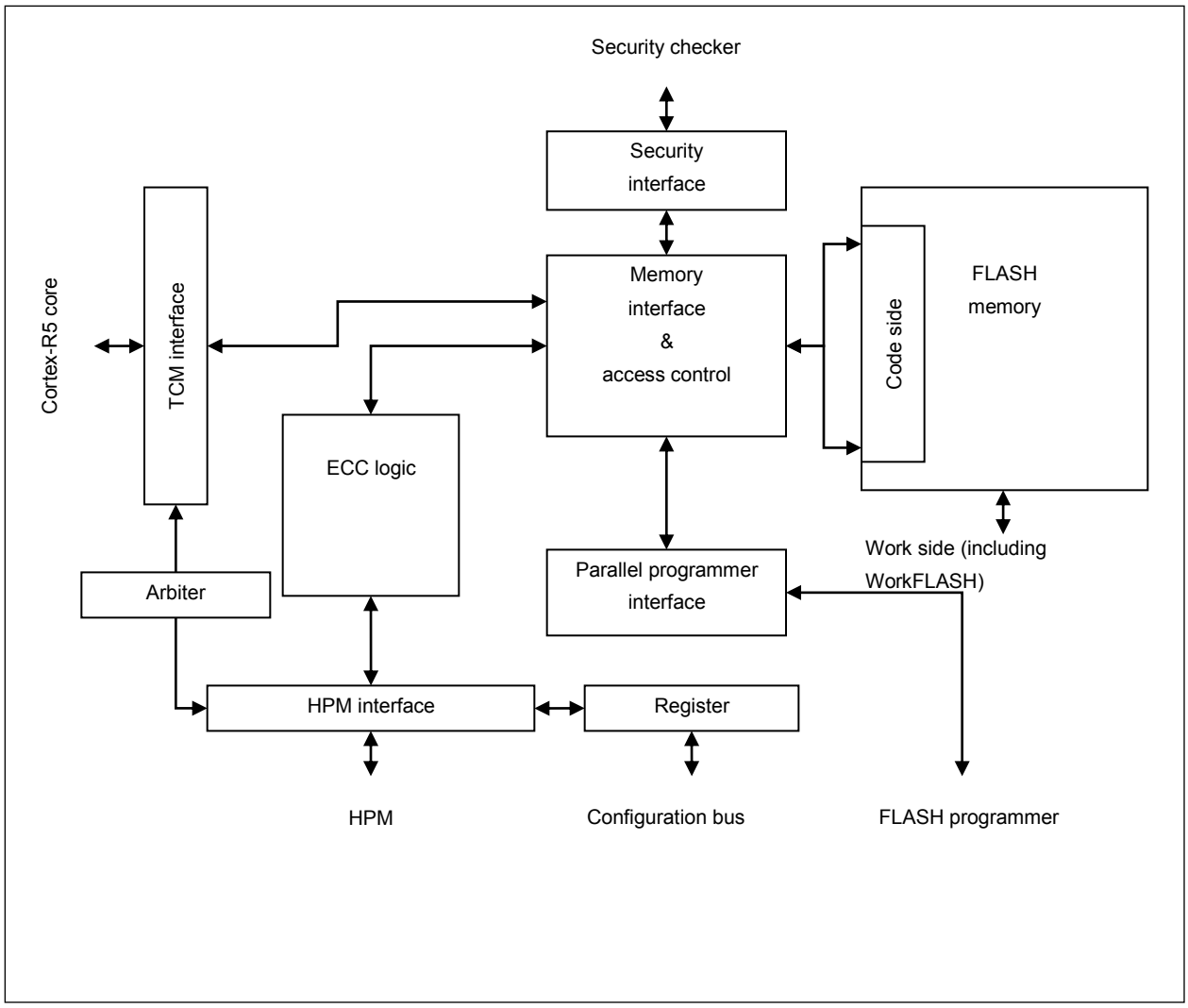


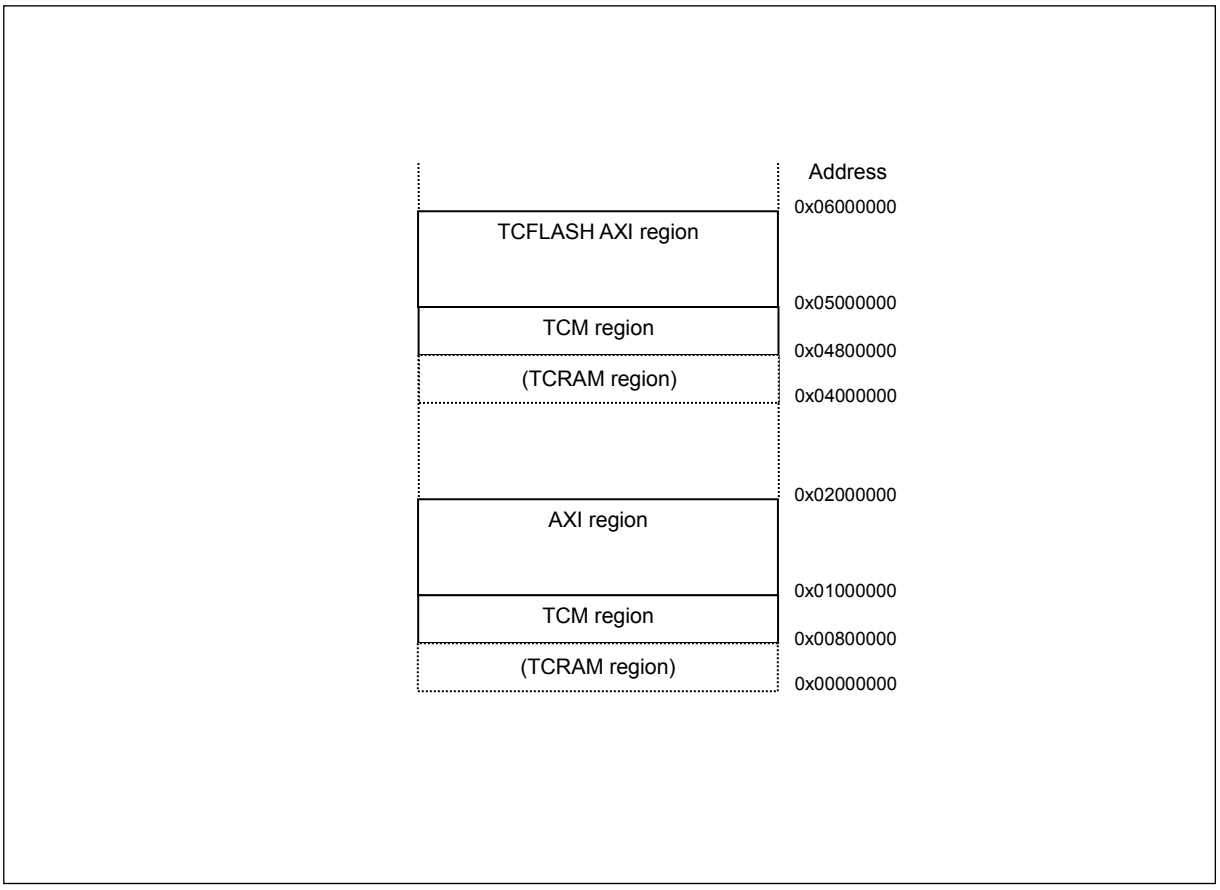
Figure 2-2 Configuration of TCFLASH



2.2. Address/Sector Map of TCFLASH

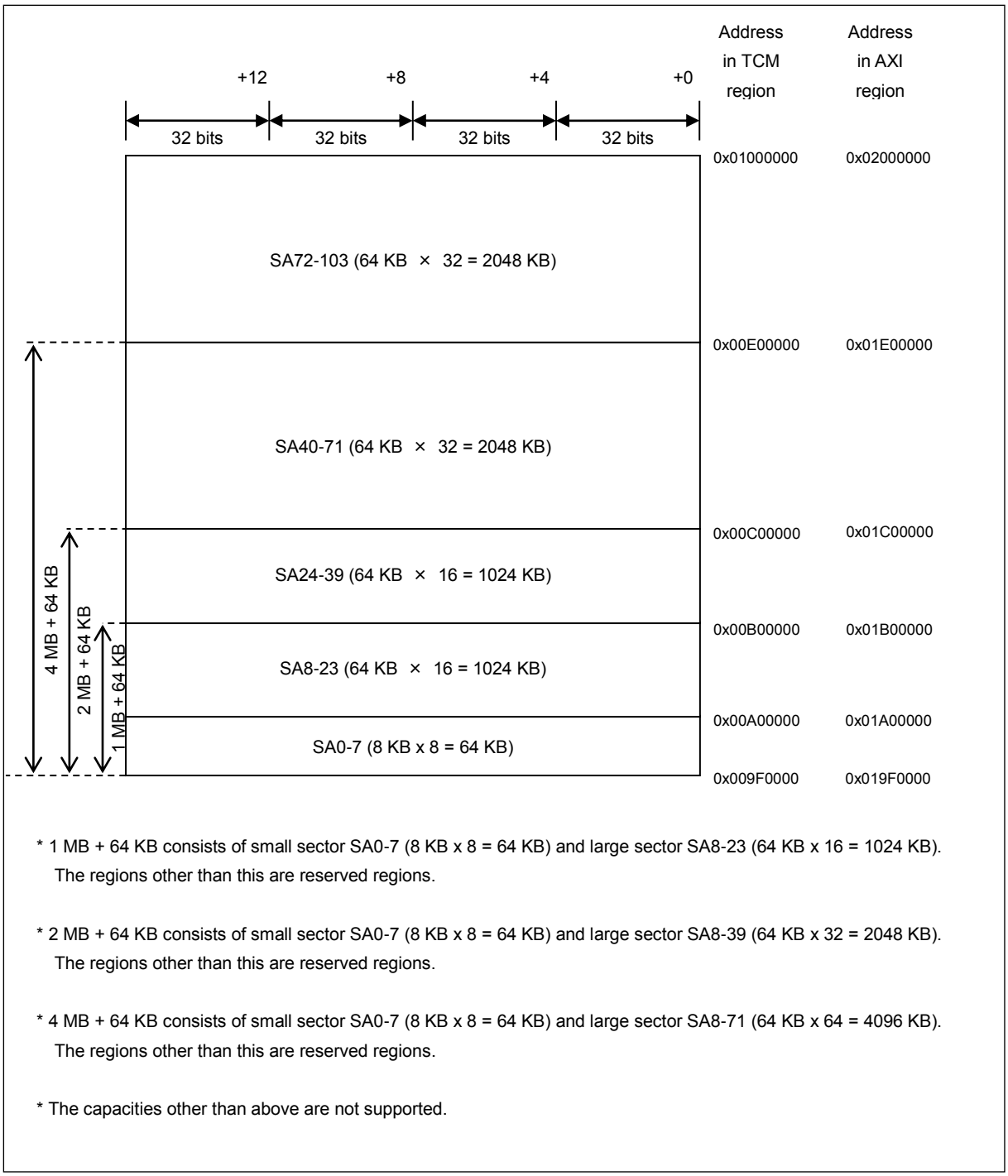
A TCFLASH exists for CPU core and is placed on the memory map by the program as shown in Figure 2-3.

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.

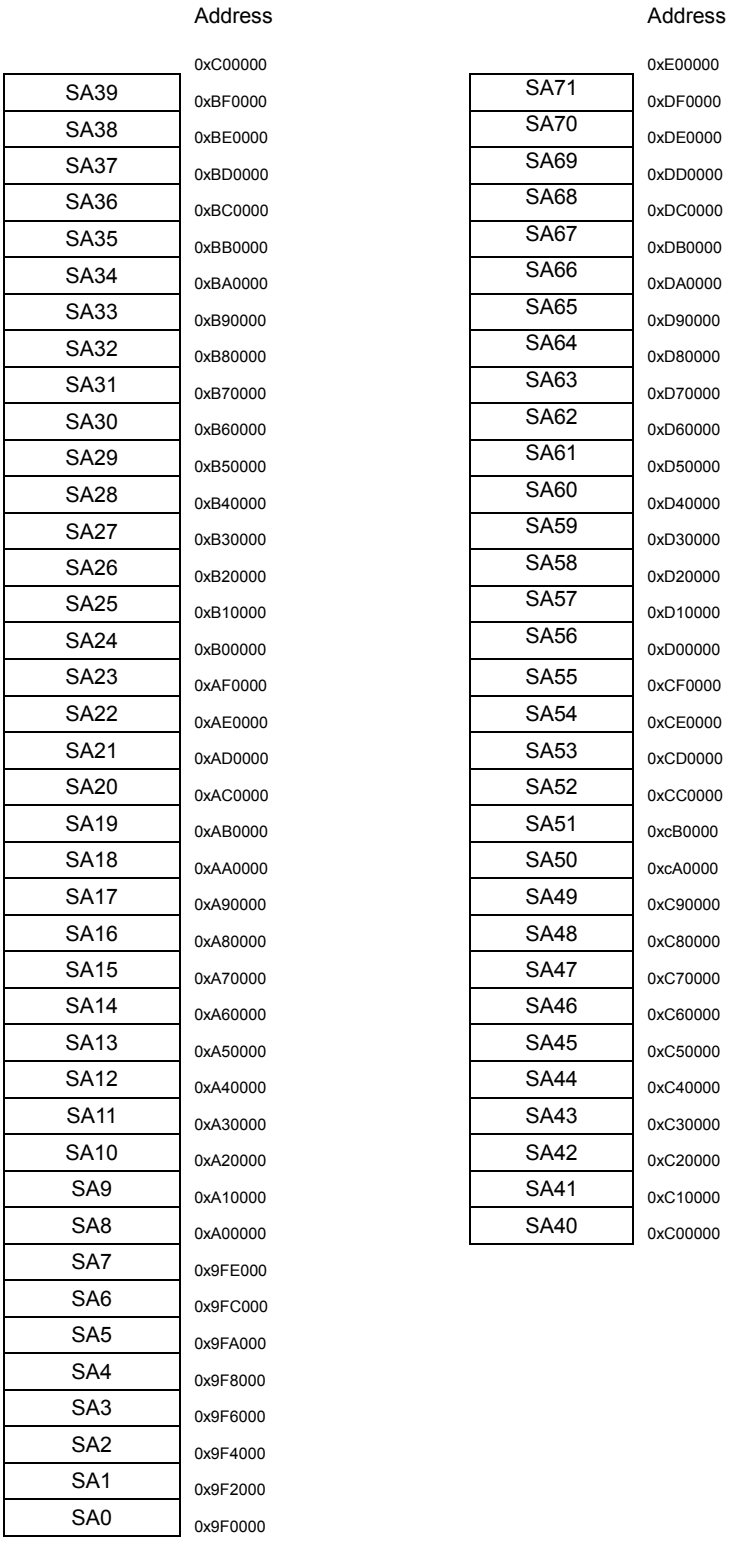
* The capacities other than above are not supported.

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



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-R5 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-R5 core, via AXI Bus masters (non-Cortex-R5 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)

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.

*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.

.

- 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.

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".

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. 16 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 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 FLASHmacros(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

- PGMS (ProGraM Status) bit

  This bit indicates the programming status.

- CERS (Chip ERase Status) bit

  This bit indicates the macro erase status.

  Any command can be received until macro erase is completed.

- SERS (Sector ERase Status) bit

  This bit indicates the sector erase status.

  During sector erase, sector erase suspend commands can be received.

- 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.

■ READ (READ rdy) bit

This bit indicates the readable status of TCFLASH.

■ HANG (HANG up1 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-R5 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-R5 core. Error detection/correction for the data and ECC bit read via the TCM space is performed in the Cortex-R5 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.

In the case of S6J312xHzC*, the maximum operating frequency of flash memory built in this LSI is 64 MHz. Therefore, if the system operating frequency is 128 MHz, for example, the setting value for FAWC[1:0] should be ""01"" as calculated using the above formula."

*: x: A/9, z: A/B

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.

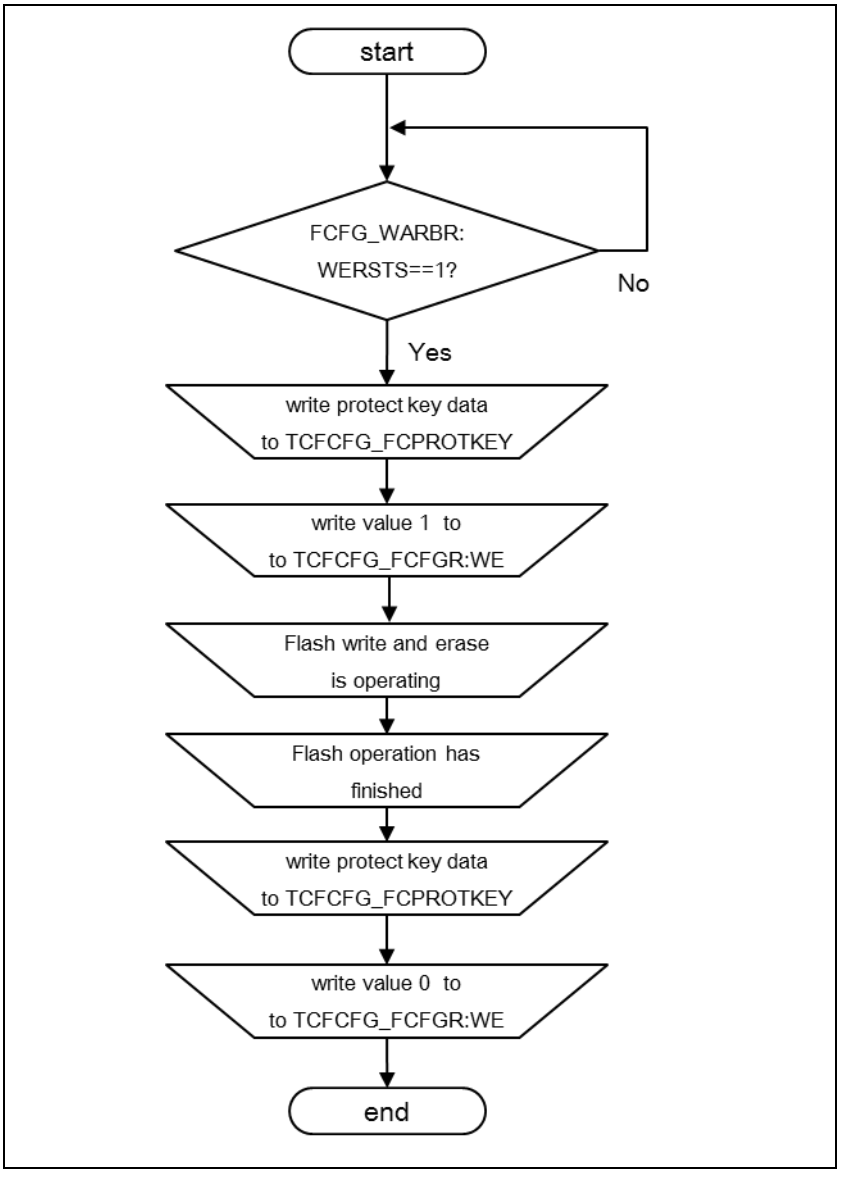
Successful program or erase operation by flash memory is not affected by programming of the read/reset command sequence. In addition, a suspended sector erase cannot be canceled by programming the read/reset command sequence.

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.

- 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

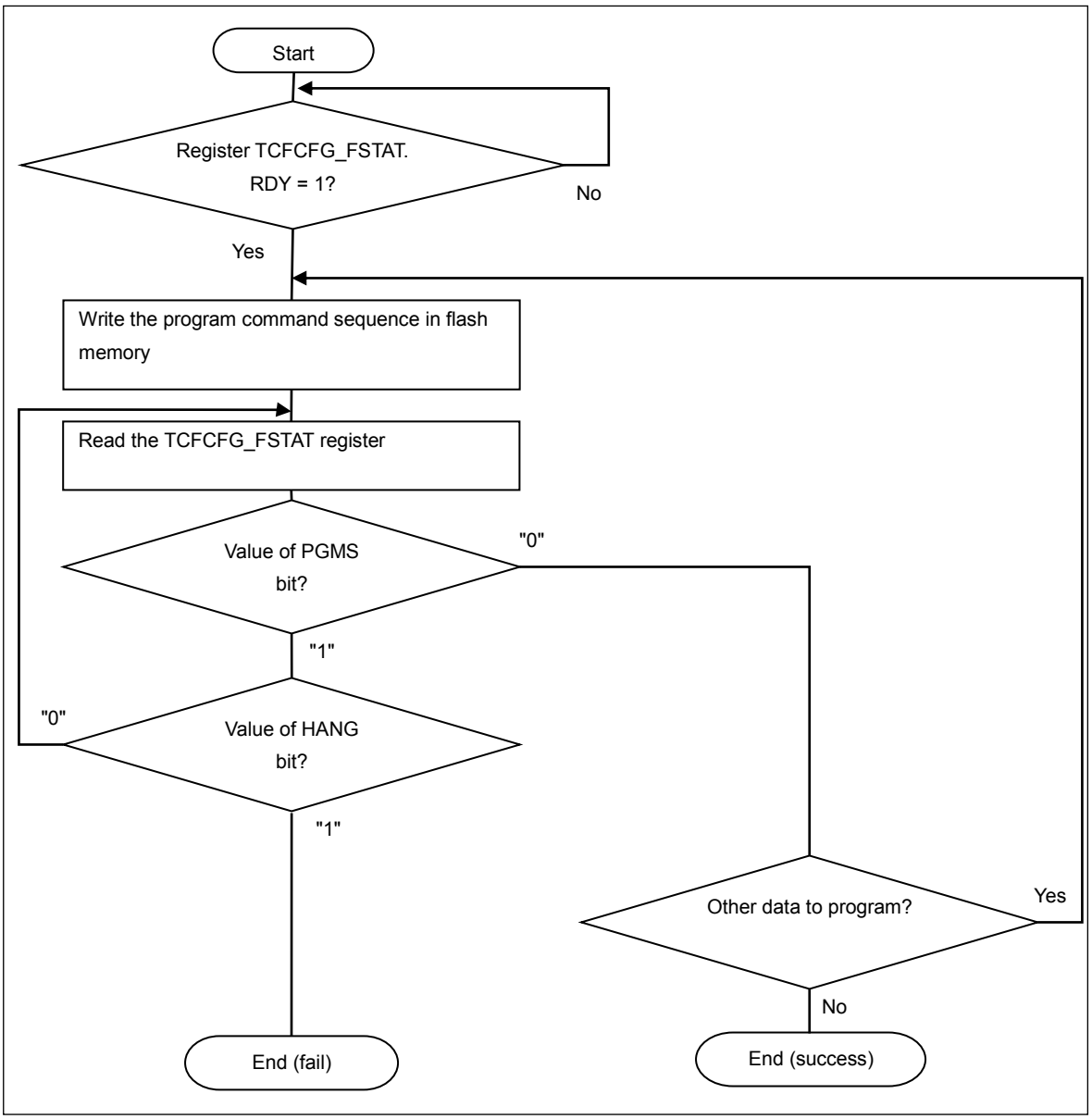
4-1: Address＝CA0, Data = 0xAA

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.

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.

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

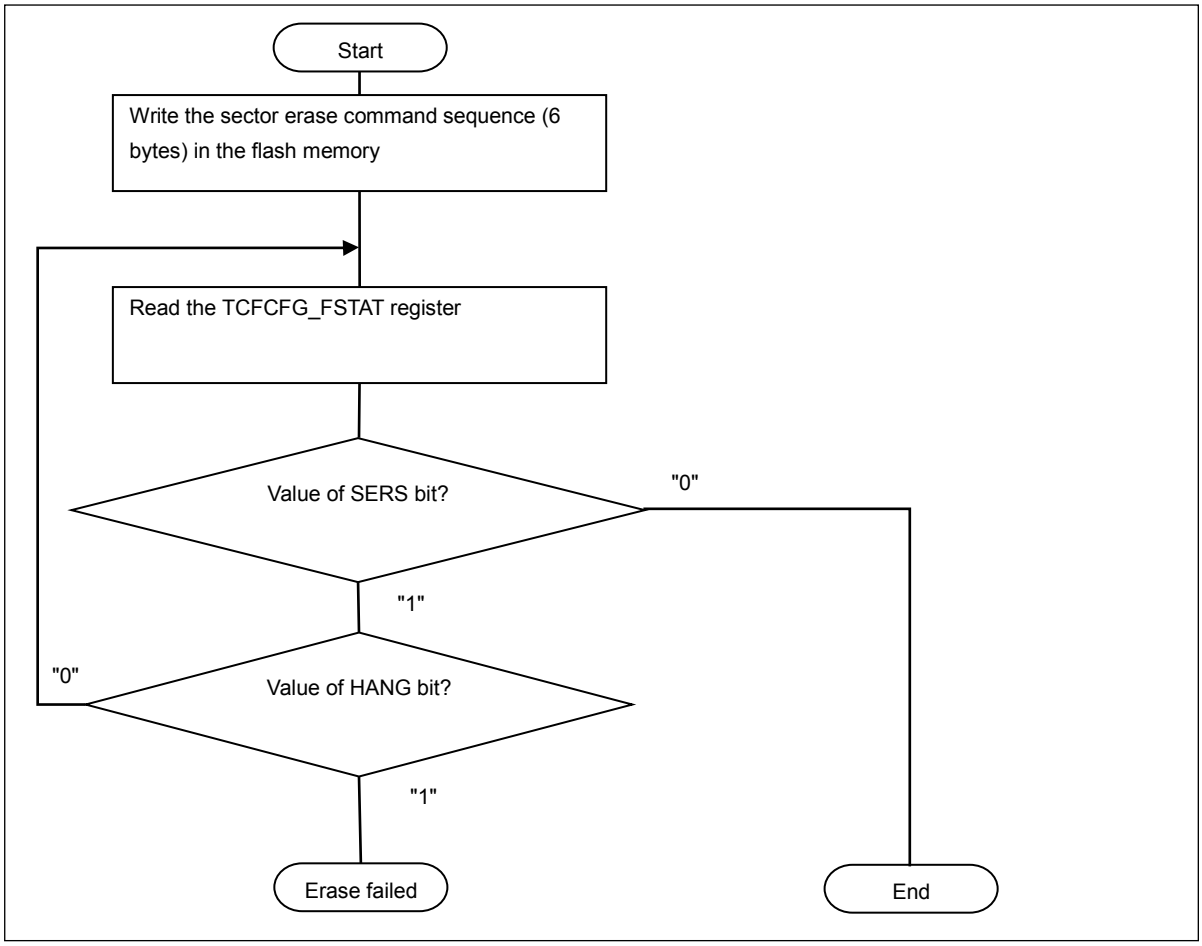
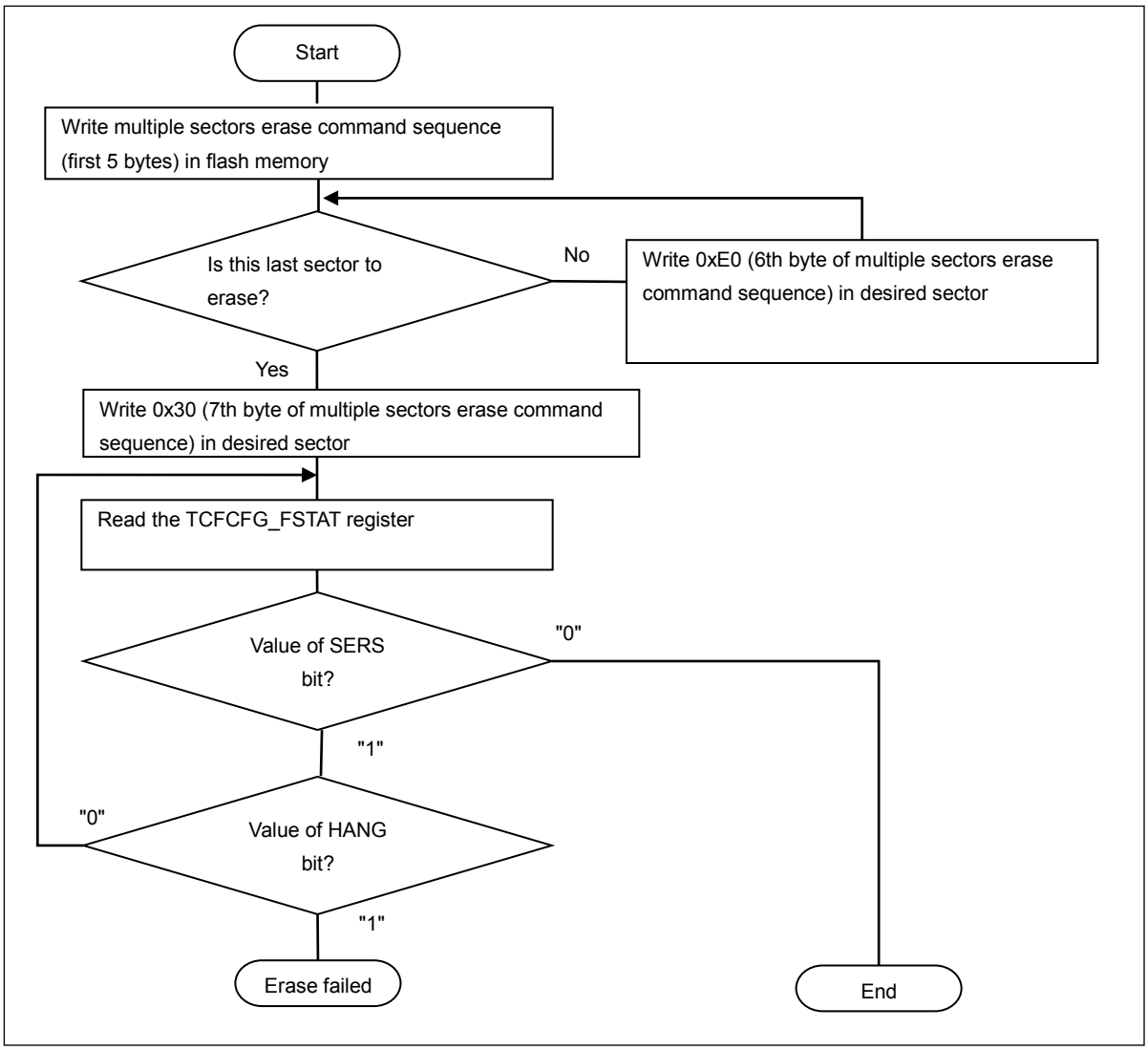


Figure 4-4 Flow Example When Multiple Sectors Are Erased



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.

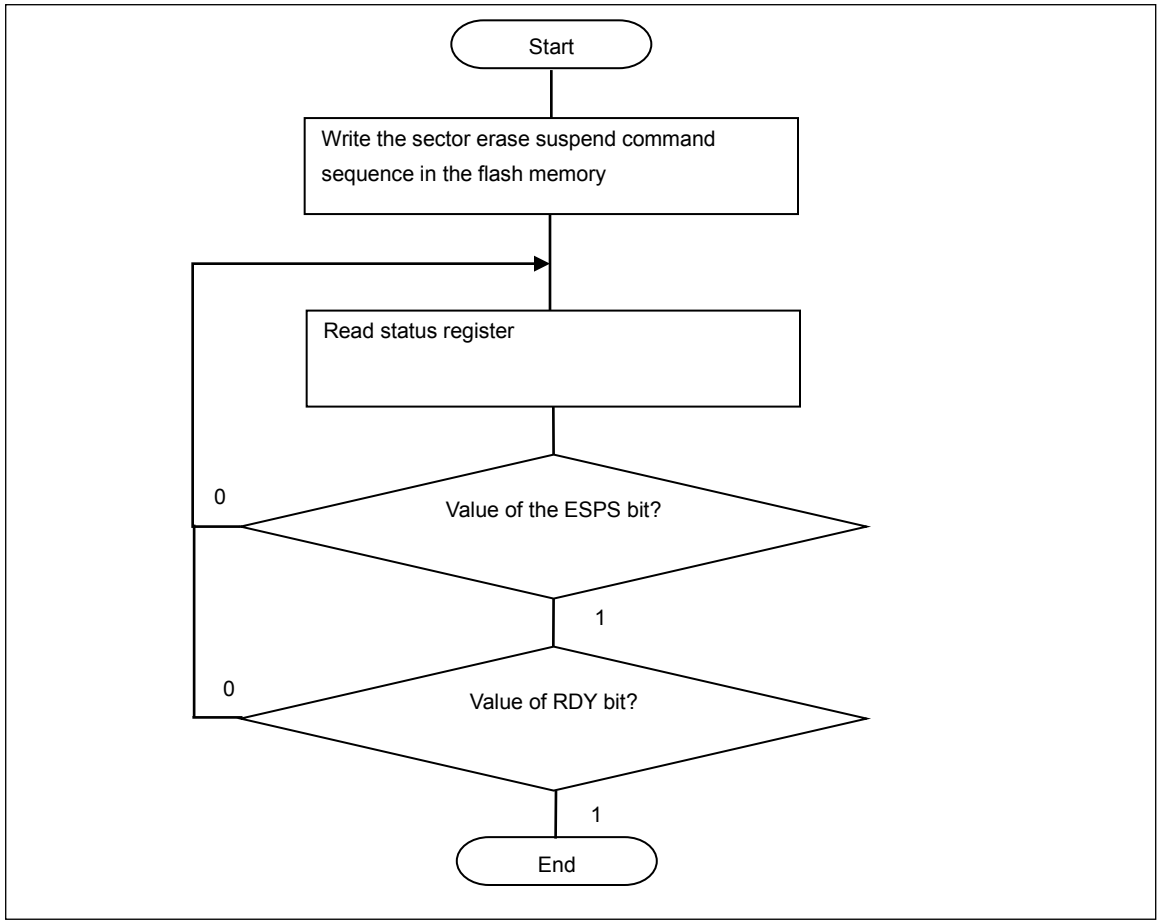
The sector erase suspend command sequence is valid only when flash memory is in the sector erase state. The sector erase suspend command sequence is ignored by the flash memory if the programming or macro erase operation is being performed or the flash memory is already in the sector erase suspend state.

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 shows the location of registers in the register area.

Table 5-1 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

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 (TCFFCFG_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.

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-R5 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 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.

5.5. 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.

5.6. TCFLASH Interrupt Control Register: TCFCFG_FICTRL (TCFLASH Interrupt Control Register)

This register allows enabling interrupt request generation for each factor and clearing interrupt factors.

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

[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.

[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.

[bit7:2] Reserved: Reserved Bits

[bit 1] HANGIE: Hang Interrupt Enable

HANGIE	Description
0	Disable the generation of hang interrupt requests.
1	Setting "1" is prohibited.

[bit 0] RDYIE: Programming/Erasing Ready Interrupt Enable

RDYIE	Description
0	Disable the generation of programming/erasing ready interrupt requests.
1	Setting "1" is prohibited.

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.

[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.

Note:

- *Initial value depends on the boot configuration.*

[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 TCFCG_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 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.

[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 TCMBUF0 to 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 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.

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, retry the suspended program or erase operation after reset completes.

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_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.

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".

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**

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. (64-bit 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.

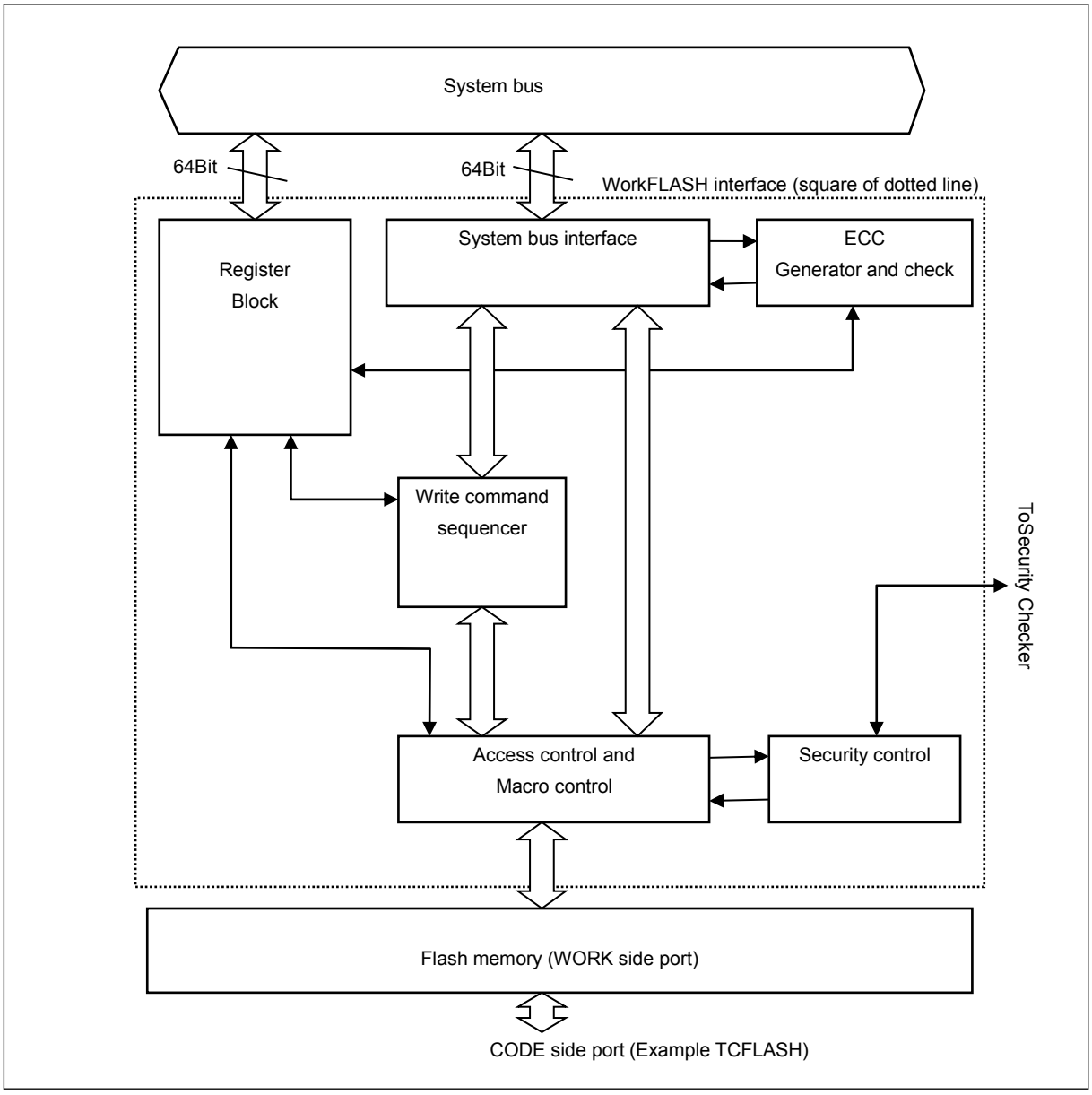
- Equipped with an ECC check circuit test function with error injection.
- Enables writing from not only a CPU but also the DMAC.
- Enables writing from an unprivileged mode.
- Equipped with a control register setting protection function using a key code.
- Equipped with a Flash security function for 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.

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 WorkFLASH is mounted depending on the model.

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.

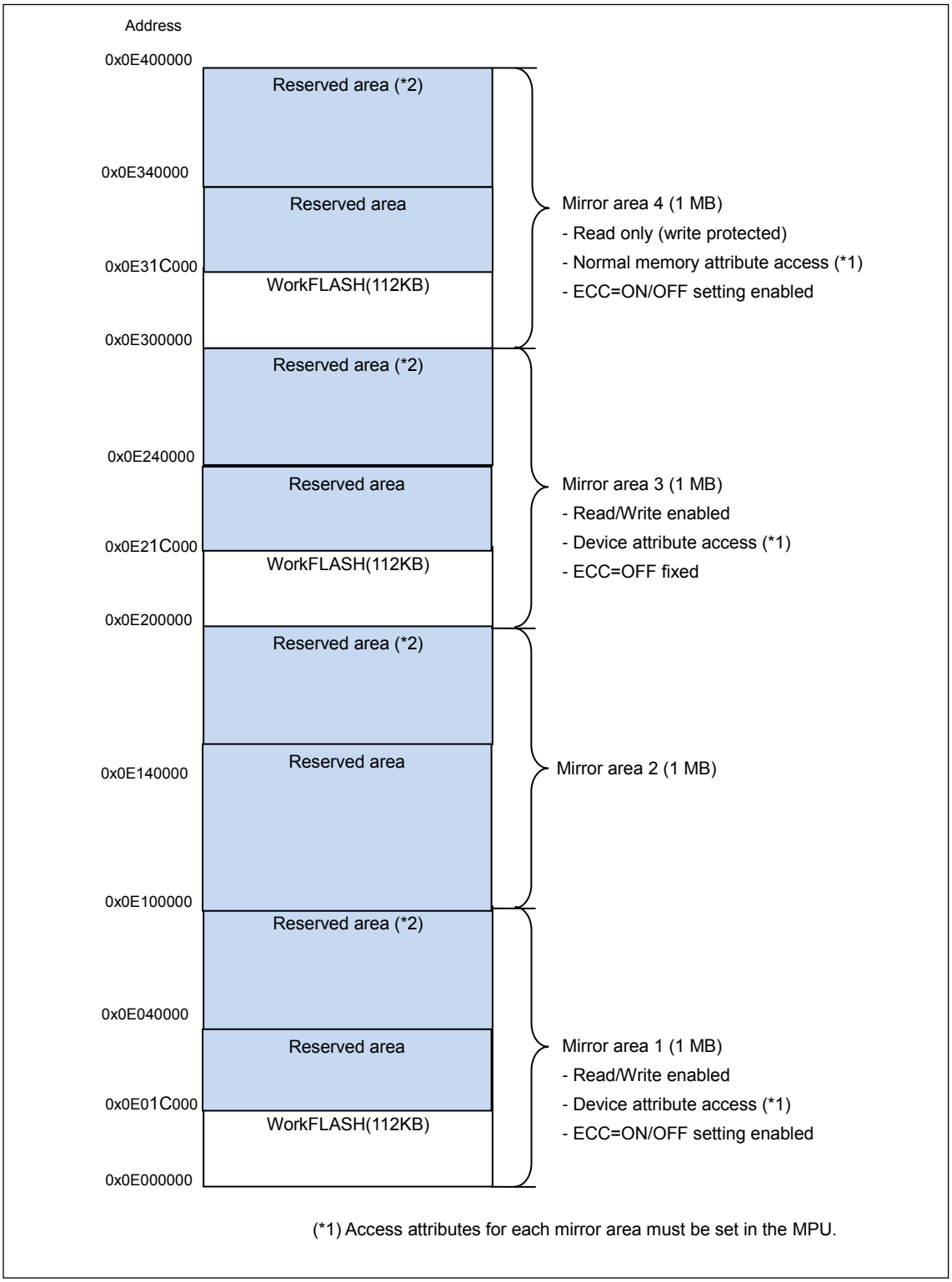
A single WorkFLASH has an area of 48 KB (or 112 KB depending on the assembly). The 48-KB area consists of 6 small sectors (or 14 or 30 sectors depending on the assembly), and the size of each small sector is 8 KB. These sectors are placed as shown in Figure 2-3. Their places in an overall address map 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

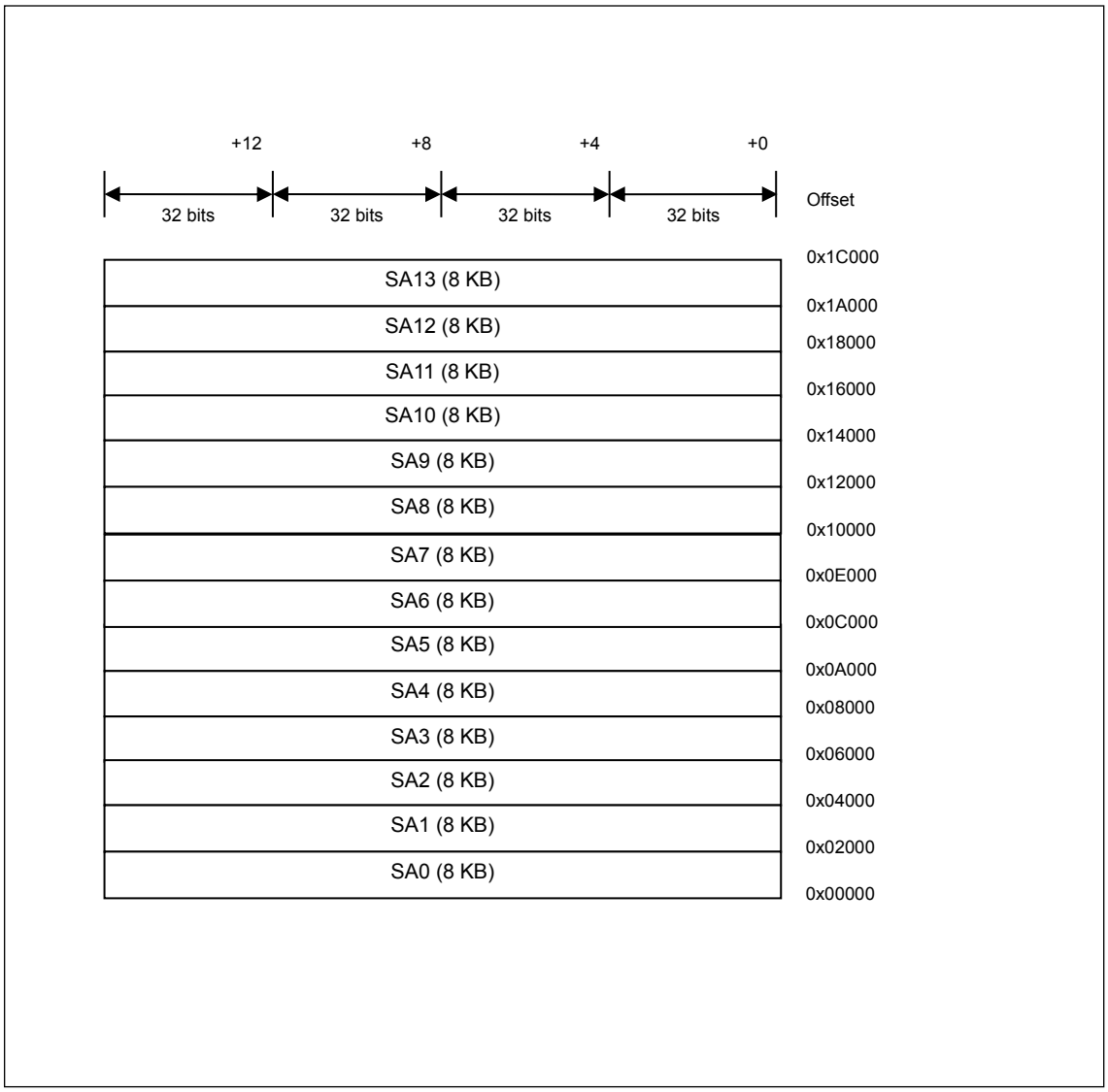
* In order to perform read and write operation as shown above, the access attribute has to be configured for each mirror area by MPU.

Figure 2-2 Address Map of WorkFLASH in User Mode



(*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.)

The Flash security function does not allow writing and erasing of protected sectors. 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)-R5 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	0	0	0	0	0	0	0	0	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 (however, if the reserved area is written, RESA flag is set instead of this bit).	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.	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	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 (the reserved area in Flash and registers)	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

3.8. WorkFLASH Access Restrictions

To prevent the data and programs in WorkFLASH from being read by third parties, operations on WorkFLASH are restricted.

4. Setting Procedure

This section explains the WorkFLASH setting procedures.

4.1. Setting the Number of Wait Cycles

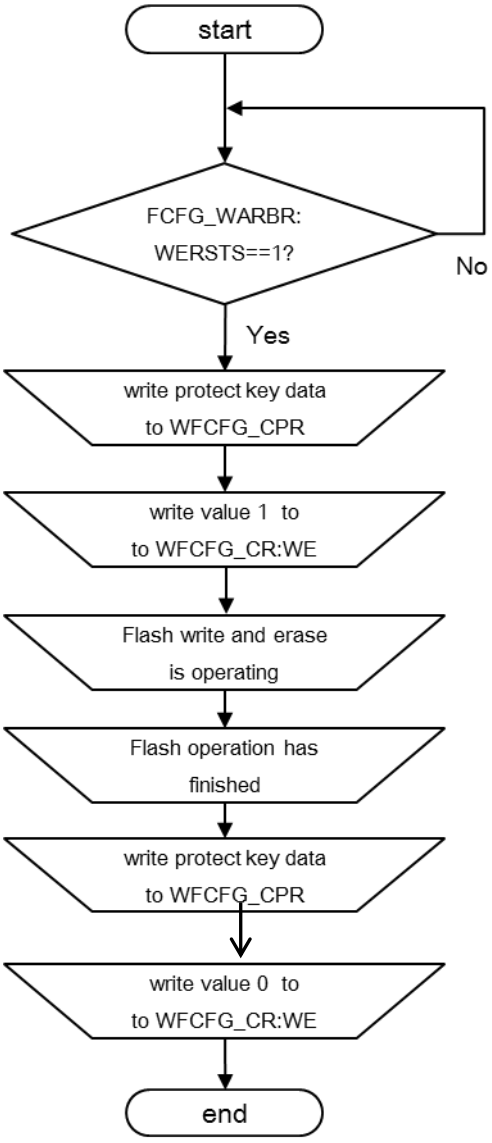
Since WorkFLASH and TCFLASH use the same Dual Port Flash memory, the FAWC bit setting in the TCFCFG_FCFGR register can insert a wait cycle at the Flash memory access time. For details on how to set it, see "4.1 Setting Wait Cycle Count" in the TCFLASH specifications.

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.

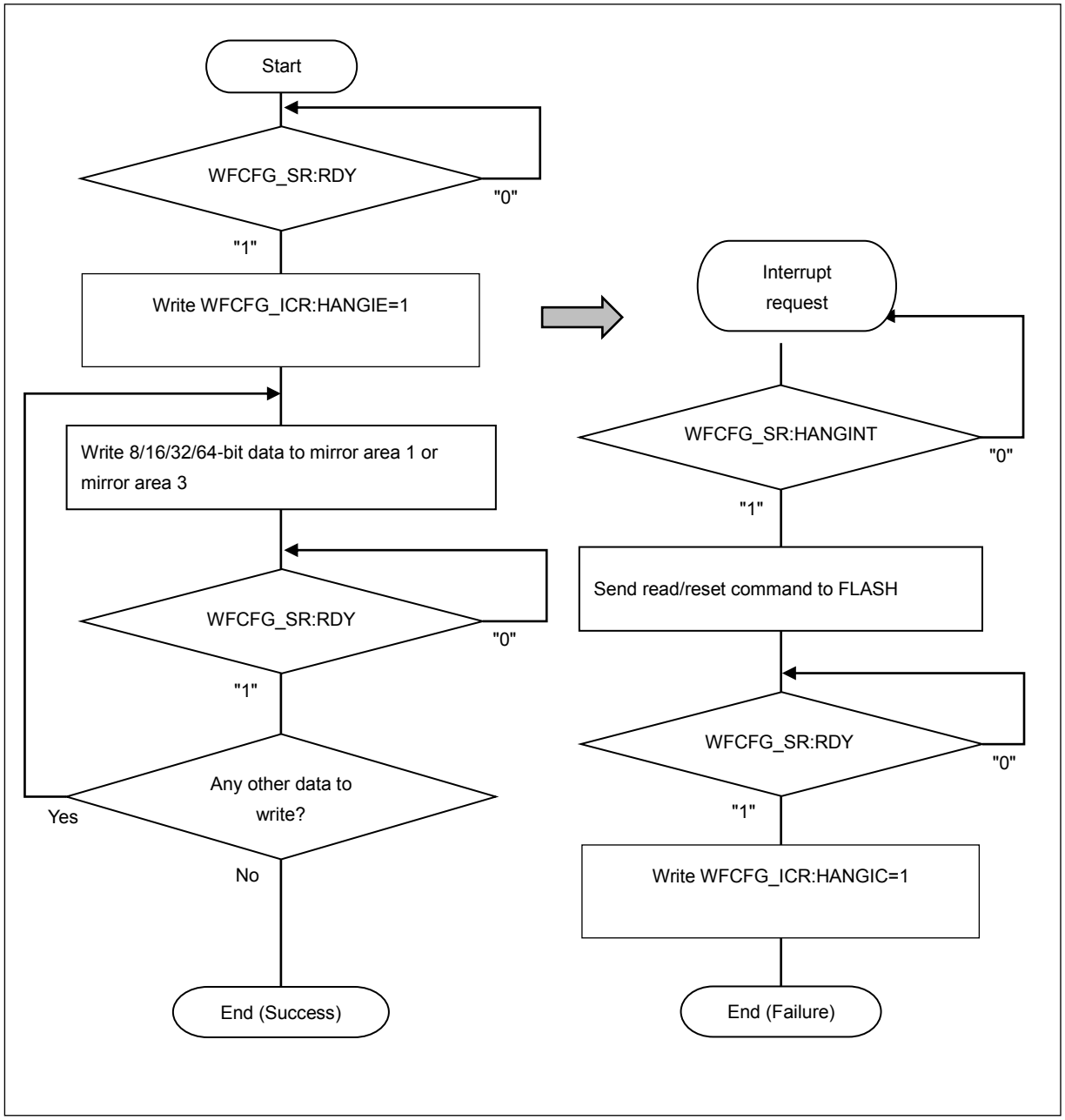
1. Writing with 64 bits when ECC is enabled
   - 1st time: Address=PA[mirror area 1], Data=PD64[63:0]

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.

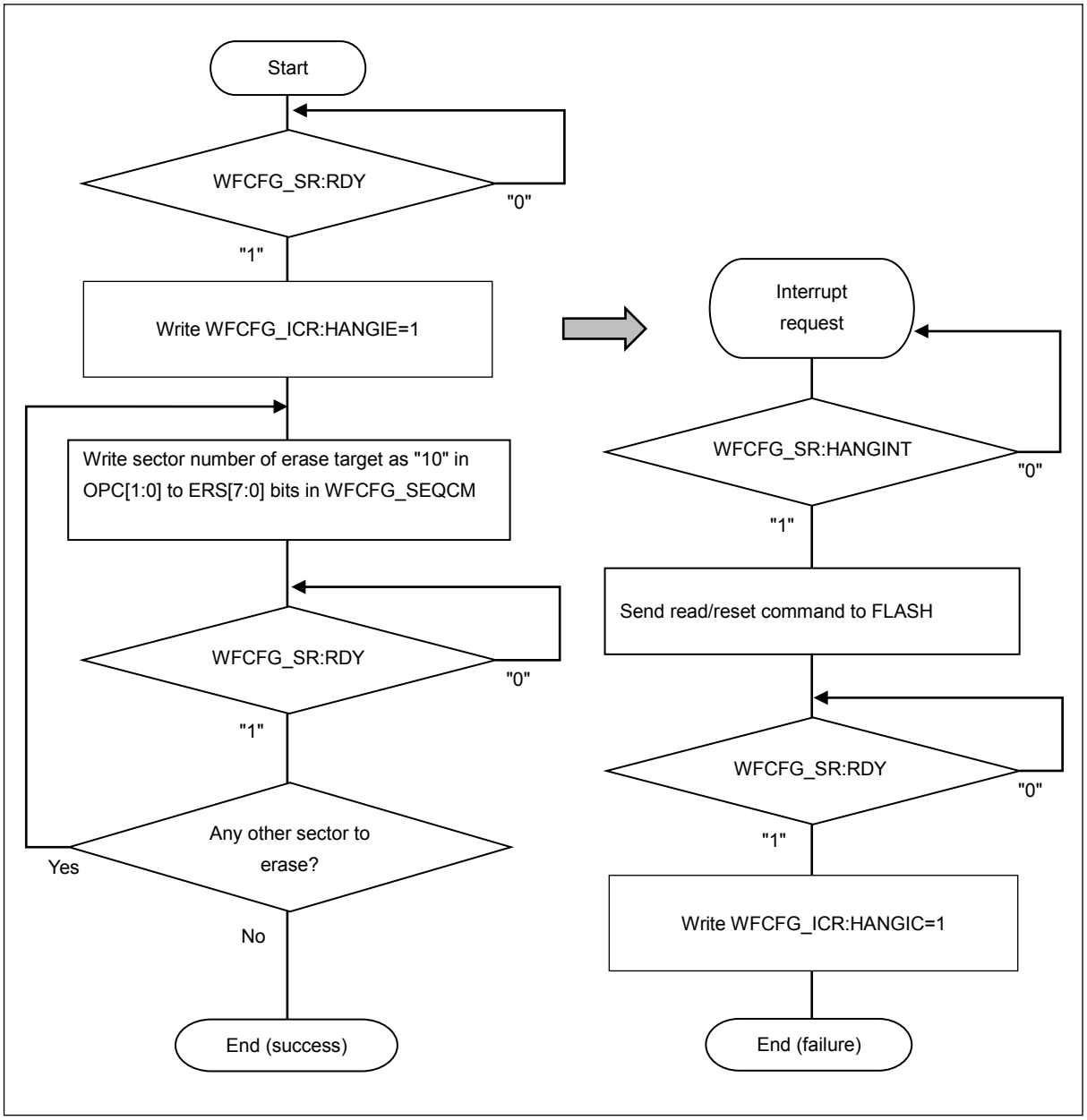
You can verify the completion of the sector erase operation by checking whether the value of the RDY bit in WFCFG_SR is "1". You can check the sector erase execution status with the SERS bit and ESPS bit in WFCFG_SR.

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.

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

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							R1/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

[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	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															

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: Reserved Area Access

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. - The sector protection function specifies the protected sectors.

[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	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															

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.

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.

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.

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

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.)

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.

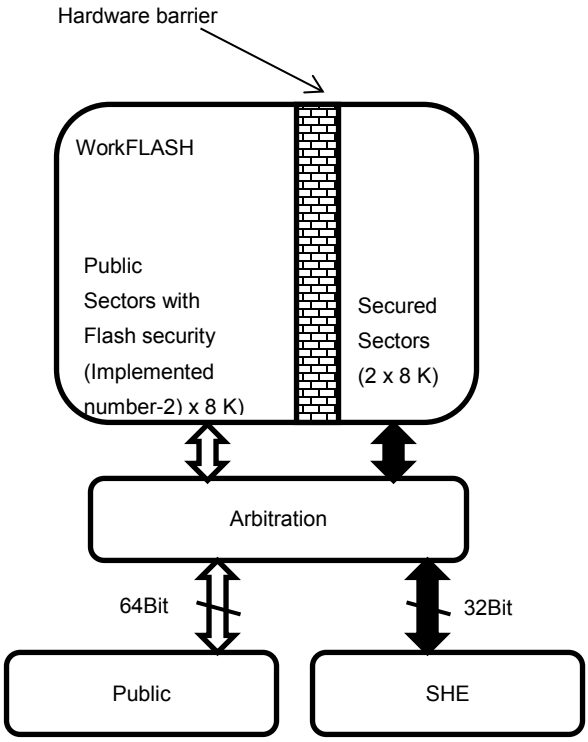
> Example: If the WFIF has valid write access, but the SHEIF has the current access right, the WFIF access is accepted, but its execution has to wait until the end of the SHEIF access.

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 hardware interface.

1. **Overview**
2. **Configuration**
3. **Explanation of Operation**
4. **Setting Procedure Example**
5. **Registers**
6. **Others**

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.

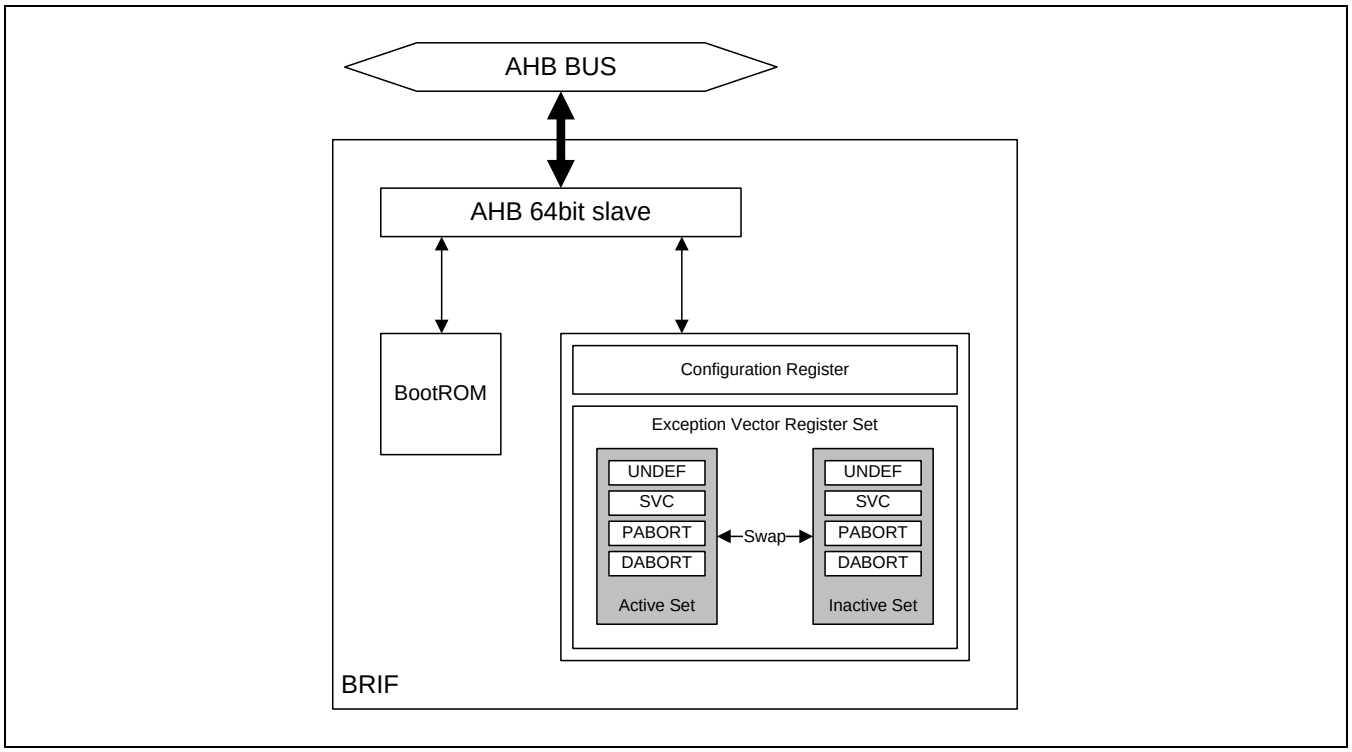
Notes:

- *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.

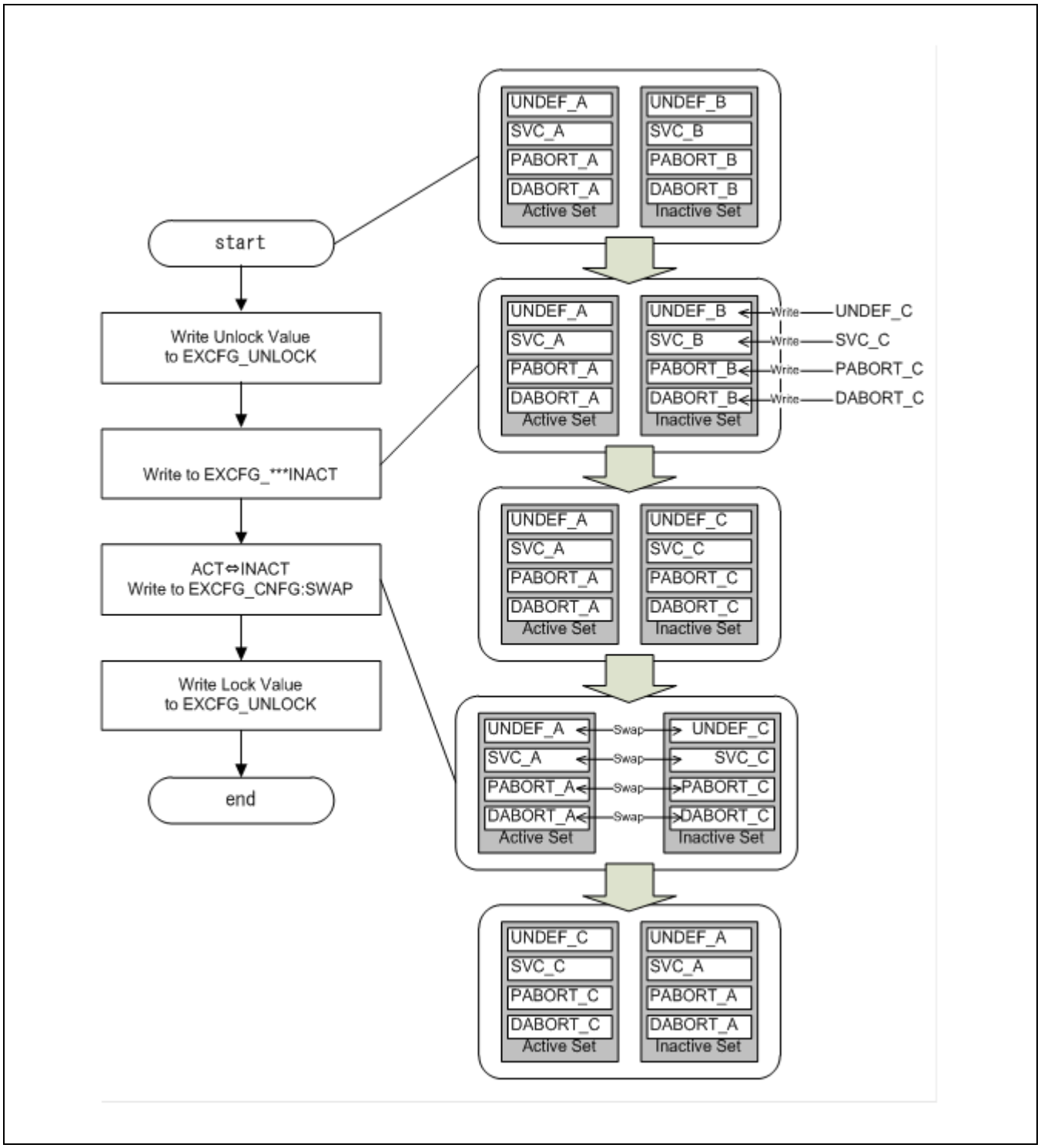
Note:

- *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

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 section Explains the BootROM Software Interface.

1. **Overview**
2. **BootROM Markers**
3. **BootROM Operation**
4. **Notes**

BOOTROMSW-T00P00R01L02-E1-GBB00

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 Connection 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 Connection 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

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 Connection Enable Marker (DDR_DSM)

This marker is used to enable debugger connection.

Register Configurations

Bit	31	0
Field	DSEM [31:0]	
R/W Attribute	R/W	
Protection Attribute	-	
Initial Value	11111111_11111111_11111111_11111111	

Register Functions

[bit31:0] DSEM: Debugger Connection Enable Marker

These bits are used to enable debugger connection. The following are the setting values.

bit31:0	Description
0x59F71234	Debugger connection enabled
Other than above	Debugger connection disabled (initial value)

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.

MK_SER	DDR_DSM	Description
Security enabled	Debugger connection enabled	A 128-bit authentication key is required when a debugger is connected.
Security disabled	Debugger connection enabled	A 128-bit authentication key is not required when a debugger is connected.
Security enabled	Debugger connection disabled	A debugger cannot be connected.
Security disabled	Debugger connection disabled	A 128-bit authentication key is not required when a debugger is connected.

2.5. Debugger Security Key Marker 0 (DDR_DSKM0)

This marker is used to set an authentication key for debugger connection.

Register Configurations

Bit	31	0
Field	DSKM [127:96]	
R/W Attribute	R/W	
Protection Attribute	-	
Initial Value	11111111_11111111_11111111_11111111	

Register Functions

[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.

2.6. Debugger Security Key Marker 1 (DDR_DSKM1)

This marker is used to set an authentication key for debugger connection.

Register Configurations

Bit	31	0
Field	DSKM [95:64]	
R/W Attribute	R/W	
Protection Attribute	-	
Initial Value	11111111_11111111_11111111_11111111	

Register Functions

[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.

2.7. Debugger Security Key Marker 2 (DDR_DSKM2)

This marker is used to set an authentication key for debugger connection.

Register Configurations

Bit	31	0
Field	DSKM [63:32]	
R/W Attribute	R/W	
Protection Attribute	-	
Initial Value	11111111_11111111_11111111_11111111	

Register Functions

[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.

2.8. Debugger Security Key Marker 3 (DDR_DSKM3)

This marker is used to set an authentication key for debugger connection.

Register Configurations

Bit	31	0
Field	DSKM [31:0]	
R/W Attribute	R/W	
Protection Attribute	-	
Initial Value	11111111_11111111_11111111_11111111	

Register Functions

[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.

2.9. SHE Secure Boot Mode Marker (BDR_SBMM)

This marker is used to set a mode of Secure Boot.

Register Configurations

Bit	31	0
Field	SBMM [31:0]	
R/W Attribute	R/W	
Protection Attribute	-	
Initial Value	11111111_11111111_11111111_11111111	

Register Functions

[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.

Register Configurations

Bit	31	0
Field	SBSM [31:0]	
R/W Attribute	R/W	
Protection Attribute	-	
Initial Value	11111111_11111111_11111111_11111111	

Register Functions

[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.

Register Configurations

Bit	31	0
Field	DWEM [31:0]	
R/W Attribute	R/W	
Protection Attribute	-	
Initial Value	11111111_11111111_11111111_11111111	

Register Functions

[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.

Register Configurations

Bit	31	0
Field	ABVM [31:0]	
R/W Attribute	R/W	
Protection Attribute	-	
Initial Value	11111111_11111111_11111111_11111111	

Register Functions

[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.

Register Configurations

Bit	31	0
Field	ABVEM [31:0]	
R/W Attribute	R/W	
Protection Attribute	-	
Initial Value	11111111_11111111_11111111_11111111	

Register Functions

[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).

Register Configurations

Bit	31	30	29	28	27	26	25	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	14	13	12	11	10	9	8
Field	Reserved							
R/W Attribute	R1/W1							
Protection Attribute	-							
Initial Value	11111111_11111111							

Register Functions

[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).

Register Configurations

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	

Register Functions

[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).

Register Configurations

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	

Register Functions

[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).

Register Configurations

Bit	31	0
Field	WDGRUNLLM [31:0]	
R/W Attribute	R/W	
Protection Attribute	-	
Initial Value	11111111_11111111_11111111_11111111	

Register Functions

[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).

Register Configurations

Bit	31	0
Field	WDGRUNULM [31:0]	
R/W Attribute	R/W	
Protection Attribute	-	
Initial Value	11111111_11111111_11111111_11111111	

Register Functions

[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).

Register Configurations

Bit	31	0
Field	WDGPSSLLM [31:0]	
R/W Attribute	R/W	
Protection Attribute	-	
Initial Value	11111111_11111111_11111111_11111111	

Register Functions

[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).

Register Configurations

Bit	31	0
Field	WDGPSSULM [31:0]	
R/W Attribute	R/W	
Protection Attribute	-	
Initial Value	11111111_11111111_11111111_11111111	

Register Functions

[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).

Register Configurations

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	

Register Functions

[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).

Register Configurations

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							

Register Functions

[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).

Register Configurations

Bit	31	0
Field	CEM [31:0]	
R/W Attribute	R/W	
Protection Attribute	-	
Initial Value	11111111_11111111_11111111_11111111	

Register Functions

[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)

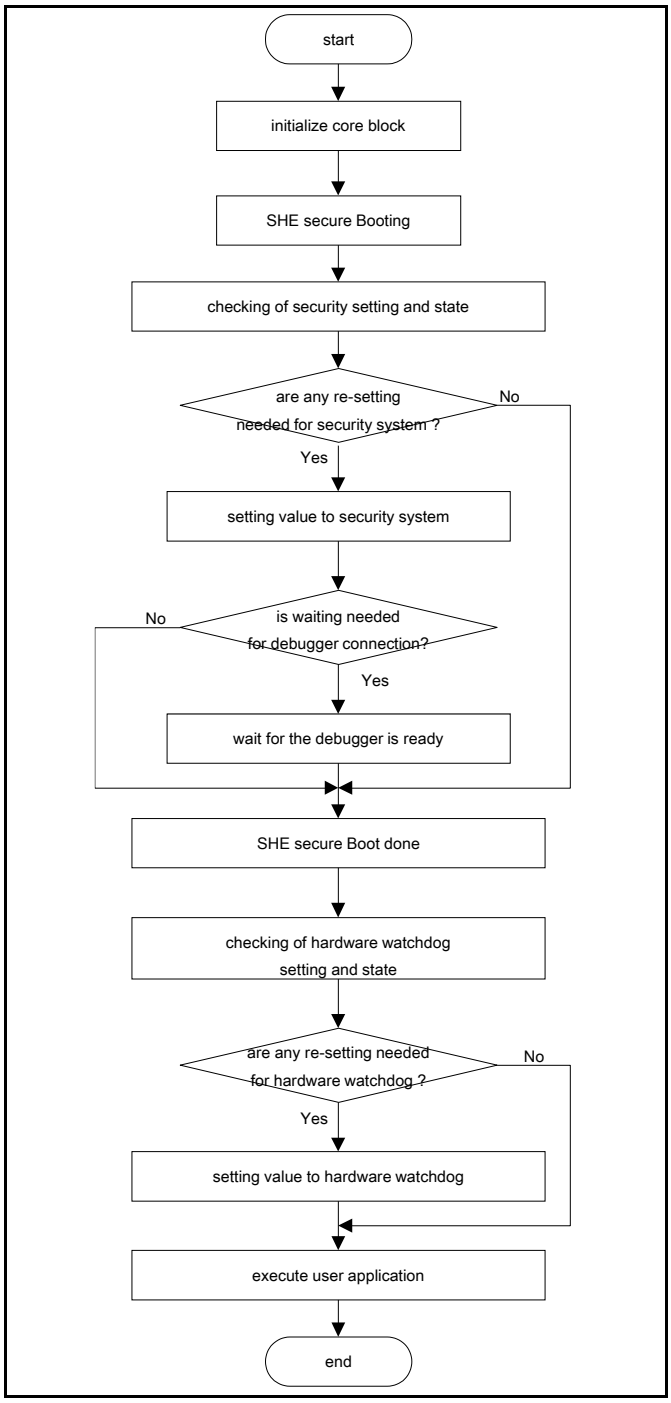
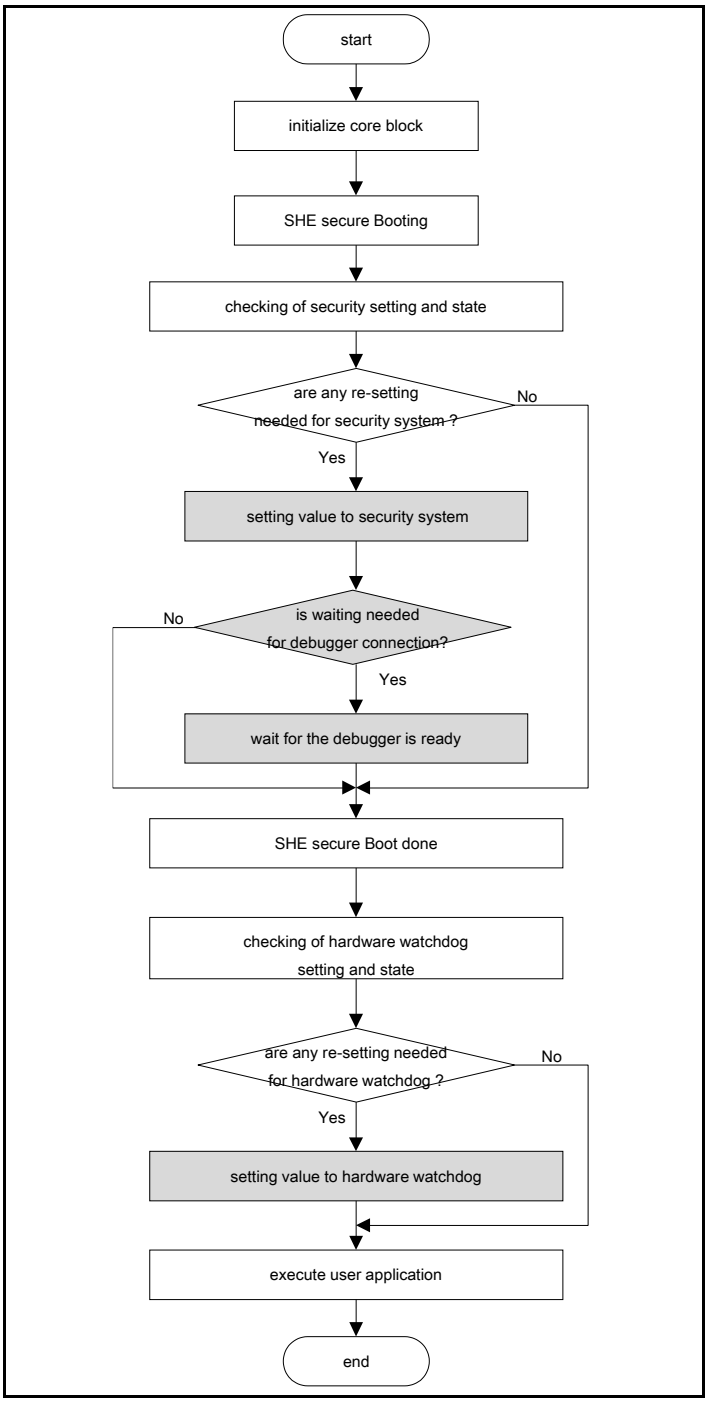
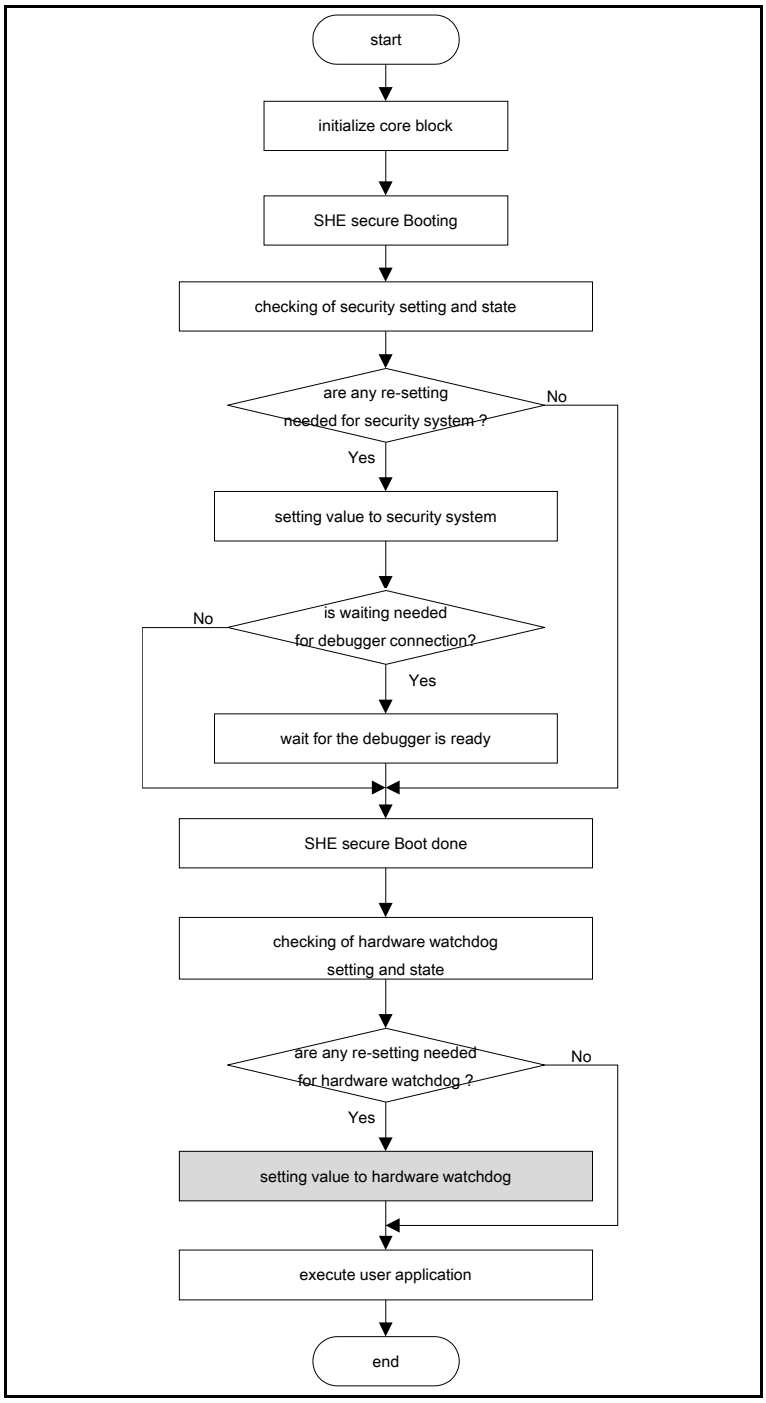


Figure 3-2 User Mode Process Flow (Soft Reset)



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
- 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	Invalid because of FPU is not mounted
Debugging reset	Not set

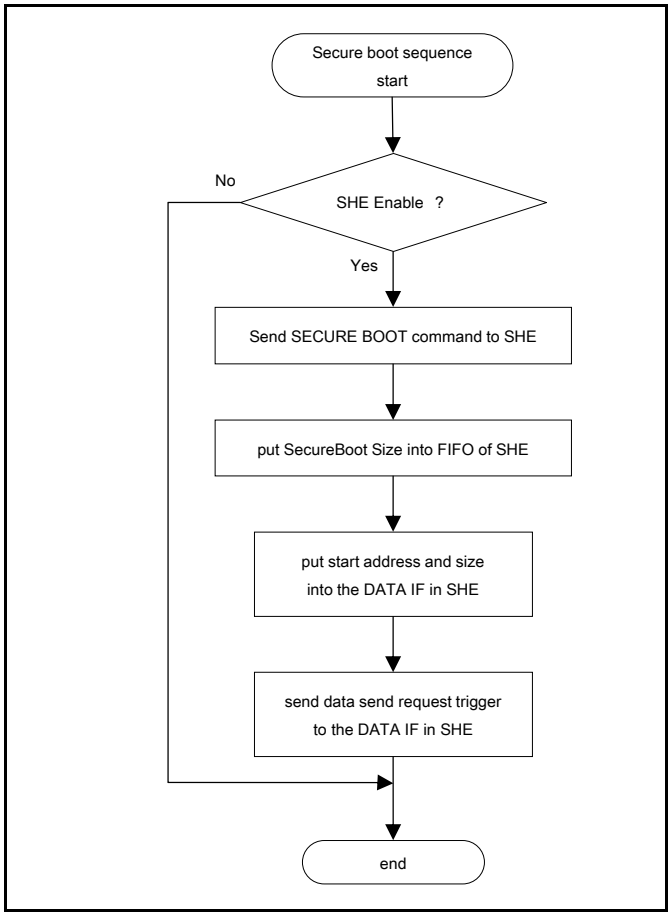
Notes:

- *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.

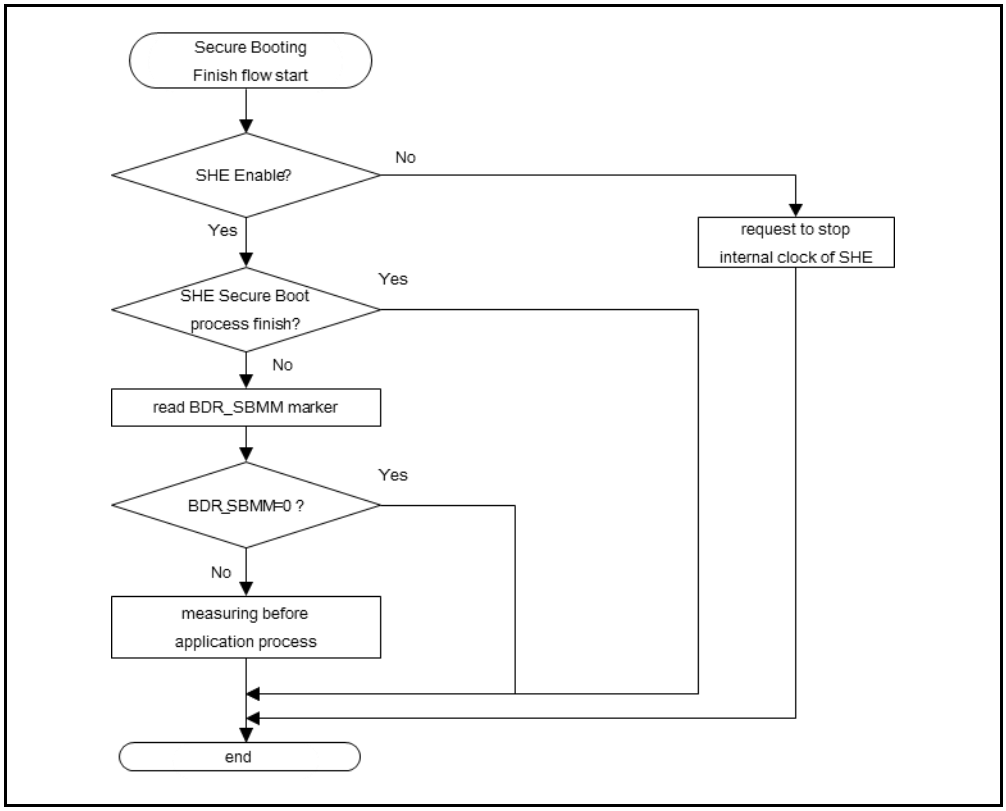
In the DATA IF register of the SHE, the BootROM program sets the Secure Boot Start address determined based on the values of BDR_ABVM and BDR_ABVEM and the Secure Boot size value read from BDR_SBSM. Then, it issues a data transfer request trigger, thus starting the testing of the program area specified 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 Connection 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 R1 to R13 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.

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-T00P00R01L02-E1-GBB00

1. Overview

This section describes the features of the exclusive access memory (EAM) block.

If a conflicting access occurs, Cortex™-R5 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-R5.
- 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-R5.) 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.

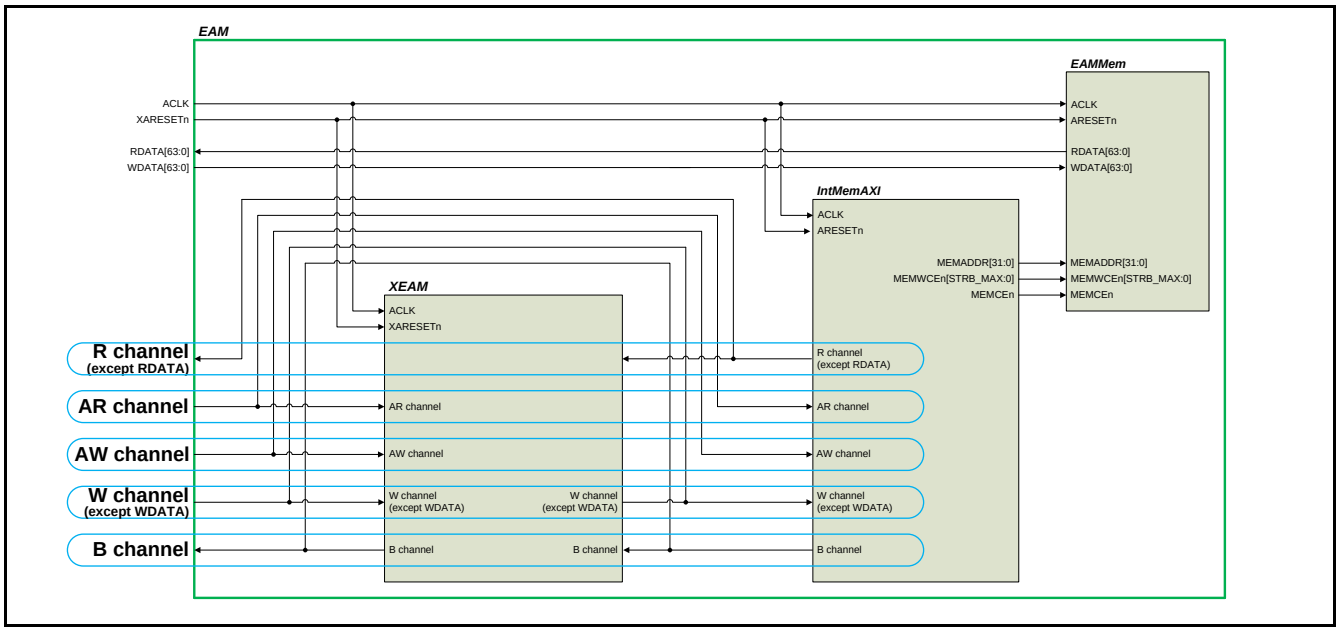
2. Configuration

This section shows the configuration of the exclusive access memory (EAM).

2.1. Block Diagram

This section shows a block diagram 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, "W" = 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	AxID	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	AxID	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	AxID	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	AxID	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	AxID	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	AxID	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	AxID	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	AxID	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	AxID	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	AxID	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	AxID	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	AxID	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	AxID	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	AxID	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	AxID	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	AxID	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	AxID	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	AxID	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	AxID	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	AxID	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	AxID	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	AxID	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	AxID	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	AxID	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	AxID	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	AxID	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	AxID	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

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.

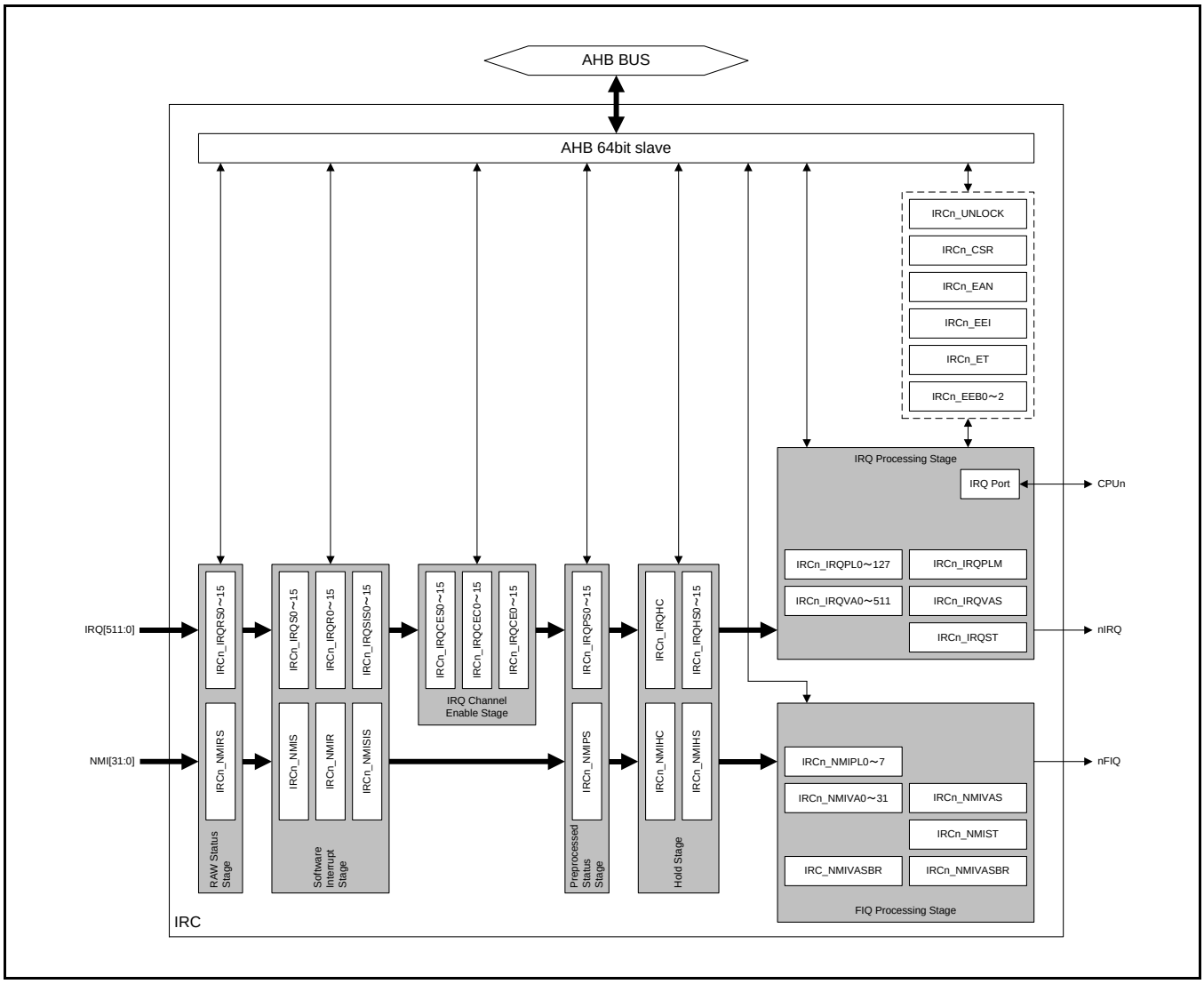
Notes:

- *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 "APPENDIXES: LIST OF INTERRUPT FACTOR AND DMA ACTIVATION FACTOR."*

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.

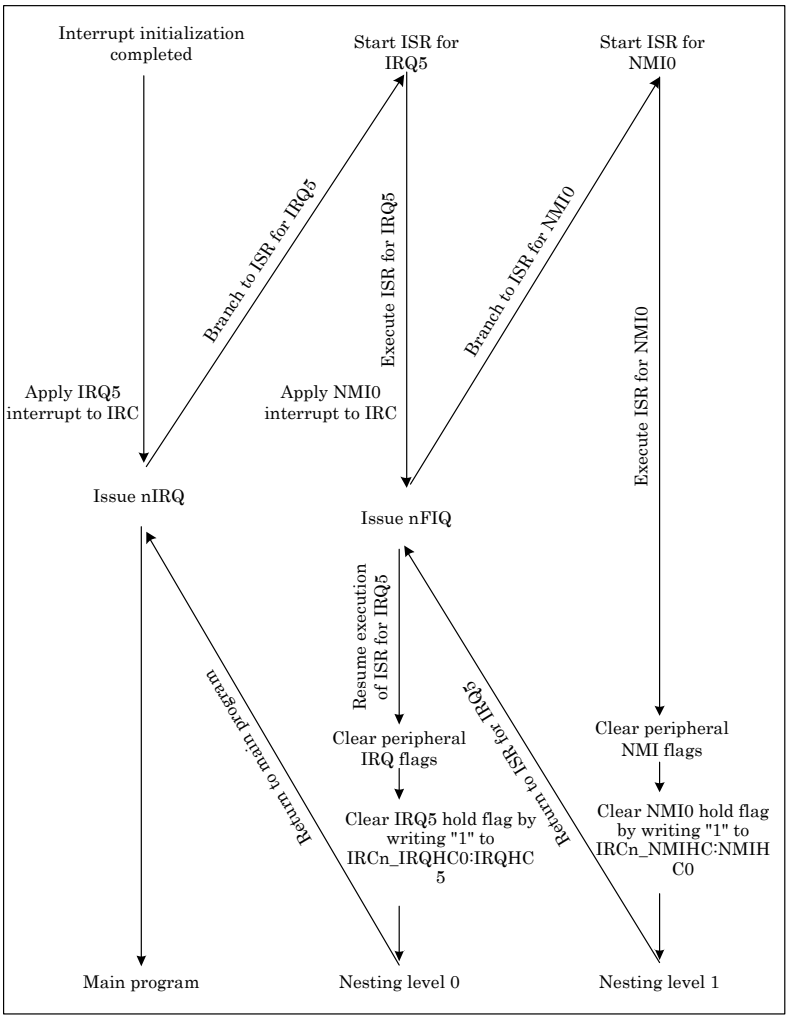
Nesting IRQ and NMI

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



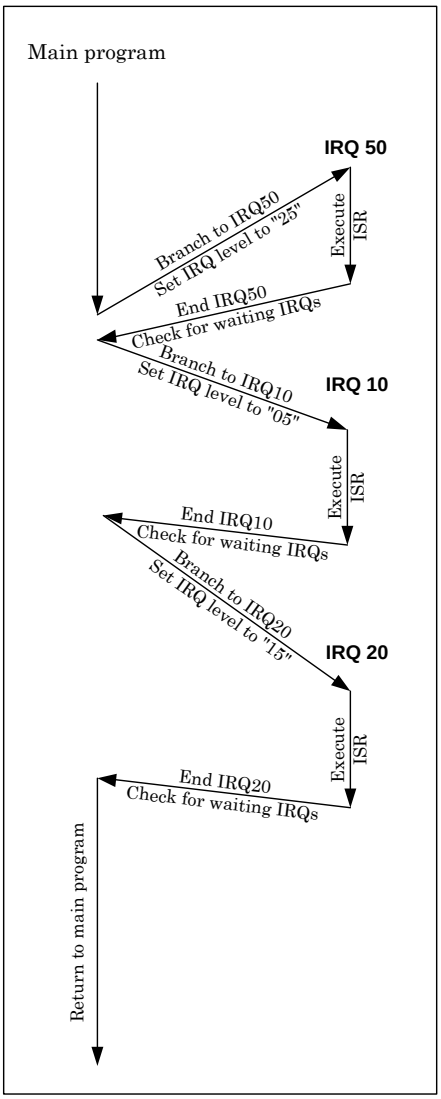
The following paragraphs explain the example shown in Figure 3-1.

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

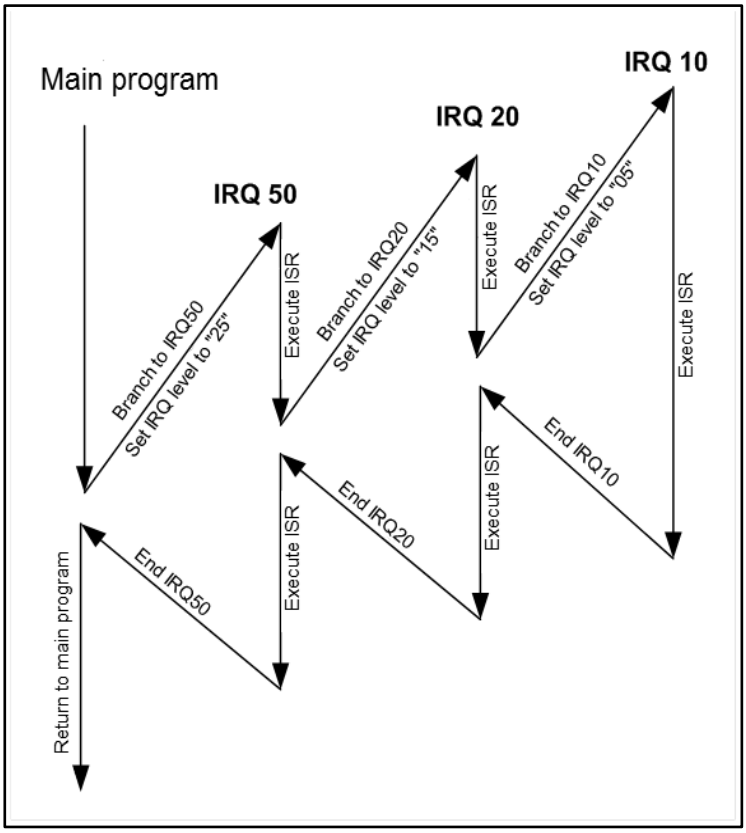
Figure 3-2. The priority levels of IRQ10, 20, and 50 used in this example are assumed to be "5", "15", and "25", respectively.

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 IRS 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.

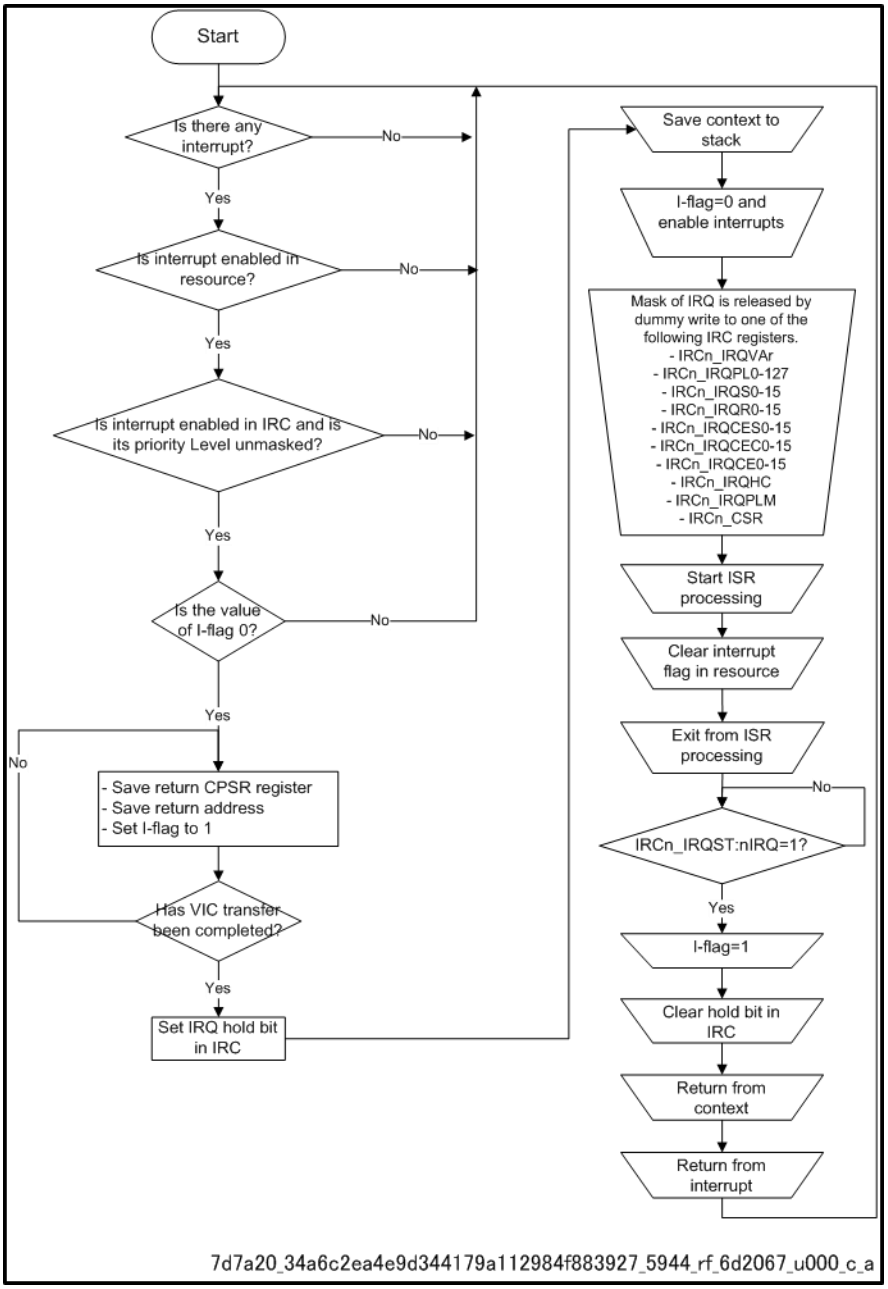
First, IRQ50 occurs. Since there are no waiting IRQs, IRQ50 is executed. At this point, "25" can be read from IRCn_IRQST:IRQPS. During the execution of the ISR for IRQ50, IRQ20 occurs. The ISR for IRQ50 is stopped and the ISR for IRQ20 is executed. At this point, "15" can be read from IRCn_IRQST:IRQPS. 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:

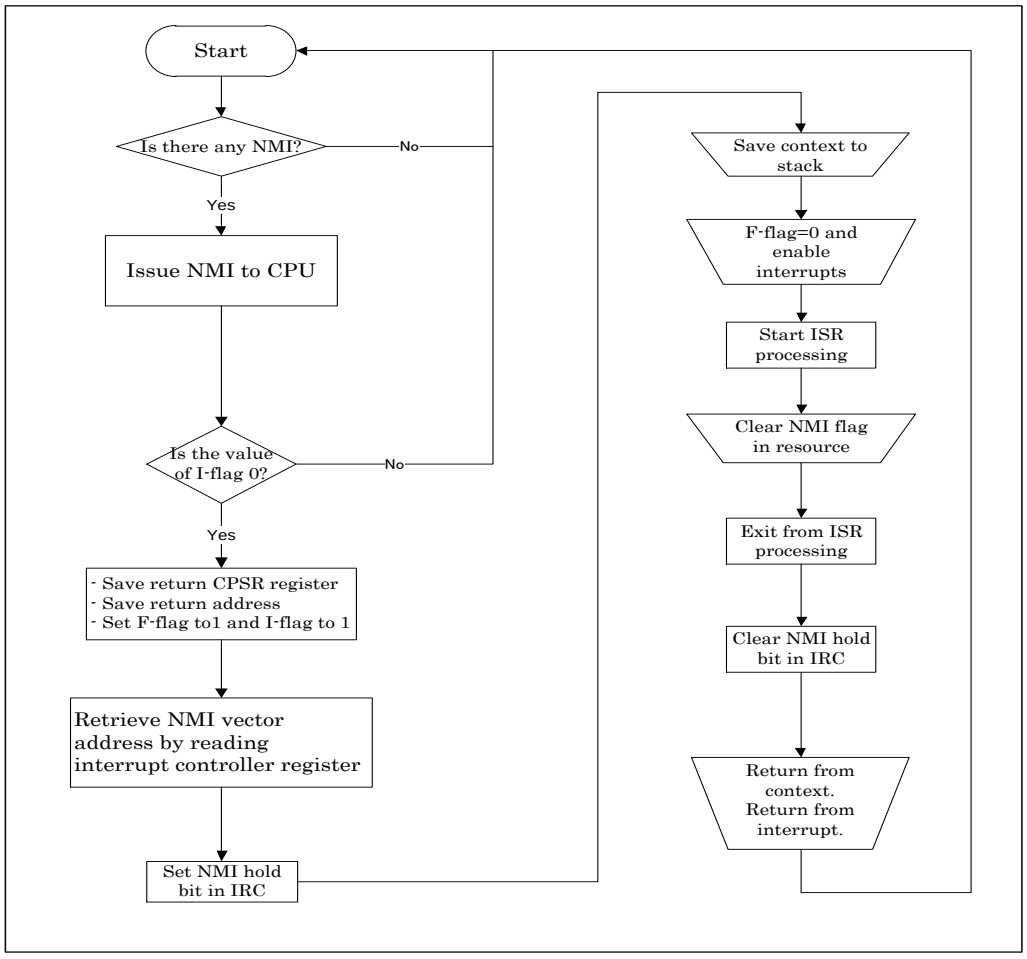
1. The NMI flag which caused the NMI interrupt must be cleared by software during the execution of the ISR. This should be completed as early as possible in the ISR to prevent the interrupt controller from accepting the same NMI flag..
2. The hold bit in the interrupt controller should be cleared during 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 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 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).

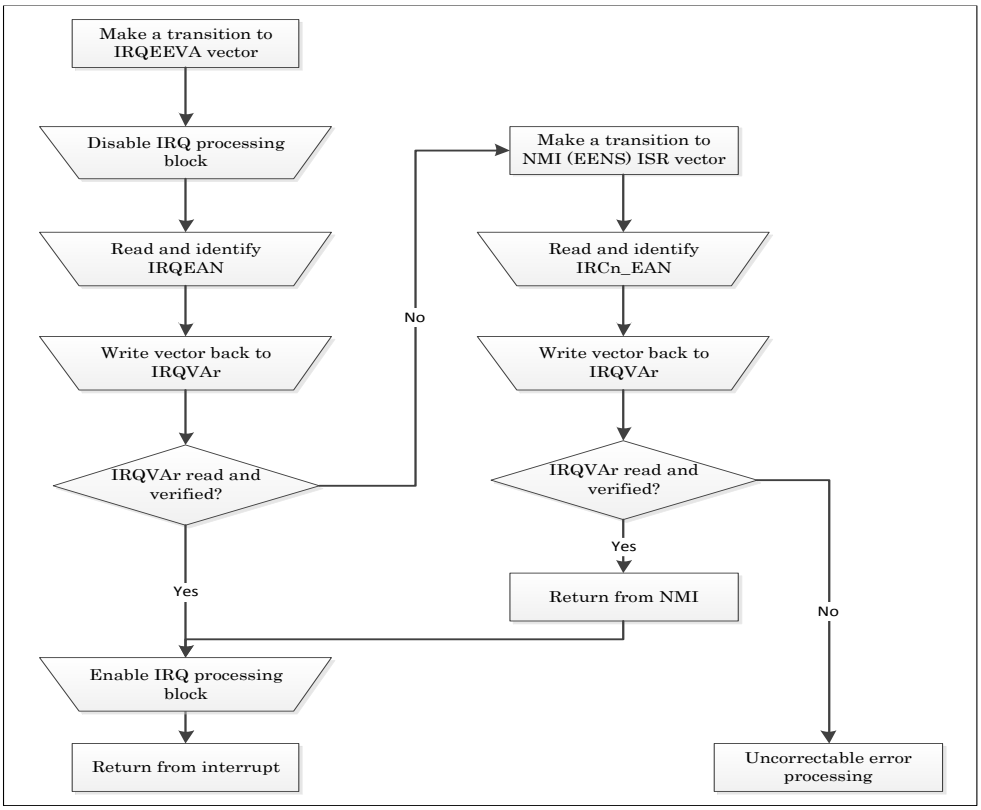
About IRCn_IRQEEVA

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:

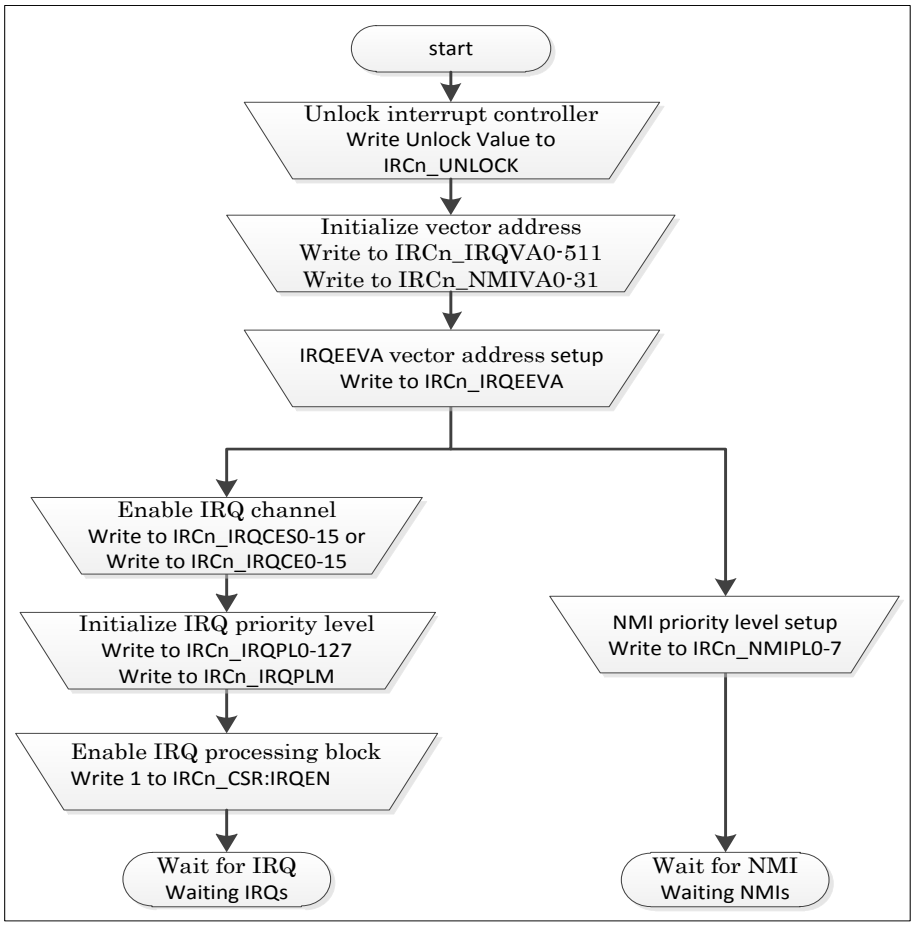
- *If dummy values are not written to any of the registers within the IRC, the generation of a new IRQ interrupt is suppressed until writing is complete. After writing to any of the registers within the IRC, the acceptance of a new IRQ interrupt is resumed.*
- *If the condition for returning a bus error 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

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.

Notes:

- *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.

Notes:

- *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)

Notes:

- 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 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-T00P00R01L02-E1-GBB00

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

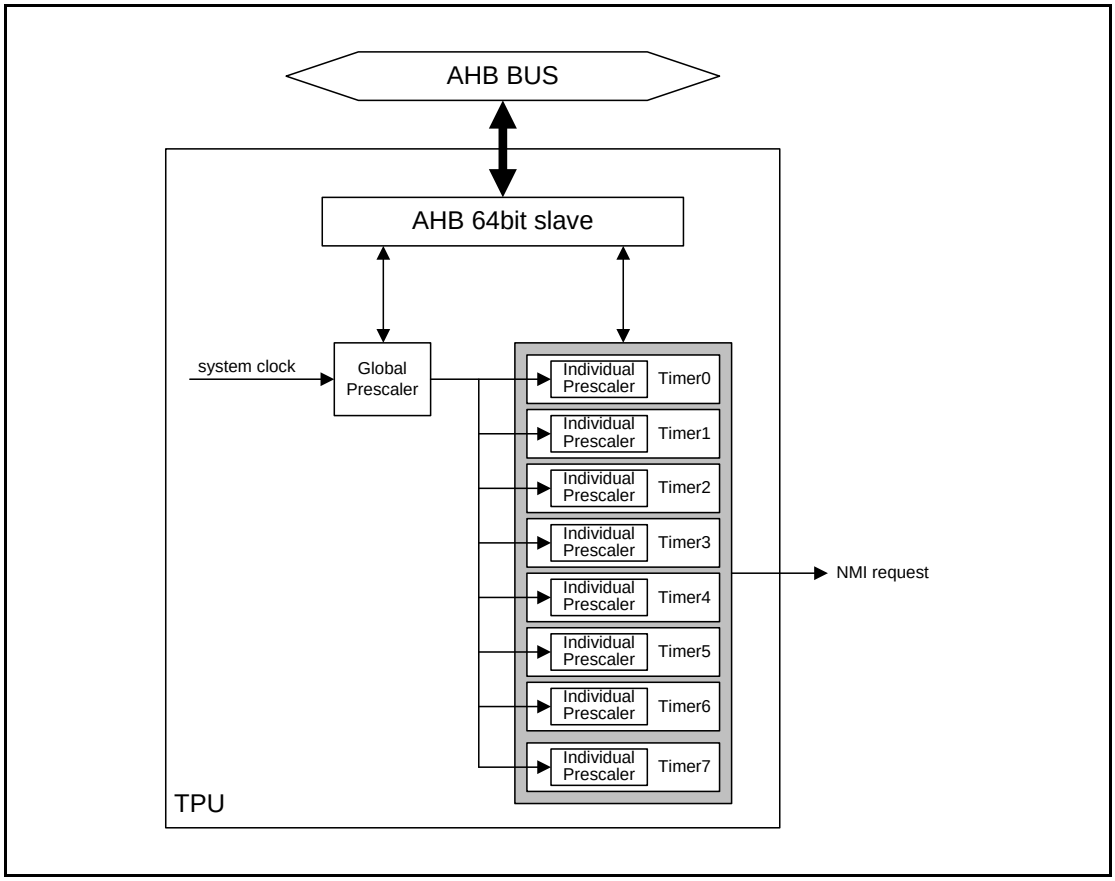
Notes:

- *This chapter sometimes refers to the time protection unit by its abbreviation TPU (Timing 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 PUn_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.

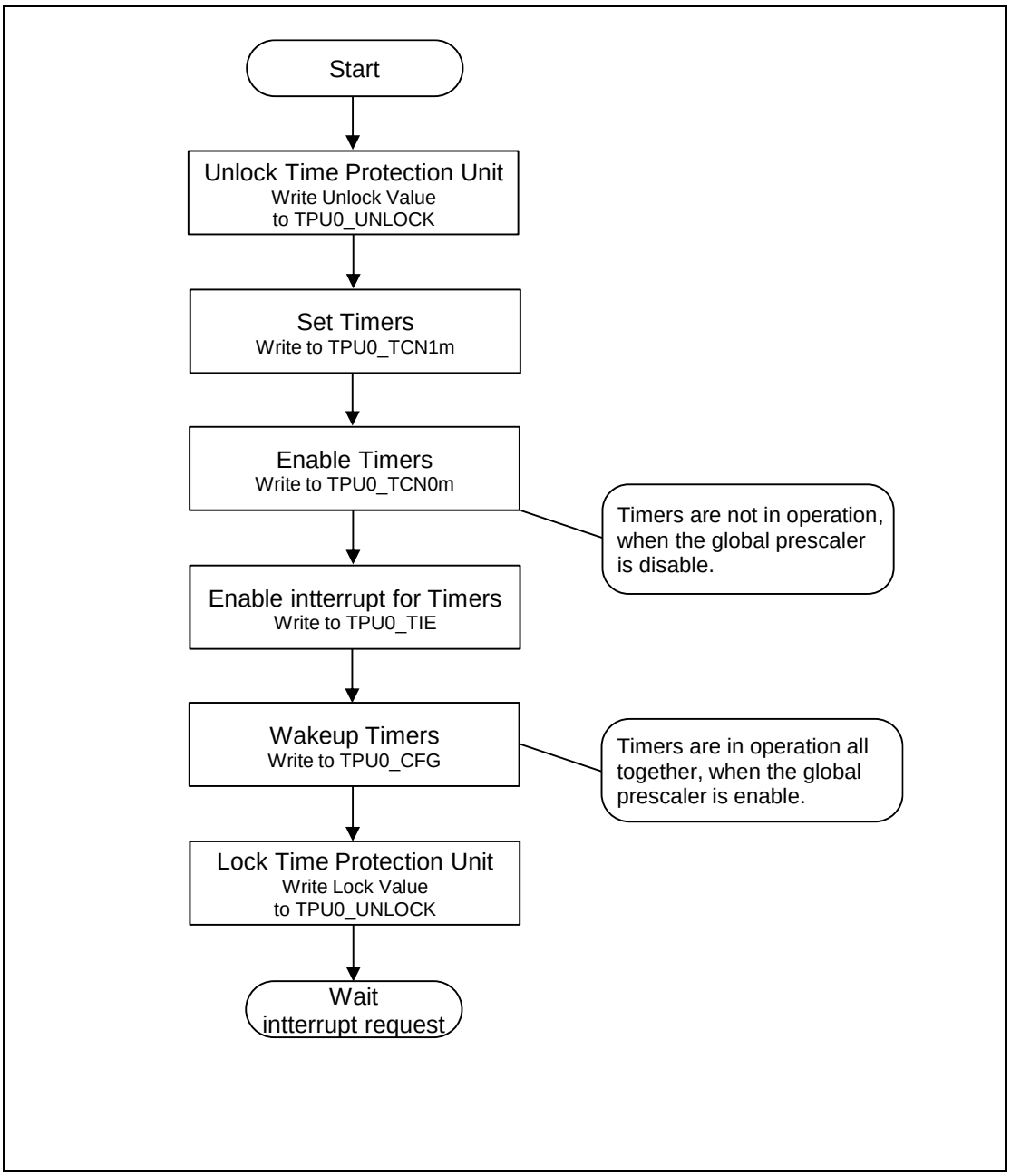
4. Setting Procedure Examples

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.

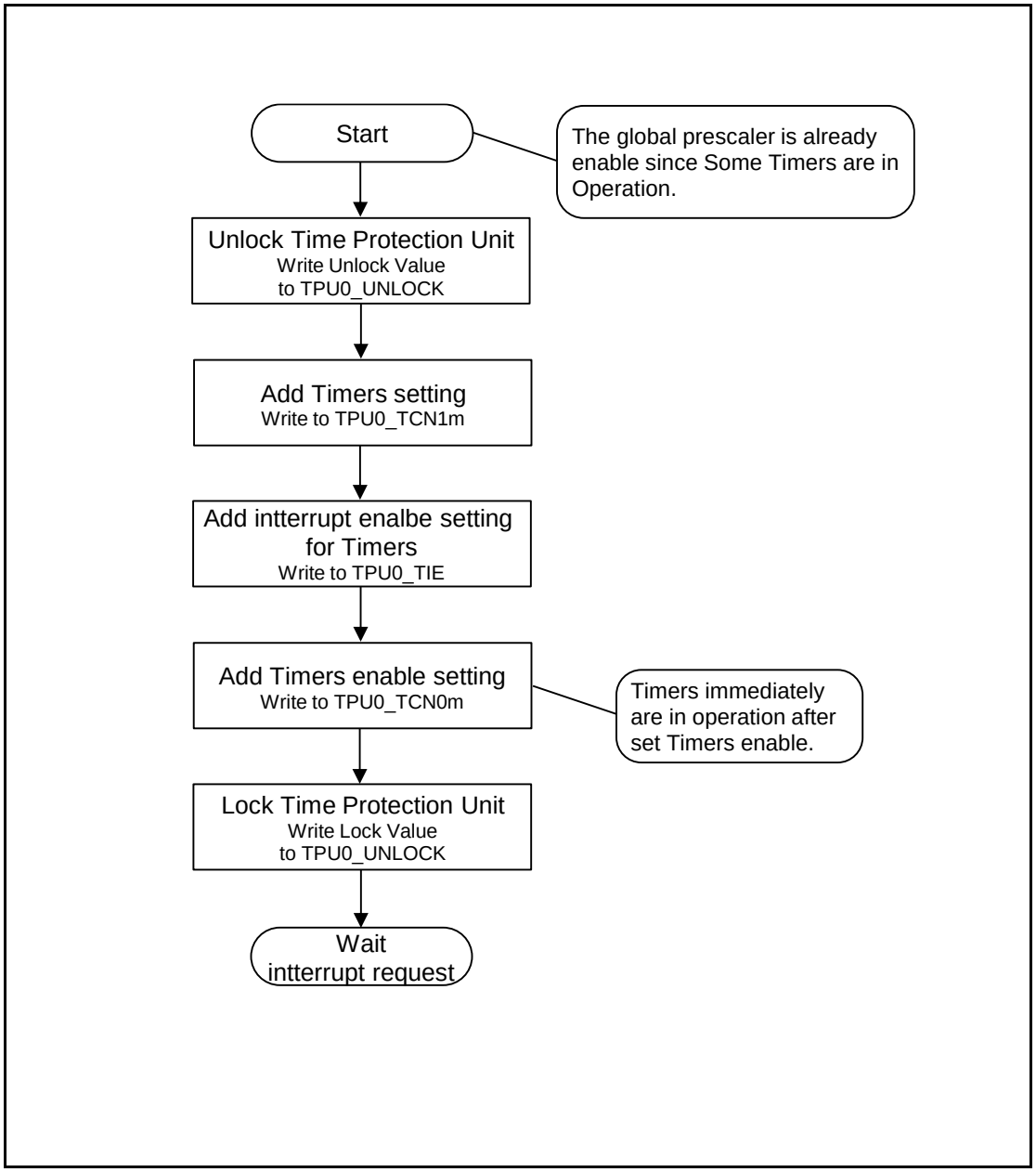
Figure 4-1 Time Protection Unit Setting Flow (When Multiple Timers are Simultaneously Started from the Initial State)



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

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 "3Explanation 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 "3Explanation 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 "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.*

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

SECURITY-T00P00R01L03-E1-GBB01

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.*

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.

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	Non-USERMODE Security= Device Protection
TCFlash Macro Erase	Enabled	Restricted (*1)	(*2)
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)	

(*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).

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.

[A]-->[B]-->[C]

[A] WorkFlash macro erase sector#13 --> sector #12 --> sector#11 --> ... --> sector #0

[B] TCFlash#0 macro erase

[C] Program any data to TCFlash and WorkFlash.

Notes:

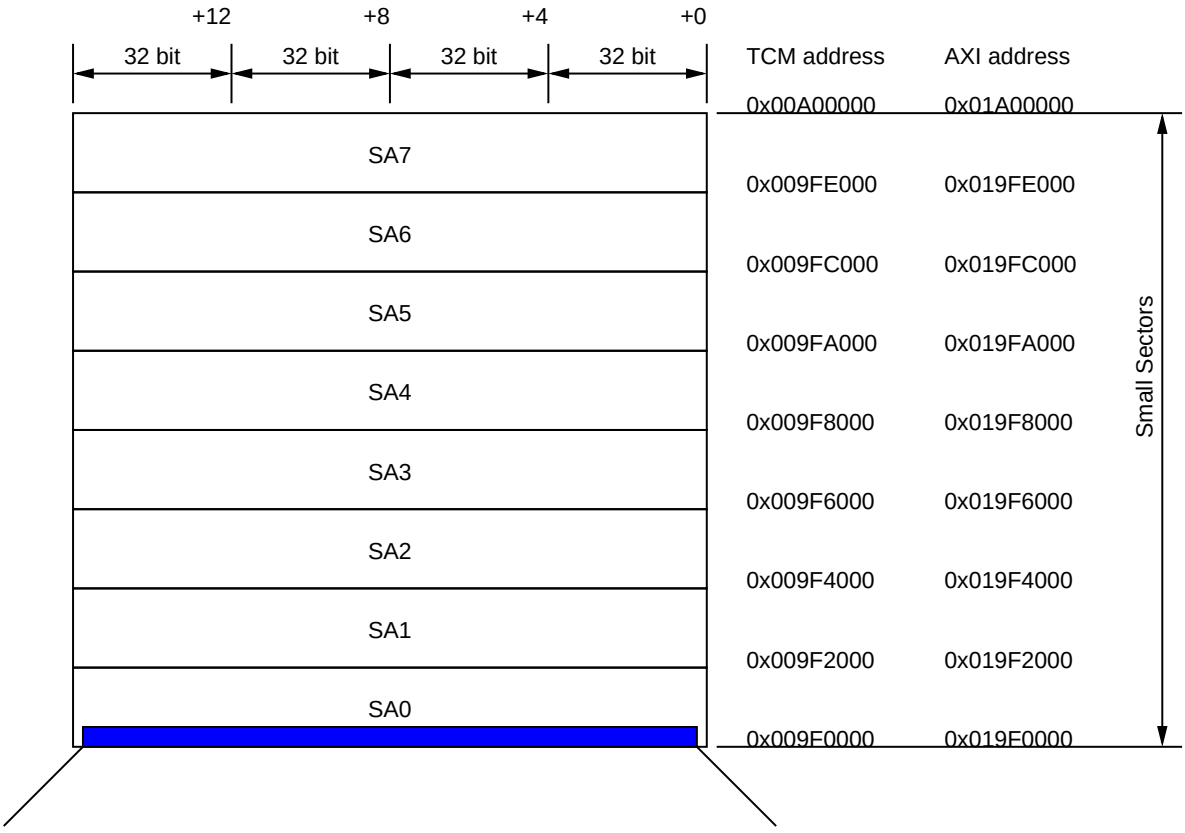
- *If there is a reset between the steps, you have to restart the steps from the first step.*
- *To disable security, program following configuration to the security marker in the last step. {MK_SER : Security = Disabled}*

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.

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”. 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.

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]

[31:0] 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]

[31:0] 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]

[31:0] 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.

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.

[31:0] 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.

[31:0] 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

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 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.

- '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.

Full-chip erase can be forbidden using the MK_CEER marker also for non-USERMODE. It is required to enable it together with 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.

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.

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 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 State 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

Notes:

- *64-bit access is not supported to the registers.*

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

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	1	0 or 1	0 or 1

[31:9] Reserved: These Bits are Not Used. Writing has No effect.

[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.

[7:6] Reserved: These Bits are Not Used. Any Write attempt will Cause a Bus Error.

[5] SFDONE: Security Fetch Done Register

Value	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.

[4] SWPOE: Sector Write Permission Overwrite Enable Register

Value	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.

[3] SECOE: Security Overwrite Enable Register

Value	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.

[2] CEEN: Chip Erase Enable Register

Value	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.

[1] SECSC: Security Scope Register

Value	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.

[0] SECEN: Security Enable Register

Value	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

[31:0] SER: Security Enable Register

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.

After RSTX_PD2H the Security Enable Marker (SEM) 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 SEM 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 MK_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

[31:0] SSR: Security Scope Register

Value	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 (SSM) 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.

Notes:

- *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

[31:0] CEER: Chip Erase Enable Register

Full-chip erase can be forbidden using this marker.

Value	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.

After each RSTX_PD2H the Chip Erase Enable Marker (CEEM) 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.

The chip erase can be triggered by JTAG equipment or 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.*

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

[31:0] SOER: Security Overwrite Enable Register

Value	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 (SOEM) 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. That 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

[31:0] SWPOER: Sector Write Permission Overwrite Enable Register

Value	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 (SWPOEM) 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”. When the TCFCFG0_SWPOER is set to “disabled”, the write access will cause a bus error. That 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: Secure Hardware Extension (SHE)

This Chapter explains the function and operation of the SHE module.

1. **Outline of the Secure Hardware Extension (SHE) Module**
2. **Secure Hardware Extension Registers**
3. **Operation of the Secure Hardware Extension**
4. **Notes on Using SHE**
5. **Enhanced Secure Hardware Extension**
6. **References**

SHE-T00P00R01L03-E1-GBB00

1. Outline of the Secure Hardware Extension (SHE) Module

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 Cypress 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:

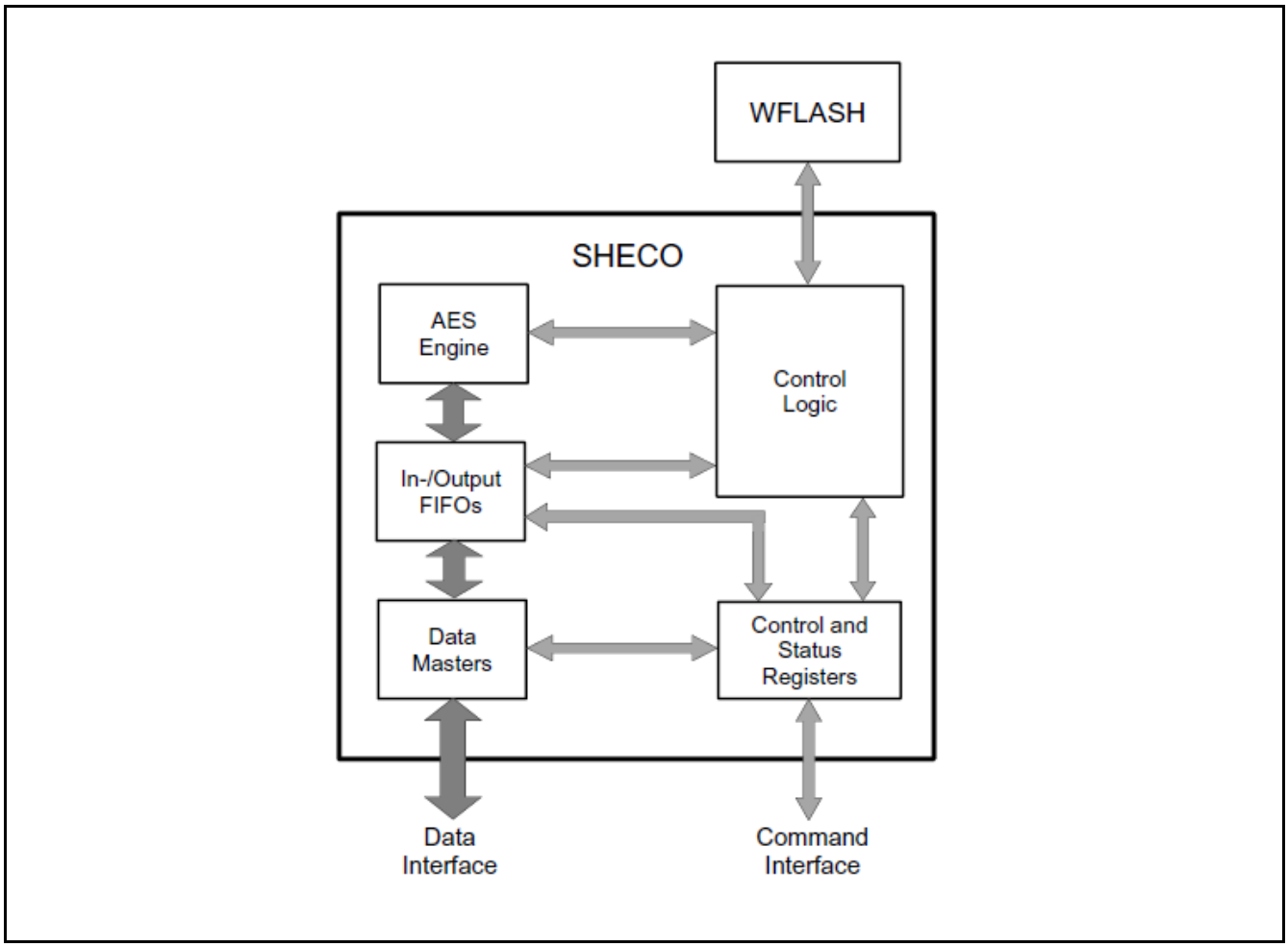
  - *The Miyaguchi-Preneel compression function is not directly accessible by the user and will only be used internally if required during command processing*

  - 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 soft- ware (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

Block Diagram of the SHE Module

Figure 1-1 Block Diagram of the SHE Module



2. 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)

2.1. Configuration Registers for the Command Interface

2.1.1. 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 2-1 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).

Note:

- *Initial value depends on the boot configuration.*

2.1.2. 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 2-2 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.

2.1.3. 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 2-3 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".

2.2. Status Registers for the Command Interface

2.2.1. 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 2-4 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.*

2.2.2. 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 2-6 lists only several often used codes, Refer to chapter 4.1.2 for the complete list of the error codes.

Bit	31	30	29	28	27	26	25	24
Field	CANCEL EXTD [7]	CANCEL EXTD [6]	CANCEL EXTD [5]	CANCELE XTD [4]	CANCEL EXTD [3]	CANCELE XTD [2]	CANCEL EXTD [1]	CANCEL EXTD [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	CANCEL MAIN[7]	CANCEL MAIN[6]	CANCEL MAIN[5]	CANCEL MAIN[4]	CANCEL MAIN[3]	CANCEL MAIN[2]	CANCEL MAIN[1]	CANCEL MAIN[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 2-5 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 contain 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

2.2.3. 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 2-6 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

2.2.4. 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 2-7 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.

2.3. Interrupt Registers for Command Interface

2.3.1. 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 2-8 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).
[15:8]	Reserved	-

Bit Position	Bit Field Name	Bit Description
[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.*

2.3.2. 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 2-9 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.
[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.

2.3.3. 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 2-10 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.

2.4. Configuration Registers for the Data Interface

2.4.1. 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 2-11 Input Channel Master Start Address Register (SHE_IMSTADDR) Bits

Bit Position	Bit Field Name	Bit Description
[31:0]	IMSTADDR	Input Channel Master Start Address

2.4.2. 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 2-12 Output Channel Master Start Address Register (SHE_OMSTADDR) Bits

Bit Position	Bit Field Name	Bit Description
[31:0]	OMSTADDR	Output Channel Master Start Address

2.4.3. 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 2-13 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

2.4.4. 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 2-14 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

2.4.5. 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 2-15 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.

2.4.6. 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 2-16 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.

2.4.7. 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 2-17 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.

2.4.8. 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 2-18 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.

2.4.9. 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 2-19 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 2-20 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

2.5. Status Registers for Data Interface

2.5.1. 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 2-21 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

2.5.2. 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 2-22 Data Master Status Register (SHE_MSTSTATUS) Bits

Bit Position	Bit Field Name	Bit Description
[31:27]	Reserved	-

Bit Position	Bit Field Name	Bit Description
[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 wishes to return an error condition to the originating master '11': DECERR (Decode error) 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 wishes to return an error condition to the originating master '11': DECERR (Decode error) 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.

2.5.3. 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 2-23 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

2.5.4. 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	OMSTERRADDR[31]	OMSTERRADDR[30]	OMSTERRADDR[29]	OMSTERRADDR[28]	OMSTERRADDR[27]	OMSTERRADDR[26]	OMSTERRADDR[25]	OMSTERRADDR[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	OMSTERRADDR[23]	OMSTERRADDR[22]	OMSTERRADDR[21]	OMSTERRADDR[20]	OMSTERRADDR[19]	OMSTERRADDR[18]	OMSTERRADDR[17]	OMSTERRADDR[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	OMSTERRADDR[15]	OMSTERRADDR[14]	OMSTERRADDR[13]	OMSTERRADDR[12]	OMSTERRADDR[11]	OMSTERRADDR[10]	OMSTERRADDR[9]	OMSTERRADDR[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]	OMSTERRA DDR[5]	OMSTERR ADDR[4]	OMSTERR ADDR[3]	OMSTERRA DDR[2]	OMSTERR ADDR[1]	OMSTERRA DDR[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 2-24 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

2.5.5. 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 2-25 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	-

Bit Position	Bit Field Name	Bit Description
[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.
[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.

2.5.6. 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 2-26 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

2.5.7. 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 2-27 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 2-28 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.

2.6. Data Transfer Registers

2.6.1. 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 2-29 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

2.6.2. 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 2-30 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

3. Operation of the Secure Hardware Extension

This chapter describes the operation of the SHE module in detail.

3.1. 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

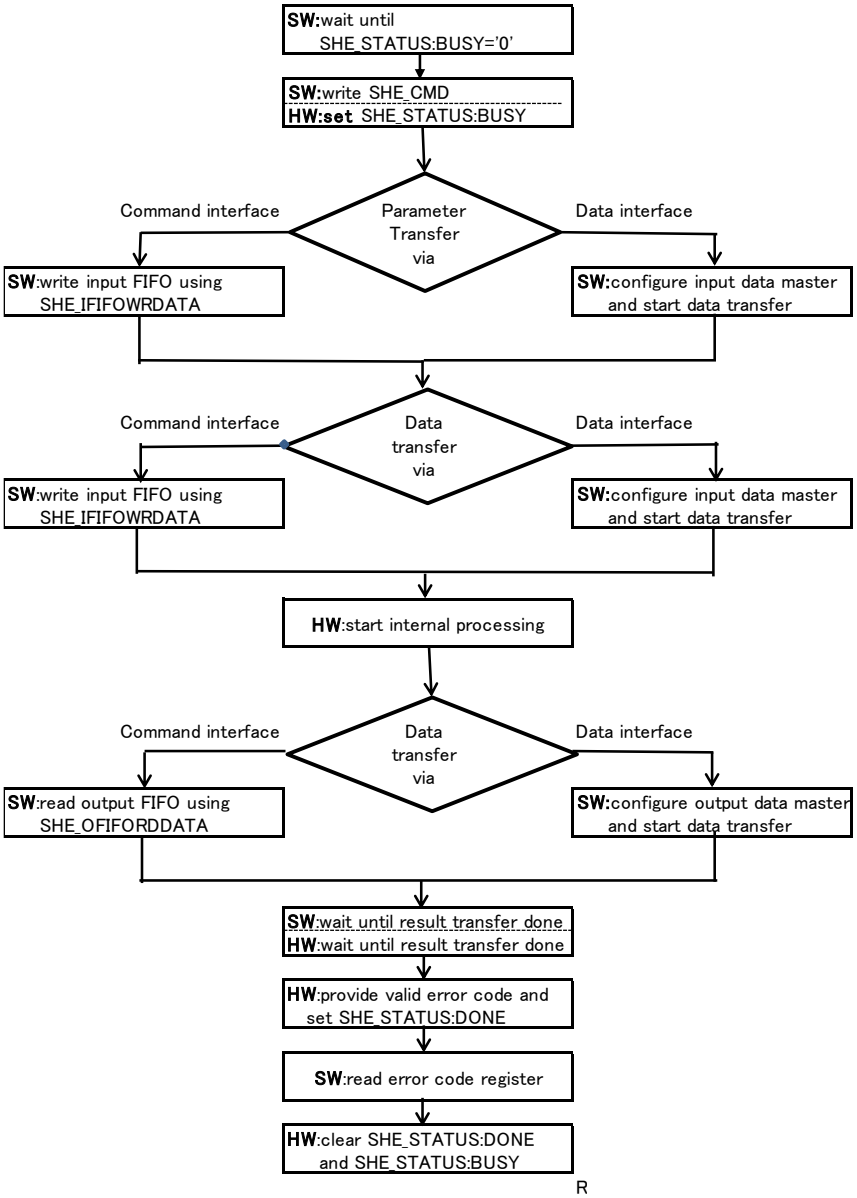
Command Execution

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 1-1 and is described in the following.

Figure 3-1 Flow Chart for the Execution of a Sequential SHE Command



R

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.

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 3-1 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 3-2 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	1	CIPHERTEXT	n x 128
	2	IV	128			
	3	PLAINTEXT	n x 128			
CMD_DEC_ECB	1	KEY_ID	64	1	PLAINTEXT	128
	2	CIPHERTEXT	128			
CMD_DEC_CBC	1	KEY_ID	64	1	PLAINTEXT	n x 128
	2	IV	128			
	3	CIPHERTEXT	n x 128			
CMD_GENERATE_MAC	1	KEY_ID	64	1	MAC	128
	2	MESSAGE_LENGTH	64			
	3	MESSAGE	n^1 x 128			
CMD_VERIFY_MAC	1	KEY_ID	64	1	VERIFICATION_STATUS	64
	2	MESSAGE_LENGTH	64			
	3	MAC_LENGTH	64			
	4	MAC	128			
	5	MESSAGE	n^1 x 128			
CMD_LOAD_KEY	1	M1	128	1	M4	256
	2	M2	256	2	M5	128
	3	M3	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,...}

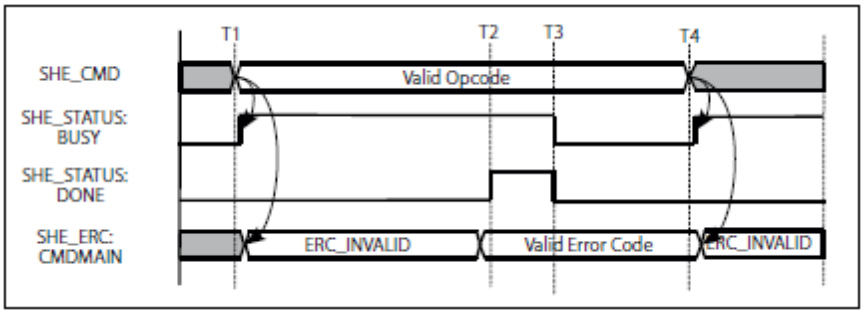
The processed results from SHE are written to the output FIFO in the order given in Table 3-2 above and can be read by the user either through the SHE command interface - see the Output FIFO Read Data Register (SHE_OFIFORDDATA0 to 31) - or data interface using the output data master. Similar to the input 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 3-2 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 3-2 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.

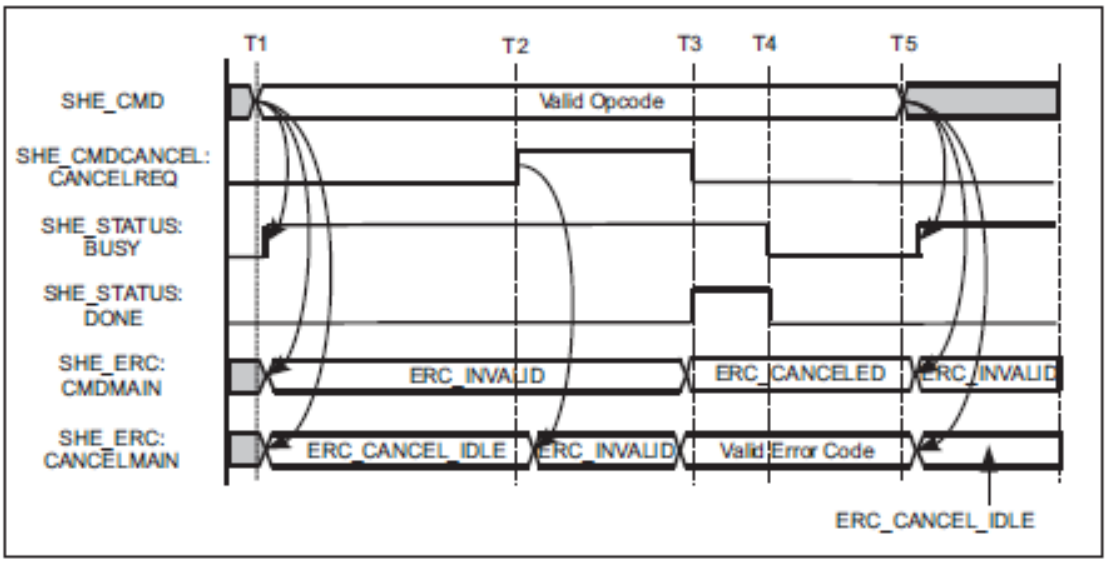
CMD_CANCEL

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 3-3 illustrates the timing flow for the execution of the CMD_CANCEL command.

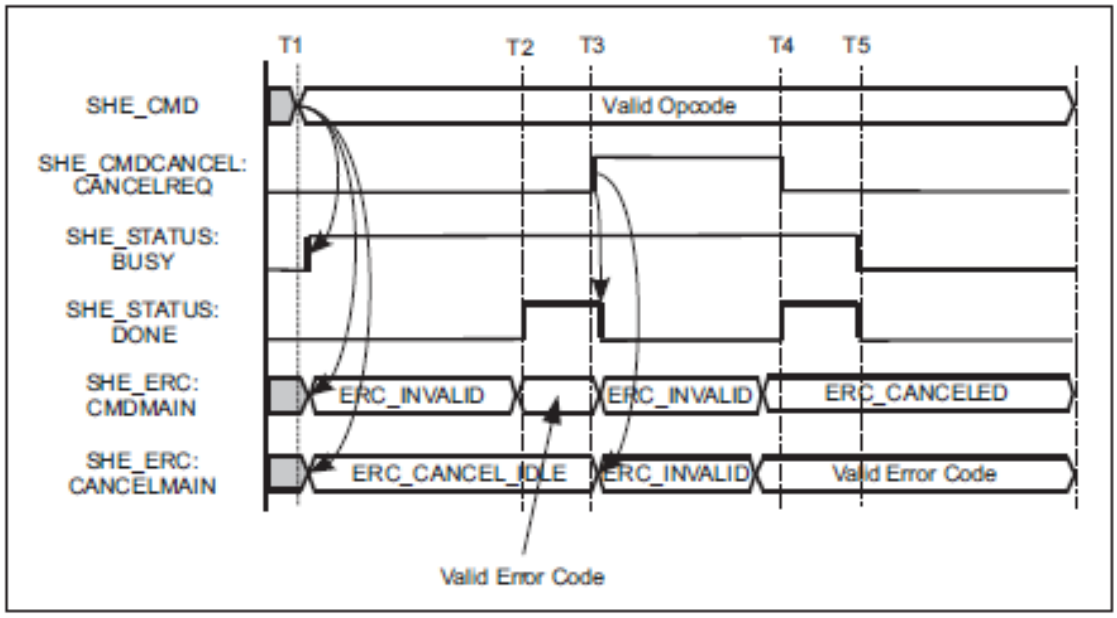
Figure 3-3 Execution of CMD_CANCEL Command



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 3-4 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 3-4 Execution of CMD_CANCEL Command Issued after the Previous Command Has Completed



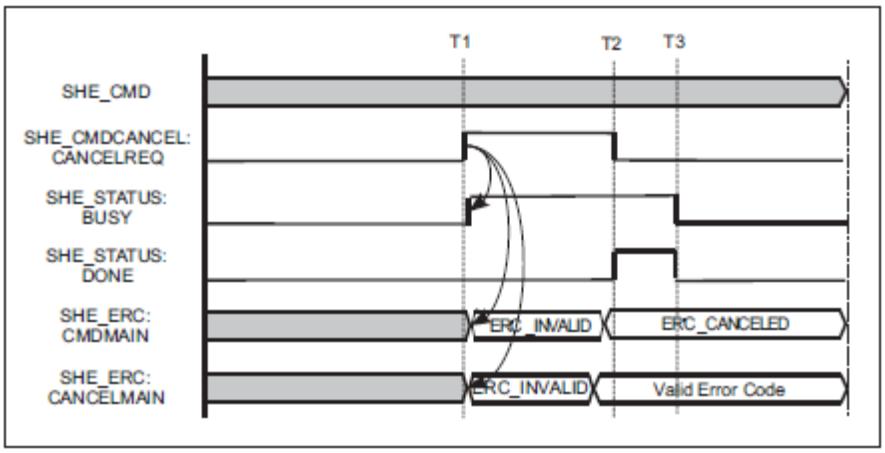
The internal processing of the sequential command, which was started at moment T1, is completed at moment T2. The cancel request is issued at moment T3. Both error codes (SHE_ERC:CMDMAIN and SHE_ERC:CANCELMAIN) are reset and the done flag (SHE_STATUS:DONE) is then cleared by the hardware. Internal processing 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 3-5. 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 3-5 Execution of CMD_CANCEL Command When No Command Execution Is Ongoing.



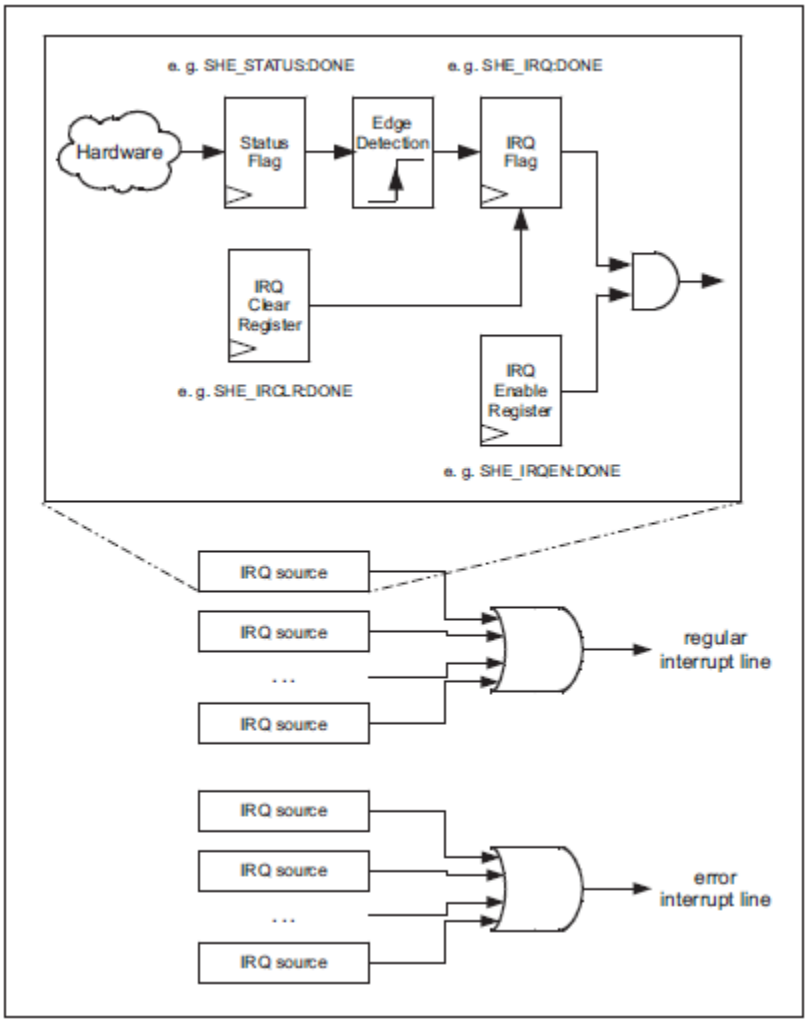
Interrupt request generation

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 3-6 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 3-6 Block Diagram of SHE Interrupt Generation Circuitry



Note:

- *All IRQ source circuits are identical to the one shown in detail.*

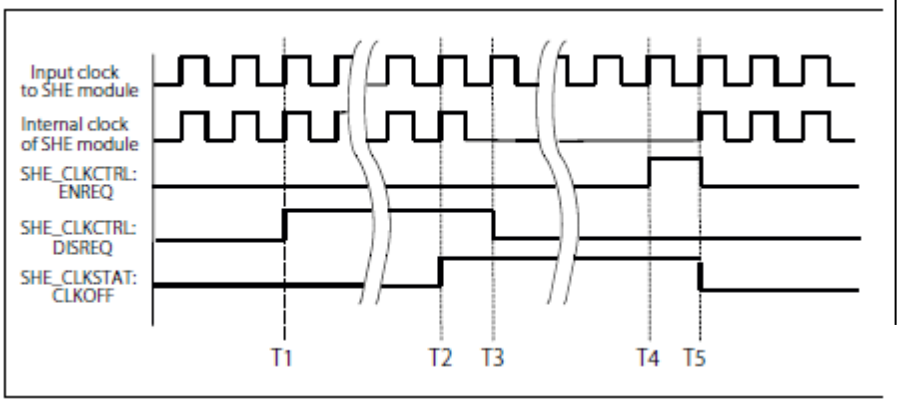
As already mentioned, several interrupt sources are combined and the resulting shared interrupt lines (one for regular and one for error interrupts) are then connected to the interrupt controller.

3.1.1. 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 3- 7.

Figure 3-7 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 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.*

3.1.2. 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.

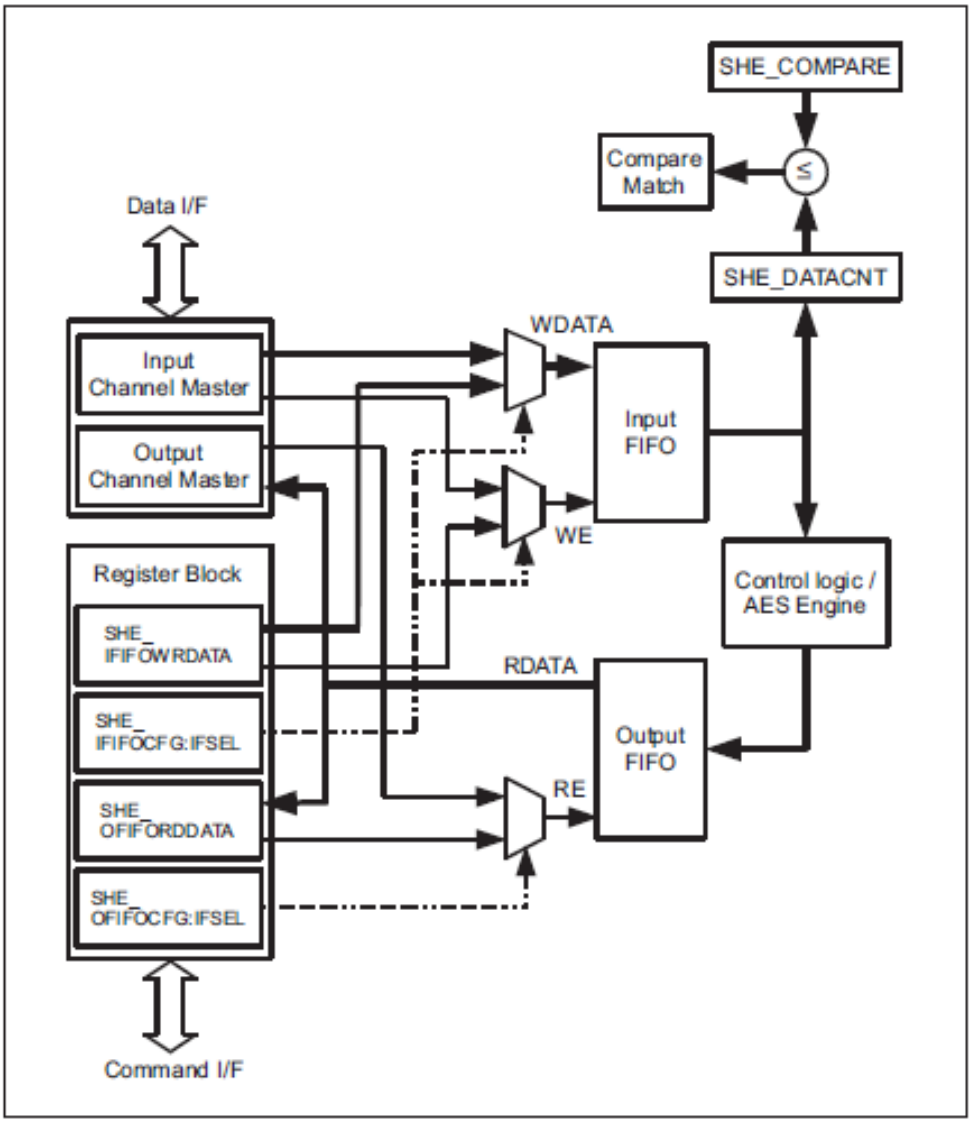
Since the command interface can also be utilized for the transfer of parameters and data used for processing, special registers exist for accessing the write port of the input FIFO (refer to the Input FIFO Write Data Register (SHE_IFIFOWRDATA0 to 31) description) and read port of the output FIFO (refer 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.

3.2. 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 3-8.

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 3-8 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 3-3 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 26 (Memory Protection Unit for the AXI) of the Traveo Cluster Series Hardware Manual.

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.

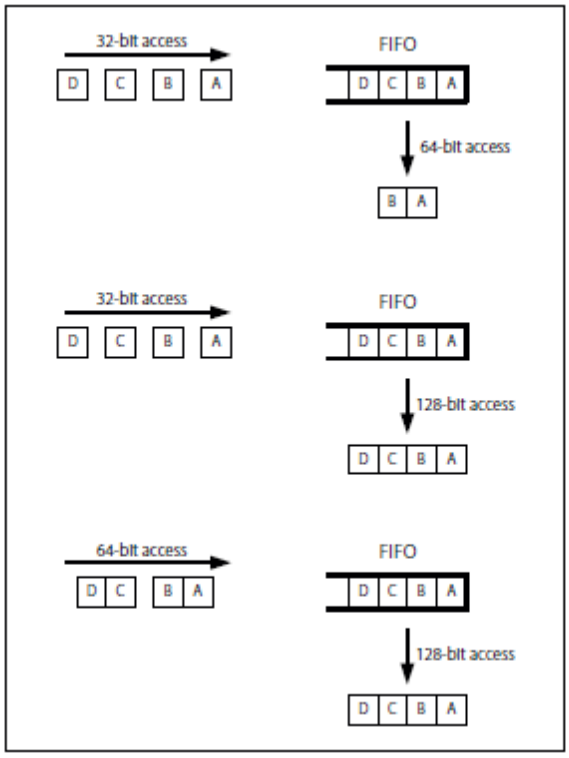
FIFO load information is also taken into account in order to maximize the efficiency of the data transfer. Incoming data transfers are only started if enough free space is available in the input FIFO. Outgoing data transfers are only started if the output FIFO contains enough data for a complete burst.

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.

3.3. 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 synchronization 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 synchronization 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 3-9 must be kept.

Figure 3-9 FIFO Data Ordering



With reference to Figure 3-9:

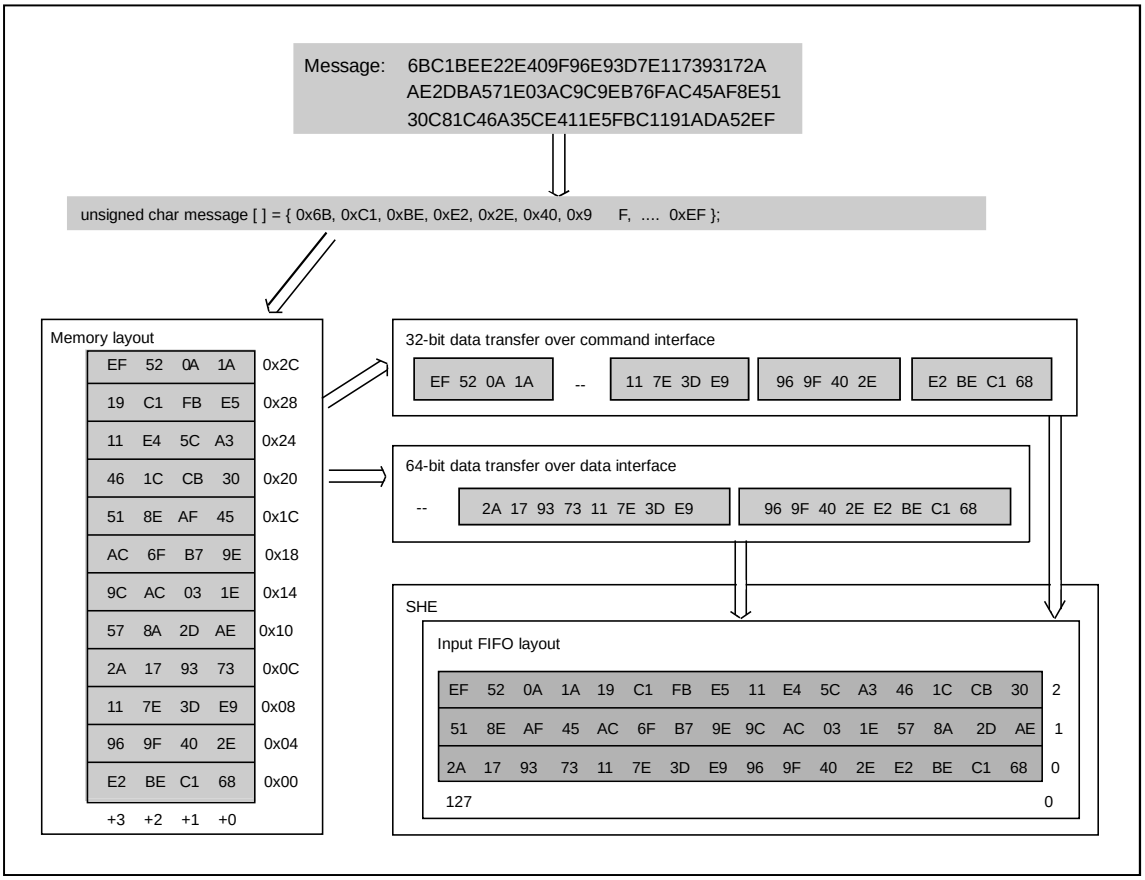
1. If a 64-bit double word has to be written/read using two 32-bit accesses, the least significant word, i.e. bits[31:0], has to be transferred first followed by the most significant word, i.e. bits [63:32].
2. If a 128-bit quad word has to be written/read using four 32-bit accesses, the least significant 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 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 3-10 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 3-10 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.

4. 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.

4.1. 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.*

4.1.1. 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.

4.1.2. SHE Used Error Codes/Used Cancel Codes

Table 4-1 SHE Used Error Codes

Cancel Error	Command Error	Main Error	Description
0x0010	0x0001	ERC_NO_ERROR	Command was normally completed.
See Table 4-2	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.

Cancel Error	Command Error	Main Error	Description
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 forbidden due to invalid device state or the NVM state.
0x0010	0x040D	ERC_GENERAL_ERROR	Unknown command request code.
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.
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.

Cancel Error	Command Error	Main Error	Description
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	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)

Cancel Error	Command Error	Main Error	Description
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 4-2 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.

5. 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.

5.1. 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 Table 4-3.

Table 5-1 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].*

5.2. 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.

VERIFY_ONLY Flag	KEY_USAGE Flag	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
  $M_2 = ENC_{CBC,K1,IV=0}(C_{ID}'|F_{ID}'|"0...0"_{94}|K_{ID}')$

5.3. 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 5-3 together with the opcode value which is required to start the execution through the SHE_CMD register.

Table 5-2 ESHE Commands

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 5-4.

Table 5-3 Parameter Order for ESHE Commands

Command	Parameter			Return Value		
	Seq. No	Name	Size [Bits]	Seq. No	Name	Size [Bits]
CMD ENC ECB ESHE	1 2	KEY ID[1] PLAINTEXT	64 128	 1	 CIPHERTEXT	 128
CMD ENC CBC ESHE	1 2 3	KEY ID1 IV PLAINTEXT	64 128 n x 128	 1	 CIPHERTEXT	 n x 128
CMD DEC ECB ESHE	1 2	KEY ID1 CIPHERTEXT	64 128	 1	 PLAINTEXT	 128
CMD DEC CBC ESHE	1 2 3	KEY ID1 IV CIPHERTEXT	64 128 n x 128	 1	 PLAINTEXT	 n x 128
CMD GENERATE MAC ESHE	1 2 3	KEY ID1 MESSAGE LENGTH MESSAGE	64 64 n2 x 128	 1	 MAC	 128
CMD VERIFY MAC ESHE	1 2 3 4 5	KEY ID1 MESSAGE LENGTH MAC LENGTH MAC MESSAGE	64 64 64 128 n2 x 128	 1	 VERIFICATION STATUS	 64
CMD LOAD KEY ESHE	1 2 3	M1 M2 M3	128 256 128	 1 2	 M4 M5	 256 128

[1] : valid values: 0x04, …, 0xD, which corresponds to KEY_11, …, KEY_20

[2] : n = ceil[3] (MESSAGE_LENGTH / 128)

[3] : ceil(x) = min{n∈Z_1 | n ≥ x}, Z_1 = {1,2,3,...}

5.3.1. 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 4-6.

Table 5-4 Parameter Order for ESHE Commands

Key to update	Authentication key	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_KEY	KEY_<n>	0x4..0xD	MASTER_ECU_KEY	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|$ENC_{ECB,K3}(C_{ID})$

$M_5 = CMAC_{K4}(M_4)$

6. 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 26: 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-T00P00R01L03-E1-GBB00

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.

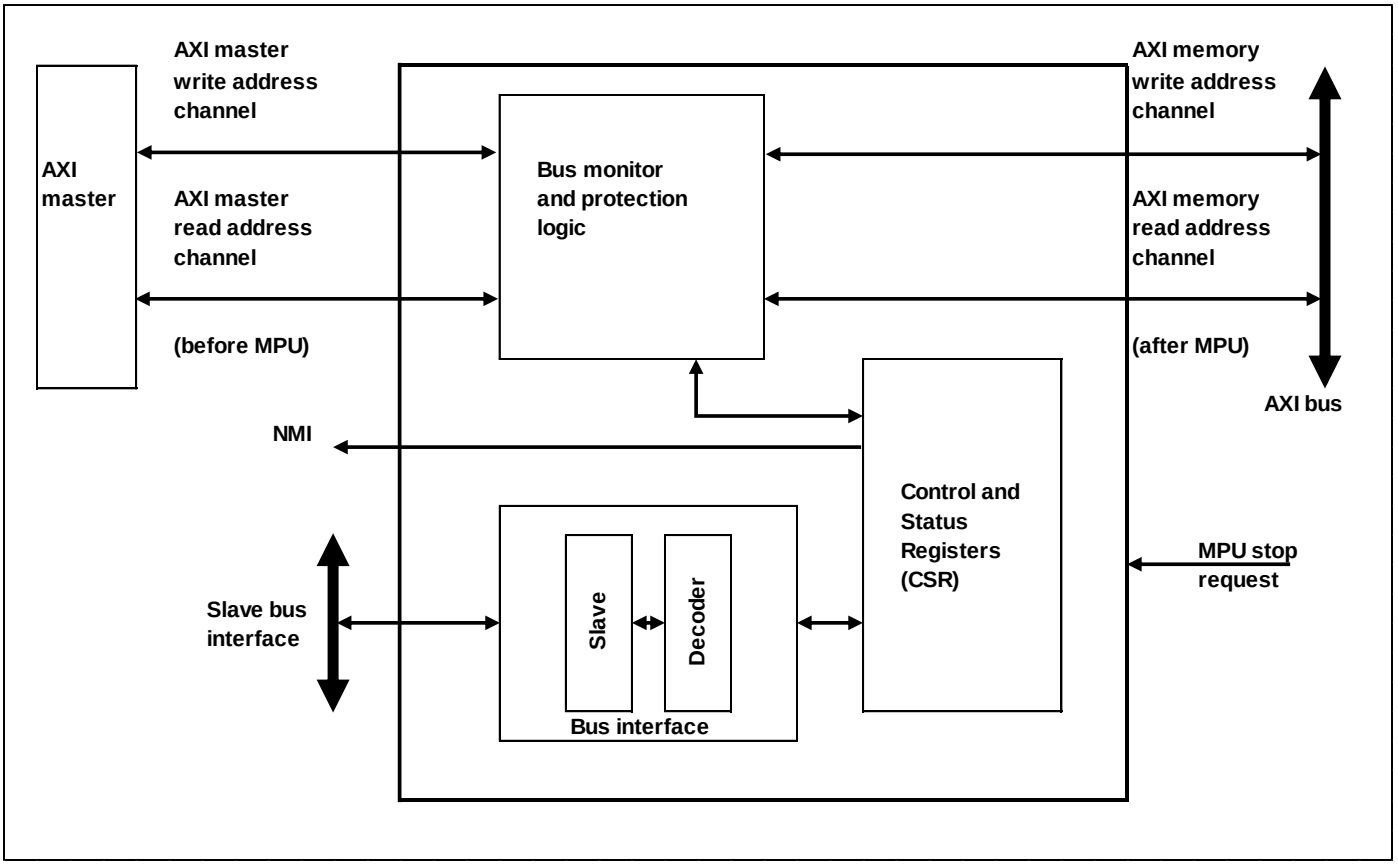
- Each of the eight regions in MPU AXI is specified using corresponding start address and end address
- Background region covers entire 4 GB address space
- On unauthorized access MPU AXI generates an NMI to the CPU
- MPU AXI collects information about the AXI master bus access that caused unauthorized 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
- Supports MPU stop feature that allows blocking of all the accesses to memory space
- 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 2-1 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

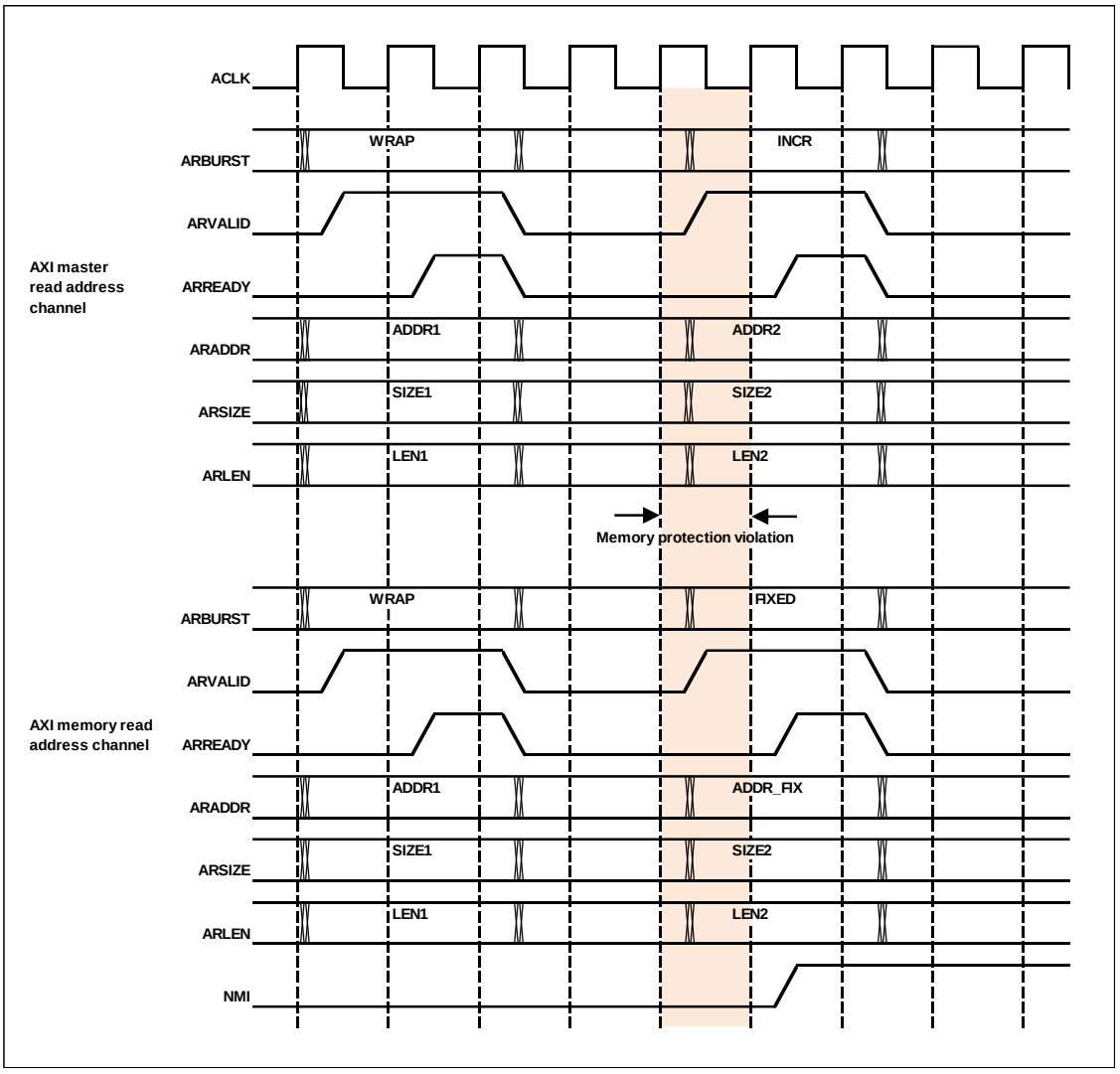
- Separate address or control and data phases
- Burst based transaction where only start address is issued
- Separate write and read address/control channels
- Separate write and read data channels

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 Figure 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 3-1 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.

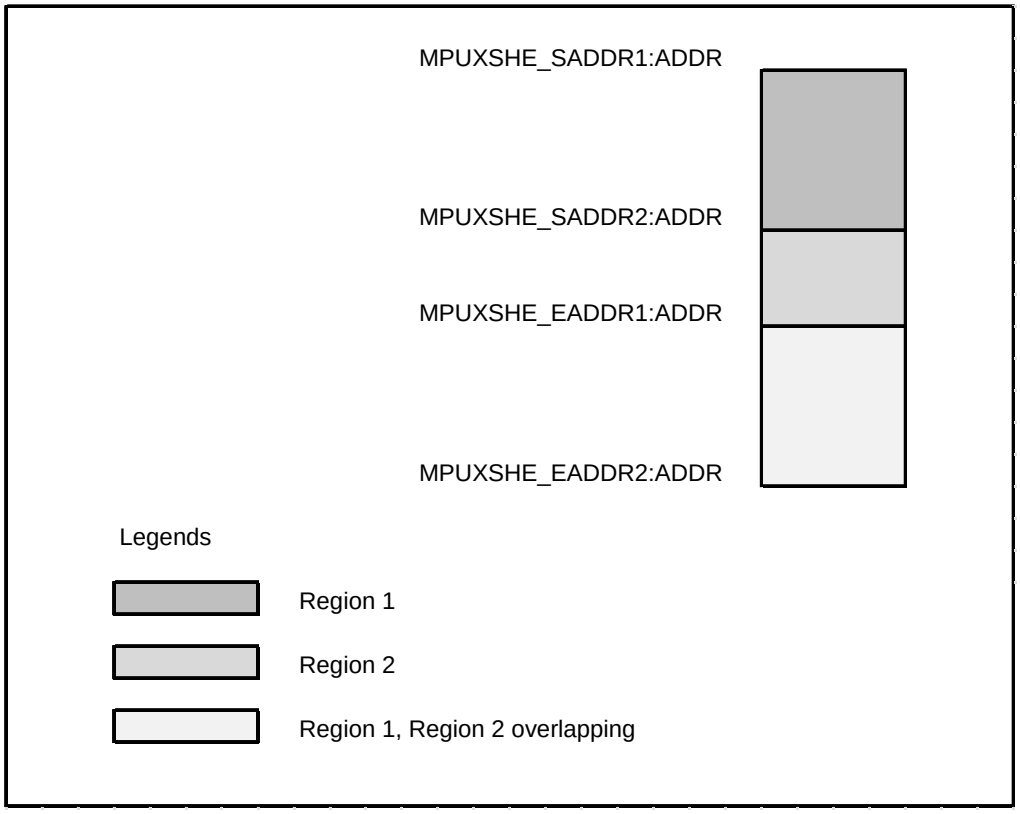
MPU Priority Decision

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 3-2 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 6.3-1 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 Table 3-1 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

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

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			
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			

Offset	+3	+2	+1	+0
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			
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 00000000 00000000 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 and MPU stop feature. Lastly it provides MPU AXI lock status and controls the status for Non-Maskable Interrupt.

■ **MPUXSHE_CTRL0**

MPUXSHE_CTRL0																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
read0	read0	read0	read0	read0	AP[2]	AP[1]	AP[0]	read0	read0	read0	read0	read0	read0	MPUENC	MPUEN	read0	read0	read0	PROT	POEN	MPUSTOPEN	MPUSTOP	LST	read0	read0	read0	read0	read0	read0	NMICL	NMI
R0	R0	R0	R0	R0	RWP	RWP	RWP	R0	R0	R0	R0	R0	R0	RWP	R	R0	R0	R0	RWP	RWP	RWP	R	R	R0	R0	R0	R0	R0	R0	R0WP	R
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Bit Position	Bit Field Name	Bit Description
[31:27]	read0	-
[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]	read0	-
[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 inter- faces are passed on to AXI memory write or read address chan- nel interfaces without any protection "1": MPU AXI monitoring and protection function is enabled
[15:13]	read0	-

Bit Position	Bit Field Name	Bit Description
[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]	MPUSTOPEN	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.
[9]	MPUSTOP	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.
[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]	read0	-
[1]	NMICL	NMI Interrupt Clear "0": No effect "1": Clears the NMI interrupt flag Read returns "0".
[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

MPUXSHE_NMIEN																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	NMIEN
R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	RWP
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit Position	Bit Field Name	Bit Description
[31:8]	read0	-
[7:1]	read0	-
[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

MPUXSHE_WERRC																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	AWSIZE[2]	AWSIZE[1]	AWSIZE[0]	AWBURST[1]	AWBURST[0]	AWLEN[3]	AWLEN[2]	AWLEN[1]	AWLEN[0]	AWPROTPRIV	AWMPV
R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R	R	R	R	R	R	R	R	R	R	R
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	X	X	X	X	X	X	X	X	X	X	0

Bit Position	Bit Field Name	Bit Description
[31:16]	read0	-
[15:11]	read0	-
[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.
[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

MPUXSHE_WERRA																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
AWADDR[31]	AWADDR[30]	AWADDR[29]	AWADDR[28]	AWADDR[27]	AWADDR[26]	AWADDR[25]	AWADDR[24]	AWADDR[23]	AWADDR[22]	AWADDR[21]	AWADDR[20]	AWADDR[19]	AWADDR[18]	AWADDR[17]	AWADDR[16]	AWADDR[15]	AWADDR[14]	AWADDR[13]	AWADDR[12]	AWADDR[11]	AWADDR[10]	AWADDR[9]	AWADDR[8]	AWADDR[7]	AWADDR[6]	AWADDR[5]	AWADDR[4]	AWADDR[3]	AWADDR[2]	AWADDR[1]	AWADDR[0]
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	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

MPUXSHE_RERRC																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	ARSIZE[2]	ARSIZE[1]	ARSIZE[0]	ARBURST[1]	ARBURST[0]	ARLEN[3]	ARLEN[2]	ARLEN[1]	ARLEN[0]	ARPROTPRIV	ARMPV
R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R	R	R	R	R	R	R	R	R	R	R
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	X	X	X	X	X	X	X	X	X	X	0

Bit Position	Bit Field Name	Bit Description
[31:16]	read0	-
[15:11]	read0	-
[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.
[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

MPUXSHE_RERRA																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
ARADDR[31]	ARADDR[30]	ARADDR[29]	ARADDR[28]	ARADDR[27]	ARADDR[26]	ARADDR[25]	ARADDR[24]	ARADDR[23]	ARADDR[22]	ARADDR[21]	ARADDR[20]	ARADDR[19]	ARADDR[18]	ARADDR[17]	ARADDR[16]	ARADDR[15]	ARADDR[14]	ARADDR[13]	ARADDR[12]	ARADDR[11]	ARADDR[10]	ARADDR[9]	ARADDR[8]	ARADDR[7]	ARADDR[6]	ARADDR[5]	ARADDR[4]	ARADDR[3]	ARADDR[2]	ARADDR[1]	ARADDR[0]
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	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

MPUXSHE_CTRL1																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	AP[2]	AP[1]	AP[0]	read0	read0	read0	read0	read0	read0	MPUENC	MPUEN
R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	RWP	RWP	RWP	R0	R0	R0	R0	R0	R0	RWP	R
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit Position	Bit Field Name	Bit Description
[31:16]	read0	-
[15:11]	read0	-
[10:8]	AP	Access Permissions These bits are used to control access permissions for region 1.
[7:2]	read0	-
[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

MPUXSHE_SADDR1																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
SADDR[31]	SADDR[30]	SADDR[29]	SADDR[28]	SADDR[27]	SADDR[26]	SADDR[25]	SADDR[24]	SADDR[23]	SADDR[22]	SADDR[21]	SADDR[20]	SADDR[19]	SADDR[18]	SADDR[17]	SADDR[16]	SADDR[15]	SADDR[14]	SADDR[13]	SADDR[12]	SADDR[11]	SADDR[10]	SADDR[9]	SADDR[8]	SADDR[7]	SADDR[6]	SADDR[5]	SADDR[4]	SADDR[3]	SADDR[2]	SADDR[1]	SADDR[0]
RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	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

MPUXSHE_EADDR1																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
EADDR[31]	EADDR[30]	EADDR[29]	EADDR[28]	EADDR[27]	EADDR[26]	EADDR[25]	EADDR[24]	EADDR[23]	EADDR[22]	EADDR[21]	EADDR[20]	EADDR[19]	EADDR[18]	EADDR[17]	EADDR[16]	EADDR[15]	EADDR[14]	EADDR[13]	EADDR[12]	EADDR[11]	EADDR[10]	EADDR[9]	EADDR[8]	EADDR[7]	EADDR[6]	EADDR[5]	EADDR[4]	EADDR[3]	EADDR[2]	EADDR[1]	EADDR[0]
RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	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

MPUXSHE_UNLOCK																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
UNLOCK[31]	UNLOCK[30]	UNLOCK[29]	UNLOCK[28]	UNLOCK[27]	UNLOCK[26]	UNLOCK[25]	UNLOCK[24]	UNLOCK[23]	UNLOCK[22]	UNLOCK[21]	UNLOCK[20]	UNLOCK[19]	UNLOCK[18]	UNLOCK[17]	UNLOCK[16]	UNLOCK[15]	UNLOCK[14]	UNLOCK[13]	UNLOCK[12]	UNLOCK[11]	UNLOCK[10]	UNLOCK[9]	UNLOCK[8]	UNLOCK[7]	UNLOCK[6]	UNLOCK[5]	UNLOCK[4]	UNLOCK[3]	UNLOCK[2]	UNLOCK[1]	UNLOCK[0]
R0WP	R0WP	R0WP	R0WP	R0WP	R0WP	R0WP	R0WP	R0WP	R0WP	R0WP	R0WP	R0WP	R0WP	R0WP	R0WP	R0WP	R0WP	R0WP	R0WP	R0WP	R0WP	R0WP	R0WP	R0WP	R0WP	R0WP	R0WP	R0WP	R0WP	R0WP	R0WP
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	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: This register can not be written by 8-bit or 16-bit write access; any such access causes protection error. For more details on lock and unlock values, refer to the device specific datasheet.

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

MPUXSHE_MID																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
MID[31]	MID[30]	MID[29]	MID[28]	MID[27]	MID[26]	MID[25]	MID[24]	MID[23]	MID[22]	MID[21]	MID[20]	MID[19]	MID[18]	MID[17]	MID[16]	MID[15]	MID[14]	MID[13]	MID[12]	MID[11]	MID[10]	MID[9]	MID[8]	MID[7]	MID[6]	MID[5]	MID[4]	MID[3]	MID[2]	MID[1]	MID[0]
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

Bit Position	Bit Field Name	Bit Description
[31:0]	MID	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 the device. This register helps in developing software to hardware version implemented in the device

5. Notes on Using MPU AXI

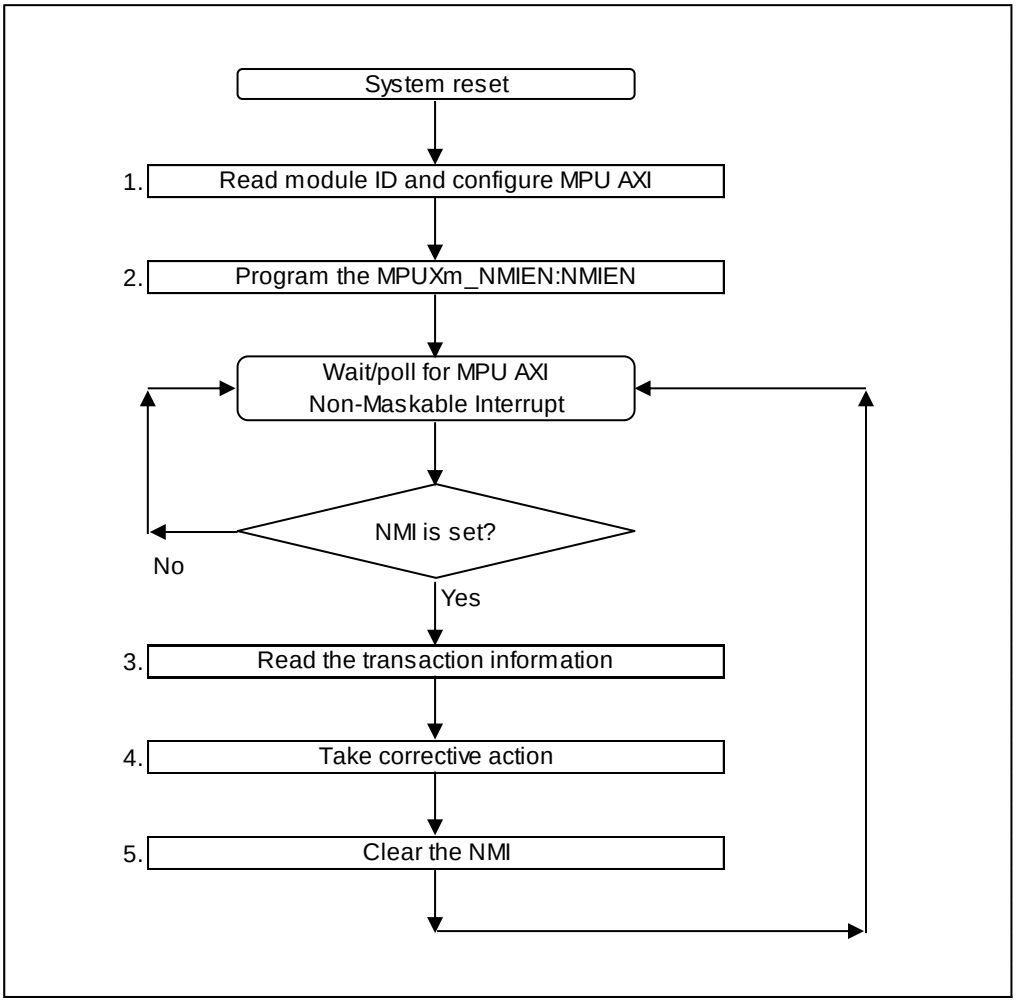
This section is the 'programmer's guide', which lists the usage notes for programming the MPU AXI module. It is recommended to read these guidelines before programming the MPU AXI module.

■ 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 27: 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-T00P00R01L04-E1-GBB01

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.

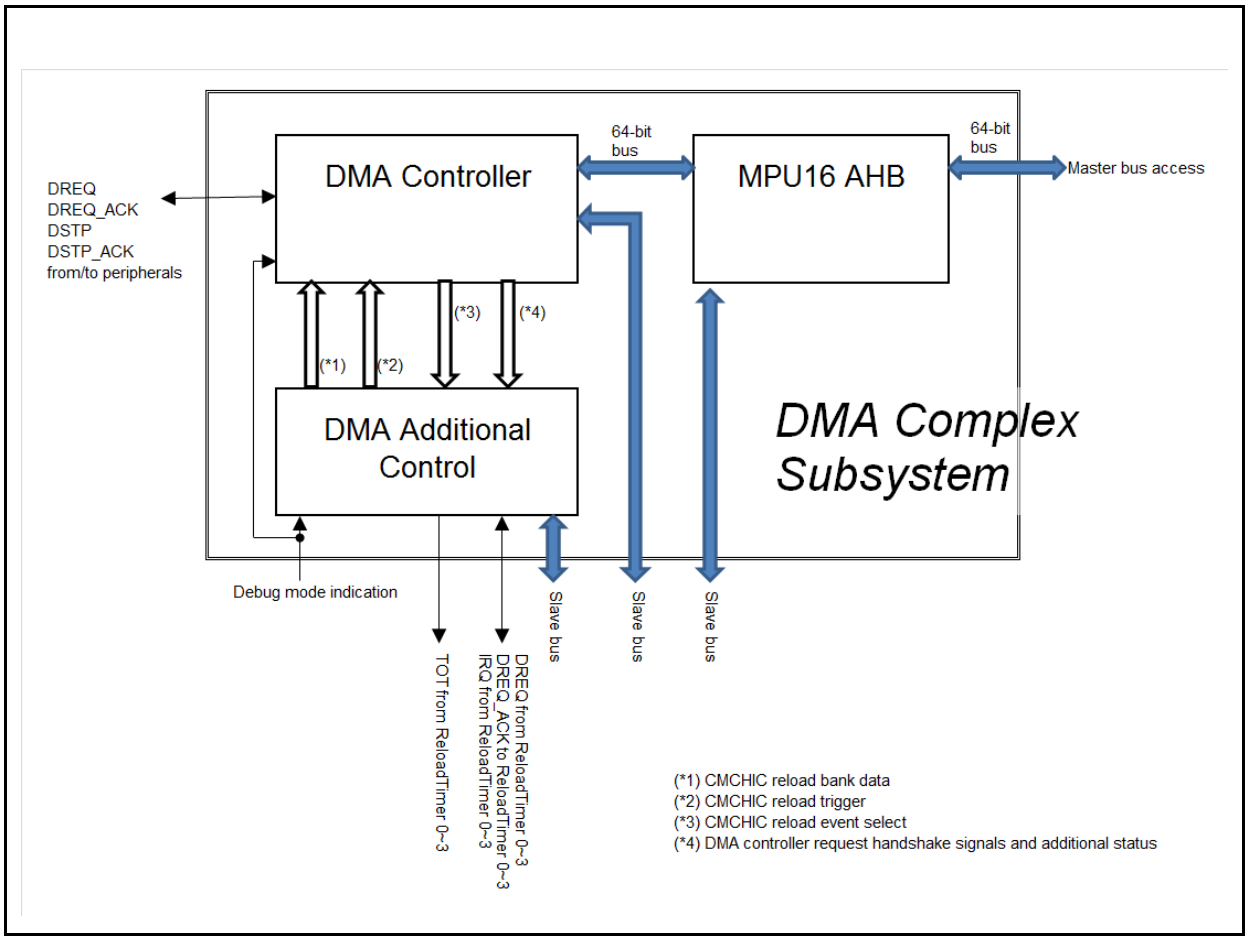
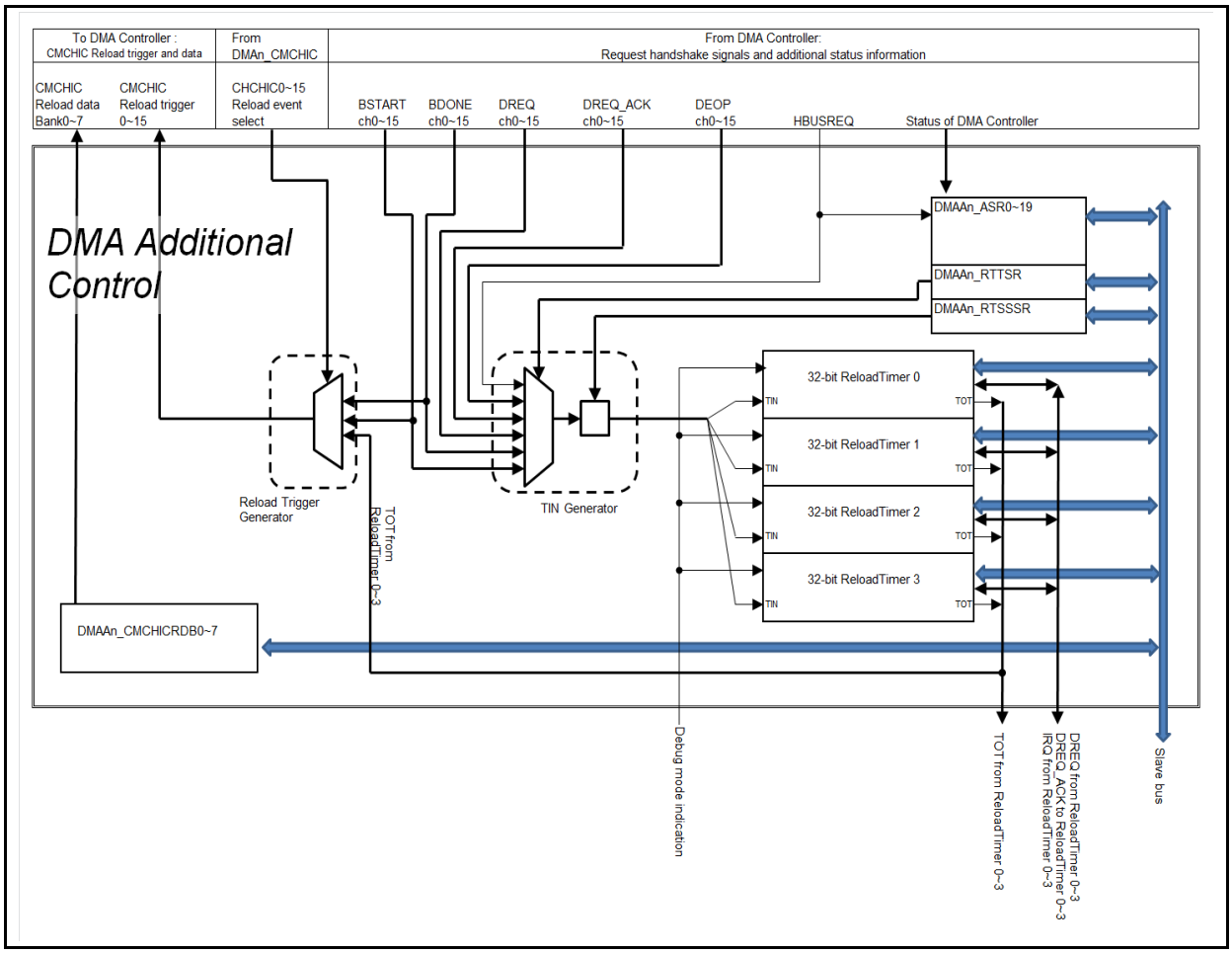


Figure 2-2 Entire Block Diagram of DMA Additional Control Block



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.

When stopping DMA_CLK and accessing a register, an error reply is returned.

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

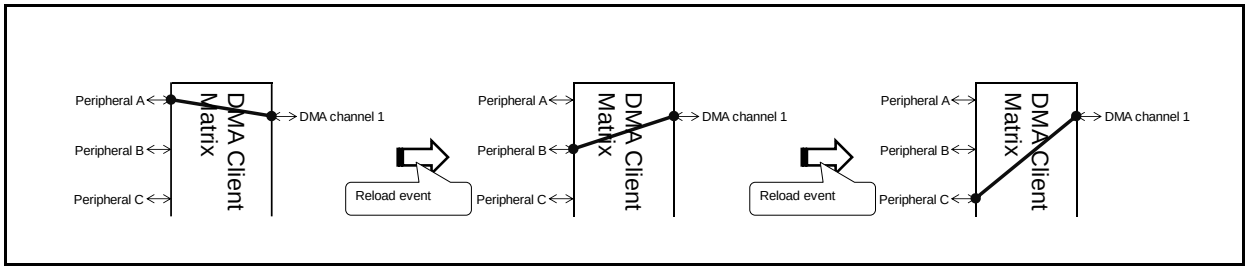
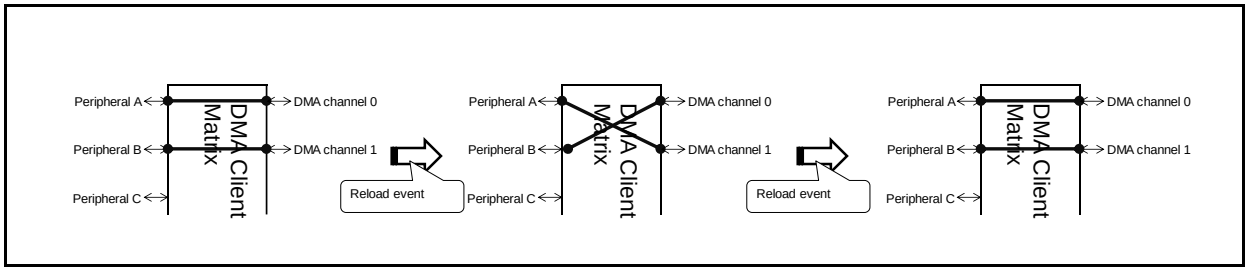


Figure 3-2 Example of CMCHIC Reload Usage: Swapping Client and Channel Connection



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 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.

'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

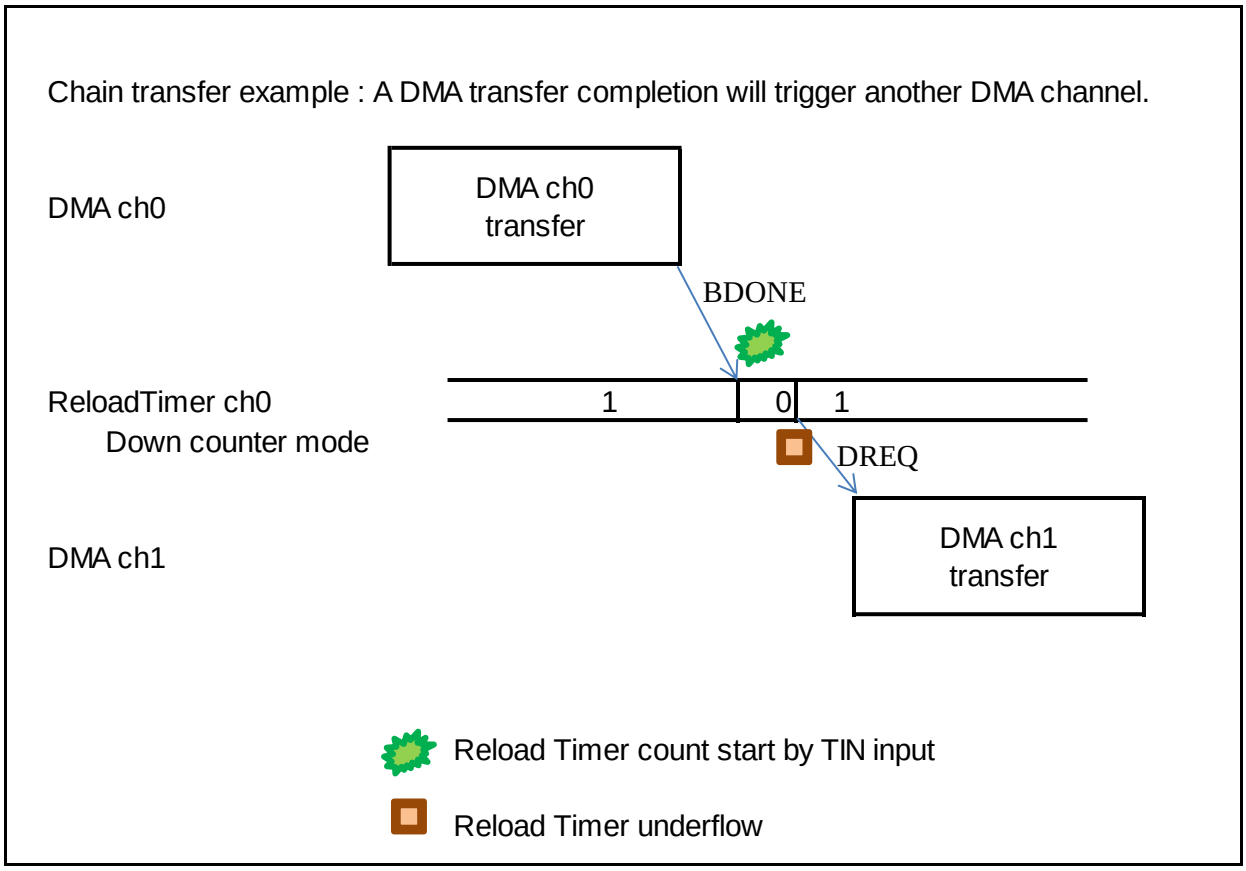


Figure 3-4 Chain Transfer Example

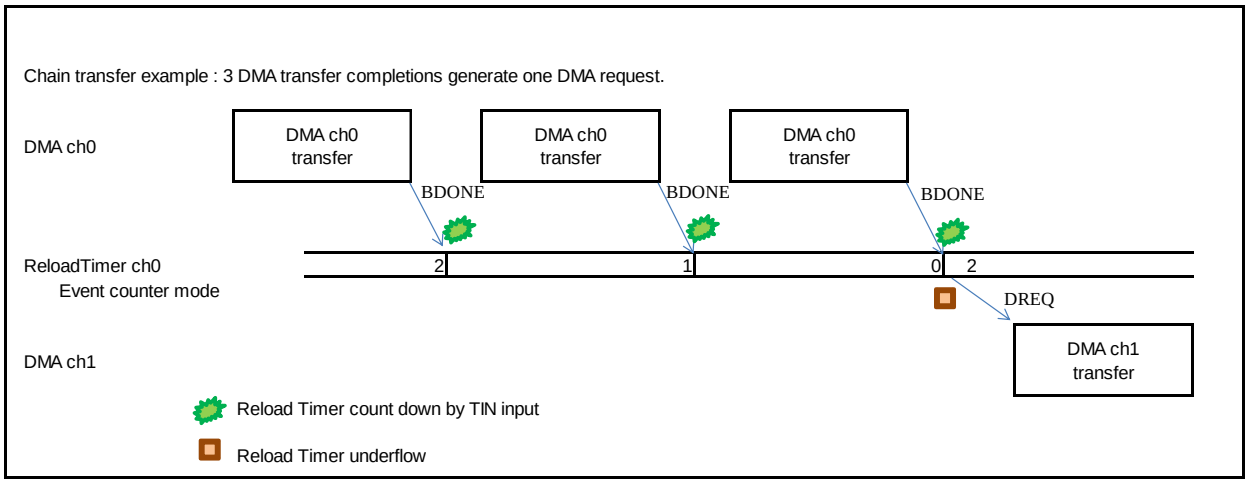
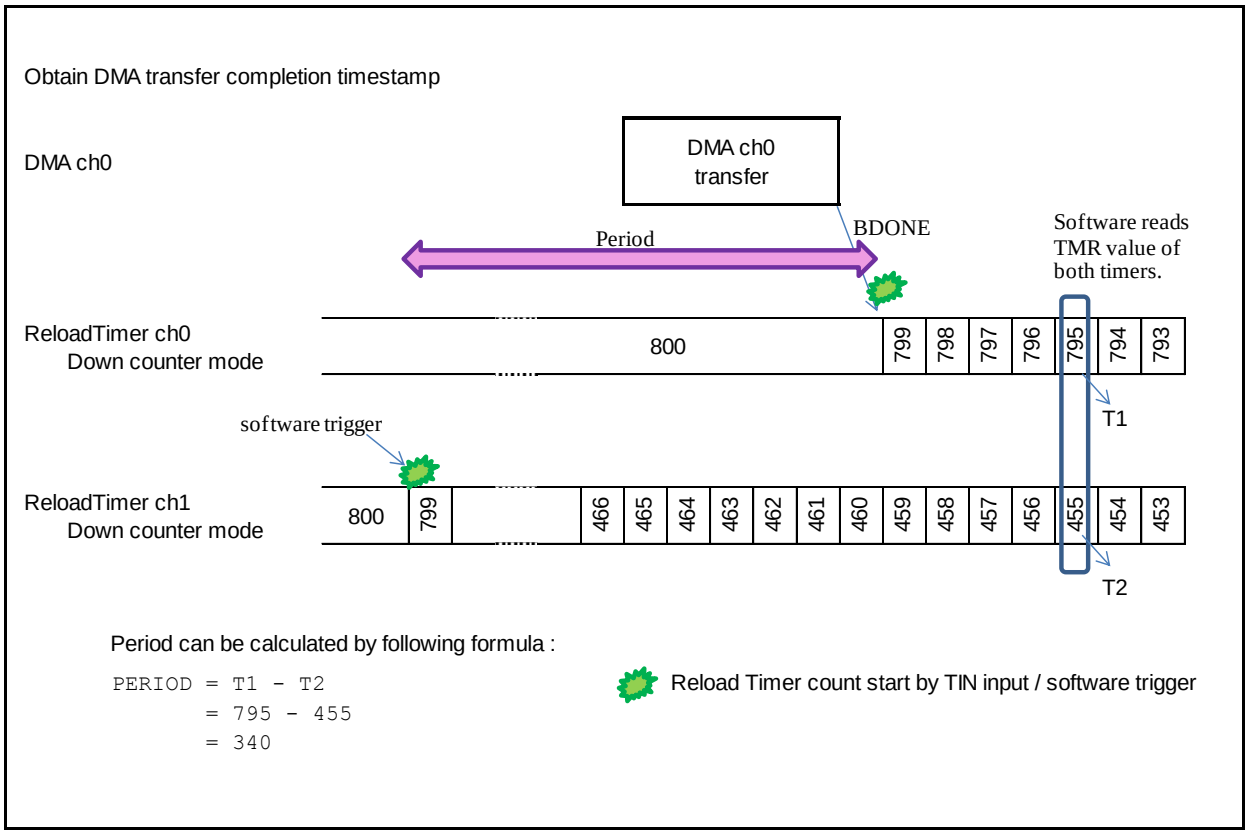


Figure 3-5 Example to Obtain Timestamp



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.

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 .

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.*

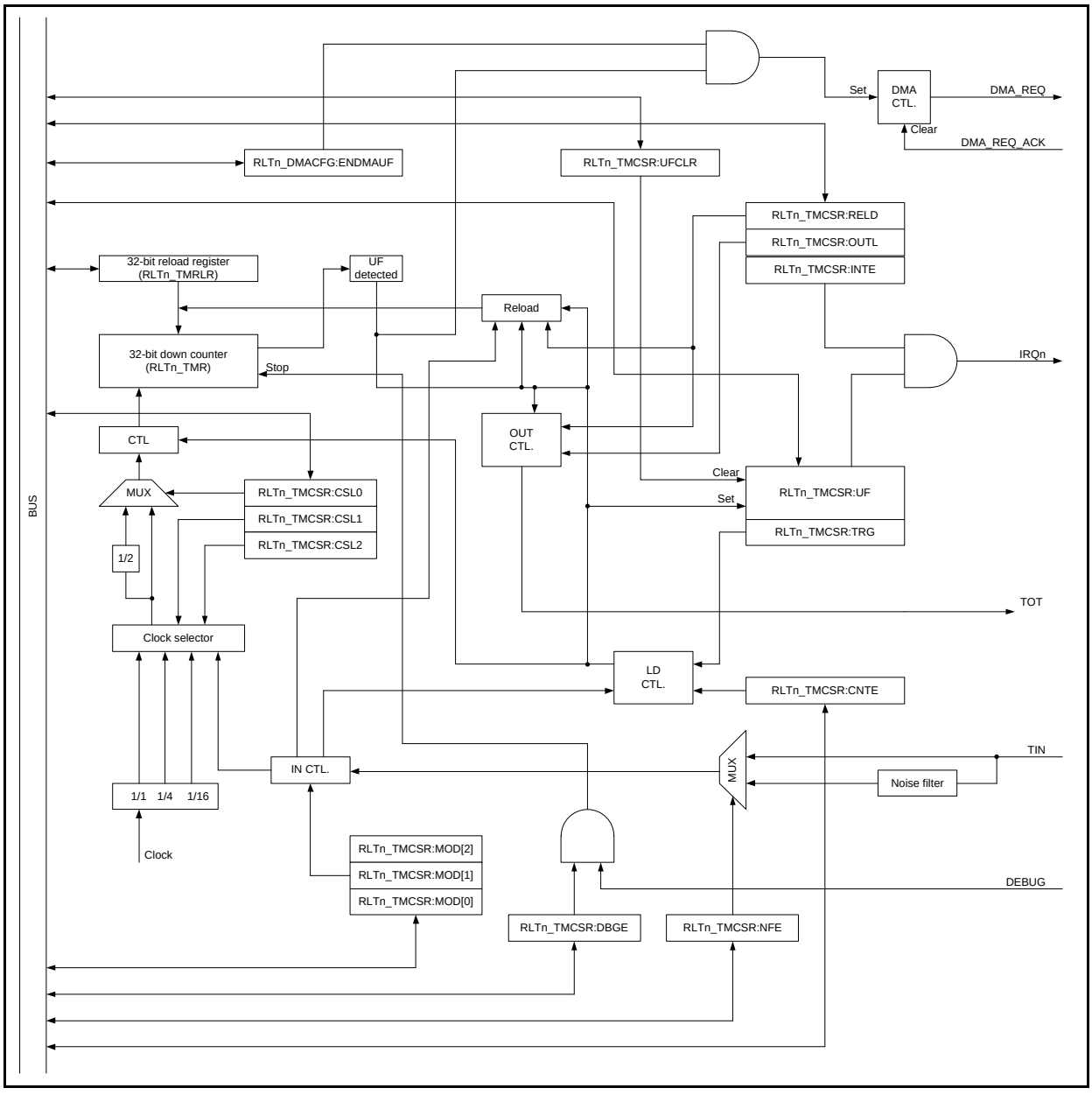
DMA and Interrupts

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
7.5 DMA

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

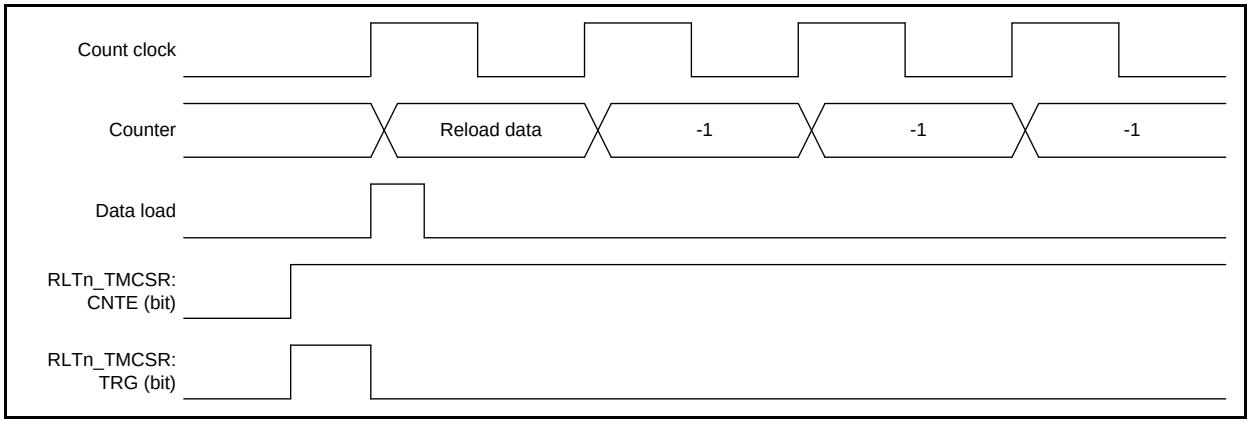
Writing "1" to both DMAAn_RLTm_TMCSR:CNTE and DMAAn_RLTm_TMCSR:TRG bits enables and starts counting at the same time. Using the DMAAn_RLTm_TMCSR:TRG bit as a trigger input is always available when the timer is enabled (DMAAn_RLTm_TMCSR:CNTE = 1), regardless of the operation mode.

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



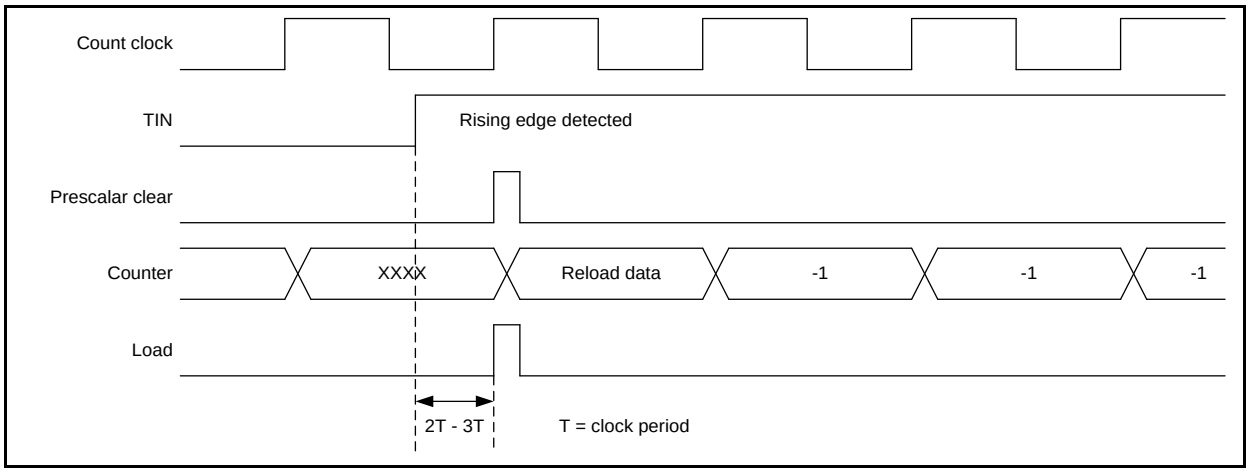
Input Functions of 32-Bit Reload Timer (In Internal Clock Mode)

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.

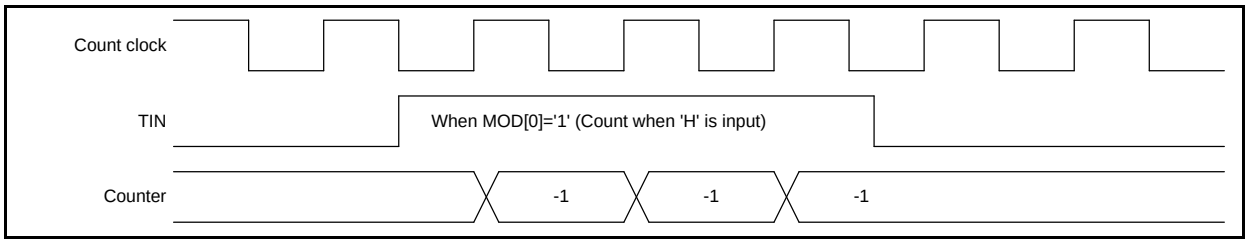
Figure 7-2 Trigger Input Operation of 32-Bit Reload Timer



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

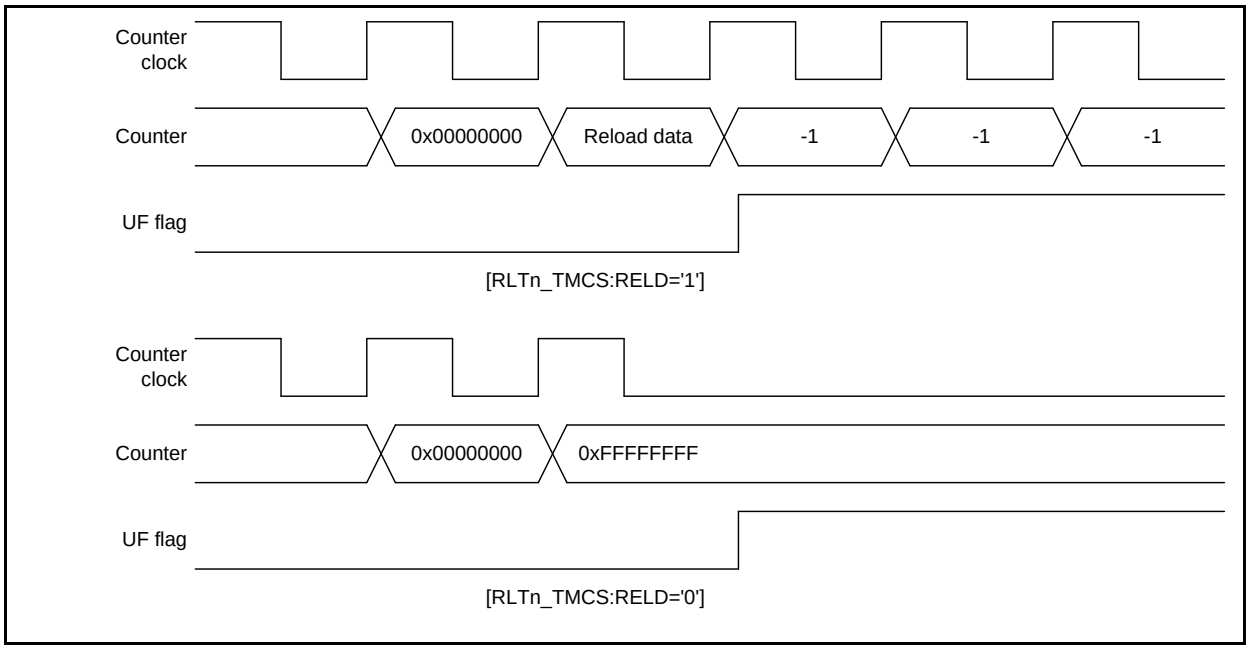
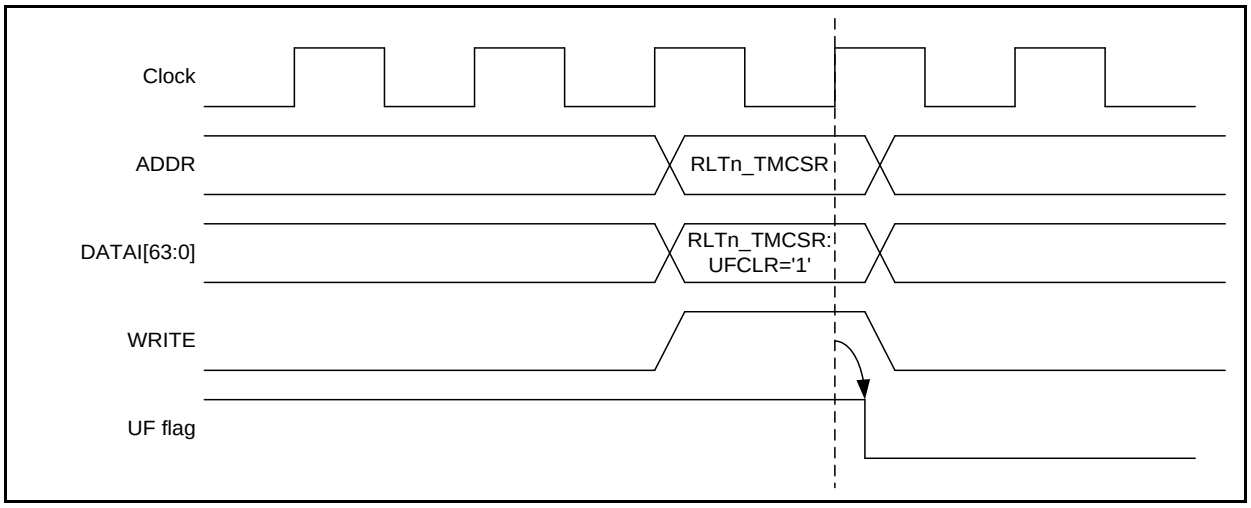


Figure 7-5 Clearing of Underflow Bit



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

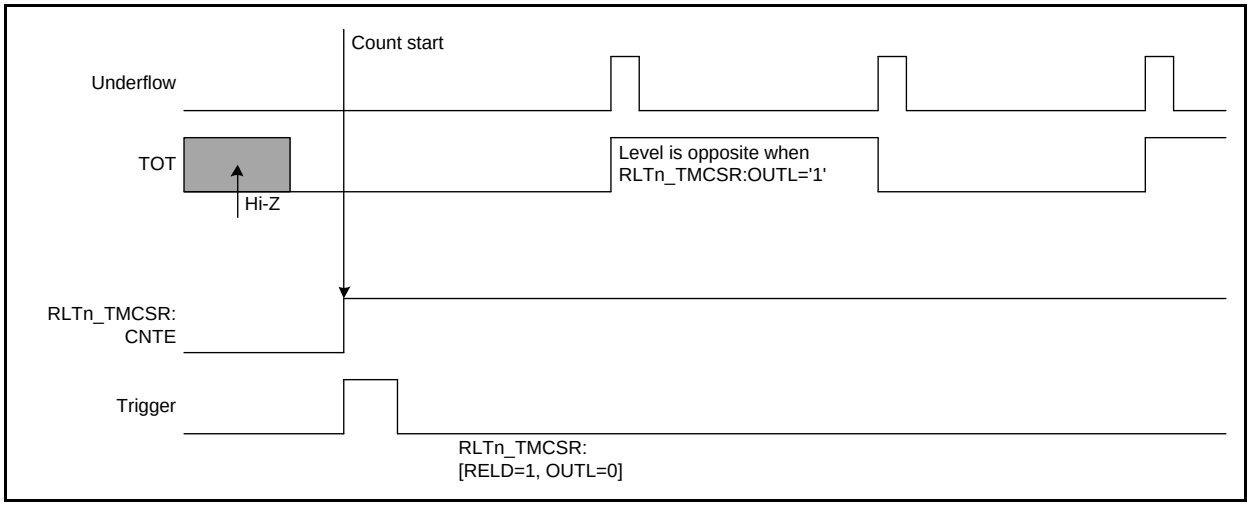
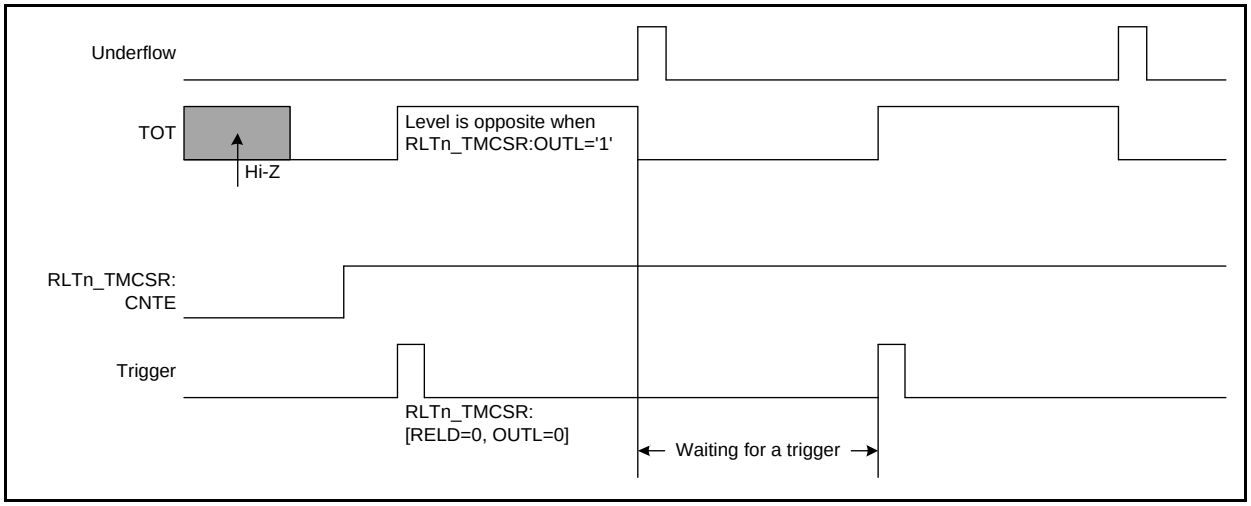


Figure 7-7 Output Signal Function of 32-Bit Reload Timer in One-Shot Mode



7.4. Counter Operation State

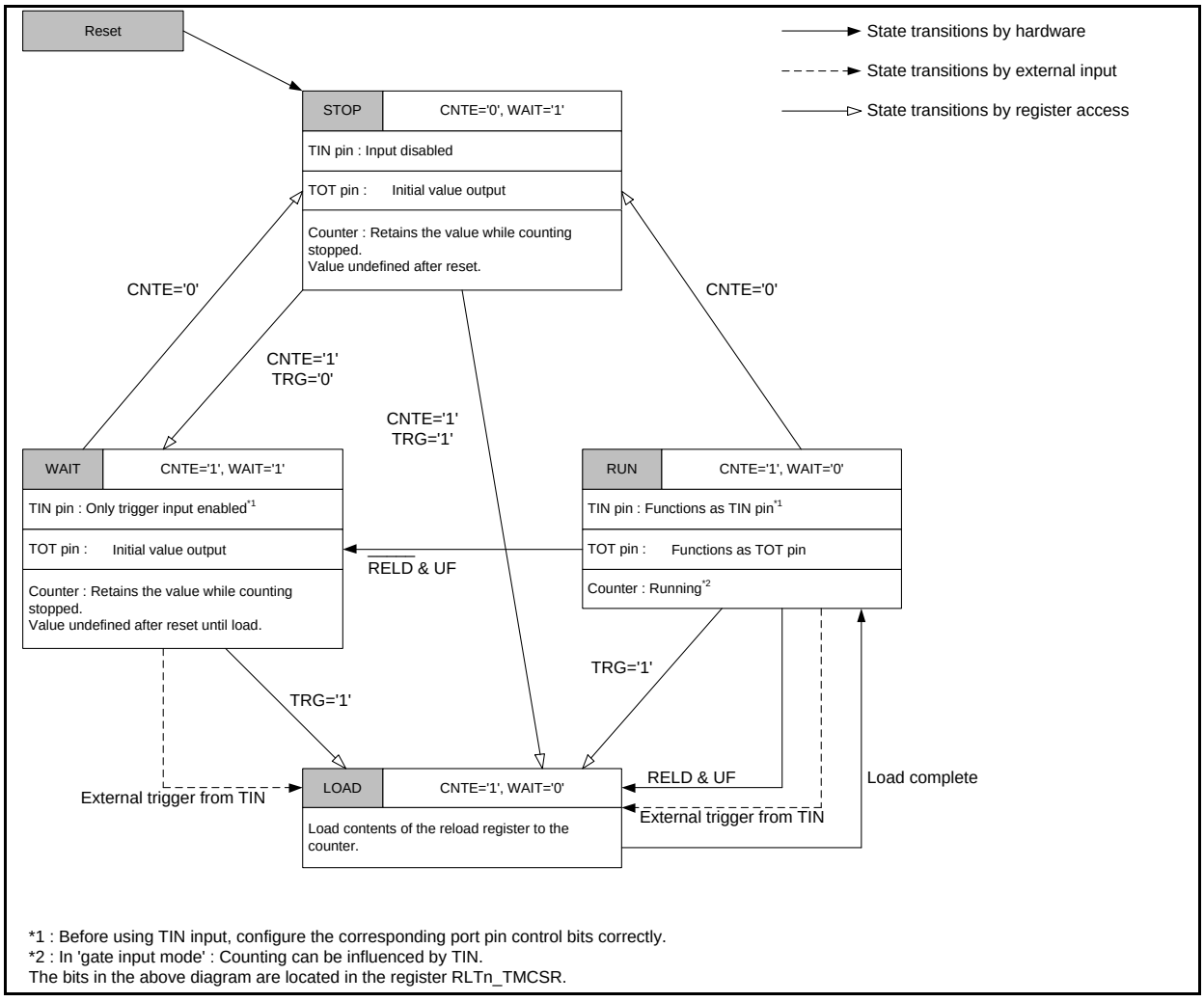
The counter state is determined by DMAAn_RLTm_TMCSR:CNTE bit in the Timer Control Status Register and the internal WAIT signal.

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.

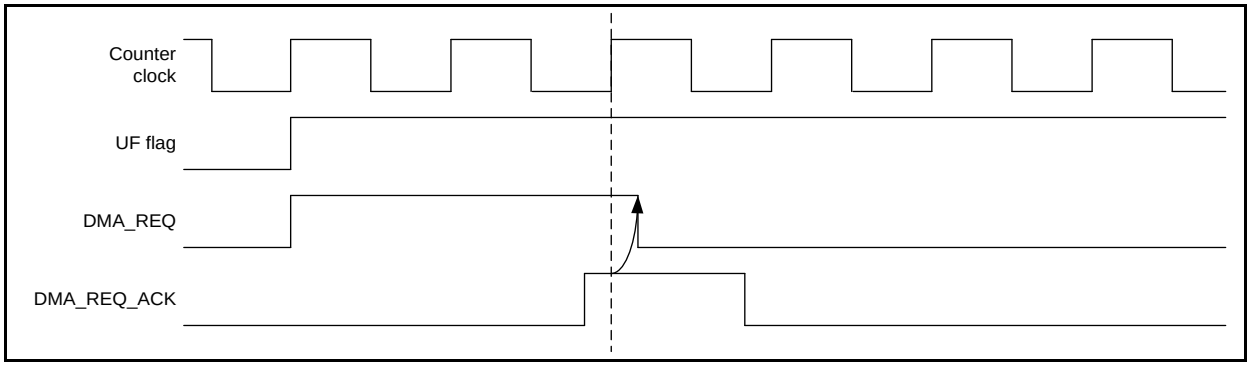
Enabling DMA Support

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.

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 8-1 and Table 8-2 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 8-3 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 8-3 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 8-3 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

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.*

[bit6] Reserved

[bit5] OUTL: Output Level

This Output Level (OUTL) bit sets the output level for the TOT.

Refer to Table 8-4.

[bit4] RELD: Reload

This Reload (RELD) bit enables reload operations.

Refer to Table 8-4.

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 8-1 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 8-2 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 8-3 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 8-4 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 28: DMA Controller

Head page of chapter "DMA Controller"

1. **Overview**
2. **Configuration**
3. **Operational Description**
4. **Registers**
5. **Additional Information**

DMA-T00P00R01L02-E1-GBB02

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.

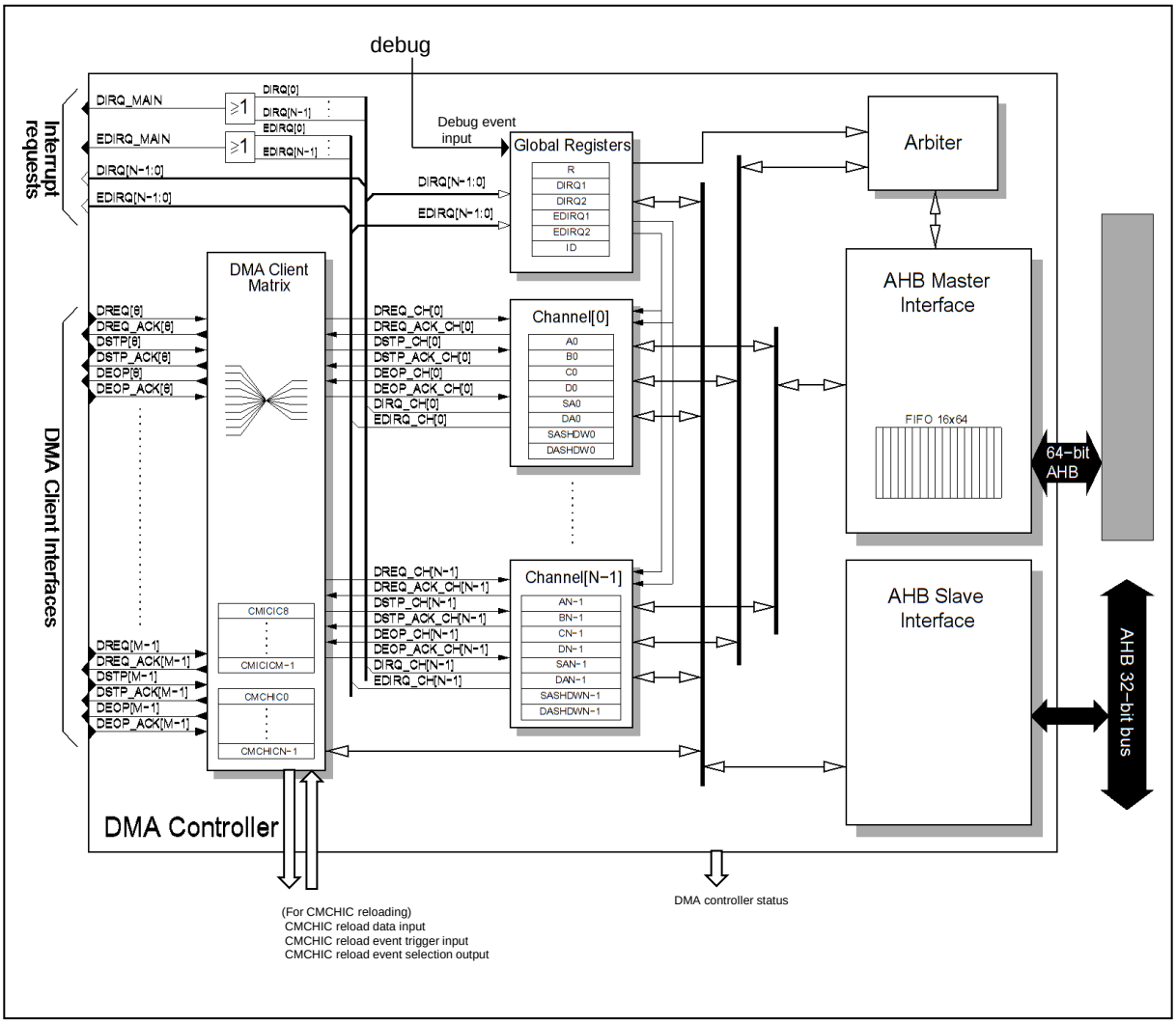
The following items are the features related to data transfer by the DMA Controller:

- 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, …, 331

'M' represents number of client I/F. M = 332 in this product.

Note:

- *Please refer "APPENDIX:DMA channel Activation Factors" for the assigned peripherals as DMA client.*

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

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)

Block Count	Beat Limit	Alternate	Resulting Sequence of AHB Transfers
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
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)

Block Count	Beat Limit	Alternate	Resulting Sequence of AHB Transfers
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

Block Count	Beat Limit	Alternate	Resulting Sequence of AHB Transfers
0	INCR8	1	1x SINGLE RD + 1x SINGLE WR
1	INCR8	1	2x (1x SINGLE RD + 1x SINGLE WR)
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

Block Count	Beat Limit	Alternate	Resulting Sequence of AHB Transfers
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)
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

Notes:

- *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.

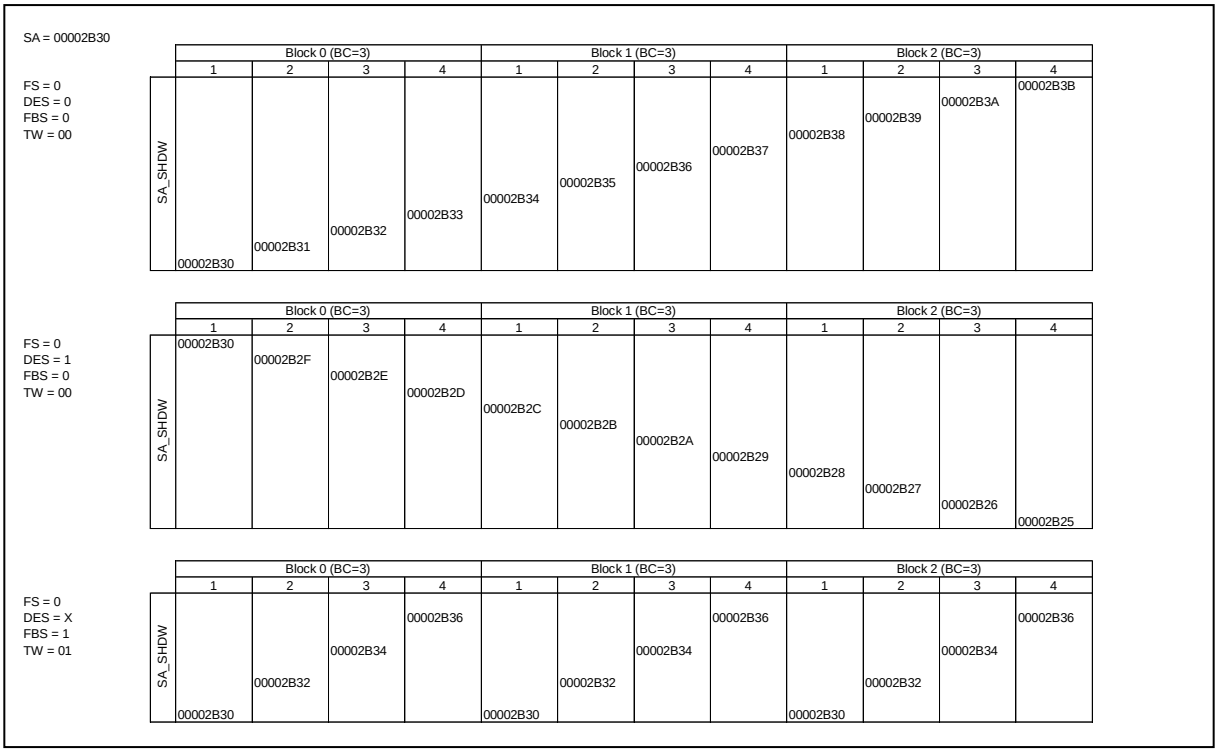
Table 3-2 Source Address Shadow Update Behavior

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.

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 4 7	2	None
2	4 7 8	4	2
3	4 7 8	7	4
4	1 4 8	8	7
5	1 4 7	1	8

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.*

Note:

- *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

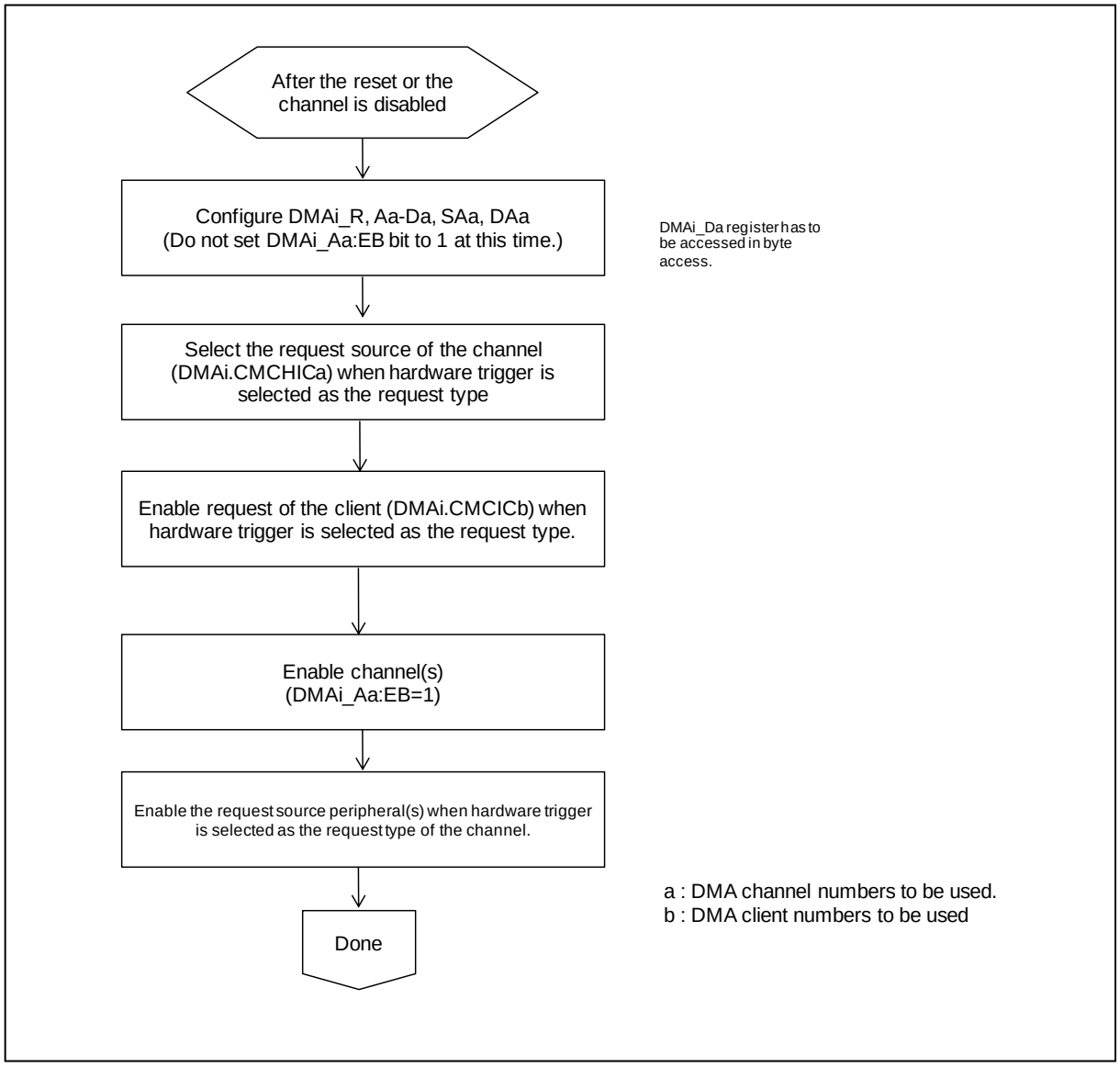
Stopping Clock for DMA Controller or entering PSS(Power Saving State) Mode

- 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

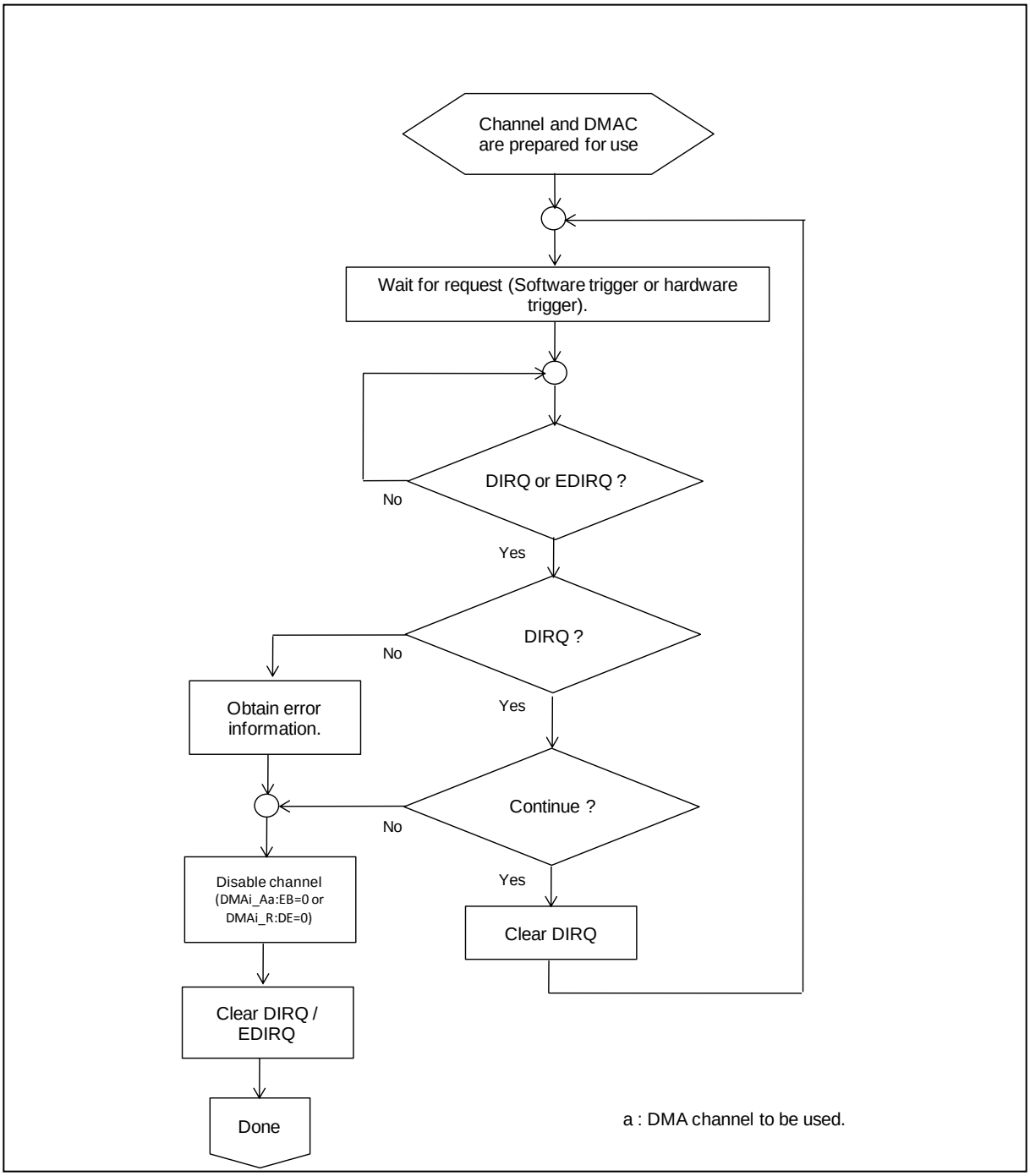
Configuring and Enabling a DMA Channel

Figure 3-2 Recommended Flow for Configuring and Enabling a DMA Channel(s)



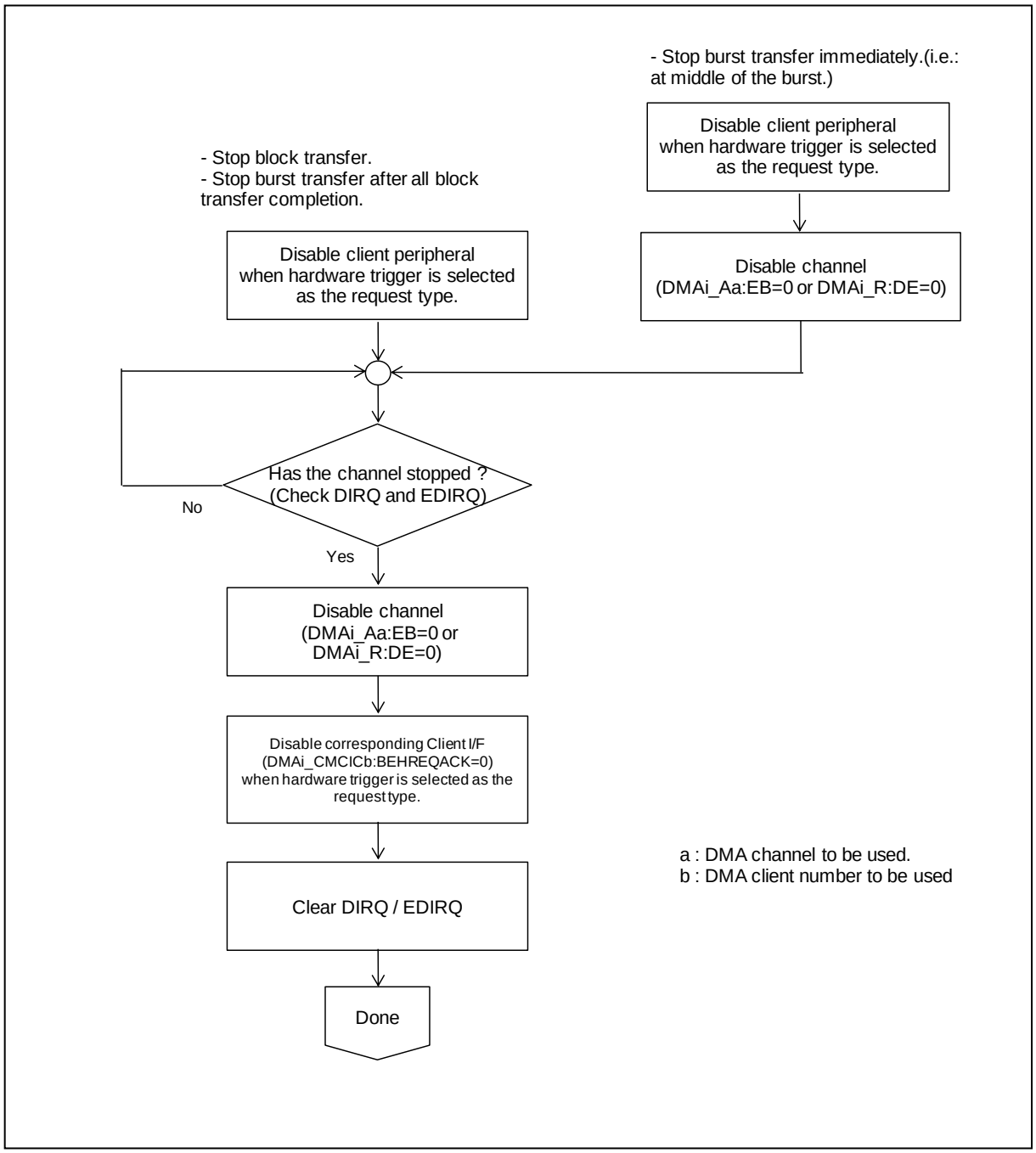
Handling DMAC Requests

Figure 3-3 Recommended Flow for Handling DMAC Requests



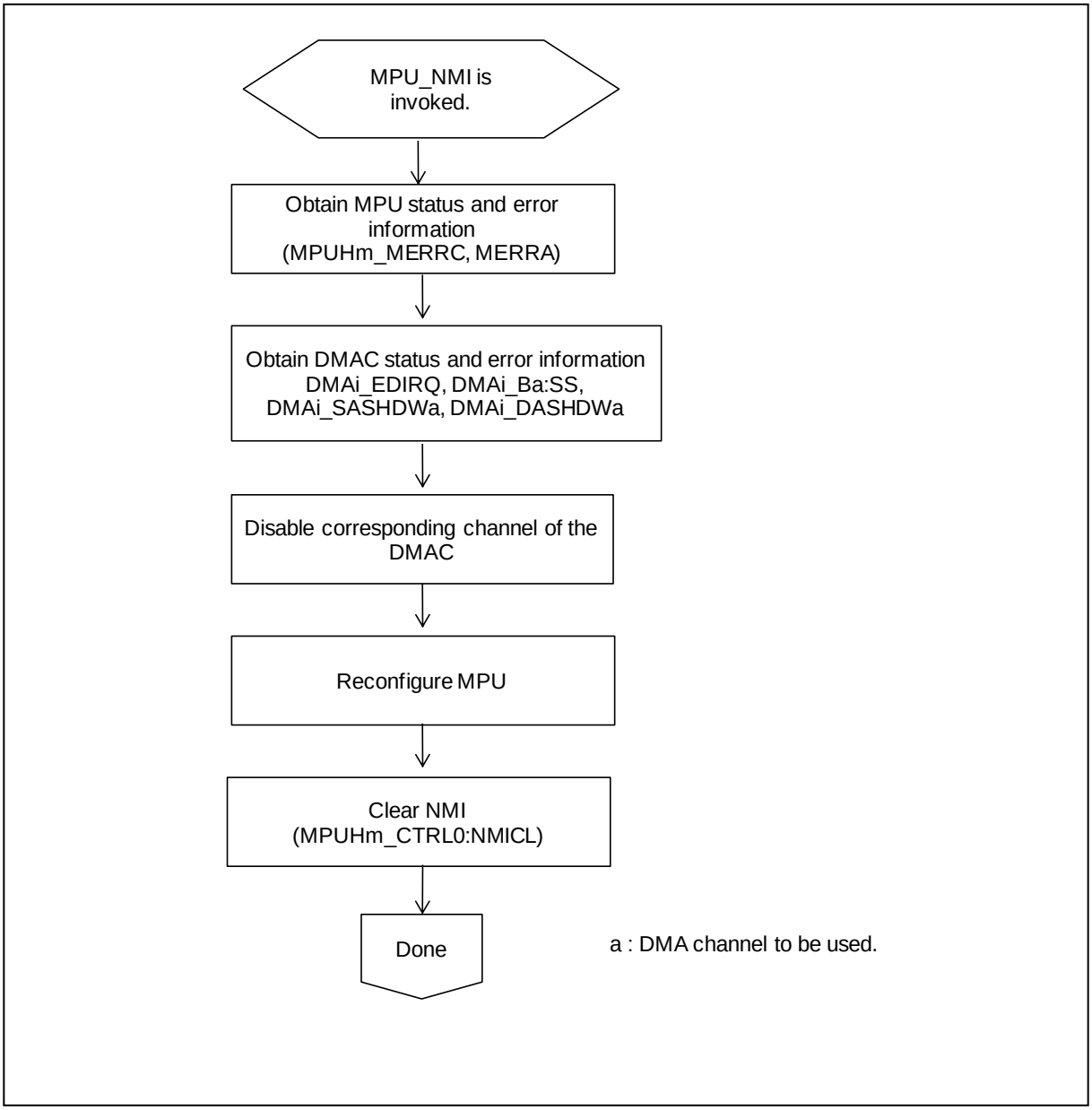
Stop Transfer

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

'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, …, 331.

'M' represents number of client I/F. M = 332 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: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

*: 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

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.

[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.

[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 DMAiAn: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 DMAiAn: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

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_CMICIC331.*

[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, … , 331)*

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,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	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

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-T00P00R01L03-E1-GBB00

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.

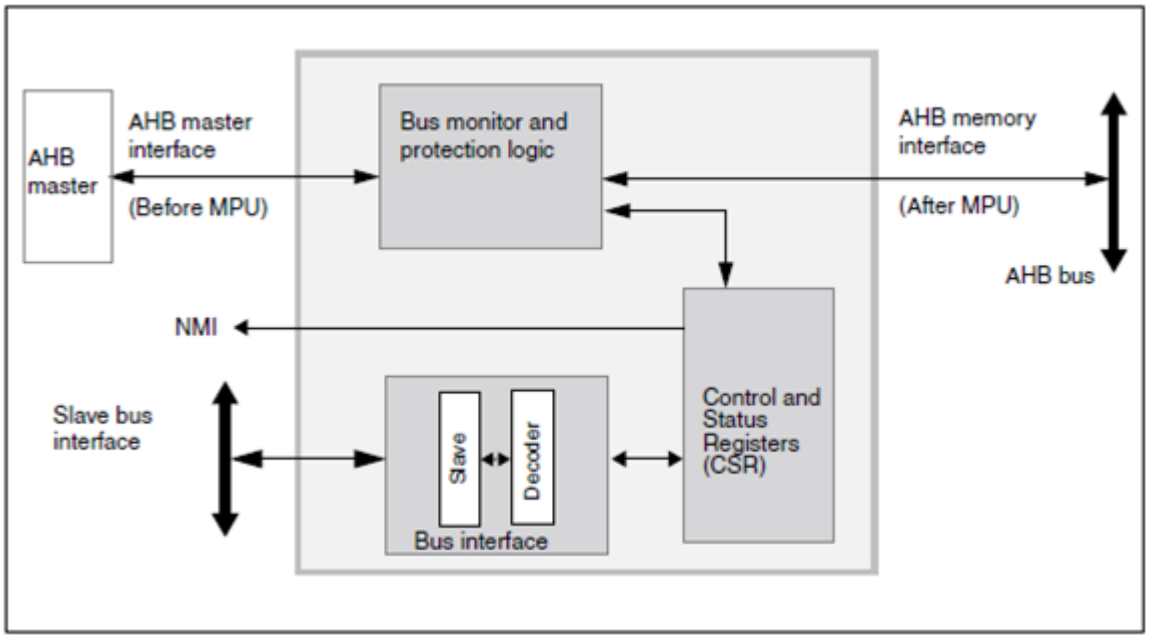
The MPU16 AHB provides 16 regions and one background region. Each region has corresponding Access Permission bits that define the permission attributes for that particular region. Any unauthorized access to that memory space is flagged using a non-maskable interrupt. MPU16 AHB also collects information 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

Block Diagram of the MPU16 AHB

Figure 1-1 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.

2. 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)

Figure 2-1 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

2.1. 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 2-1 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	-

Bit Position	Bit Field Name	Bit Description
[17]	MPUENC	MPU AHB Enable Control 0: MPU AHB Monitoring and Protection Function is disabled 1: MPU AHB Monitoring and Protection Function is enabled
[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], MPUHm_CTRL0:POEN and MPUHn_CTRL0:PROT can only be written when the MPU is disabled (MPUHn_CTRL0:MPUEN = 0).*

2.2. 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 2-2 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.

2.3. 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 2-3 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.*

2.4. 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 2-4 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.*

2.5. 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 2-5 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

2.6. 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 2-6 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 (MPUHm_CTRL1 to 16:MPUEN = 0) or the MPU is disabled (MPUHn_CTRL0:MPUEN = 0).*

2.7. 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 2-7 MPU16 AHB End Address Register for Region 1 (MPUHm_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).*

2.8. 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 2-8 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:0x ACCABB56 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.

2.9. 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 2-9 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.*

3. 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_EADDR16n 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.

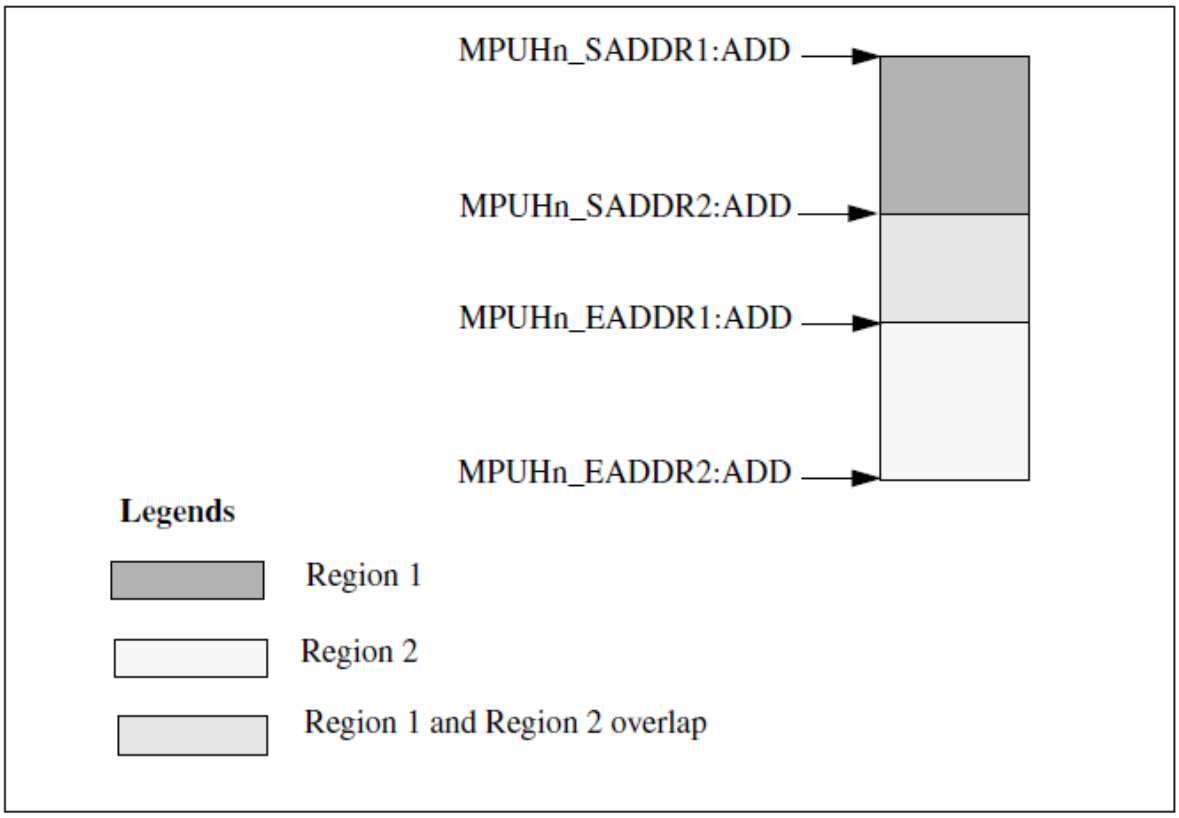
The priority of region 16 is the highest, and the priority of region 15 is high in the second.

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 3-1 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 3-1 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 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**

CANFDMRE-T00P00R02L00-E1-GBB00

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

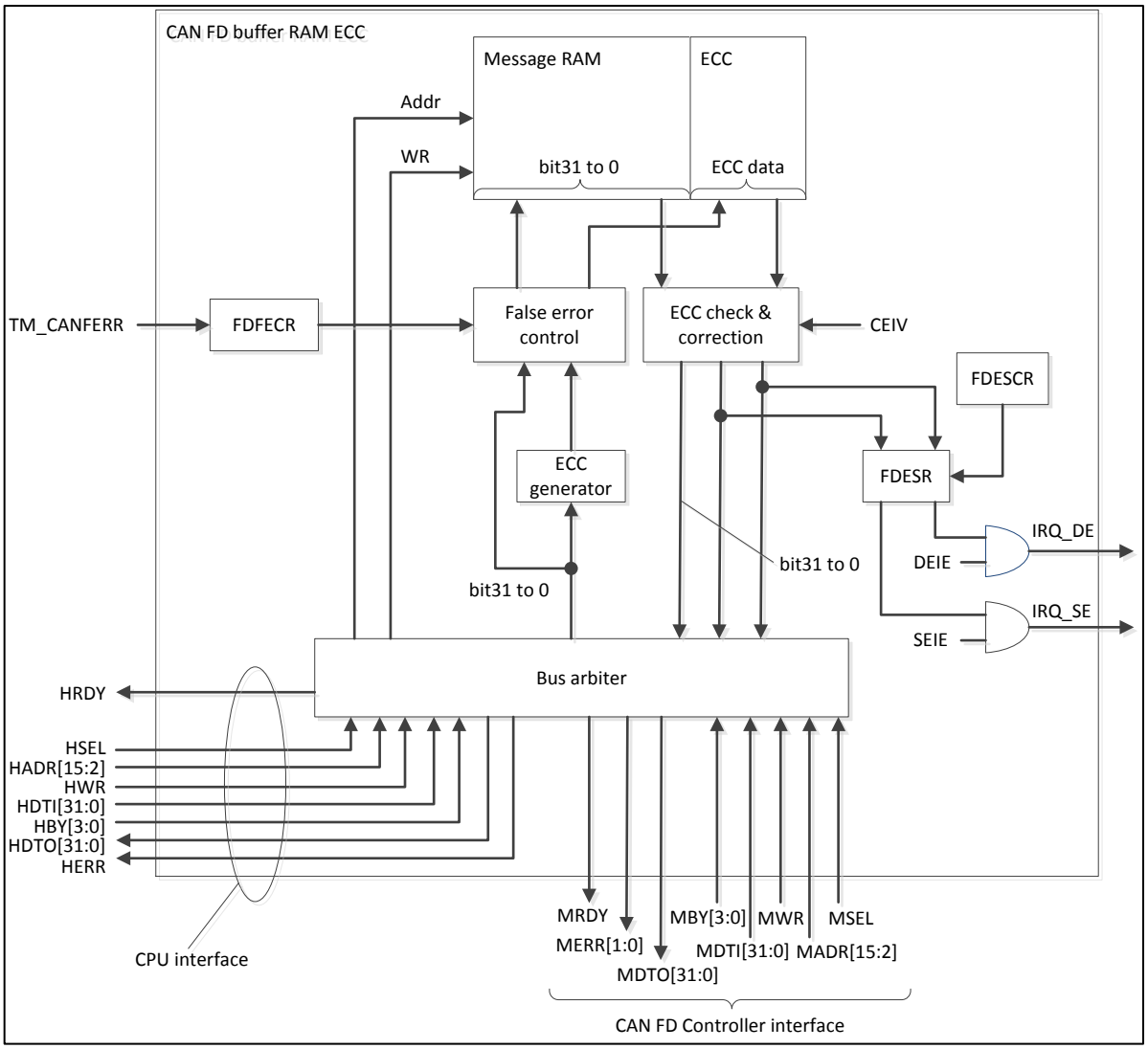
Note:

- *ECC error generation function of the message RAM is not supported for Traveo 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 programed 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
FDECR	CAN FD ECC error control register	6.1
FDESR	CAN FD ECC error status register	6.2
FDESCR	CAN FD ECC error status clear register	6.3
FDDEAR	CAN FD ECC double-bit error address register	6.4
FDSEAR	CAN FD ECC single-bit error address register	6.5
FDFECR	CAN FD ECC false error control register	6.6

6.1. CAN FD ECC Error Control Register (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 (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 (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 (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 (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 (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 31: CAN FD Controller

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

CANFD-T00P00R02L00-E1-GBB01

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

- CAN-FD (V3.2.0)
- 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 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

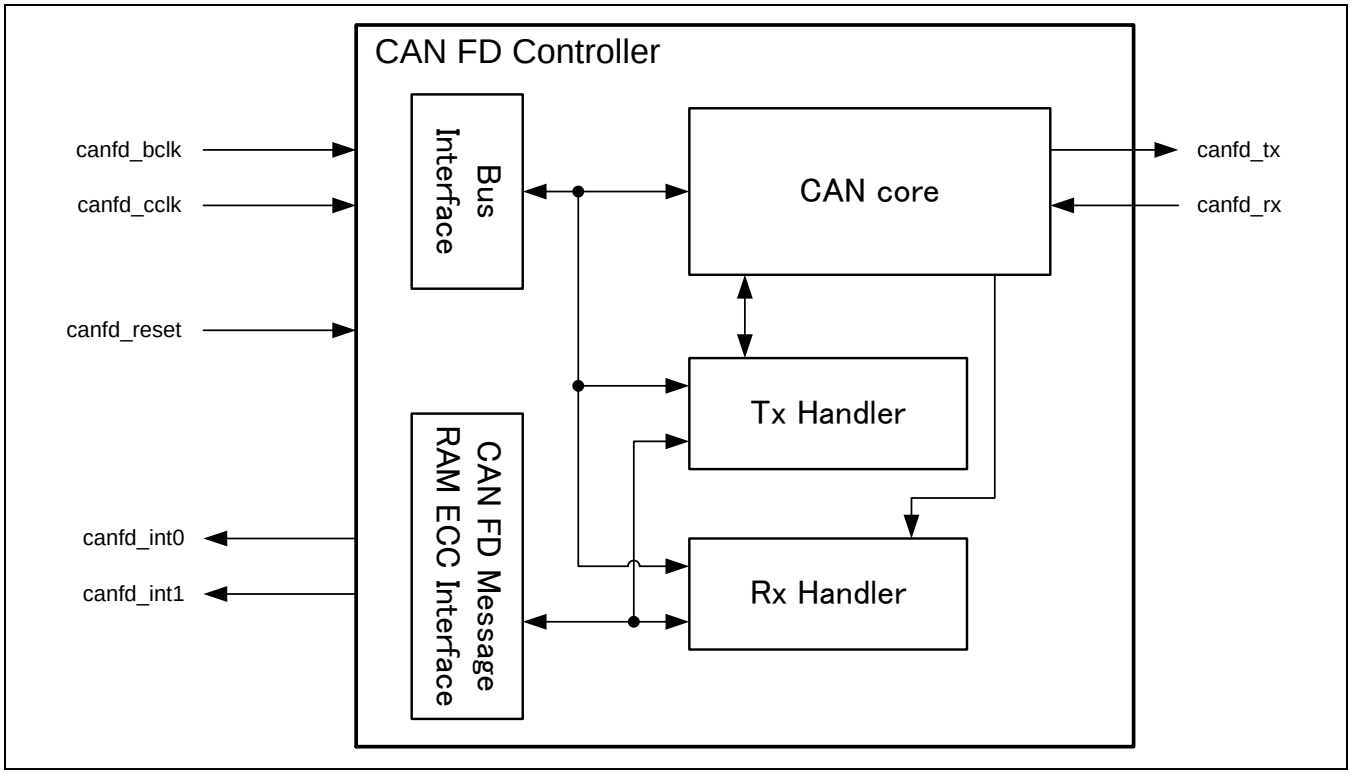
2. Configuration

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 (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		
	250 Kbps	**500 Kbps**	**1 Mbps**
Required Frequency (Bus Clock: canfd_bclk)	5.1 MHz	10.1 MHz	20.3 MHz

Note:

- *CLK_LCP is the source clock of canfd_bclk, which is used as bus clock for CAN-FD macro.*

Table 2-2 Required Frequencies [min] for CAN FD

	Bit Rate					
	250 Kbps (nominal) 1 Mbps (FD)	**250 Kbps (nominal) 2 Mbps (FD)**	**500 Kbps (nominal) 2 Mbps (FD)**	**500 Kbps (nominal) 4 Mbps (FD)**	**500 Kbps (nominal) 5 Mbps (FD)**	**1 Mbps (nominal) 2 Mbps (FD)**
Required Frequency (Bus Clock: canfd_bclk)	9.4 MHz	12.0 MHz	18.8 MHz	23.9 MHz	25.3 MHz	26.3 MHz

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.

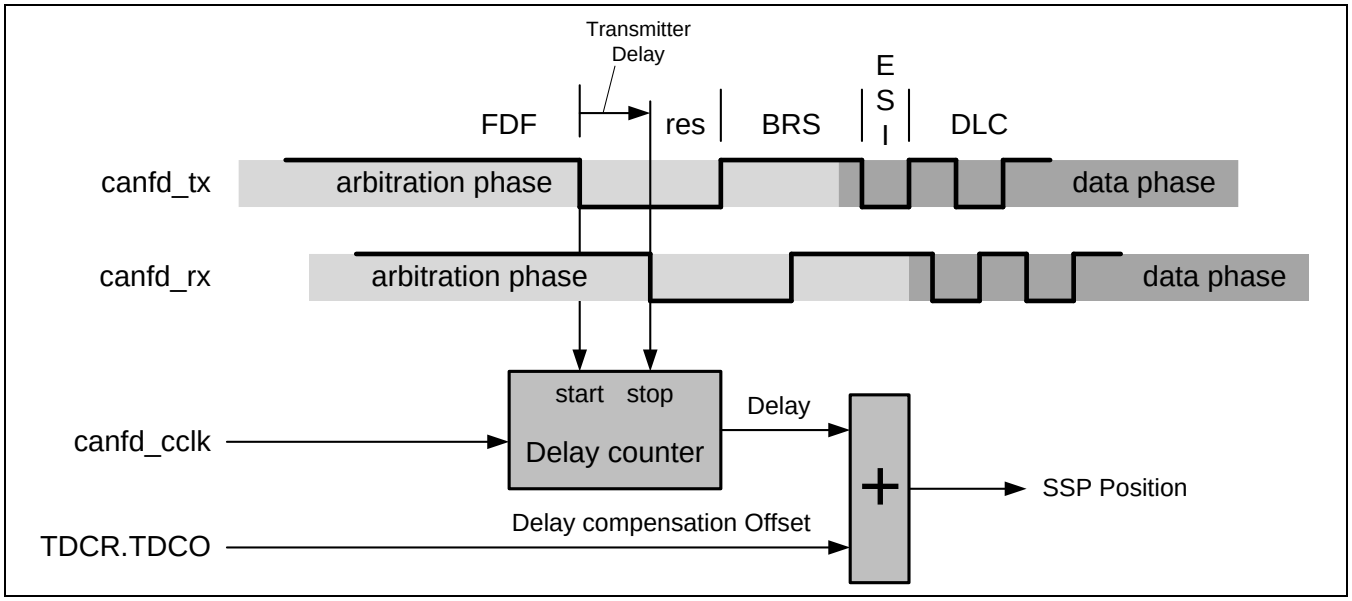
The following boundary conditions have to be considered for the transmitter delay compensation implemented in the CAN FD Controller:

- 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:

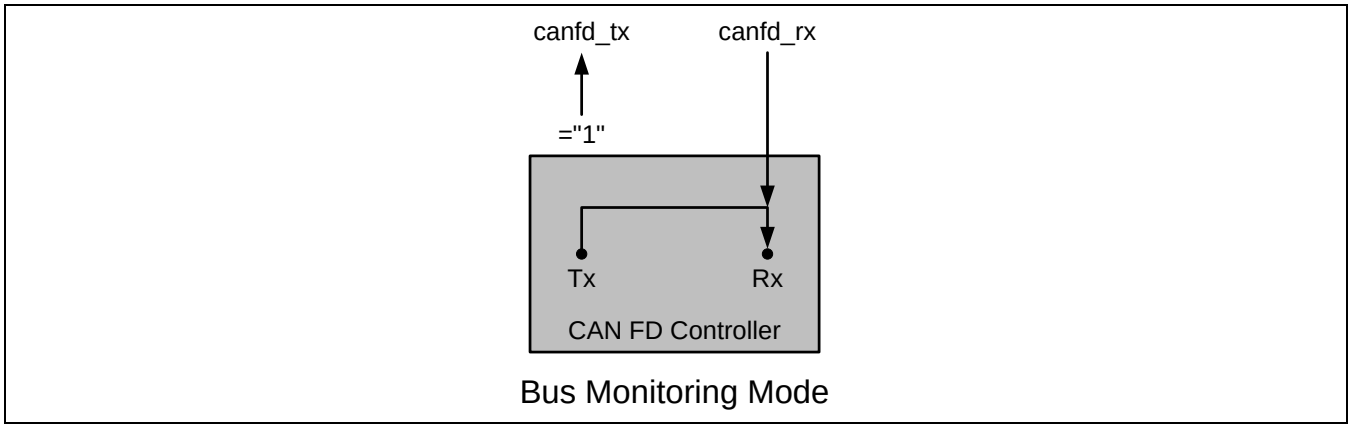
- *The Restricted Operation Mode must not be combined with the Loop Back Mode (internal or external).*

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

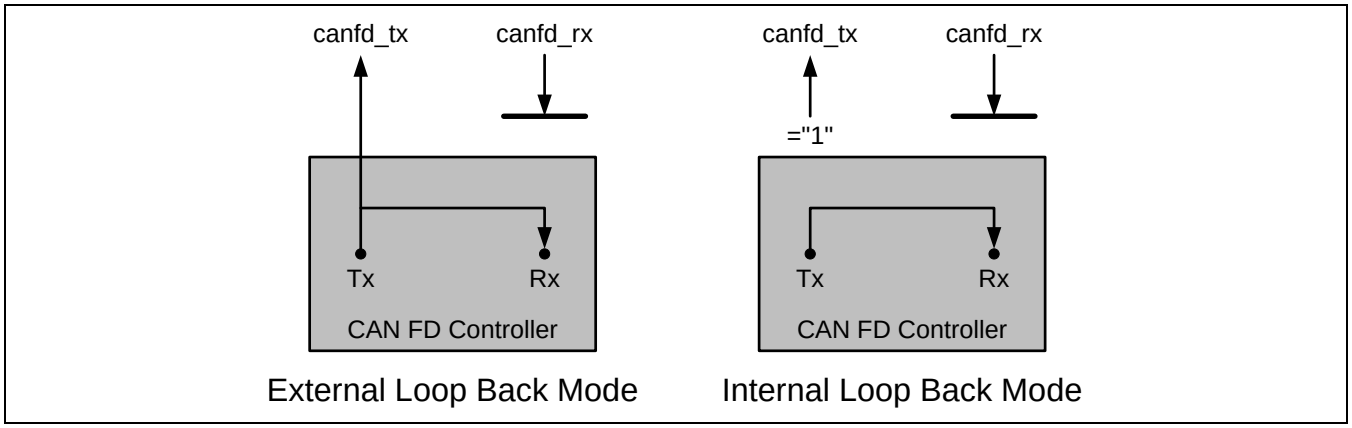
The CAN FD Controller can be set in External Loop Back Mode by programming the Loop Back Mode bit (TEST.LBCK) to one. In External Loop Back Mode, the CAN FD Controller treats its own transmitted messages as received messages and stores them (if they pass acceptance filtering) into a dedicated Rx Buffer 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 (TSCC)" and "5.9.Timestamp Counter Value (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 (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

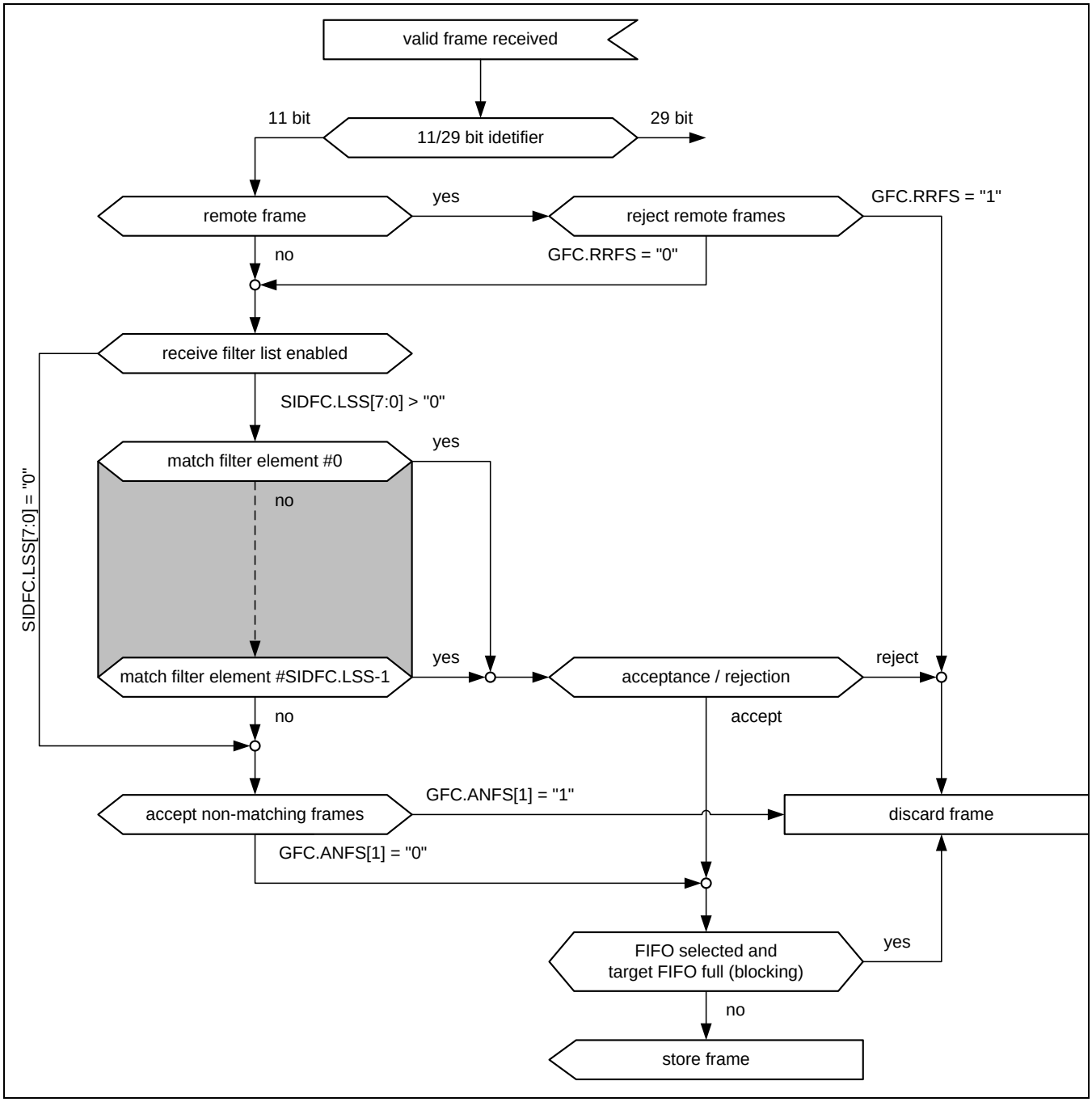
A filter for a single dedicated ID stores received messages in dedicated Rx Buffers if the received ID matches SFID1[10:0]/EFID1[28:0]. SFID2[10:0]/EFID2[28:0] is used to configure the offset to the Rx Buffer Start Address (RXBC.RBSA[15:2]) of where the message is to be stored, and also to define how 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



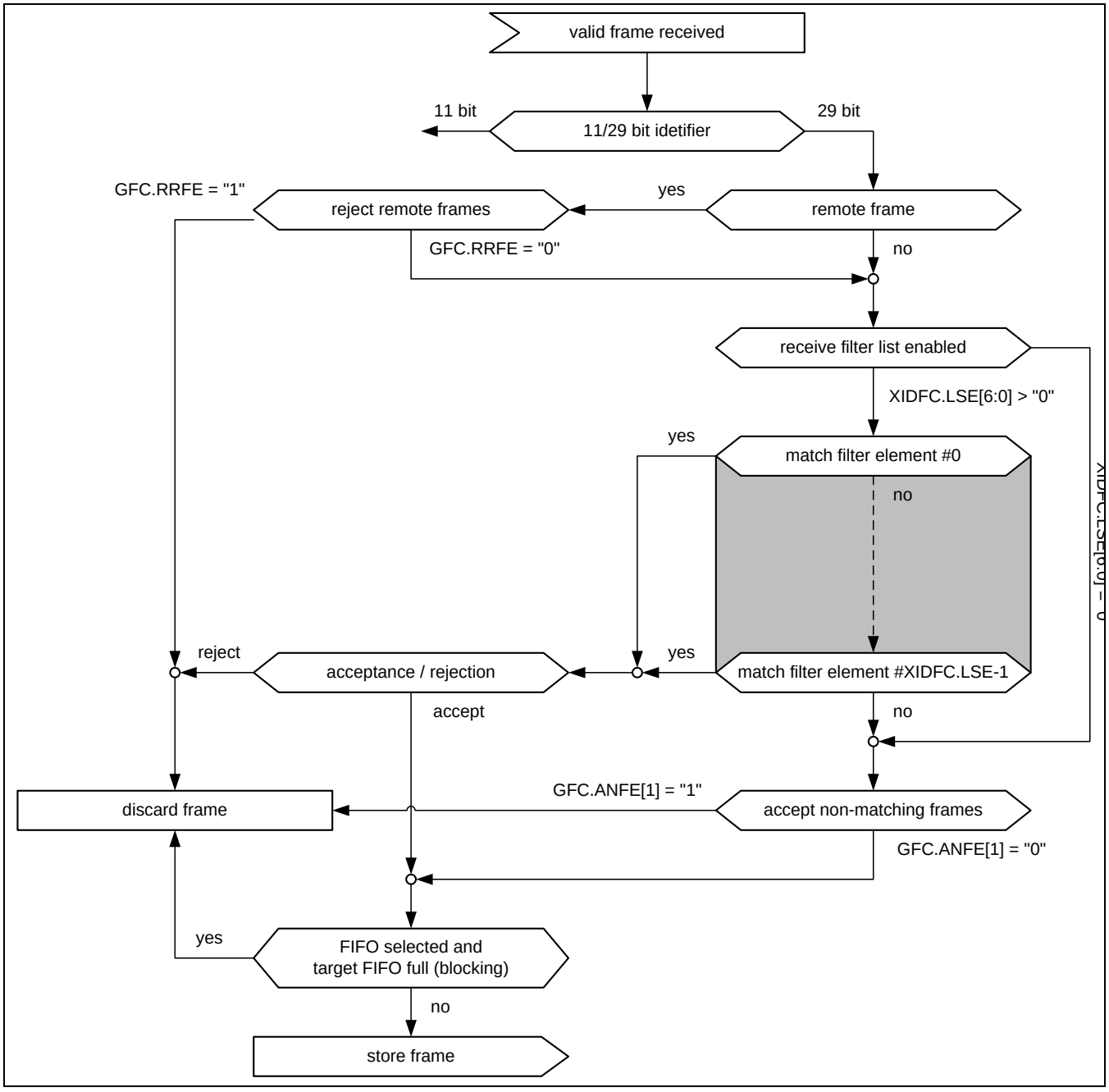
3.4.1.3 Extended Message ID Filtering

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



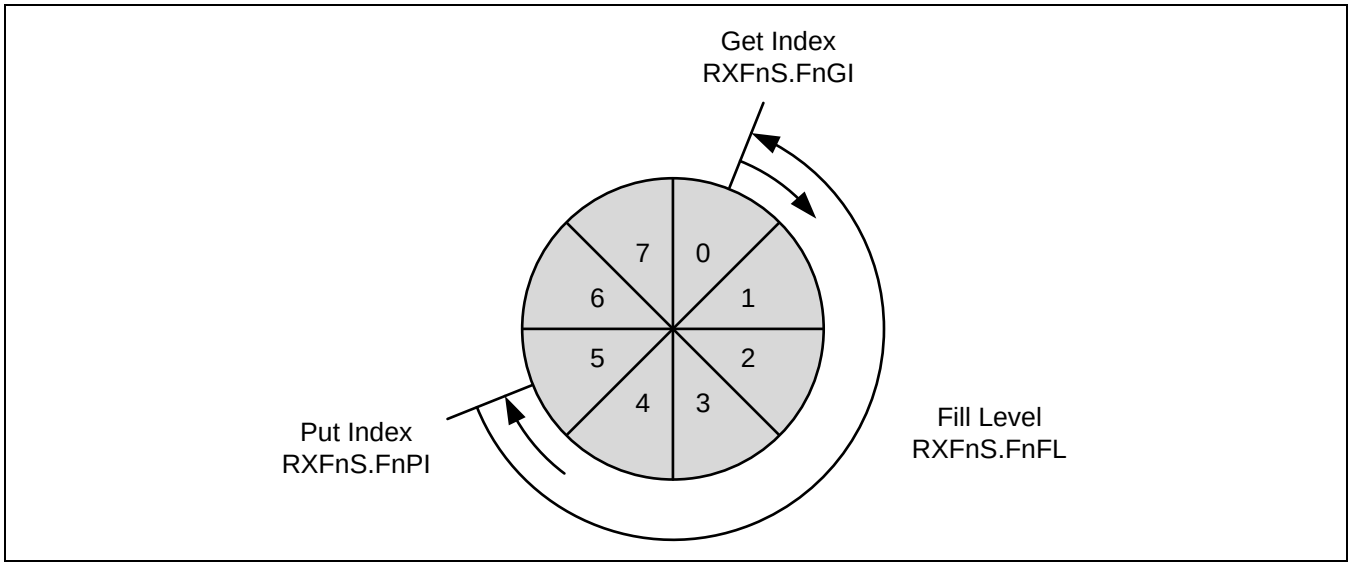
3.4.2. Rx FIFOs

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.

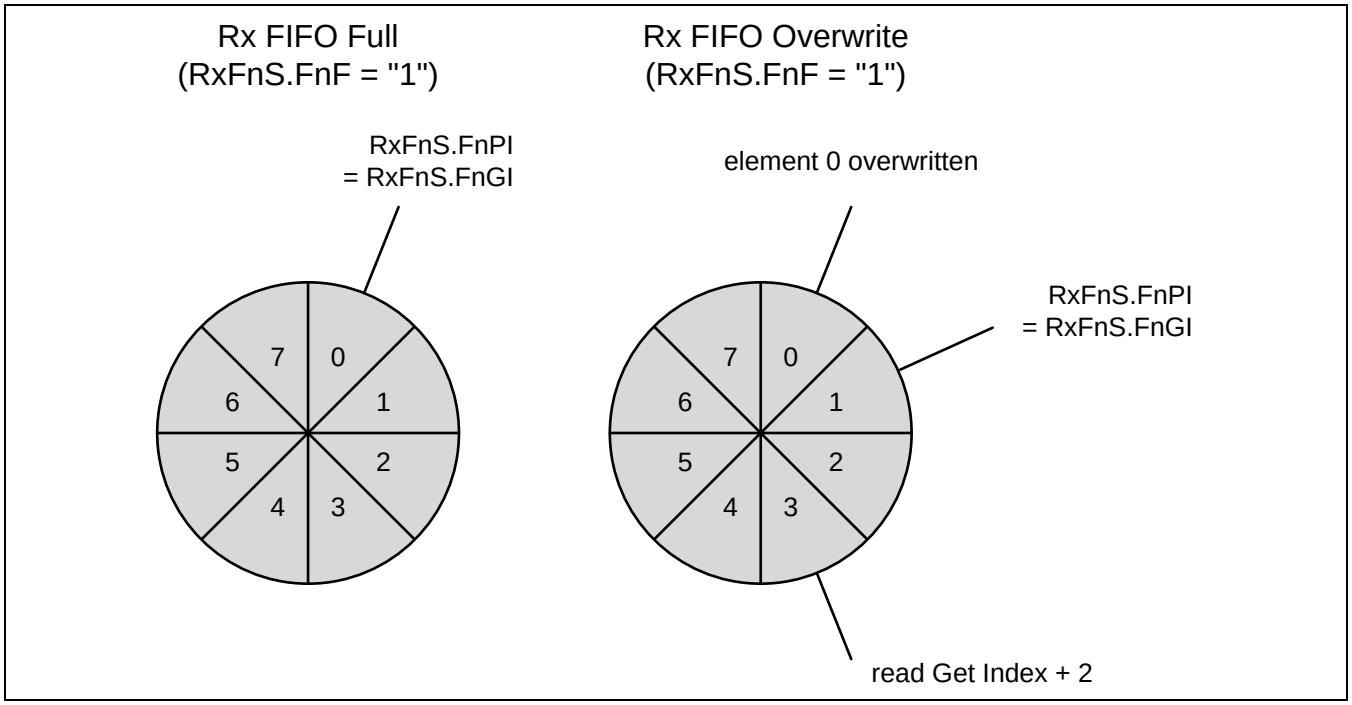
3.4.2.2 Rx FIFO Overwrite Mode

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:

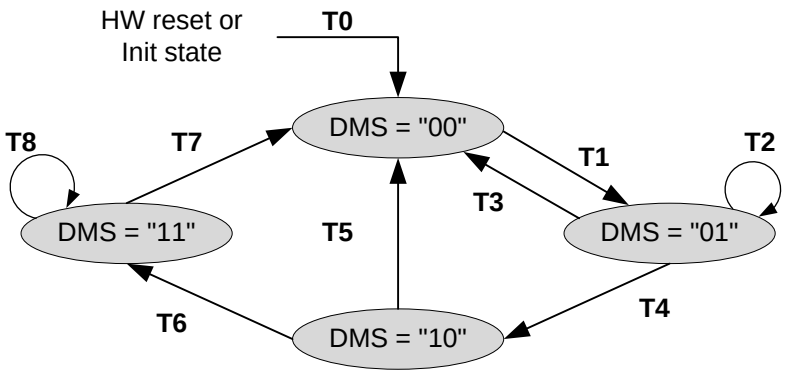
- *Debug on CAN feature is not supported for Traveo Platform.*
- *DMA transfer request is not supported also.*

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 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]).

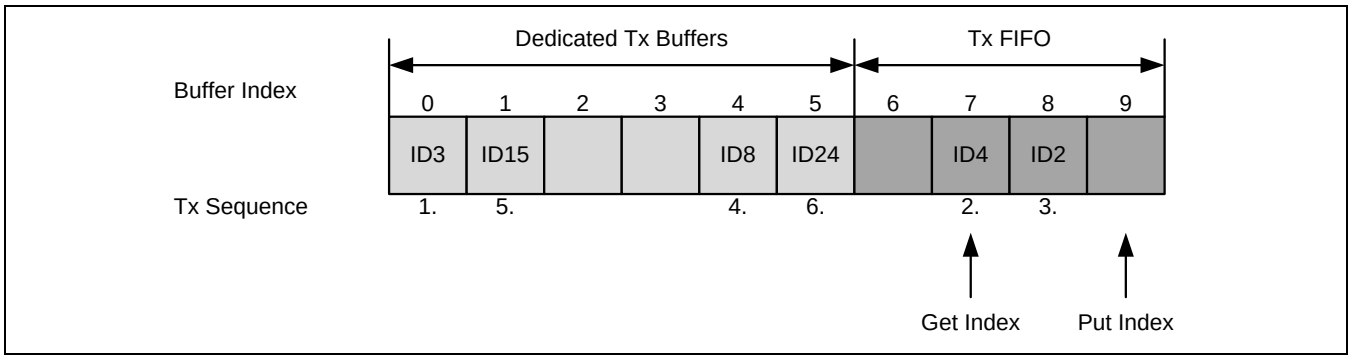
3.5.5. Mixed Dedicated Tx Buffers/Tx FIFO

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

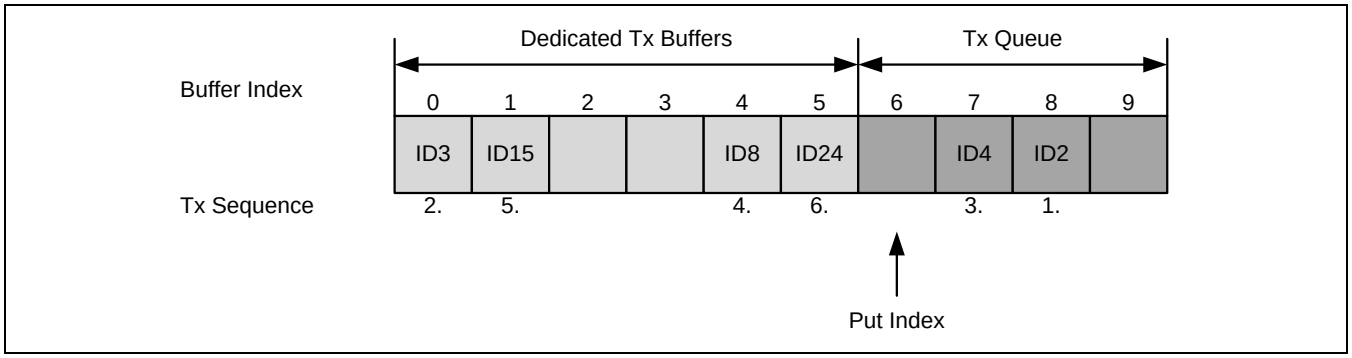
3.5.6. Mixed Dedicated Tx Buffers/Tx Queue

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.

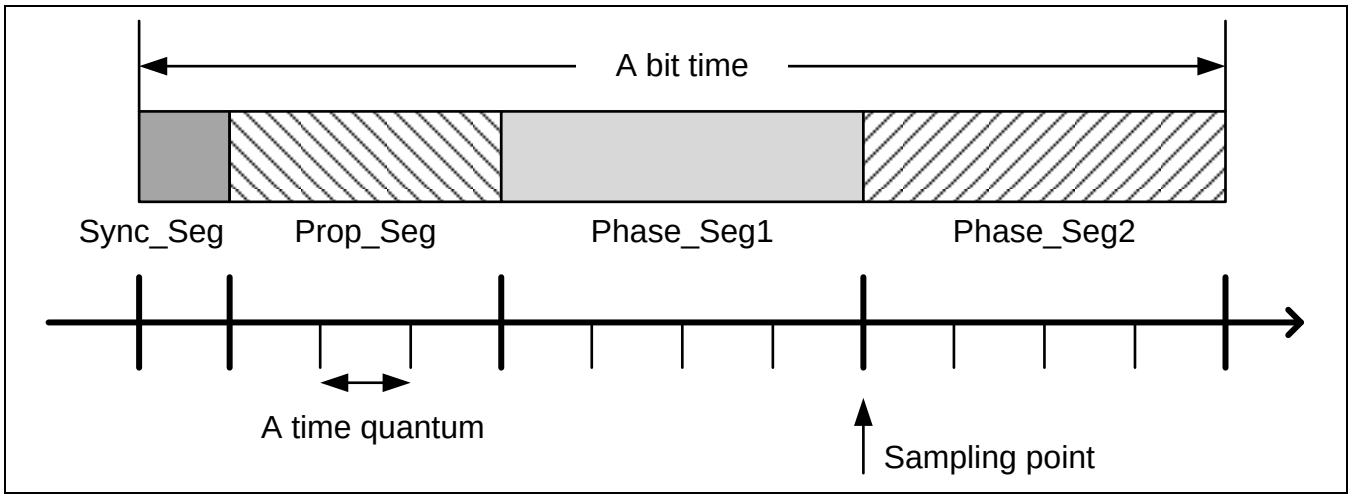
The oscillator frequencies vary slightly because of changes in temperature or voltage, or deterioration of components. As long as the frequencies vary only within the tolerance range of the oscillators, the CAN nodes can compensate for the different bit rates by resynchronizing to the bit stream.

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	16tq	1	20tq	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 tqds per bit time	DBTP.DBRP + 1	Number of tqds per bit time	DBTP.DBRP + 1	Number of tqds per bit time	DBTP.DBRP + 1	Number of tqds per bit time	DBTP.DBRP + 1	Number of tqds per bit time	DBTP.DBRP + 1	Number of tqds 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.*

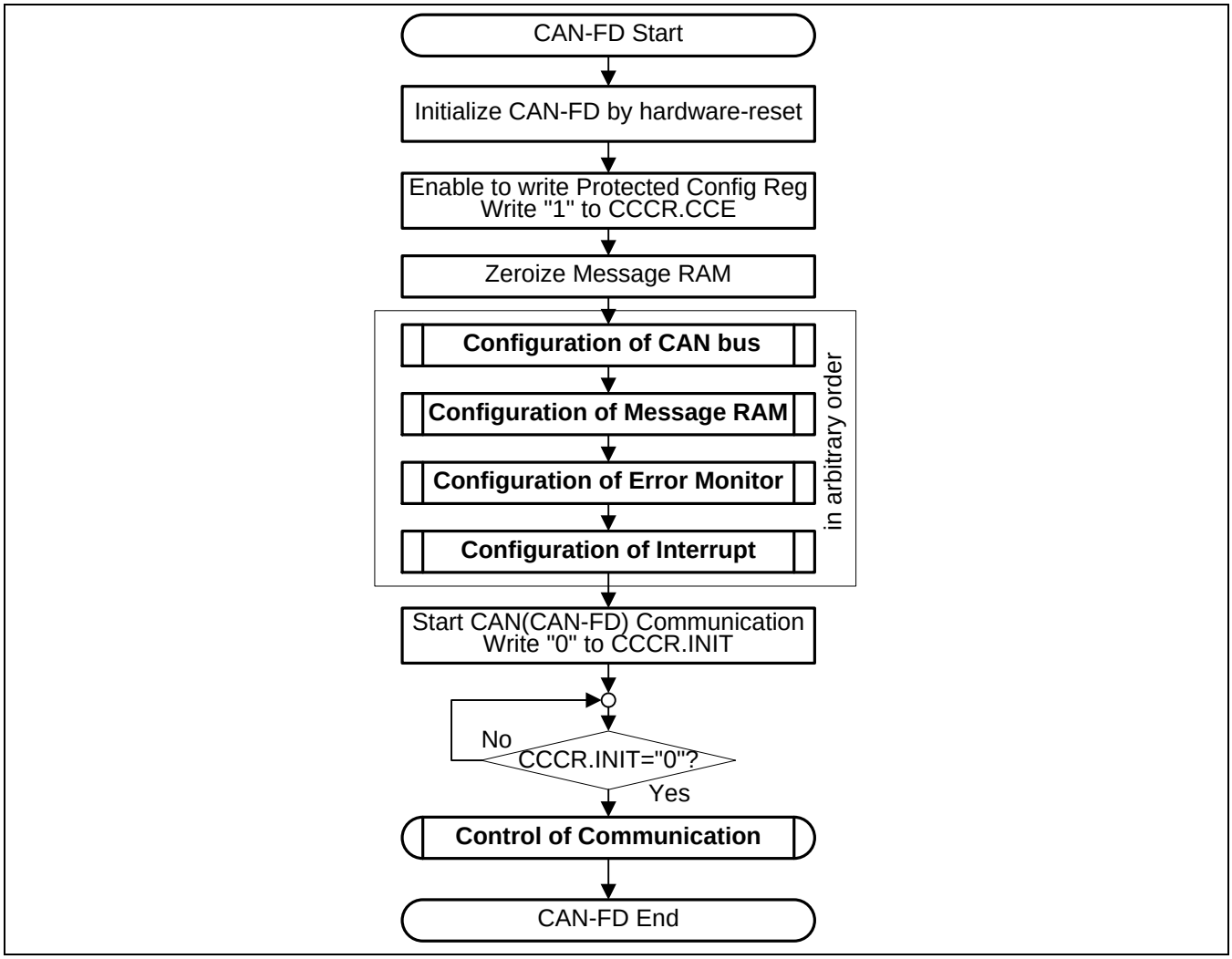
4. Setup Procedure Examples

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.

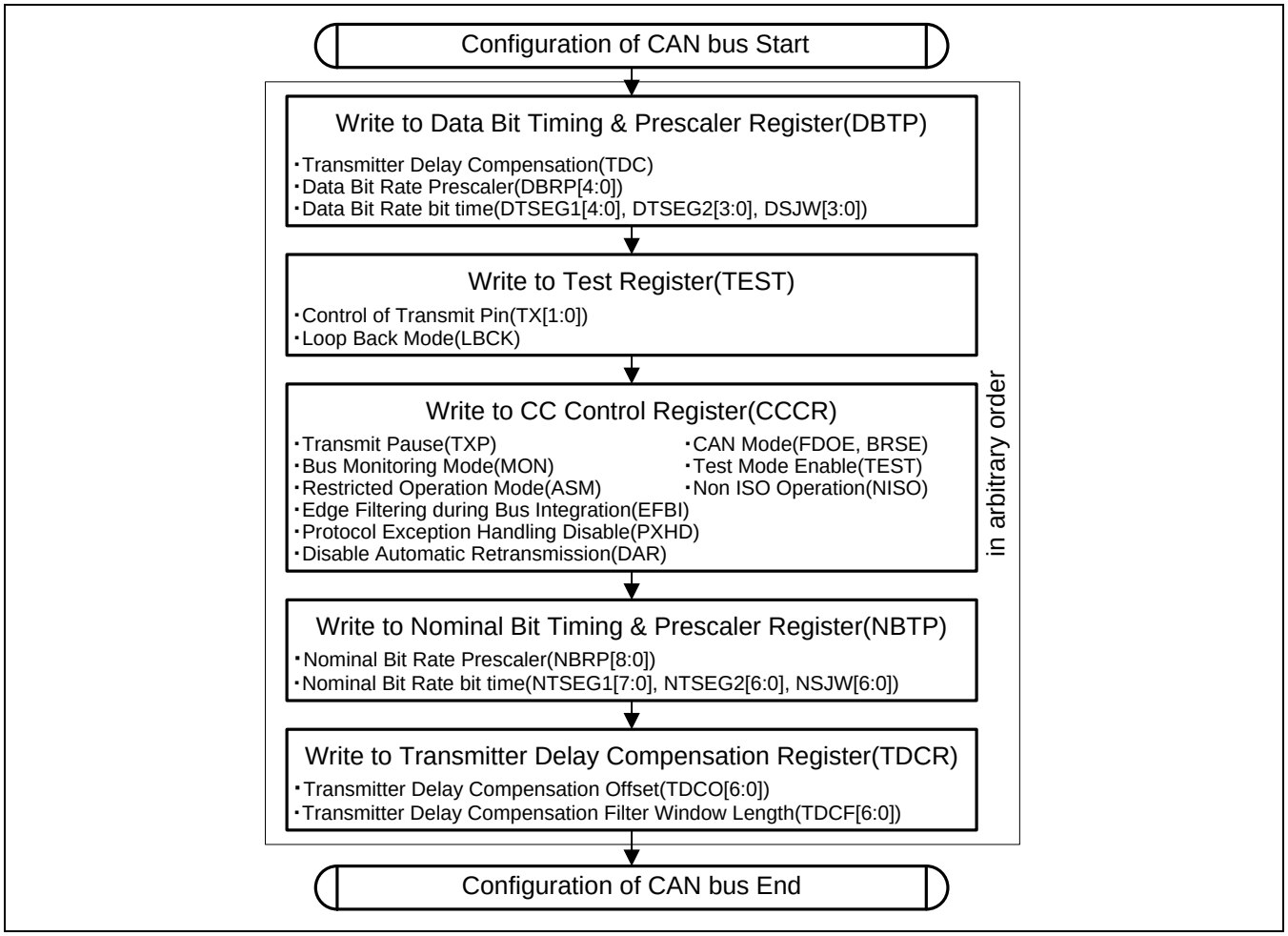
Figure 4-1 General Program Flow



4.1. Configuration of CAN Bus

Figure 4-2 lists what to configure in order to set up the CAN bus related functions.

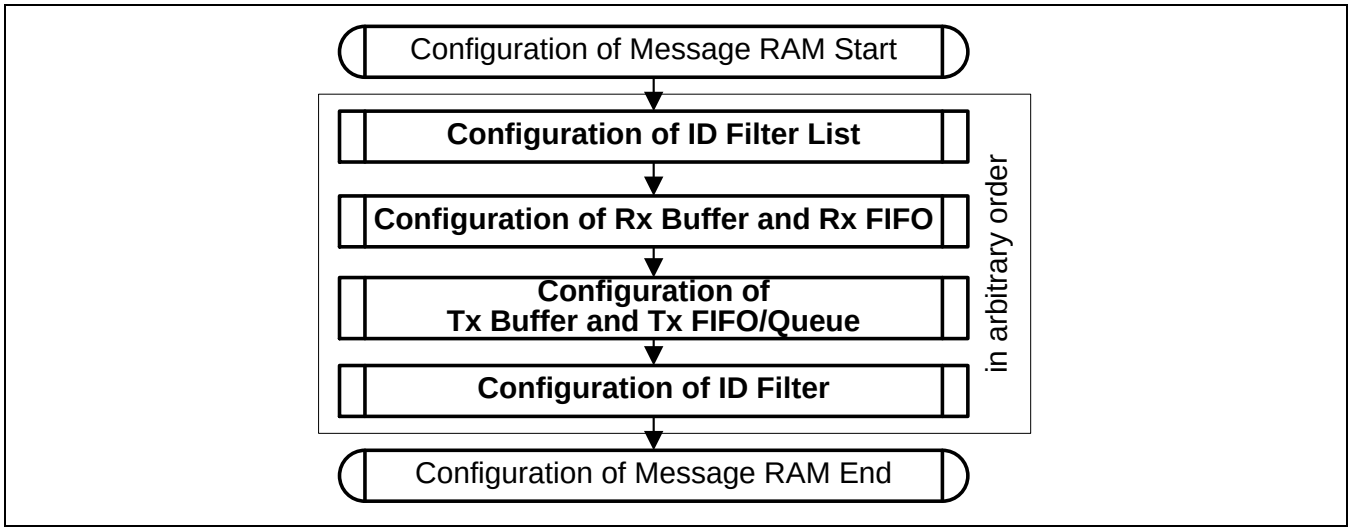
Figure 4-2 Flow Diagram "Configuration of CAN Bus"



4.2. Configuration of Message RAM

Figure 4-3 shows the overall configuration procedure of the Message RAM.

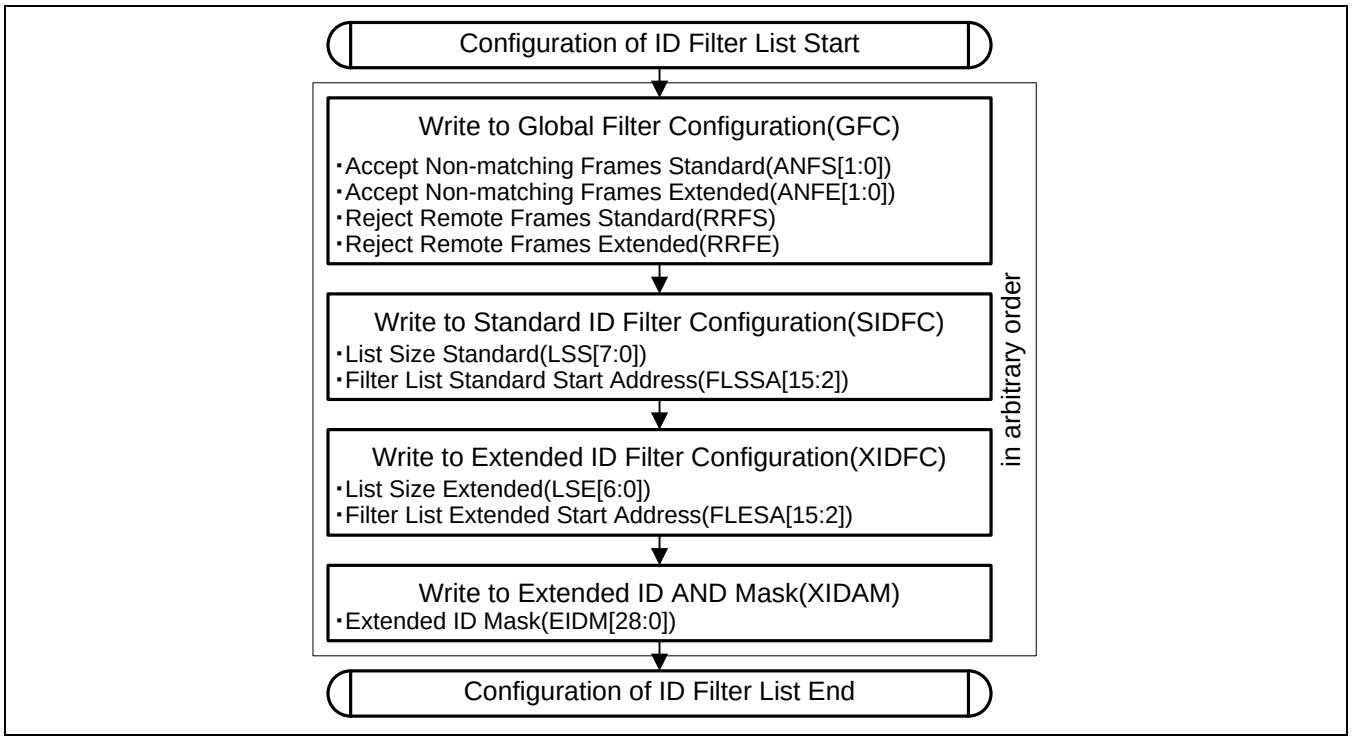
Figure 4-3 Flow Diagram "Configuration of Message RAM"



4.2.1. Configuration of ID Filter List

Figure 4-4 lists what to configure in order to set up the ID Filter lists.

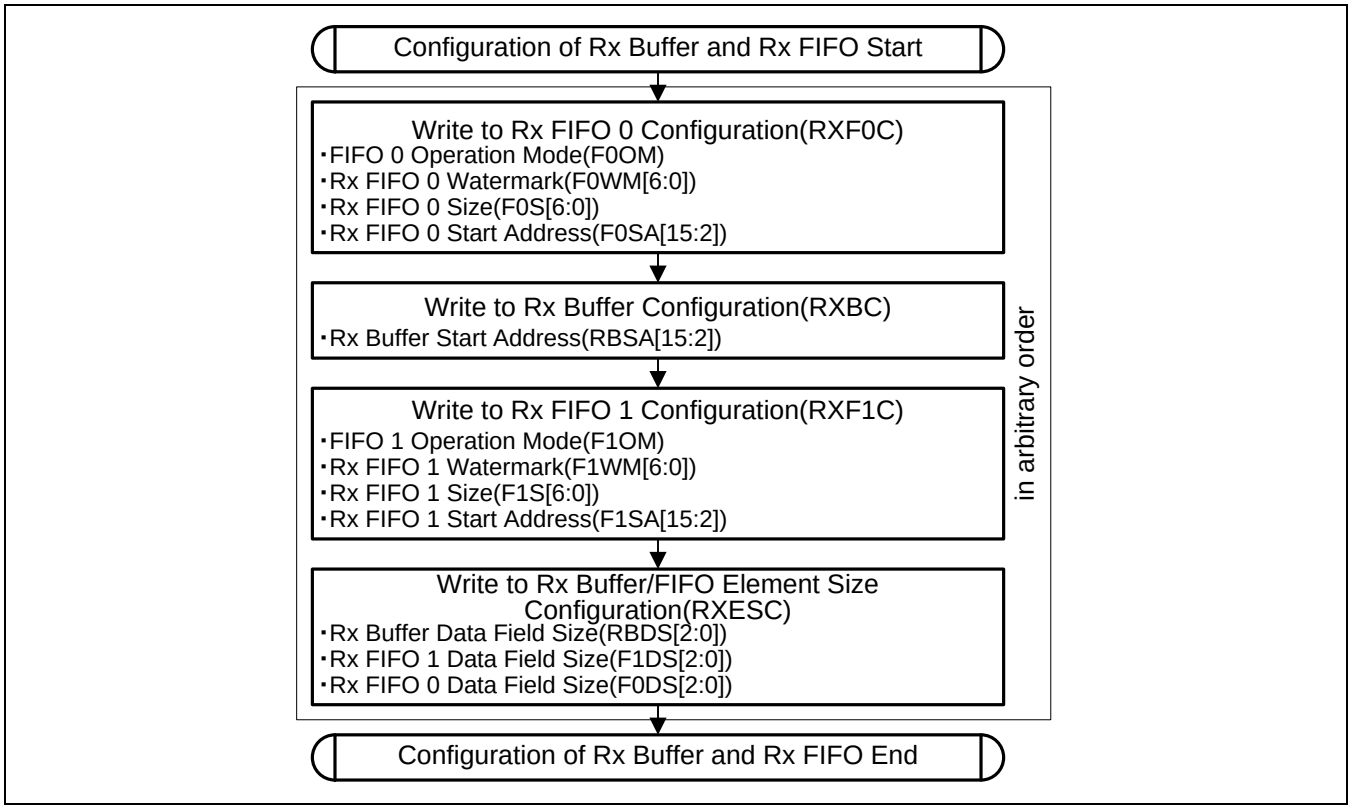
Figure 4-4 Flow Diagram "Configuration of ID Filter List"



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.

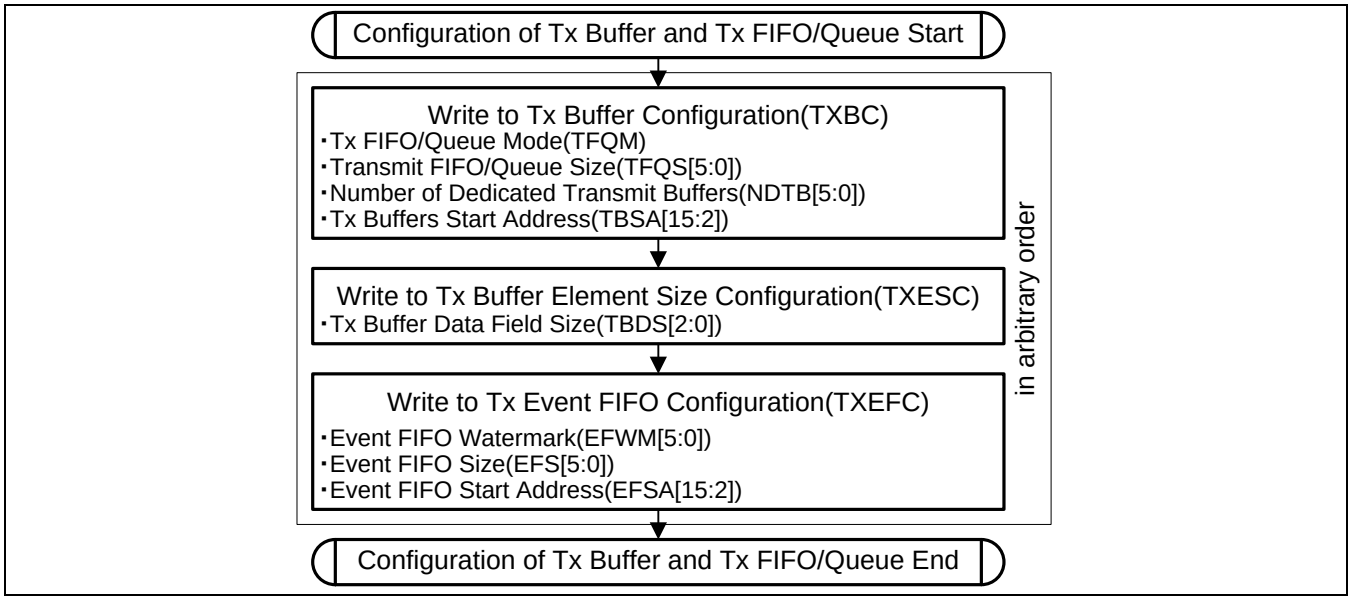
Figure 4-5 Flow Diagram "Configuration of Rx Buffer and Rx FIFO"



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.

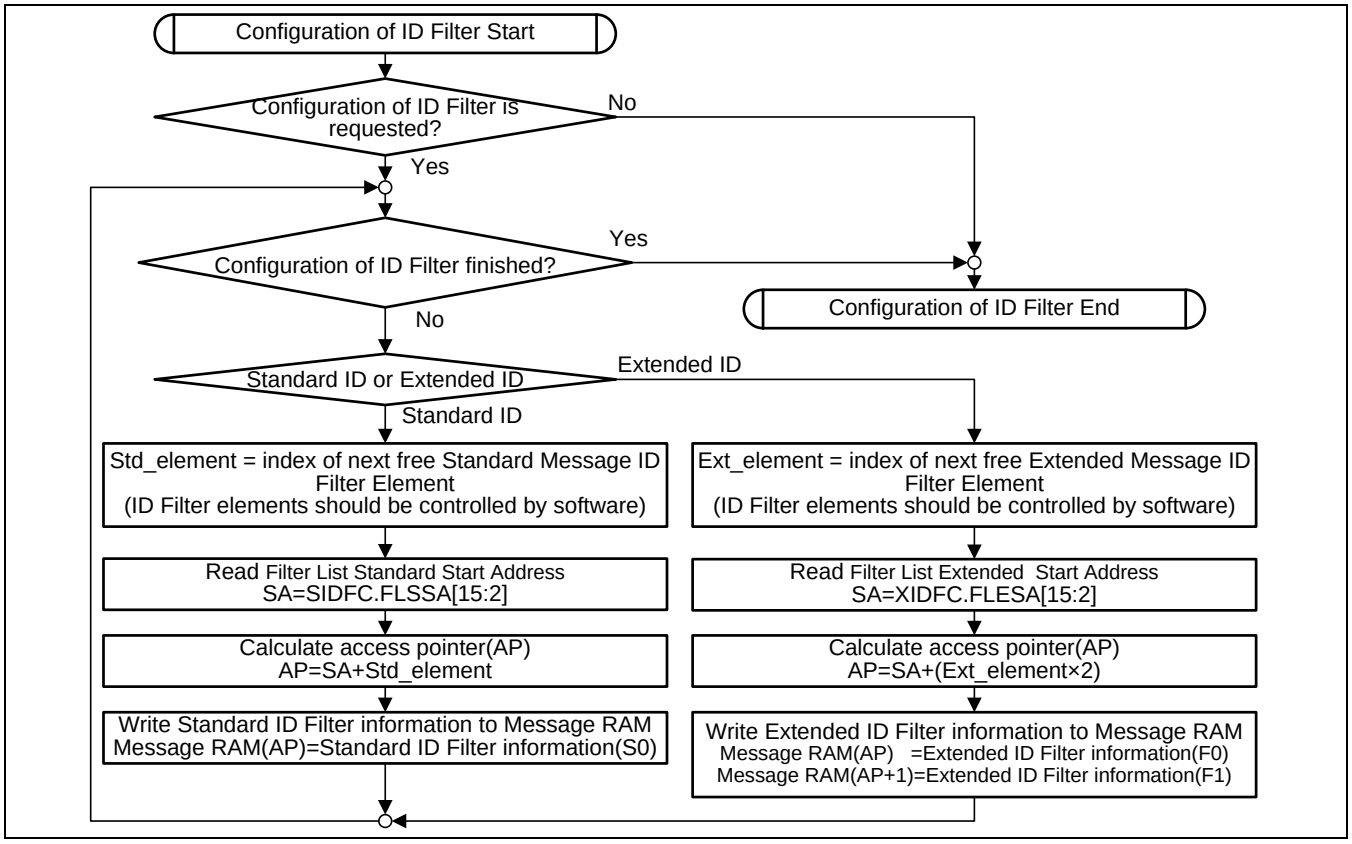
Figure 4-6 Flow Diagram "Configuration of Tx Buffer and Tx FIFO/Queue"



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"



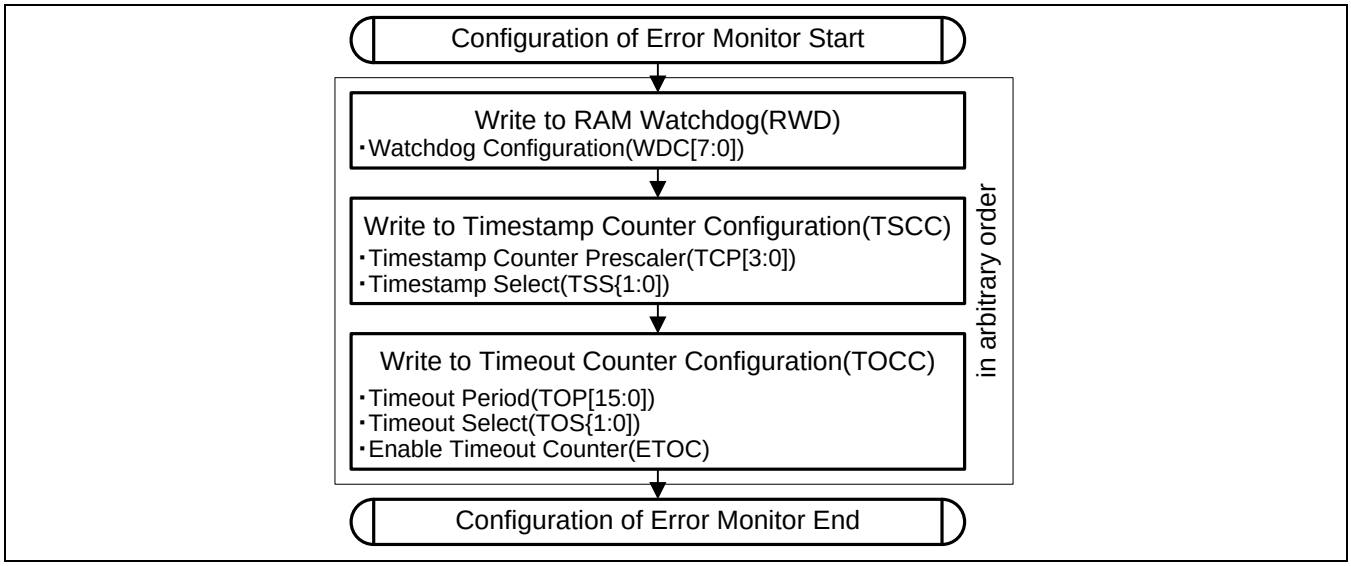
Note:

- *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.

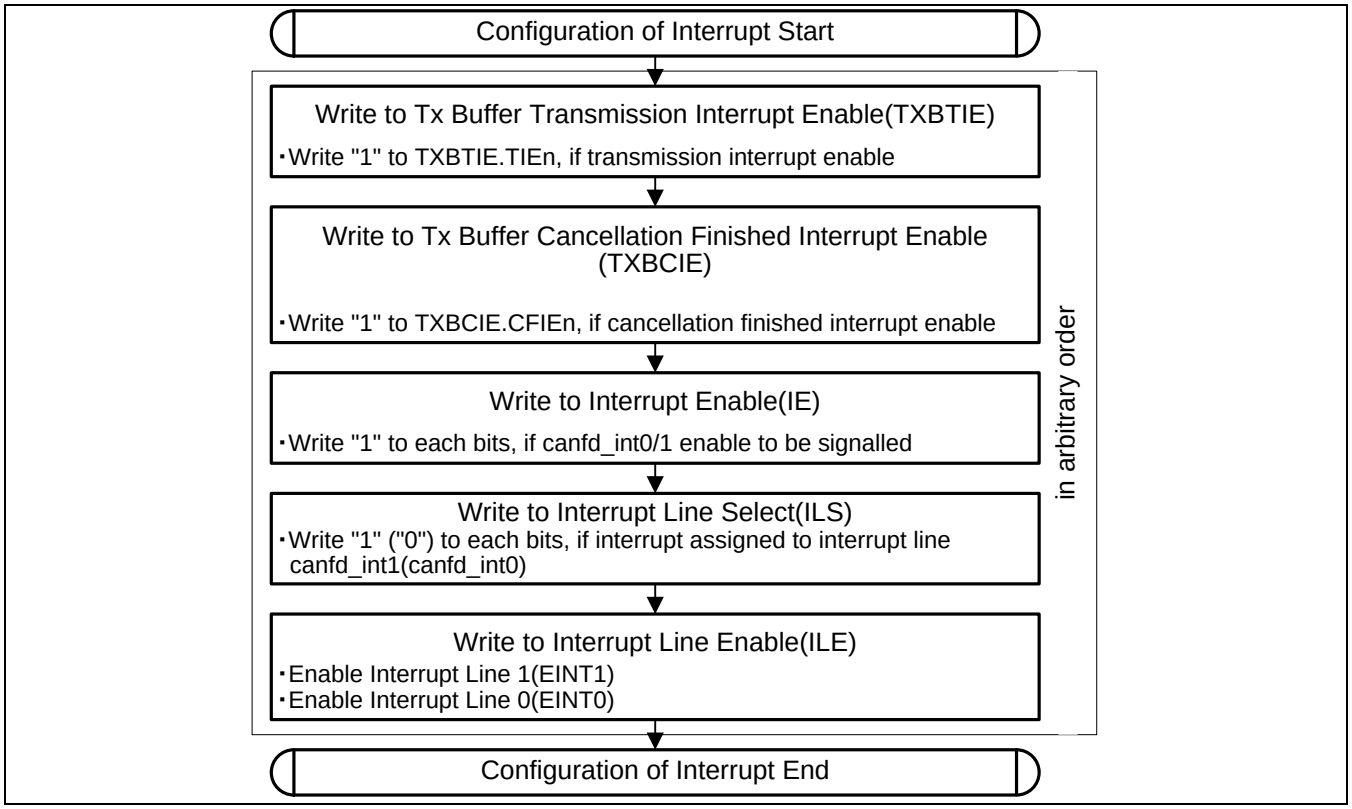
Figure 4-8 Flow Diagram "Configuration of Error Monitor"



4.4. Configuration of Interrupt

Figure 4-9 lists what to configure in order to set up the Interrupt related logic.

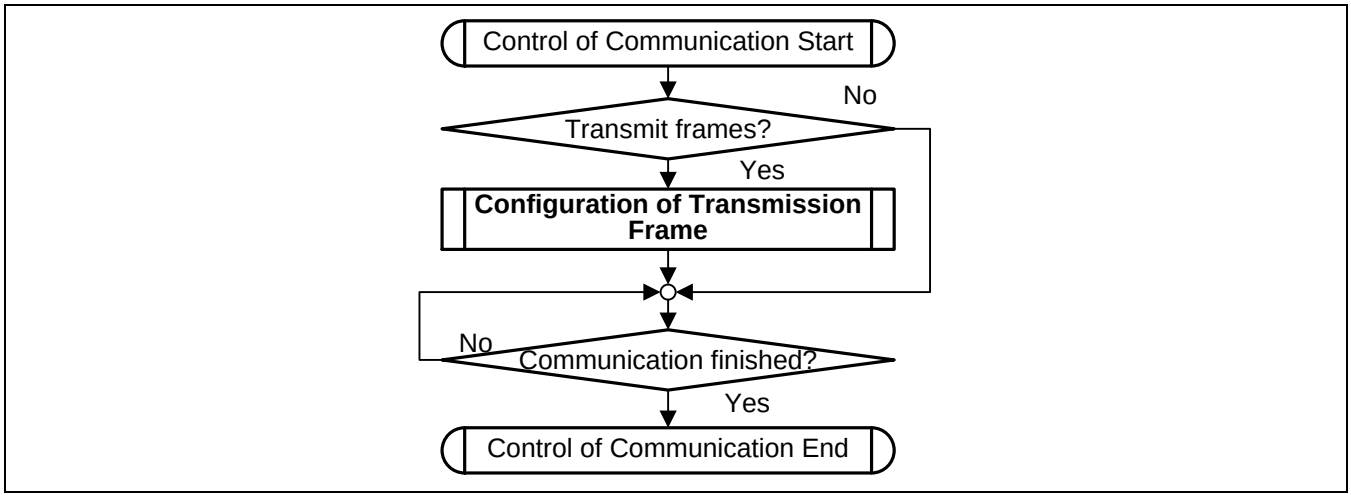
Figure 4-9 Flow Diagram "Configuration of Interrupt"



4.5. Control of Communication

Figure 4-10 shows the overall configuration procedure for communication on the CAN bus.

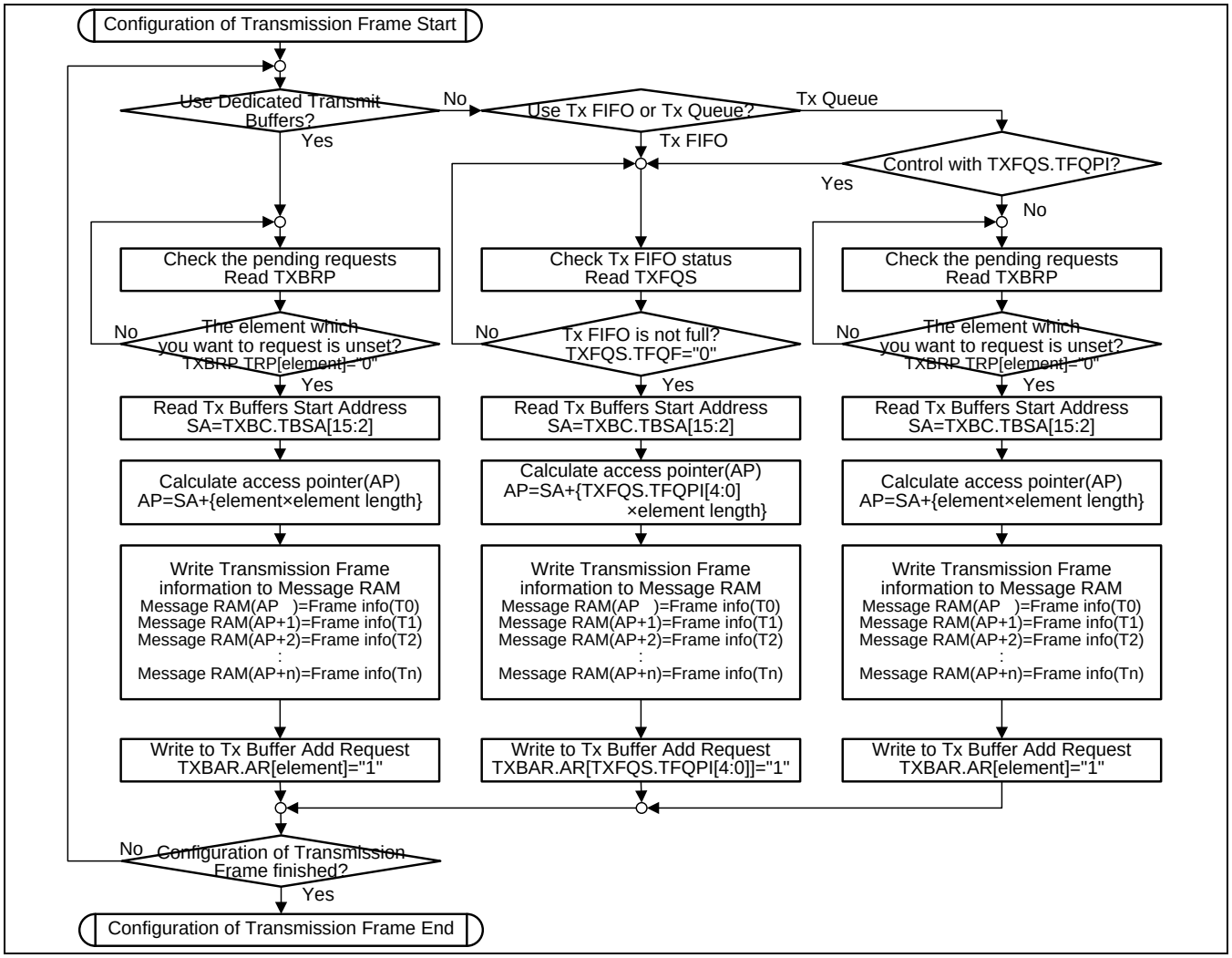
Figure 4-10 Flow Diagram "Control of Communication"



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"



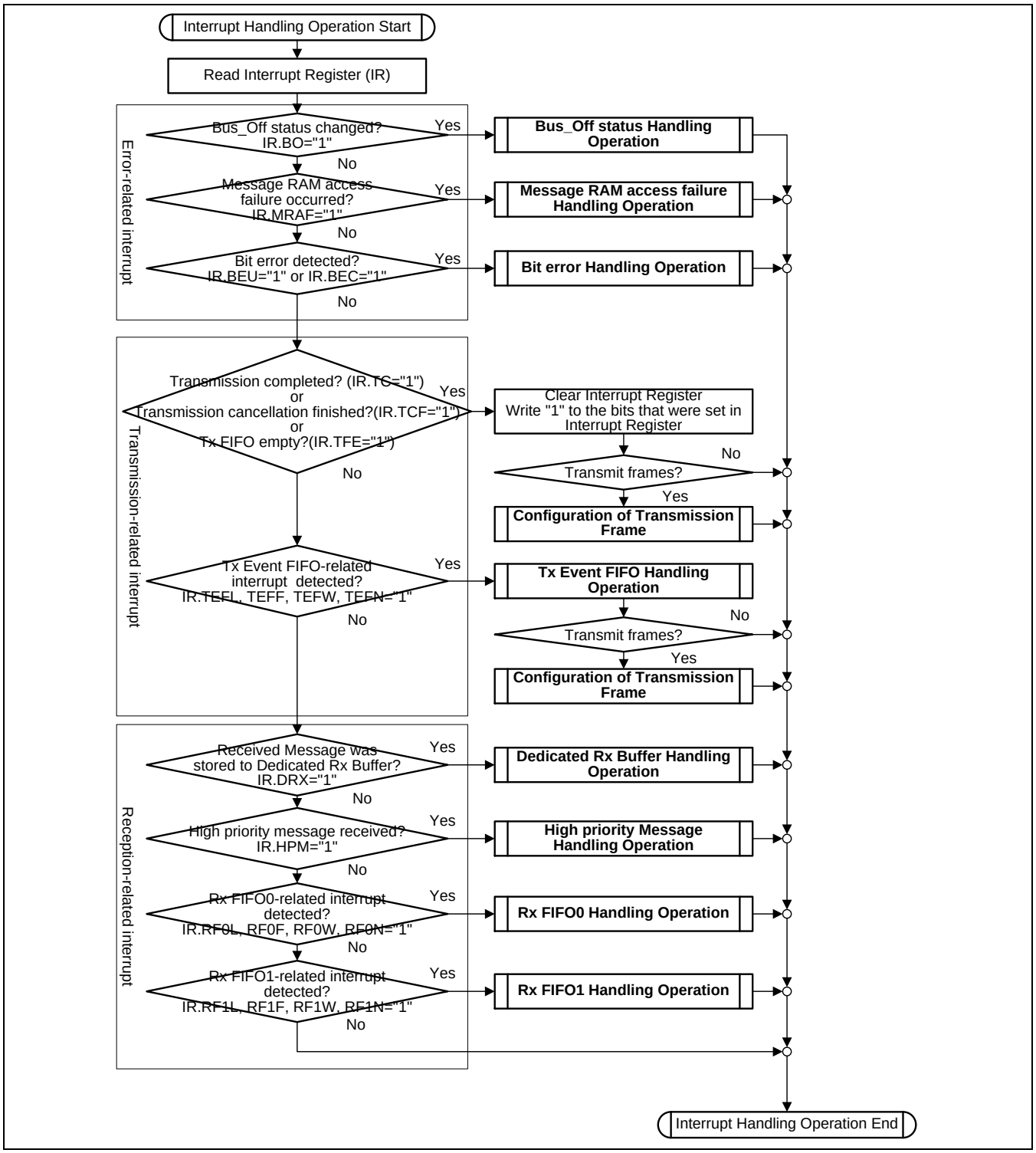
Notes:

- *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.

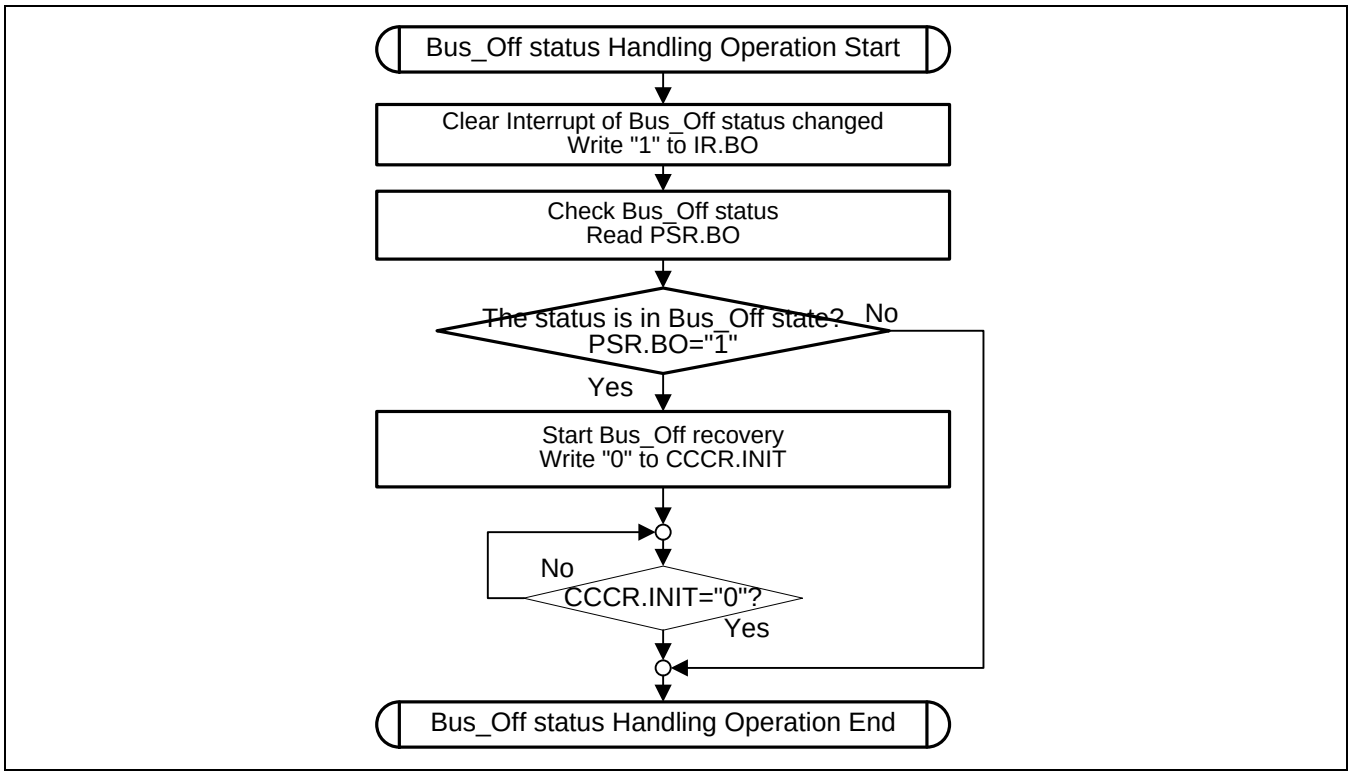
Figure 4-12 Flow Diagram "Interrupt Handling Operation"



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"



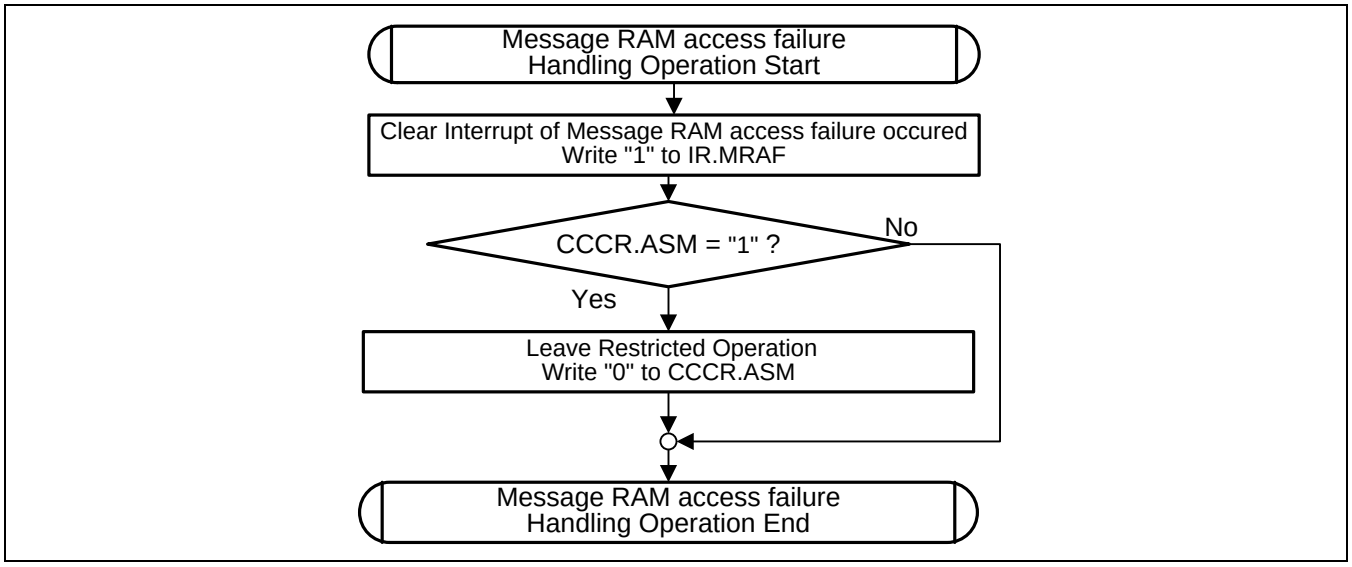
Note:

- *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"



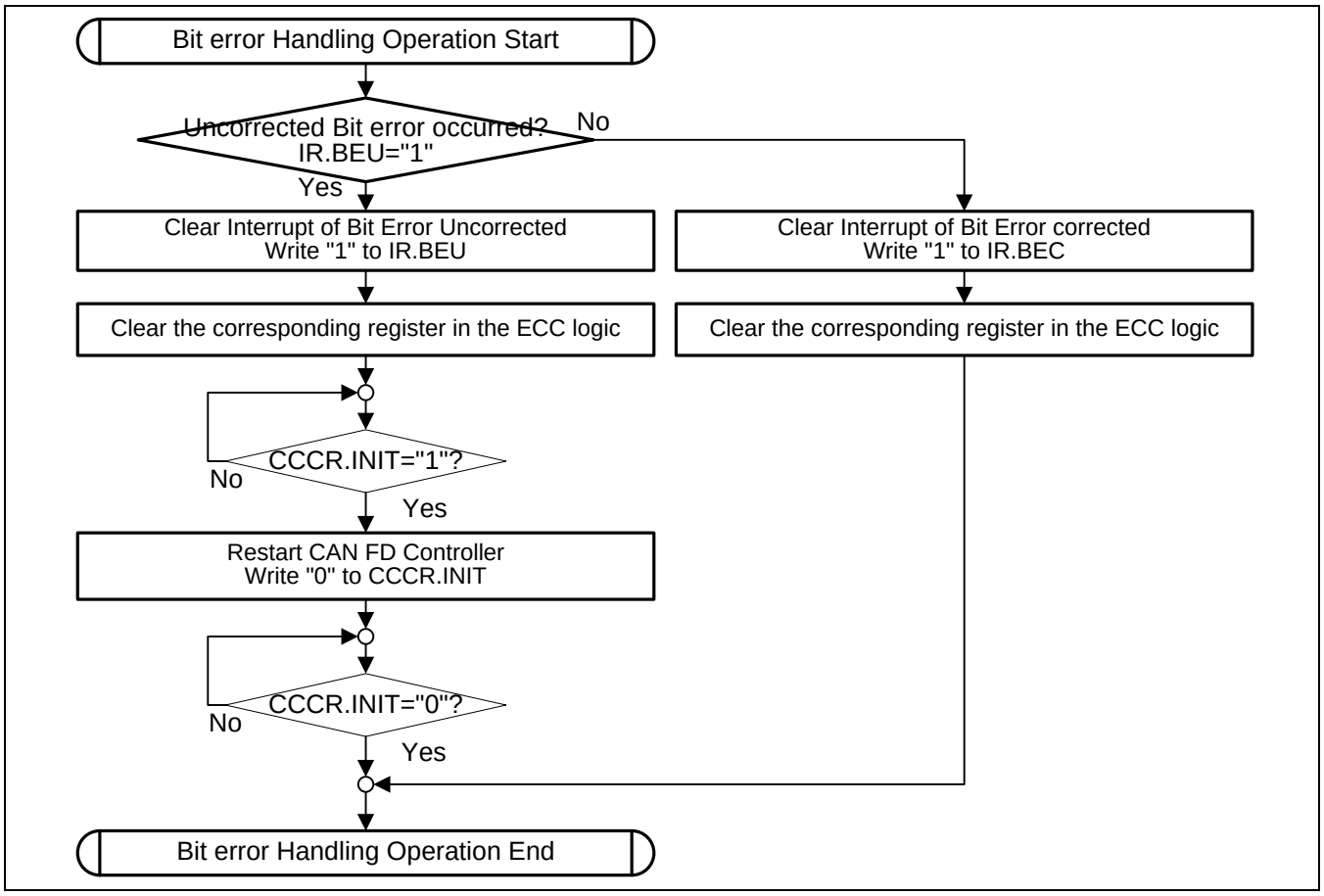
Note:

- *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"



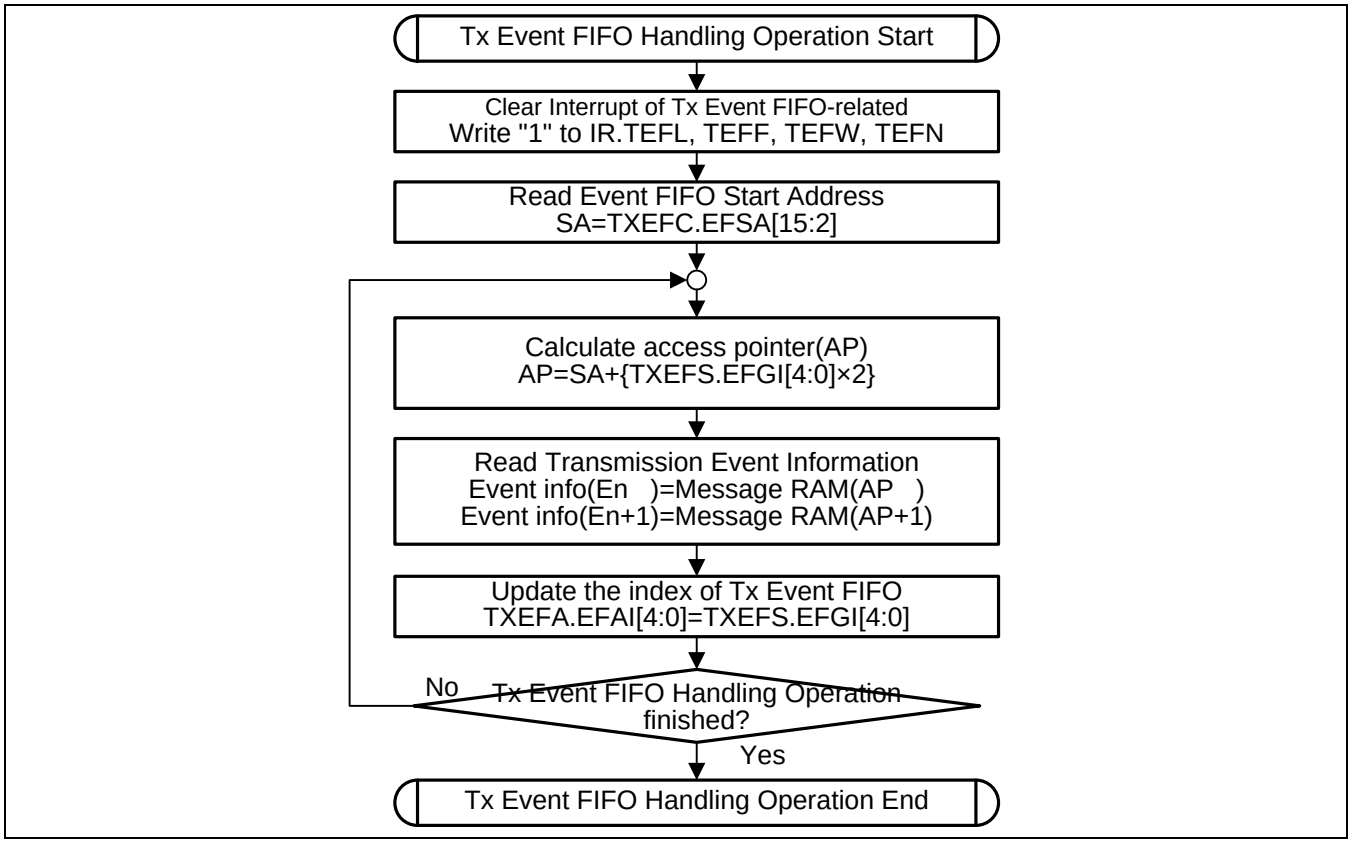
Note:

- *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"



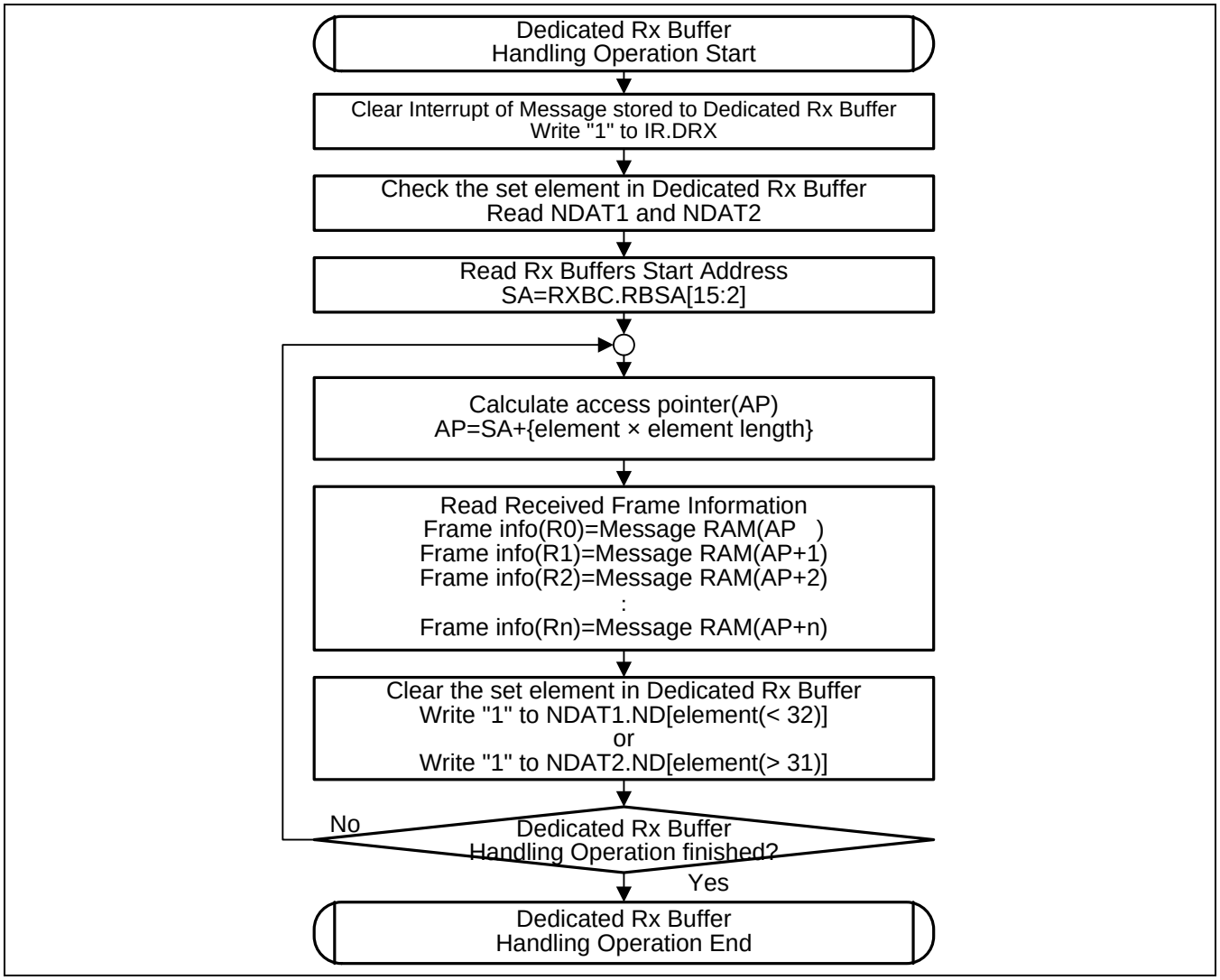
Note:

- *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"



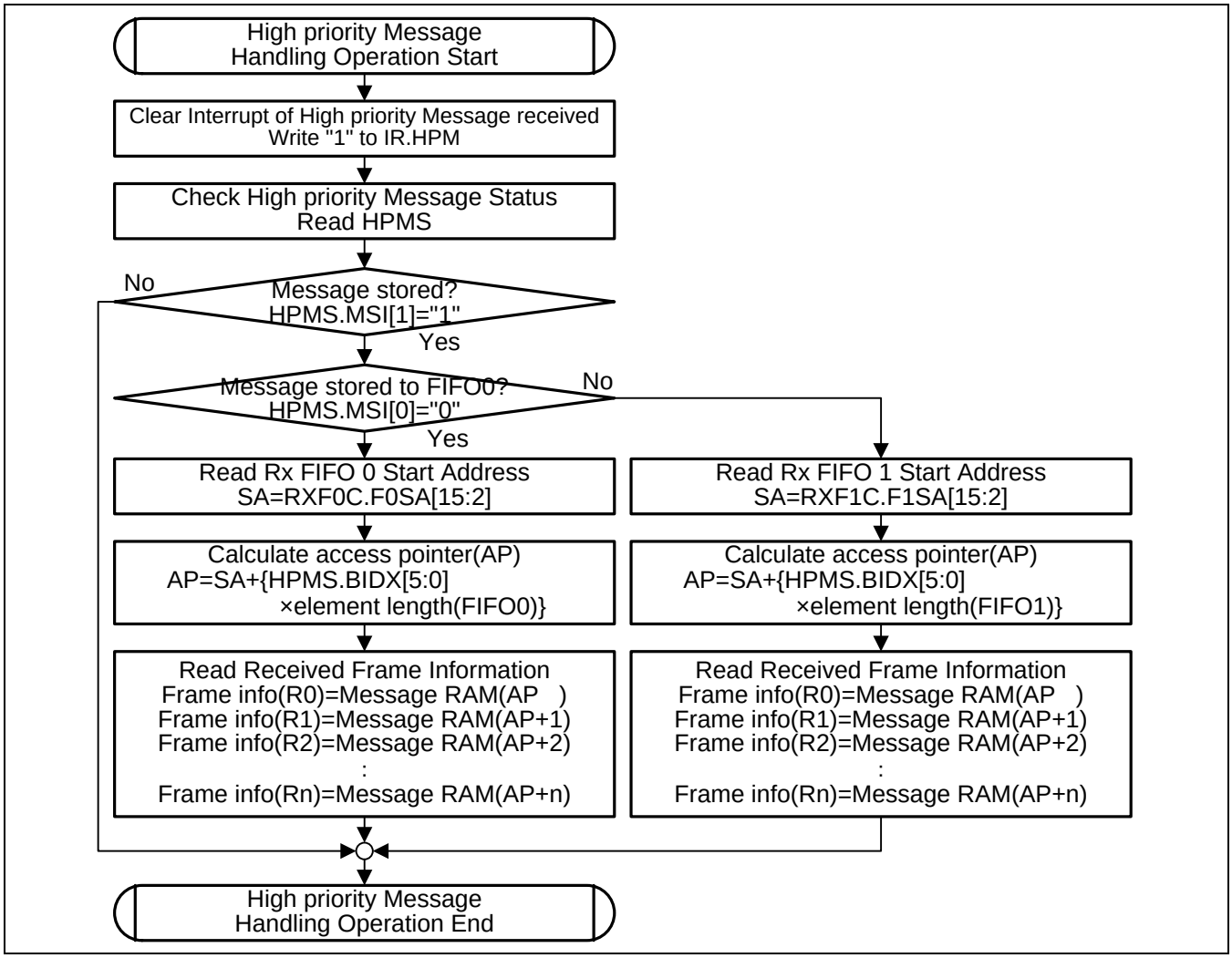
Note:

- *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"



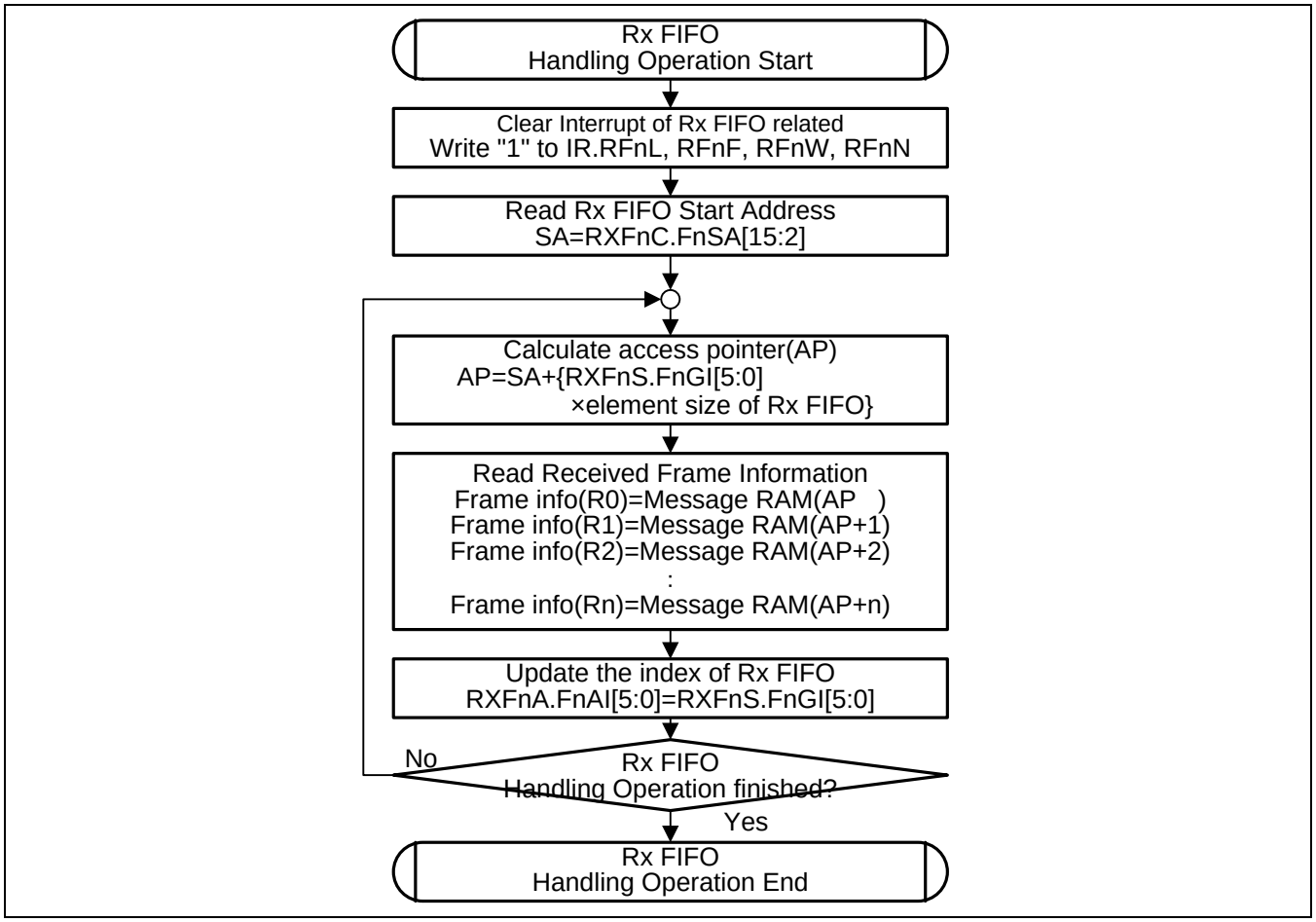
Note:

- *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
CCCR	CC Control Register	5.6

Abbreviation	Register Name	Reference
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

5.1. Core Release Register (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 5-3 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 (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 (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. 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 (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 (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 (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	R0,W0	R/W	R,WX	R,WX	R/W		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.

[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.

[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 (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							00

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,WX	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
0x00-0x7F	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.

5.8. Timestamp Counter Configuration (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	Reserved. Do not use this setting.
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 (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 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.

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 (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 (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 (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 (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	0xXX	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 (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	R,WX							
Protection Attribute	-							
Initial value	0x00							

bit	23	22	21	20	19	18	17	16
Field	Reserved							
R/W Attribute	R,WX							
Protection Attribute	-							
Initial value	0x00							

bit	15	14	13	12	11	10	9	8
Field	Reserved	TDCO[6:0]						
R/W Attribute	R/W	R/W						
Protection Attribute	-	-						
Initial value	0	0000000						

bit	7	6	5	4	3	2	1	0
Field	Reserved	TDCF[6:0]						
R/W Attribute	R/W	R/W						
Protection Attribute	-	-						
Initial value	0	0000000						

[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 (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

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

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.

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).

Bit	Description
0	No Message RAM access failure occurred.
1	Message RAM access failure occurred.

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.*

[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 (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 (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	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

[bit31:30] 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 (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 (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 (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 (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 (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 (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.

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 (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 (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 (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 (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	X		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 (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			R/W					
Protection Attribute	R0,W0		-					
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 (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 (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 (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 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 (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 (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 (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.*

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 (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 (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 (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 (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 (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 (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 (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 (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 (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 (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 (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 (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

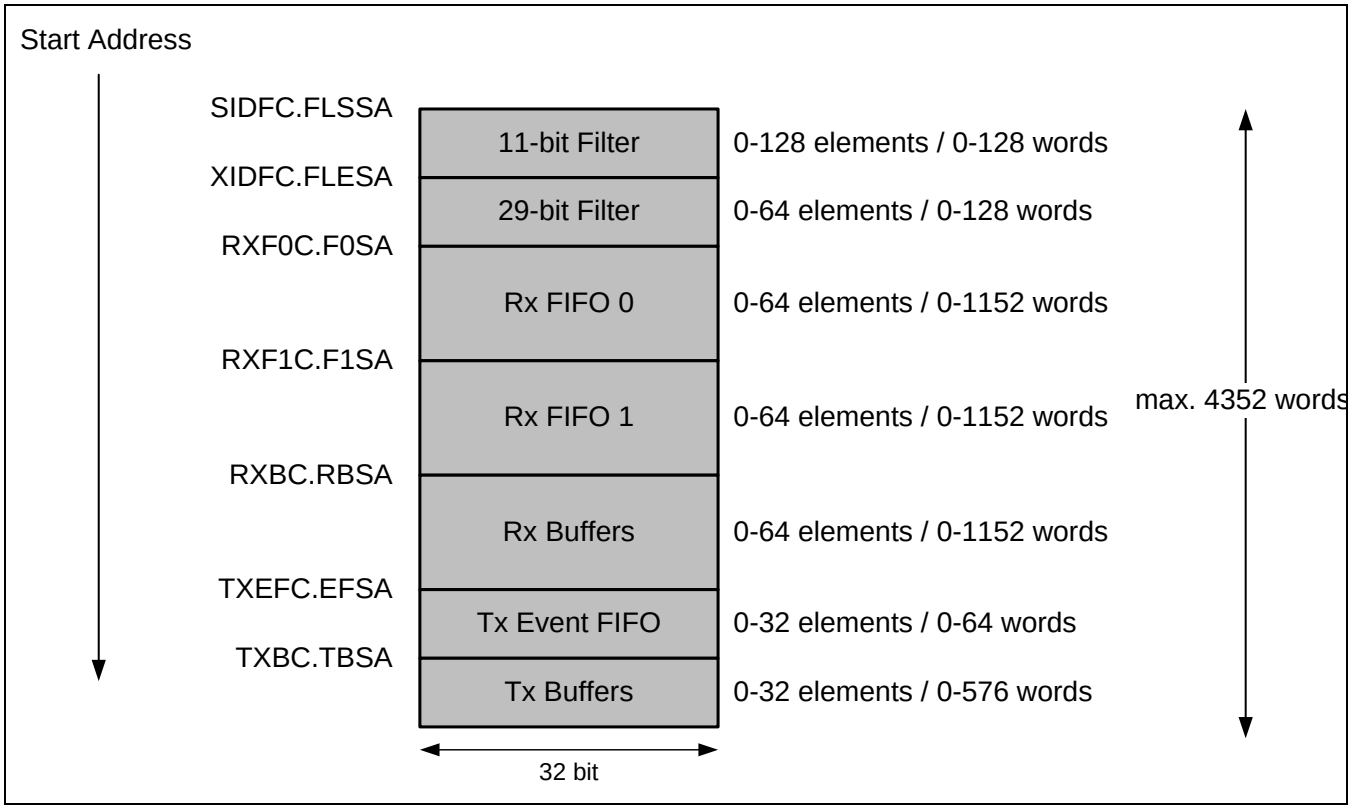
6.5.Standard Message ID Filter Element

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.*

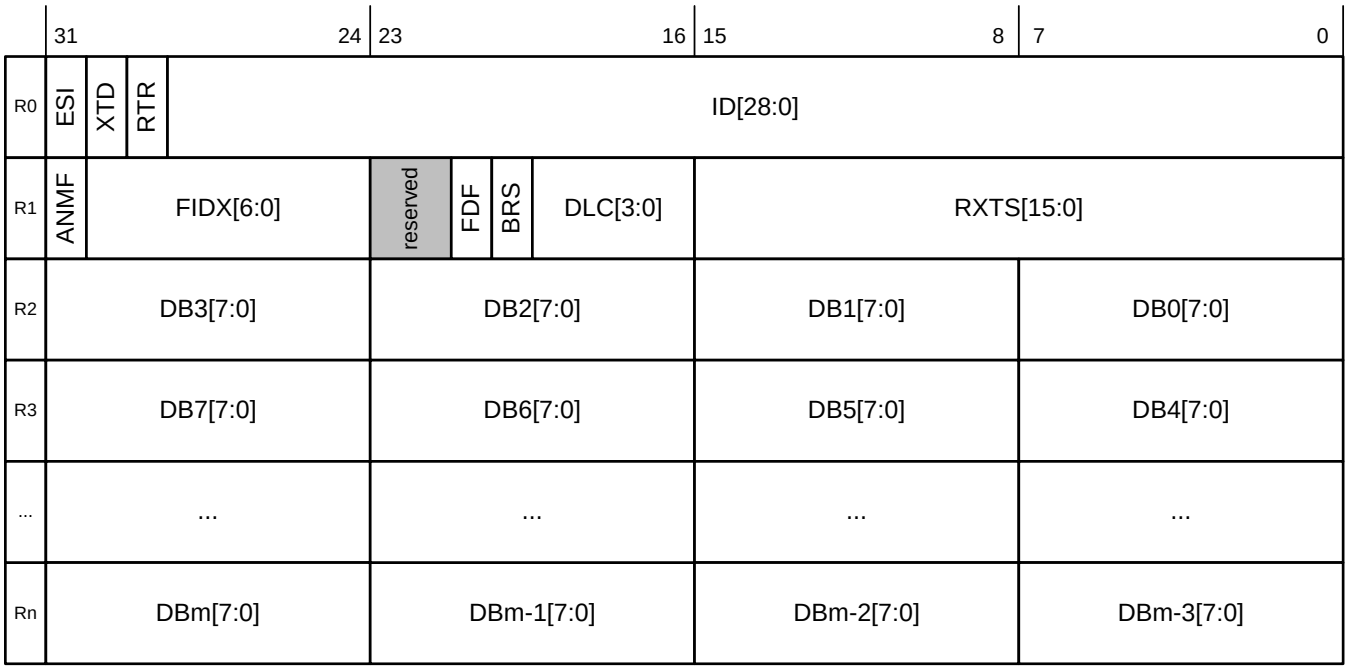
List of Message RAM Elements

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

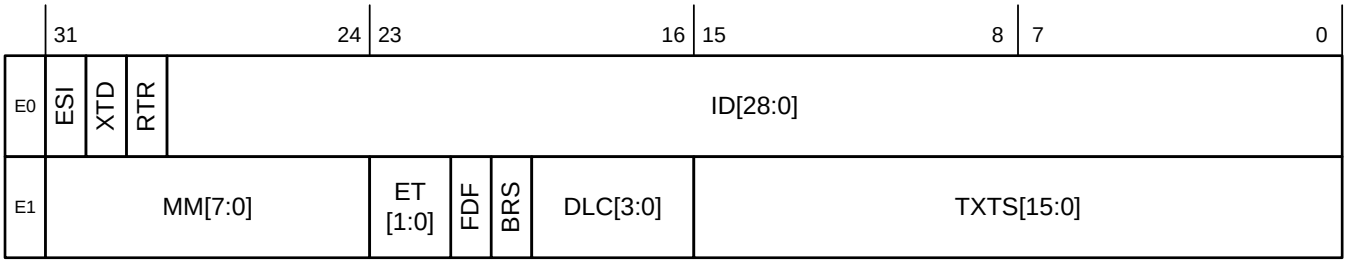
Tn [bit7:0] DBm-3[7:0]: Data Byte m-3

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

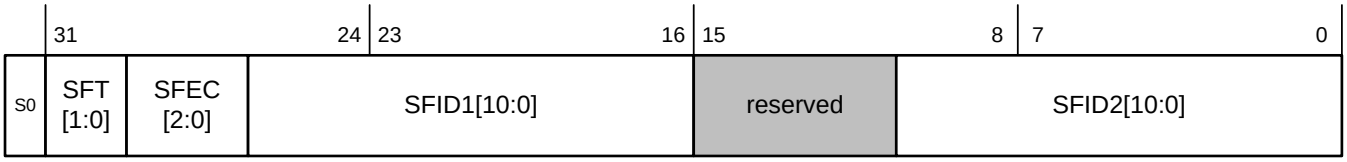
Timestamp Counter value captured on start of frame transmission. Resolution depending on configuration of the Timestamp Counter Prescaler (TSCC.TCP[3:0]).

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:

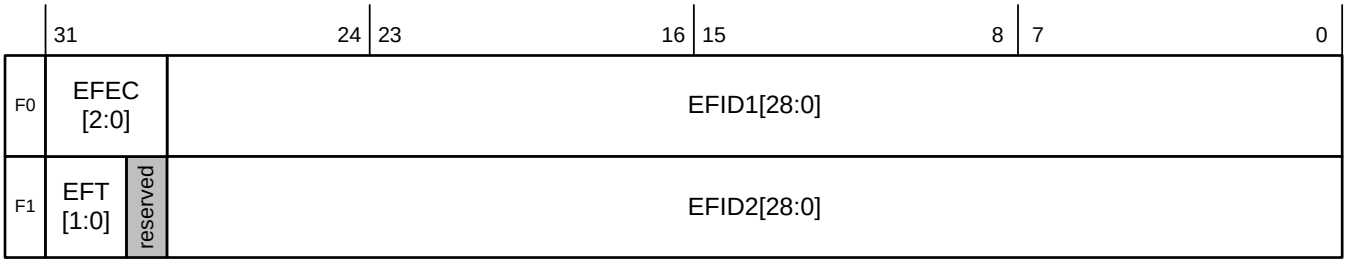
- *Debug message is used for Debug on CAN feature.*
- *Debug on CAN feature is not supported for Traveo Platform.*

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 Platform.*

CHAPTER 32: CAN Prescaler

This chapter describes the CAN prescaler.

1. **Overview**
2. **Configuration**
3. **Explanation of Operation**
4. **Setting Procedure**
5. **Registers**
6. **Restriction**

CANP-T00P00R01L02-E1-GBB00

1. Overview

This section provides an overview of CAN prescaler.

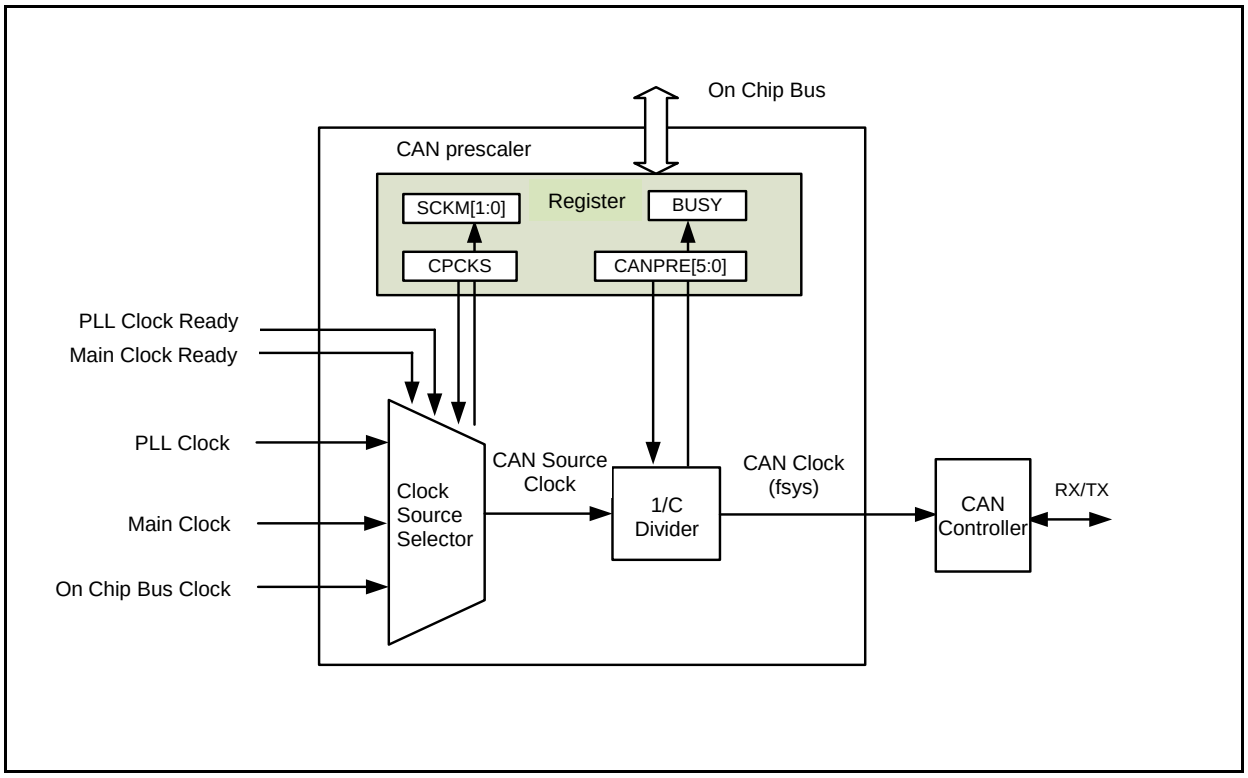
This module generates clocks (fsys) to be supplied from each clock source to the CAN macro.

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 clock and main clock are selectable for a source clock of the CAN prescaler.

If the PLL clock is selected but PLL 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

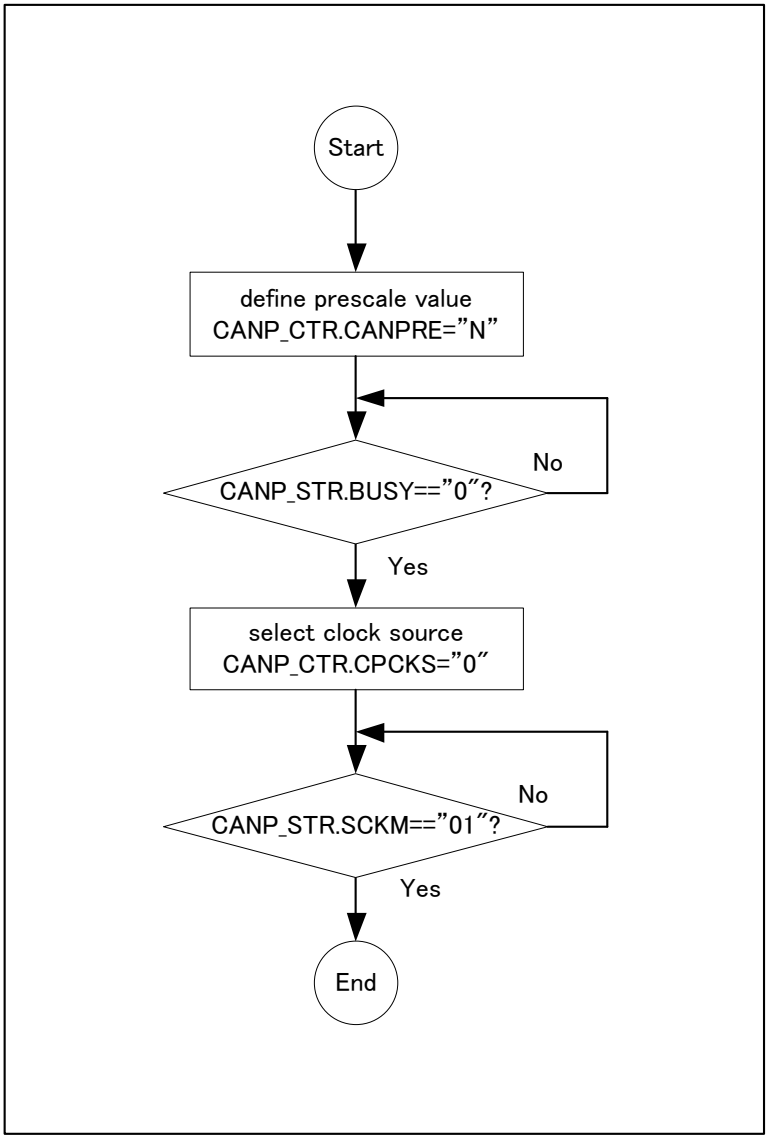
This module has the C division circuit that can change the CAN system clock (fsys) cycle. A division ratio from C = 1 to 64 can be set.

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.

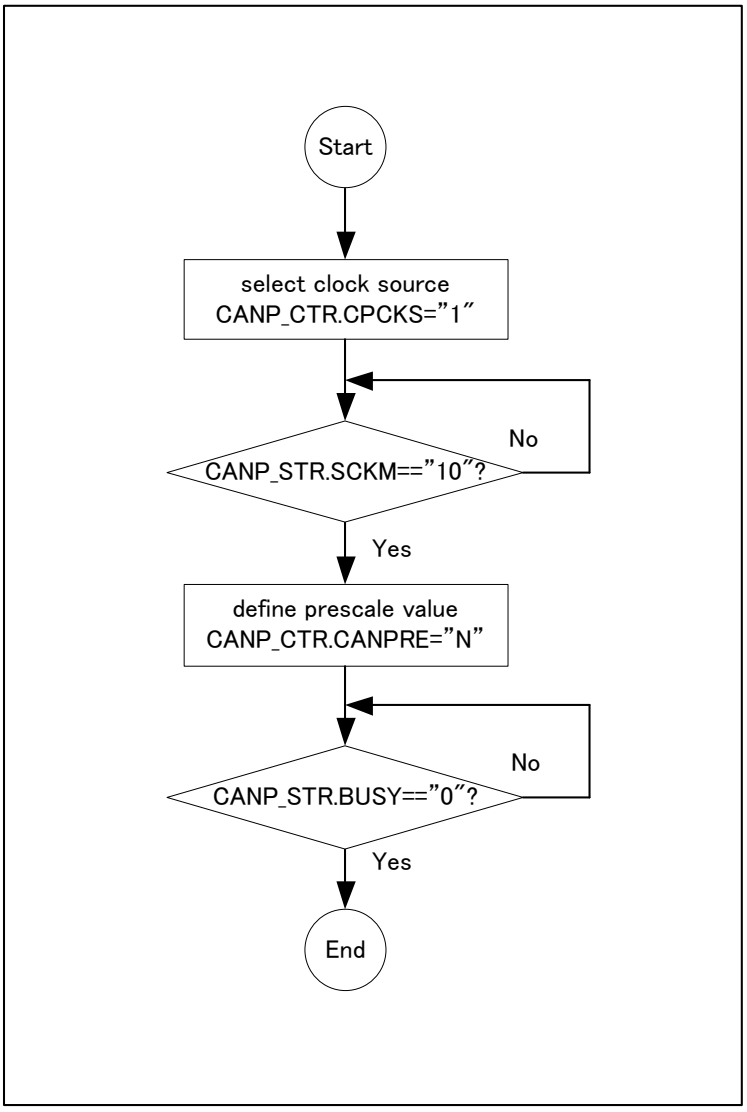
4. Setting Procedure

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 clock.



The following shows the procedure for switching from the PLL 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

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 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 clock, confirm that the PLL 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 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 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.

Note:

- *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 33: 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**

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.)

	Function	Description
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[x] 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

	Function	Description
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 SyncField error, and framing error)

Note:

− *The Wake Up function of the LIN is not supported.*

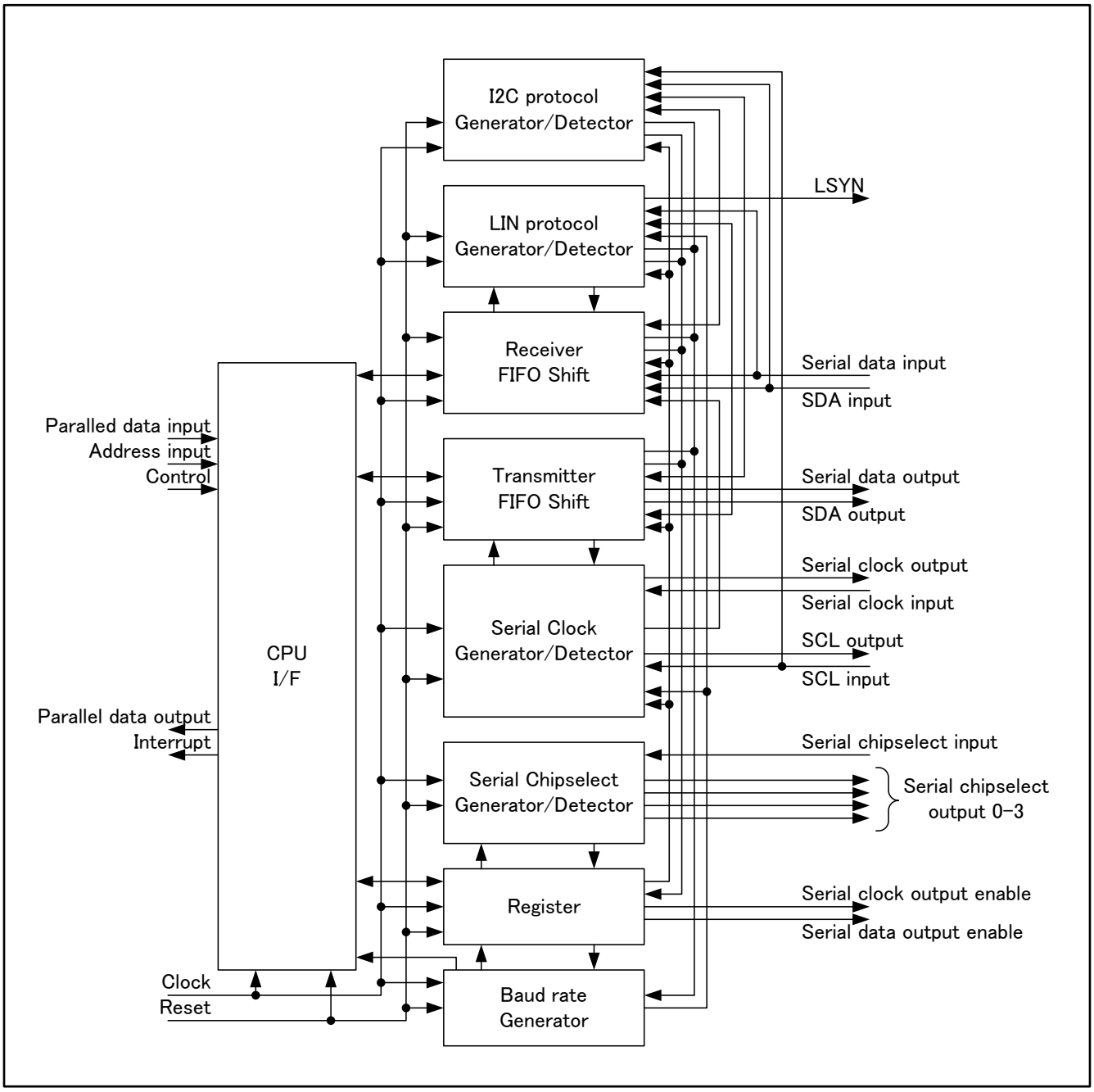
4. Overview of the I²C Interface (I²C Communication Control Interface)

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

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	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	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

Registers	Mode	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
EIBCR/ NFCR	4	-	-	SDLS	SCLS	SDLC	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	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	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

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

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 34: 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)**

Note:

- *ACK Mode of Block Transfer is not supported for this device.*

UART-T00P00R01L04-E1-GBB01

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.

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)
					Reception of as much data as the amount set in FBYTE		Reading of reception data (RDR) until the reception FIFO is empty
					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)
	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)
	RBERR	ESR	Appl	Appl	Reception block transfer error	ECR:REIE	Writing "1" to the reception block transfer error clear bit (ESRC:RBERRC)

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	ESFR	Appl	Appl	Transmission FIFO overrun	ECR:TEIE	Writing "1" to the transmission FIFO overrun flag clear bit (ESRC:TXOVRC)
	TBERR	ESFR	Appl	Appl	Transmission block transfer error	ECR:TEIE	Writing "1" to the transmission block transfer error clear bit (ESRC:TBERRC)
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)

*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 (SSR: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

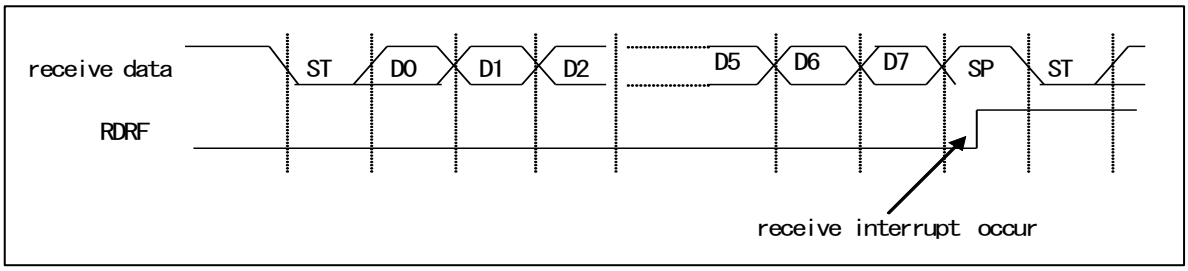
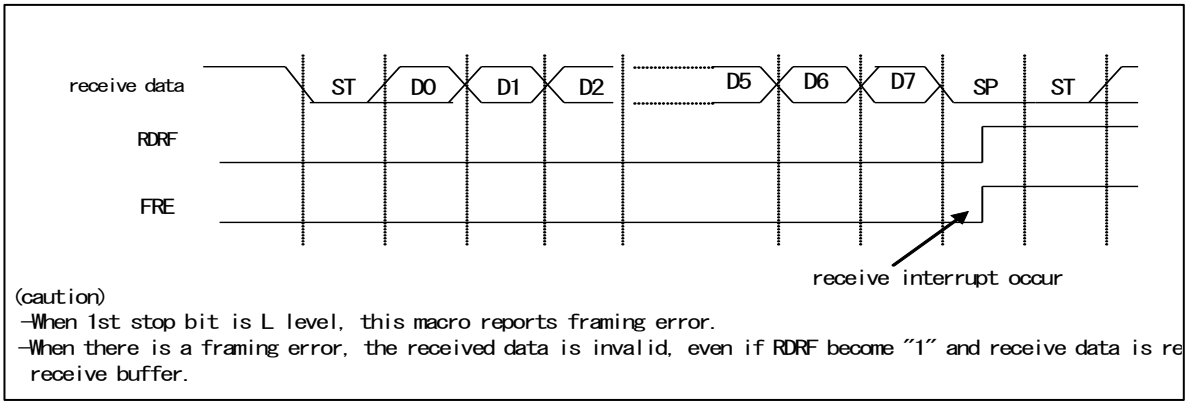


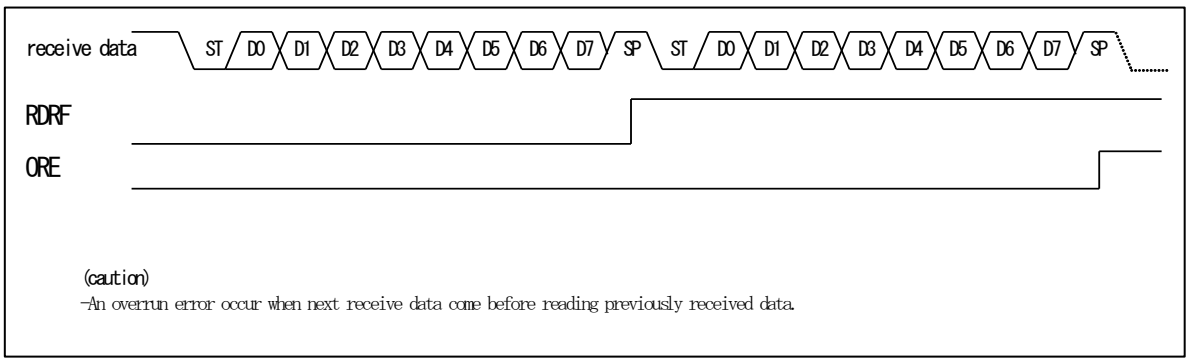
Figure 2-2 Set Timing of FRE (Framing Error) Flag Bit



Notes:

- *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

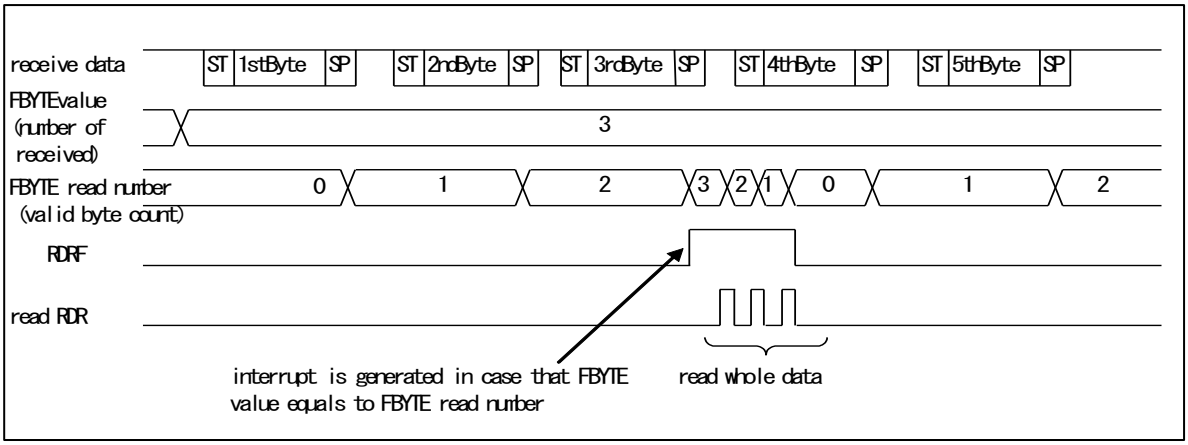
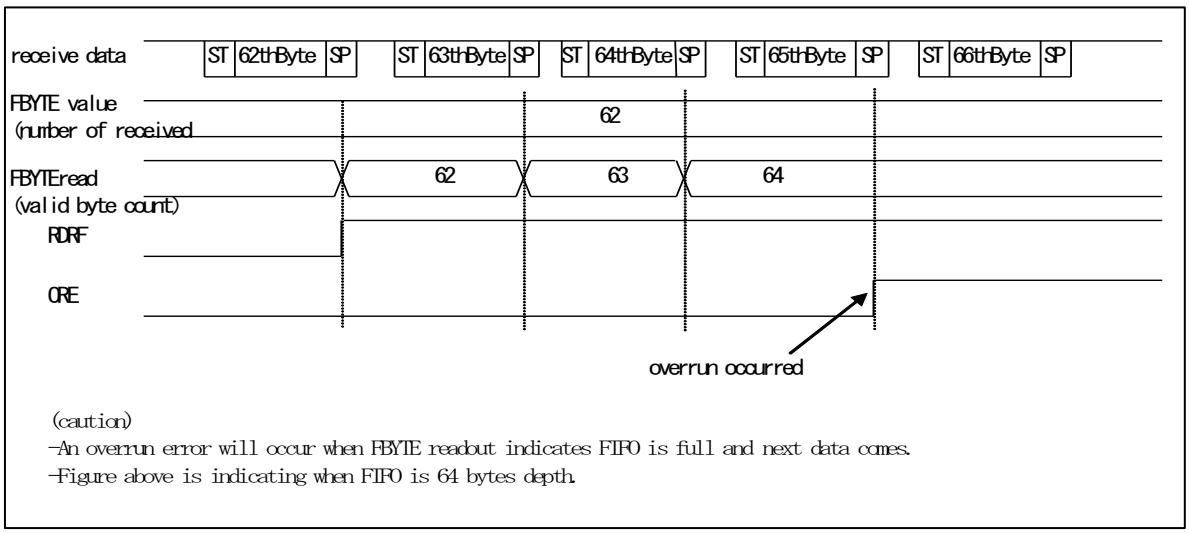


Figure 2-5 Set Timing of ORE (Overrun Error) Flag Bit



2.3. Occurrence of Transmission Interrupts and Flag Set Timing

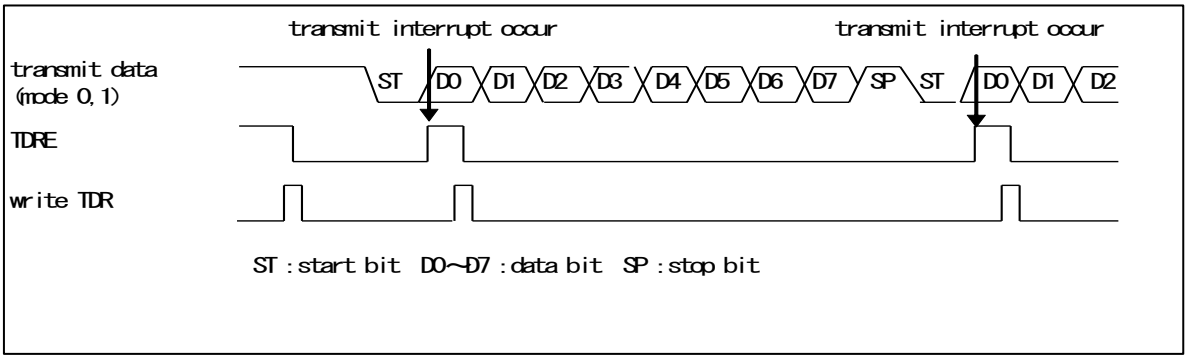
A data transmission interrupt occurs in the following cases: 1. The transmission data is transferred from the transmission data register (TDR) to the transmission shift register (SSR:TDRE = 1) to start data transmission. 2. The transmission operation is not performed (SSR:TBI = 1).

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).

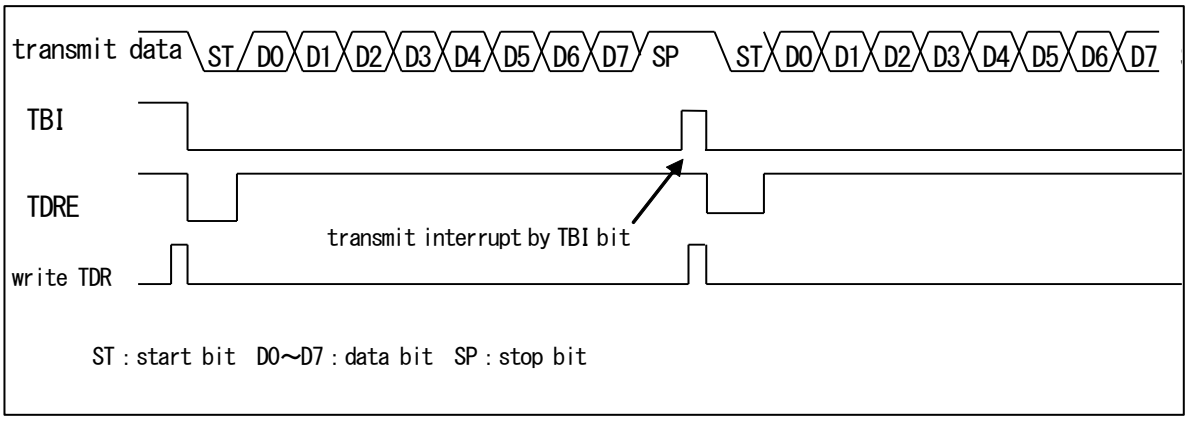
Figure 2-6 Set Timing of Transmission Data Empty Flag (SSR:TDRE)



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

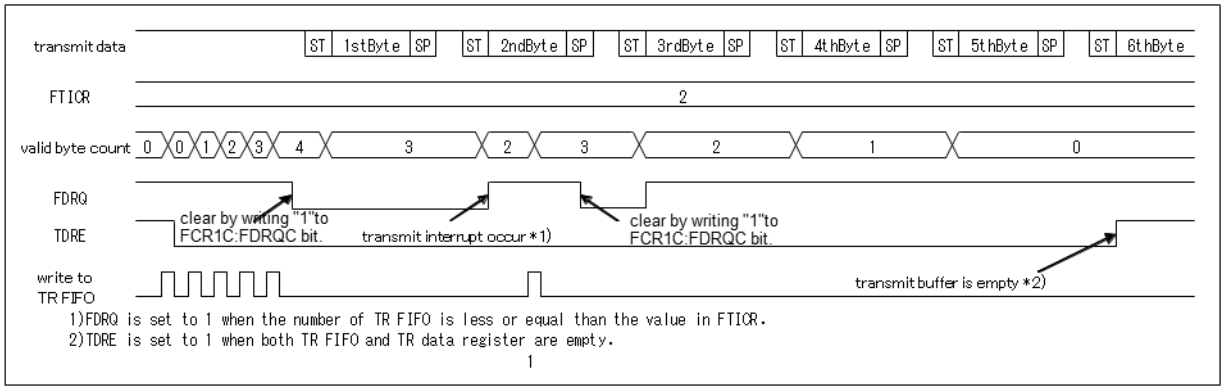
When the transmission FIFO is used, an interrupt occurs if the number of data items stored in the transmission FIFO is equal to or less than the setting of the FTICR register (FTICR).

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



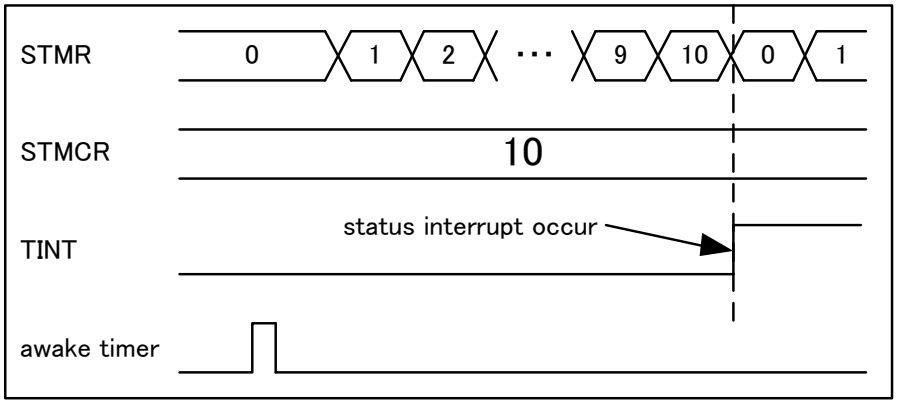
2.5. Occurrence of Timer Interrupts and Flag Set Timing

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.

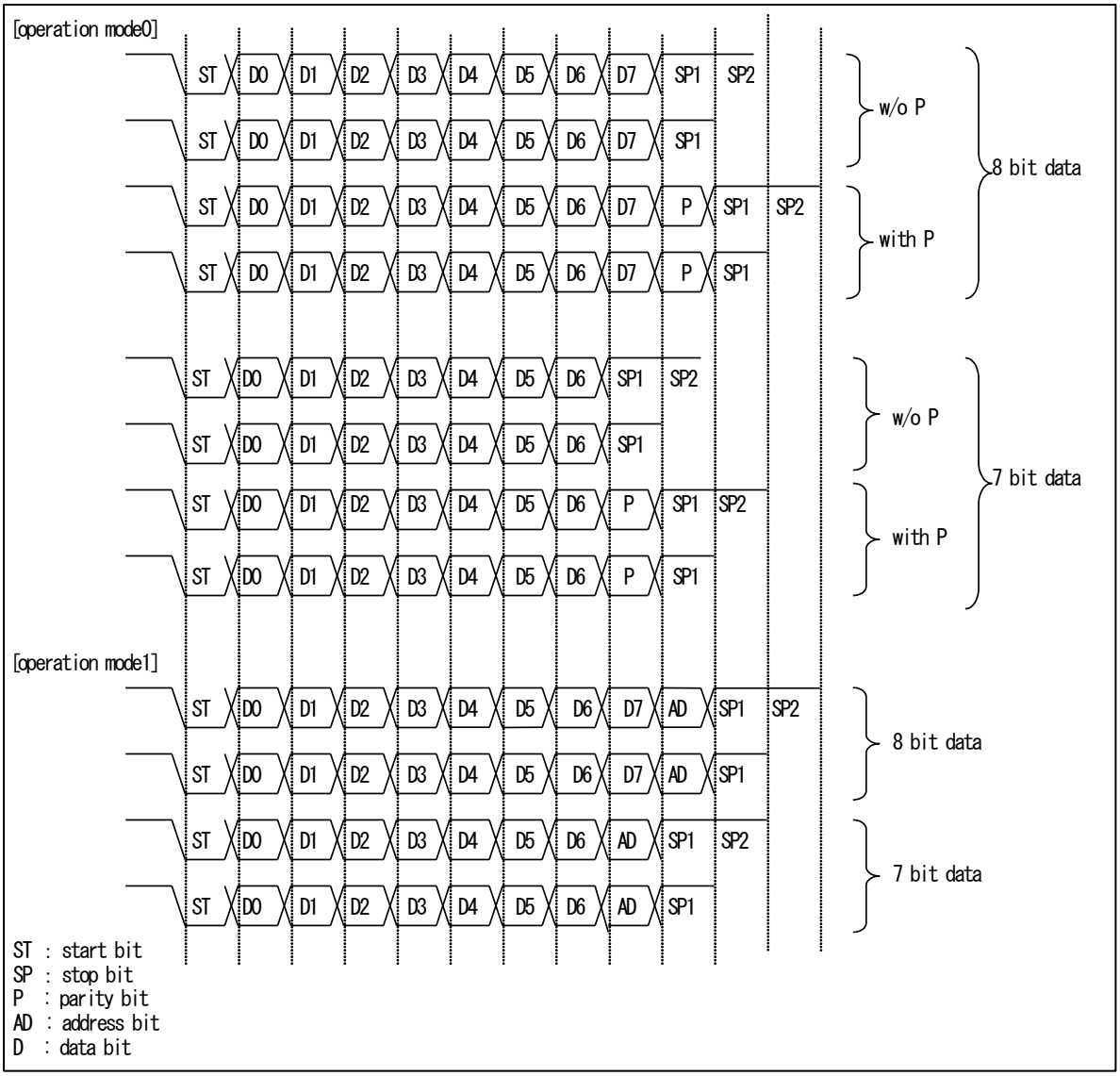
UART Operation

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)

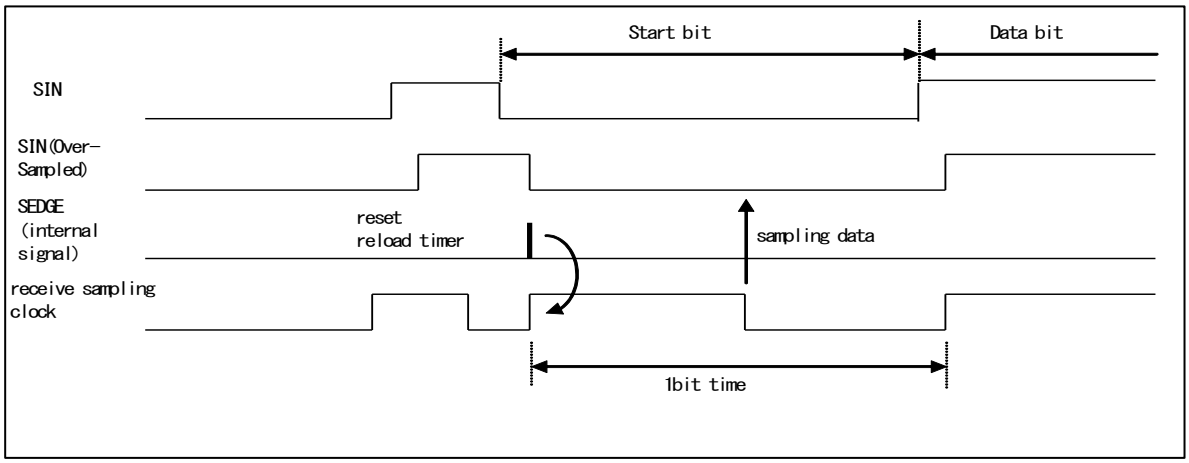
Clock Selection

- 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

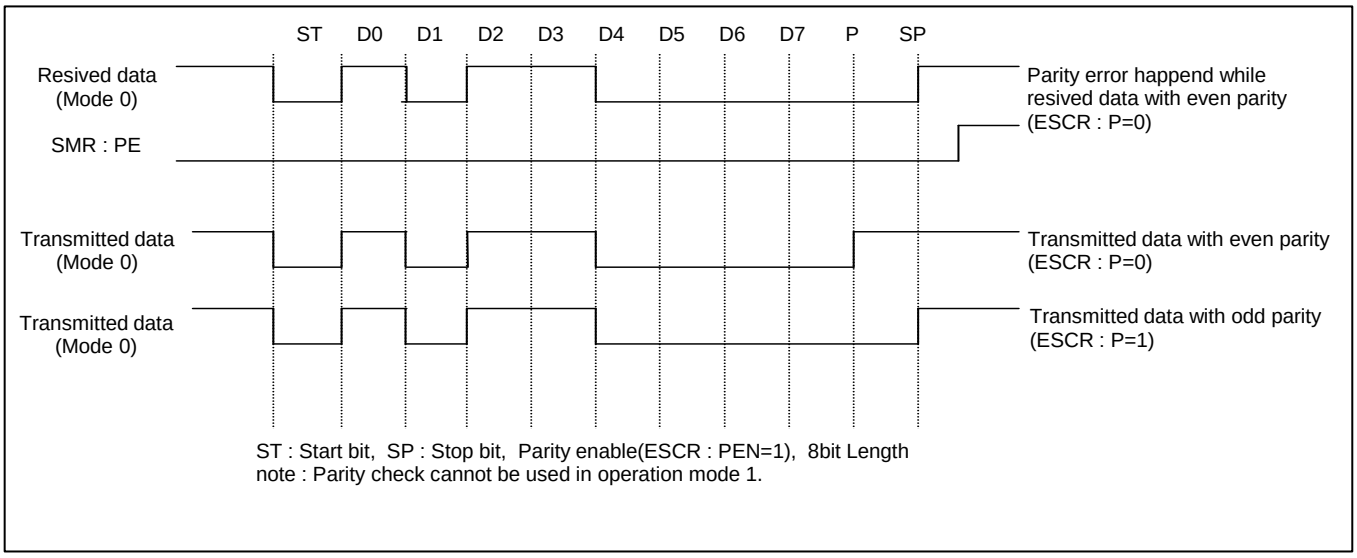
- Operation mode 0 allows the detection of parity errors, overrun errors, and framing errors.
- Operation mode 1 allows the detection of overrun errors and framing errors. The detection of parity errors is not possible.

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

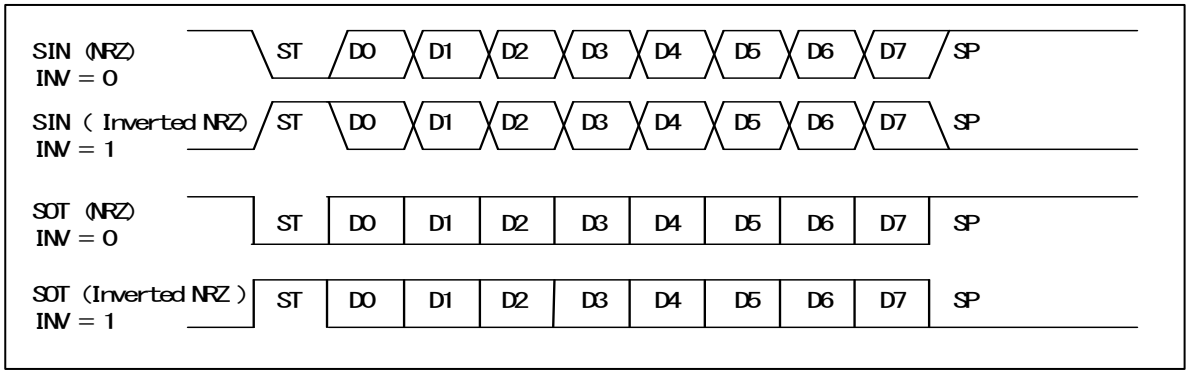


Data Signaling Method

- 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.

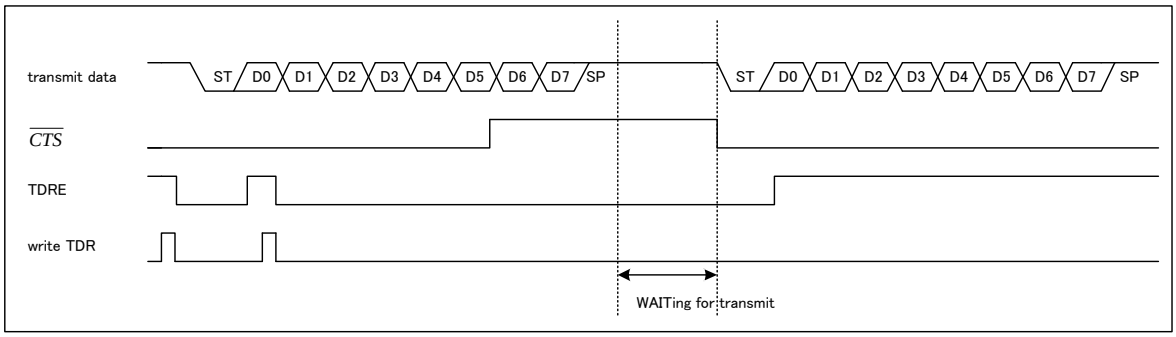
Hardware Flow Control

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.

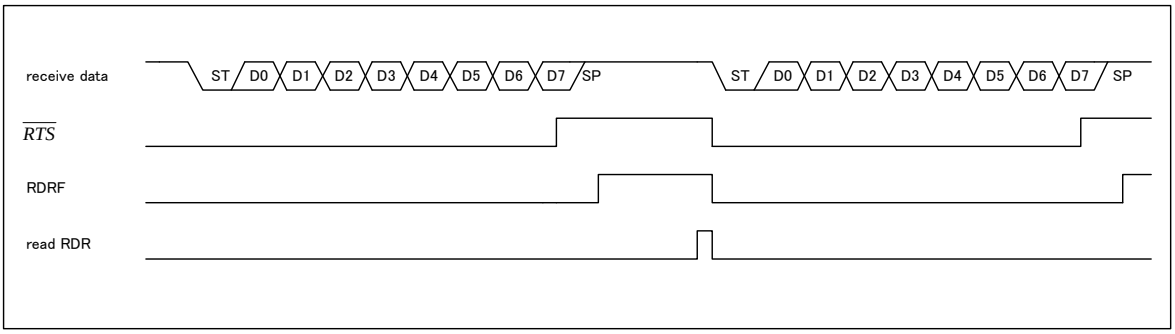
Figure 3-5 Hardware Flow Control for Data Transmission (SMR:SBL = 0, ESCR:ESBL = INV = PEN = L2 = L1 = L0 = 0)



- 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)

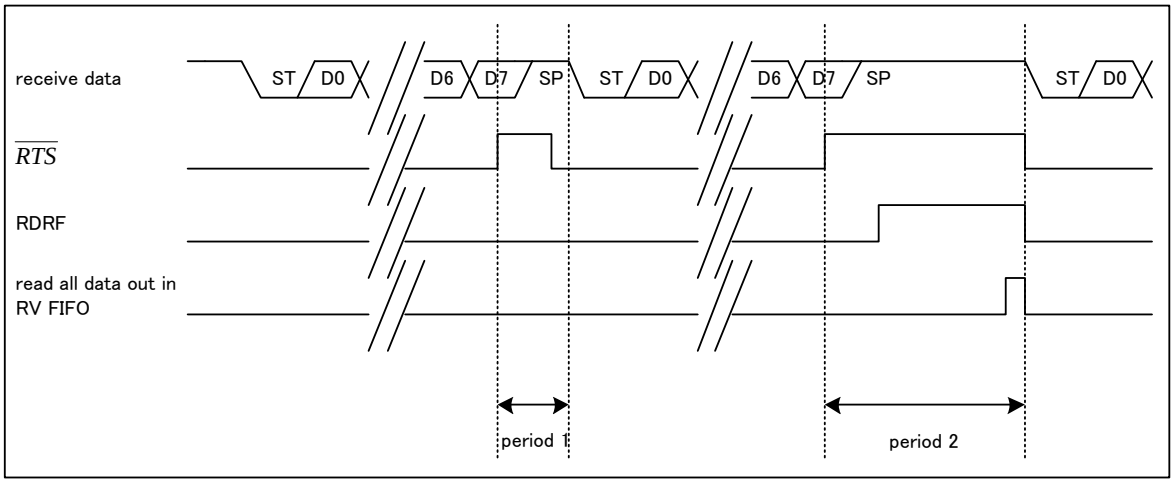


- FIFO used

  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

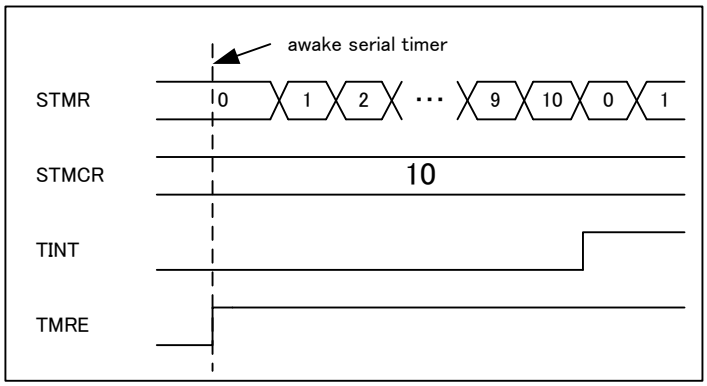
Activating the Serial Timer

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.

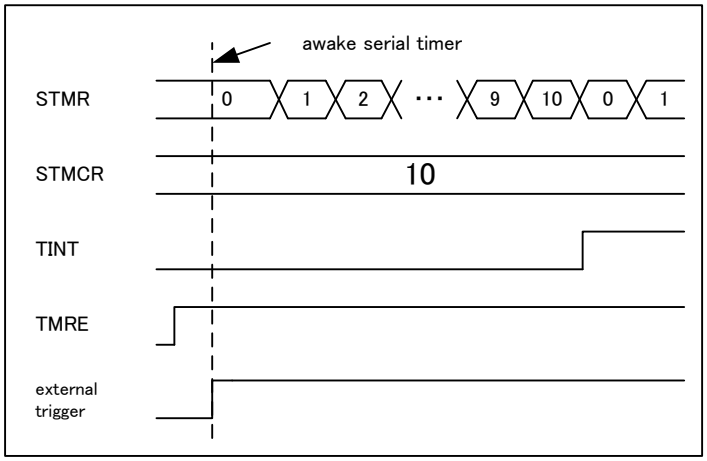
Figure 4-1 Activation by Serial Timer Enable Bit (STMCR = 10, SACSR:TRGE = 0, TSYNE = 0)



- 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)

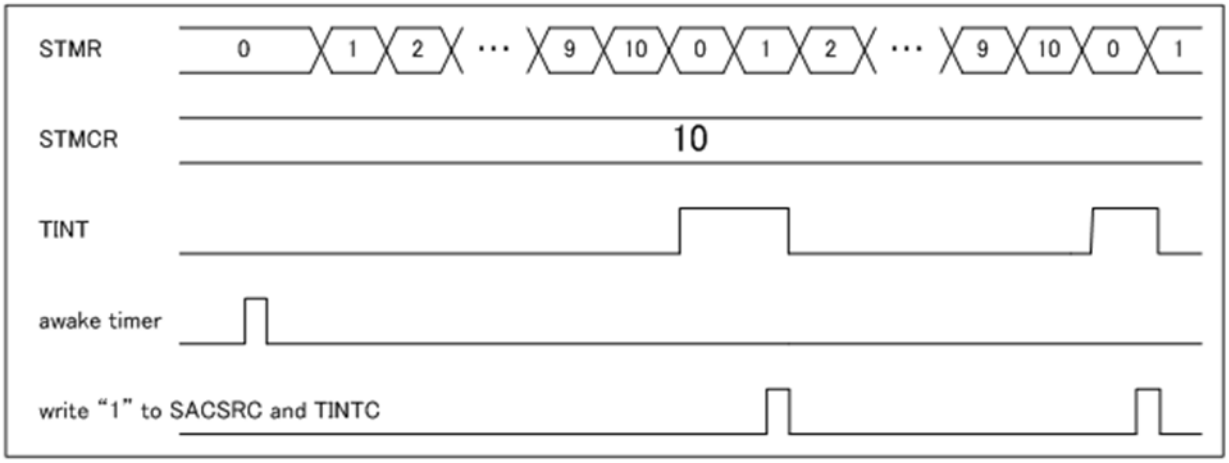


Figure 4-4 Flow Chart of Serial Timer Initialization

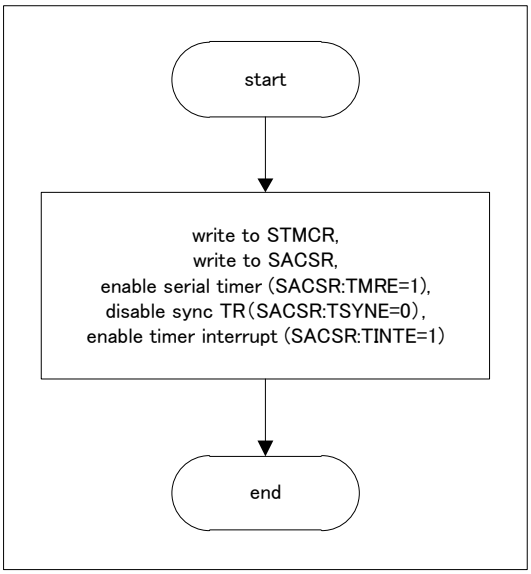
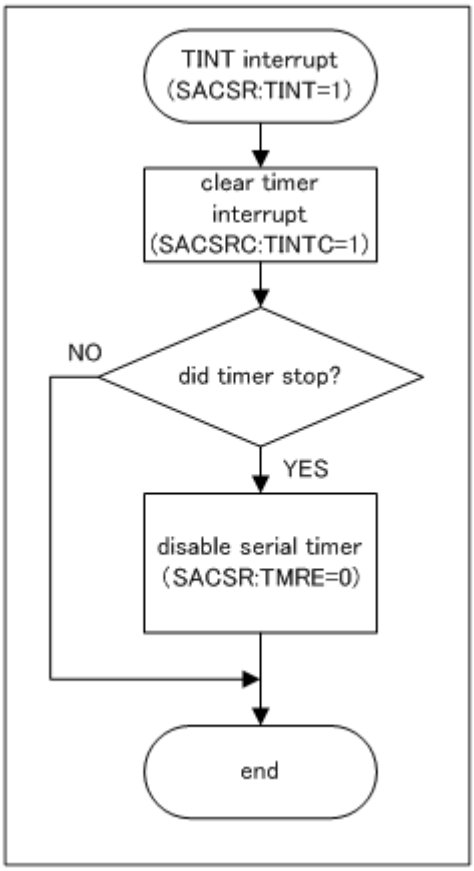


Figure 4-5 Flow Chart of Serial Timer Interrupt Processing



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:TDIV) of the timer operation clock is set to 0b0000, the timer interrupt flag (SACSR:TINT) is fixed to "1".*

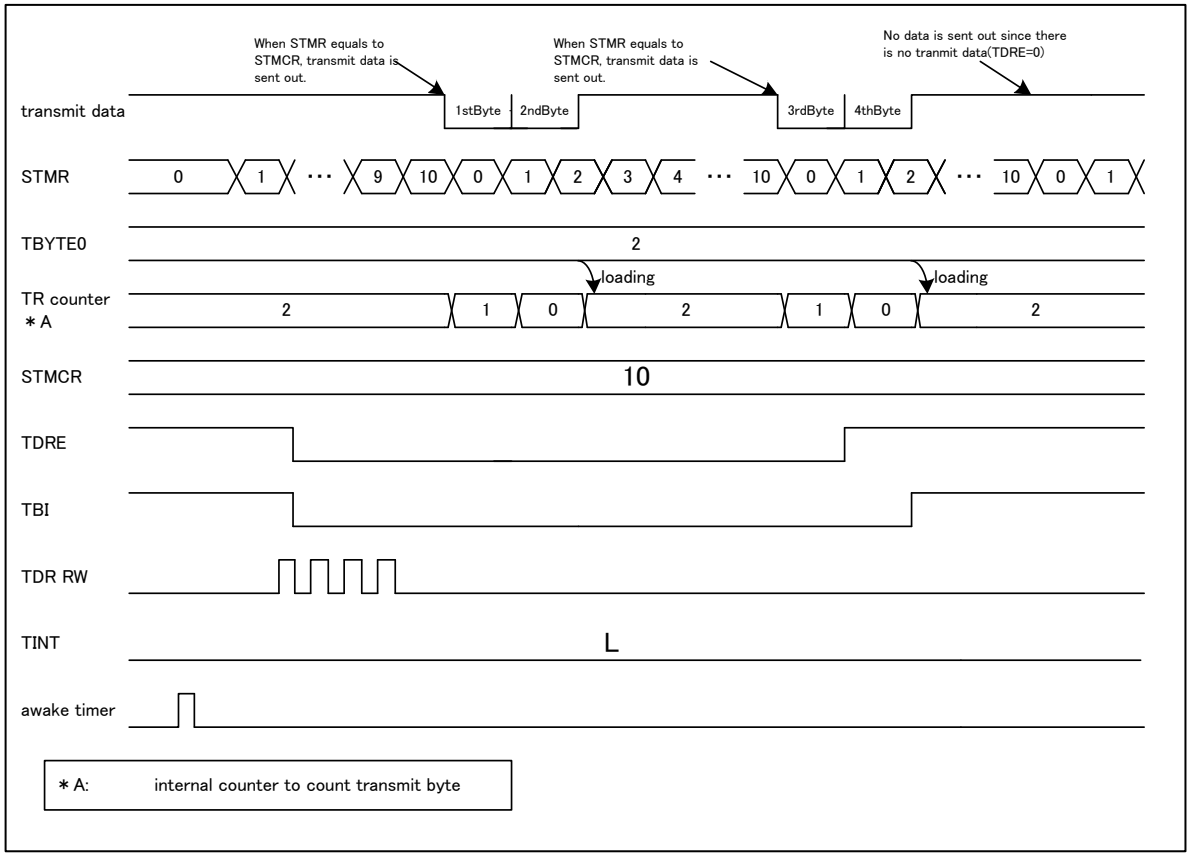
Transmission Operation Synchronized with the Timer

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

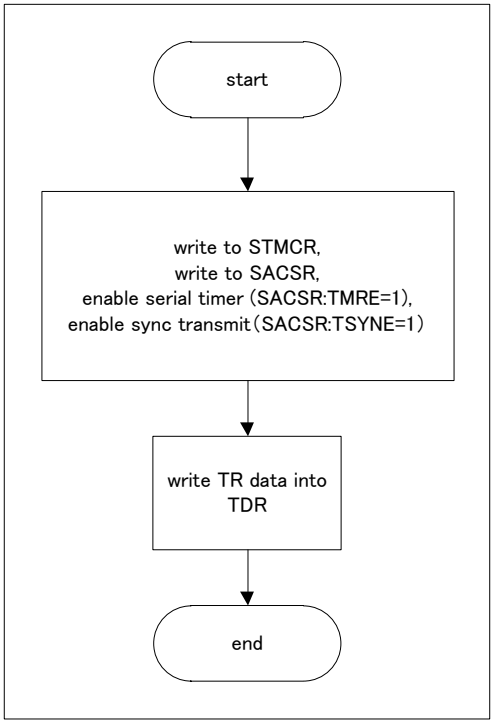
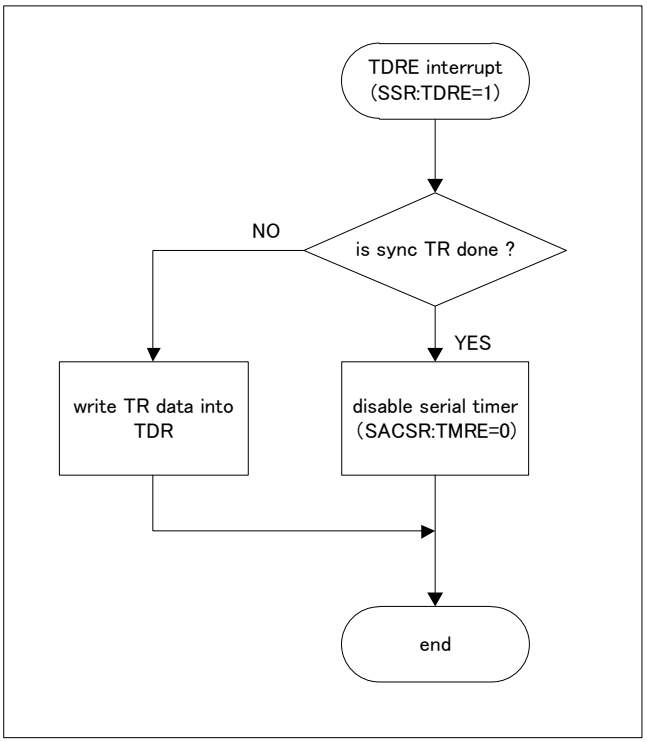


Figure 4-8 Flow Chart of Interrupt Processing in Transmission Synchronized with Timer



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.*

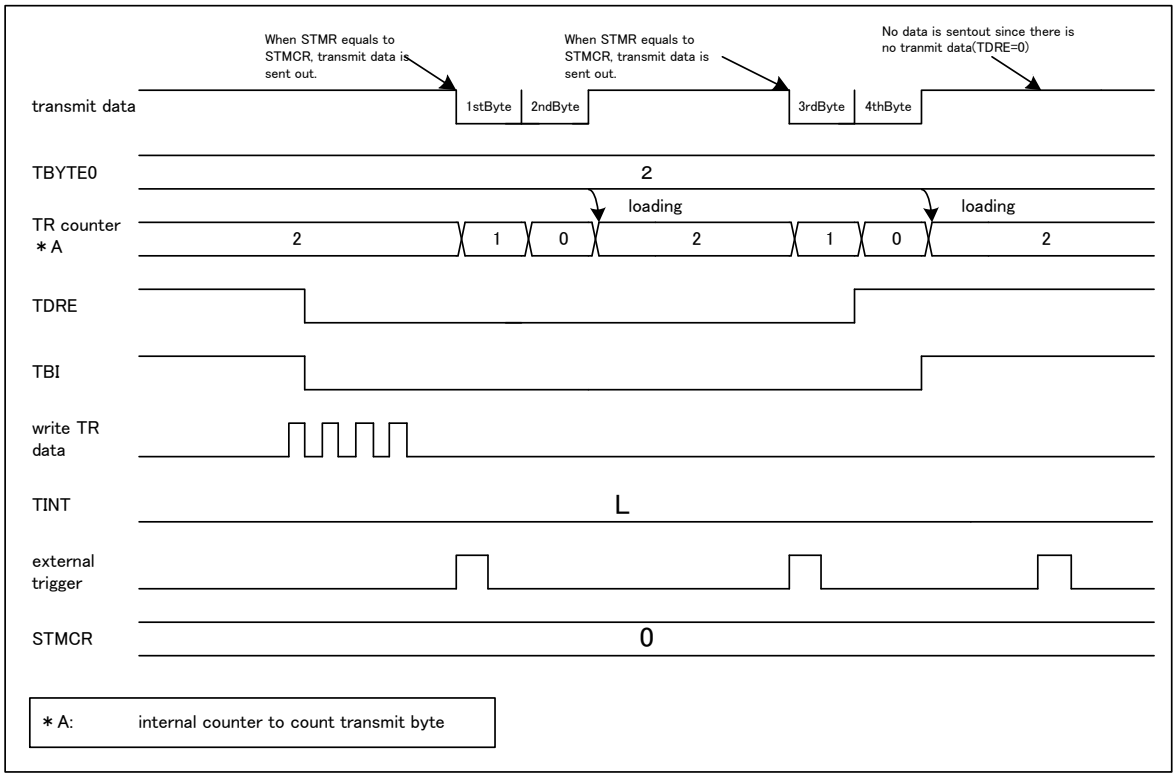
Transmission Operation Synchronized with an External Trigger

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

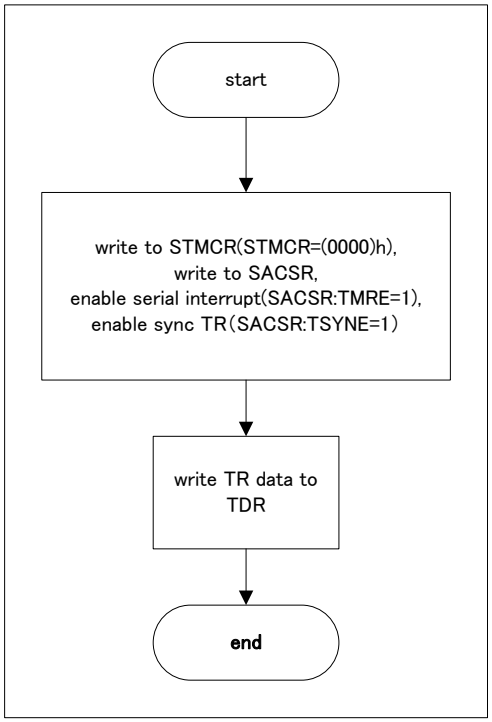
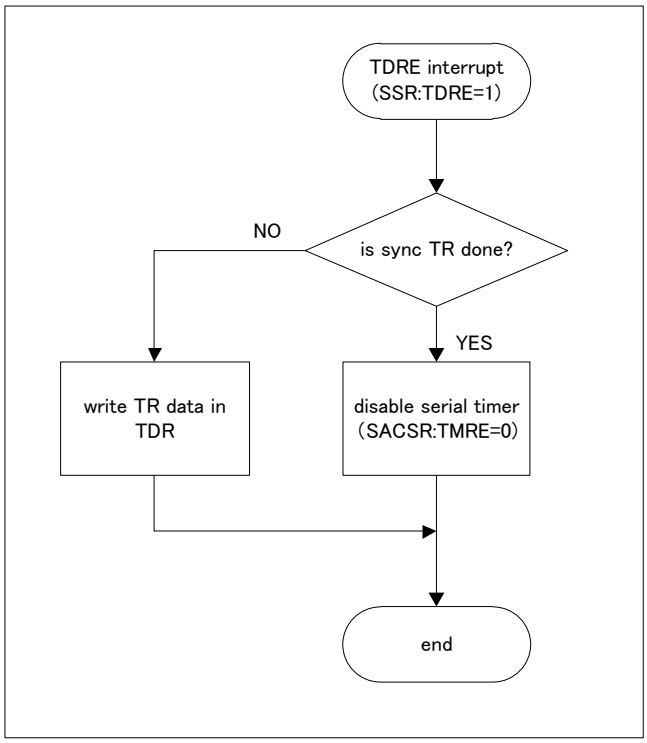


Figure 4-11 Flow Chart of Interrupt Processing in Transmission Synchronized with External Trigger



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

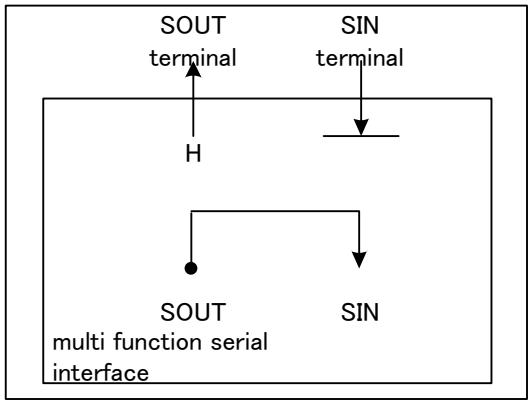
This section describes operation in a test mode.

Serial Test Mode

When serial test mode is enabled (SACSR:STST = 1), SOUT and SIN are connected together inside the multifunction 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)

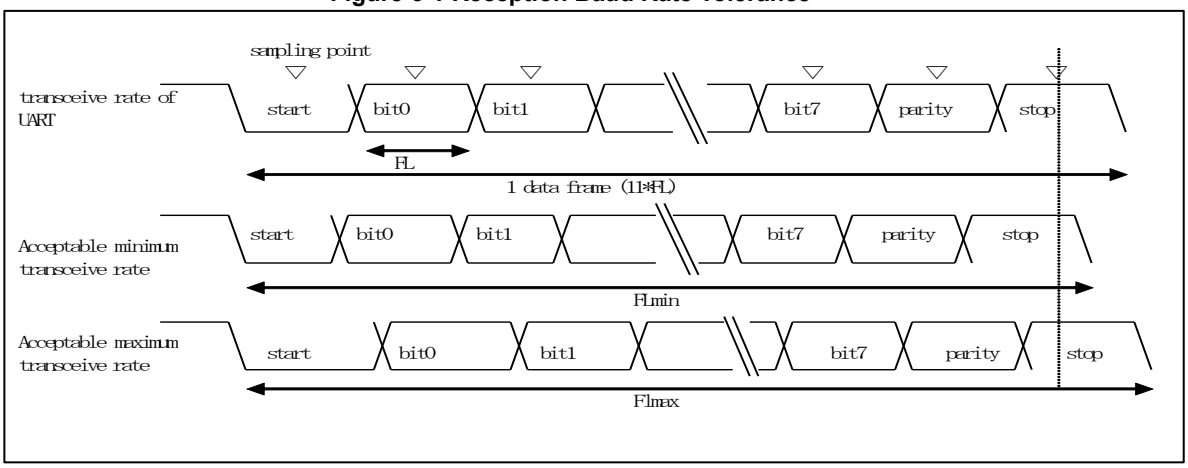
Baud Rate	40 MHz		48 MHz		72 MHz		80 MHz	
(bps)	Value	ERR	Value	ERR	Value	ERR	Value	ERR
4 M	9	0	11	0	17	0	19	0
2.5 M	15	0	-	-	-	-	31	0
2 M	19	2	23	0	35	0	39	0
1 M	39	0	47	0	71	0	79	0
500000	79	0	95	0	143	0	159	0
460800	86	-0.22	103	0.16	155	0.16	173	-0.22
250000	159	0	191	0	287	0	319	0
230400	173	-0.22	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 * Flmax = (11 bits * (V + 1) - (V + 1) / 2) / Φ V: Reload value, Φ: Bus clock

Flmax = (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:

Flmax = (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 acknowledgment standard (ACK mode) and 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 reception block size register.

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.

Note:

- *ACK Mode of Block Transfer is not supported for this device.*

7.1. ACK Mode

This section describes block transfer in ACK mode.

Reception Block Transfer

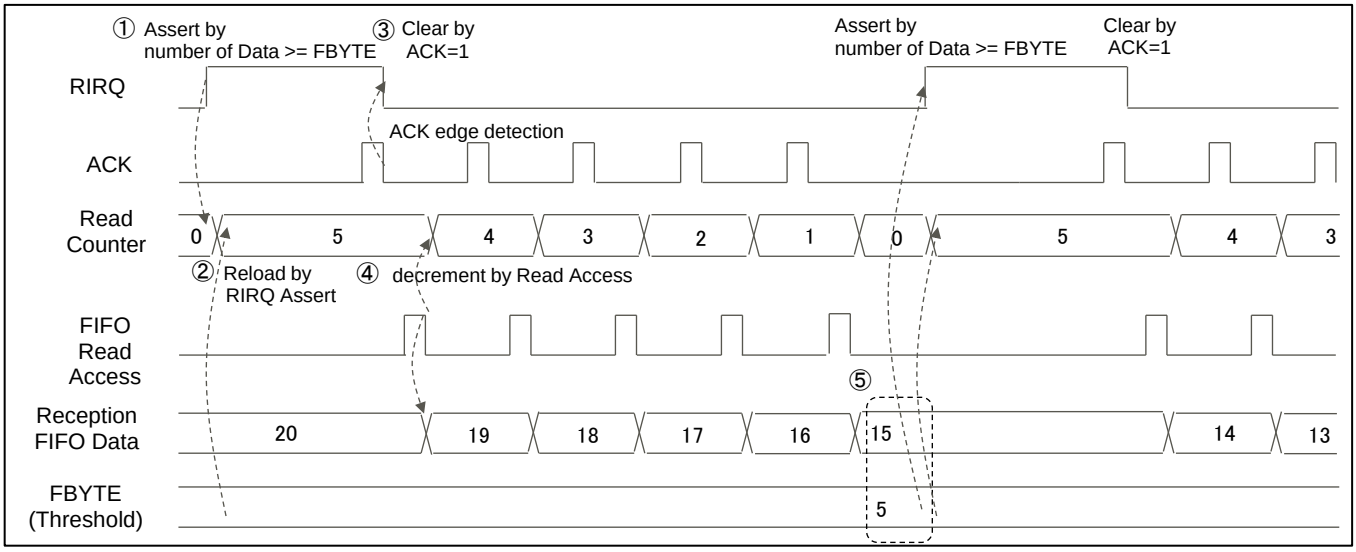
Reception block transfer is performed as described below.

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 reception acknowledgment is input, the reception block transfer request (RIRQ) is cleared. With the read of 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.

Figure 7-1 Reception Block Transfer in ACK Mode



Transmission Block Transfer

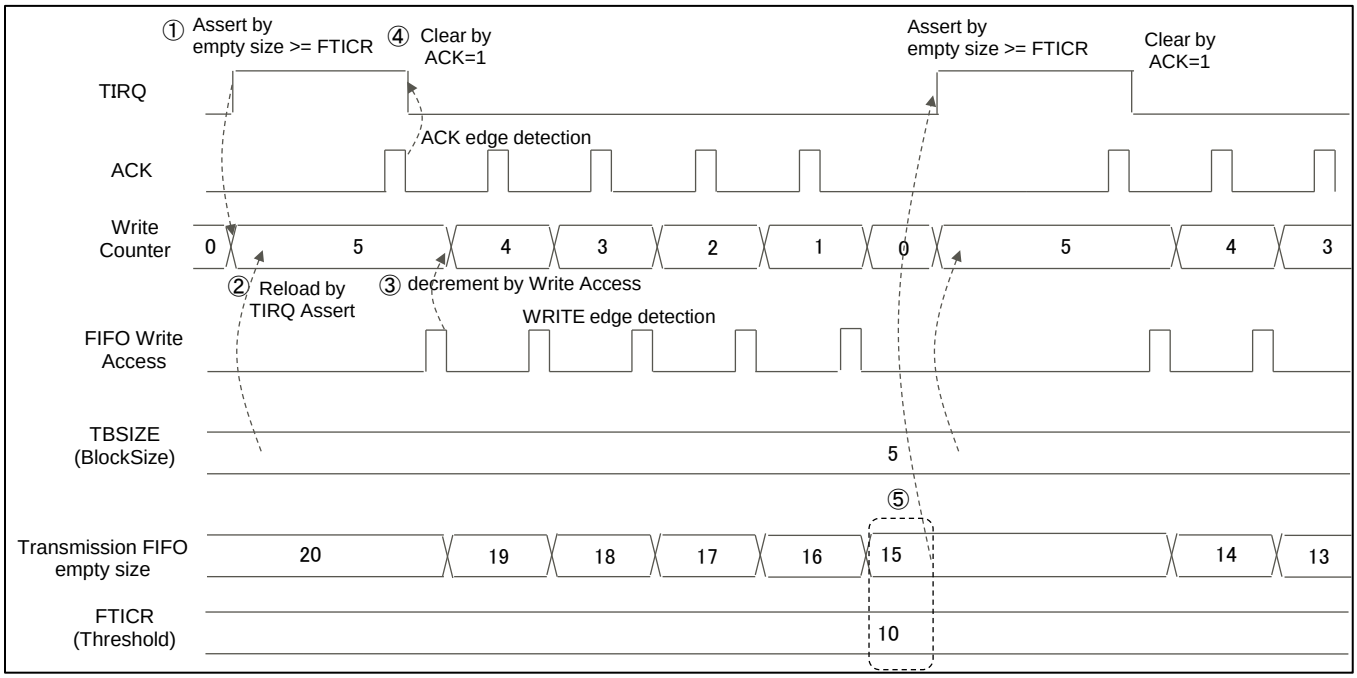
Transmission block transfer is performed as described below.

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 transmission acknowledgment is input, the transmission block transfer request (TIRQ) is cleared. With the write of the 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.

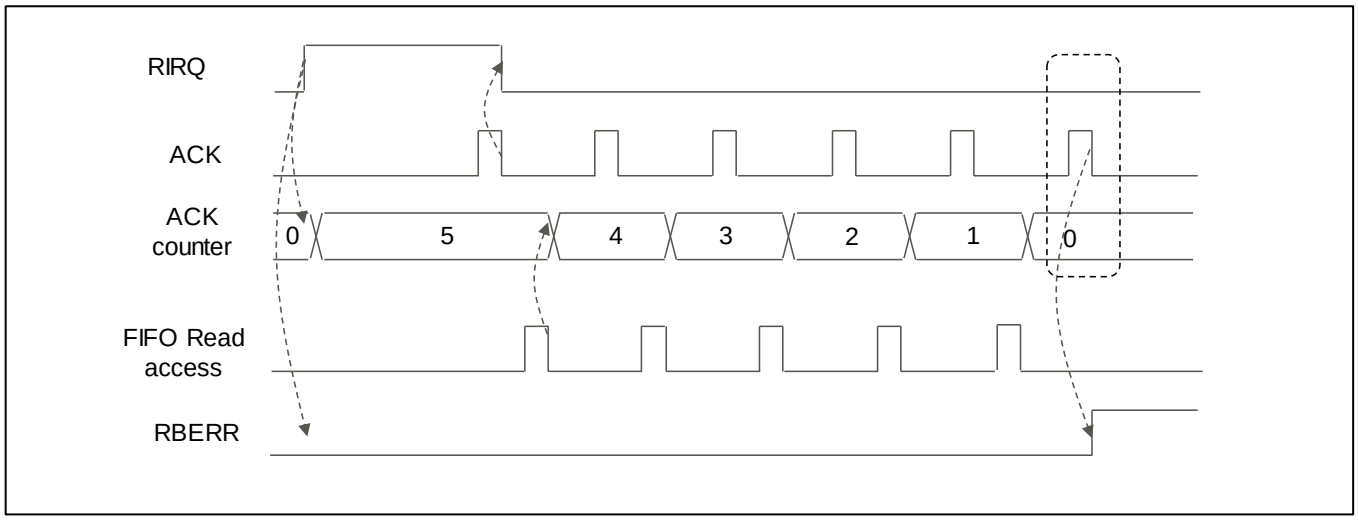
Figure 7-2 Transmission Block Transfer in ACK Mode



Reception Block Transfer Error

If, while the Read counter is "0", an acknowledgment is input before the next reception block transfer request (RIRQ) is asserted, a reception block transfer error occurs, and the ESR:RBERR bit is asserted.

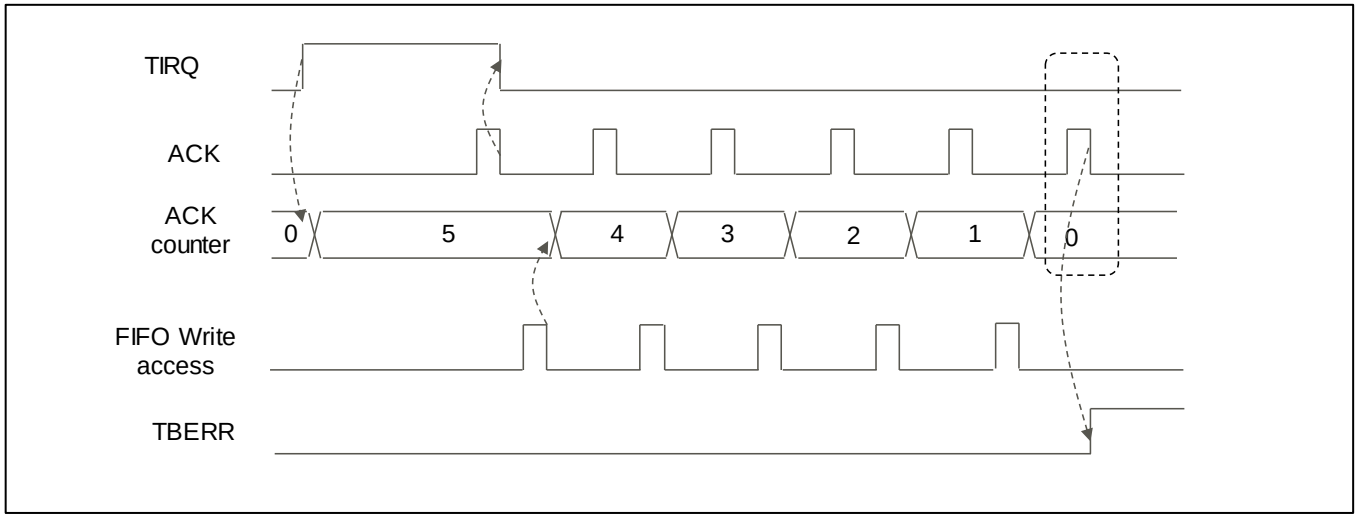
Figure 7-3 Reception Block Transfer Error in ACK Mode



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.

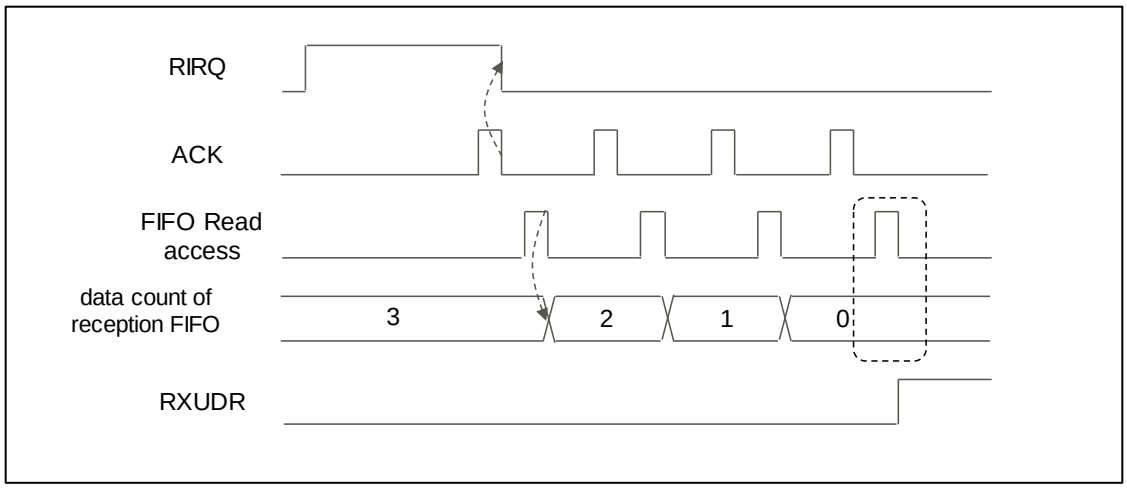
Figure 7-4 Transmission Block Transfer Error in ACK Mode



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.

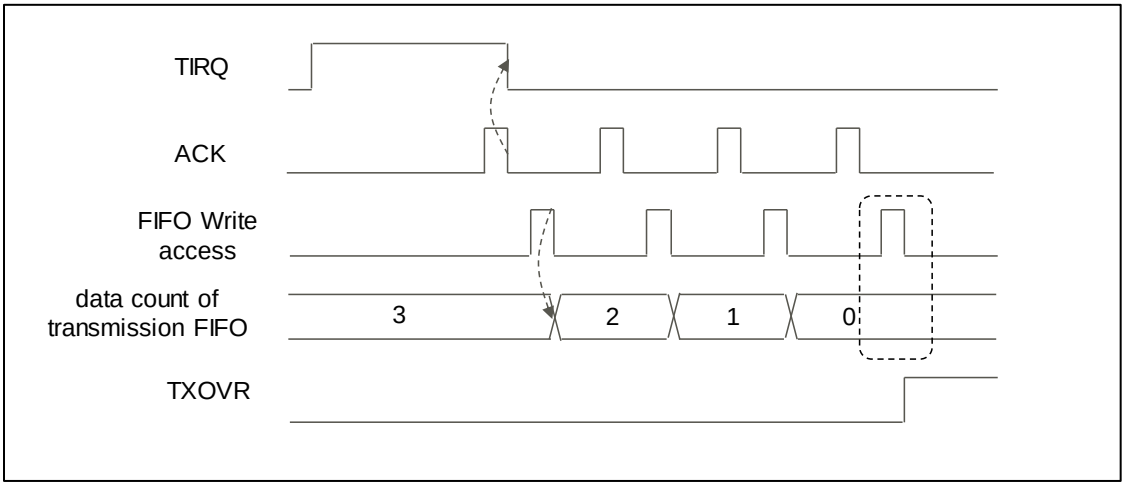
Figure 7-5 Reception FIFO Under Run in ACK Mode



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 ACK Mode



7.2. RW Access Mode

This section describes block transfer in RW access mode.

Reception Block Transfer

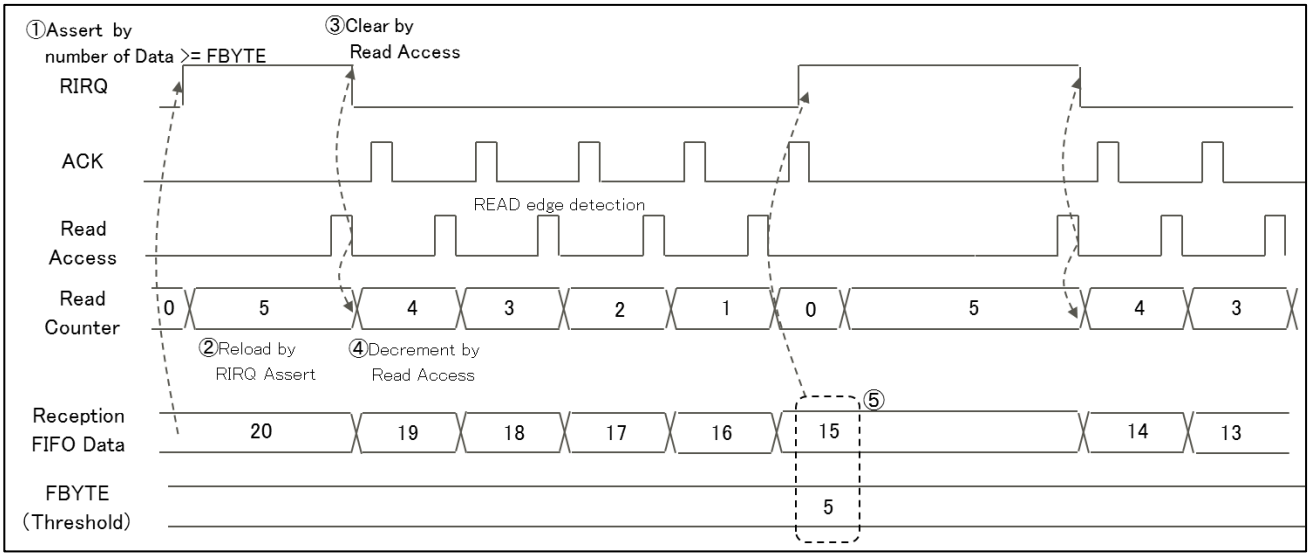
Reception block transfer is performed as described below.

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.

Figure 7-7 Reception Block Transfer in RW Access Mode



Transmission Block Transfer

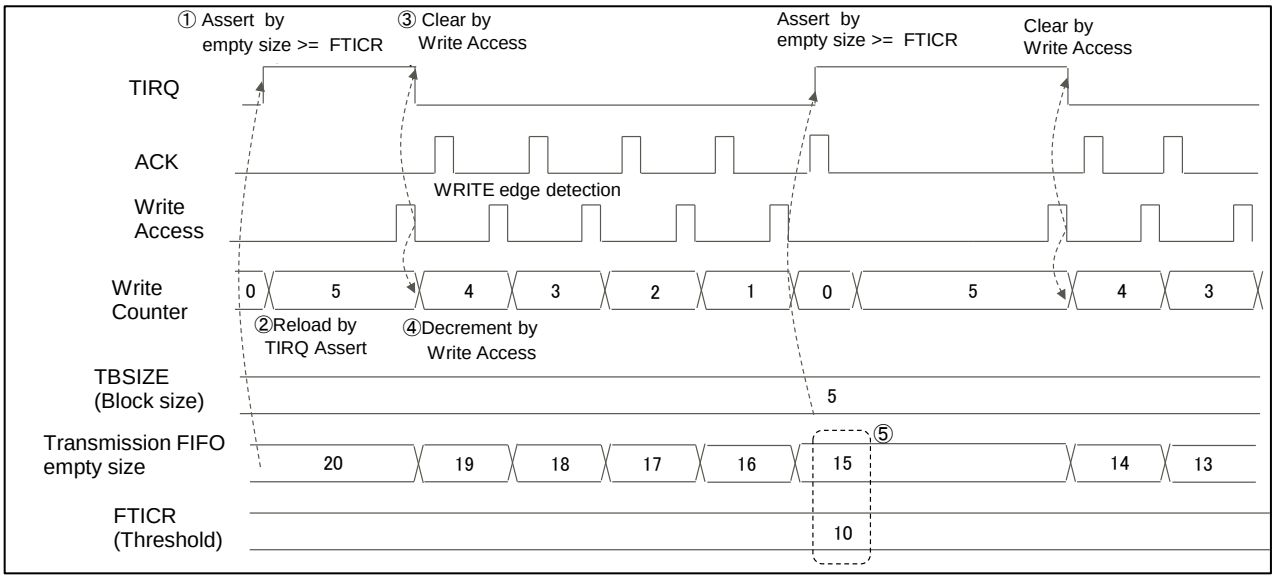
Transmission block transfer is performed as described below.

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.

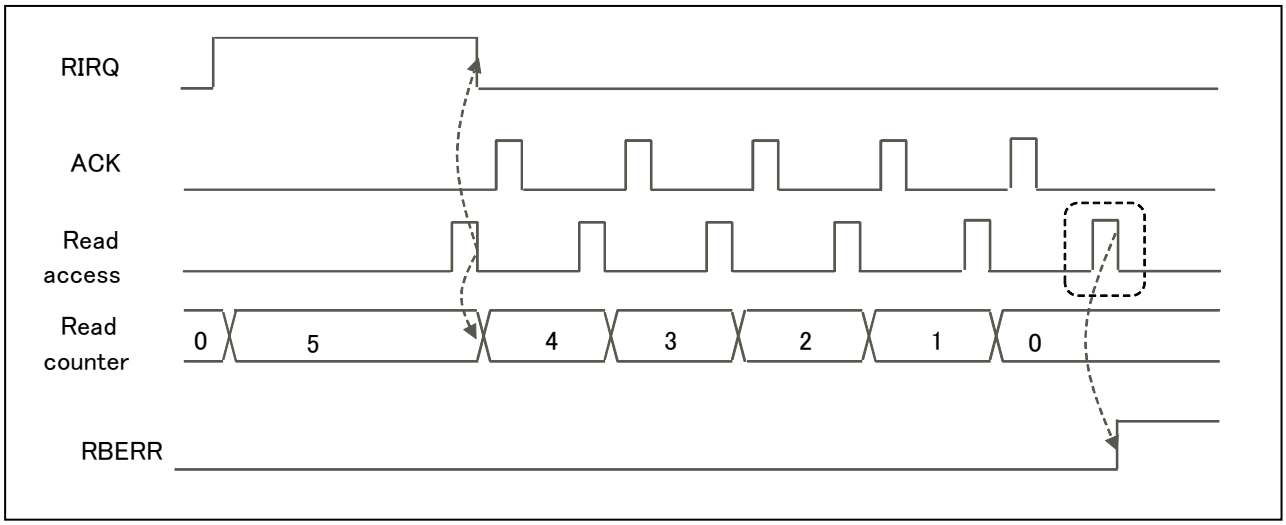
Figure 7-8 Transmission Block Transfer in RW Access Mode



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.

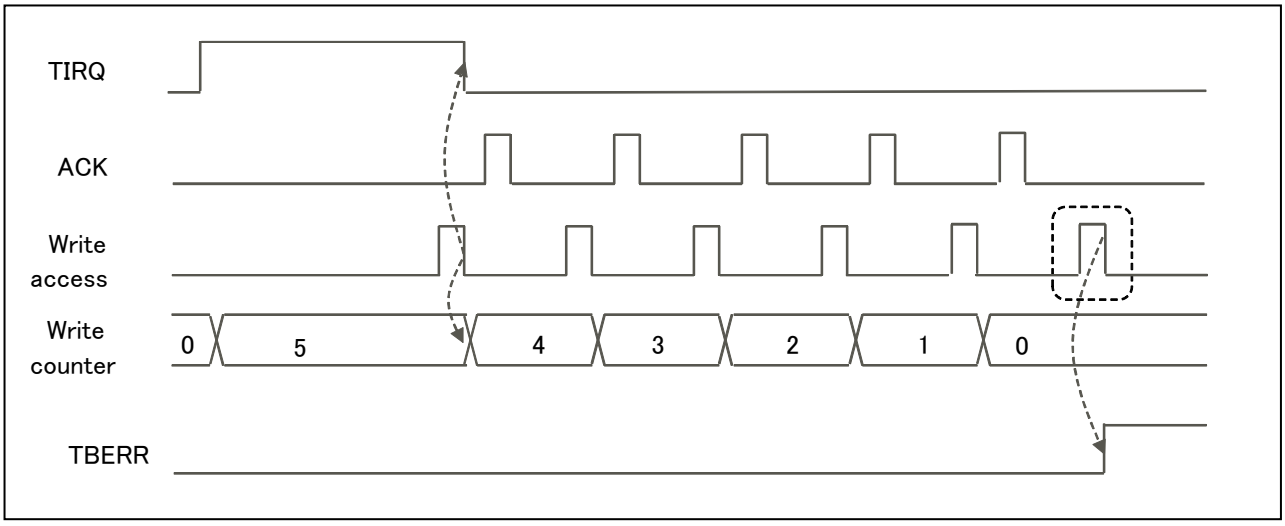
Figure 7-9 Reception Block Transfer Error in RW Access Mode



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 FSR:TBERR bit is asserted.

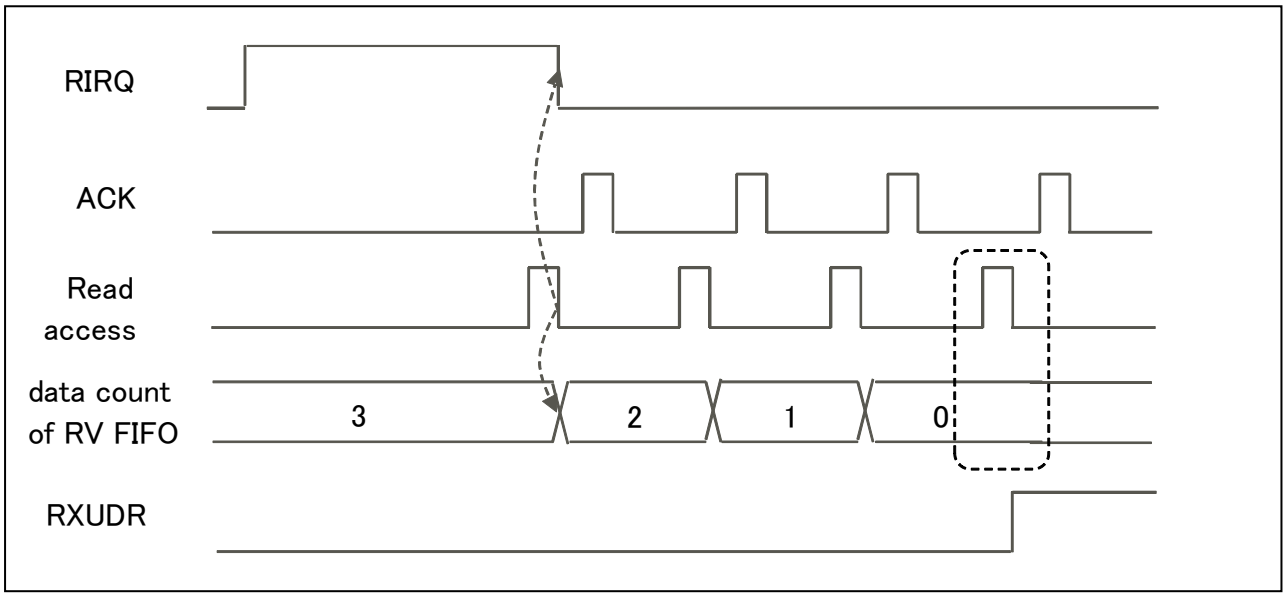
Figure 7-10 Transmission Block Transfer Error in RW Access Mode



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.

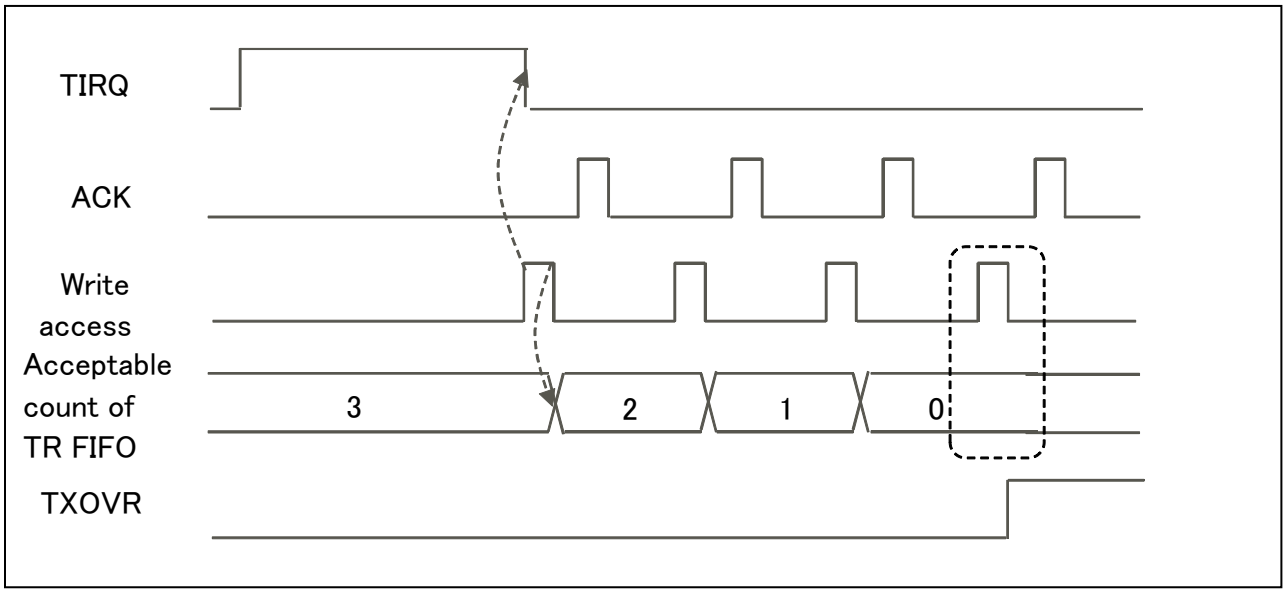
Figure 7-11 Reception FIFO Under Run in RW Access Mode



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 7-12 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)

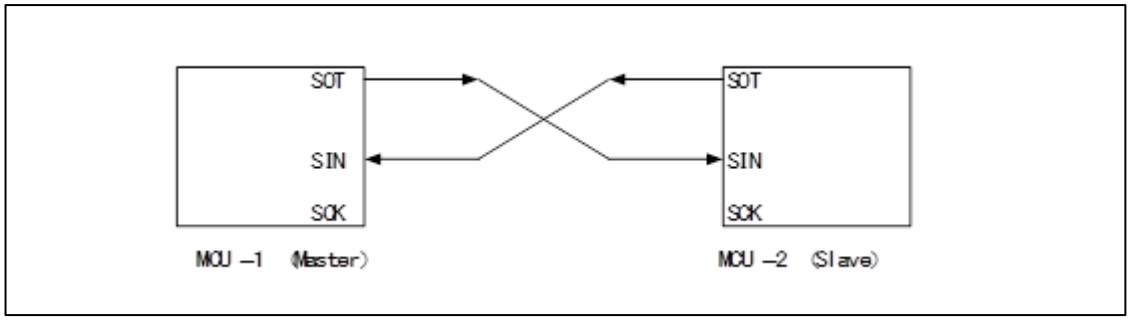
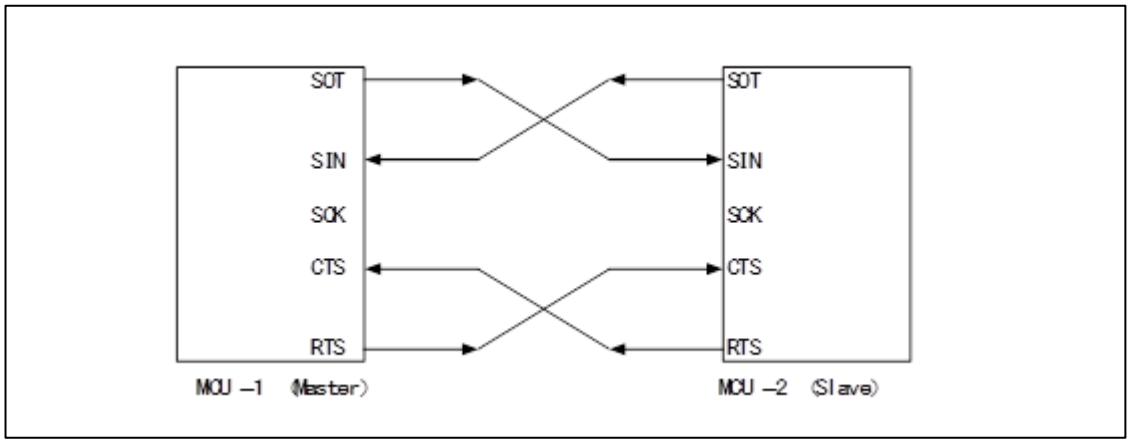


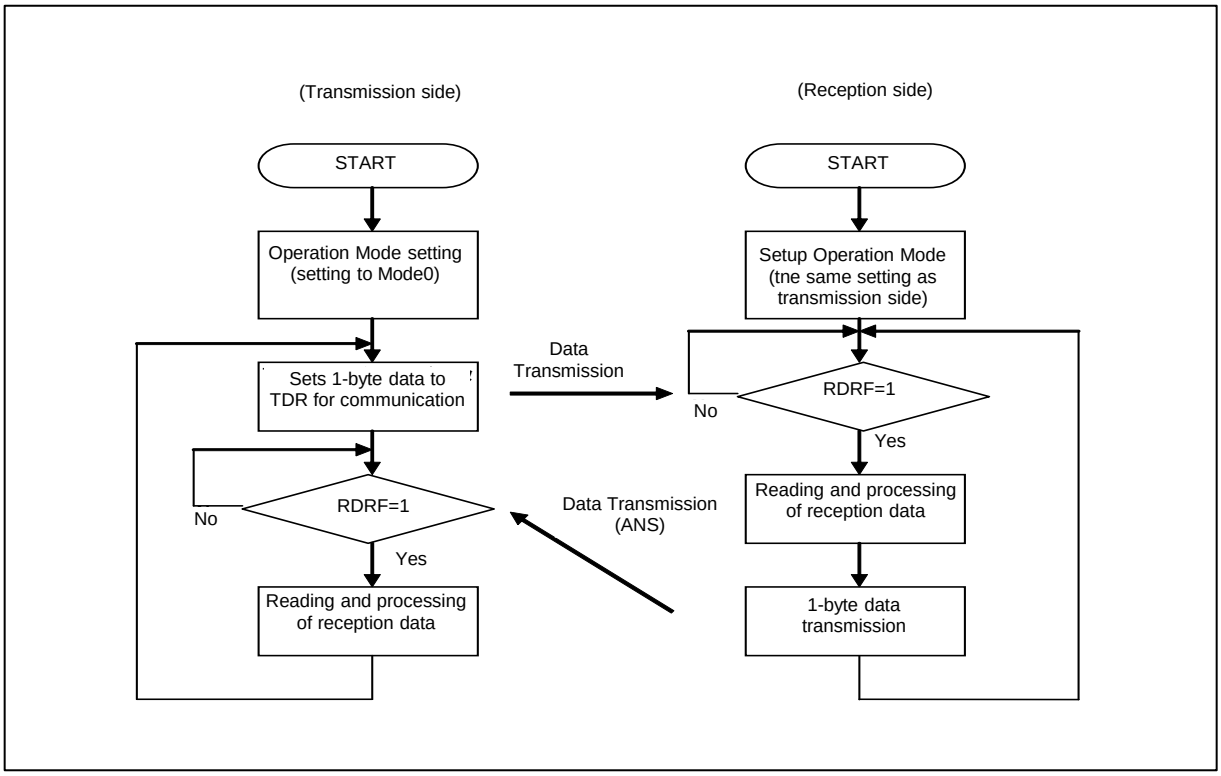
Figure 8-2 Example of Bidirectional Communication Connection for UART Operation Mode 0 (Flow Control)



Flow Chart

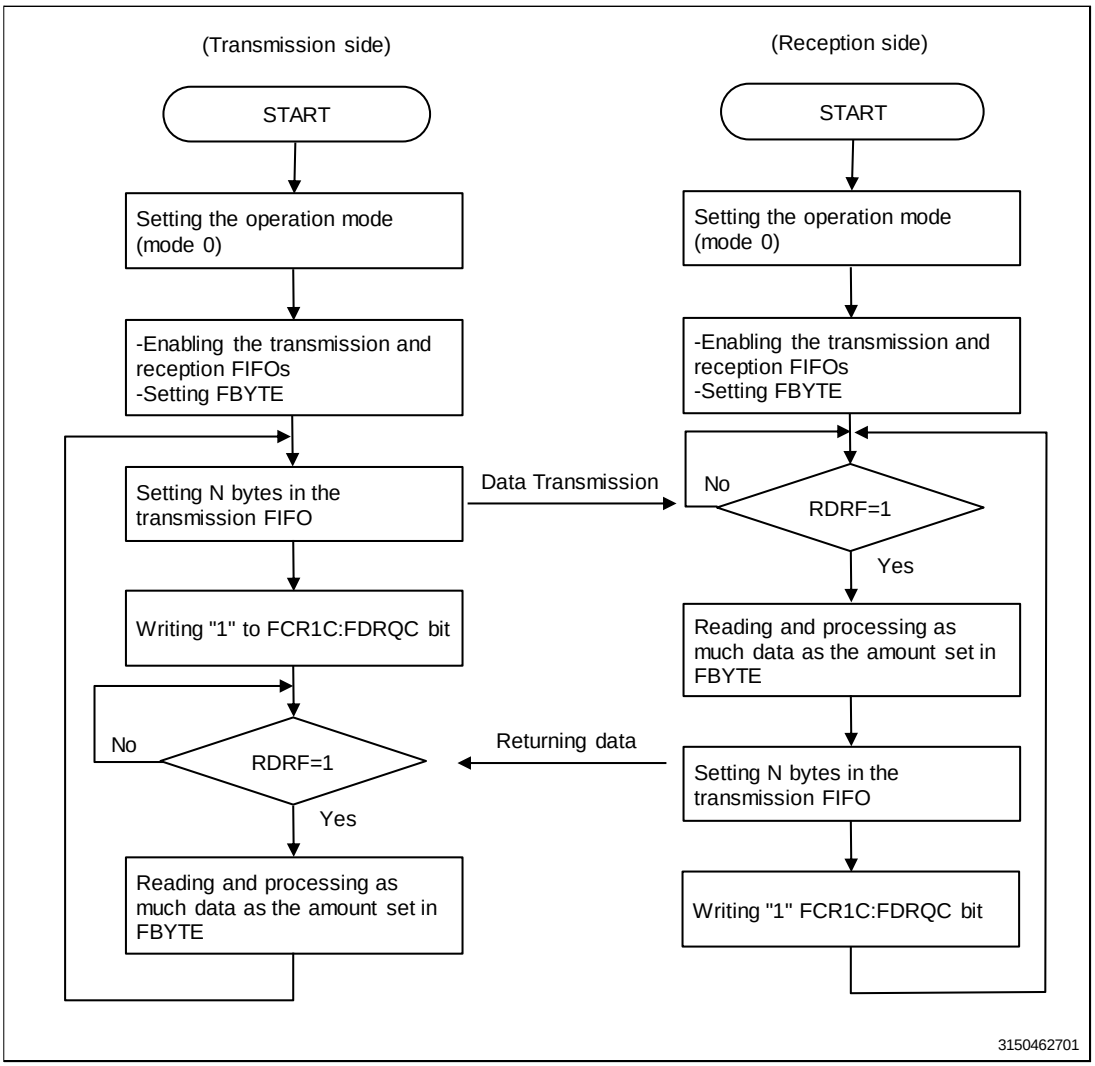
FIFO Not Used

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)



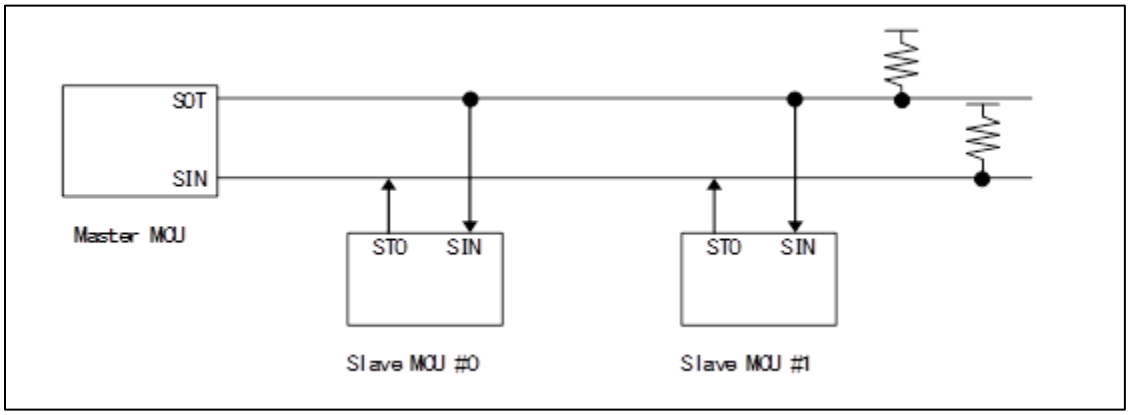
9. Setup Procedure and Program Flow for Operation Mode 1 (Asynchronous Multi-processor Mode)

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

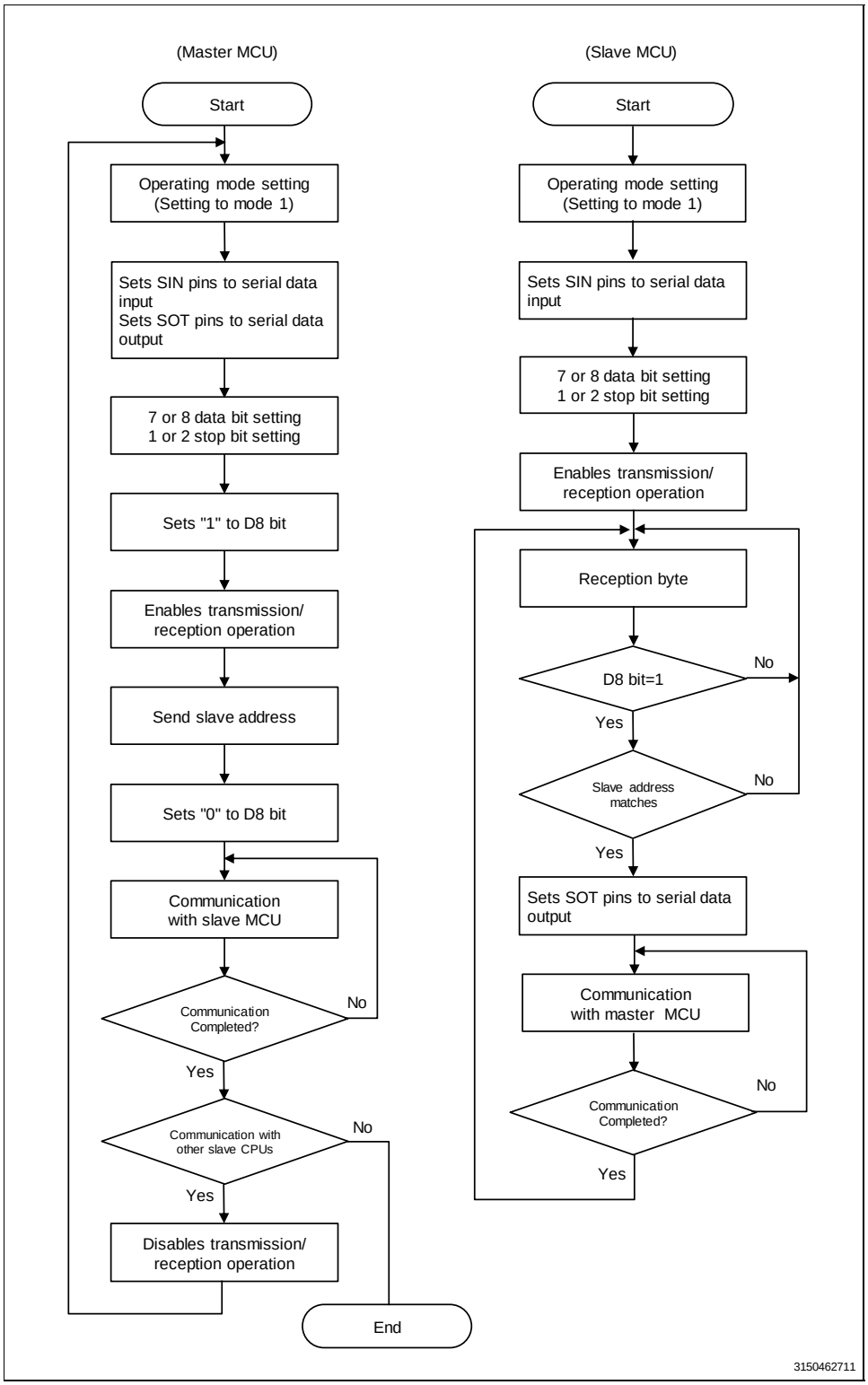
Communication starts when the master MCU transmits address data. Address data, for which the D8 bit is set to "1", selects the destination slave MCU. Each slave CPU checks the address data by its program, and if it coincides with the allocated address, the MCU communicates (by normal data) with the master MCU.

Figure 9-2 and Figure 9-3 provide flow charts of master/slave-type communication (multi-processor mode).

Flow Chart

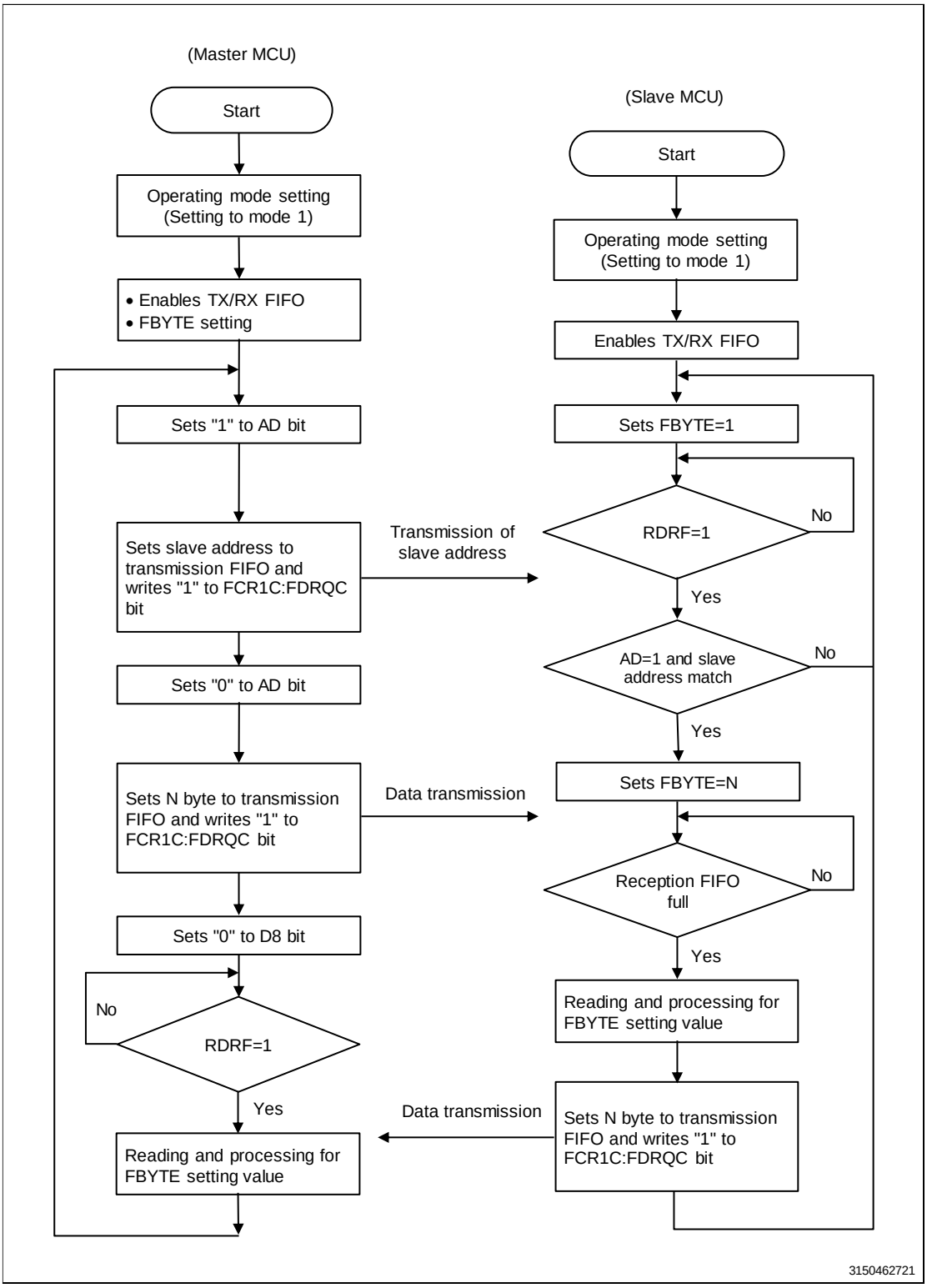
FIFO Not Used

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)
	-	-

	Bit15 Bit8	Bit7 Bit0
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 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)
	-	-

	Bit15 Bit8	Bit7 Bit0
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
	-	-
	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
	-	-

*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.

Notes:

- *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.

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).

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

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.

[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.

Note:

- *Set a SOUT pin as a peripheral output pin (set with PPC_PCFGRijj). See "CHAPTER: I/O PORT" for how to make setups.*

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

[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

[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

[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.

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).*

[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" when the UPCL bit in the serial control register (SCR) is set to "1".
- For set/reset timing of the TDRE bit when the transmission FIFO is used, see "CHAPTER 34:2.4 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" when the UPCL bit in the serial control register (SCR) is set to "1".
- 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 and transmission data registers 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 by16-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 34:2.4 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	R0,W0				R/W	R/W	R,WX
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 multifunction 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:INT) 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 Resister

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 0x00000. 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 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,WX	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 FRC1C: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.*

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

- FIFO reset (writing "1" to FCL)
- 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.*

[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.*

[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 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.
- 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.*
- *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.*
- *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 after disabling the reception operation.*
- *Any setting that exceeds the FIFO capacity is prohibited.*

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.*

[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.*

[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).

- 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 35: CSIO (Clock Synchronous Serial Interface)

This chapter explains the CSIO functions of the multifunction 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)**

Note:

- *ACK Mode of Block Transfer is not supported for this device.*

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 buffer (when FIFO is not used) − Transmission and reception FIFOs (64 bytes each)*1 (when FIFO is used)
2	Transfer format	− Clock synchronization (no start/stop bits) − Master/slave functions − Supports SPI (supports both master and slave).
3	Baud rate	− A 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, reception block transfer error) − Reception FIFO interrupt (reception FIFO under run) − Transmission interrupt (transmission data empty, transmission bus idle, chip error interrupt, 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) − A DMA Transfer support function is provided for both transmission and reception. − Status interrupt (serial timer interrupt)
7	Serial chip select	− 4-channel control (independent control, rounding control) − The setup/hold/deselect times can be made 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. − An operating clock division value can be selected (division by 1 to 256). − Can be activated with an external trigger.
10	Synchronous mode	Master or slave function
11	Pin access	The serial data output pin can be set to "1".
12	FIFO option	− Incorporates transmission and reception FIFOs (64 bytes for transmission FIFO, 64 bytes for reception FIFO) − The transmission and reception FIFOs 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)
			Reception of as much data as the value set in FBYTE		Reading of reception data until the reception FIFO is empty
			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)
	RXUDR	ESR	Reception FIFO under run	ECR:REIE	Writing "1" to the transmission FIFO under run flag clear bit (ESRC:RXUDRC)
	RBERR	ESR	Reception block transfer error	ECR:REIE	Writing "1" to the reception block transfer error clear bit (ESRC:RBERRC)

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
Transmission	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)
	TXOVR	ESR	Transmission FIFO overrun	ECR:TEIE	Writing "1" to the transmission FIFO overrun flag clear bit (ESRC:TXOVRC)
	TBERR	ESR	Transmission block transfer error	ECR:TEIE	Writing "1" to the transmission block transfer error clear bit (ESRC:TBERRC)
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)

*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 (SSR: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

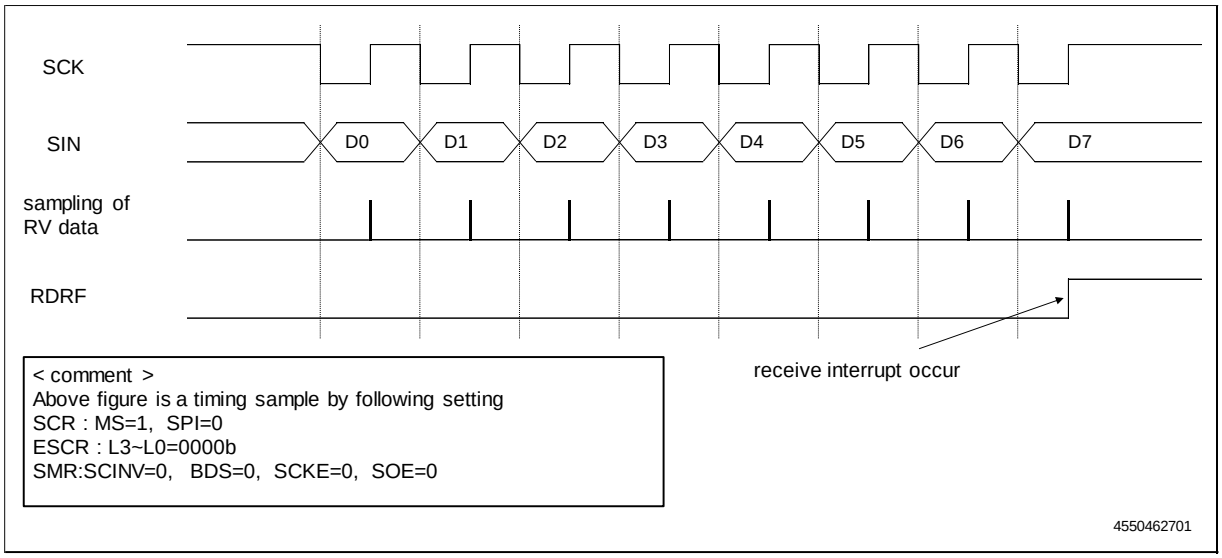
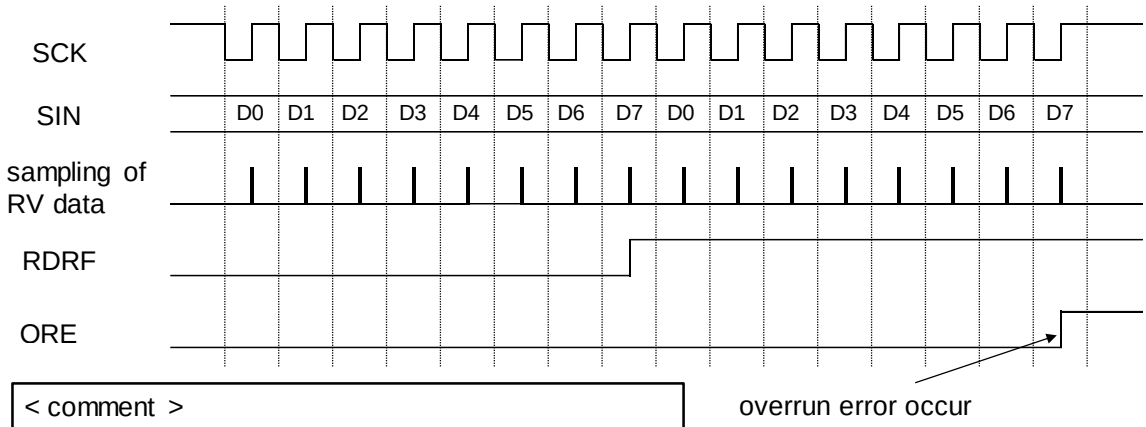


Figure 2-2 ORE (Overrun Error) Flag Set Timing



< 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

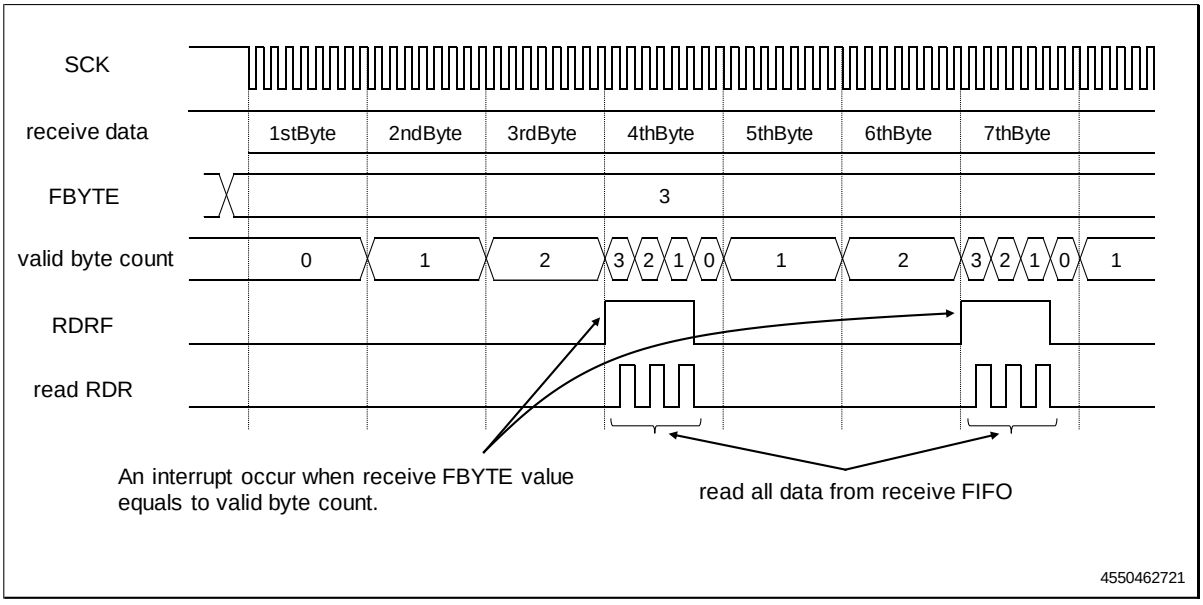
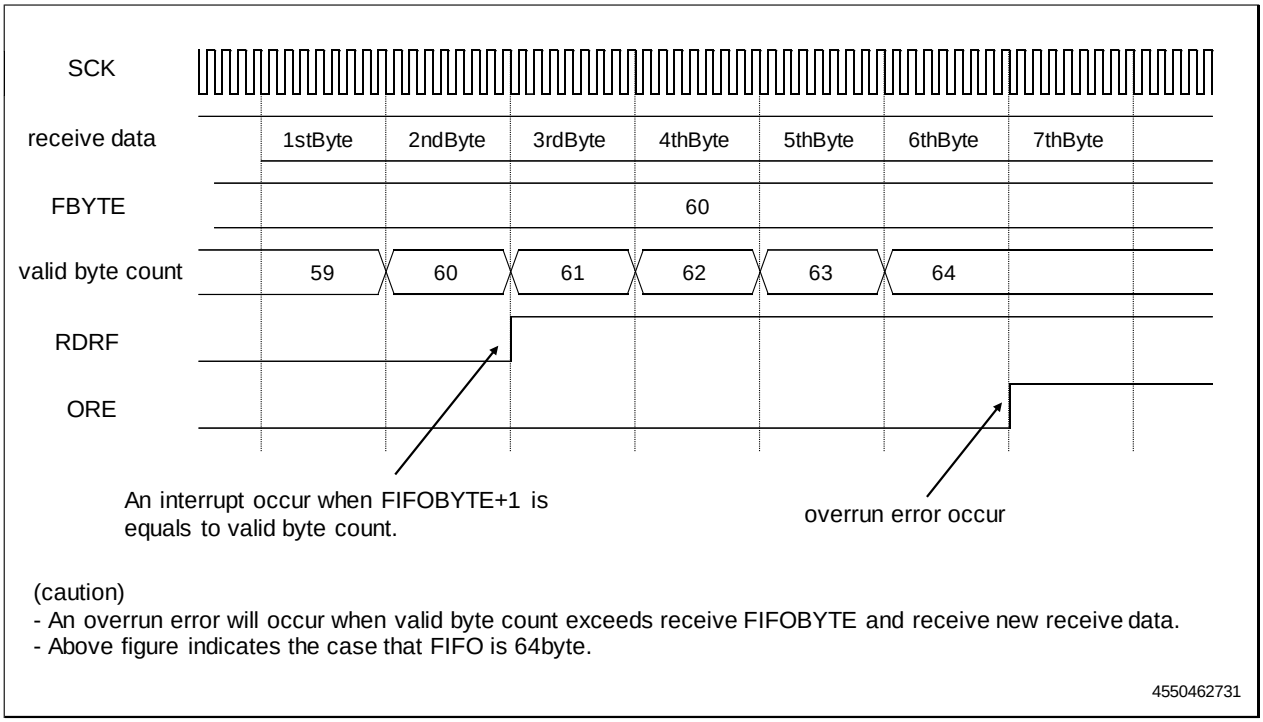


Figure 2-4 Set Timing of ORE (Overrun Error) Flag Bit



2.3. Transmission Interrupt Generation and Flag Set Timing

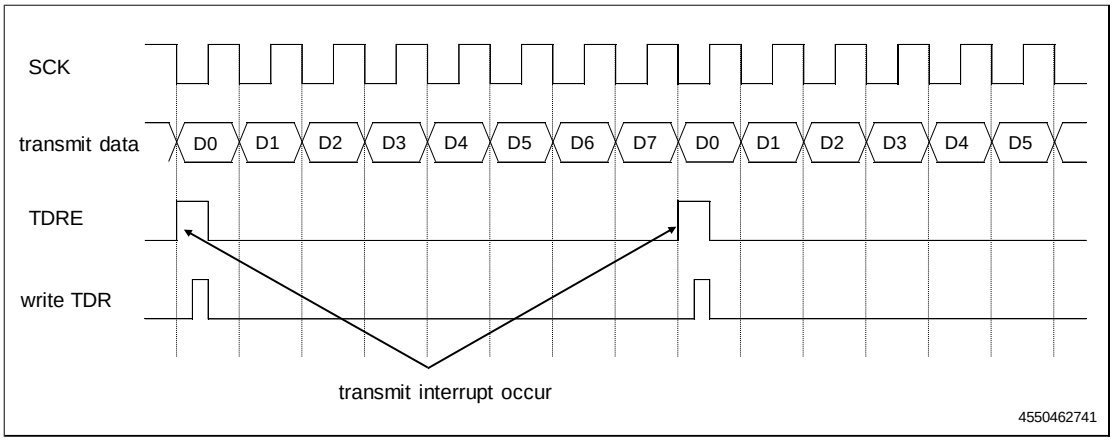
An interrupt during transmission is generated if transmission data is transferred from the transmission data register (TDR) to the transmission shift register (SSR:TDRE = 1) and transmission is started, or if transmission is not being performed (SSR:TBI = 1).

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).

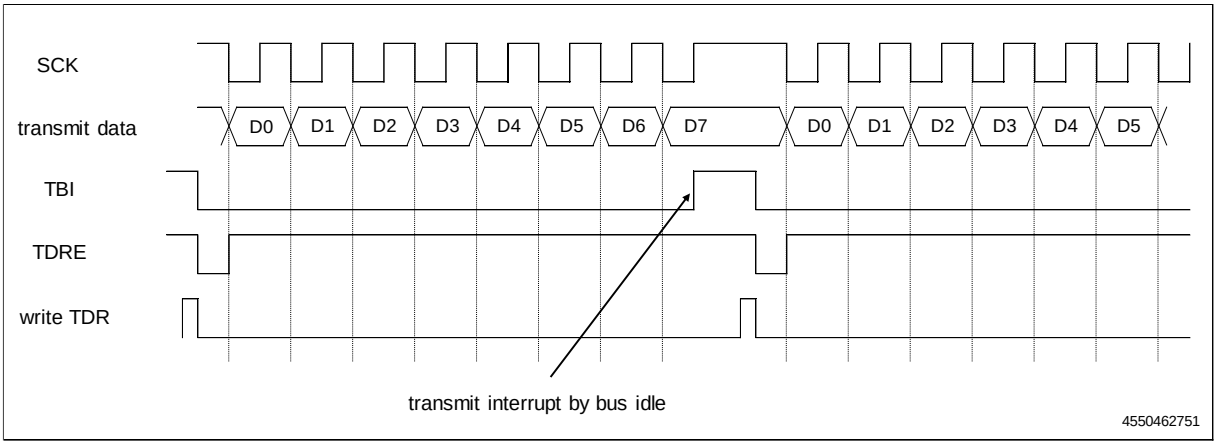
Figure 2-5 Set Timing of Transmission Data Empty Flag (SSR:TDRE) (SCR:MS = 0, SCR:SPI = 0)



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

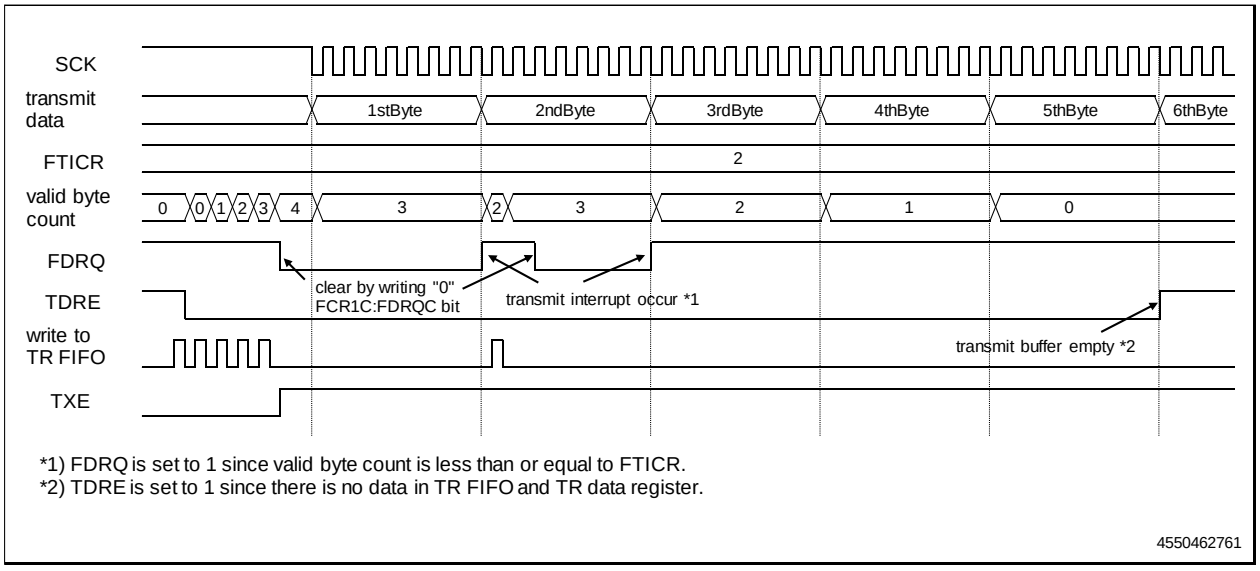
When the transmission FIFO is used, an interrupt is generated if the number of data items stored in the transmission FIFO is equal to or less than the setting made for the FTICR register (FTICR).

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



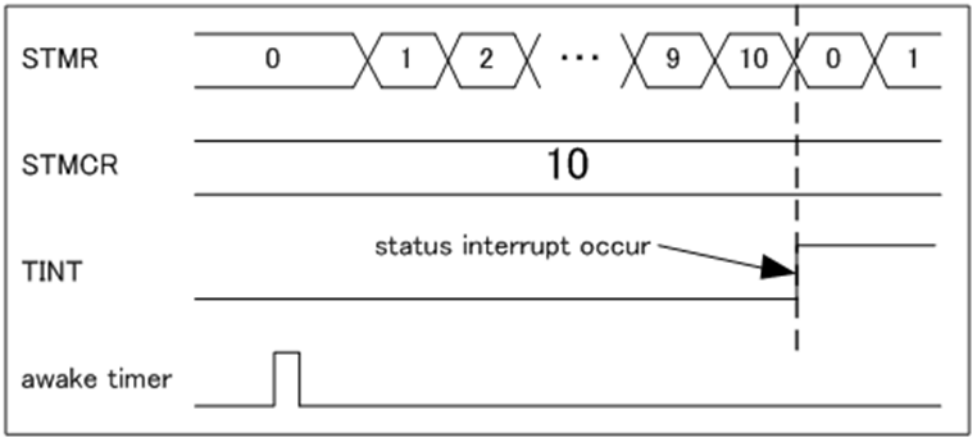
2.5. Timer Interrupt Generation and Flag Set Timing

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

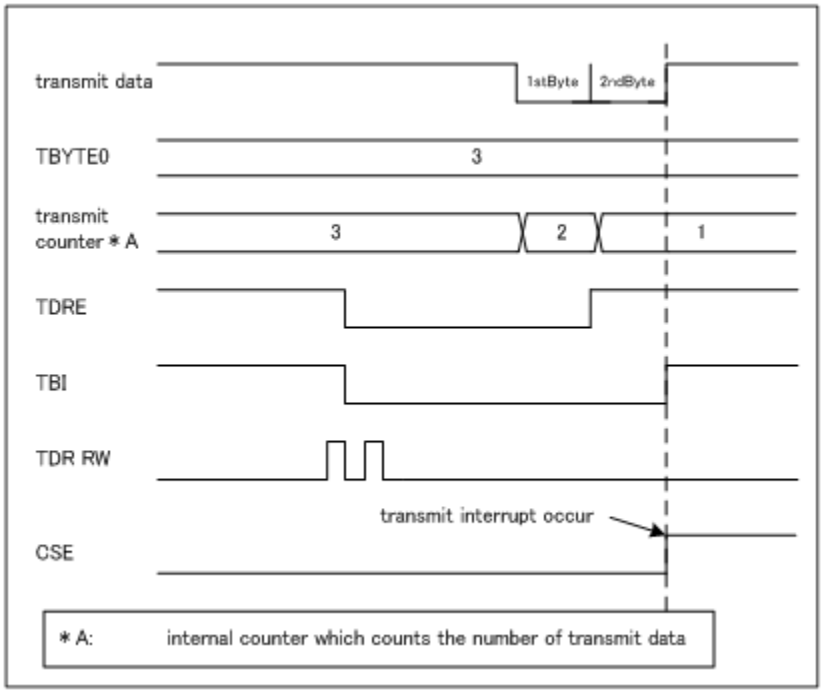
Master Mode (SCR:MS = 0)

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)

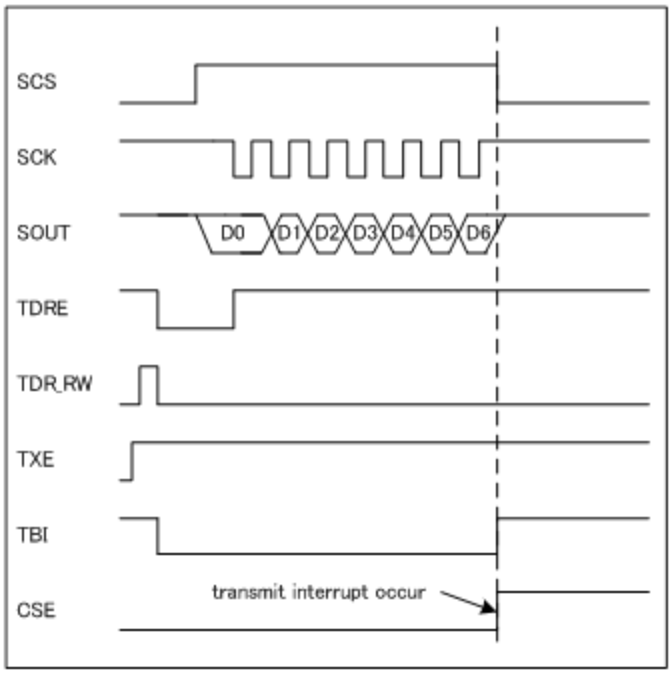
A chip select error occurs if chip select becomes inactive in any of the following cases.

- 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

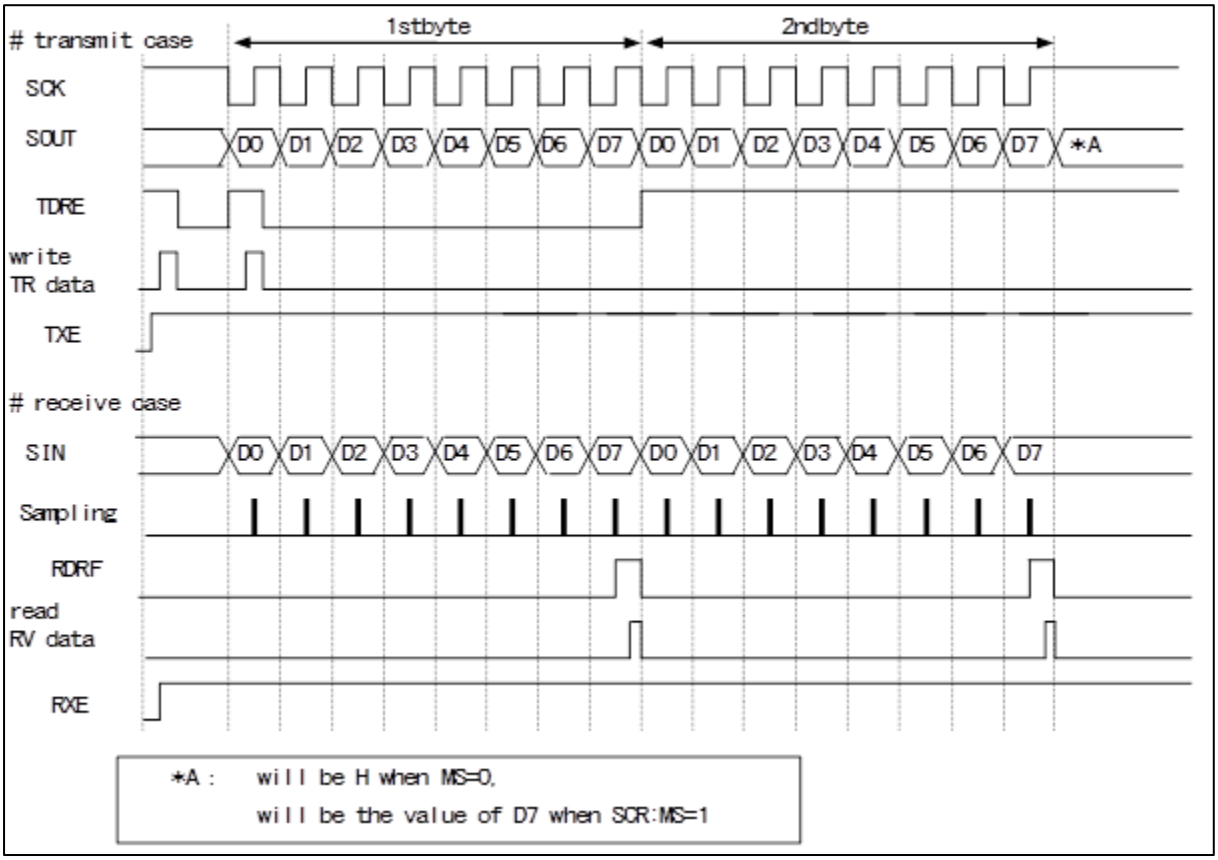
During slave operation: SCR:MS = 1, SMR:SCKE = 0

*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:

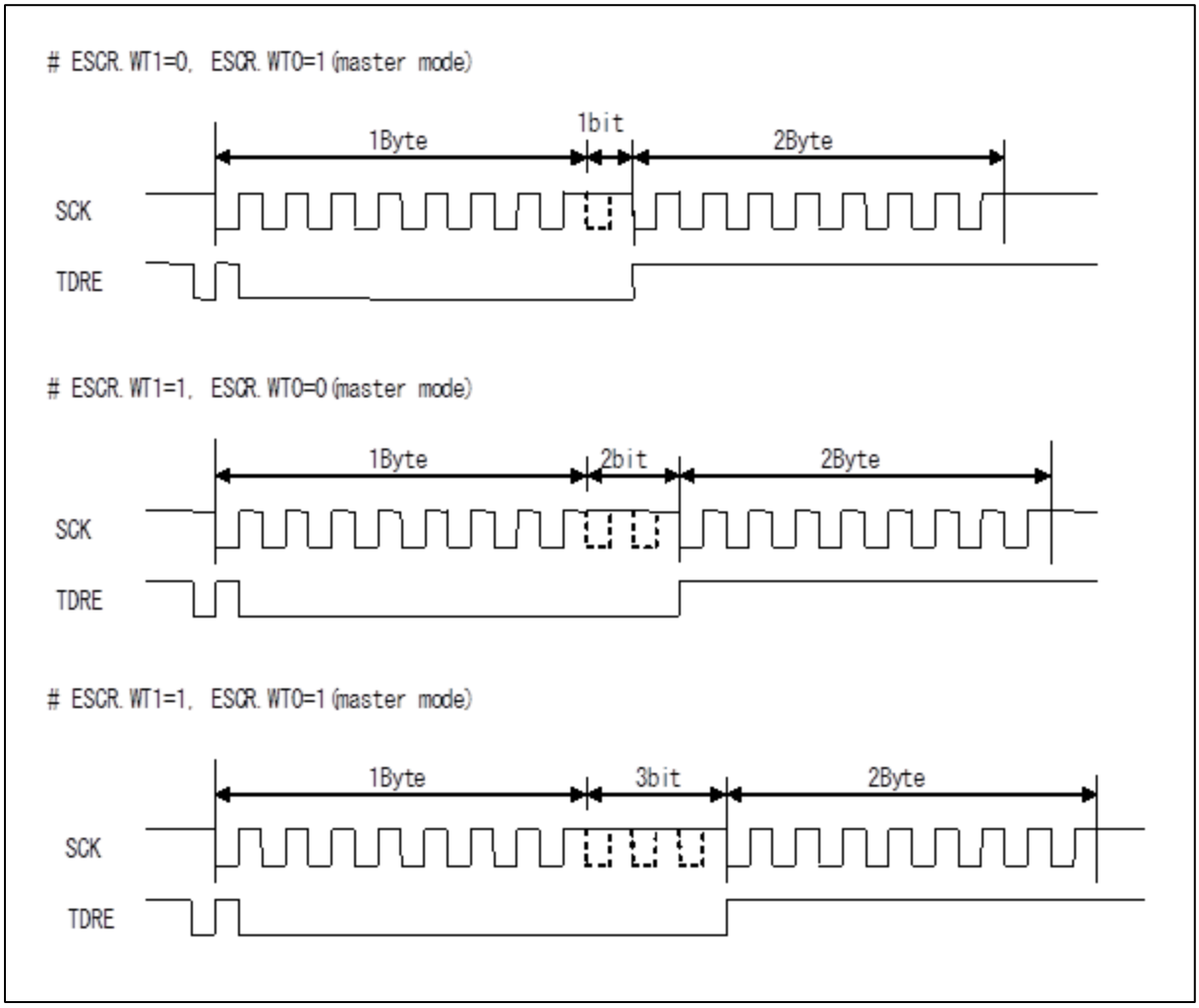
- *If performing reception operation only, write dummy data to TDR to output the serial clock (SCK).*
- *When the transmission/reception FIFOs are enabled, setting the number of frames to transfer in the FBYTE register causes as many serial clock (SCK) pulses as the number of frames to be output..*

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.*

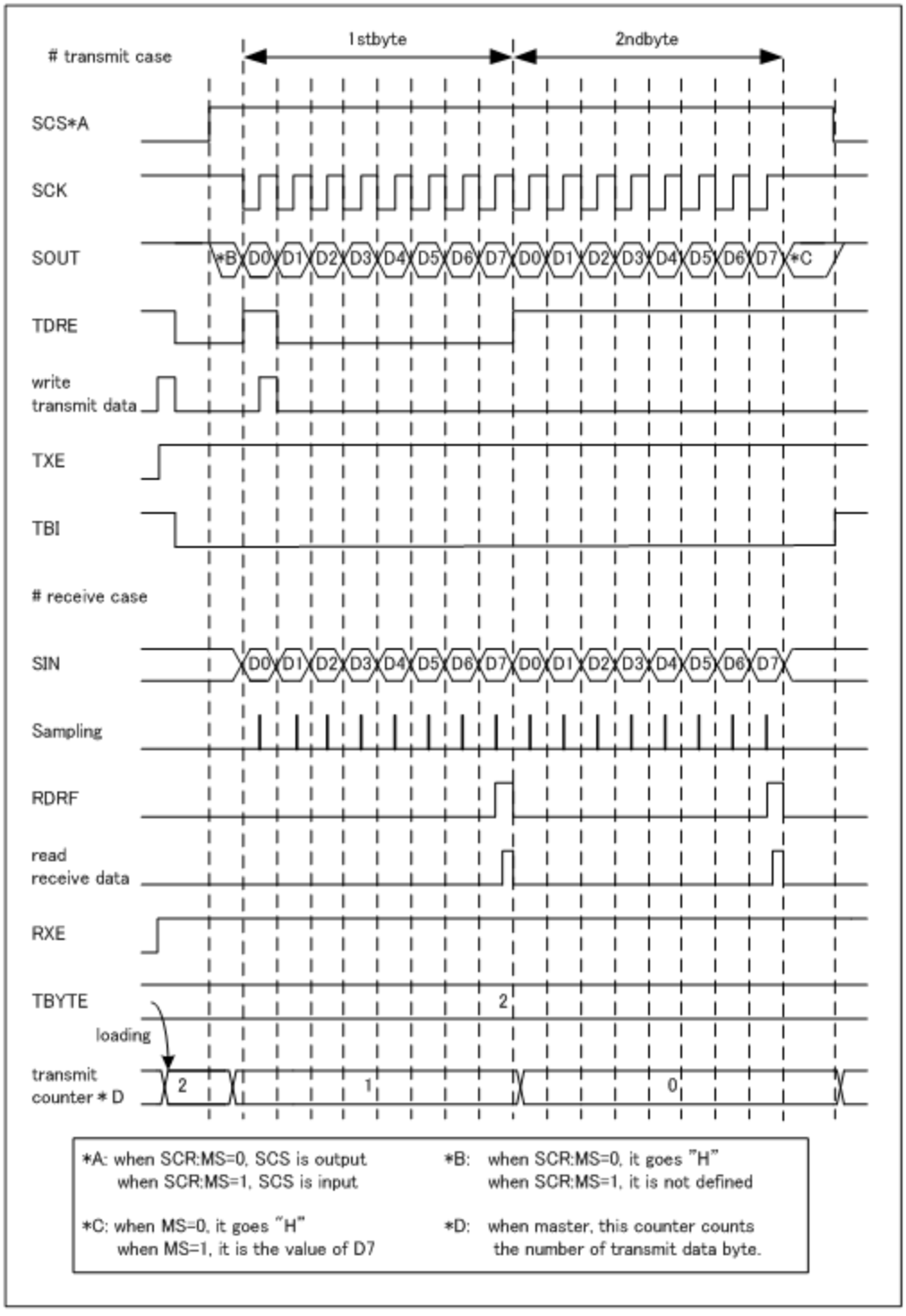
Reception Operation

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

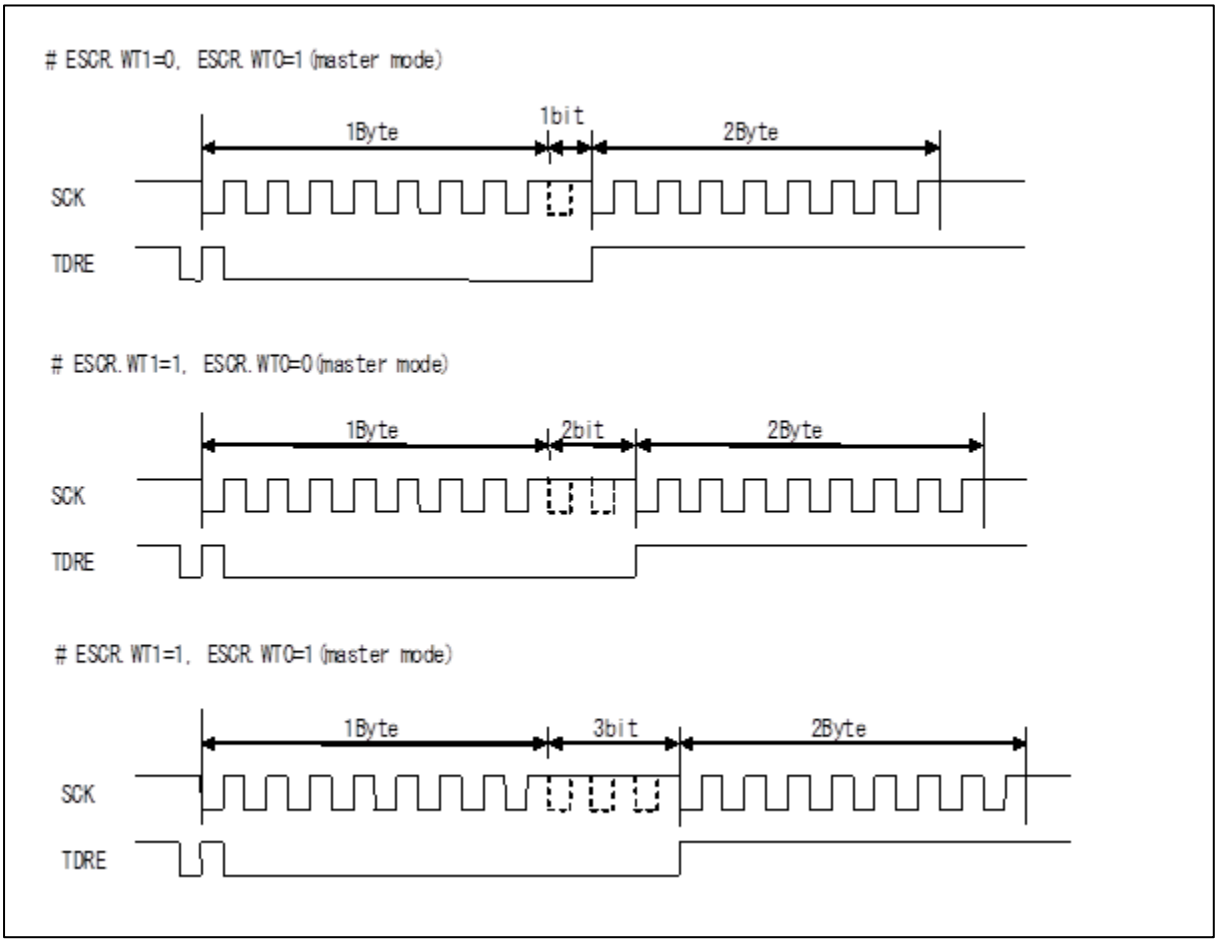
1. To perform transmission/reception at the same time, enable serial data output (SMR:SOE = 1) as well as transmission/reception (SCR:TXE, RXE = 1).
2. When transmission data is written to TDR, SSR:TDRE is set to 0. Later, the serial chip select pin (SCS) becomes active, and after the elapse of the serial 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

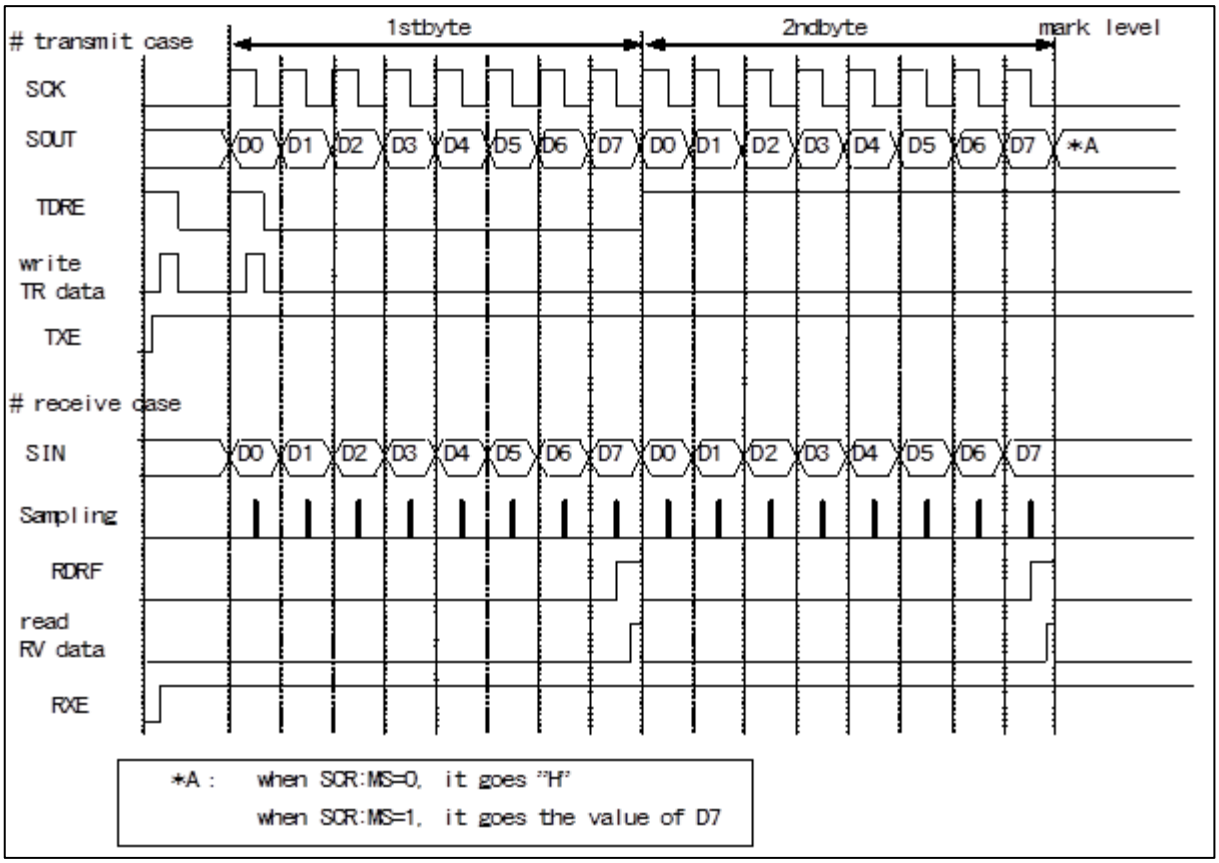
During slave operation: SCR:MS = 1, SMR:SCKE = 0

*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:

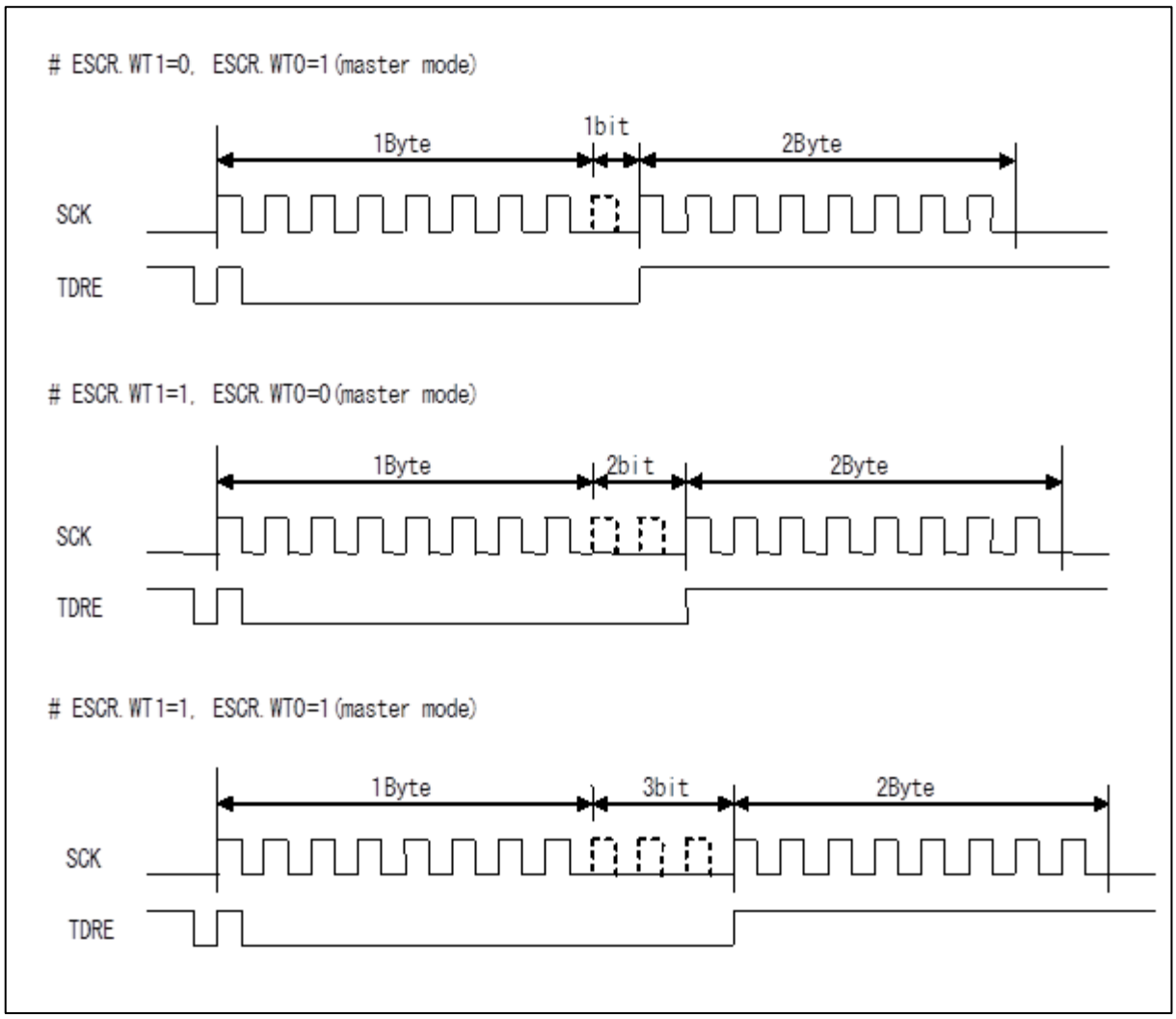
- *When performing reception only, write dummy data to TDR to output the serial clock (SCK).*
- *When the transmission/reception FIFOs are enabled, setting the number of frames to transfer in the FBYTE register causes as many serial clock (SCK) pulses as the number of frames to be output.*

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.*

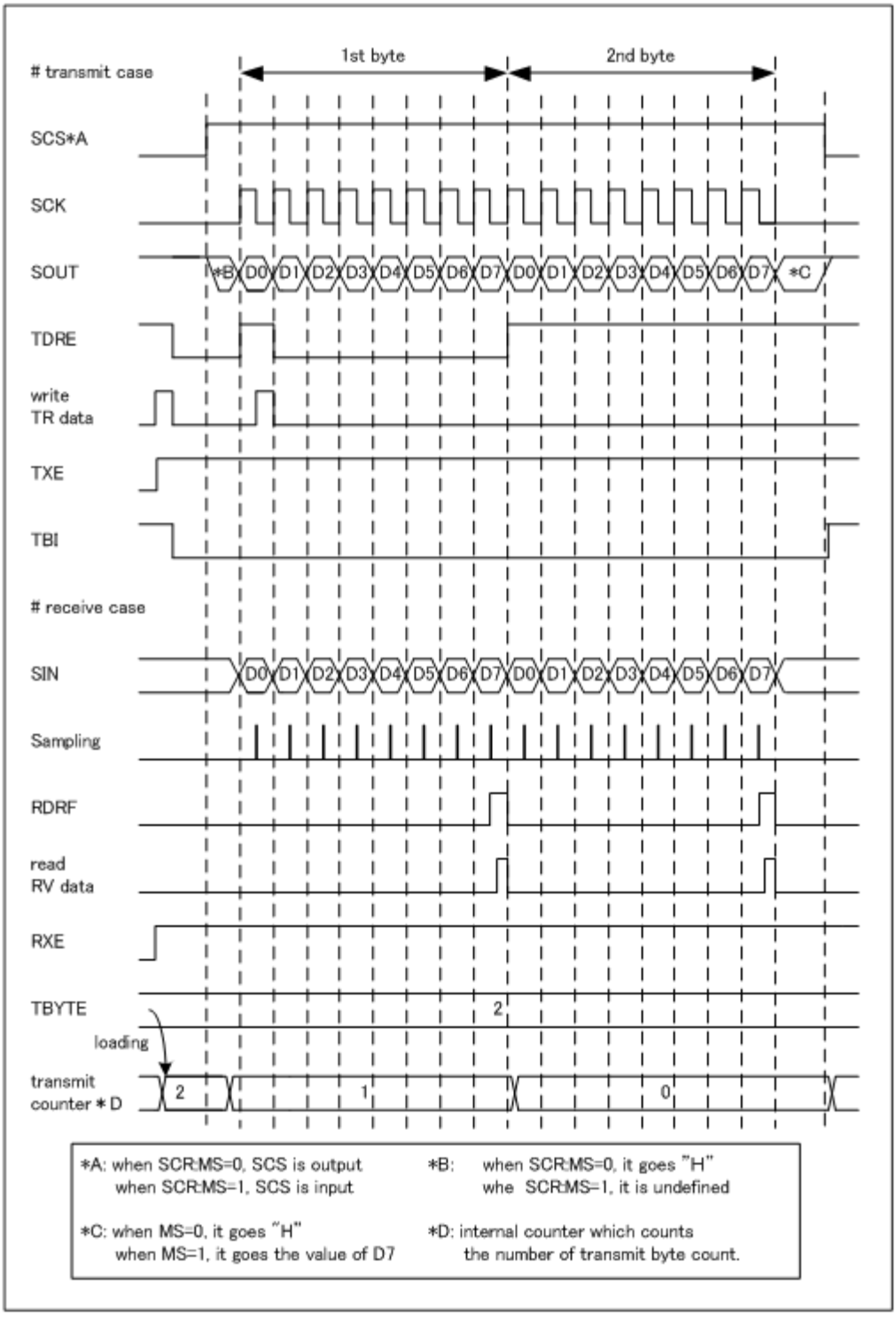
Reception Operation

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

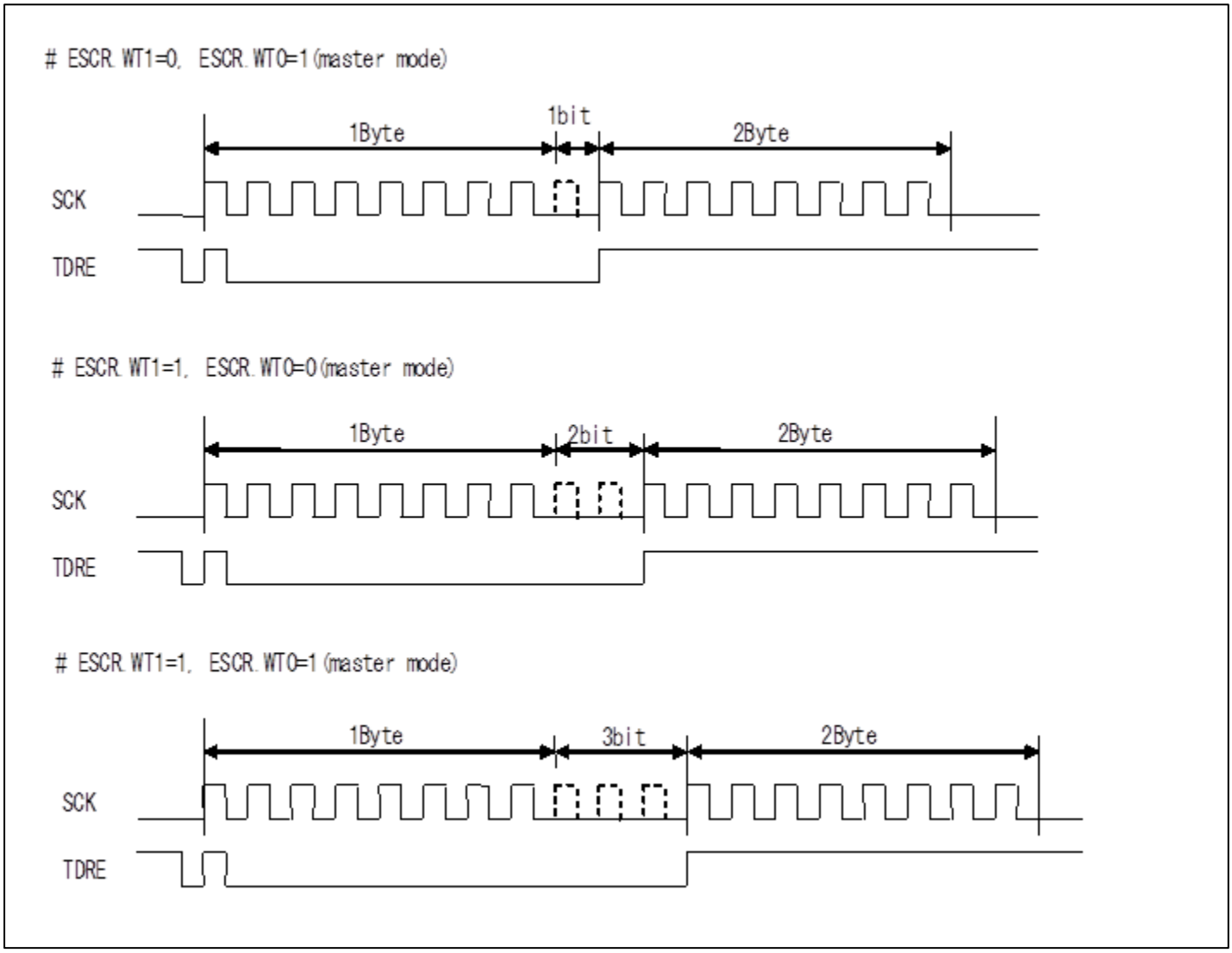
1. To perform transmission/reception at the same time, enable serial data output (SMR:SOE = 1) and also transmission/reception (SCR:TXE, RXE = 1).
2. When transmission data is written to TDR, SSR:TDRE is set to 0. Later, the serial chip select pin (SCS) becomes active, and after the elapse of the serial 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

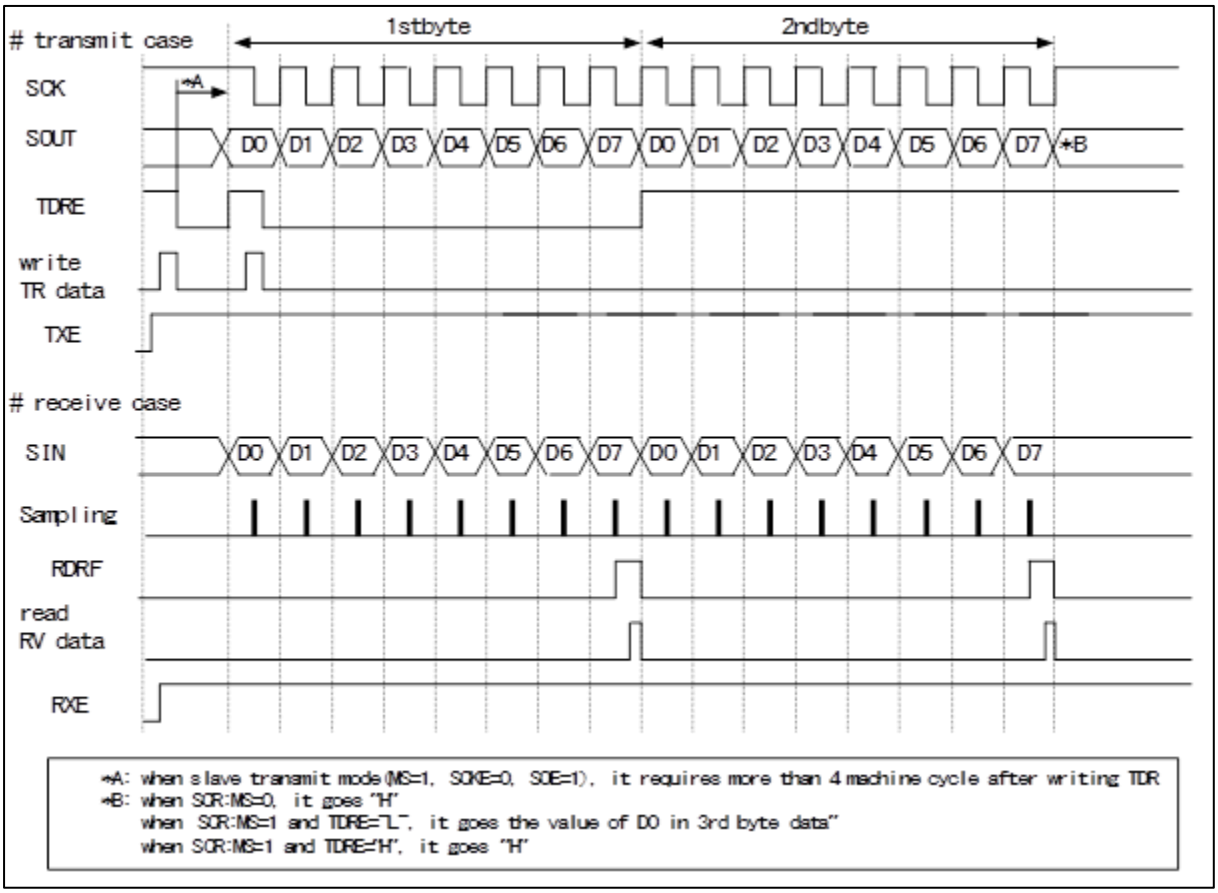
During slave operation: SCR:MS = 1, SMR:SCKE = 0

*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:

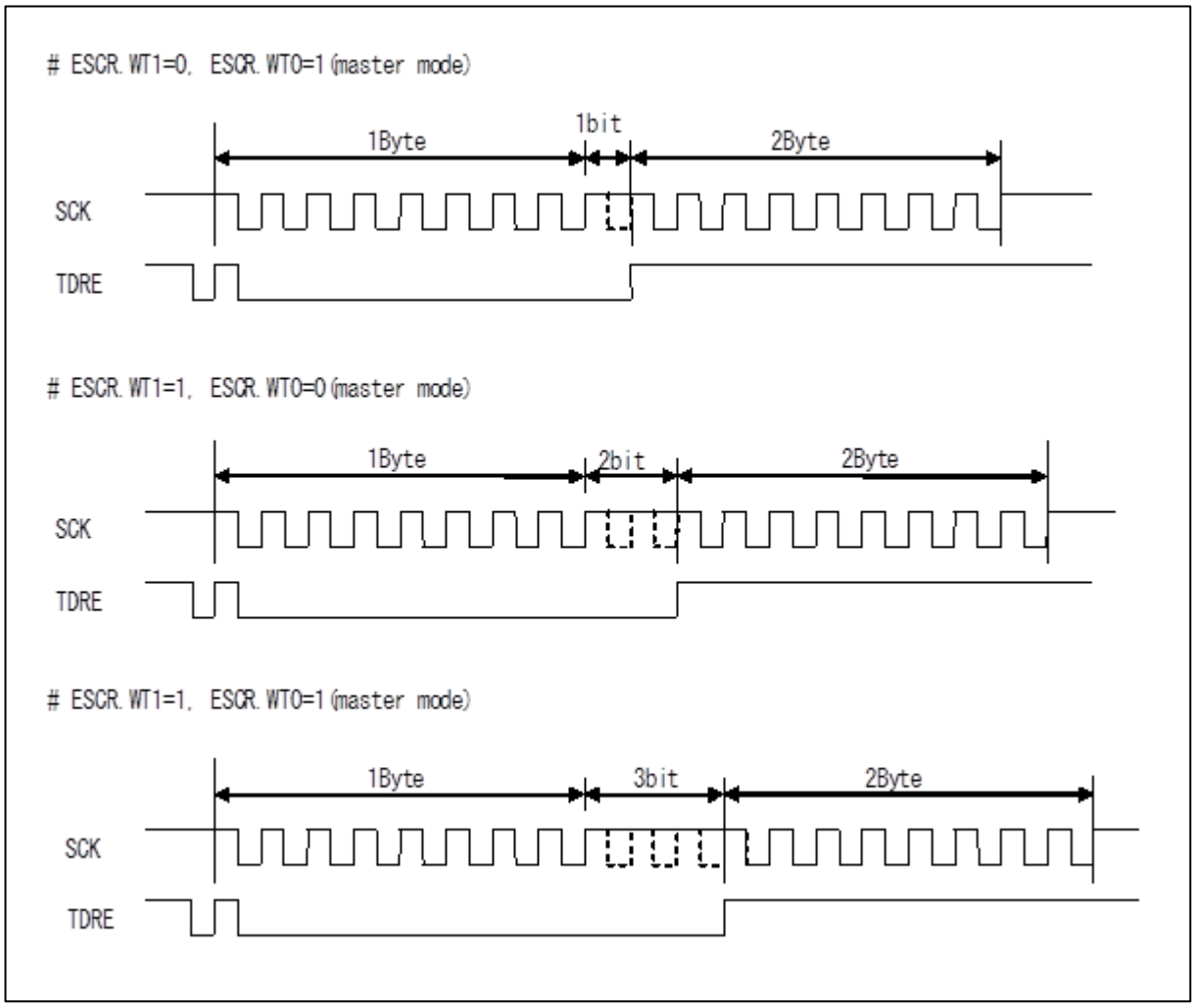
- *When performing reception only, write dummy data to TDR to output the serial clock (SCK).*
- *When the transmission/reception FIFOs are enabled, setting the number of frames to transfer in the FBYTE register causes as many serial clock (SCK) pulses as the number of frames to be output.*

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".

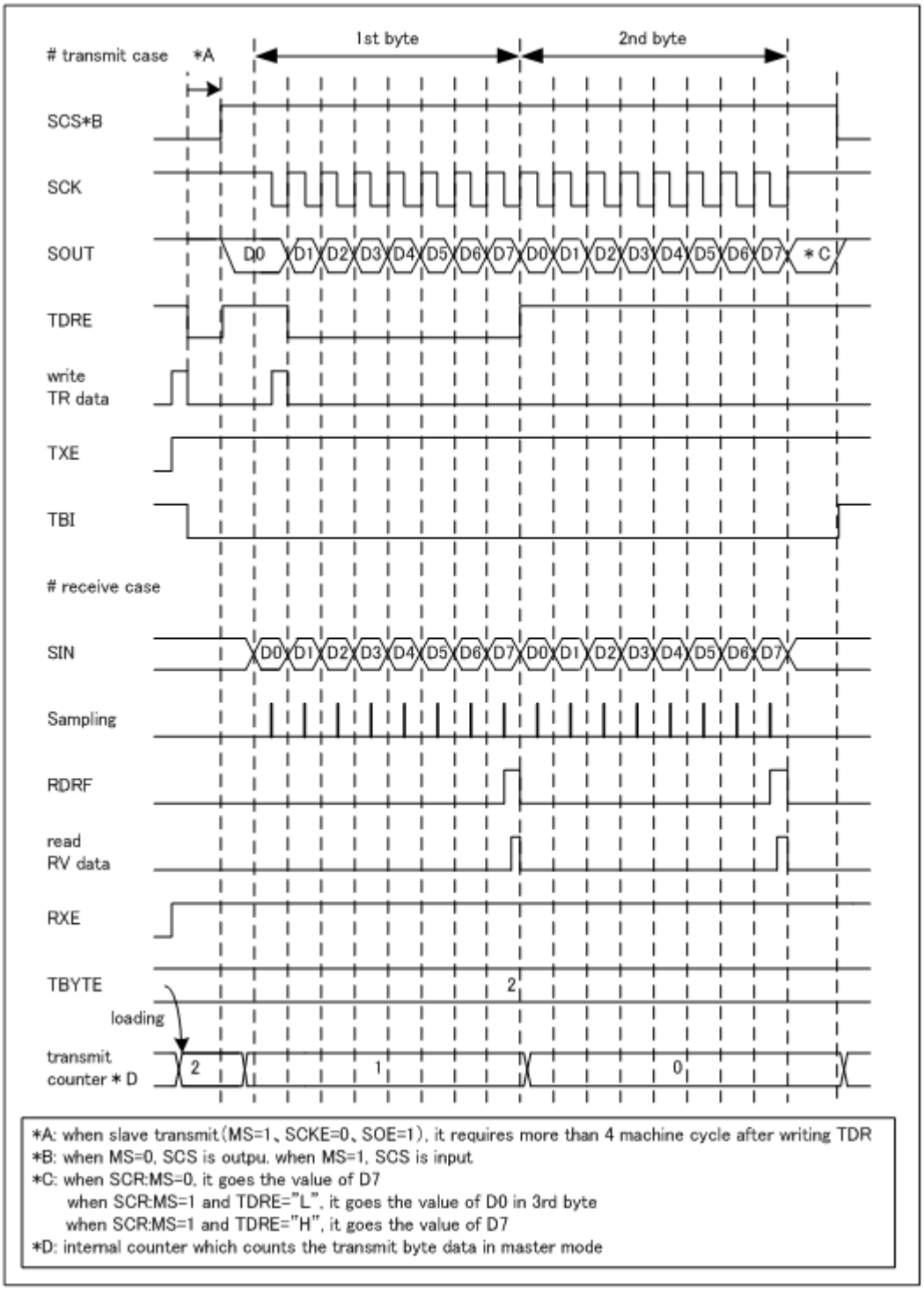
Transmission/Reception Operations

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)



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

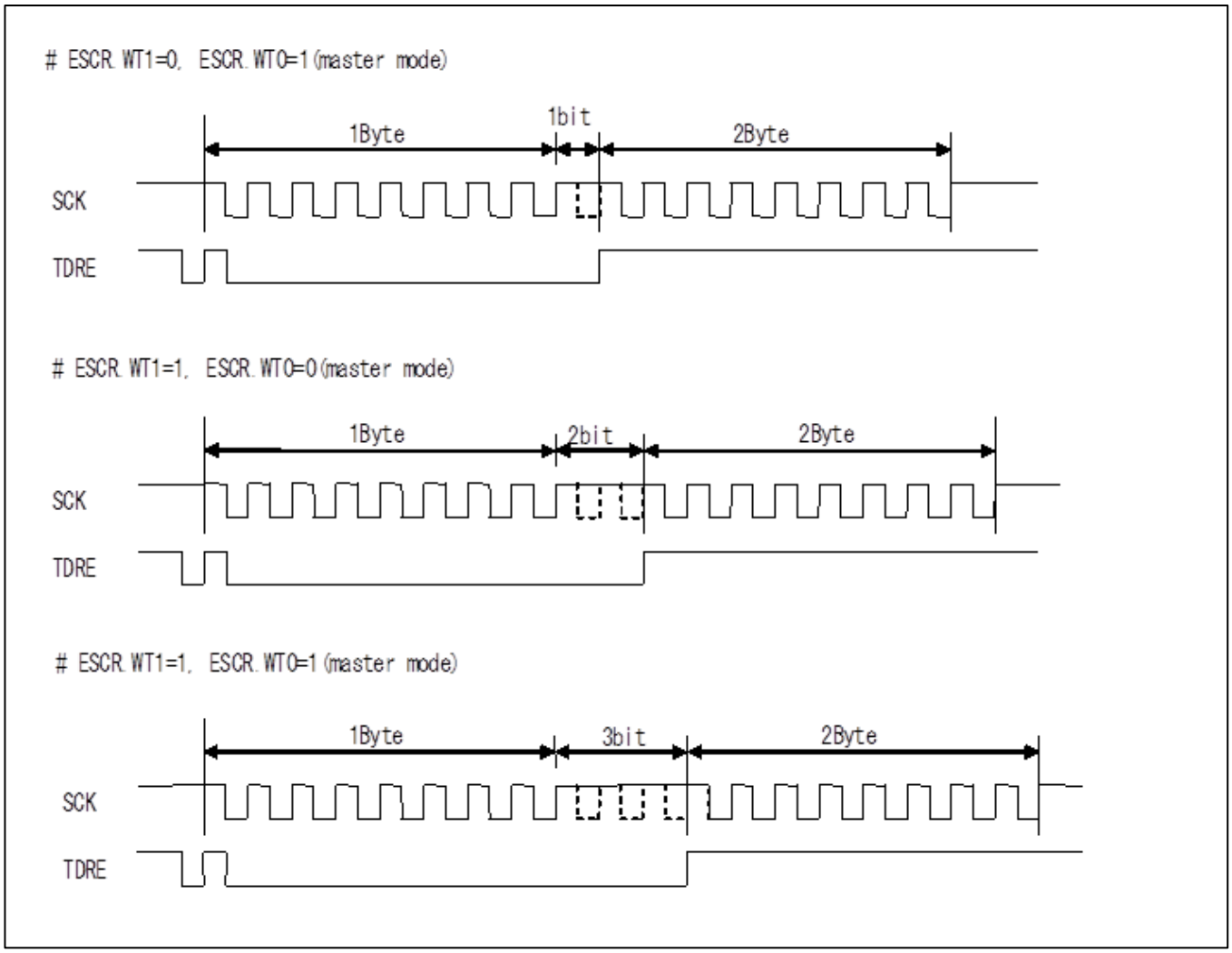
1. To perform transmission/reception at the same time, enable serial data output (SMR:SOE = 1) and also transmission/reception (SCR:TXE, RXE = 1).
2. When transmission data is written to TDR, SSR:TDRE is set to 0. Later, at the same time as the first bit is output, the serial chip select pin (SCS) becomes 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

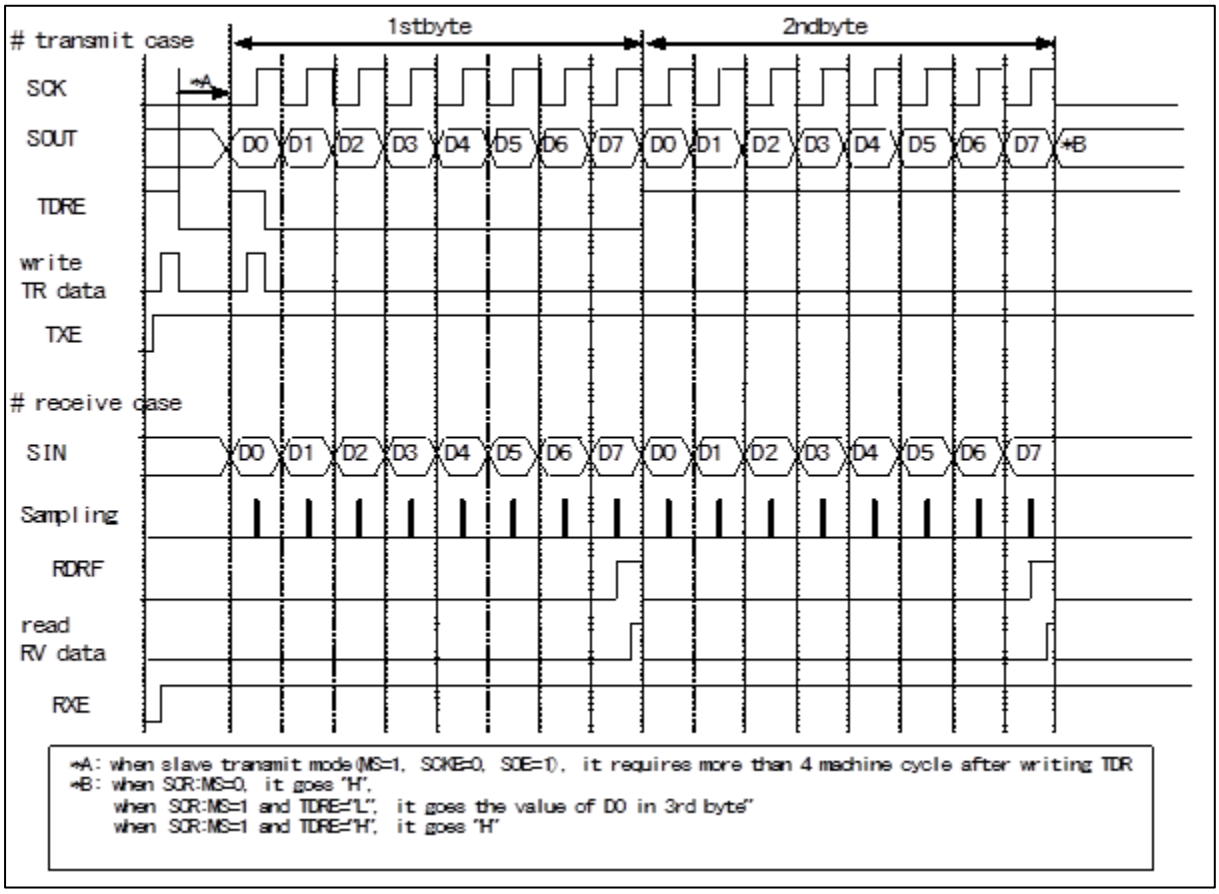
During slave operation: SCR:MS = 1, SMR:SCKE = 0

*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:

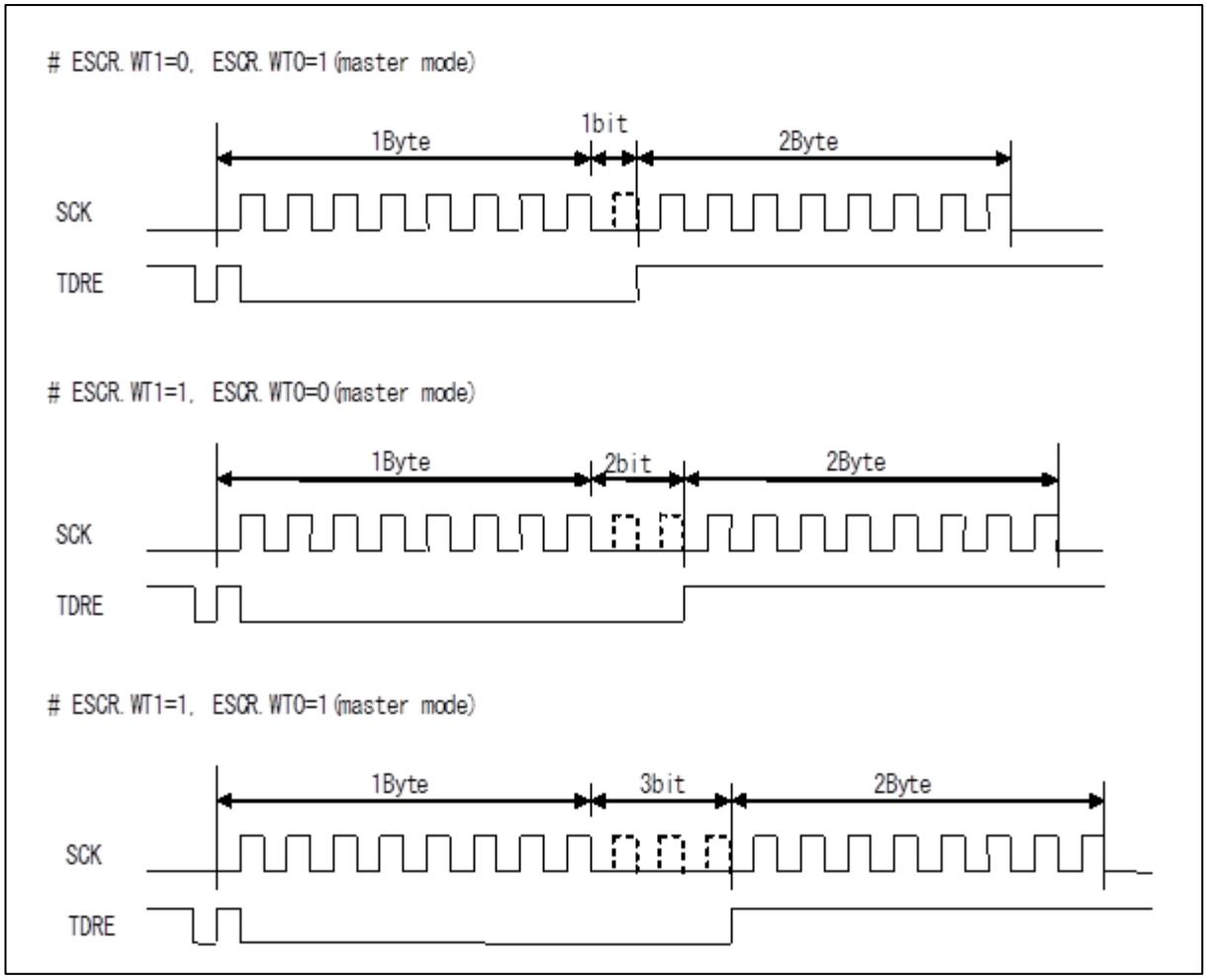
- *When performing reception only, write dummy data to TDR to output the serial clock (SCK).*
- *When the transmission/reception FIFOs are enabled, setting the number of frames to transfer in the FBYTE register causes as many serial clock (SCK) pulses as the number of frames to be output.*

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".

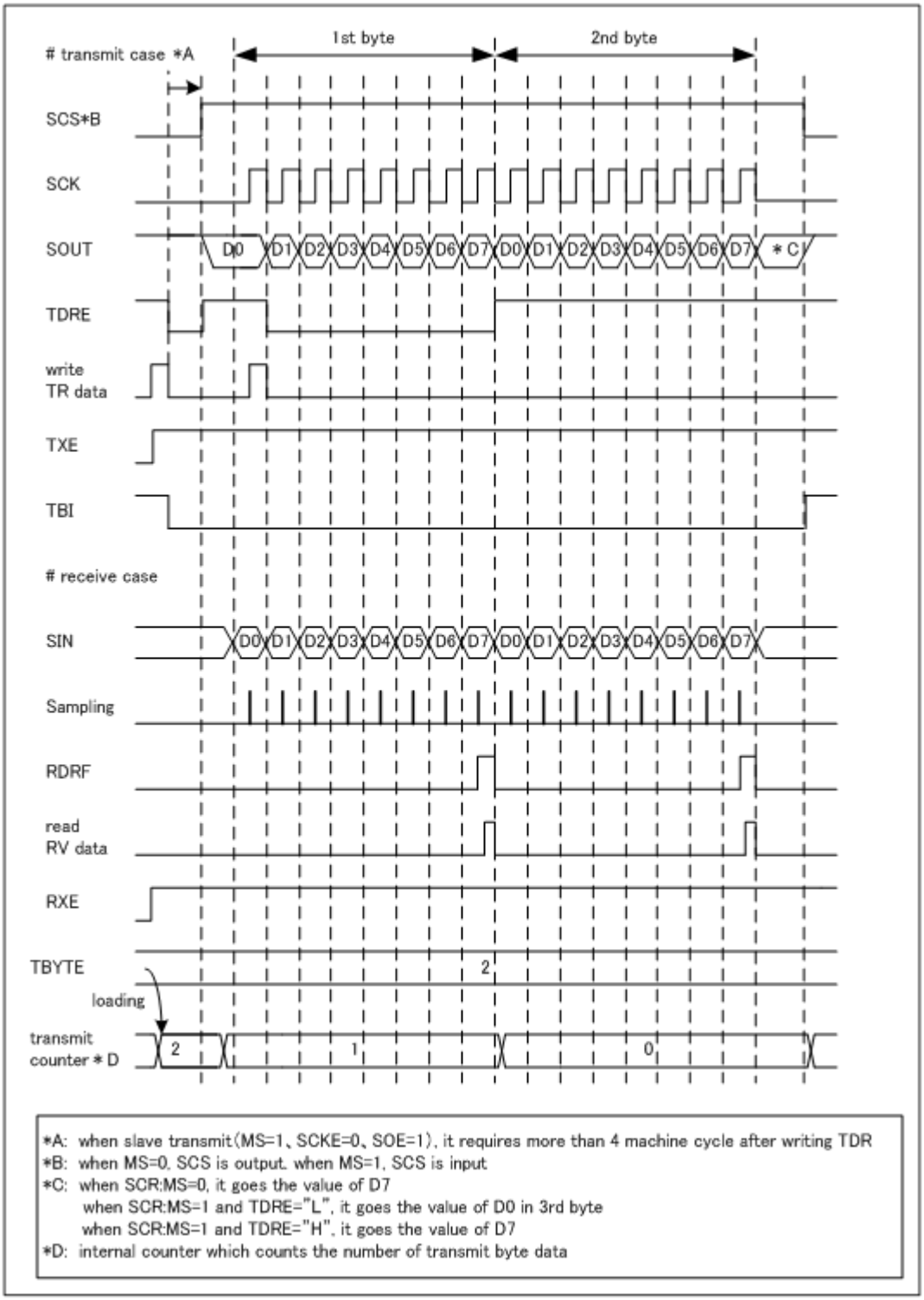
Transmission/Reception Operations

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)



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:

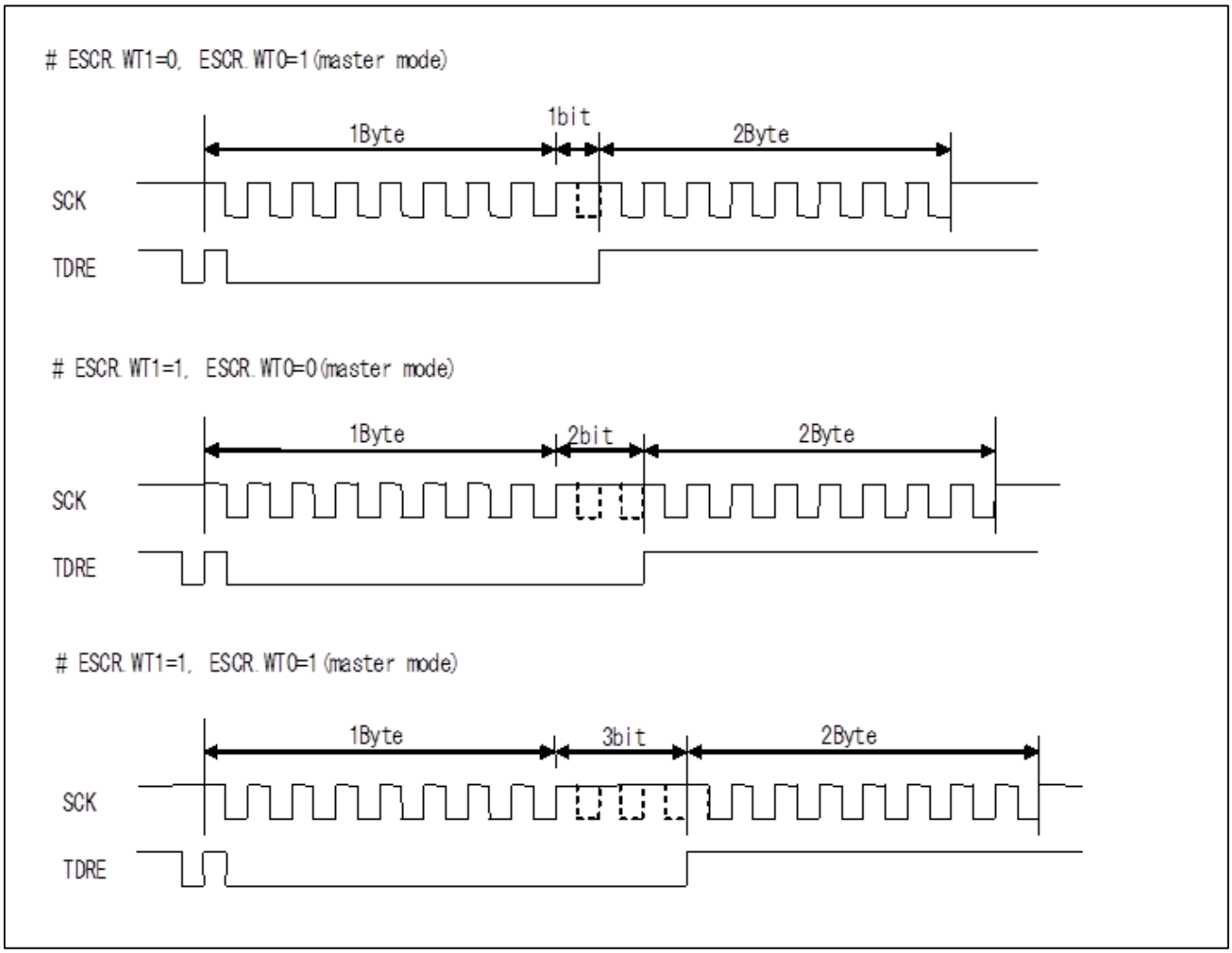
- *When performing only reception, write dummy data to TDR to cause the serial clock (SCK) to be output.*
- *When the transmission/reception FIFOs are enabled, setting the number of frames to transfer in the FBYTE register causes as many serial clock (SCK) pulses as the number of frames to be output.*

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.

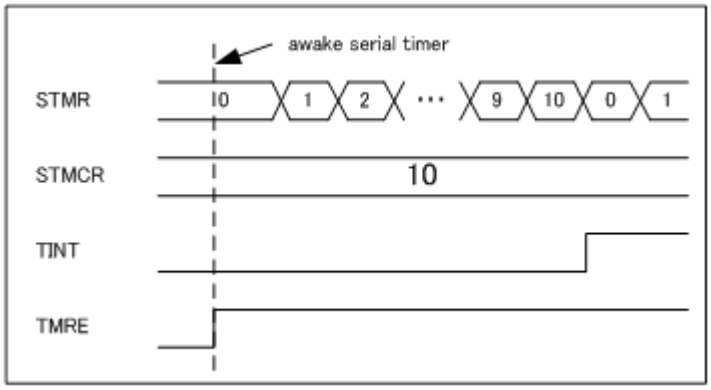
Activating the Serial Timer

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.

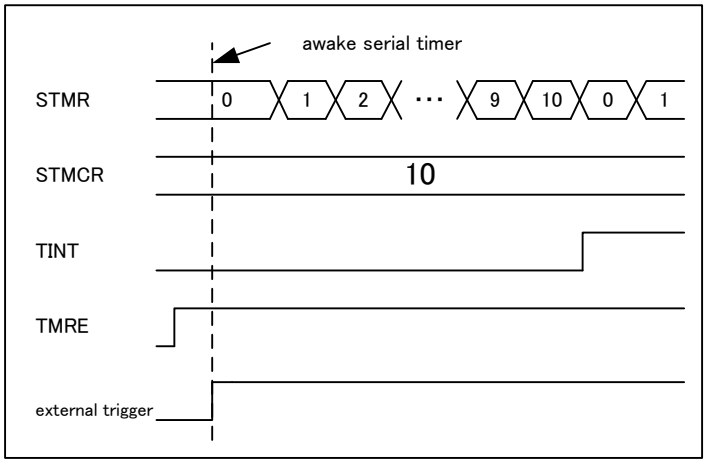
Figure 4-1 Activation by Serial Timer Enable Bit (STMCR = 10, SACSR:TRGE = 0, TSYNE = 0)



- 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)

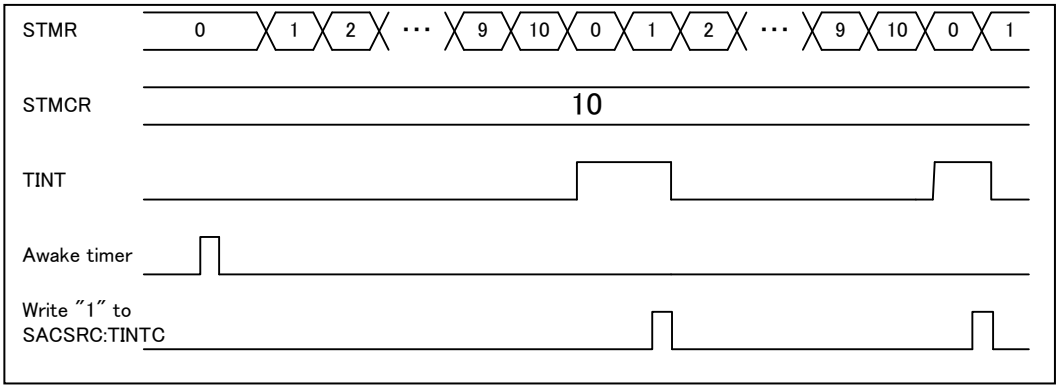


Figure 4-4 Flow Chart of Serial Timer Initialization

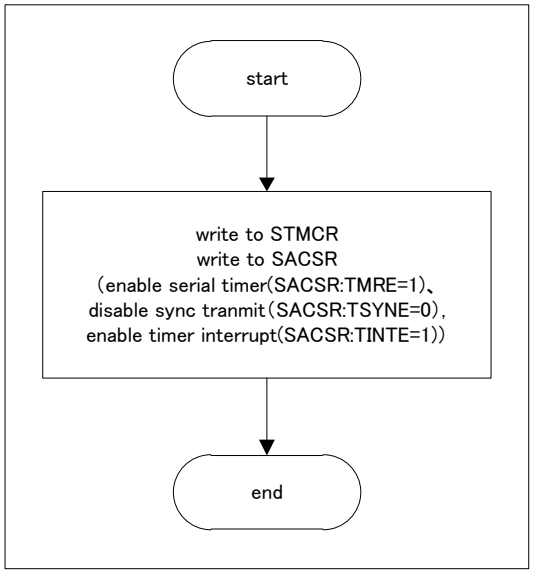
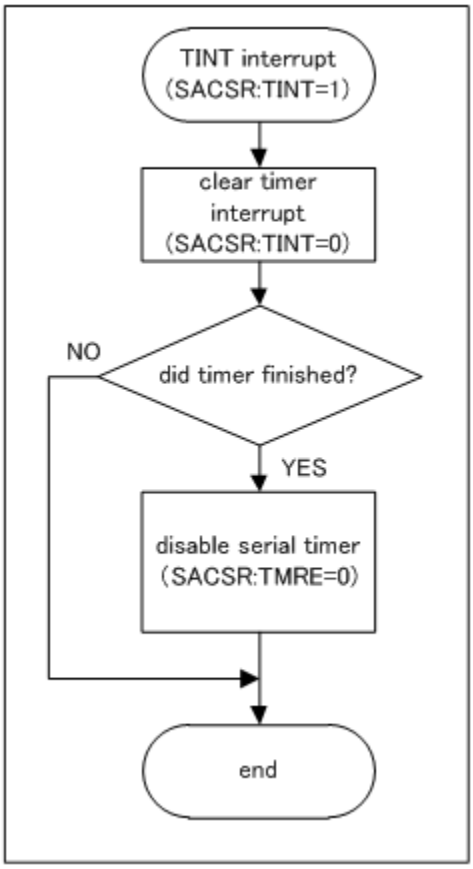


Figure 4-5 Flow Chart of Serial Timer Interrupt Processing



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".*

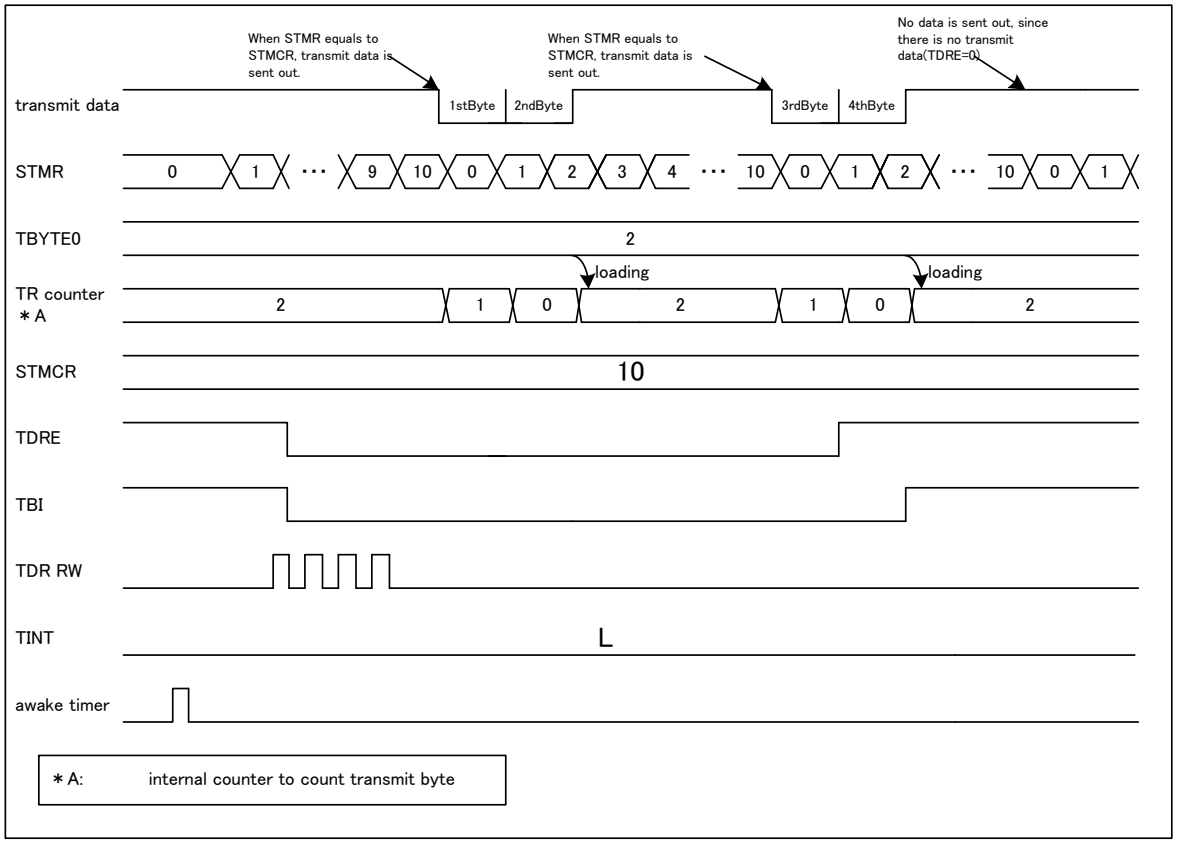
Transmission Operation Synchronized with the Timer

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

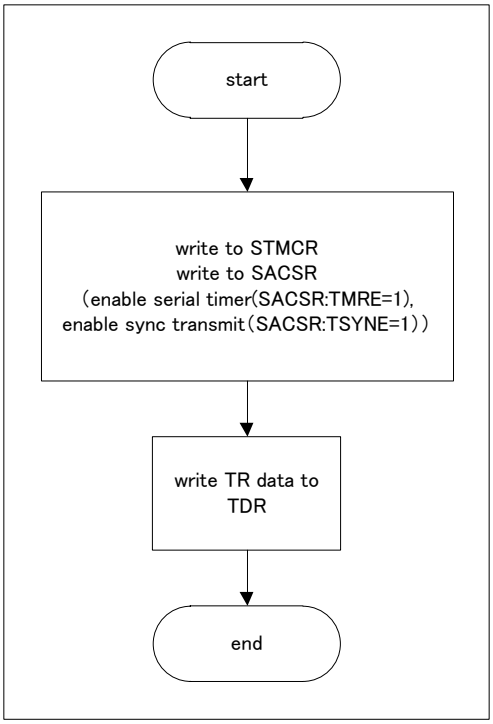
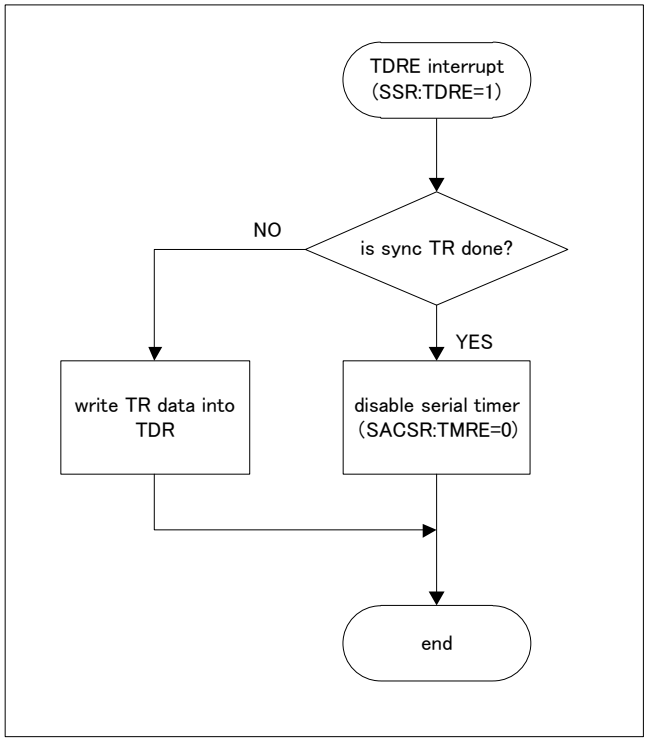


Figure 4-8 Flow Chart of Interrupt Processing in Transmission Synchronized with Timer



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.

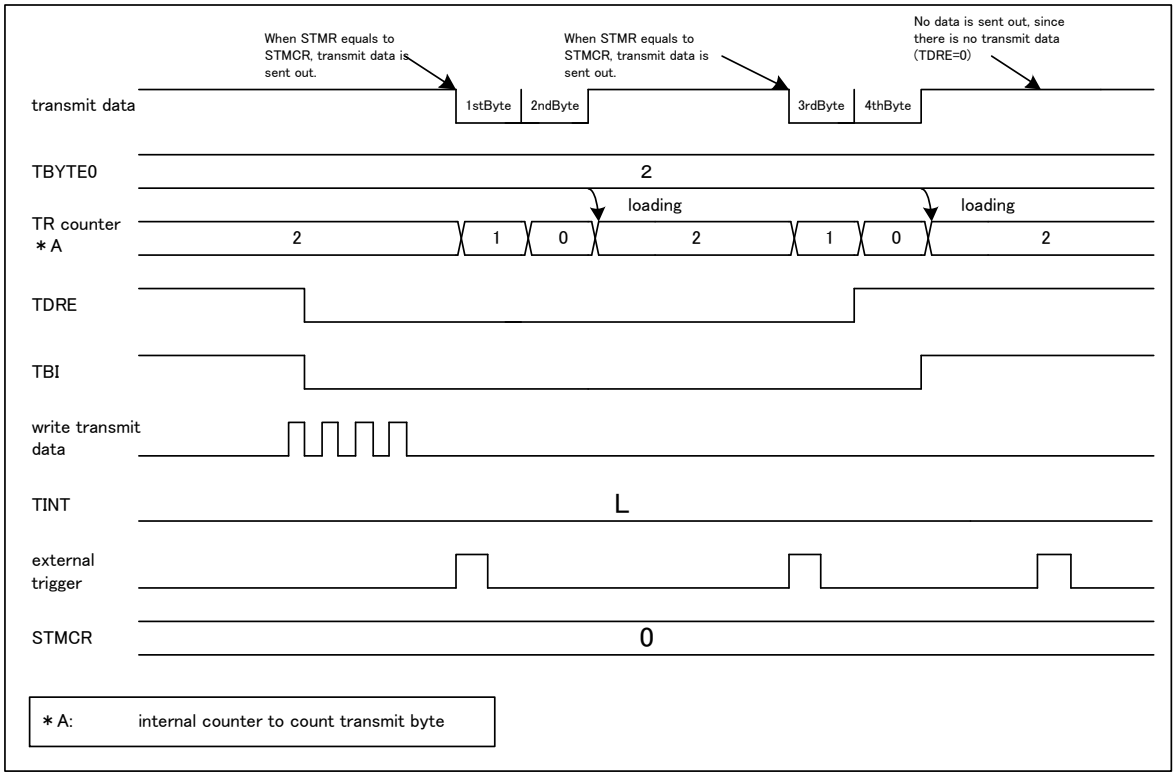
Transmission Operation Synchronized with an External Trigger

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

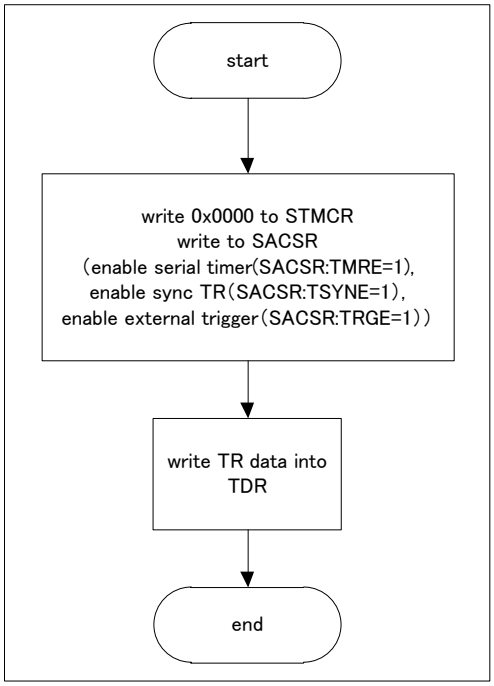
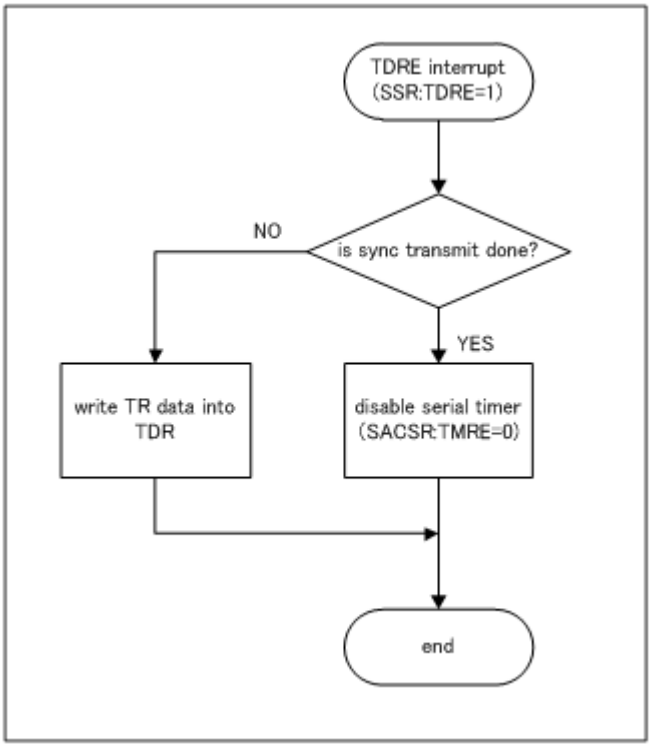


Figure 4-11 Flow Chart of Interrupt Processing in Transmission Synchronized with External Trigger



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)

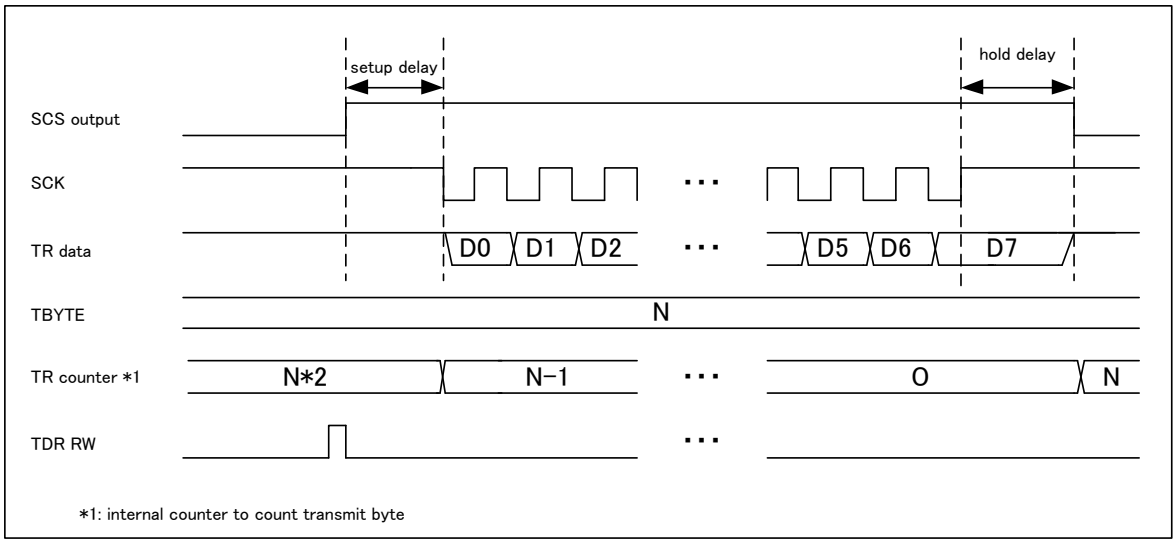
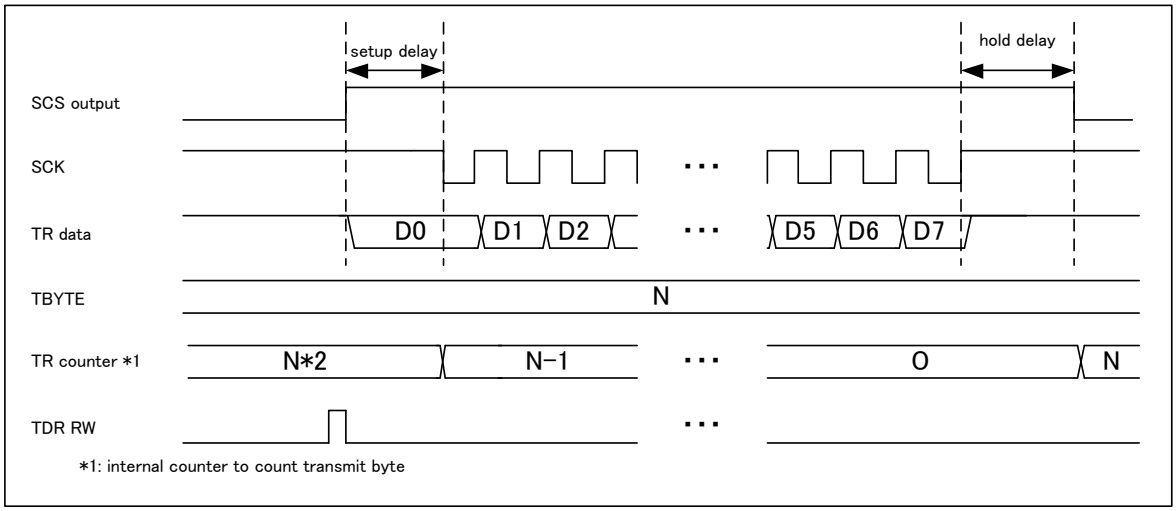


Figure 5-2 Serial Chip Select Operation (Master Transmission (MS = 0), SPI Transfer (SPI = 1), SCINV = 0)



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 CSSU70).

- 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)

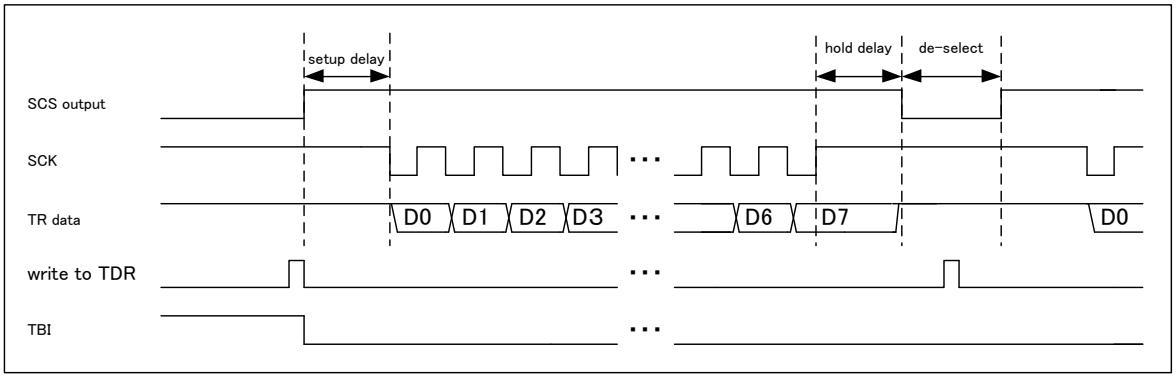
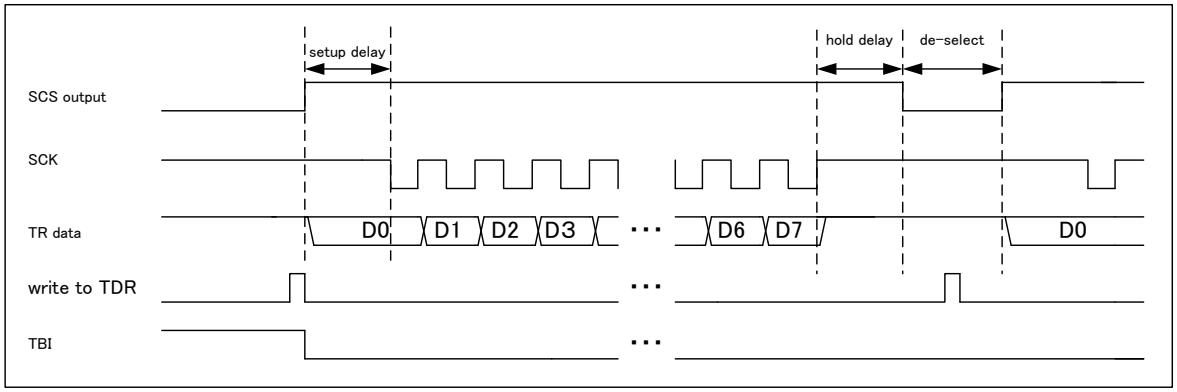


Figure 5-4 Timing Adjustment (Normal Transfer (SPI = 1), SCINV = 0)



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.*

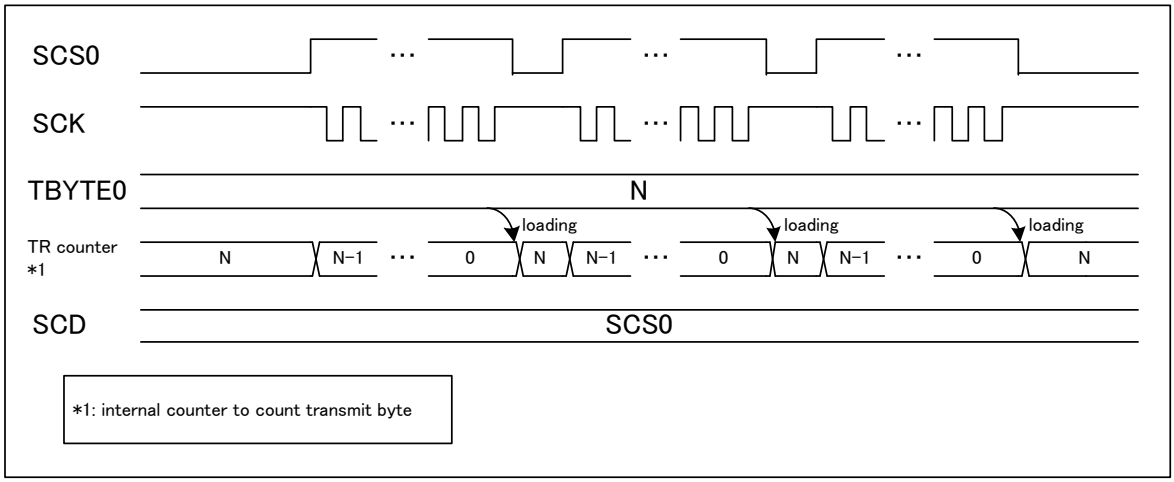
Independent Operation of the Chip Select Pin (Valid Only in Master Mode (SCR:MS = 0))

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)

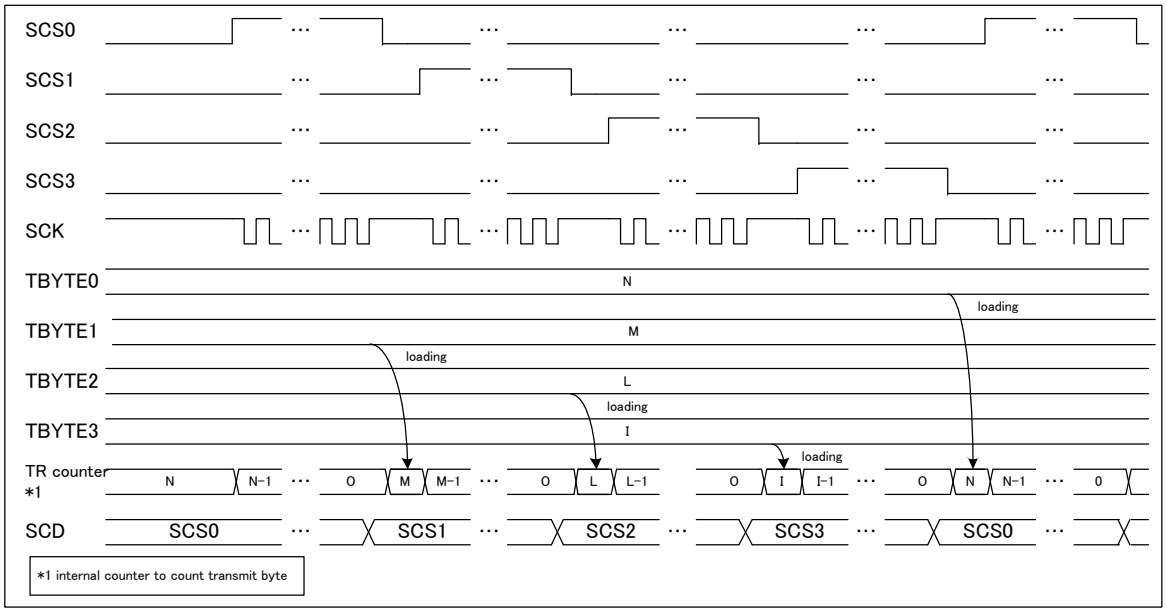


Figure 5-7 shows the timing chart that is assumed if the serial chip select start pin is SCS1 (SST1 and SST0 = 1) and the end pin is SCS2 (SED1 and SED0 = 2).

Figure 5-7 Rounding Operation of Chip Select (SST1-0 = 1, SED1-0 = 2, CSEN3 = 0, CSEN2 = 1, CSEN1 = 1, CSEN0 = 0, SCAM = 0)

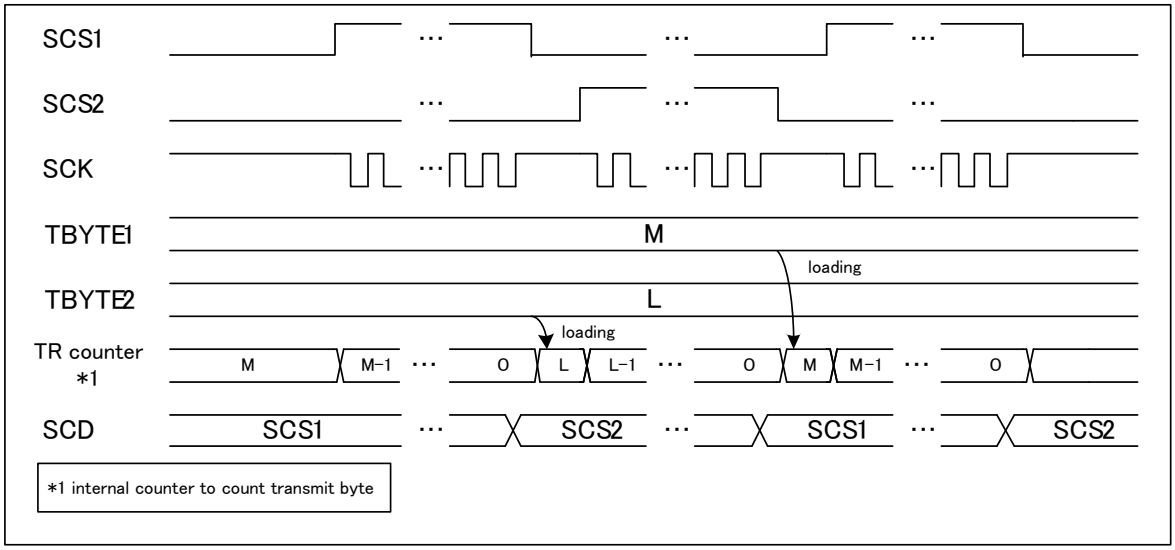
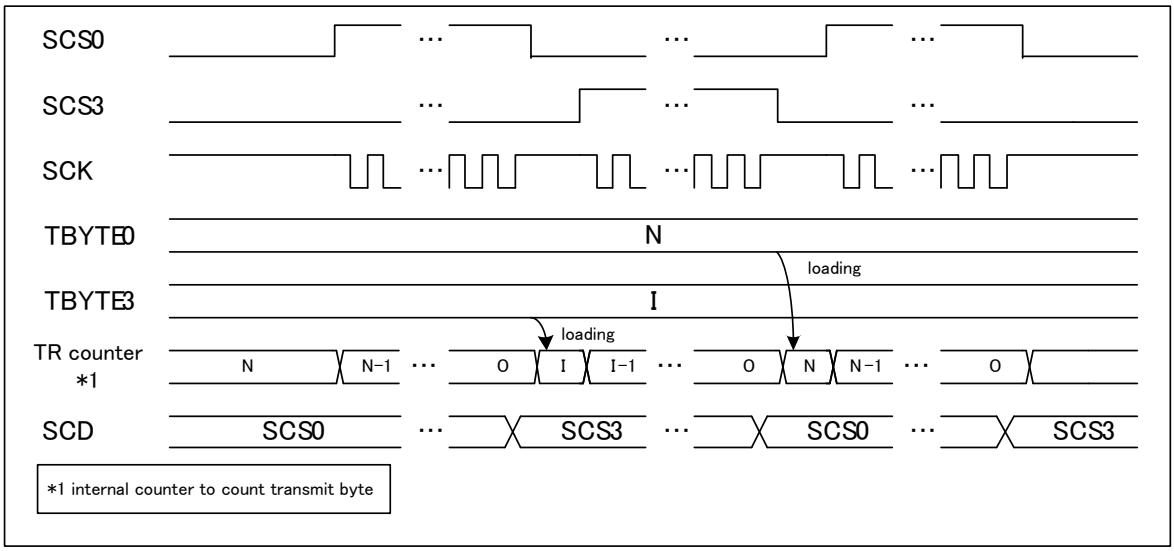


Figure 5-8 shows the timing chart that is assumed if the serial chip select start pin is SCS0 (SST1 and SST0 = 0), the end pin is SCS3 (SED1 and SED0 = 3), and chip select pins 1 and 2 are disabled (CSEN1 and CSEN2 = 0b00). After serial chip select pin 0 becomes active, pins 1 and 2 are skipped, and 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)			

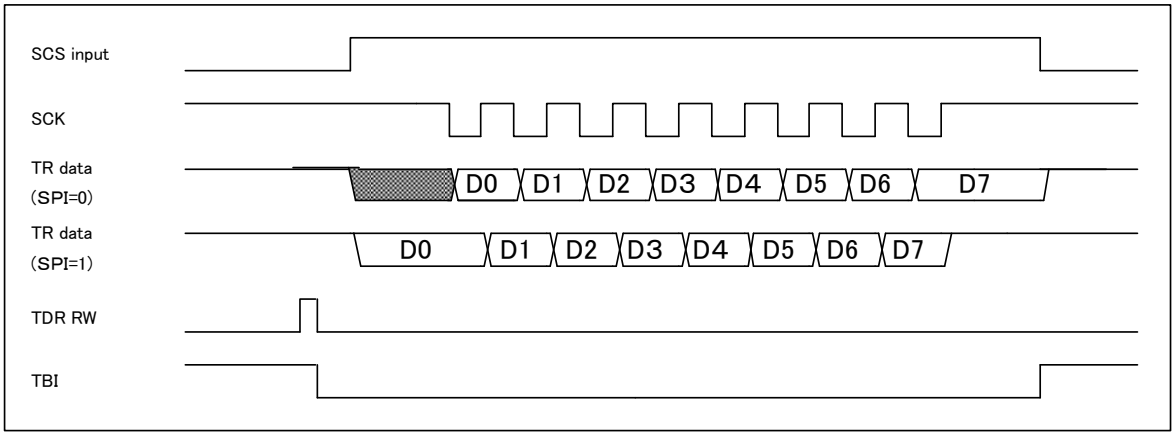
Notes:

- *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

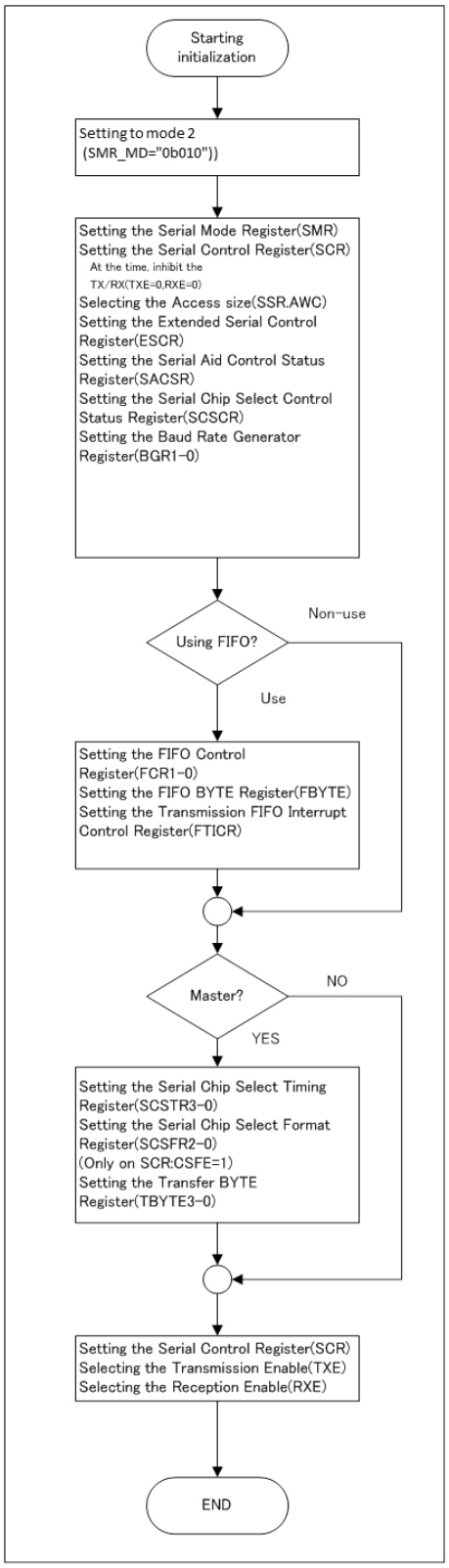
The active level of chip select of each serial chip select pin, mark level of the serial clock, SPI mode enabling and disabling, and data direction and data length of serial data output can be set with the bits listed in Table 5-2.

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:SCLVL	SMR:SCINV	SCR:SPI	SMR:BDS	ESCR:L3 to L0
	Serial chip select pin 1 output	SCSFR0: CS1SCLVL	SCSFR0: CS1SCINV	SCSFR0: CS1SPI	SCSFR0: CS1BDS	SCSFR0: CS1L3 to CS1L0
	Serial chip select pin 2 output	SCSFR1: CS2SCLVL	SCSFR1: CS2SCINV	SCSFR1: CS2SPI	SCSFR1: CS2BDS	SCSFR1: CS2L3 to CS2L0
	Serial chip select pin 3 output	SCSFR2: CS3SCLVL	SCSFR2: CS3SCINV	SCSFR2: CS3SPI	SCSFR2: CS3BDS	SCSFR2: CS3L3 to CS3L0
Chip select format disable (SCR:CSFE = 0)		SCSCR0:SCLVL	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

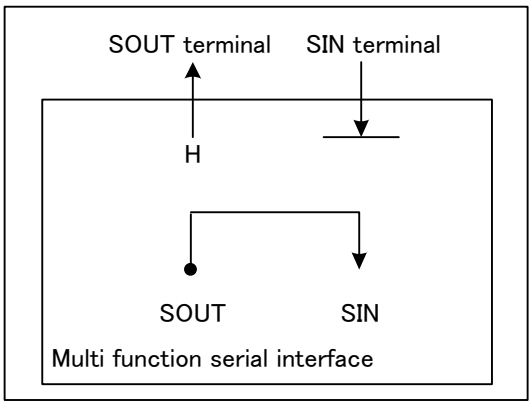
This section describes operation in test mode.

Serial Test Mode

When serial test mode is enabled (SACSR:STST = 1), SOUT and SIN are connected together inside the multifunction 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 1s 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
```

(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	40 MHz		48 MHz		72 MHz		80 MHz	
(bps)	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)

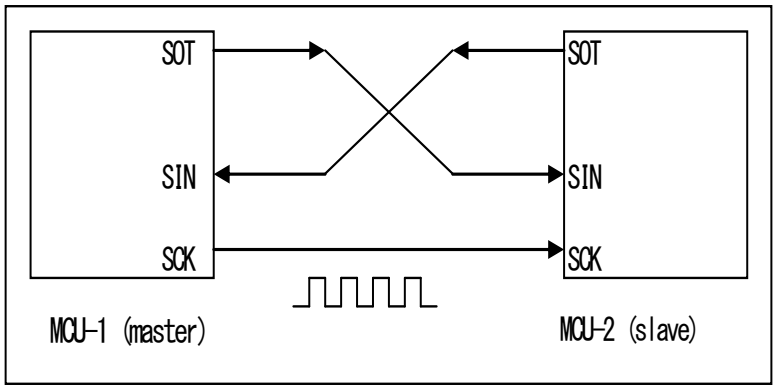
7.2. Setup Procedure and Program Flow for the CSIO (Clock Synchronous Serial Interface)

The CSIO (Clock Synchronous Serial Interface) can perform serial bidirectional transmission.

Connection between MCUs

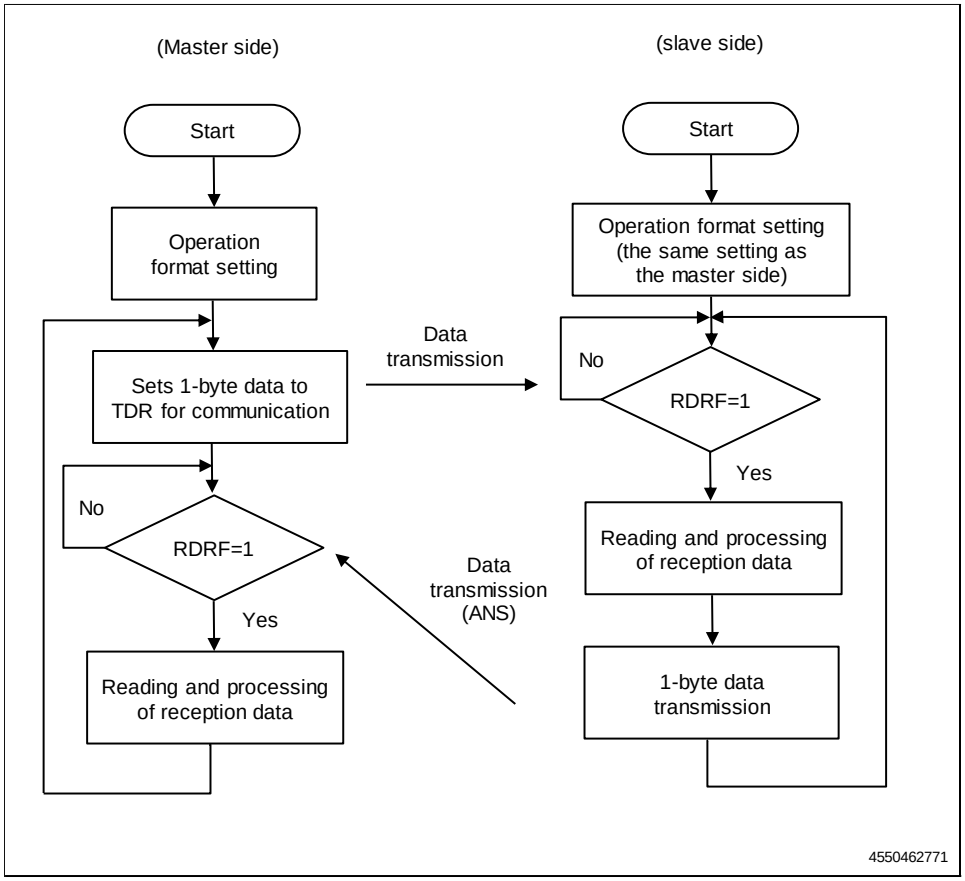
For the CSIO (Clock Synchronous Serial Interface), select bidirectional communication. As shown in Figure 7-1, 2 MCUs are interconnected.

Figure 7-1 Connection Example for CSIO (Clock Synchronous Serial Interface) Bidirectional Communication



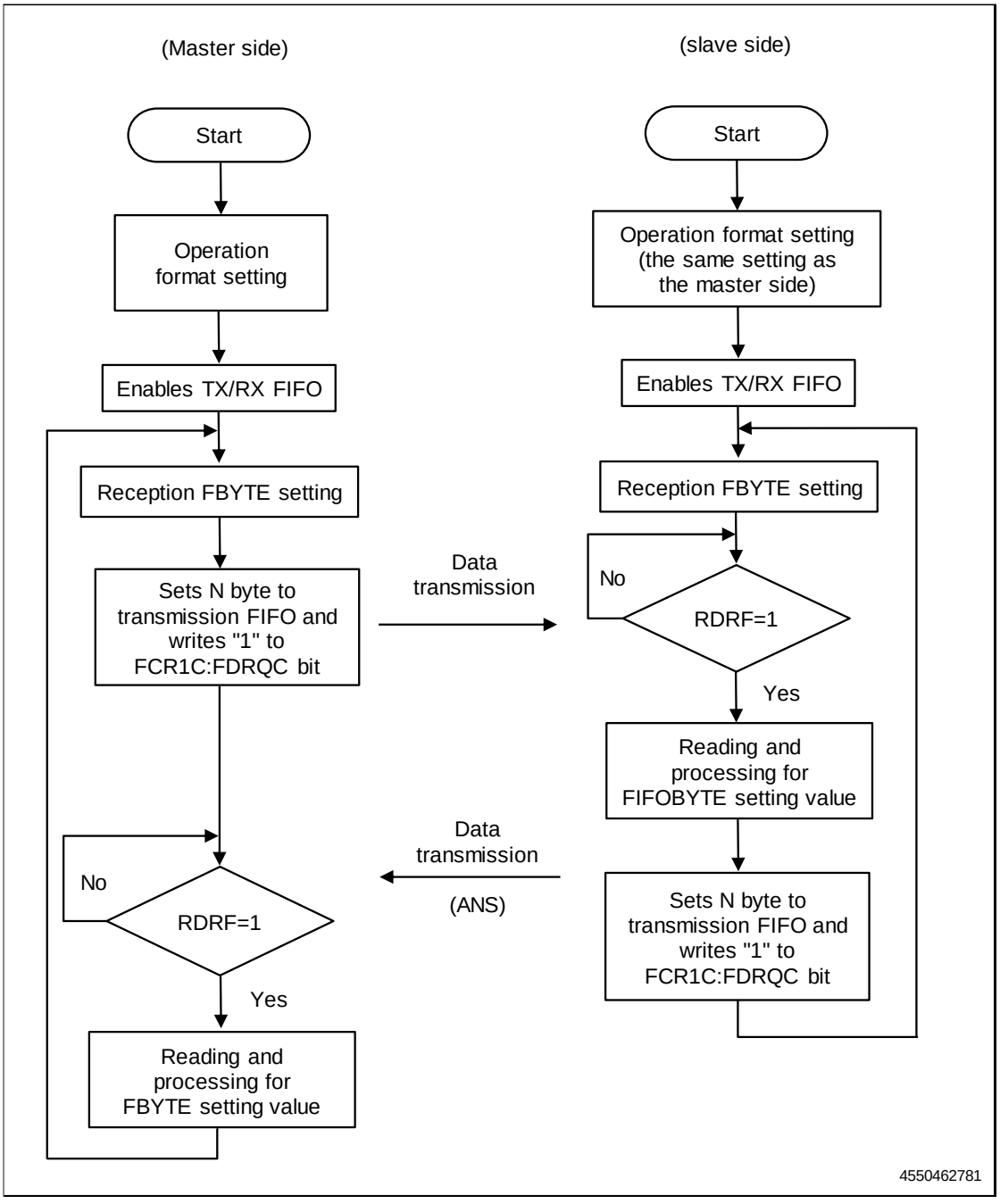
- FIFO not used

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 acknowledgment standard (ACK mode) and 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 reception block size register.

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.

Note:

- *ACK Mode of Block Transfer is not supported for this device.*

8.1. ACK Mode

This section describes block transfer in ACK mode.

Reception Block Transfer

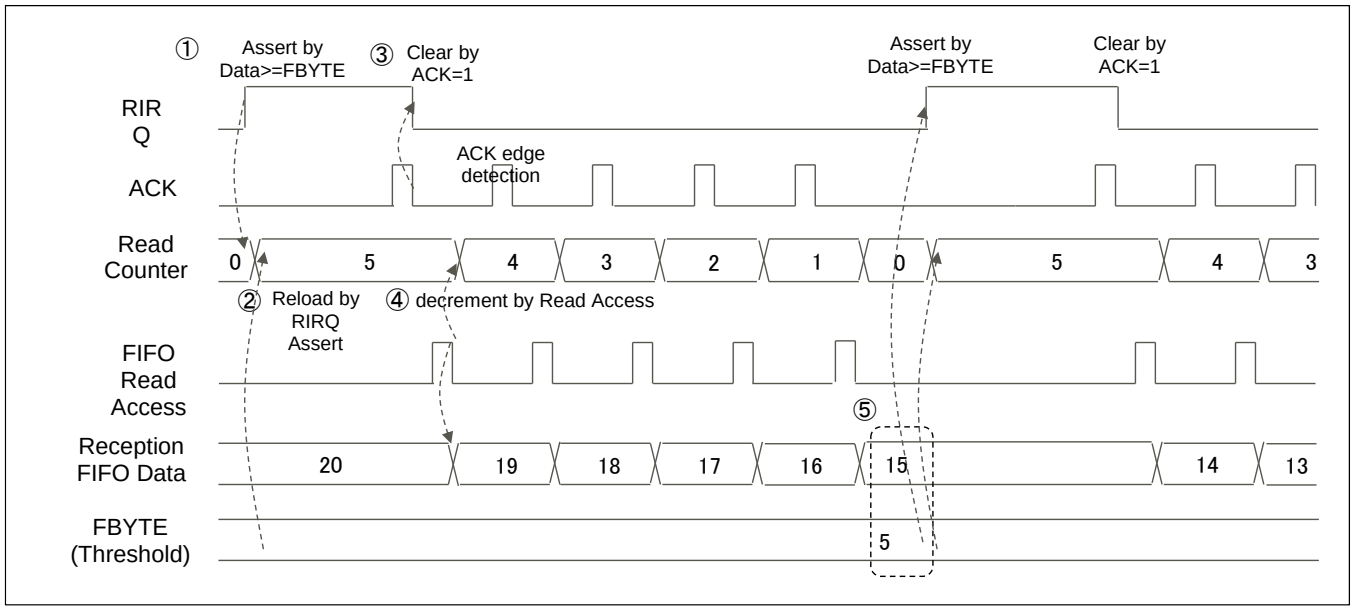
Reception block transfer is performed as described below.

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 reception acknowledgment is input, the reception block transfer request (RIRQ) is cleared. With the read of 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.

Figure 8-1 Reception Block Transfer in ACK Mode



Transmission Block Transfer

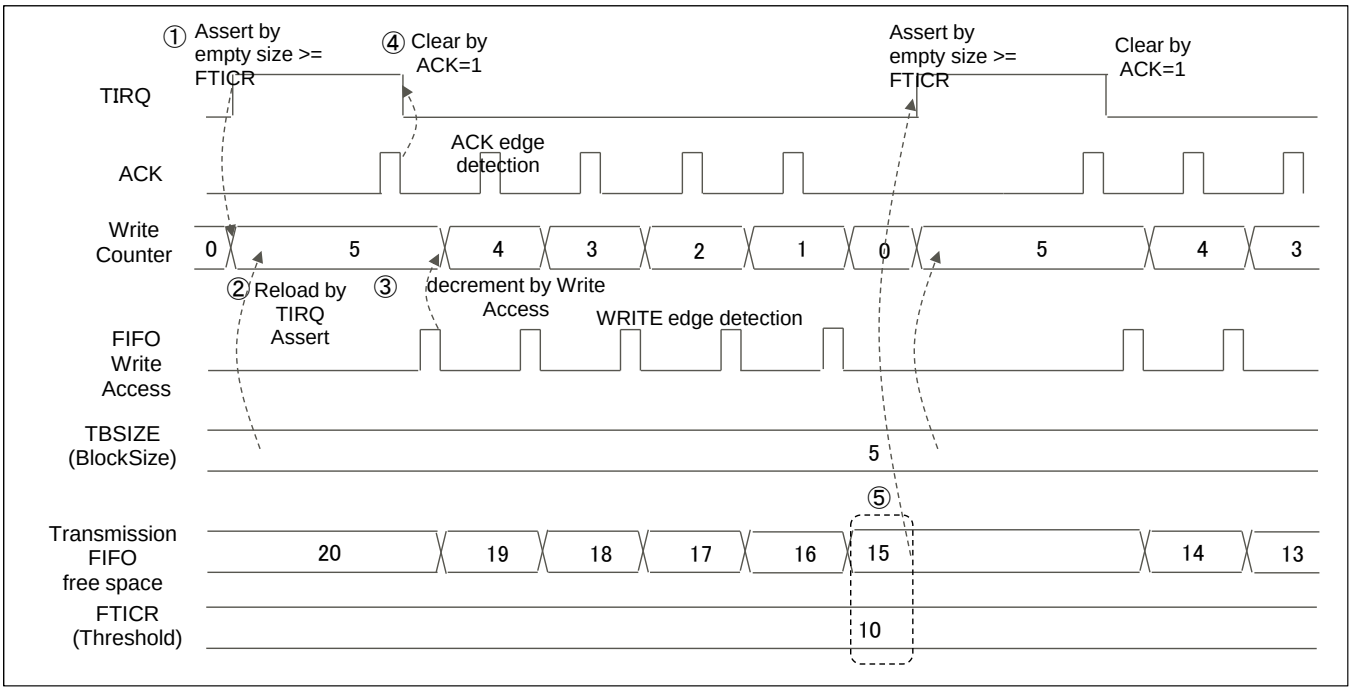
Transmission block transfer is performed as described below.

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 transmission acknowledgment is input, the transmission block transfer request (TIRQ) is cleared. With the write of the 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.

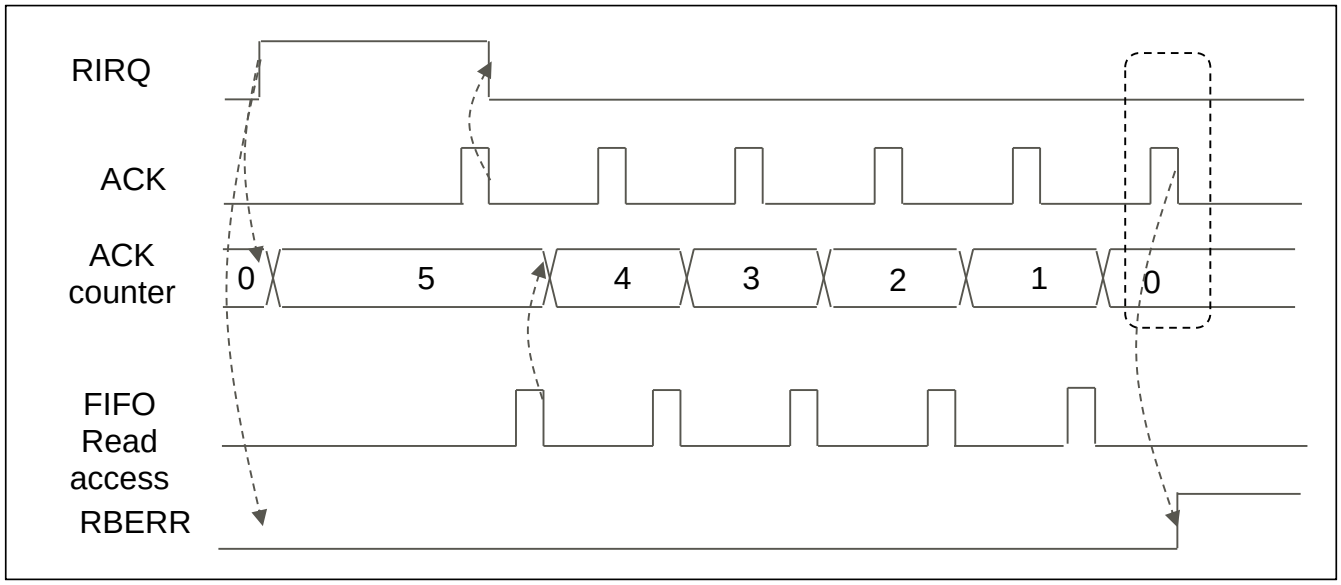
Figure 8-2 Transmission Block Transfer in ACK Mode



Reception Block Transfer Error

If, while the Read counter is "0", an acknowledgment is input before the next reception block transfer request (RIRQ) is asserted, a reception block transfer error occurs, and the ESR:RBERR bit is asserted.

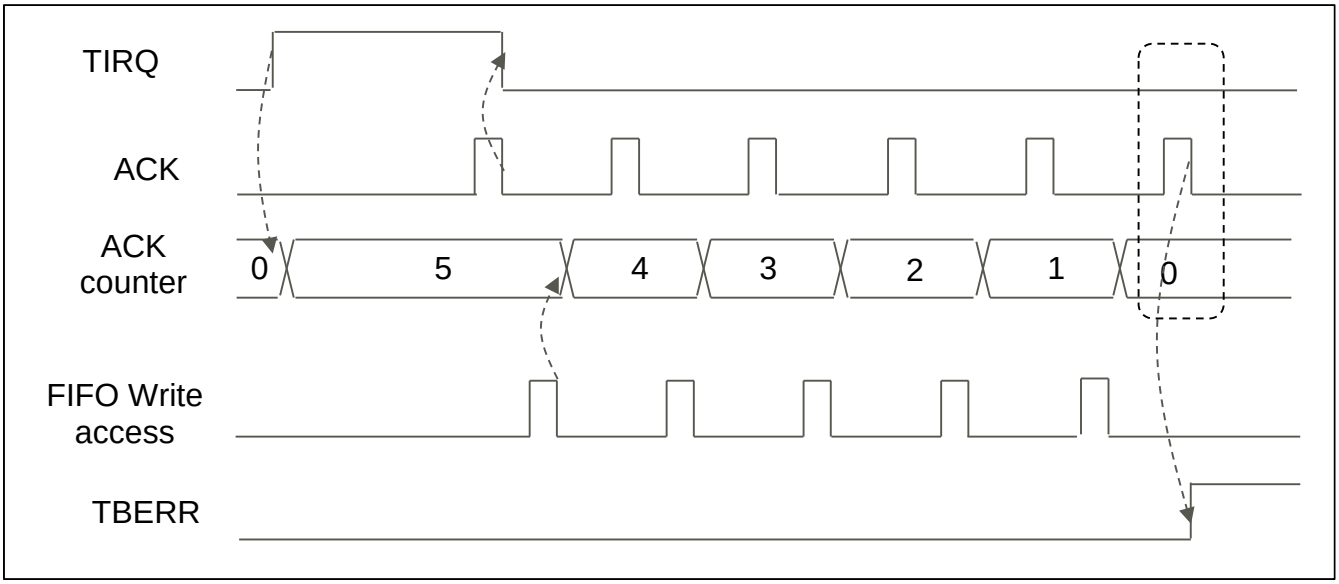
Figure 8-3 Reception Block Transfer Error in ACK Mode



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.

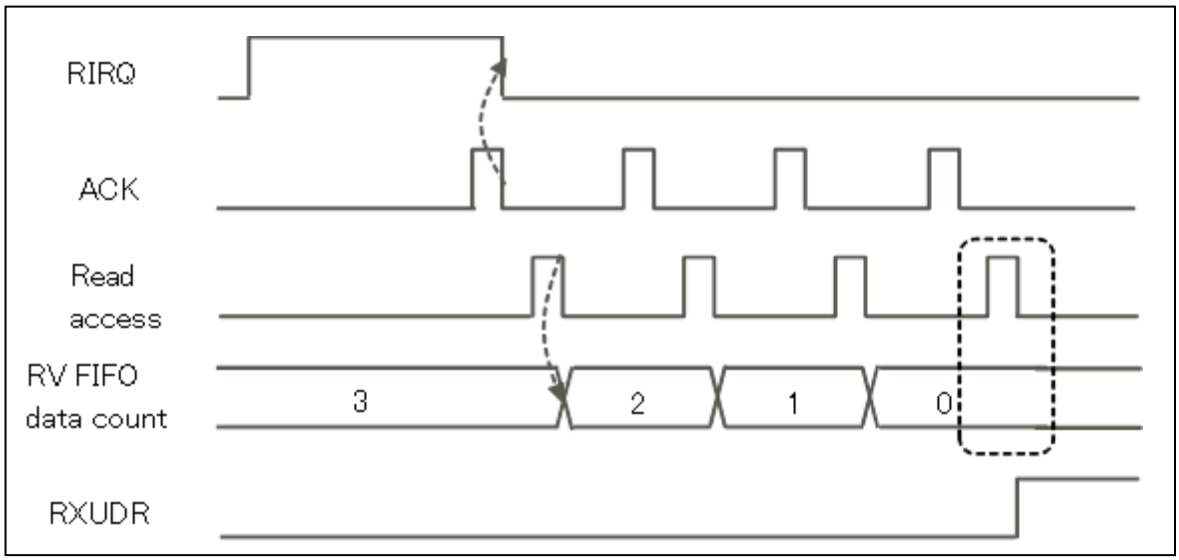
Figure 8-4 Transmission Block Transfer Error in ACK Mode



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.

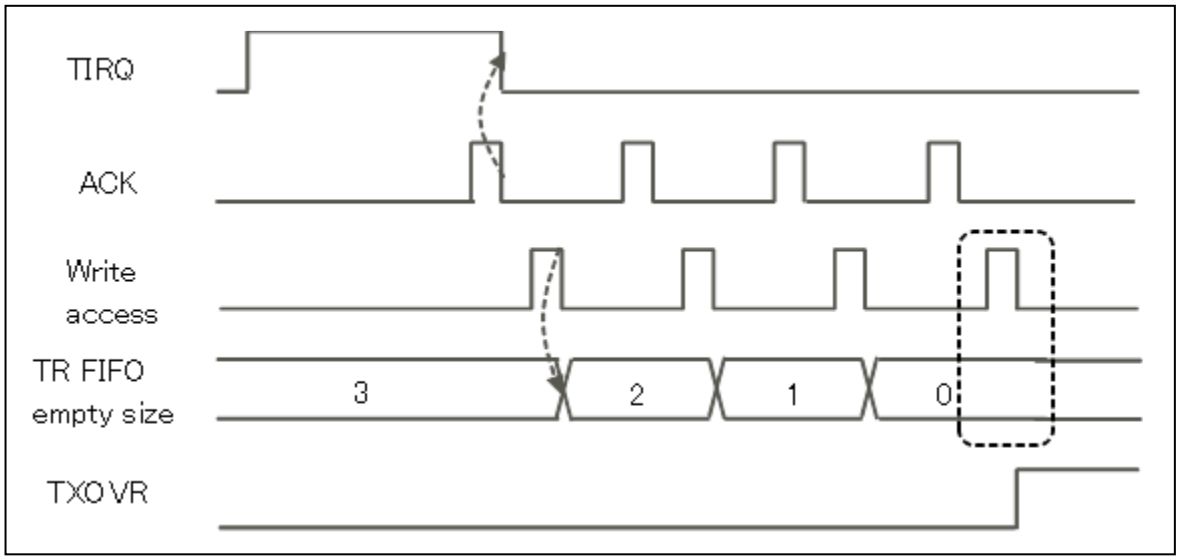
Figure 8-5 Reception FIFO Under Run in ACK Mode



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 ACK Mode



8.2. RW Access Mode

This section describes block transfer in RW access mode.

Reception Block Transfer

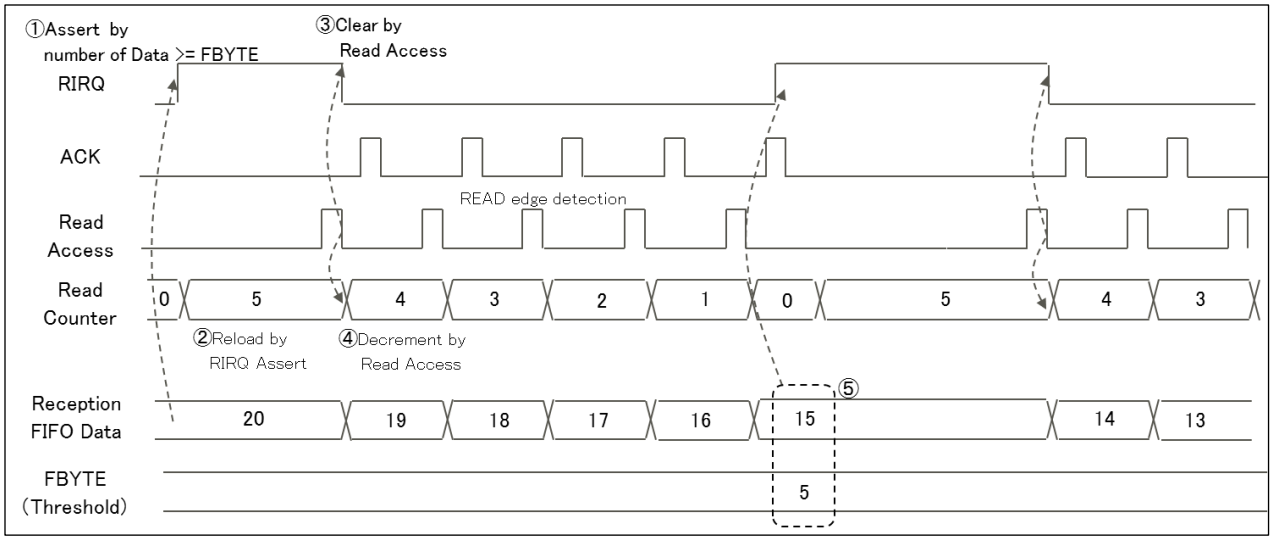
Reception block transfer is performed as described below.

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 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.

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.

Figure 8-7 Reception Block Transfer in RW Access Mode



Transmission Block Transfer

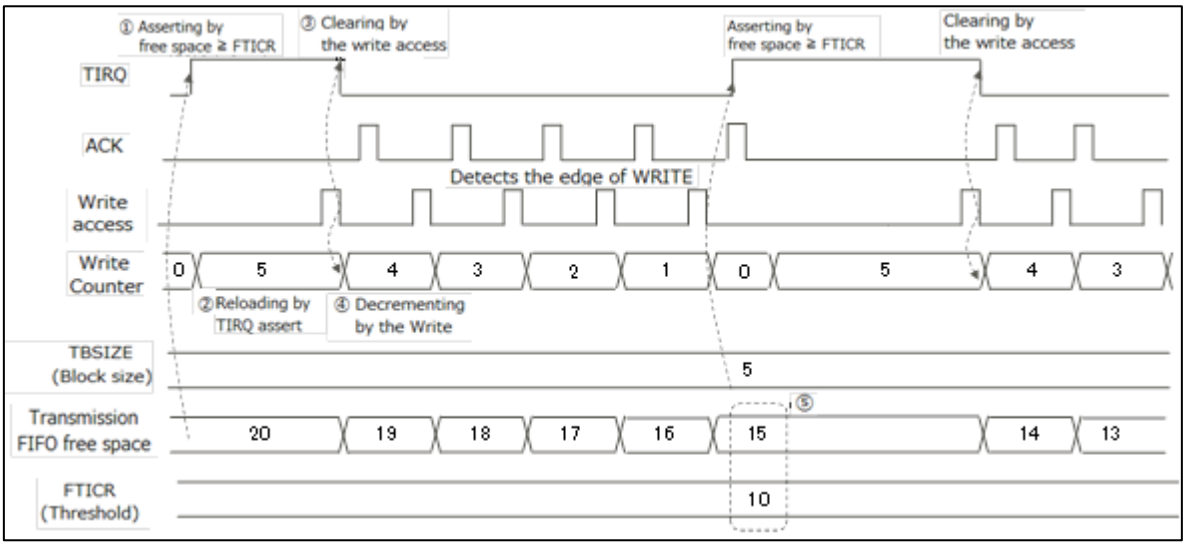
Transmission block transfer is performed as described below.

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 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.

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.

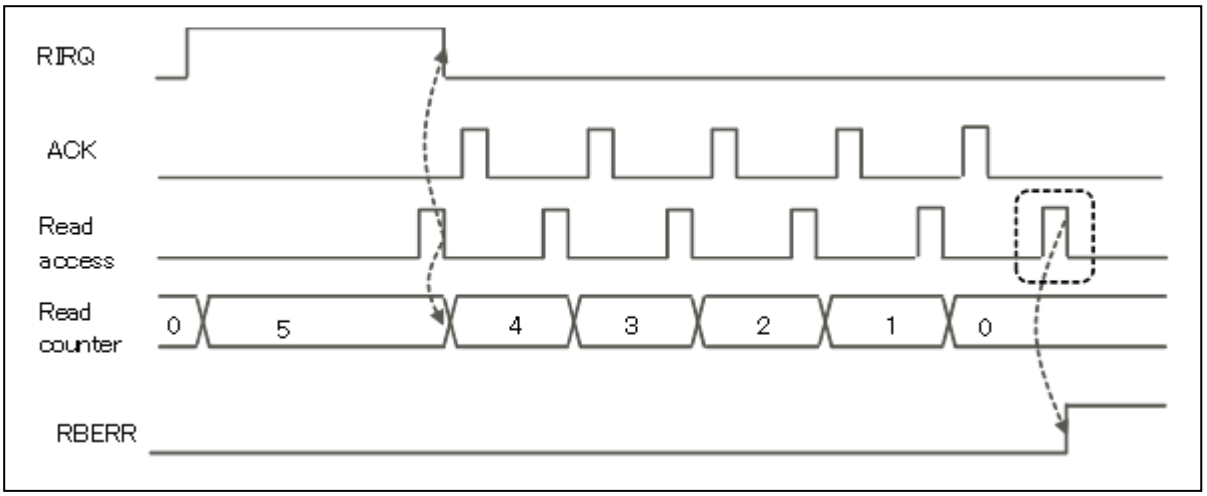
Figure 8-8 Transmission Block Transfer in RW Access Mode



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.

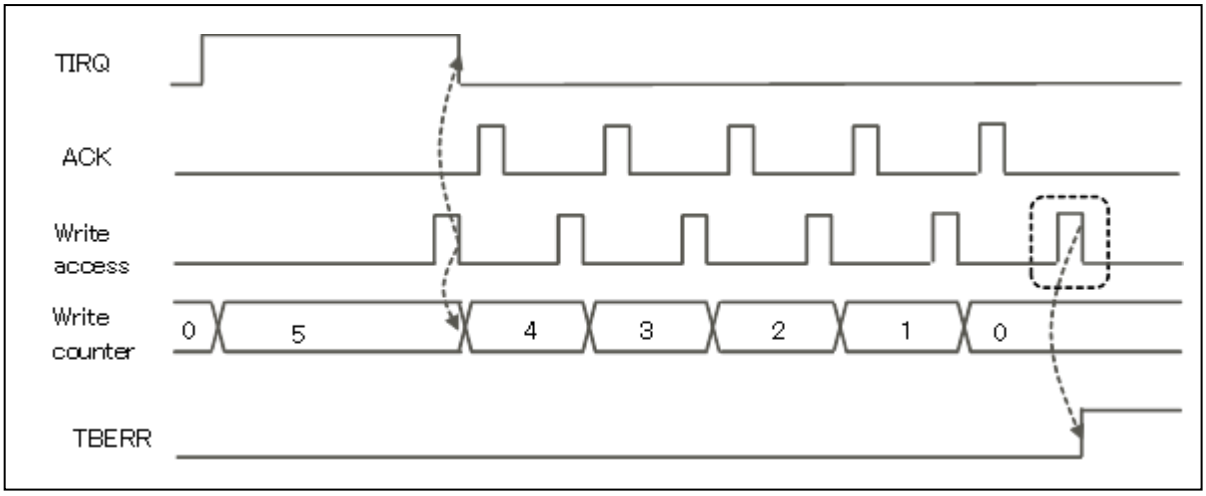
Figure 8-9 Reception Block Transfer Error in RW Access Mode



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 FSR:TBERR bit is asserted.

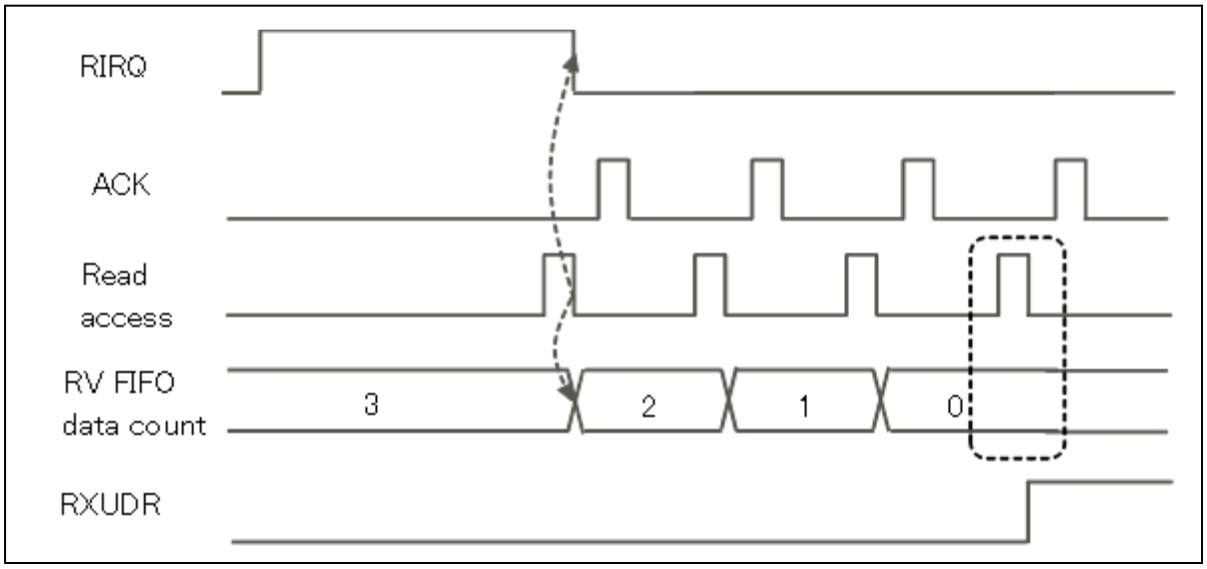
Figure 8-10 Transmission Block Transfer Error in RW Access Mode



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.

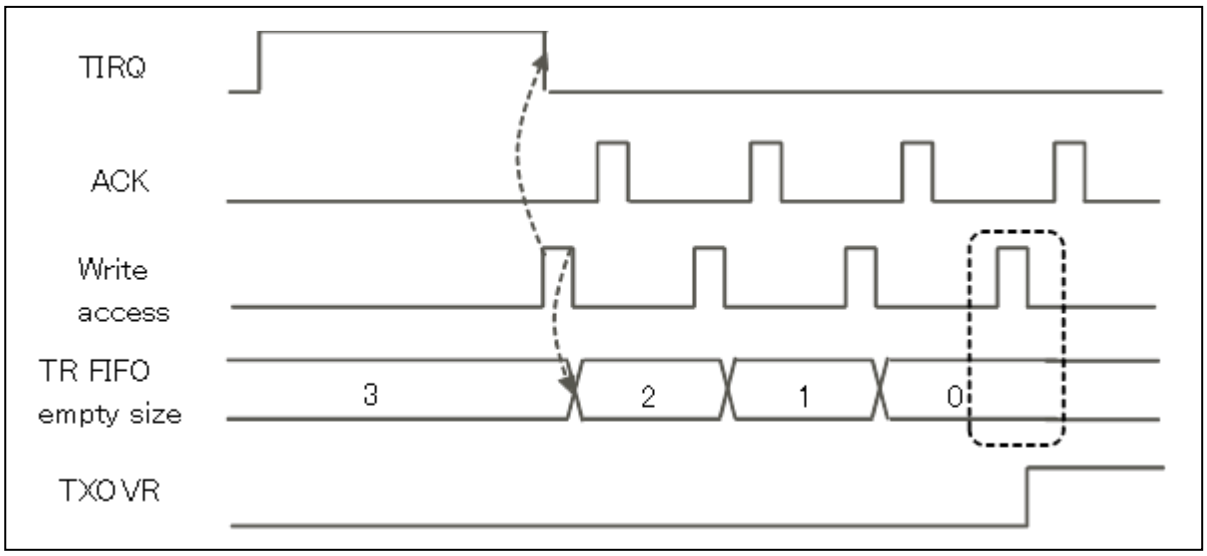
Figure 8-11 Reception FIFO Under Run in RW Access Mode



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 8-12 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)
	-	-

	bit15 bit8	bit7 bit0
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 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)
	-	-

	bit15 bit8	bit7 bit0
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
	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 both "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".

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.

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).

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

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.

[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.*
- *To use the SCK pin as a serial clock input (SCKE = 0), set the SCK pin as a peripheral input pin (set with PPC_PCFGRijj and RIC_RESINn).*
- *When you use a SCK pin as a serial clock output, set the SCK pin as a peripheral output pin (set with PPC_PCFGRijj). See "CHAPTER:I/O PORT" for how to make setups.*

[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:

- *Set a SOUT pin as a peripheral output pin (set with PPC_PCFGRijj). See "CHAPTER: I/O PORT" for how to make setups.*

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.*

[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

[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).*

[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".
- If the UPCL bit of the serial control register (SCR) is set to "1", the TDRE bit is set to "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".
- If the UPCL bit of the serial control register (SCR) is set to "1", the TDRE bit is set to "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 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.

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 serial 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 (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.*

[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 (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.*

[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:CDIV = 0b000.*

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, BDS = 0) or transfer of the highest bit first (MSB first, BDS = 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, BDS = 0) or transfer of the highest bit first (MSB first, BDS = 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 Lengtht
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, BDS = 0) or transfer of the highest bit first (MSB first, BDS = 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

- Set values in baud rate generator registers 1, 0 (BGR1, BGR0).
- 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 FLST bit 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.

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
  - FTICR setting value ≥ FTICR read value (The amount of data in the transmission FIFO is at the interrupt trigger level or lower.)
  - Reset of the transmission FIFO

FDRQ reset 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	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.*

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

- FIFO reset (writing "1" to FCL)
- 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.*

[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.*

[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 "0" to the FCR1:FDRQ 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, except when receiving 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.*
- *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.*
- *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.*

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 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.

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.*

[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.*

[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 (SSR: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", 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".*

[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. Transmission-side DMA transfer stops. At that time, a transmission interrupt occurs if transmission interrupt is enabled (SSR: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".*
- *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".*
- *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.

[bit9] 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 36: 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. 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.

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. LIN Interface (V2.1) Registers

Note:

- *ACK Mode of Block Transfer is not supported for this device.*

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 FIFO is not used) − Transmission and reception FIFOs (64 bytes each) (when FIFO is used)
2	Serial Input	Over sampling is performed by the bus clock 3 times. The reception value is determined by the rule of majority.
3	Transfer Mode	Asynchronous
4	Baud Rate	− Dedicated baud rate generator is provided (consisting of 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 with 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 underrun) − Transmission interrupt (transmission data empty, transmission bus idle, and transmission block transfer error) − Status interrupts (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 threshold 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 available. − A division value can be selected for the operation clock (1 to 256 divisions). − Activation by an 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 (can be set to 13 to 16 bits) − LIN Break delimiter generation (can be set to 1 to 4 bits) − LIN Break Field detection − Detection of the start/stop edge of the LIN Sync Field connected to an input capture

	Function	Description
12	FIFO Option	− Transmission and reception FIFOs are provided (64 bytes for transmission FIFO: 64 bytes for 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 FIFO is not used) − Transmission and reception FIFOs (64 bytes each)(when a FIFO is used)
2	Serial Input	Over sampling is performed by the bus clock 3 times. The reception value is determined by the rule of majority.
3	Transfer Mode	Asynchronous
4	Baud Rate	− Dedicated baud rate generator is provided (consisting of 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 with falling edges of the start bit
8	Reception Error Detection	− - An error is detected by a self-check on the transmission side - ➢ LIN bus error − - An error is detected by a self-check on the transmission side or is detected at the reception side - ➢ Framing error ➢ Overrun error ➢ LIN ID parity error ➢ LIN checksum error − - An error is detected on the reception side when auto baud rate adjustment is disabled - ➢ LIN Sync Data error
9	Interrupt Request	− Transmission interrupts (1) Data transmission interrupt (transmission data empty, transmission bus idle, and transmission block transfer error) ➢ (2) Transmission FIFO interrupt (when the transmission FIFO is not higher than the interrupt threshold or when the transmission FIFO is empty , transmission FIFO overrun) − Reception interrupt (1) Data reception interrupt (reception completion and reception block transfer error) (2) Reception FIFO interrupt (when reception FIFO is equal to or higher than the interrupt threshold, reception FIFO underrun) (3) 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 (1) Auto header completion interrupt (2) Sync Field detection interrupt (3) Checksum calculation completion interrupt − DMA Transfer is supported for both transmission and reception.

	Function	Description
10	Timer Function	− A 16-bit serial timer is provided. − A division value can be selected for the operation clock (1 to 256 divisions). − Activation by an 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 (1) LIN Break Field generation (can be set to 13 to 20 bits) (2) LIN Break delimiter generation (can be set to 1 to 4 bits) (3) Automatic generation of Sync Field and automatic check of data value (0x55) (4) Automatic parity generation/check in the ID Field (5) Automatic generation/check of a checksum (support for standard/extended checksum)
12	FIFO Option	− Transmission and reception FIFOs are provided (64 bytes for transmission FIFO: 64 bytes for 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 Data 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)
			Reception of as much data as the amount set in FBYTE		Reading of reception data until the reception FIFO becomes empty
			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)
	FRE	SSR	Framing error		
	RXUDR	ESR	Reception FIFO underrun	ECR:REIE	Writing "1" to the reception FIFO underrun clear flag bit (ESRC:RXUDRC)
	RBERR	ESR	Reception block transfer error	ECR:REIE	Writing "1" to the reception block transfer error clear bit (ESRC:RBERRC)

Interrupt Type	Interrupt Request Flag Bit	Flag Register	Interrupt Factor	Interrupt Factor Enable Bit	Clearing of Interrupt Request Flag
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
	TXOVR	ESR	Transmission FIFO overrun	ECR:TEIE	Writing "1" to the transmission FIFO overrun flag clear bit (ESRC:TXOVRC)
	TBERR	ESR	Transmission block transfer error	ECR:TEIE	Writing "1" to the transmission block transfer error clear bit (FSRC:TBERRC)
Status (Manual Mode)	LBD	SSR	LIN Break Field detection	ESCR:LBIE	Writing "0" to the SSR:LBD bit Writing "1" to the SSRC:LBDC bit
	SFD	SACSR	Sync Field detection	SACSR:SFDE	Writing "1" to the Sync Field detection flag clear bit (SACSRC:SFDC)
	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)
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

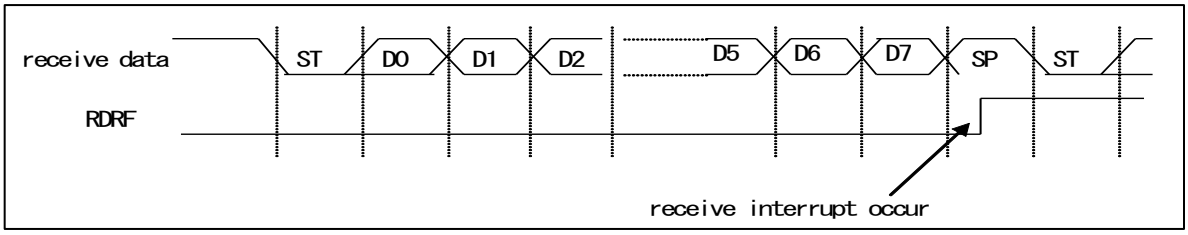
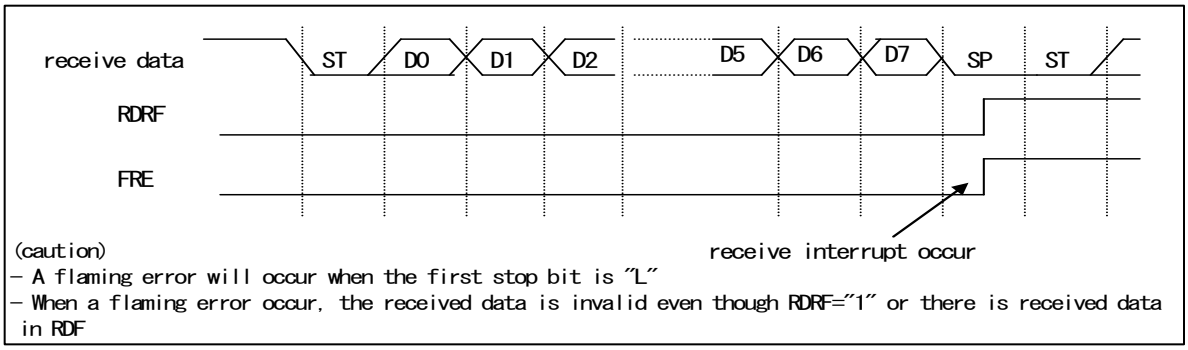


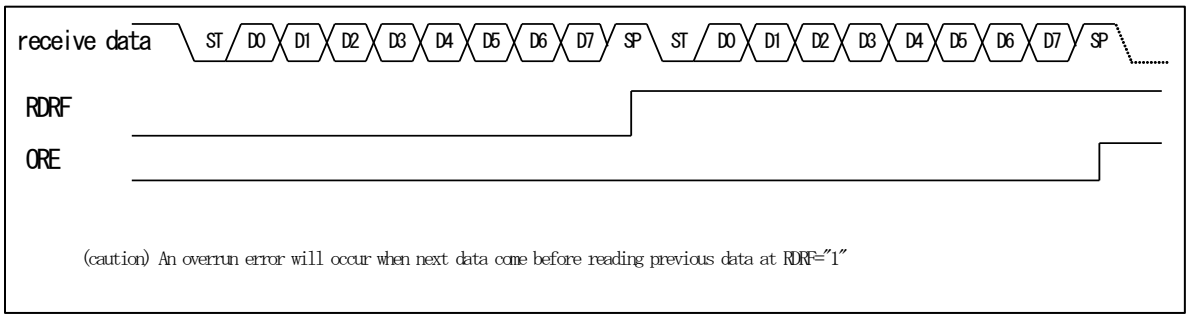
Figure 2-2 Set Timing of FRE (Framing Error) Flag Bit



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

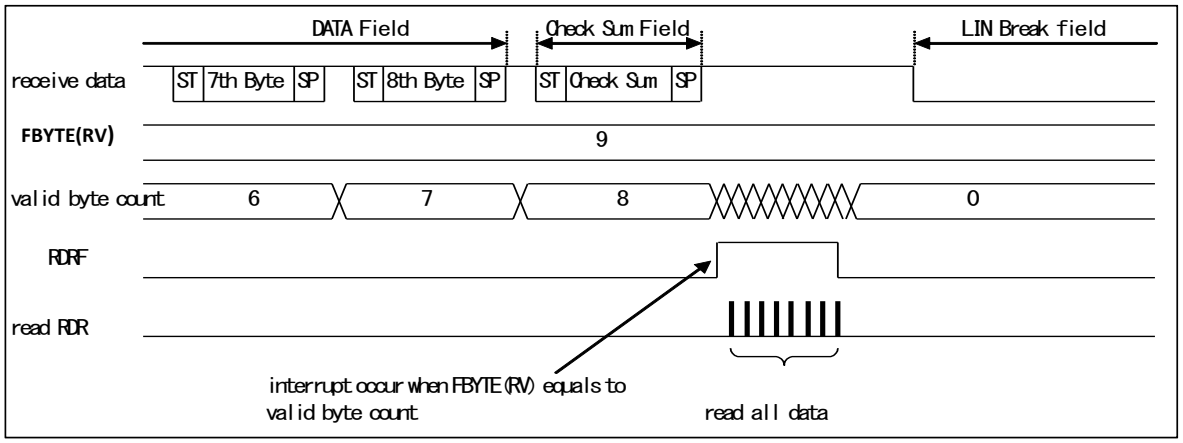
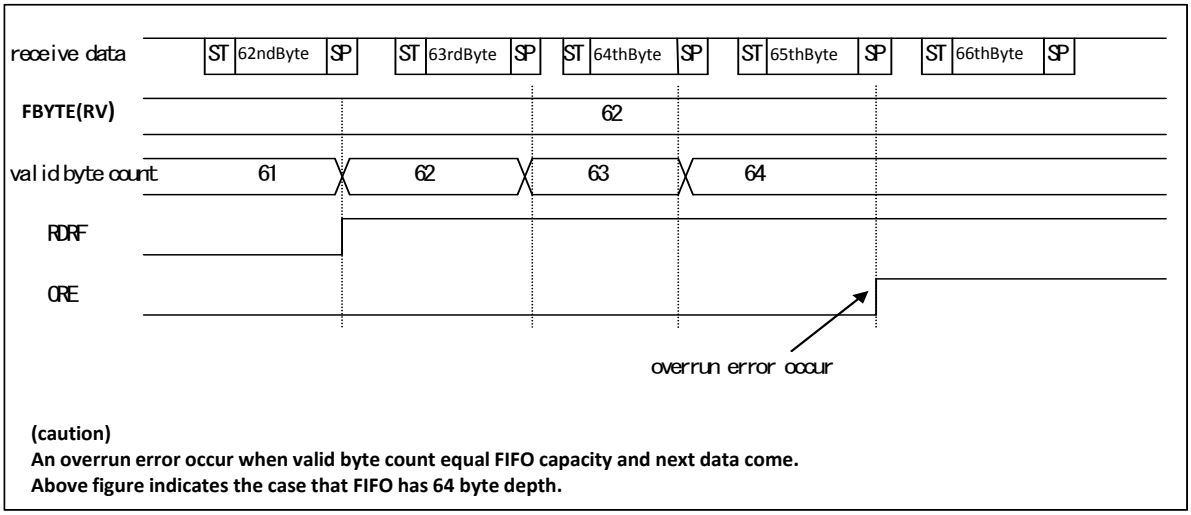


Figure 2-5 Set Timing of ORE (Overrun Error) Flag Bit



(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

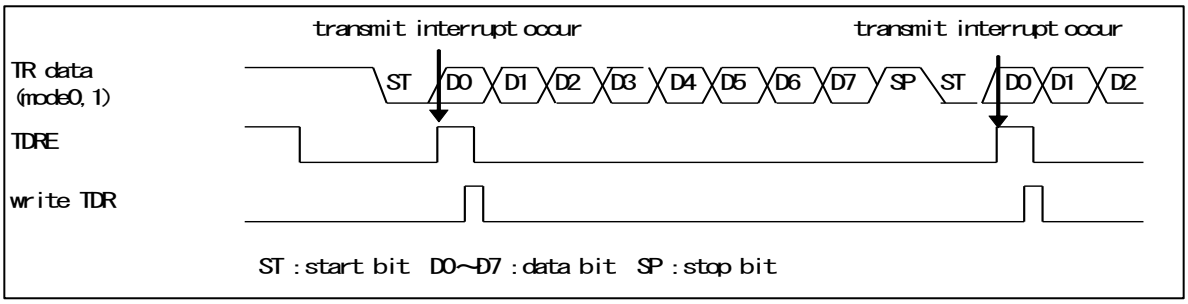
A data transmission interrupt occurs in the following cases: 1. The transmission data is transferred from the transmission data register (TDR) to the transmission shift register (SSR:TDRE = 1) to start data transmission. 2. Transmission is not performed (SSR:TBI = 1).

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).

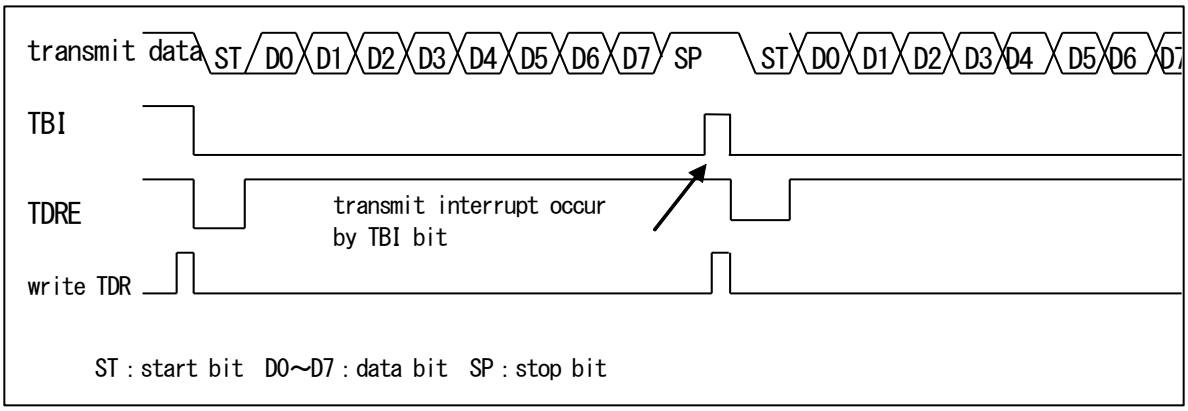
Figure 2-6 Set Timing of Transmission Data Empty Flag (SSR:TDRE)



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

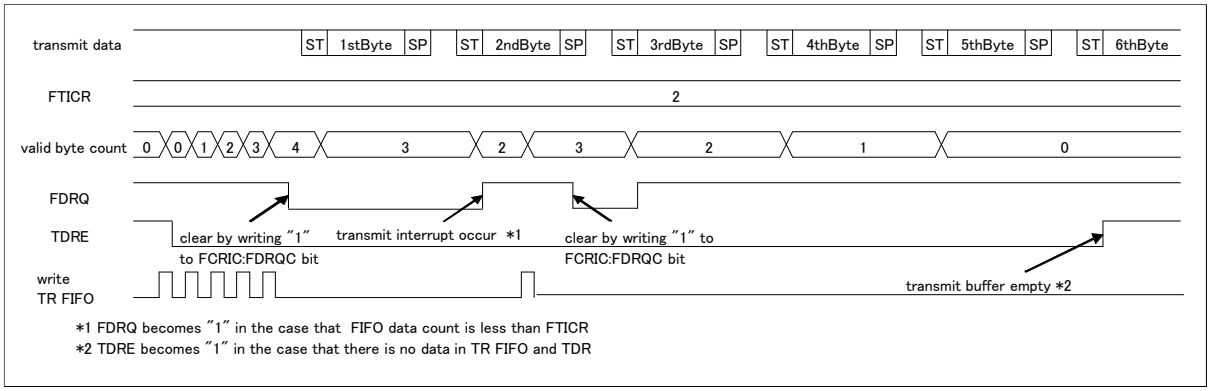
An interrupt occurs during the use of the transmission FIFO when the data count in the transmission FIFO is equal to or less than the value set in the transmission FIFO interrupt control register (FTICR).

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



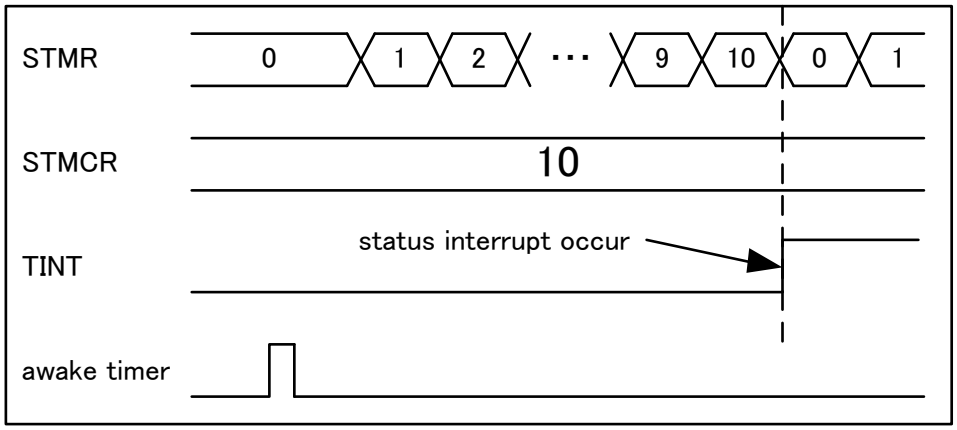
2.1.5. Occurrence of Timer Interrupts and Flag Set Timing

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).

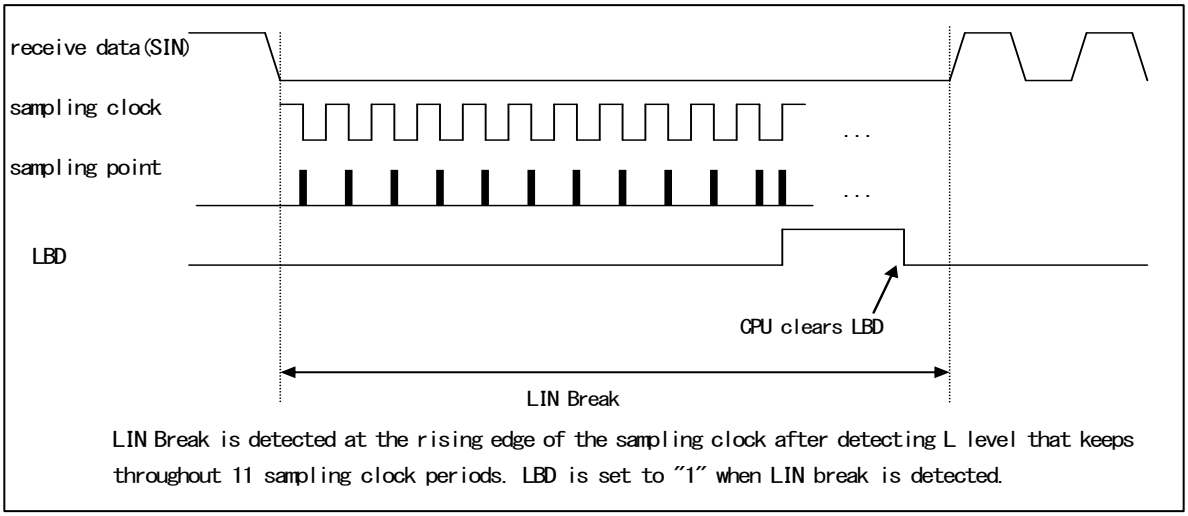
Figure 2-9 Occurrence Timing of Timer Interrupt



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

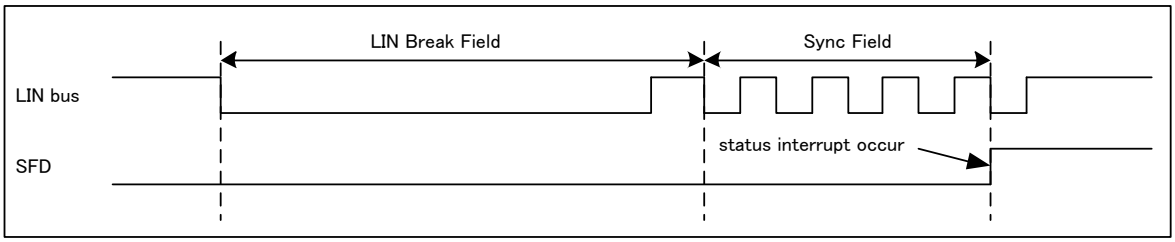
A Sync Field detection interrupt occurs if the detection of the Sync Field is complete.

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)
			Reception of as much data as the amount set in FBYTE		Reading of reception data (RDR) until the reception FIFO becomes empty
			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)
	FRE	SSR	Framing error		
	LBSER	LAMESR	LIN bus error detection	LAMIER: LBSERIE	Writing "1" to LAMESRC:LBSERC
	LSFER	LAMESR	LIN Sync Data error detection	LAMIER: LSFERIE	Writing "1" to LAMESRC:LSFERC
	LPTER	LAMESR	LIN ID parity error detection	LAMIER: LPTERIE	Writing "1" to LAMESRC:LPTERC
	LCSER	LAMESR	LIN checksum error detection	LAMIER: LCSERIE	Writing "1" to LAMESRC:LCSERC
	RXUDR	ESR	Reception FIFO underrun	ECR:REIE	Writing "1" to the reception FIFO underrun flag clear bit (ESRC:RXUDRC)
	RBERR	ESR	Reception block transfer error	ECR:REIE	Writing "1" to the reception block transfer error clear bit

Interrupt Type	Interrupt Request Flag Bit	Flag Register	Interrupt Factor	Interrupt Factor Enable Bit	Clearing of Interrupt Request Flag
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:TVOVRC)
	TBERR	ESR	Transmission block transfer error	ESR:TEIE	Writing "1" to the transmission block transfer error clear bit (ESRC:TBERRC)
Status (Assist Mode)	LBD	SSR	LIN Break Field detection	ESCR:LBIE	Writing "1" to the SSRC:LBDC bit
	SFD	SACSR	Sync Field detection	SACSR: SFDE	Writing "1" to the Sync Field detection flag clear bit (SACSRC:SFDC)
	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)
	LAHC	LAMSR	Completion of auto header	LAMIER: LAHCIE	Writing "1" to LAMSRC:LAHCC or reading from the ID register (LAMRID)
	LCSC	LAMSR	Completion of checksum calculation	LAMIER: LCSCIE	Writing "1" to LAMSRC:LCSCC

*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 Iterrupt 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).

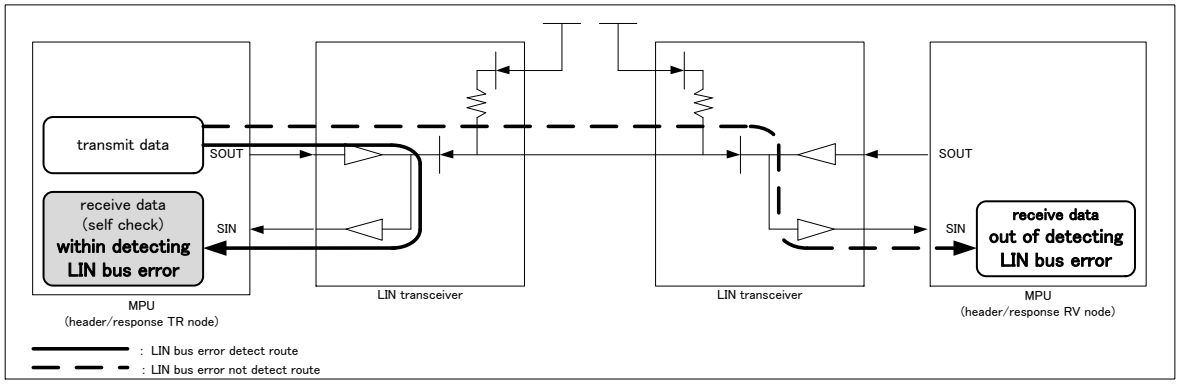
The timing at which the overrun error flag bit (SSR:ORE) is set is the same as that in manual mode, as described in "2.1.1 Occurrence of Reception Interrupts and Flag Set Timing." See "Figure 2-3 Set Timing of ORE (Overrun Error) Flag Bit."

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.

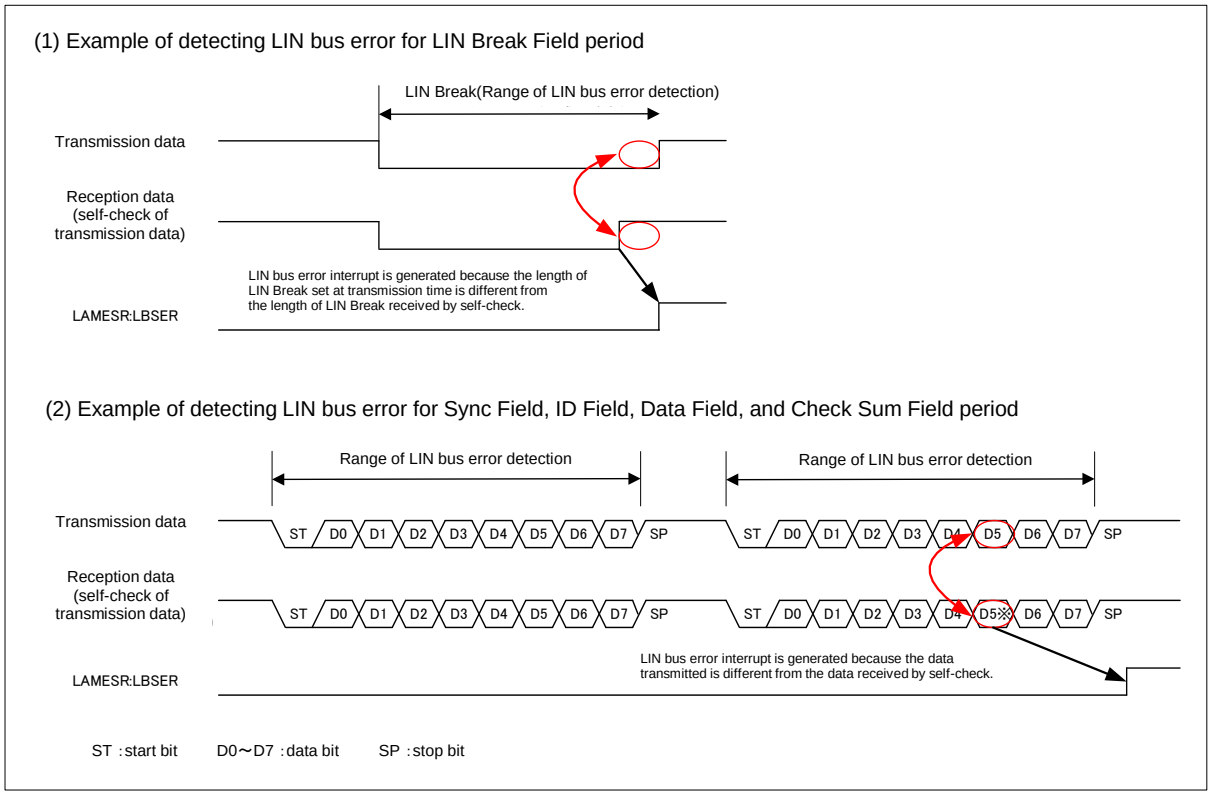
If an abnormality is detected as a result of comparing the transmission LIN Break length and the reception LIN Break length or the transmission data and reception data, then a LIN bus error is detected and a flag is set (LAMESR:LBSER = 1). A reception interrupt occurs if the interrupt is enabled (LAMIER:LBSERIE = 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)

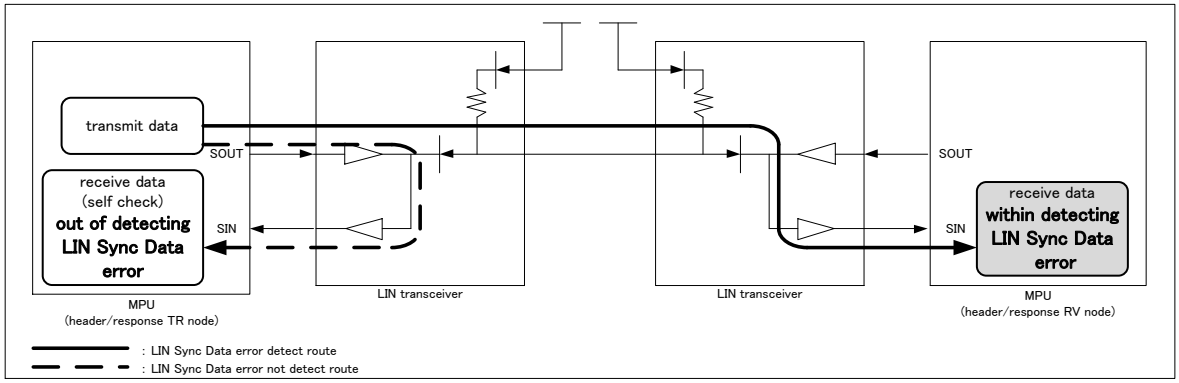


Occurrence of LIN Sync Data Error Detection Interrupt and Flag Set Timing

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

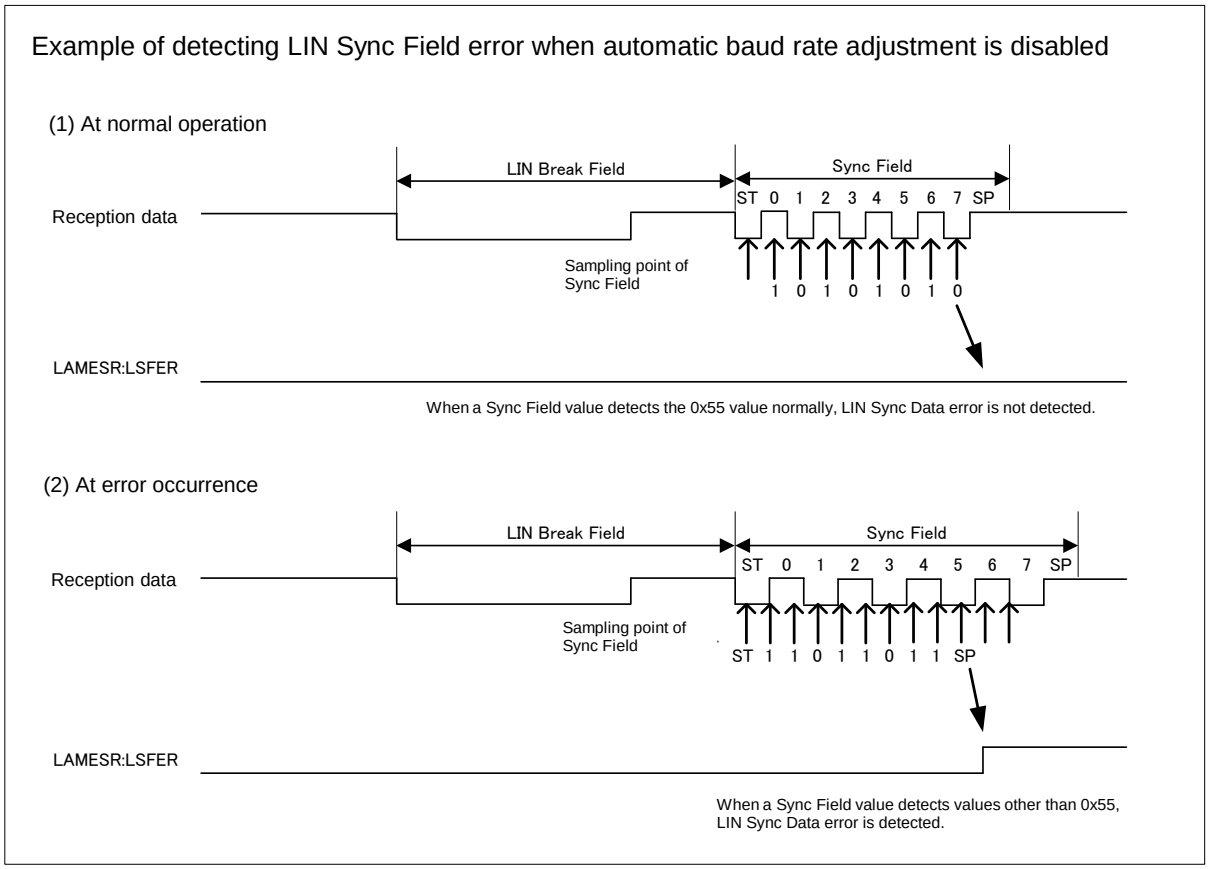


The target range of LIN Sync Data error detection is the start bit and byte data of the Sync Field. The stop bit is not included in the LIN Sync Data error detection target. A framing error (SSR:FRE = 1) is detected when the "L" level is detected on the stop bit.

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)

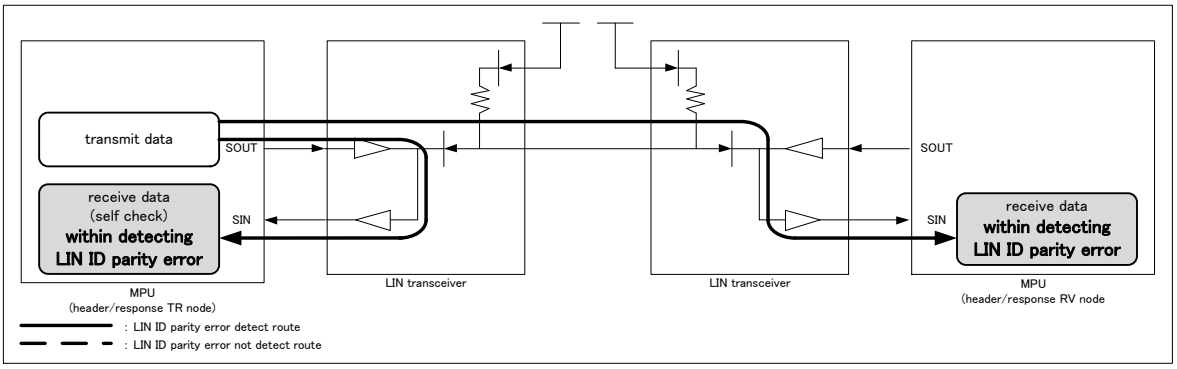


Occurrence of LIN ID Parity Error Detection Flag Interrupt and Flag Set Timing

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).

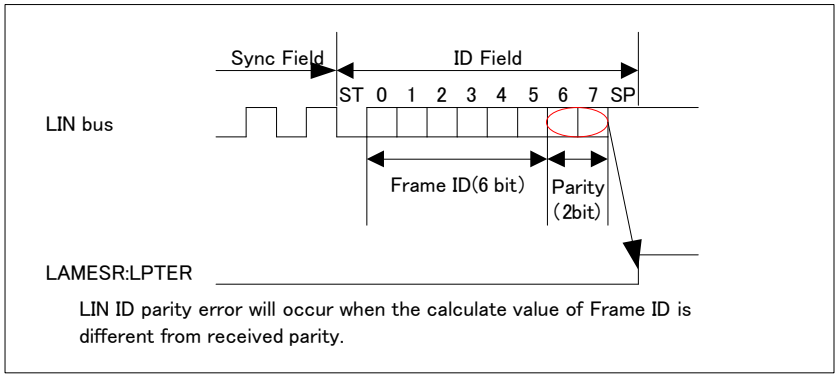
At this time, a reception interrupt occurs if the interrupt is enabled (LAMIER:LPTERIE = 1).

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)

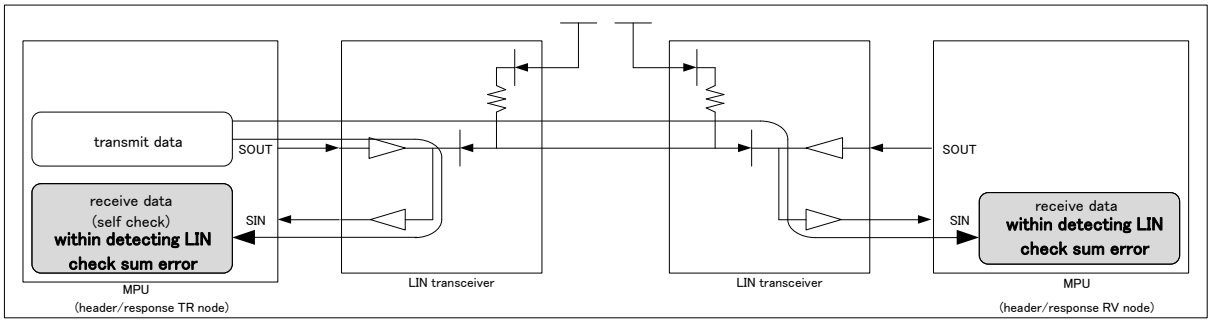


Occurrence of LIN Checksum Error Detection Flag Interrupt and Flag Set Timing

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:LDL). 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:LDL). 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).

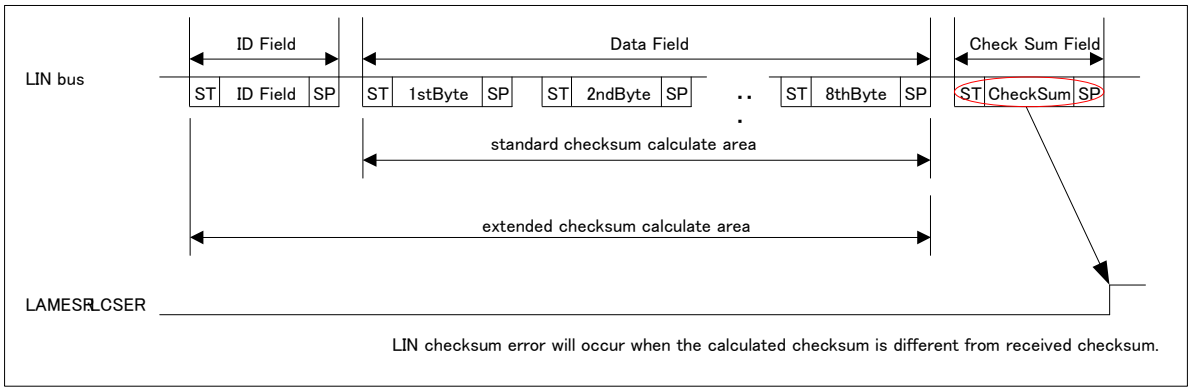
Occurrence of LIN Checksum Error Detection Interrupt and Flag Set Timing When Setting Extended Checksum Calculation

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:LDL). 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:LDL). 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)

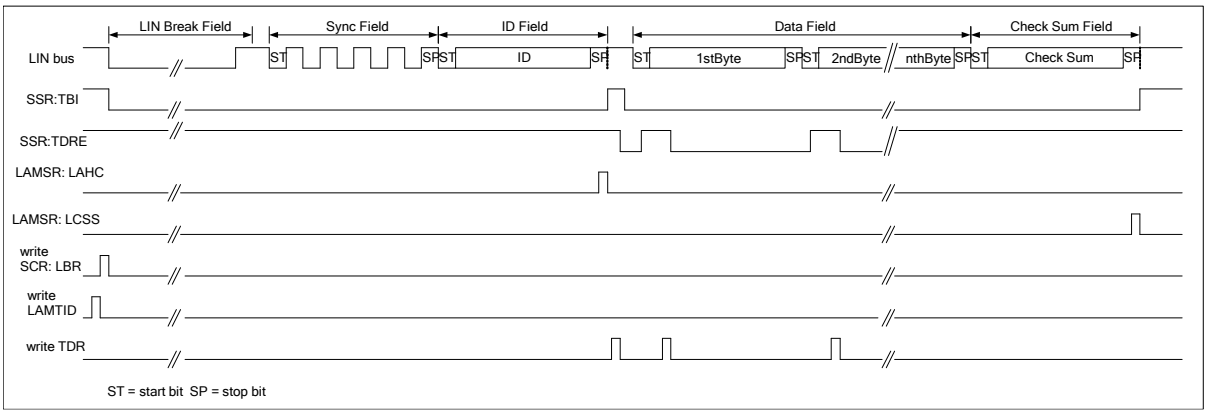


Figure 2-21 Set Timing of Transmission Bus Idle Flag (TBI) (When Master Response Is Transmitted and ID Register Is Not Used)

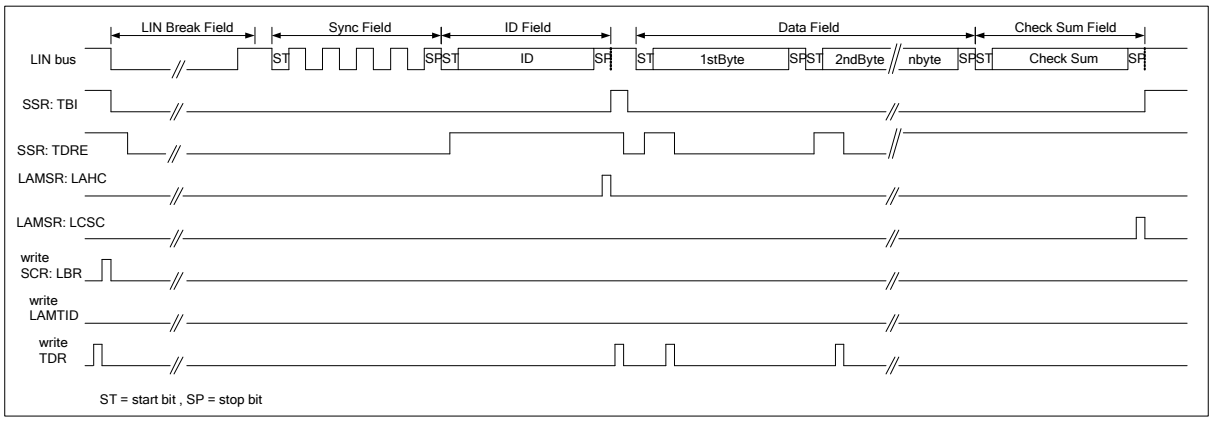
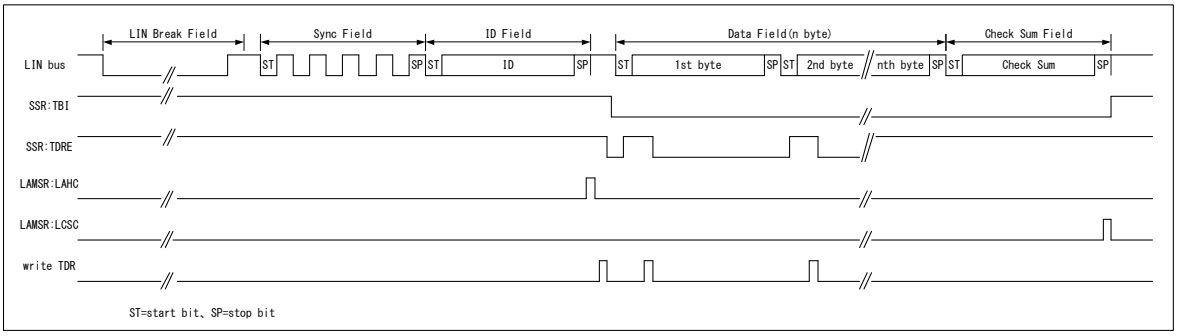


Figure 2-22 Set Timing of Transmission Bus Idle Flag (TBI) (When Slave Response Is Transmitted)



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"

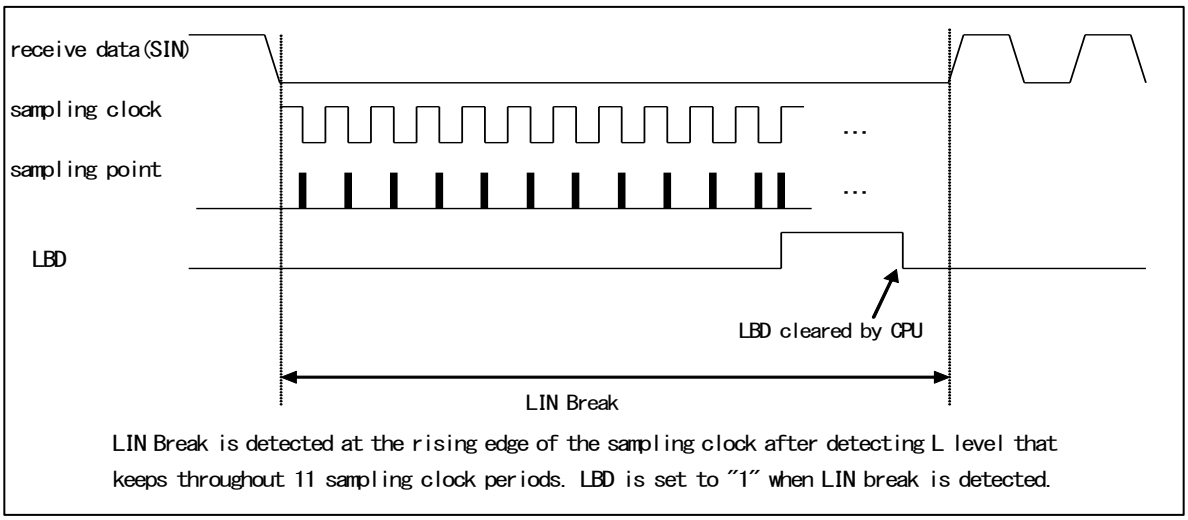
2.2.6. Occurrence of Status Interrupt and Flag Set Timing in Assist Mode

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:

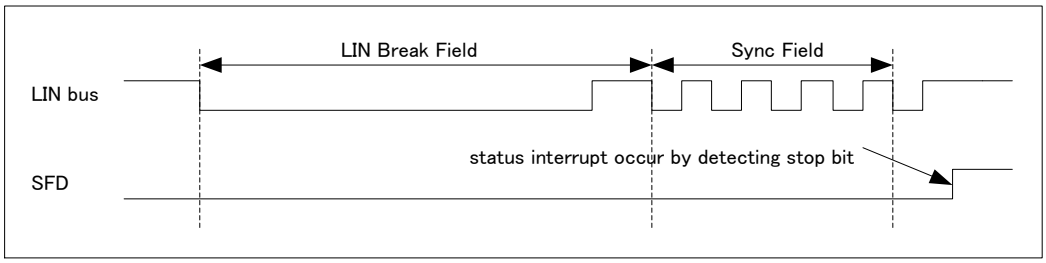
- *During the reception of a LIN Break Field, a framing error is detected before the detection of a LIN Break Field if reception is enabled (SCR:RXE = 1).*

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)

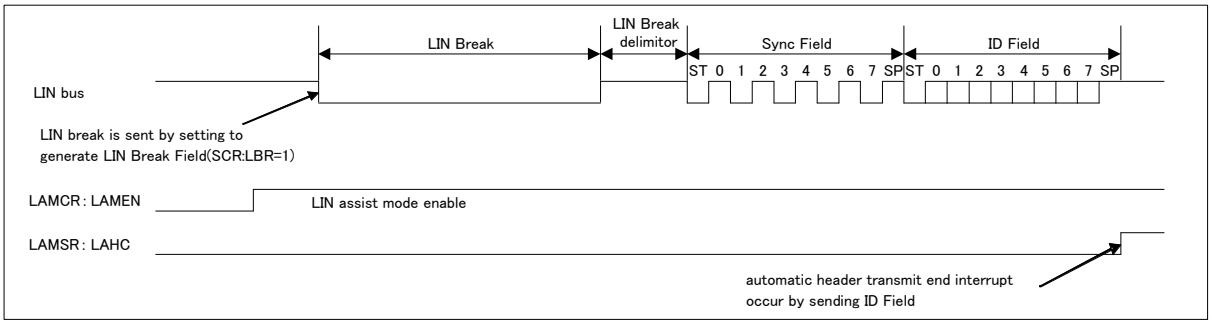


Auto Header Completion Interrupt and Flag Set Timing at Transmission

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)

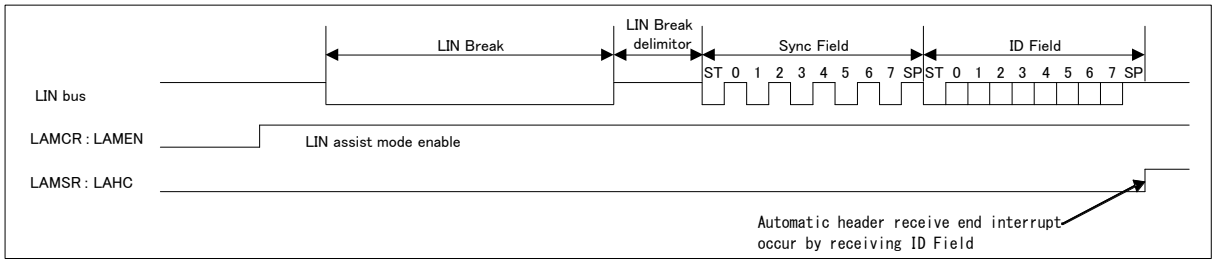


Auto Header Reception Completion Interrupt and Flag Set Timing at Reception

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)

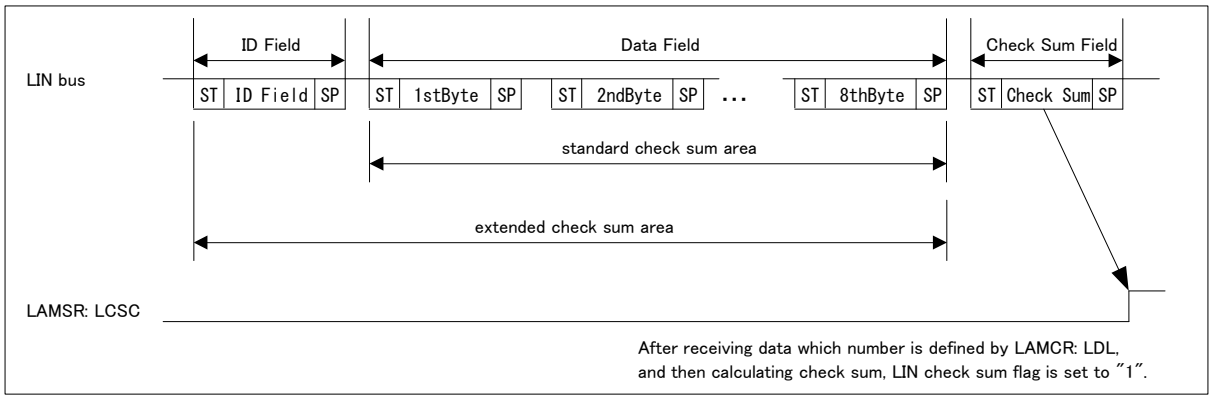


Occurrence of LIN Checksum Detection Completion Flag Interrupt and Flag Set Timing

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

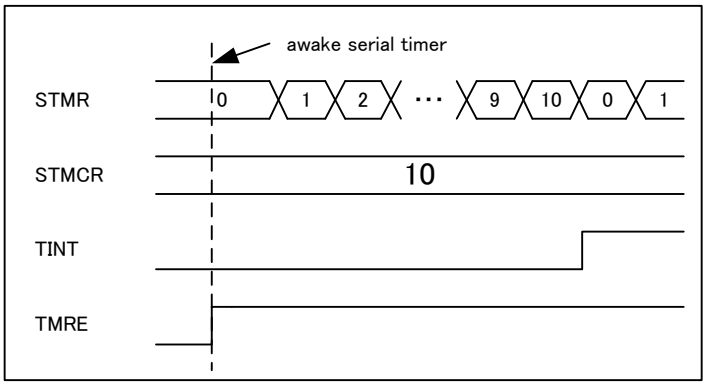
Activating Serial Timer

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.

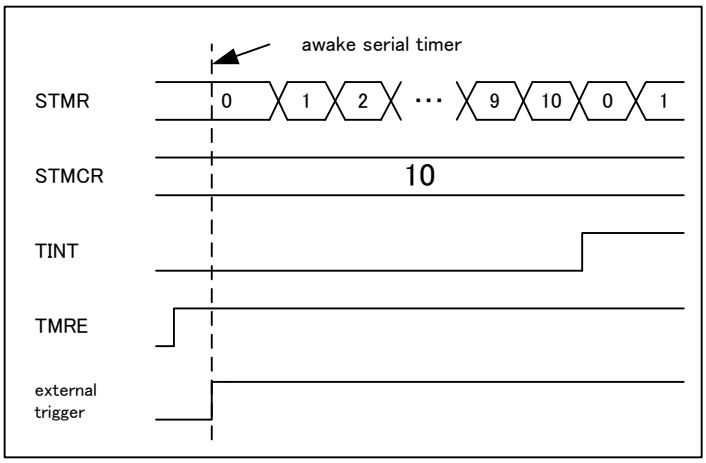
Figure 3-1 Activation by Serial Timer Enable Bit (STMCR = 10, SACSR:TRGE = 0)



- 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".

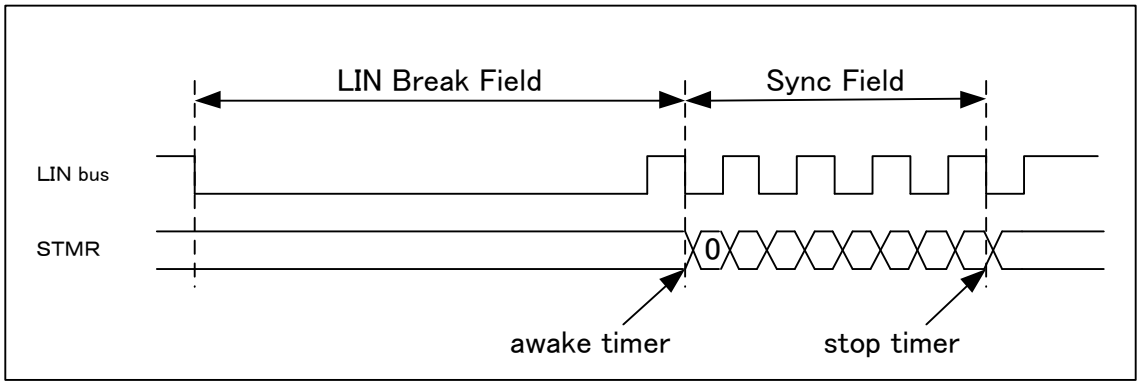
Figure 3-2 Activation by an External Trigger (STMCR = 10, SACSR:TRGE = 1, TRG1 = 0, TRG0 = 0)



■ Activation by Sync Field reception

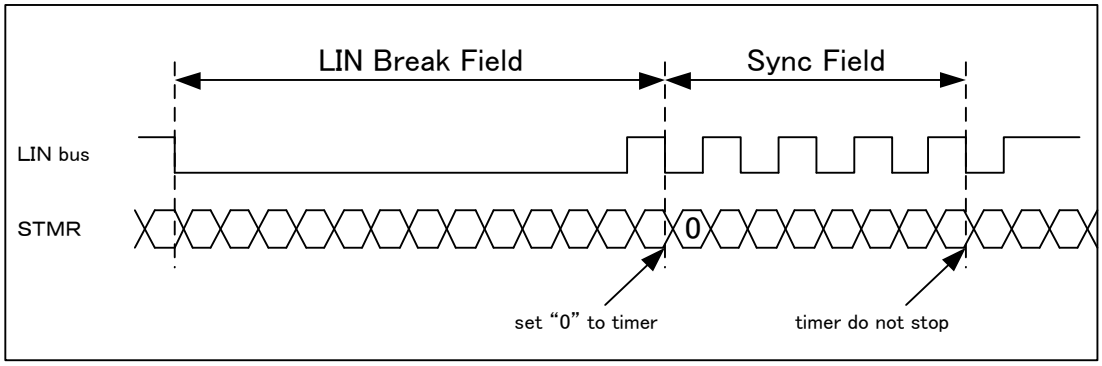
The serial timer is activated and counting of the serial timer register (STMR) starts from 0 when all of the following are satisfied: 1. The serial timer is stopped. 2. The auto baud rate adjustment bit (SACSR:AUTE) is set to "1". 3. The LIN interface (v2.1) detects the first falling edge of the Sync 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)

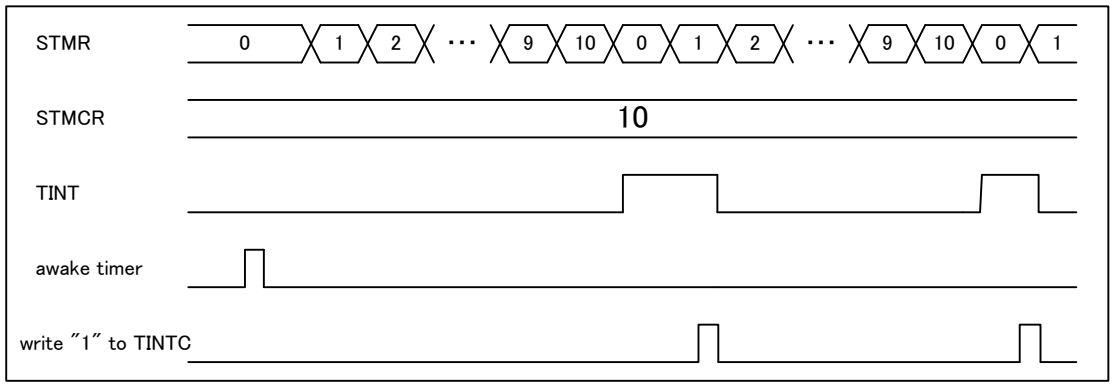


Figure 3-6 Serial Timer Initialization

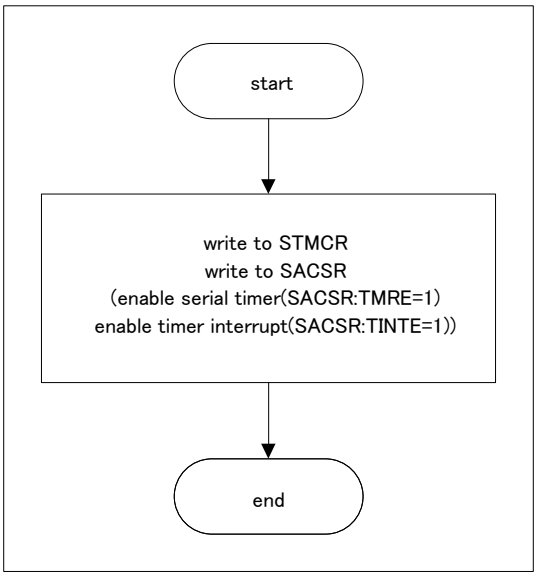
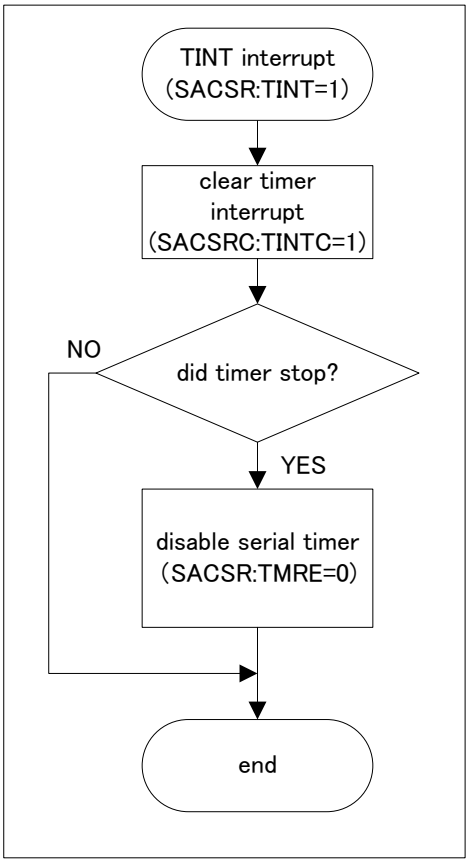


Figure 3-7 Serial Timer Interrupt



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.

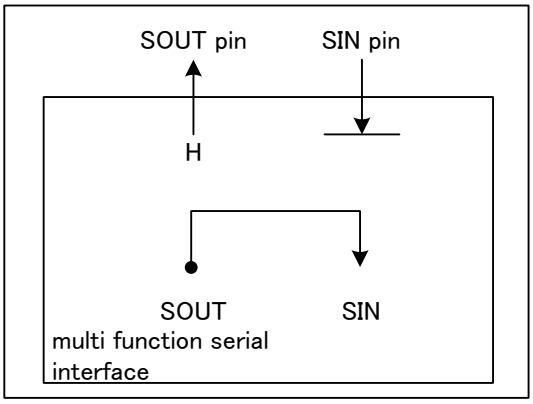
4.1. Manual Mode

Serial Test Mode

When serial test mode is enabled (SACSR:STST = 1), SOUT and SIN are connected together within the multifunction 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 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.

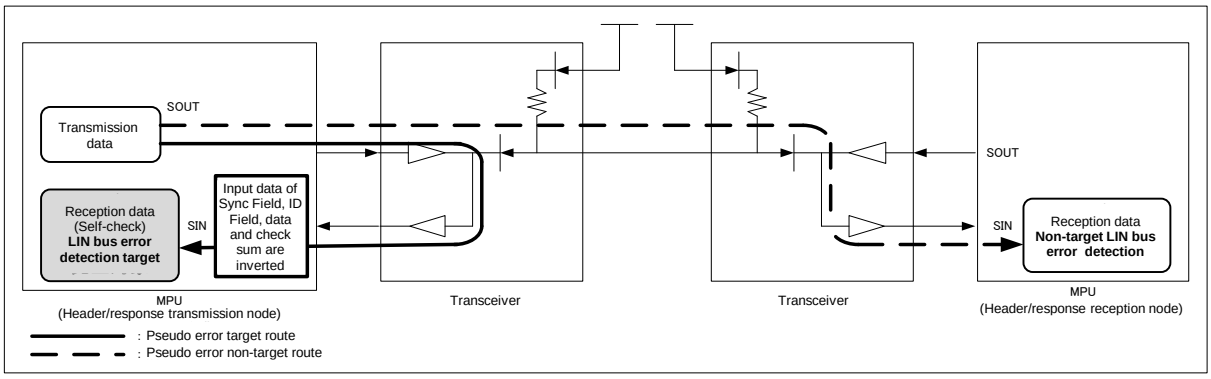
Notes:

- *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:

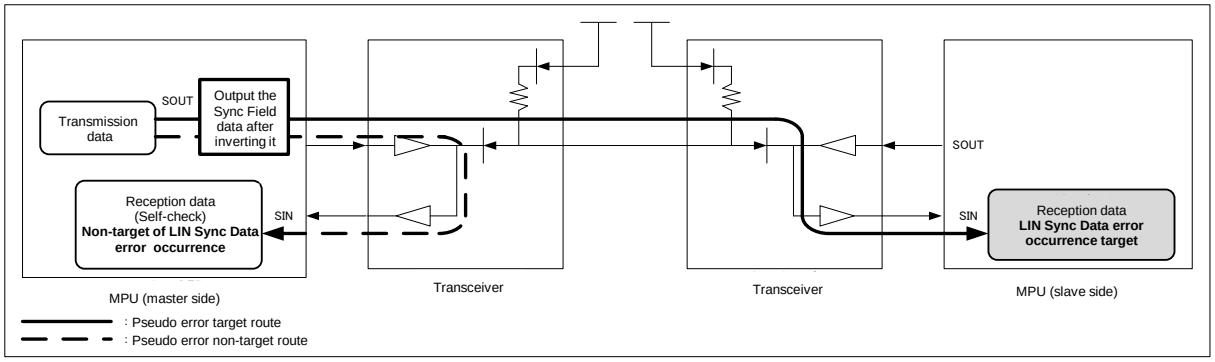
- *When a LIN bus error is detected (LAMESR:LBSER = 1), the transmission/reception processing of the header and response sections in assist mode stops.*
- *Enable this test mode only when transmitting a header or response.*

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.

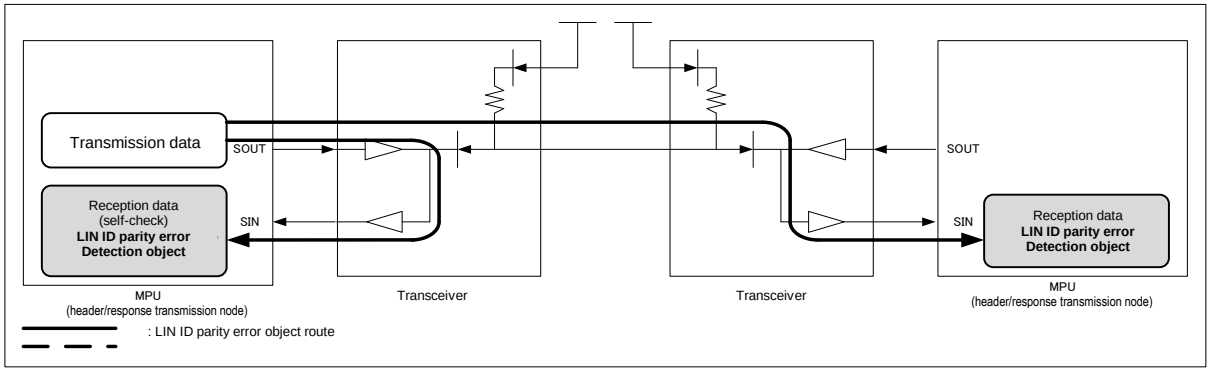
Notes:

- *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".

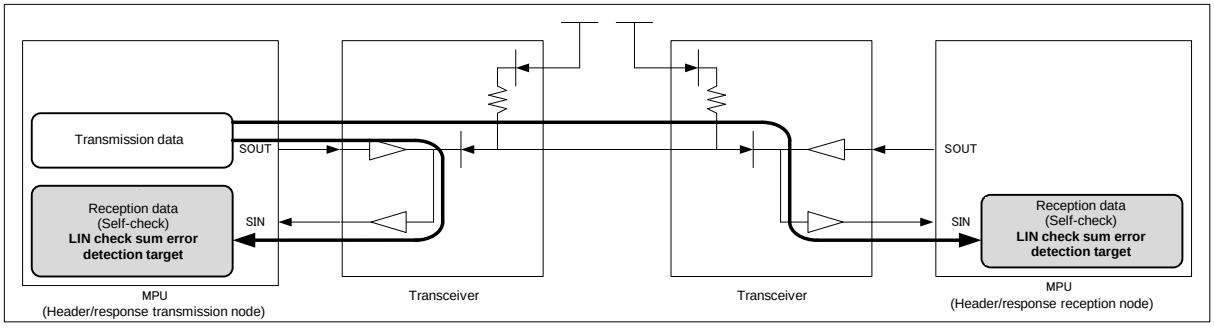
Notes:

- *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.

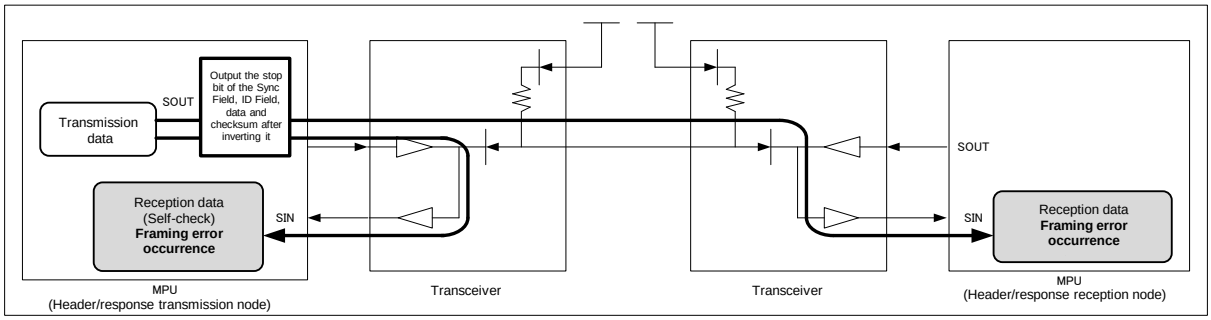
Notes:

- *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.

Note:

- *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 (%)
```

Note:

- *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)

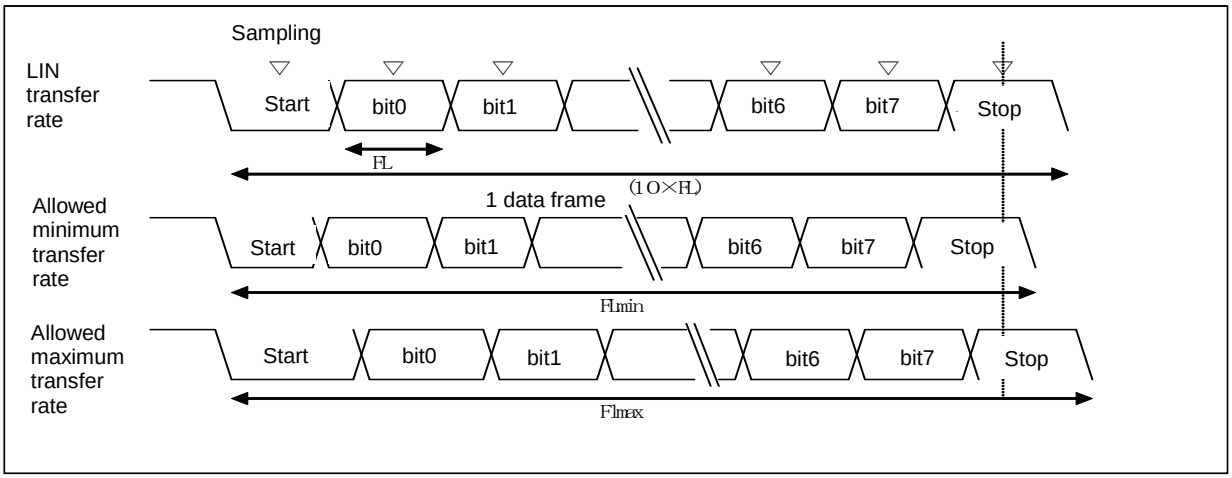
Baud Rate	40 MHz		48 MHz		72 MHz		80 MHz	
(bps)	Value	ERR	Value	ERR	Value	ERR	Value	ERR
8 M	4	0	5	0	8	0	9	0
6 M	-	-	7	0	11	0	-	-
5 M	7	0	-	-	-	-	15	0
4 M	9	0	11	0	17	0	19	0
2.5 M	15	0	-	-	-	-	31	0
2 M	19	0	23	0	35	0	39	0
1 M	39	0	47	0	71	0	79	0
500000	79	0	95	0	143	0	159	0
460800	86	-0.22	103	0.16	155	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 * Flmax = (10 bits * (V + 1) - (V + 1) / 2) / ϕ V: Reload value, ϕ: Bus clock

Flmax = (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 * Flmax = (10 bits * (V + 1) - (V + 1) / 2 - 2) / ϕ V: Reload value, ϕ: Bus clock

Flmax = (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 acknowledgment standard (ACK mode) and 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 reception block size register.

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.

Note:

- *ACK Mode of Block Transfer is not supported for this device.*

6.1. ACK Mode

This section describes block transfer in ACK mode.

Reception Block Transfer

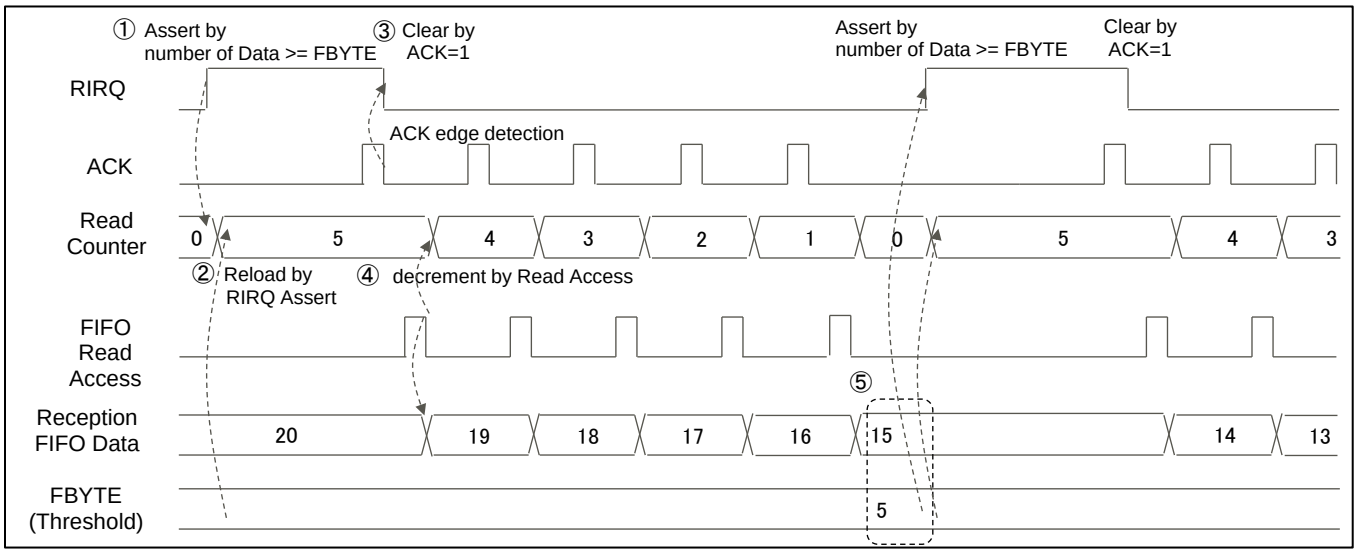
Reception block transfer is performed as described below.

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 a reception acknowledgment is input, the reception block transfer request (RIRQ) is cleared. With the lead of the 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.

Figure 6-1 Reception Block Transfer in ACK Mode



Transmission Block Transfer

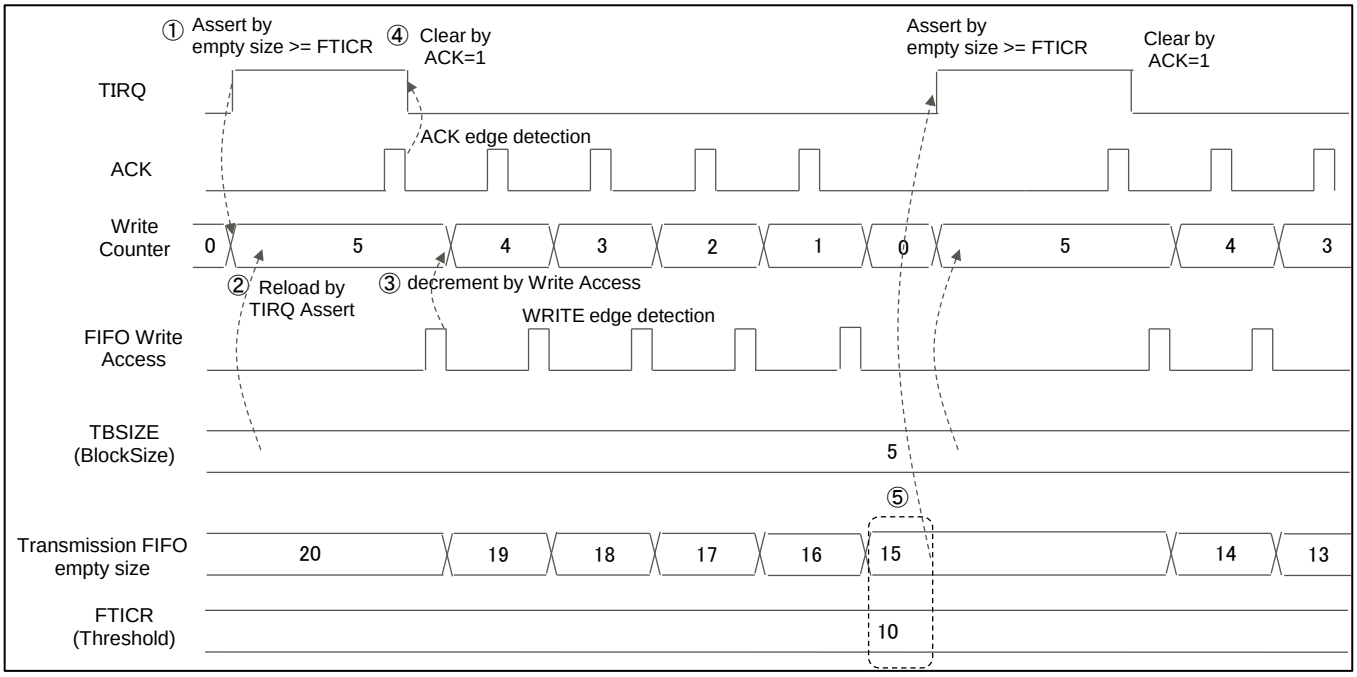
Transmission block transfer is performed as described below.

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 transmission acknowledgment is input, the transmission block transfer request (TIRQ) is cleared. With the write of the 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.

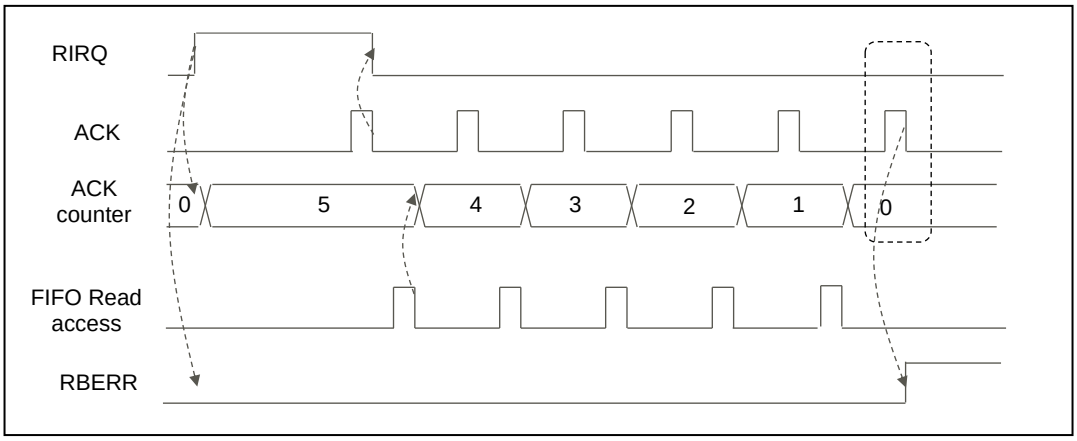
Figure 6-2 Transmission Block Transfer in ACK Mode



Reception Block Transfer Error

If, while the Read counter is "0", an acknowledgment is input before the next reception block transfer request (RIRQ) is asserted, a reception block transfer error occurs, and the ESR:RBERR bit is asserted.

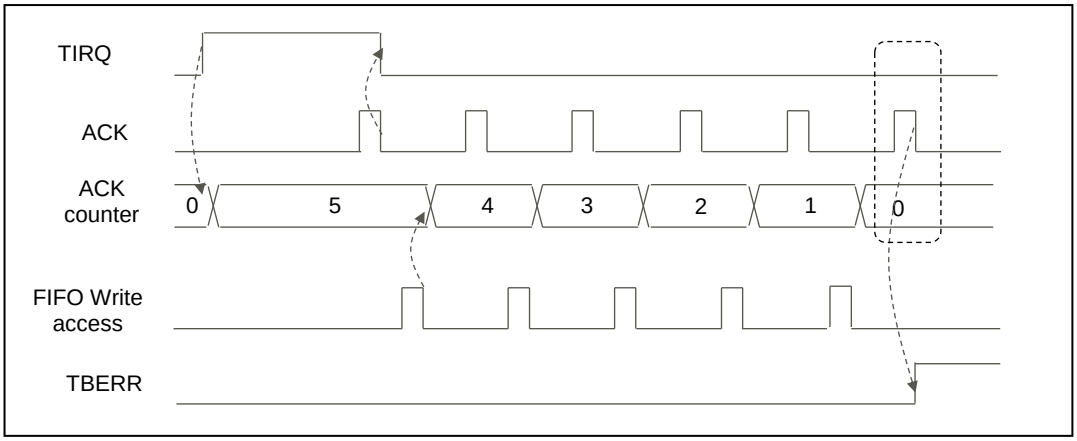
Figure 6-3 Reception Block Transfer Error in ACK Mode



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.

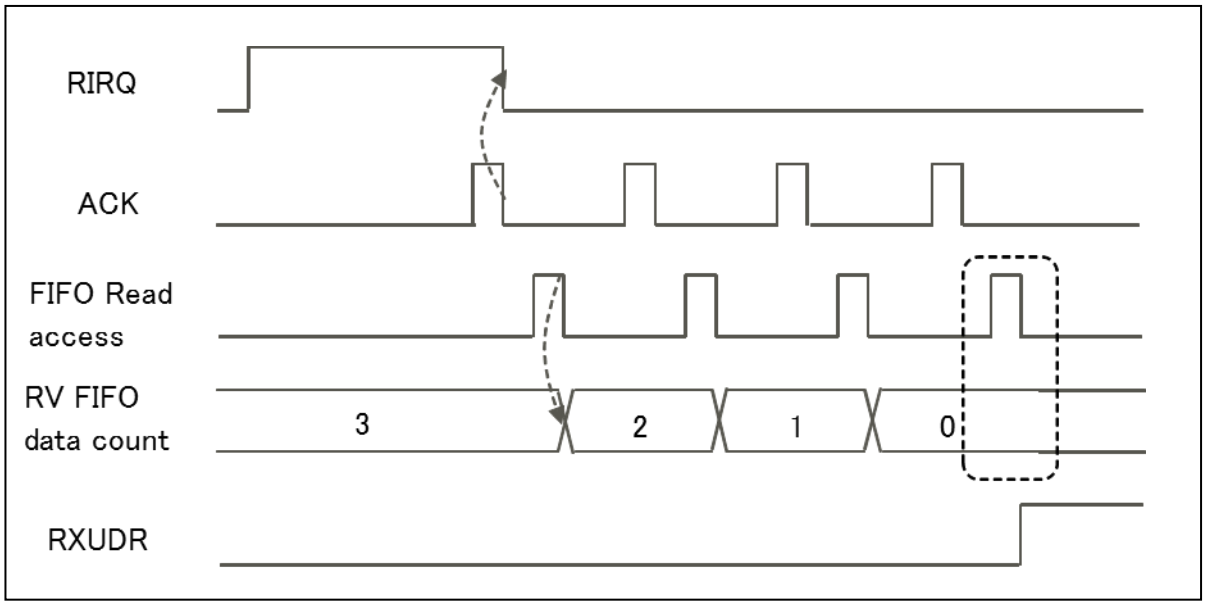
Figure 6-4 Transmission Block Transfer Error in ACK Mode



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.

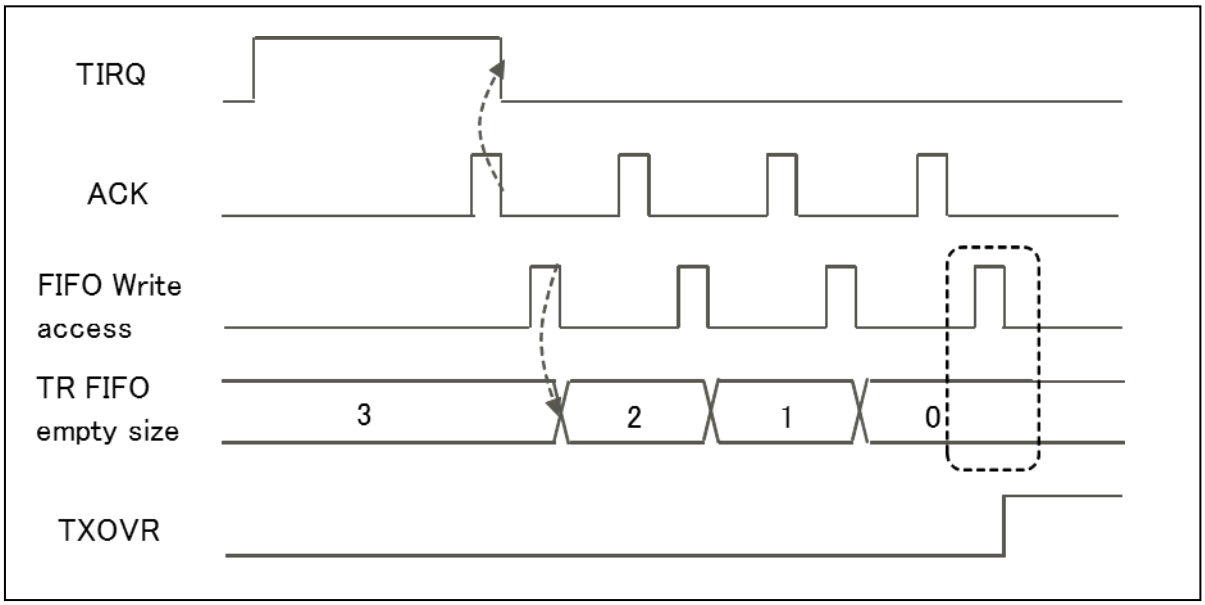
Figure 6-5 Reception FIFO Underrun in ACK Mode



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 ACK Mode



6.2. RW Access Mode

This section describes block transfer in RW access mode.

Reception Block Transfer

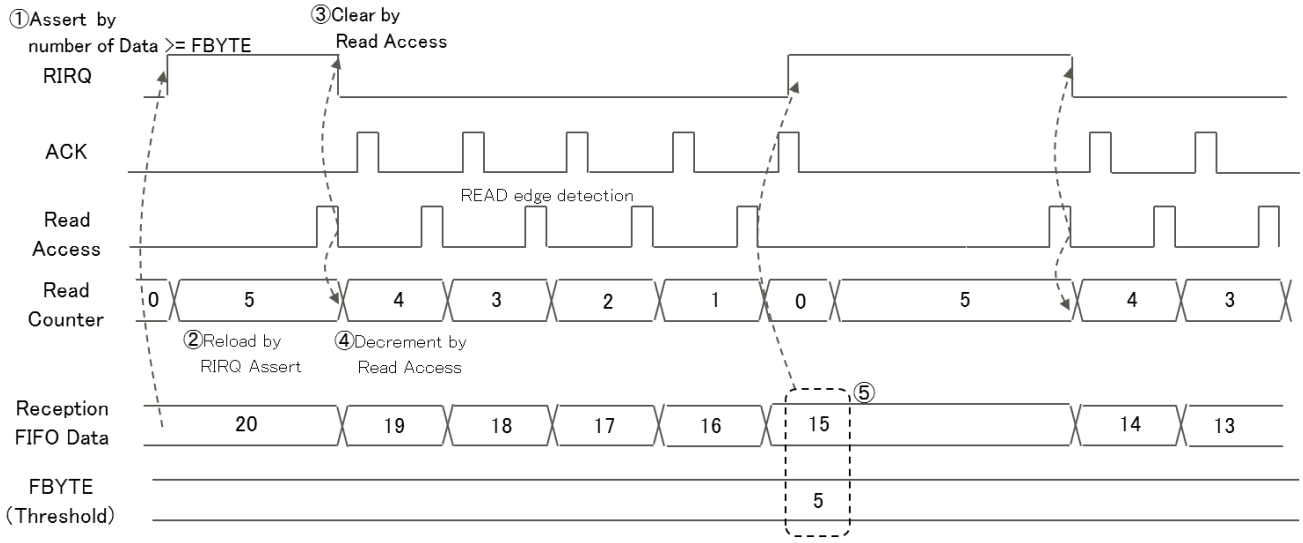
Reception block transfer is performed as described below.

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 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.

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.

Figure 6-7 Reception Block Transfer in RW Access Mode



Transmission Block Transfer

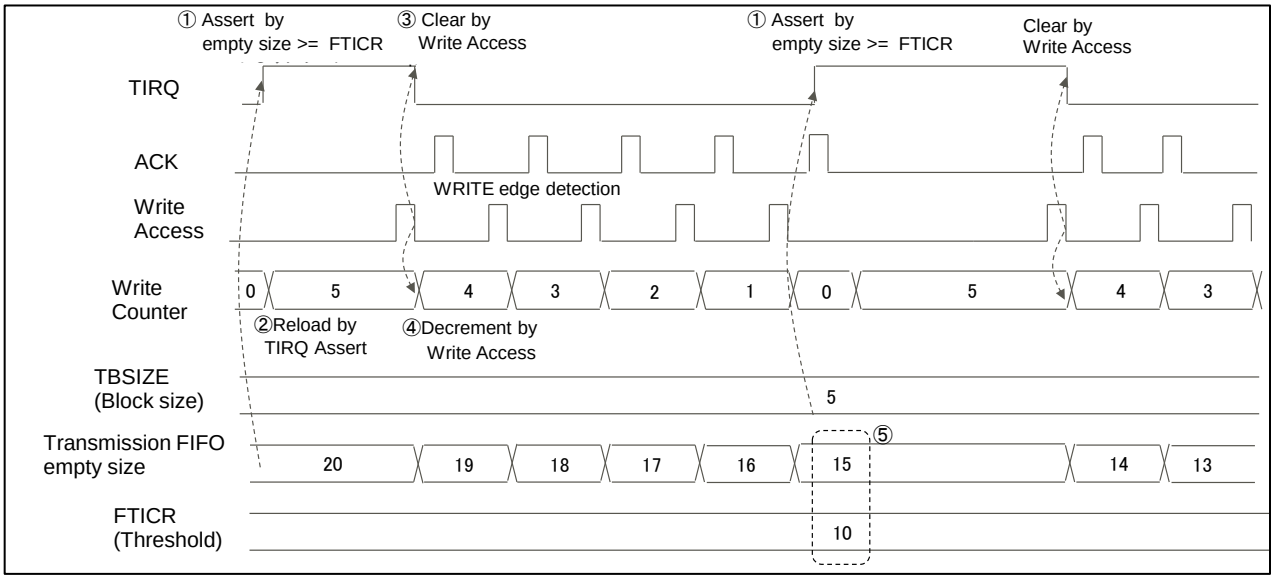
Transmission block transfer is performed as described below.

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 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.

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.

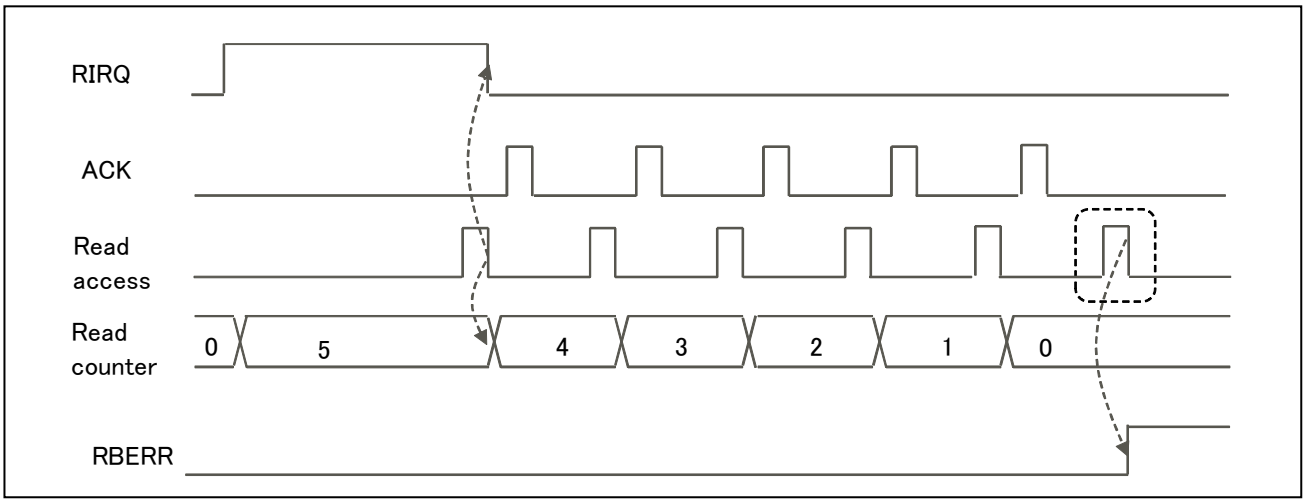
Figure 6-8 Transmission Block Transfer in RW Access Mode



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.

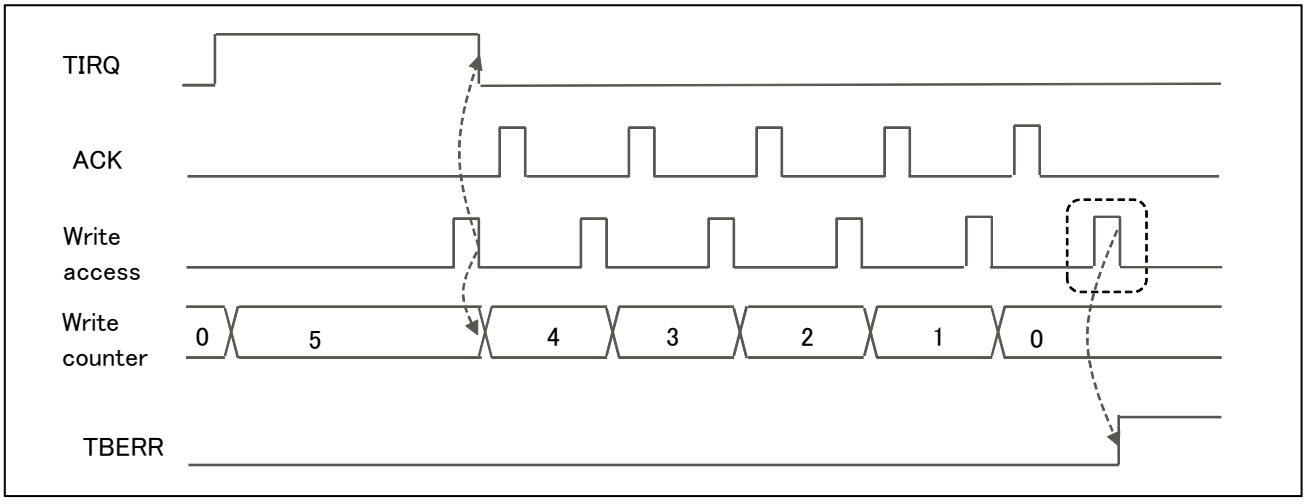
Figure 6-9 Reception Block Transfer Error in RW Access Mode



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 FSR:TBERR bit is asserted.

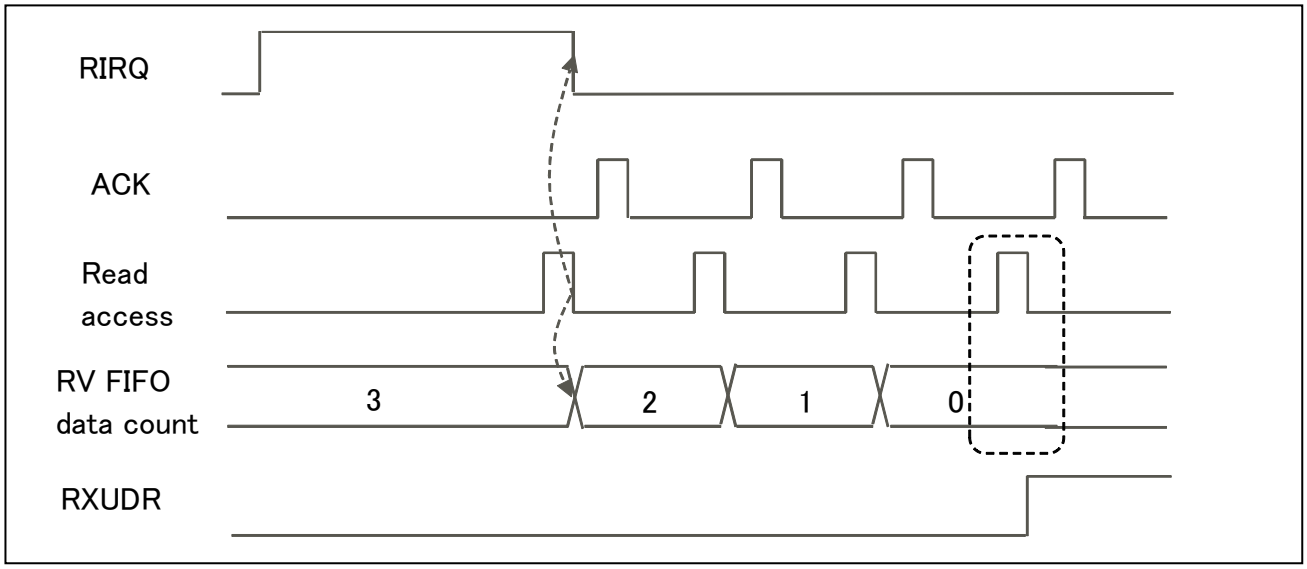
Figure 6-10 Transmission Block Transfer Error in RW Access Mode



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.

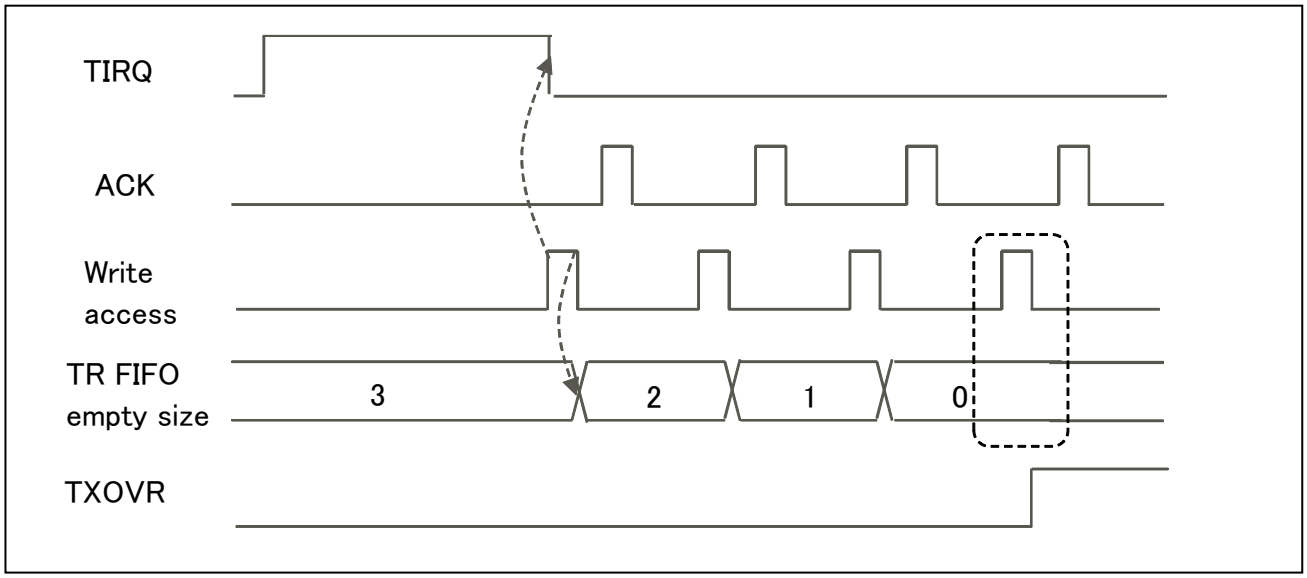
Figure 6-11 Reception FIFO Underrun in RW Access Mode



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 6-12 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".

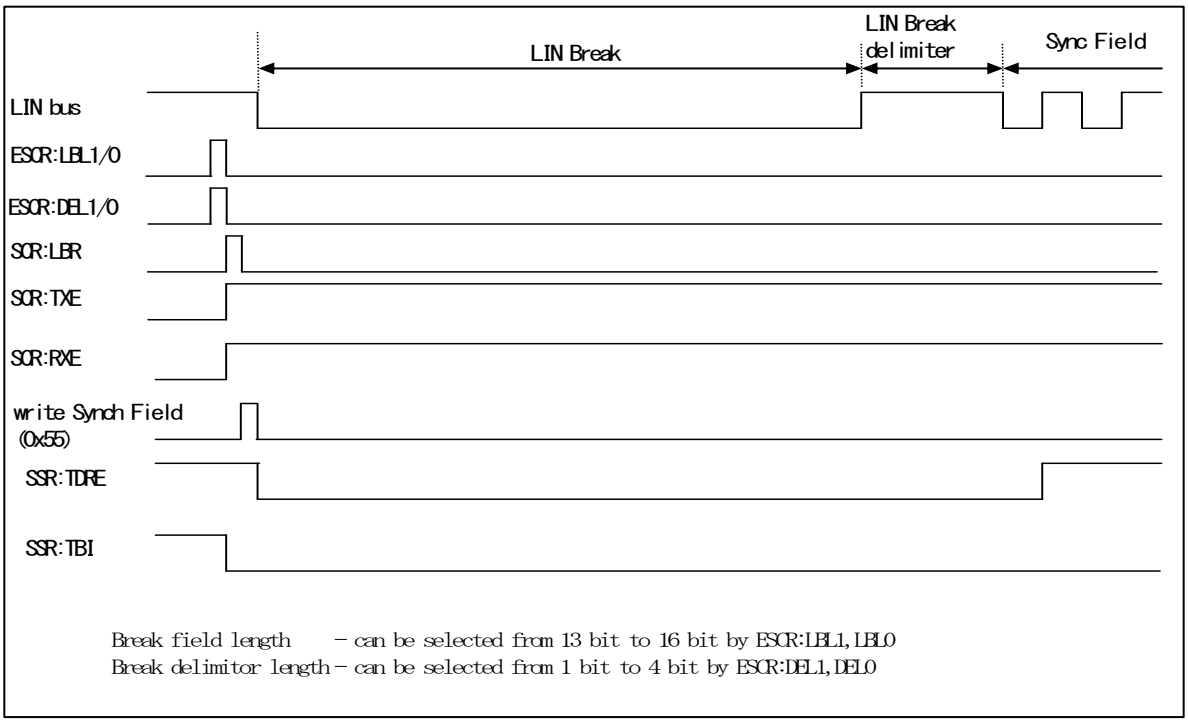
LIN Break Field Transmission to Sync Field Transmission

- 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.*

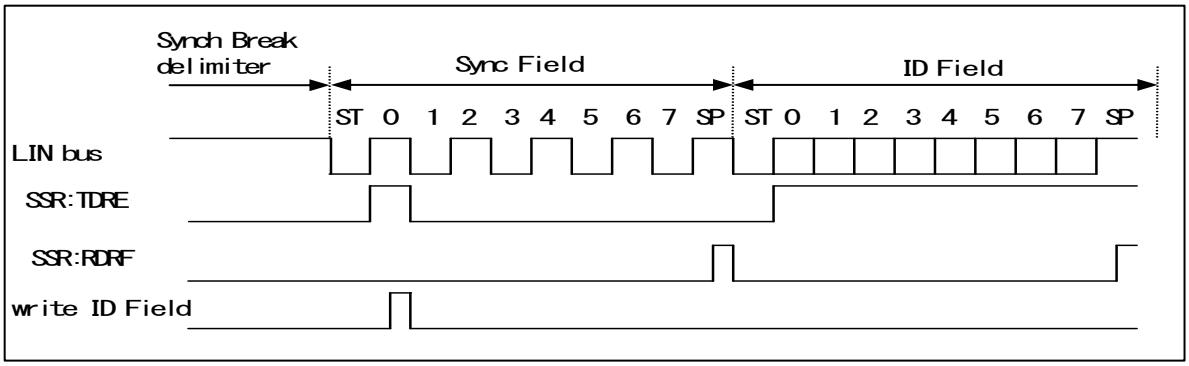
Figure 7-1 LIN Break Field to Sync Field Transmission



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



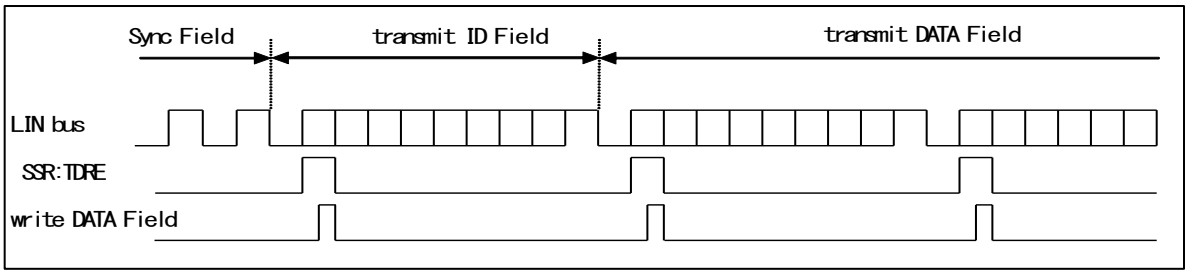
ID Field Transmission to DATA Field Transmission and Reception

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

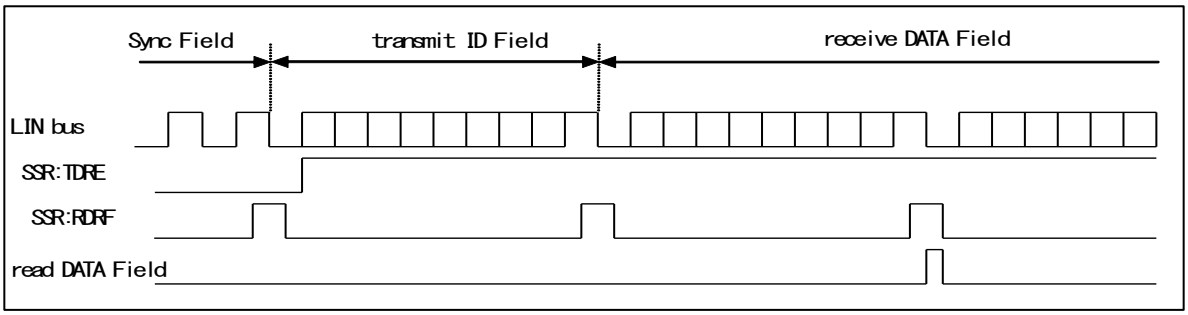


(When the DATA Field is received)

- 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 (SSR: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)

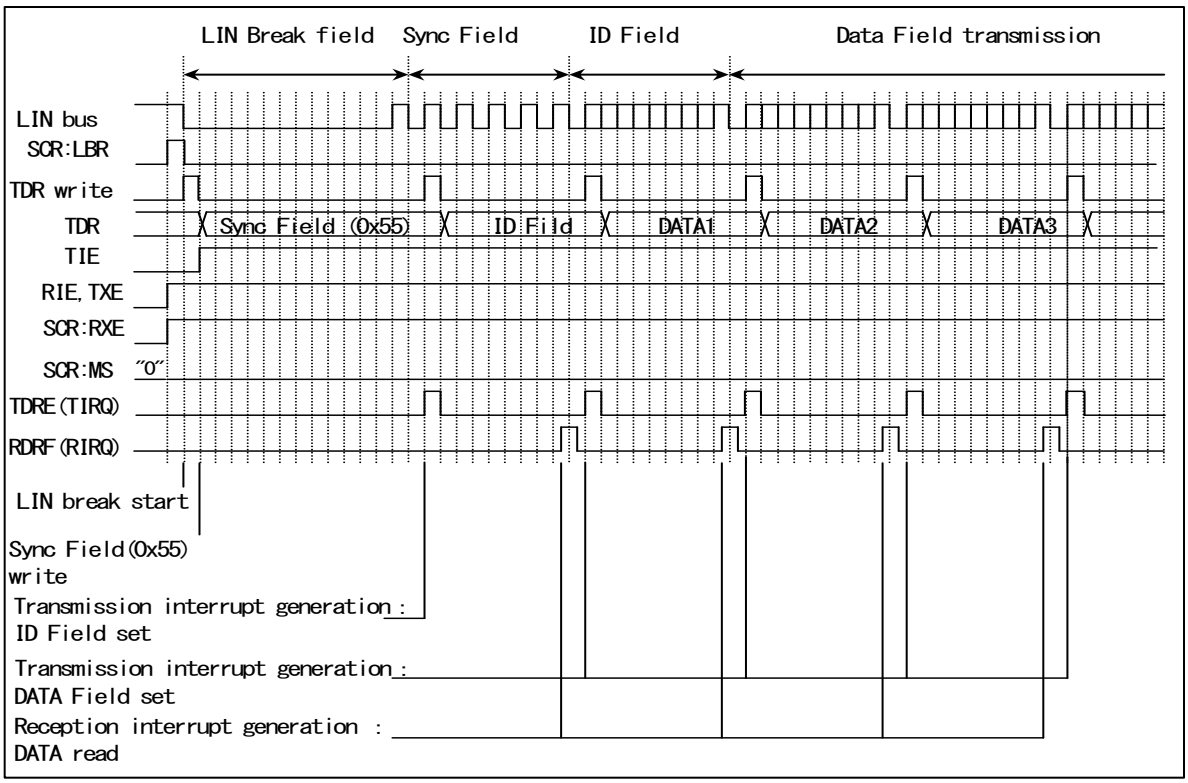
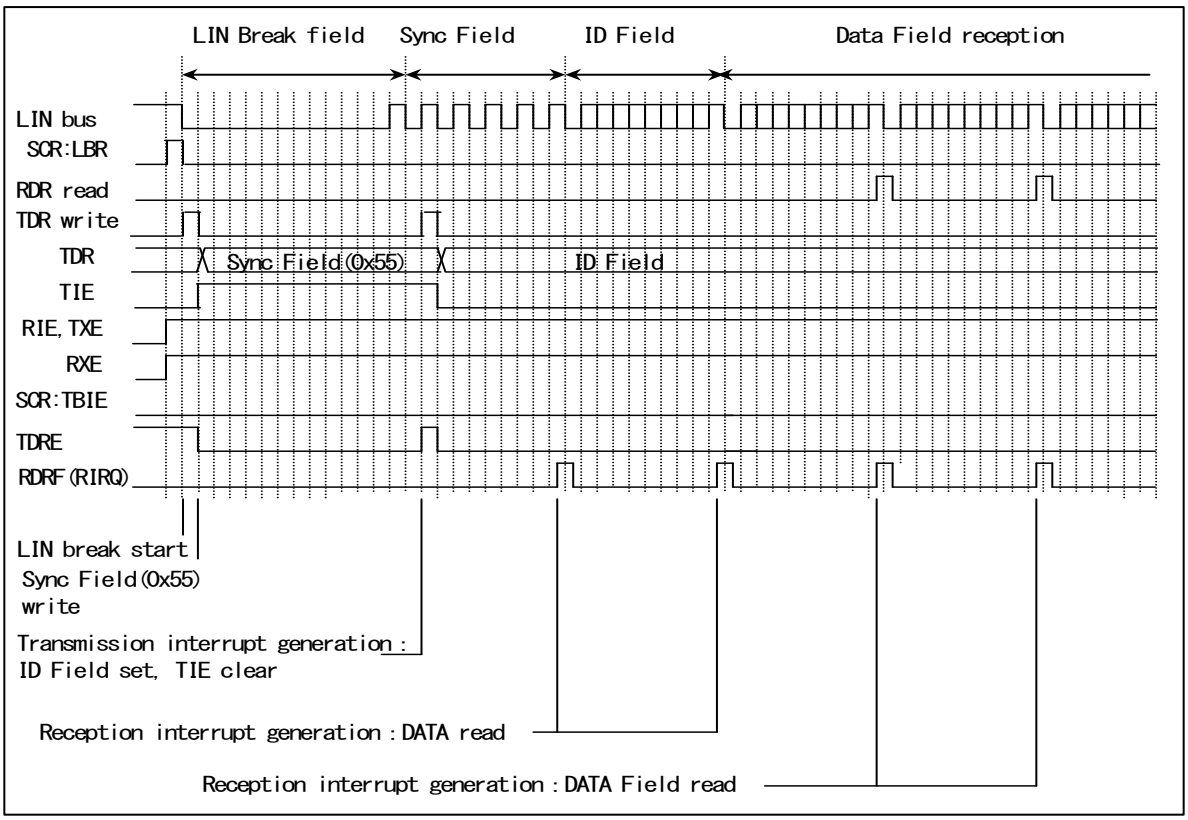


Figure 7-6 LIN Bus Timing (When a DATA Field Is Received: When a FIFO Is Not Used)



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)

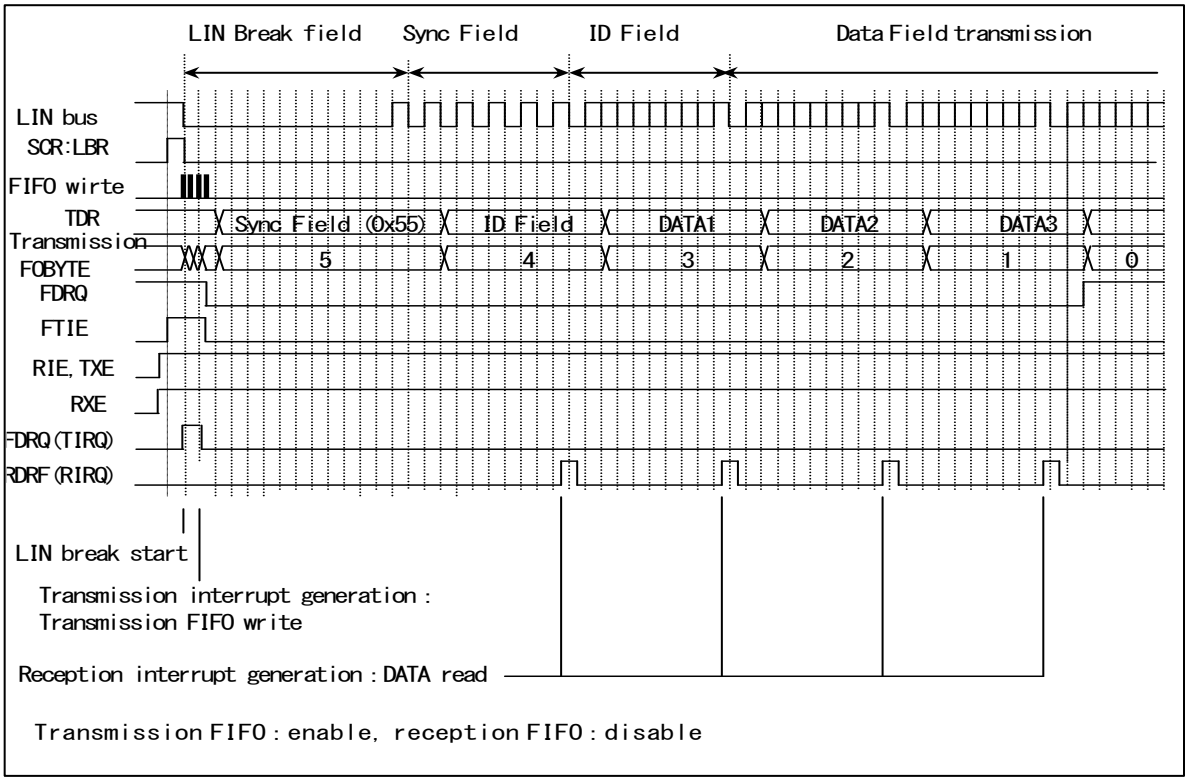
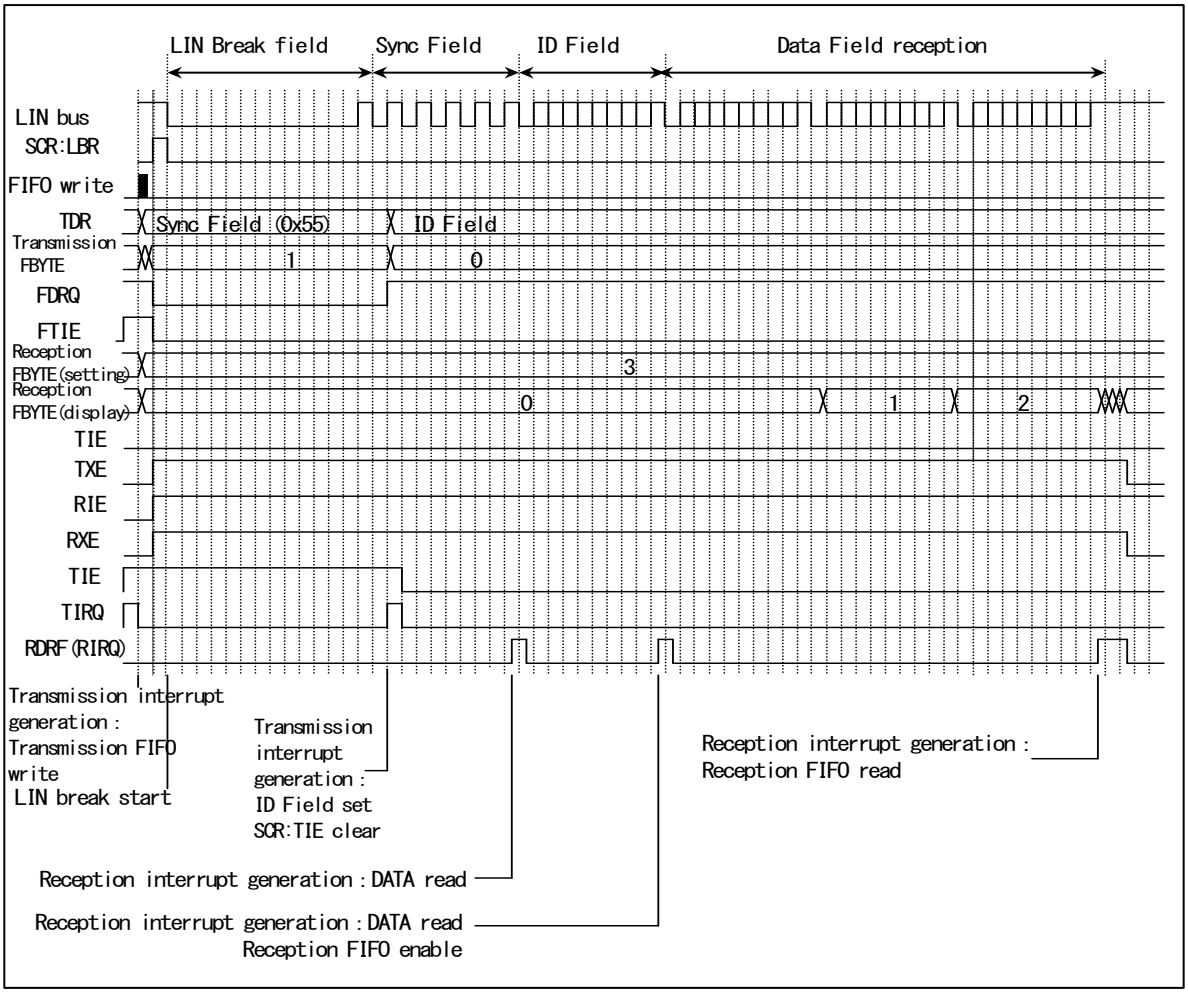


Figure 7-8 LIN Bus Timing (When a DATA Field Is Received: When a FIFO Is Used)



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 (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".

Figure 7-9 LIN Break Field Reception to Sync Field Reception (When STMR Is between SFUR and SFLR)

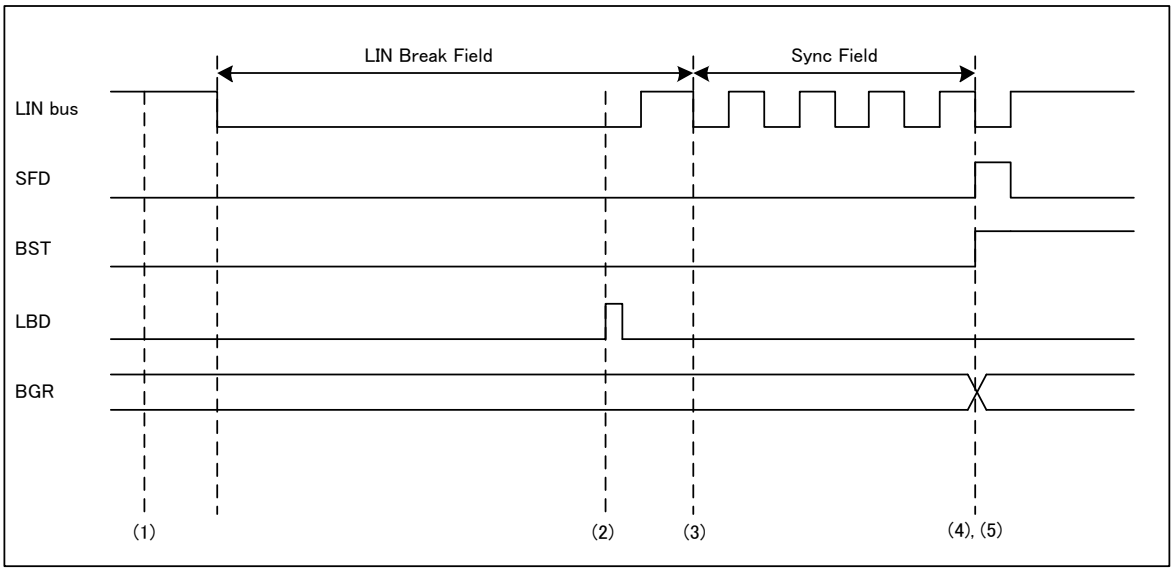
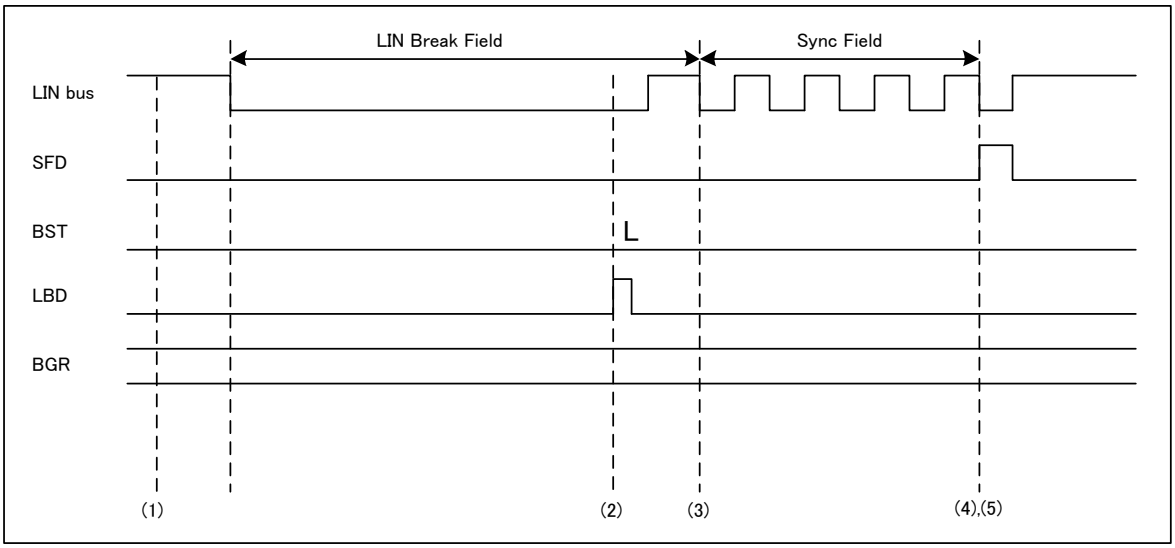


Figure 7-10 LIN Break Field Reception to Sync Field Reception (When STMR Is Not between SFUR and SFLR)



Note:

- *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),*

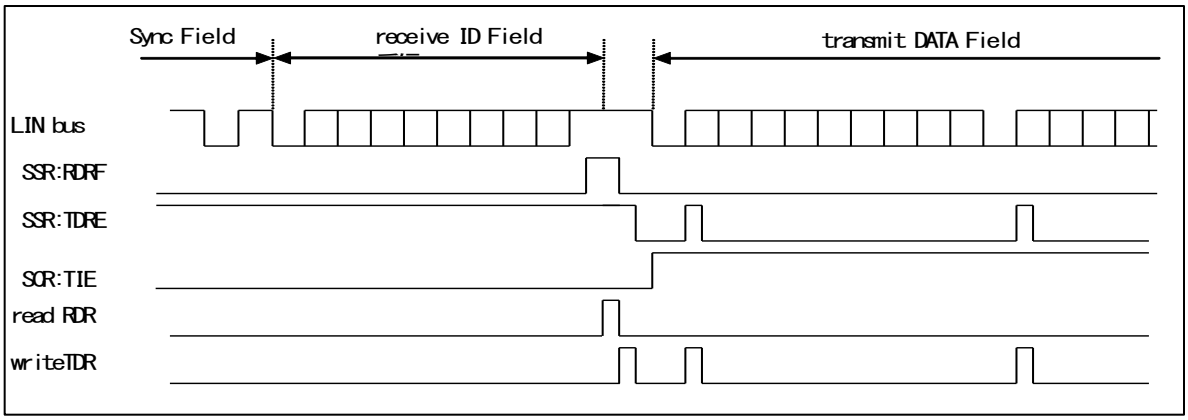
ID Field Reception to DATA Field Transmission and Reception

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).

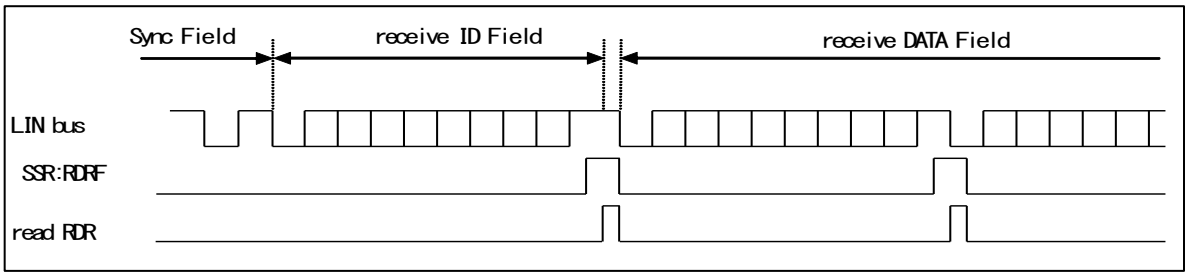
Figure 7-11 ID Field Reception to DATA Field Transmission



(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)

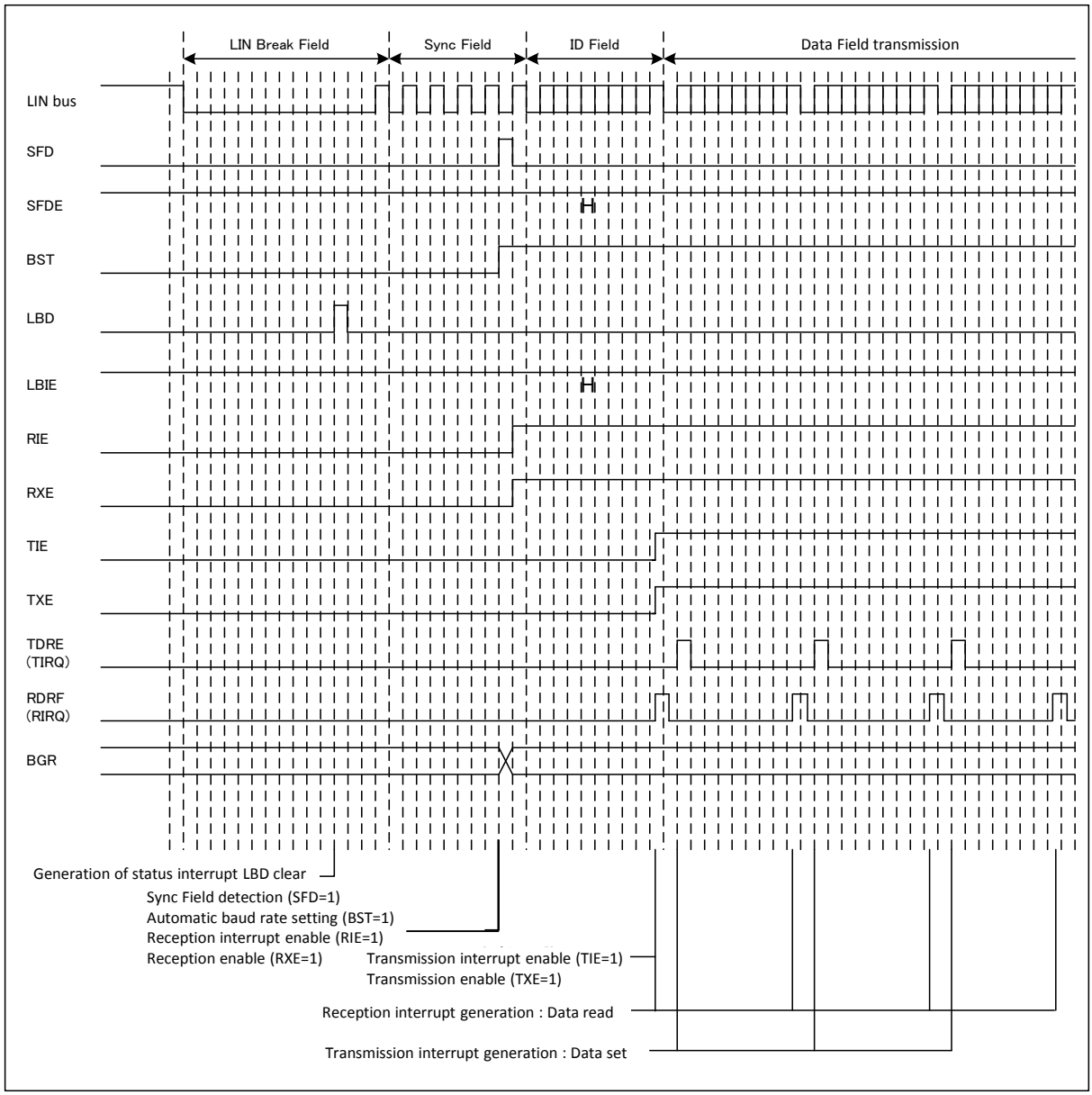
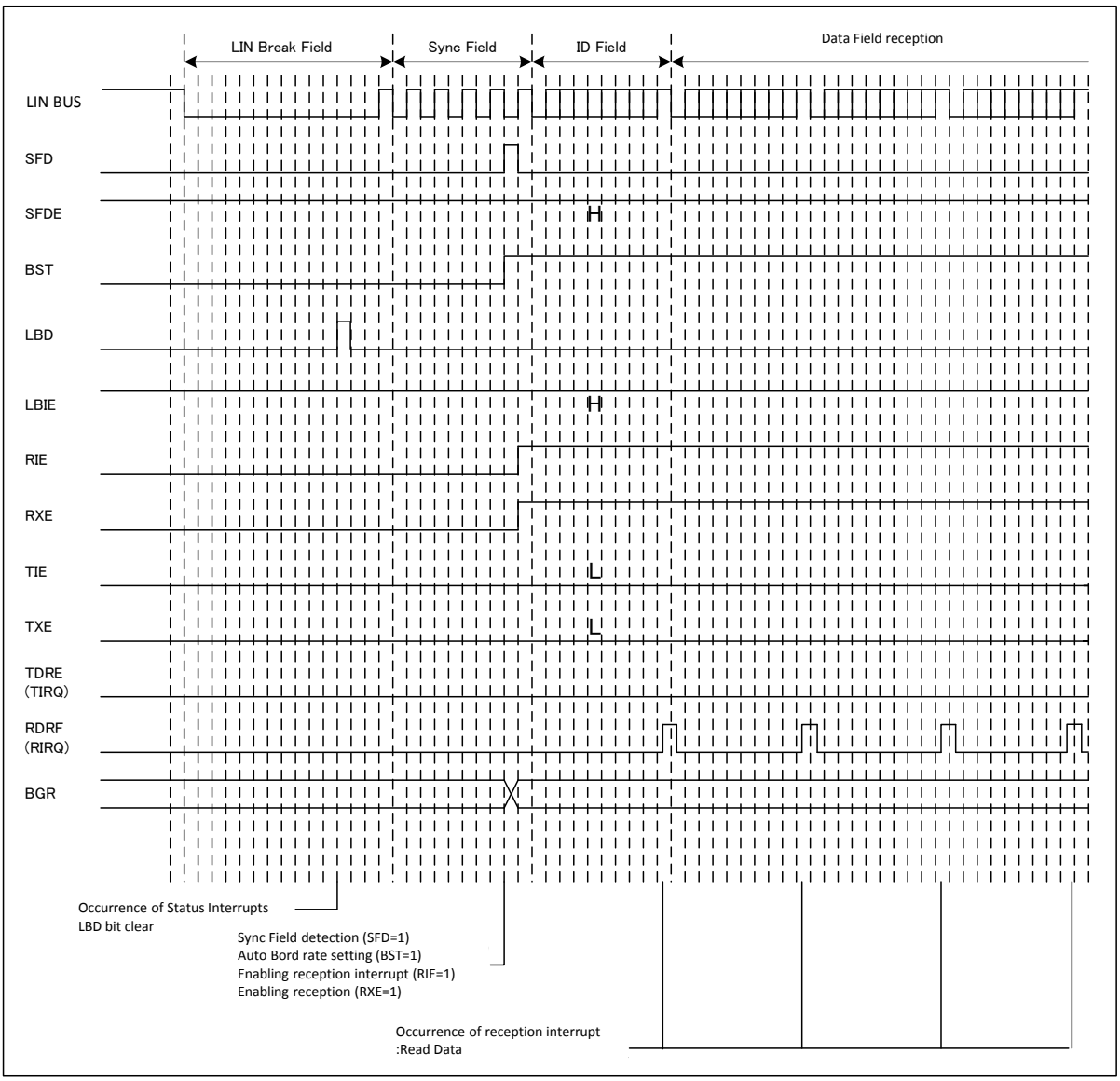


Figure 7-14 LIN Bus Timing (When the DATA Field Is Received: When a FIFO Is Not Used and SACSR:AUTE = 1)



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)

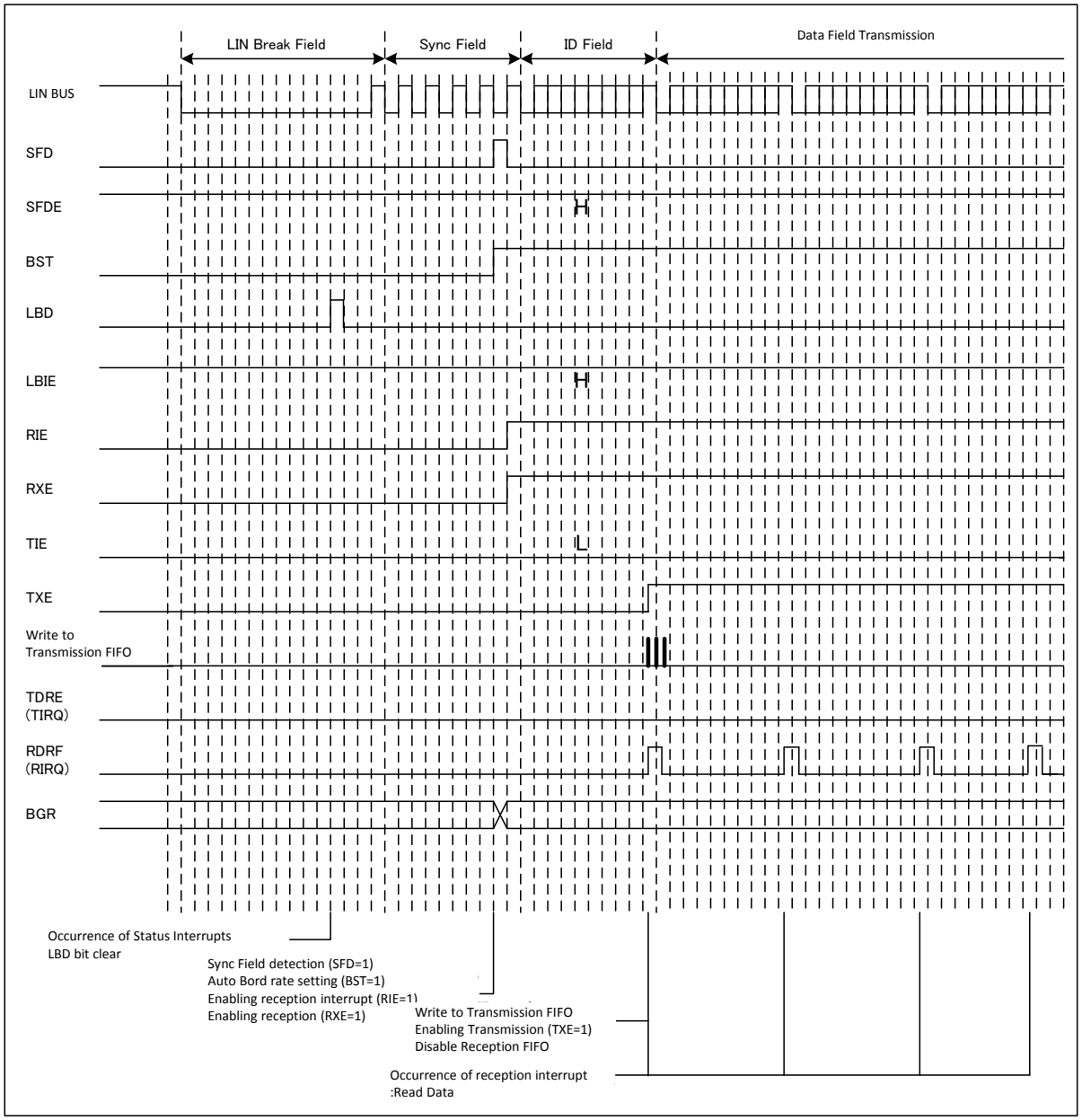
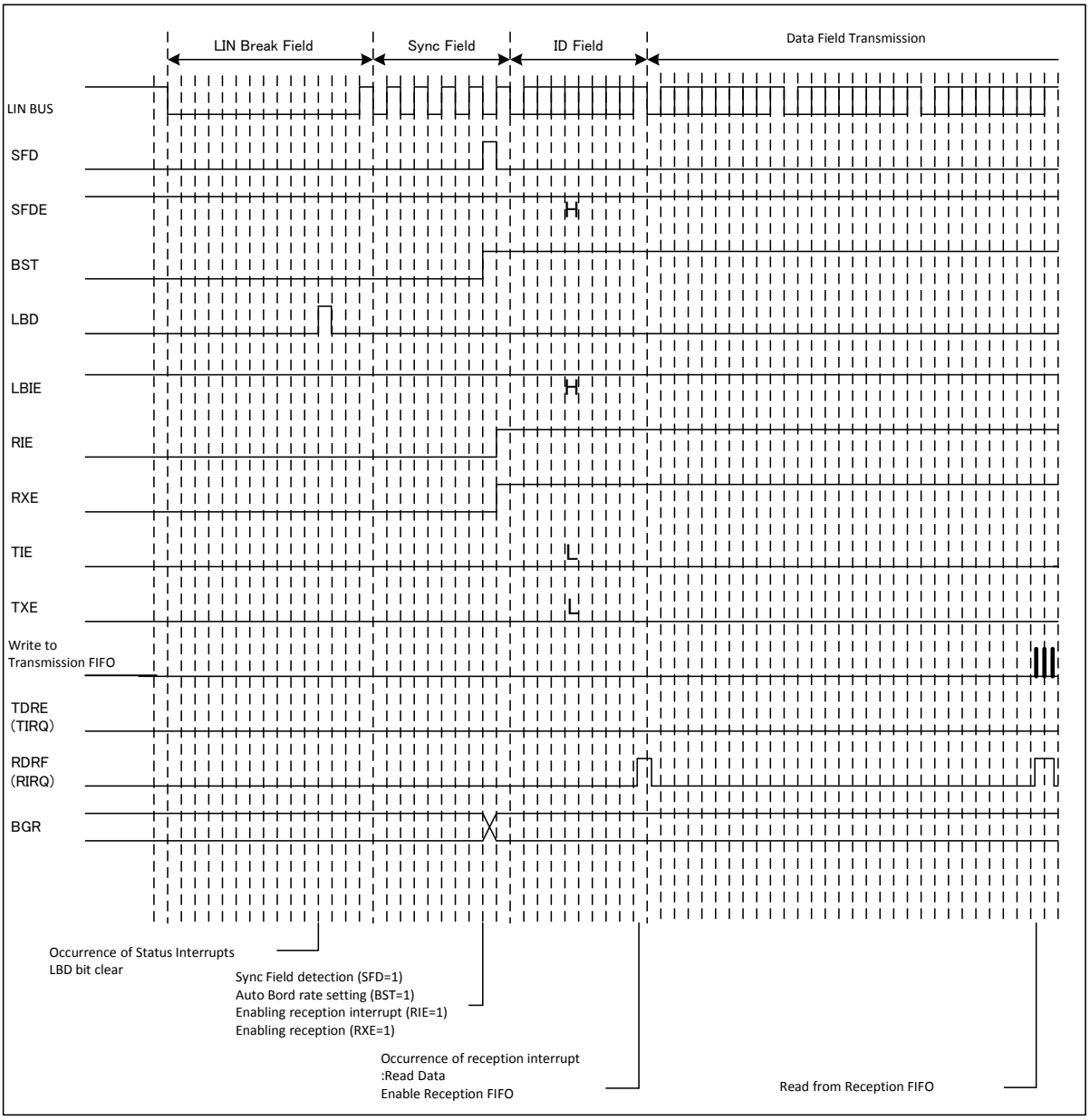


Figure 7-16 LIN Bus Timing (When DATA Field Is Received: When a FIFO Is Used and SACSR:AUTE = 1)



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

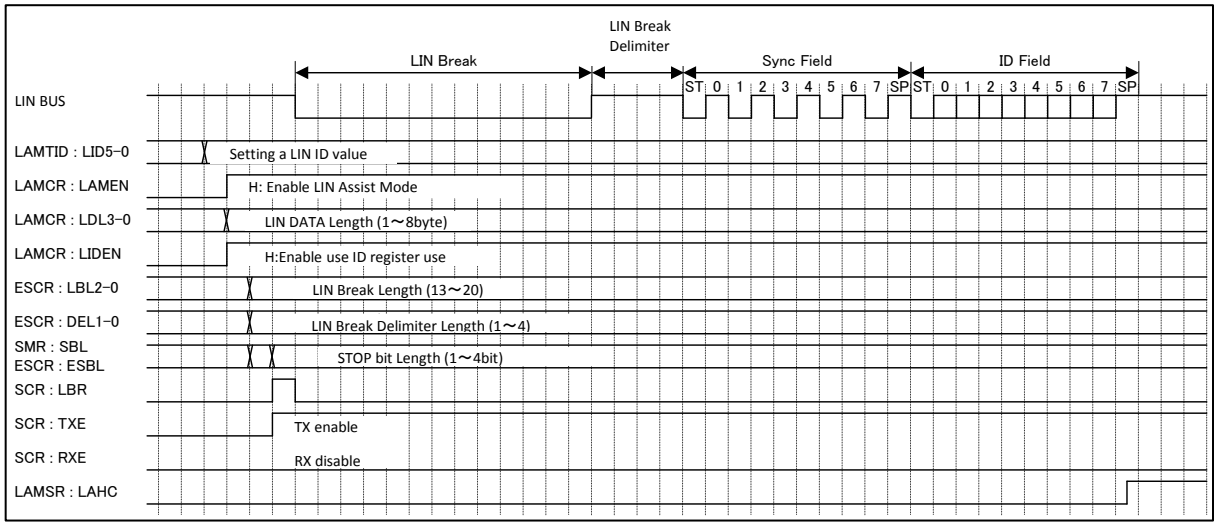
- Set the transmission operation enable bit (SCR:TXE) to "1" (transmission enabled).
- Set the LIN Break Field setting bit (SCR:LBR) to "1" (LIN Break Field generation). The LIN Break Field set at ESCR:LBL2 to LBL0 is transmitted.
- The master receives the LIN Break Field transmitted by the master, and 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 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.*
- *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:LDL[3:0]).*

Figure 7-18 ID Field Transmission to DATA Field Transmission (When ID Register Is Used and a FIFO Is Not Used)

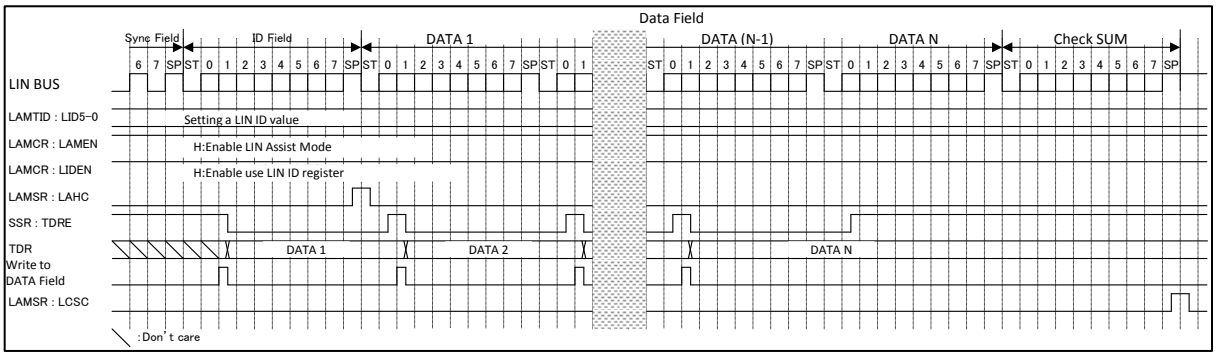
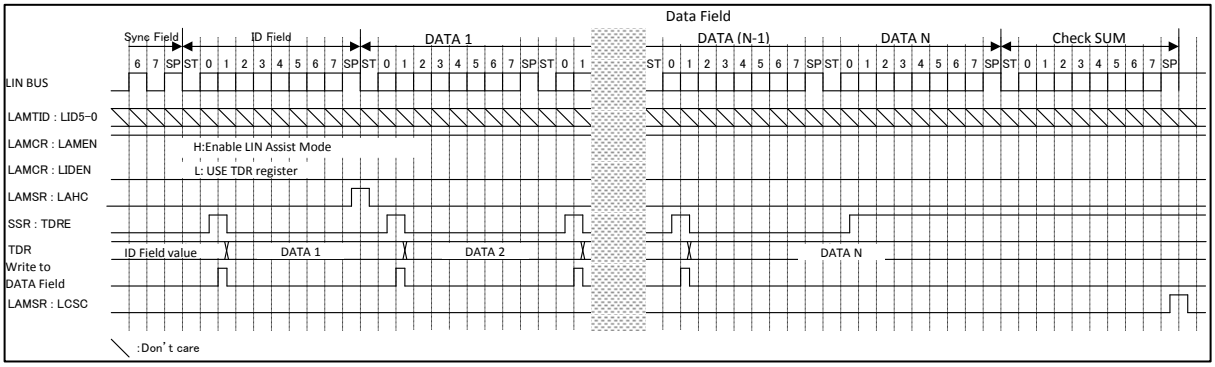


Figure 7-19 ID Field Transmission to DATA Field Transmission (When ID Register Is Not Used and a FIFO Is Not Used)



(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 (SSR: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:LDL[3:0]).*

Figure 7-20 ID Field Transmission to DATA Field Reception (When the ID Register Is Used and a FIFO Is Not Used)

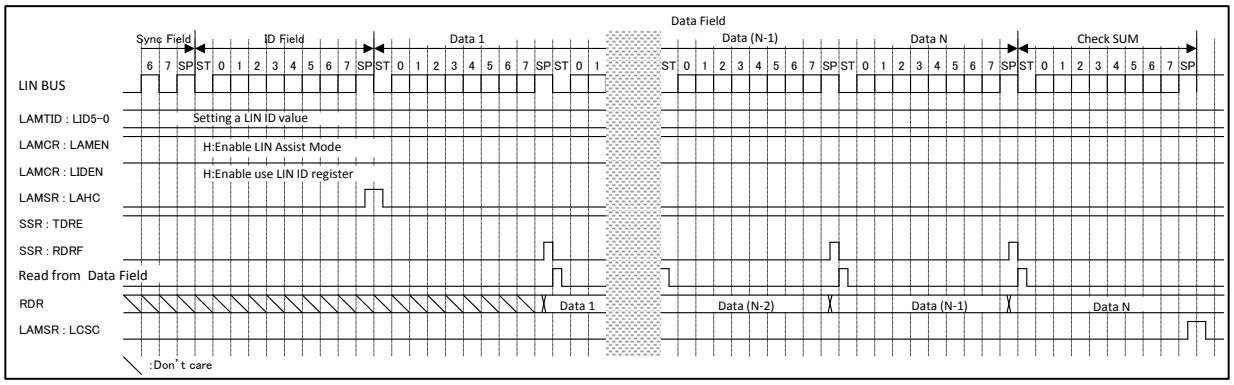
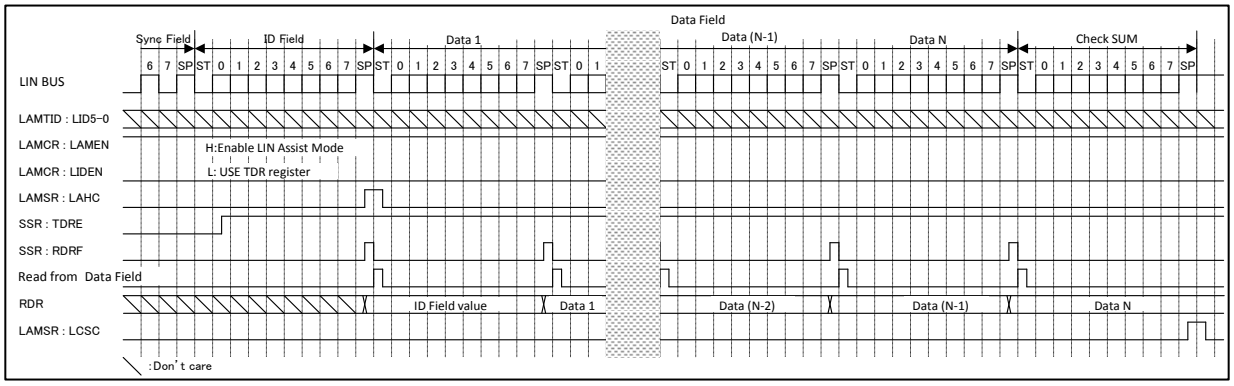


Figure 7-21 ID Field Transmission to DATA Field Reception (When the ID Register Is Not Used and a FIFO Is Not Used)



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)

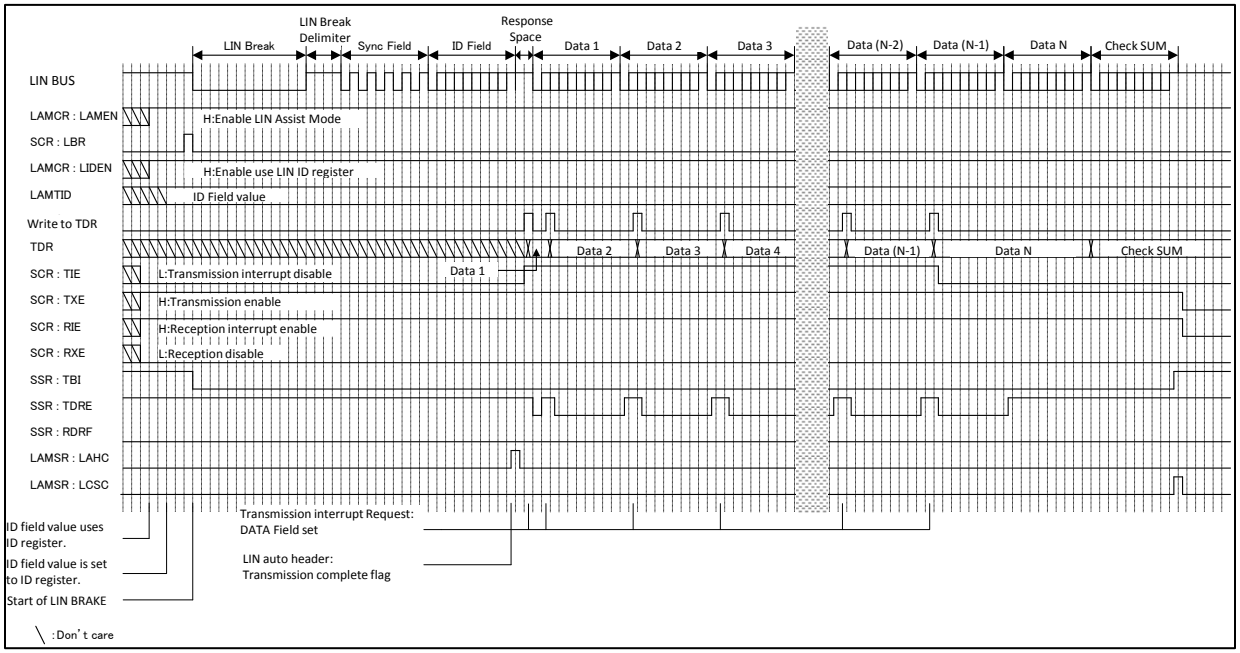


Figure 7-23 LIN Bus Timing (the ID Register Is Not Used, the DATA Field Is Transmitted, and a FIFO Is Not Used)

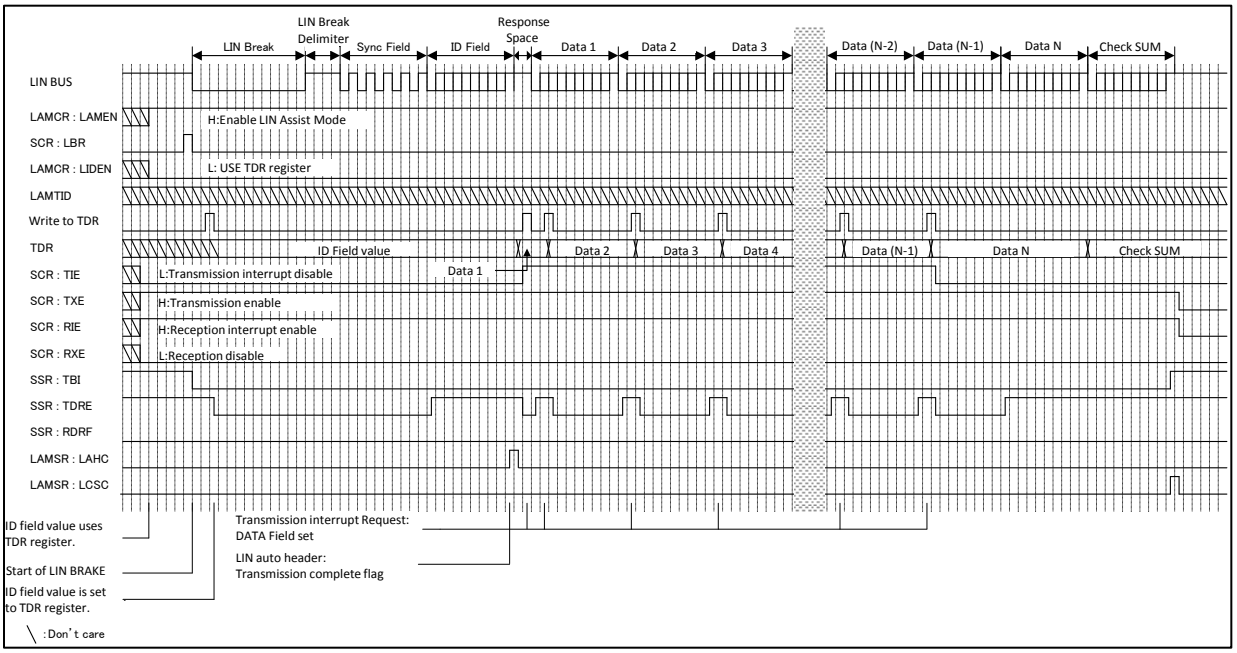


Figure 7-24 LIN Bus Timing (the ID Register Is Used, the DATA Field Is Received, and a FIFO Is Not Used)

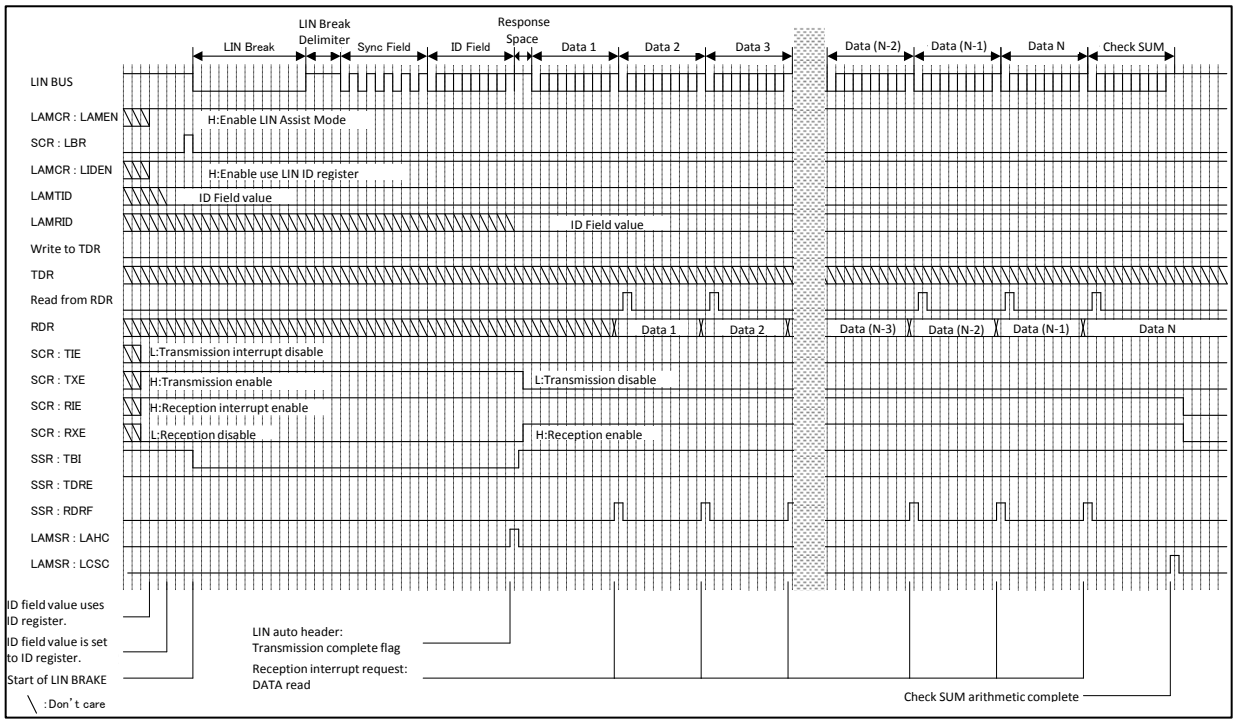


Figure 7-25 LIN Bus Timing (the ID Register Is Not Used, the DATA Field Is Received, and a FIFO Is Not Used)

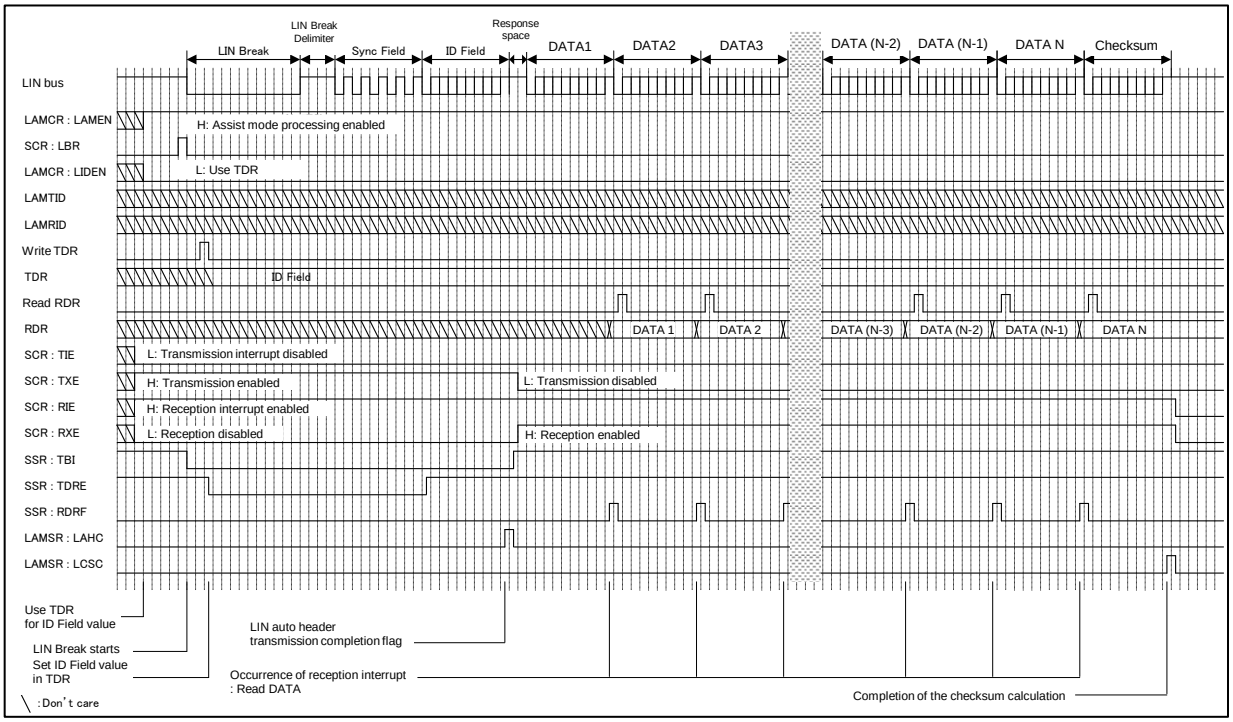


Figure 7-26 LIN Bus Timing (the ID Register Is Used, the DATA Field Is Transmitted, and a FIFO Is Used)

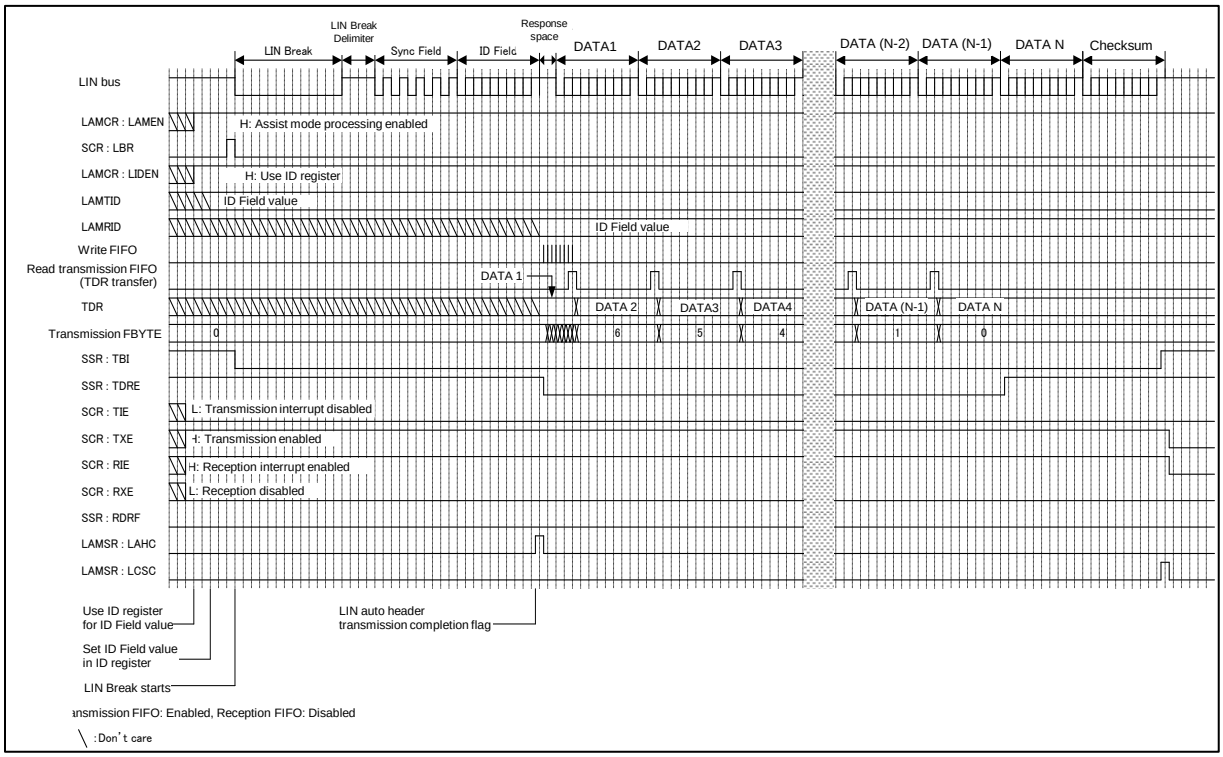


Figure 7-27 LIN Bus Timing (the ID Register Is Not Used, the DATA Field Is Transmitted, and a FIFO Is Used)

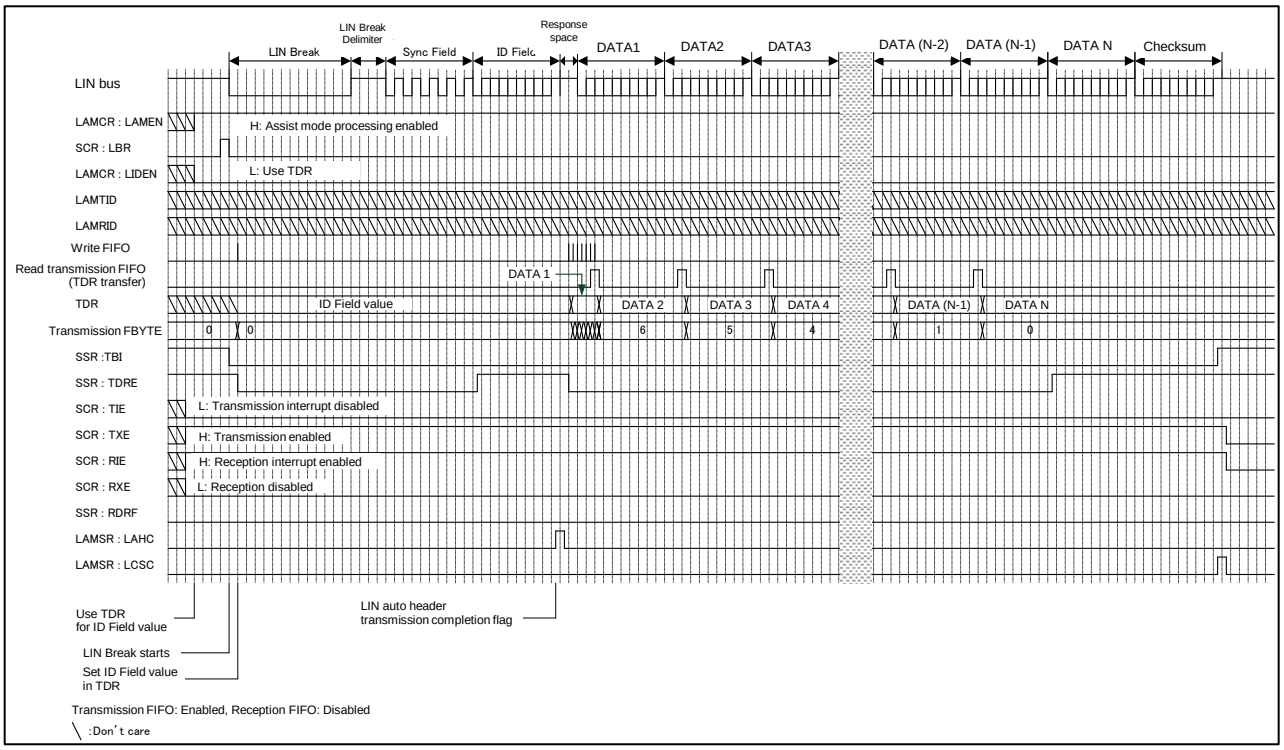


Figure 7-28 LIN Bus Timing (the ID Register Is Used, the DATA Field Is Received, and a FIFO Is Used)

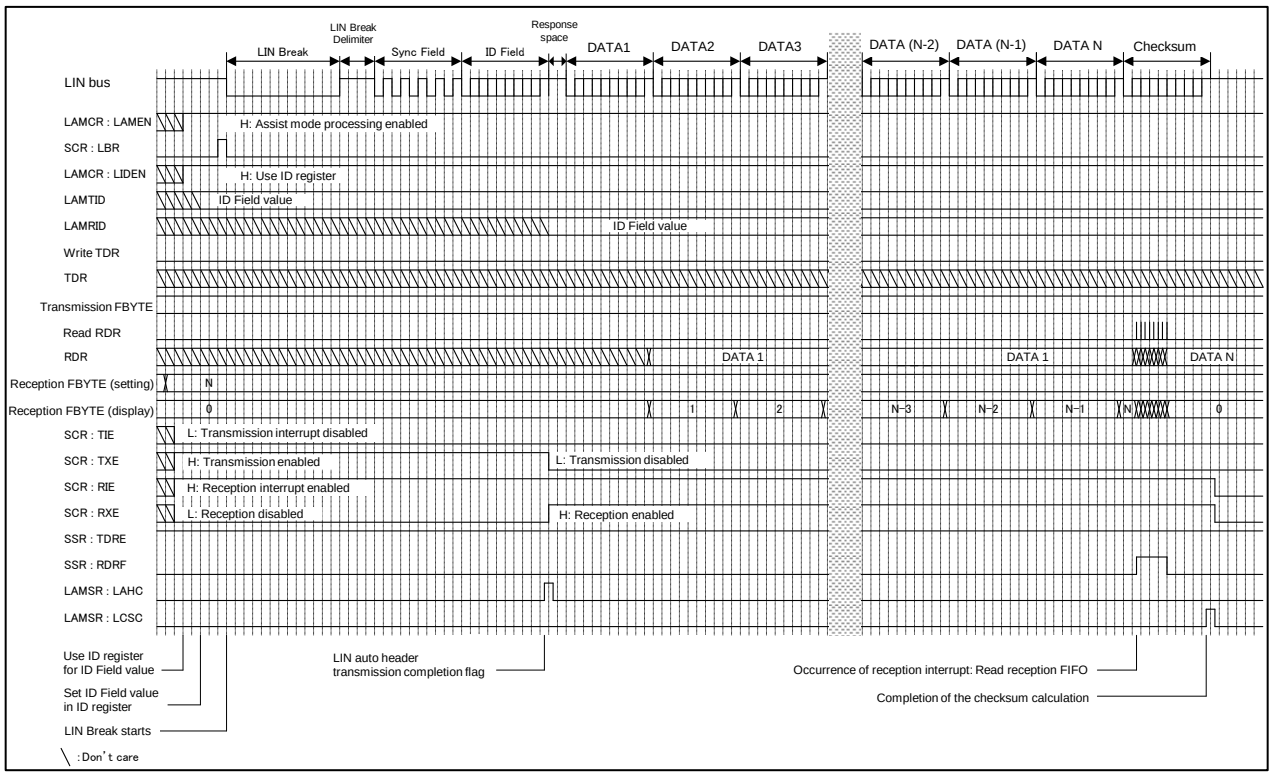
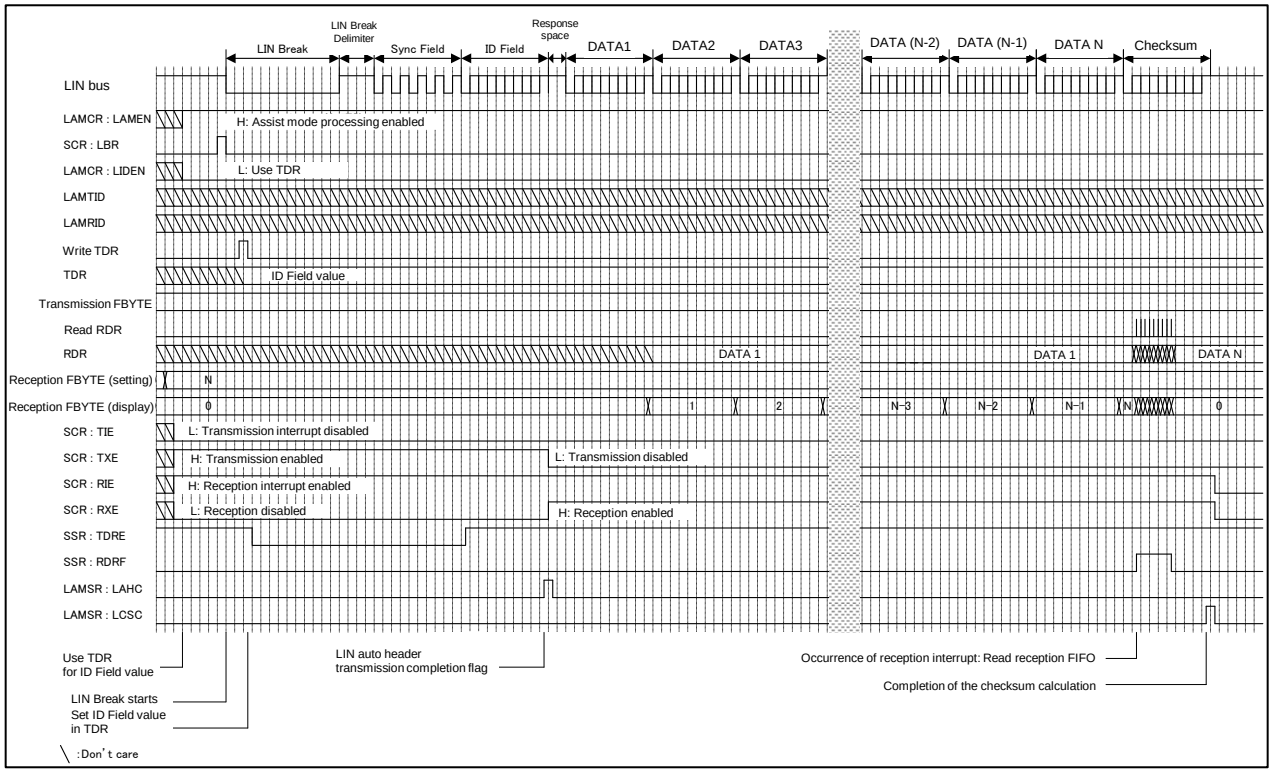


Figure 7-29 LIN Bus Timing (the ID Register Is Not Used, the DATA Field Is Received, and a FIFO Is Used)



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 (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".

Figure 7-30 LIN Break Field to ID Field Reception (When STMR Is between SFUR and SFLR)

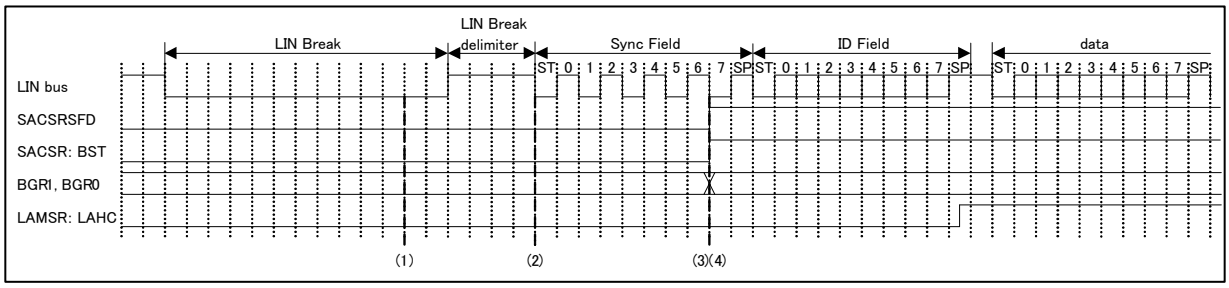
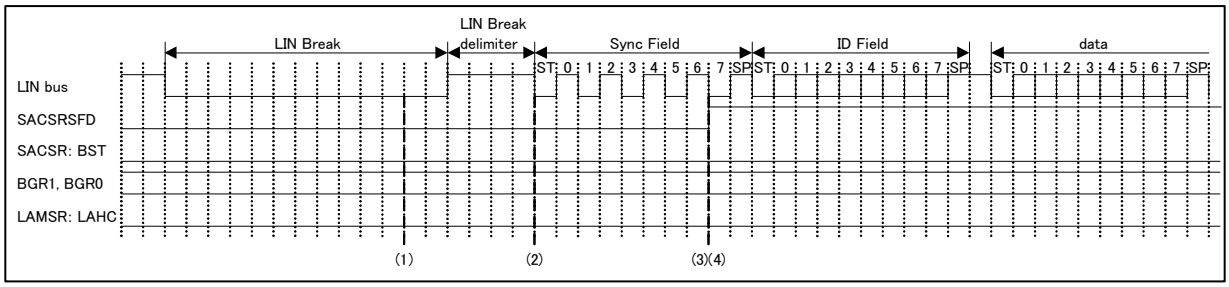


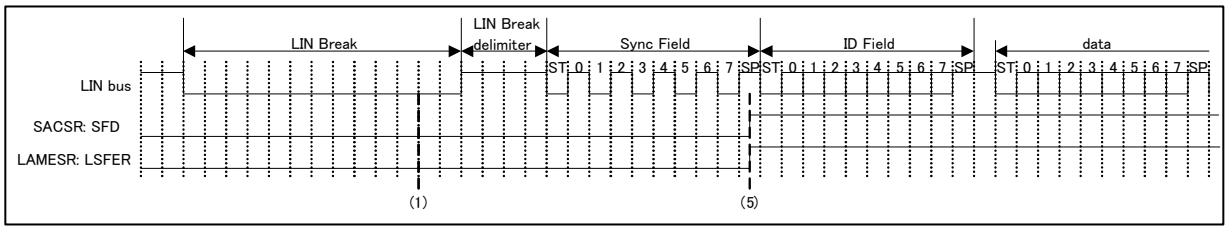
Figure 7-31 LIN Break Field to ID Field Reception (When STMR Is Not between SFUR and SFLR)



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)

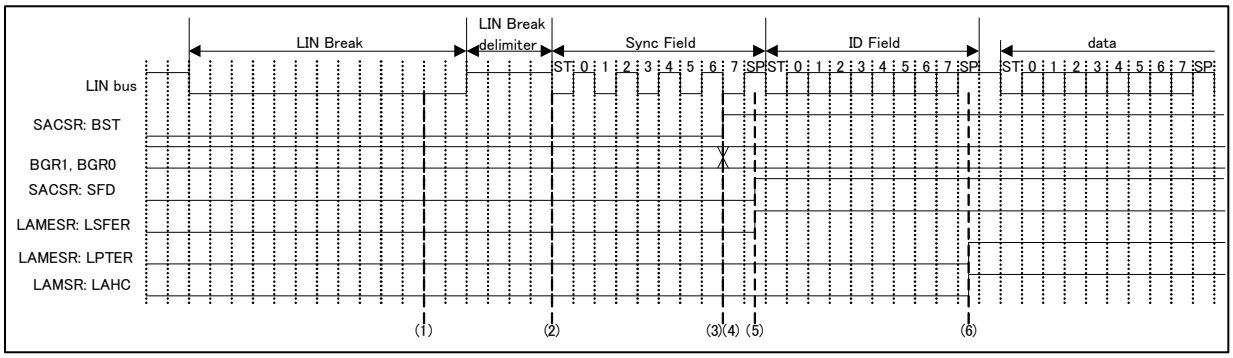
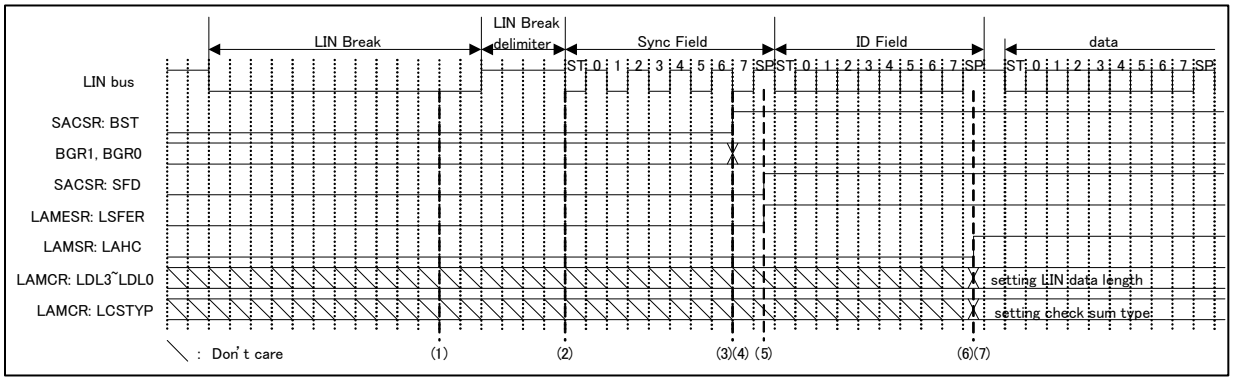


Figure 7-34 LIN Break Field to ID Field Reception (When the LIN Data Length Is Set)



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)

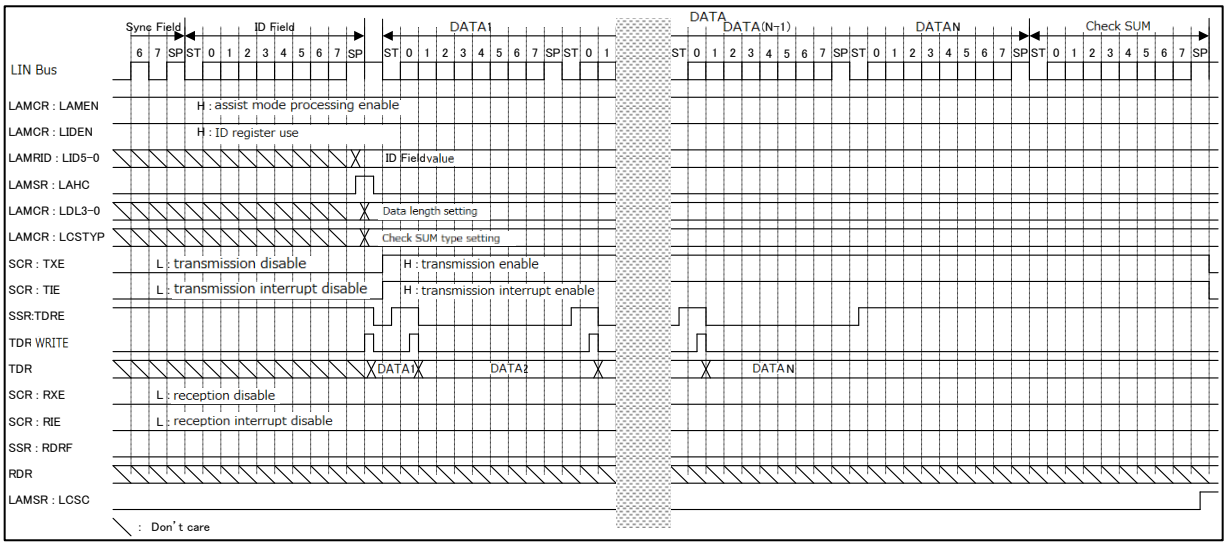
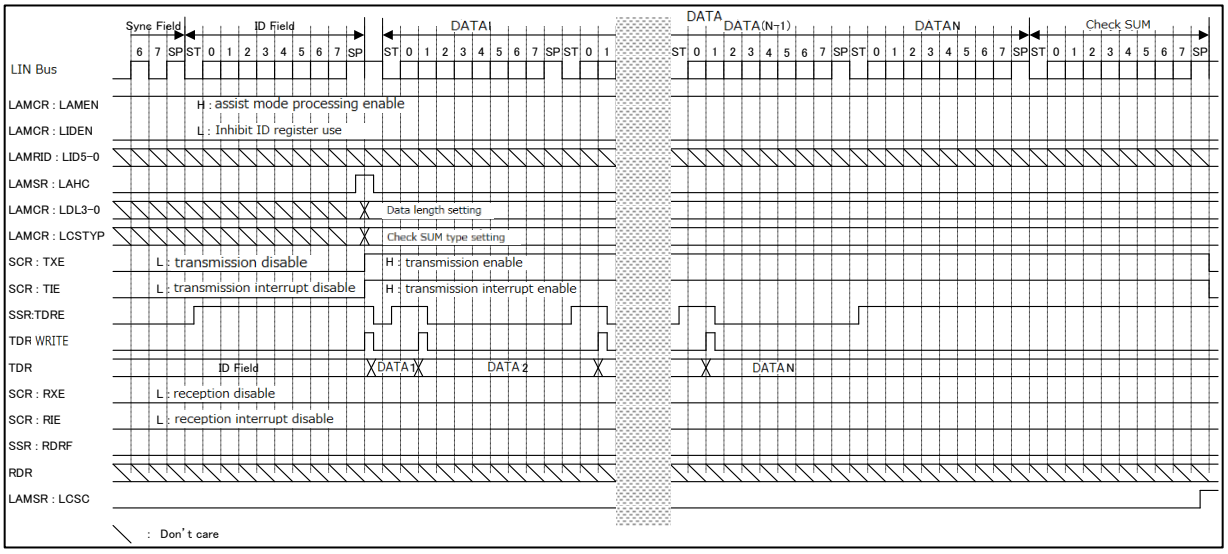


Figure 7-36 ID Field Reception to DATA Field Transmission (When the ID Register Is Not Used)



(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: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.)*

Figure 7-37 ID Field Reception to DATA Field Reception (When the ID Register Is Used)

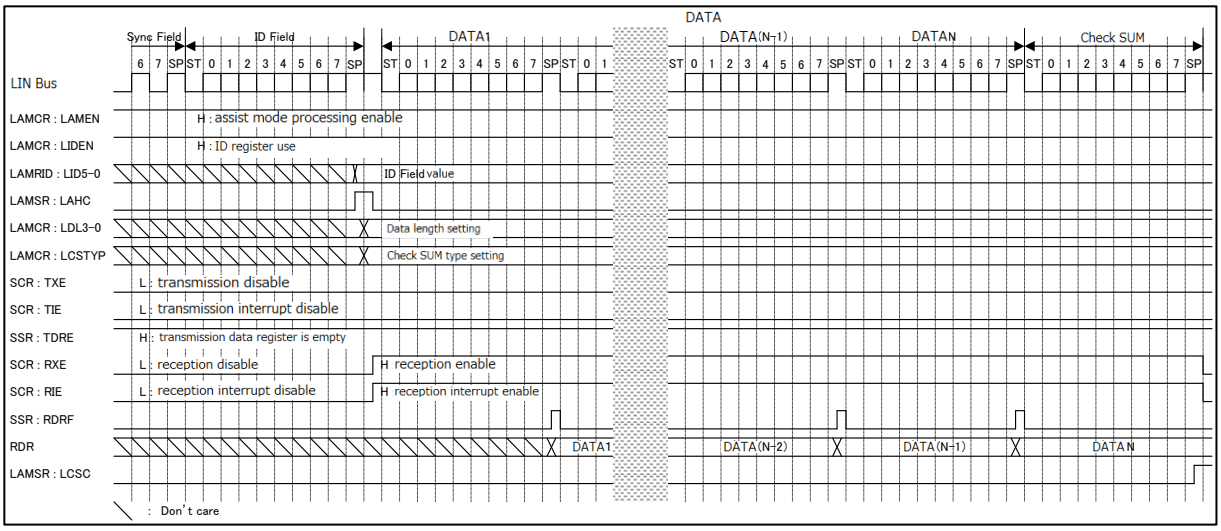
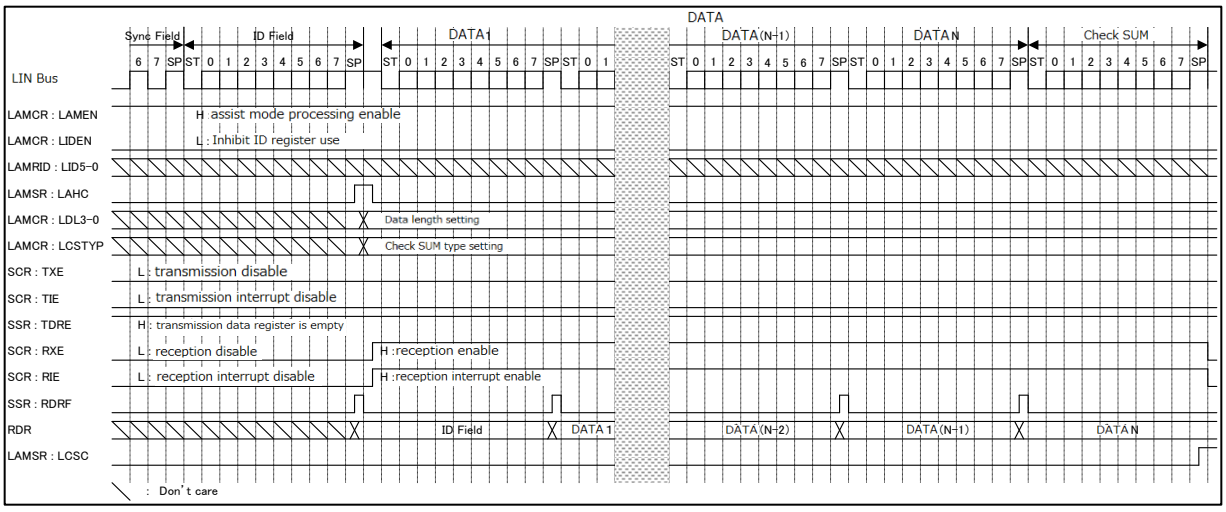


Figure 7-38 ID Field Reception to DATA Field Reception (When the ID Register Is Not Used)



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)

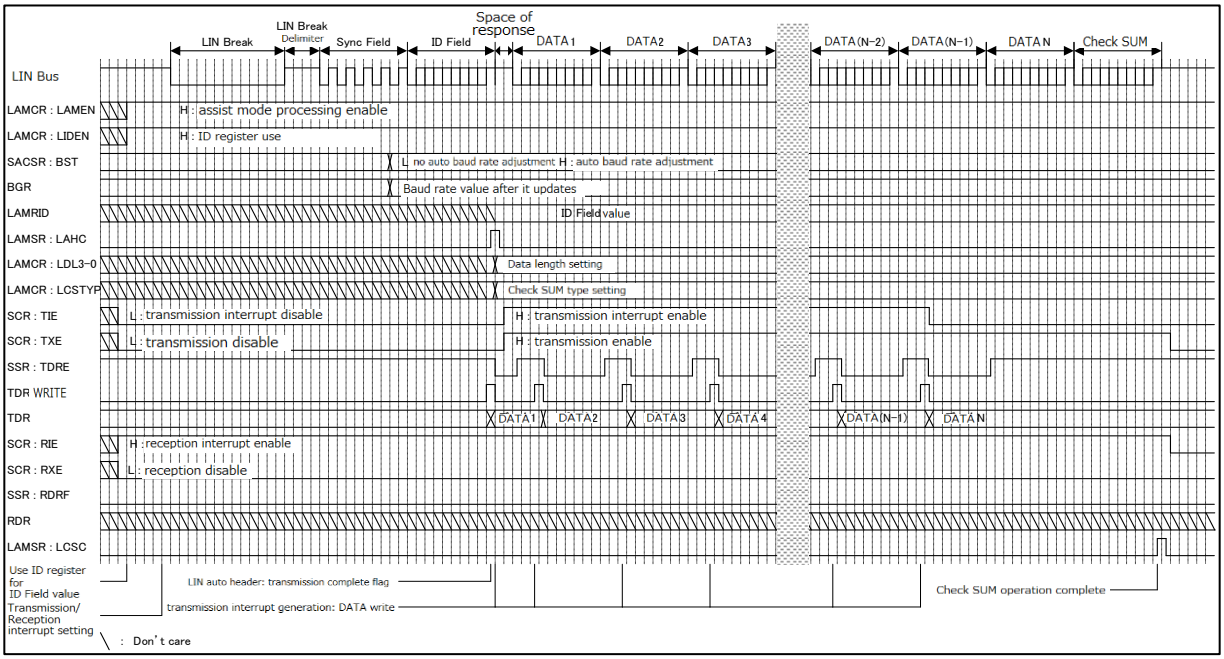


Figure 7-40 LIN Bus Timing (When the DATA Field Is Transmitted: a FIFO Is Not Used, SACSR:AUTE = 1, and the ID Register Is Not Used)

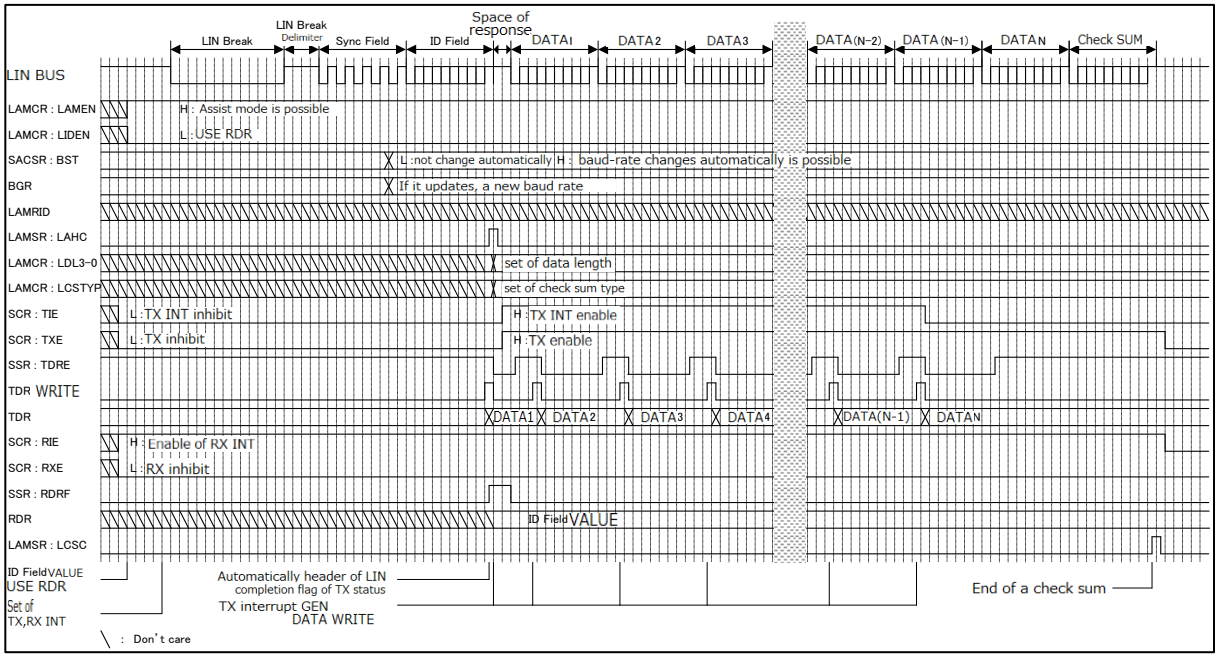


Figure 7-41 LIN Bus Timing (When the DATA Field Is Received: a FIFO Is Not Used, SACSR:AUTE = 1, and an ID Register Is Used)

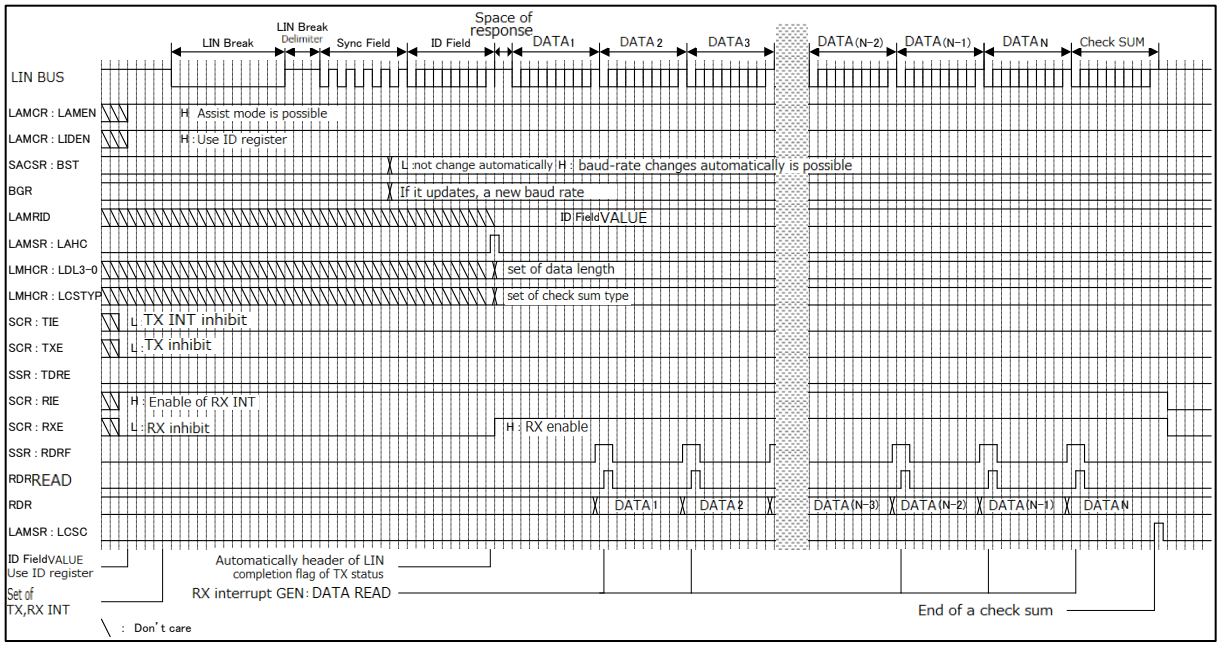


Figure 7-42 LIN Bus Timing (When the DATA Field Is Received: a FIFO Is Not Used, SACSR:AUTE = 1, and an ID Register Is Not Used)

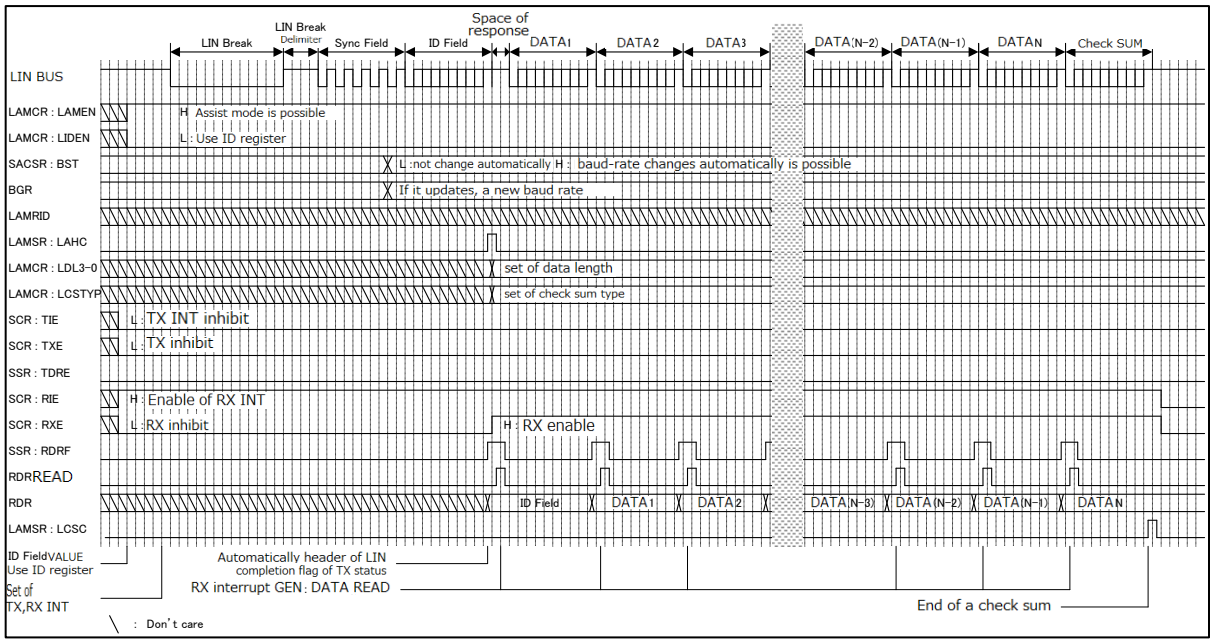


Figure 7-43 LIN Bus Timing (When the DATA Field Is Transmitted: a FIFO Is Used, SACSR:AUTE = 1, and the ID Register Is Used)

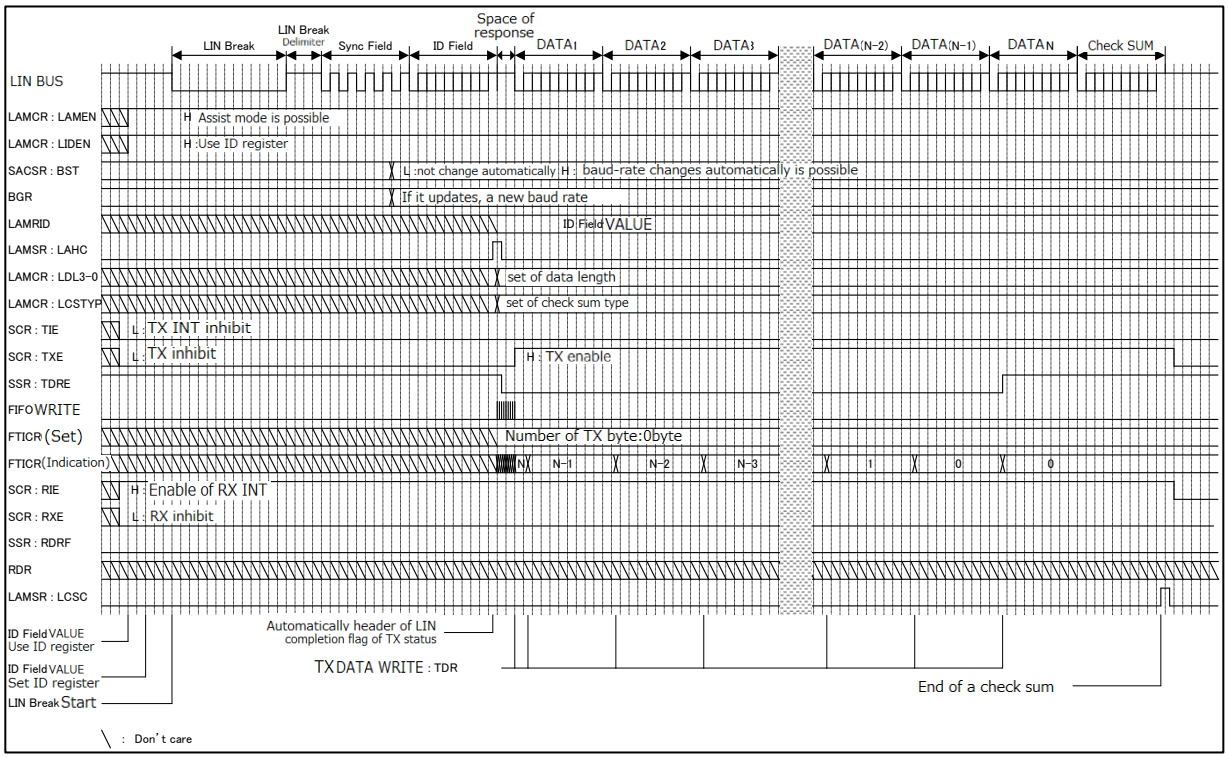


Figure 7-44 LIN Bus Timing (When the DATA Field Is Transmitted: a FIFO Is Used, SACSR:AUTE = 1, and the ID Register Is Not Used)

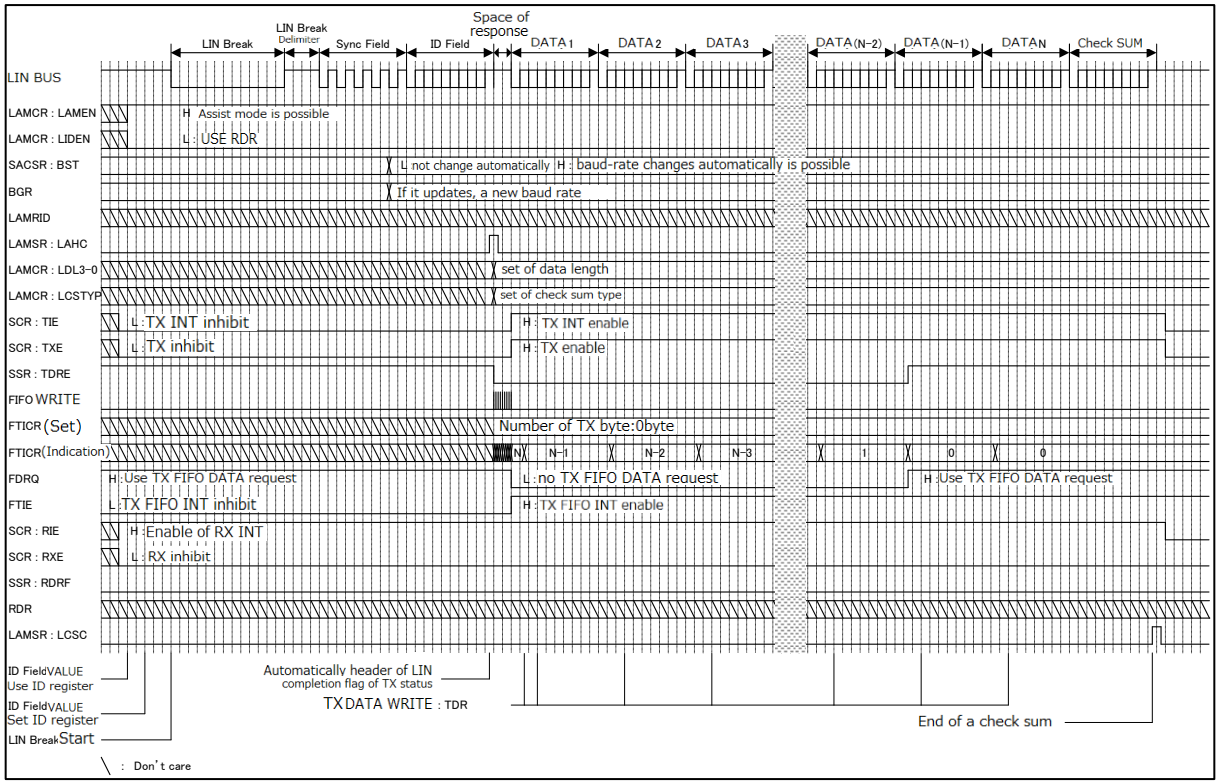


Figure 7-45 LIN Bus Timing (When the DATA Field Is Received: a FIFO Is Used, SACSR:AUTE = 1, and the ID Register Is Used)

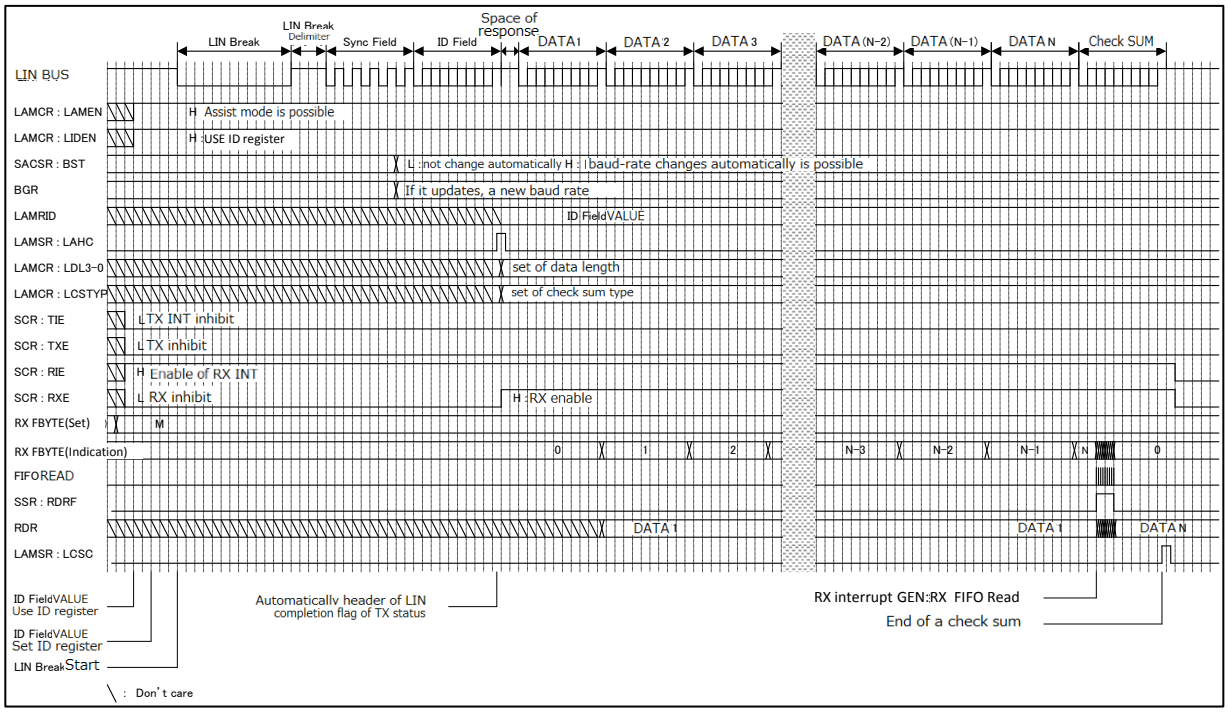
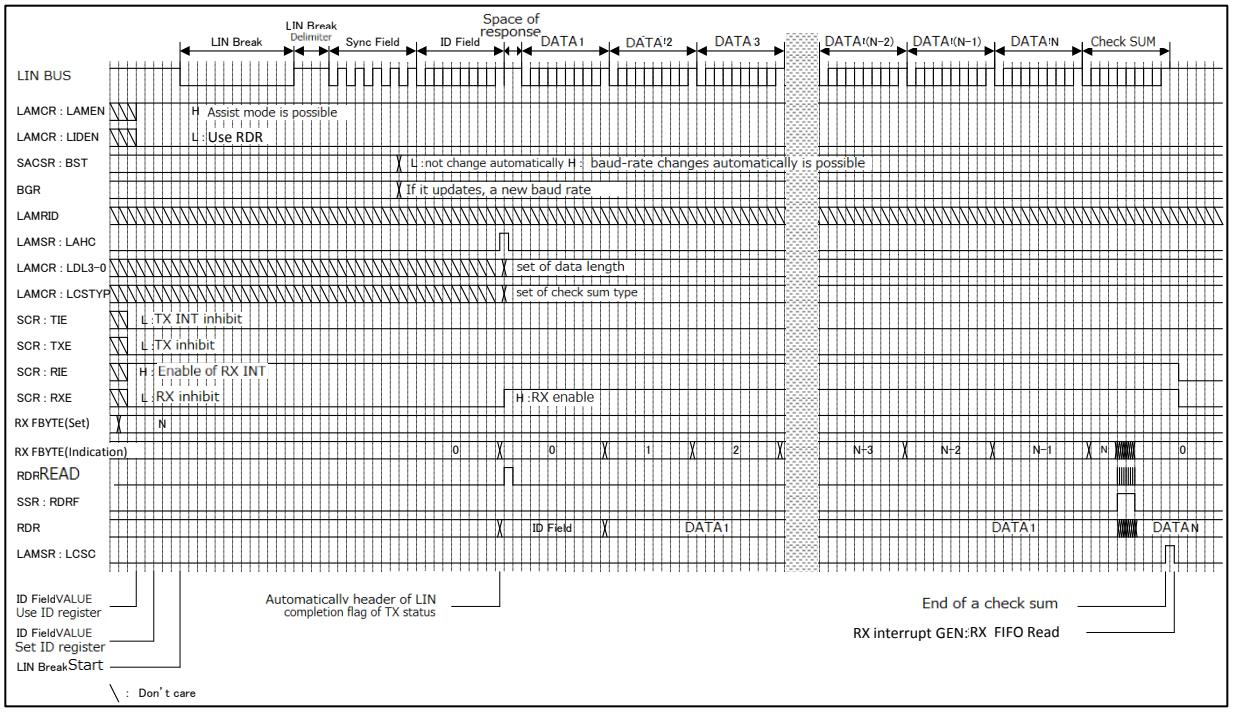


Figure 7-46 LIN Bus Timing (When the DATA Field Is Received: a FIFO Is Used, SACSR:AUTE = 1, and the ID Register Is Not Used)



8. Setup Procedure and Program Flow for Operation Mode 3 (LIN Communication Mode)

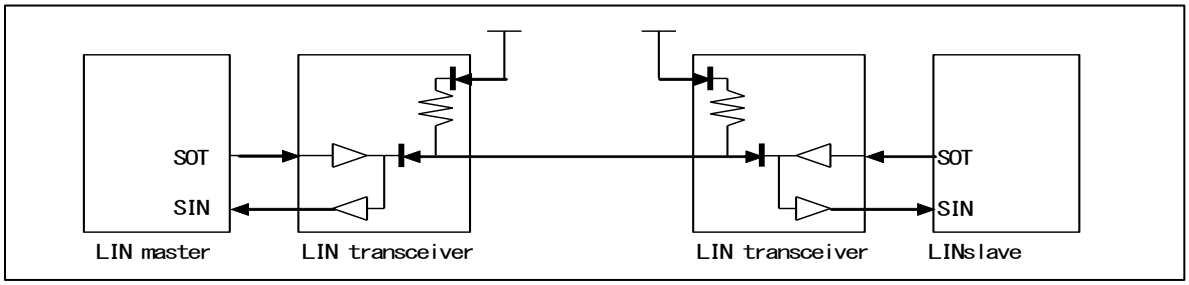
Operation mode 3 (LIN communication mode) can be used for either the LIN master system or the LIN slave system.

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)

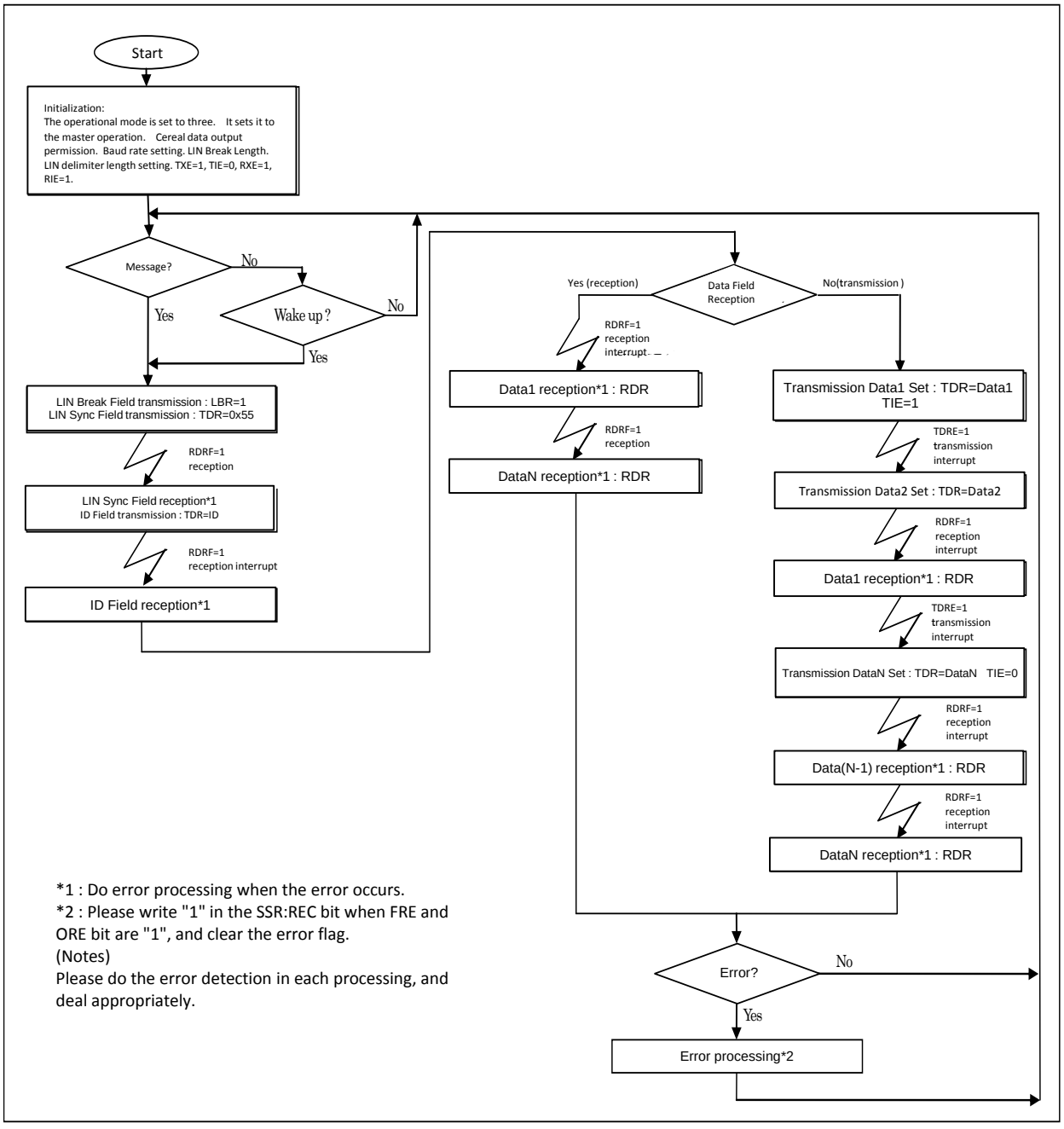
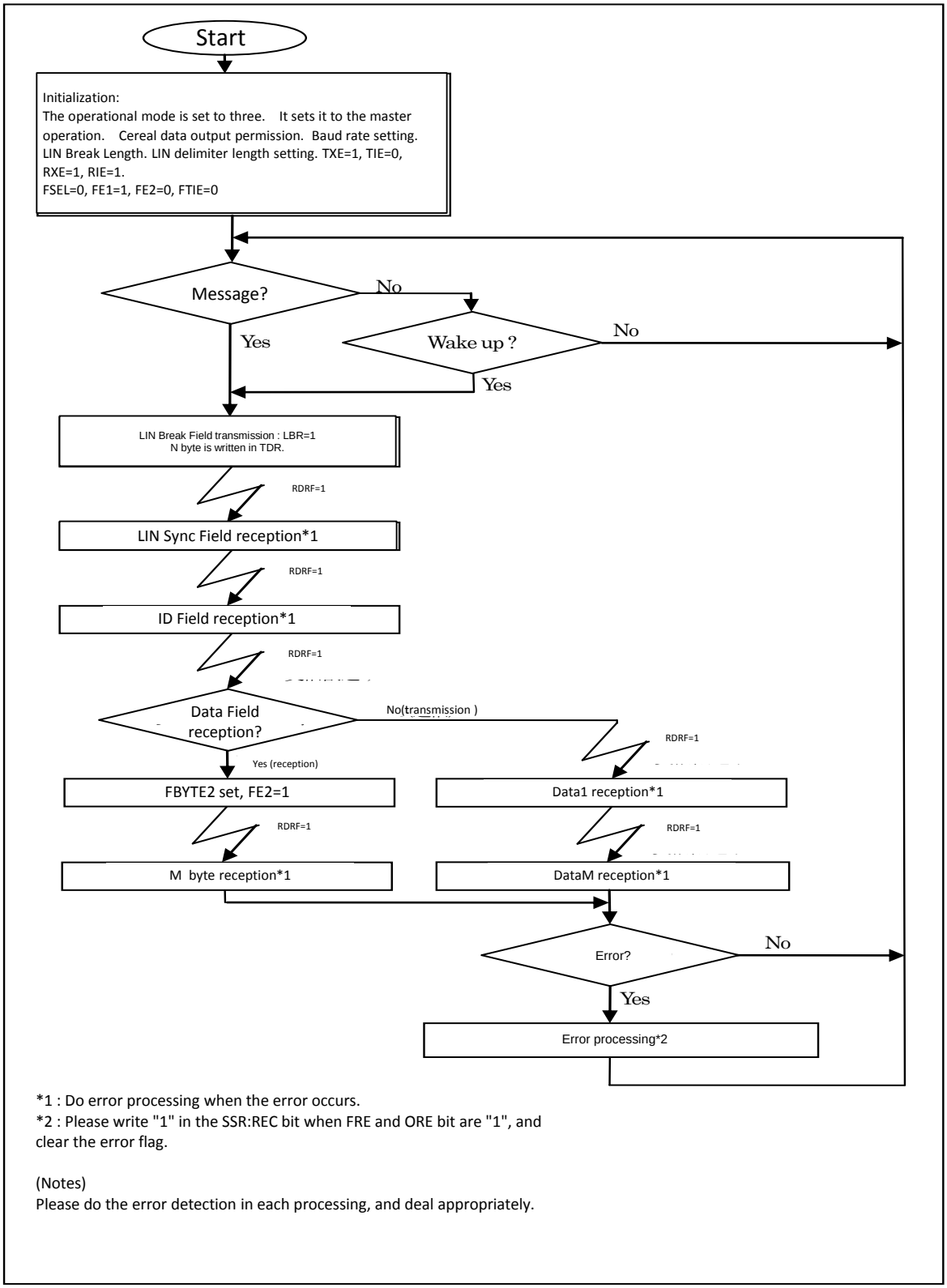


Figure 8-3 Example of LIN Communication Master Mode (a FIFO Is Used)

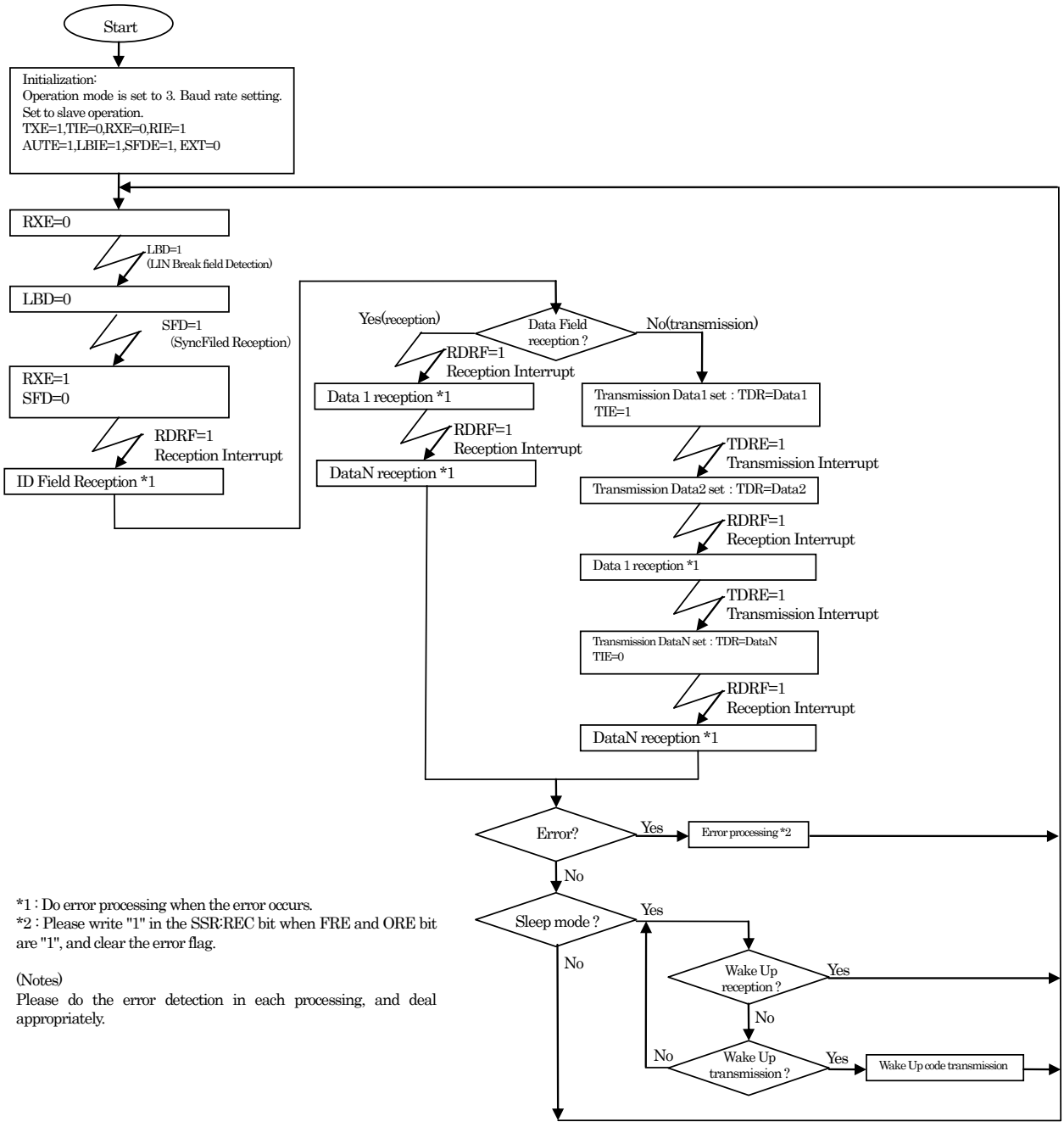


*1 : Do error processing when the error occurs.
*2 : Please write "1" in the SSR:REC bit when FRE and ORE bit are "1", and clear the error flag.

(Notes)
Please do the error detection in each processing, and deal appropriately.

Slave Operation

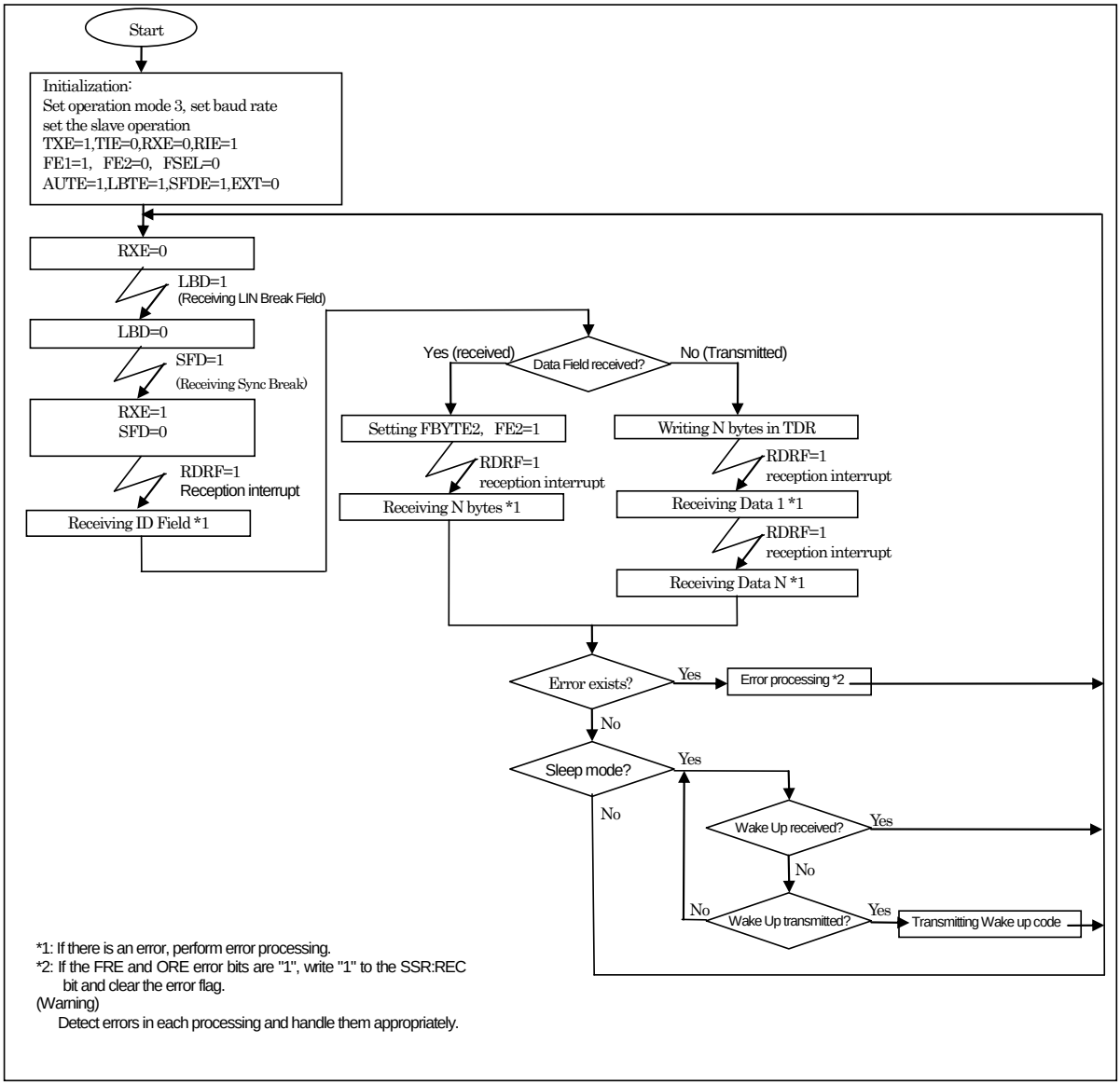
Figure 8-4 Example of LIN Communication Slave Mode (FIFO Is Not Used, Auto Baud Rate Adjustment is Allowed (SACSR:AUTE = 1))



*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)

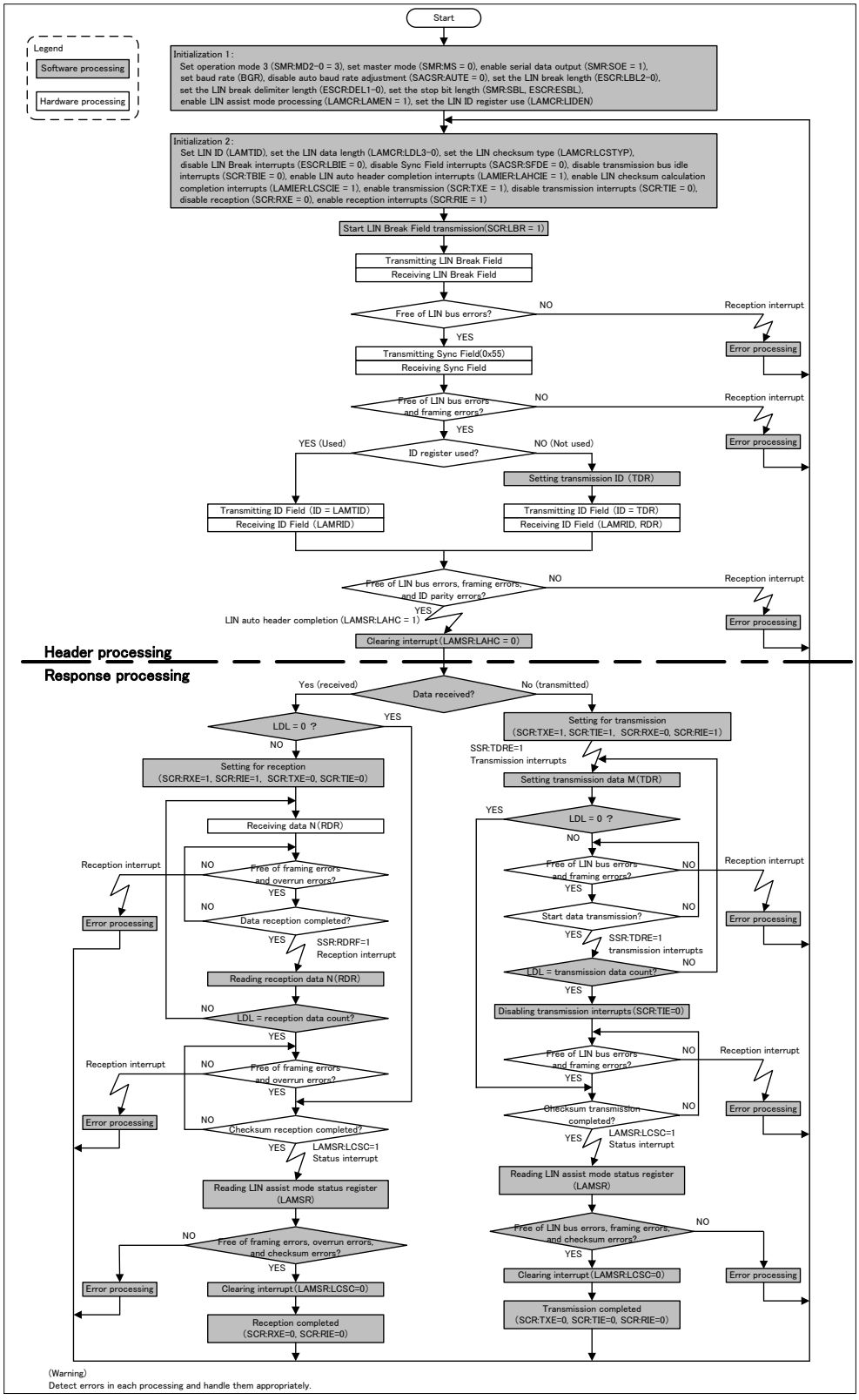
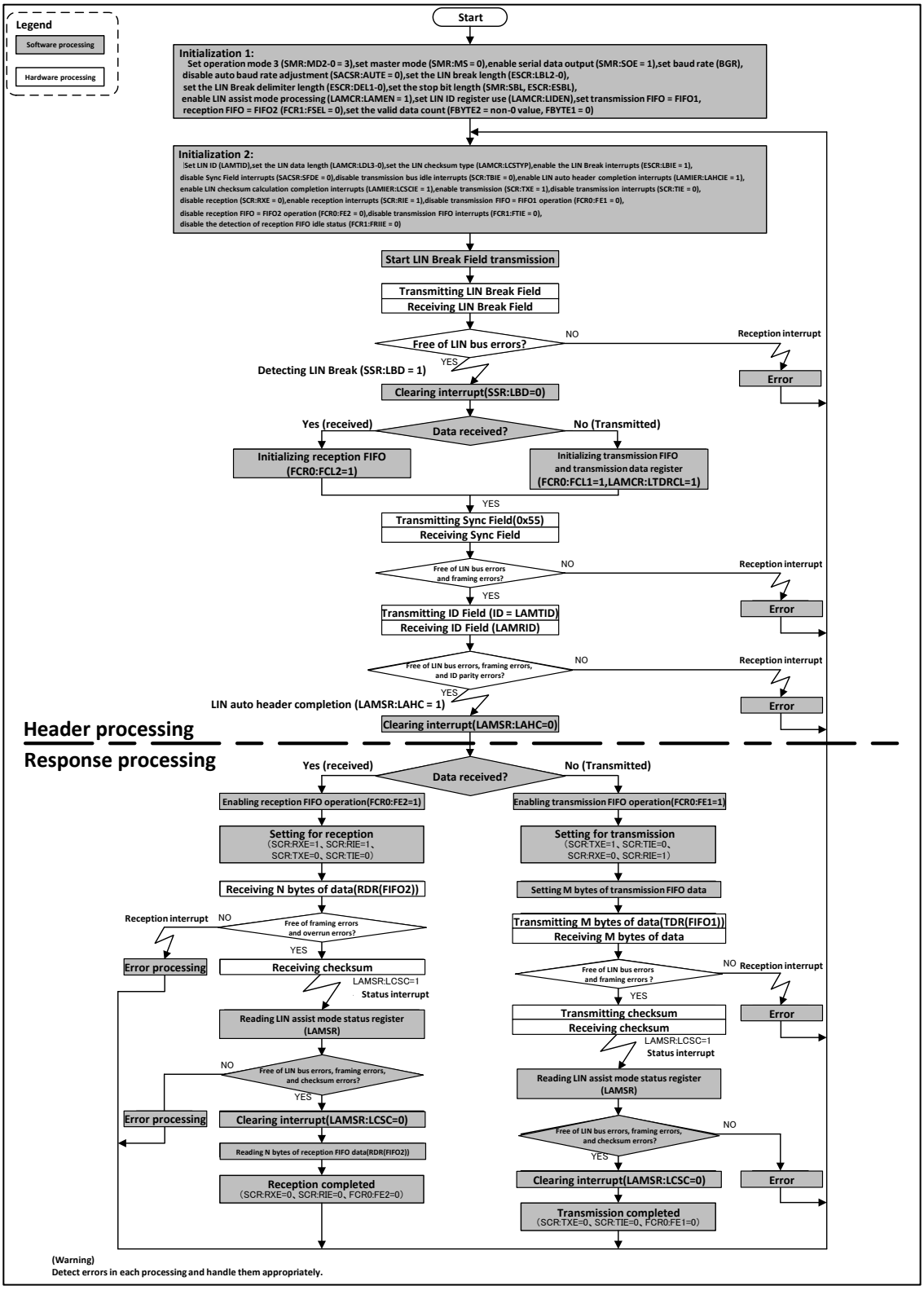


Figure 8-7 Example of LIN Communication Master Mode (Assist Mode/a FIFO Is Used)[*]



Slave Operation

Figure 8-8 Example of LIN Communication Slave Mode (Assist Mode/a FIFO Is Not Used)

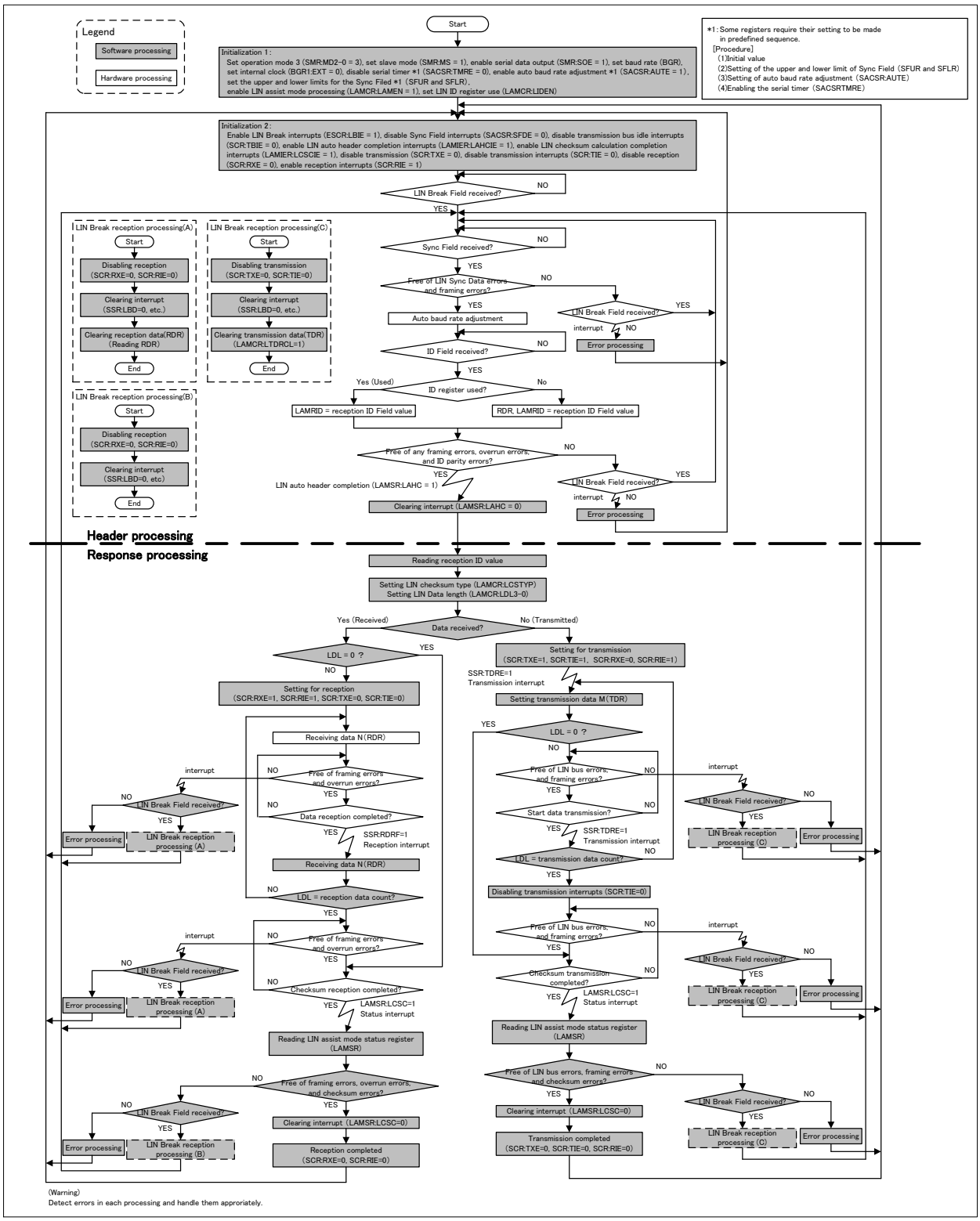
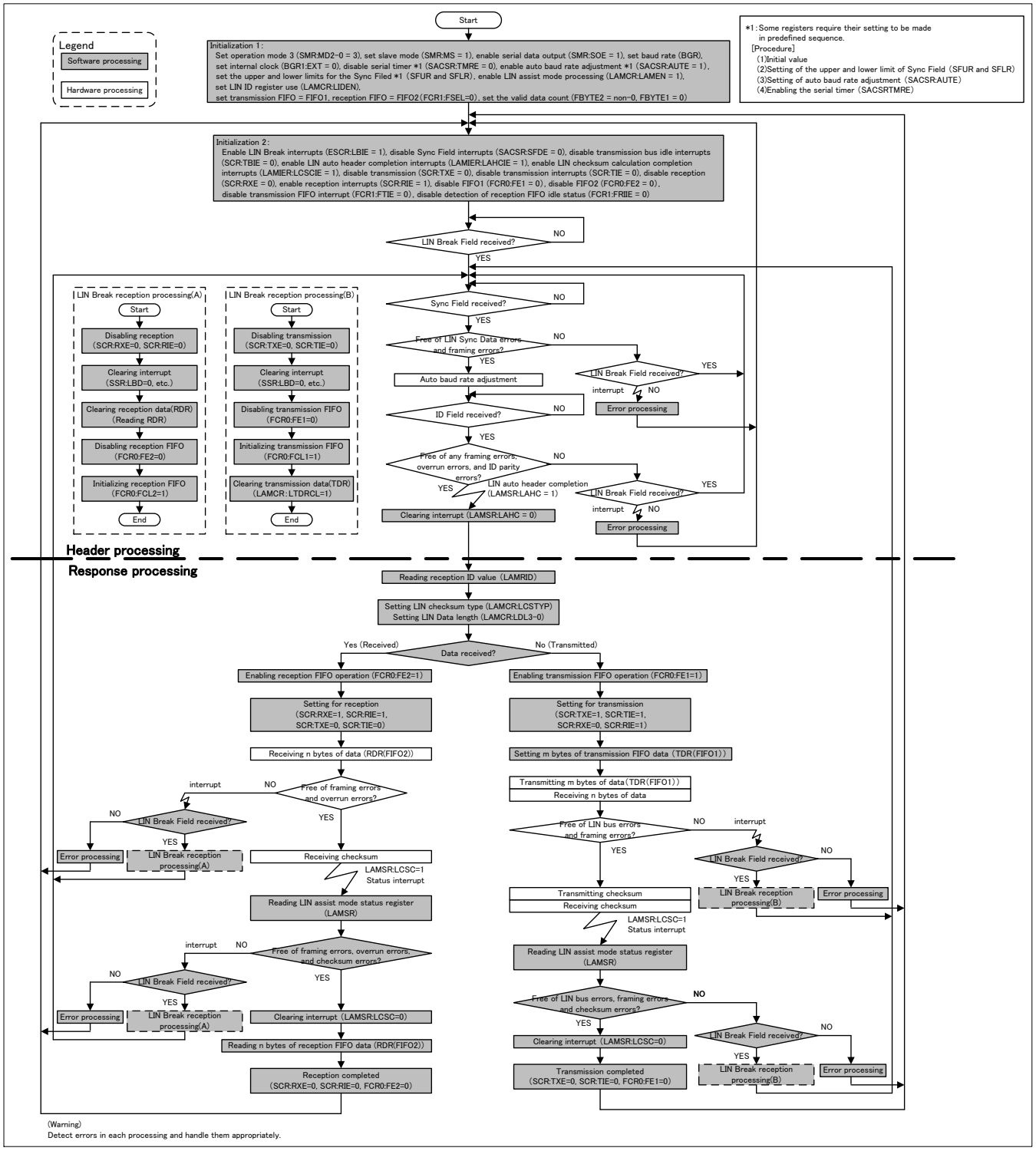


Figure 8-9 Example of LIN Communication Slave Mode (Assist Mode/a FIFO Is Used)



9. LIN 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)
	-	-

	bit15 bit8	bit7 bit0
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 Registers	SCRS (serial control set register)	SMRS (serial mode set register)
	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)
	-	-

	bit15 bit8	bit7 bit0
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
	-	-
	-	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
	-	-

*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 LINN 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.

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).

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

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.

[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

Note:

- *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.

"0": Set the SOUT pin disables serial data output.
"1": Set the SOUT pin as the serial data output pin (SOUT).

SOE	Serial Data Output Enable Bit
0	Disables serial data output
1	Serial data output pin

Notes:

- *Set a SOUT pin as a peripheral output pin (set with PPC_PCFGRijj). See "CHAPTER: I/O PORT" for how to make setups.*

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

[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.*

[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

[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".
- 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.*

[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 "1" is written to the UPCL bit in the serial control register (SCR).
- 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 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.

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 LBR bit in the serial control register (SCR) to "1" (LIN Break Field transmission).

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 and transmission data registers 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

Note:

- *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

Note:

- *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.

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).*
- *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

Note:

- *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".

Notes:

- *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.*
- *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 (BGR) 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 (BGR) 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 (BGR) 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 (BGR) 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.

Note:

- *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.

Note:

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).*

[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.

Note:
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).*

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:LCERIES 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.

Note:

- *This function is effective only in LIN assist mode (LAMCR:LAMEN = 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.

Note:

- *This function is effective only in LIN assist mode (LAMCR:LAMEN = 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.

Note:

- *This function is effective only in LIN assist mode (LAMCR:LAMEN = 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.

Note:

- *This function is effective only in LIN assist mode (LAMCR:LAMEN = 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 FLST bit 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	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
  - 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.
- 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

[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

- FIFO reset (writing "1" to FCL)
- 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
0	Do not save.
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 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. Therefore, set 0x00 as the transfer count of the FBYTE to the transmission FIFO selected by the FSEL.
- 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.
- 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.*
- *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.*
- *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 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.

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.*

[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.*
- *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.

[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 (SSR: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".*

[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:CSEIE 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:CSE 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	
Protection Attribute	-	-	-	-	-	-	-	-	
Initial Value	0	0	0	0	0	0	0	0	

[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

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	
Protection Attribute	-	-	-	-	-	-	-	-	
Initial Value	0	0	0	0	0	0	0	0	

[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

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,W	R0,W	R0,W	R0,W
Protection Attribute		-	-	-	-	-	-	-	-
Initial Value		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	
Protection Attribute	-	-	-	-	-	-	-	-	
Initial Value	0	0	0	0	0	0	0	0	

[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 37: 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 no FIFO is used) − Transmission and reception FIFOs (64 bytes each) (when the FIFOs are used)
2	Serial input	For serial clock and data input, noise from 2 to up to 38 bus clocks is filtered out.
3	Transfer mode	Synchronization
4	Baud rate	− Dedicated baud rate generators are provided (consisting of 15-bit reload counters).
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, 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.
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 the master and slave − An interrupt can be generated at the transfer time and bus error time. − The 10-bit addressing function is supported in programs. − 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 (1 to 256 divisions). − Activation by external trigger is possible.
10	FIFO	− Transmission and reception FIFOs are provided (64 bytes for the transmission FIFO and 64 bytes for the reception FIFO). − The transmission FIFO and reception FIFO can be selected. − Transmission data can be retransmitted. − The reception FIFO interrupt timing 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 "0" to timer interrupt flag bit (SACSR:TINT) 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)
	RBERR	ESR	Reception block transfer error	ECR:REIE	Writing "1" to reception block transfer error clear bit (ESRC:RBERRC)
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)
	TBERR	ESR	Transmission block transfer error	ECR:TEIE	Writing "1" to transmission block transfer error clear bit (ESRC:TBERRC)

*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)

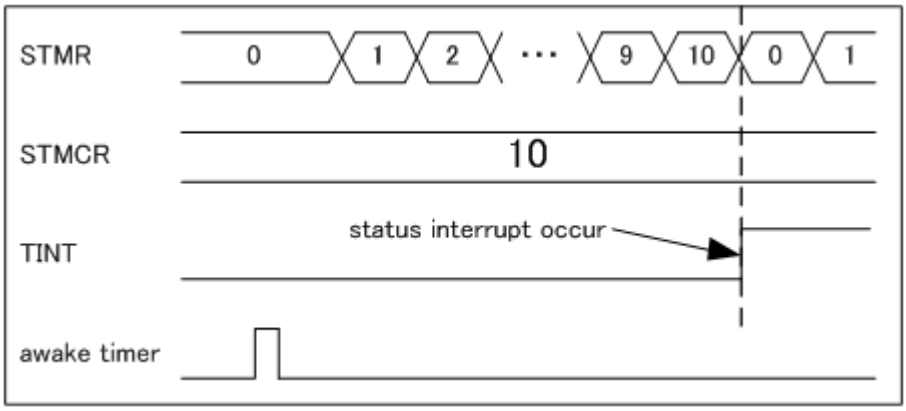
2.1. Timer Interrupt Generation and Flag Set Timing

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



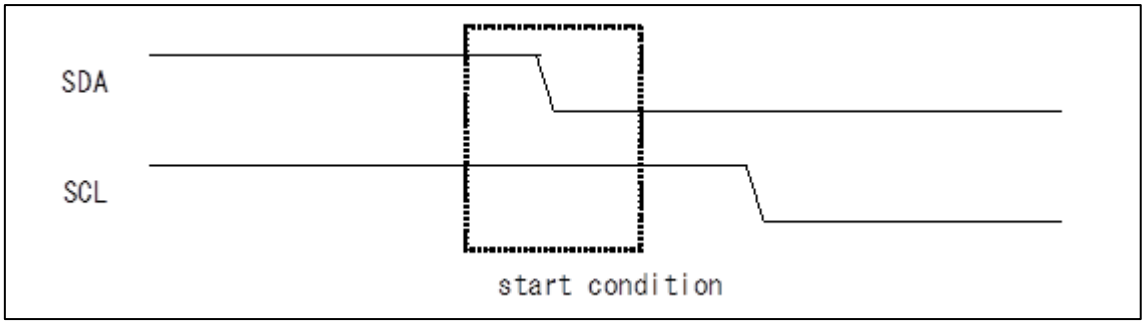
3. I²C Interface Operation

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.

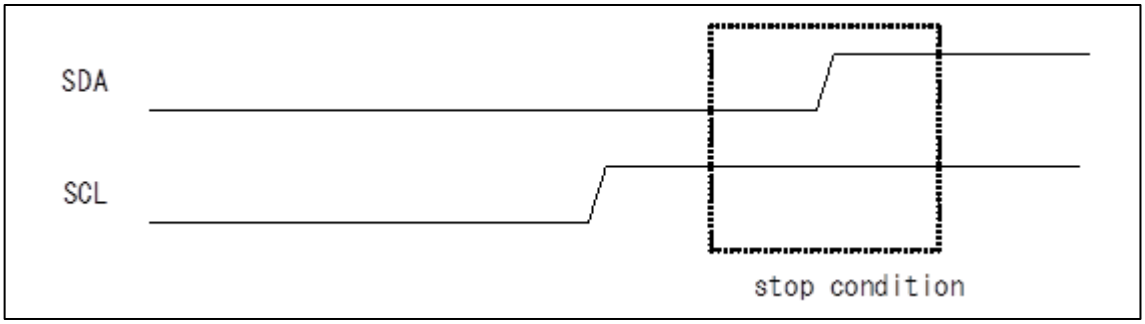
Figure 3-1 Start Condition



I²C Bus Stop Condition

The following figure shows the stop condition of the I²C bus.

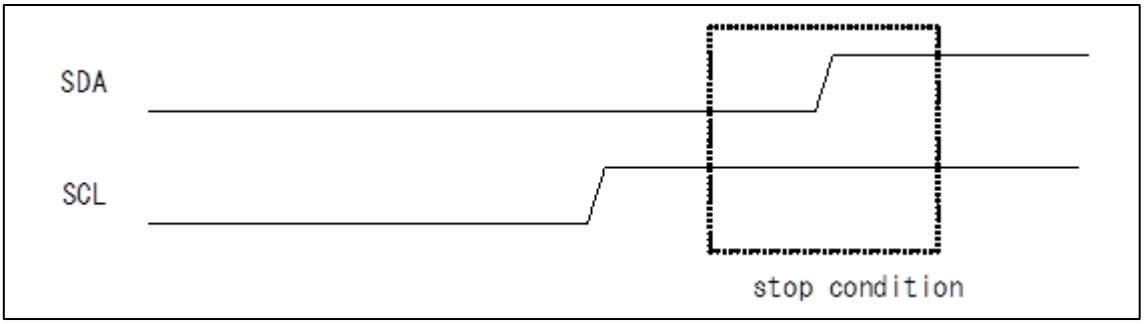
Figure 3-2 Stop Condition



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".

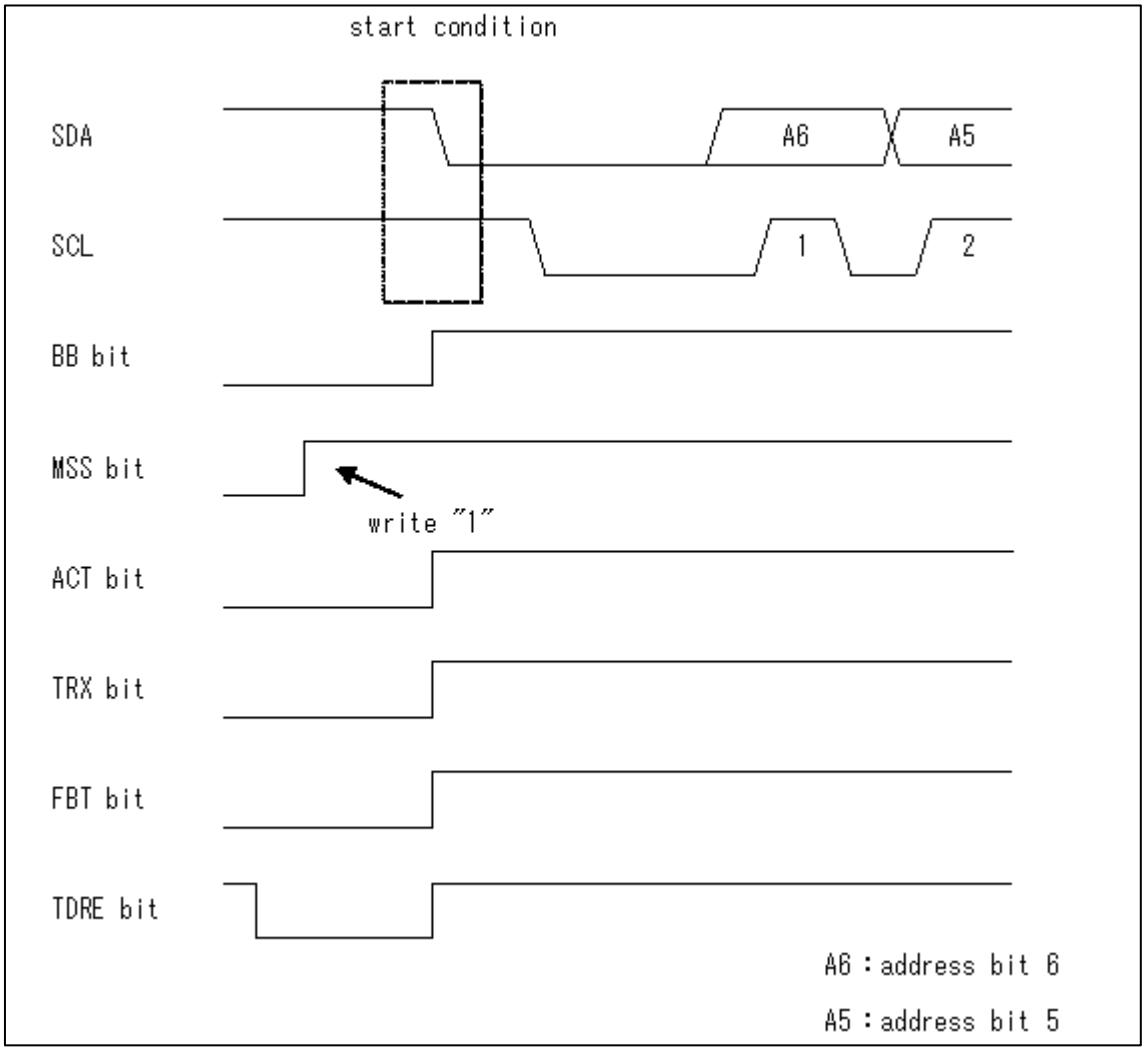
Start Condition Occurrence

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 (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.*

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)

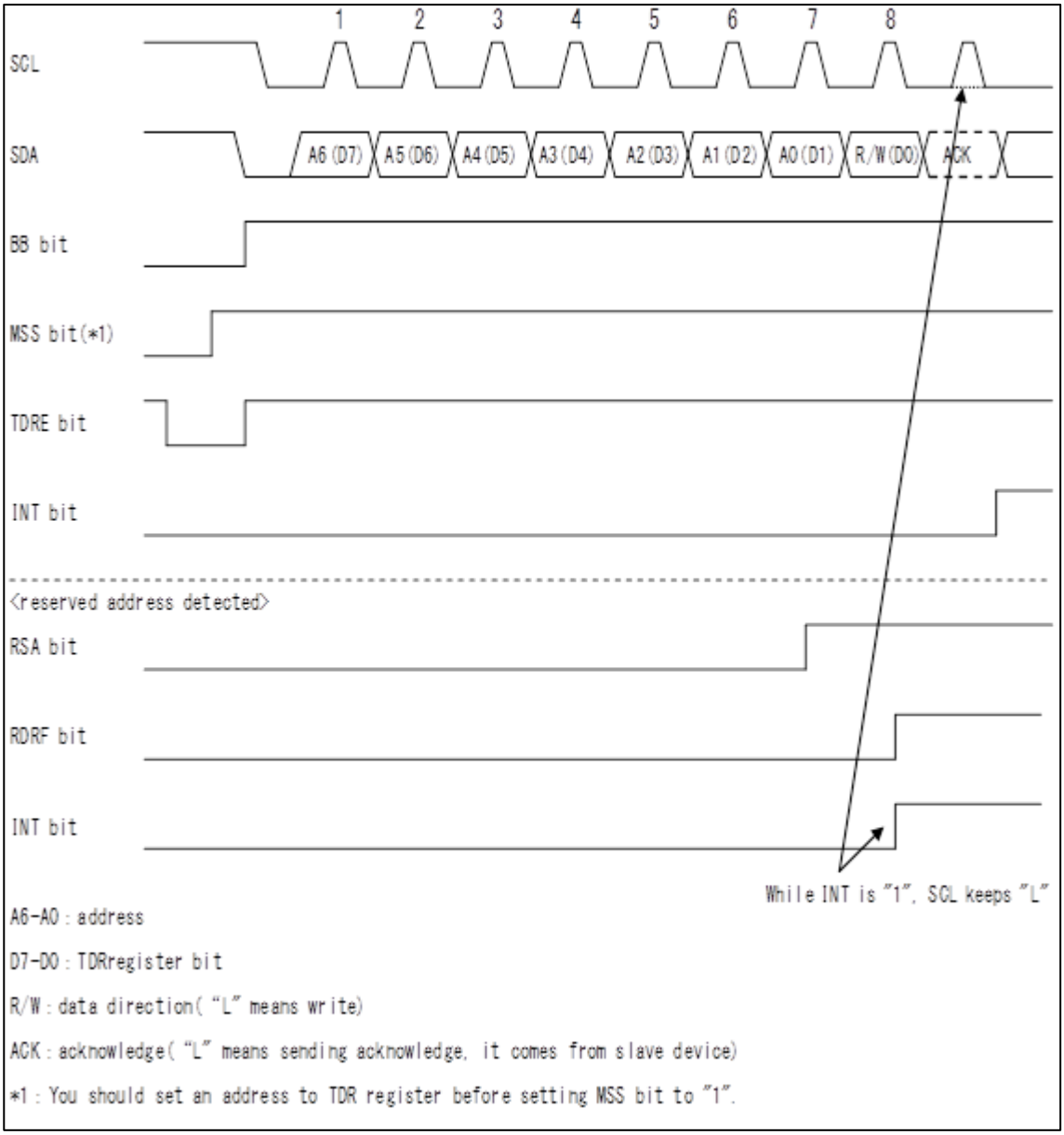
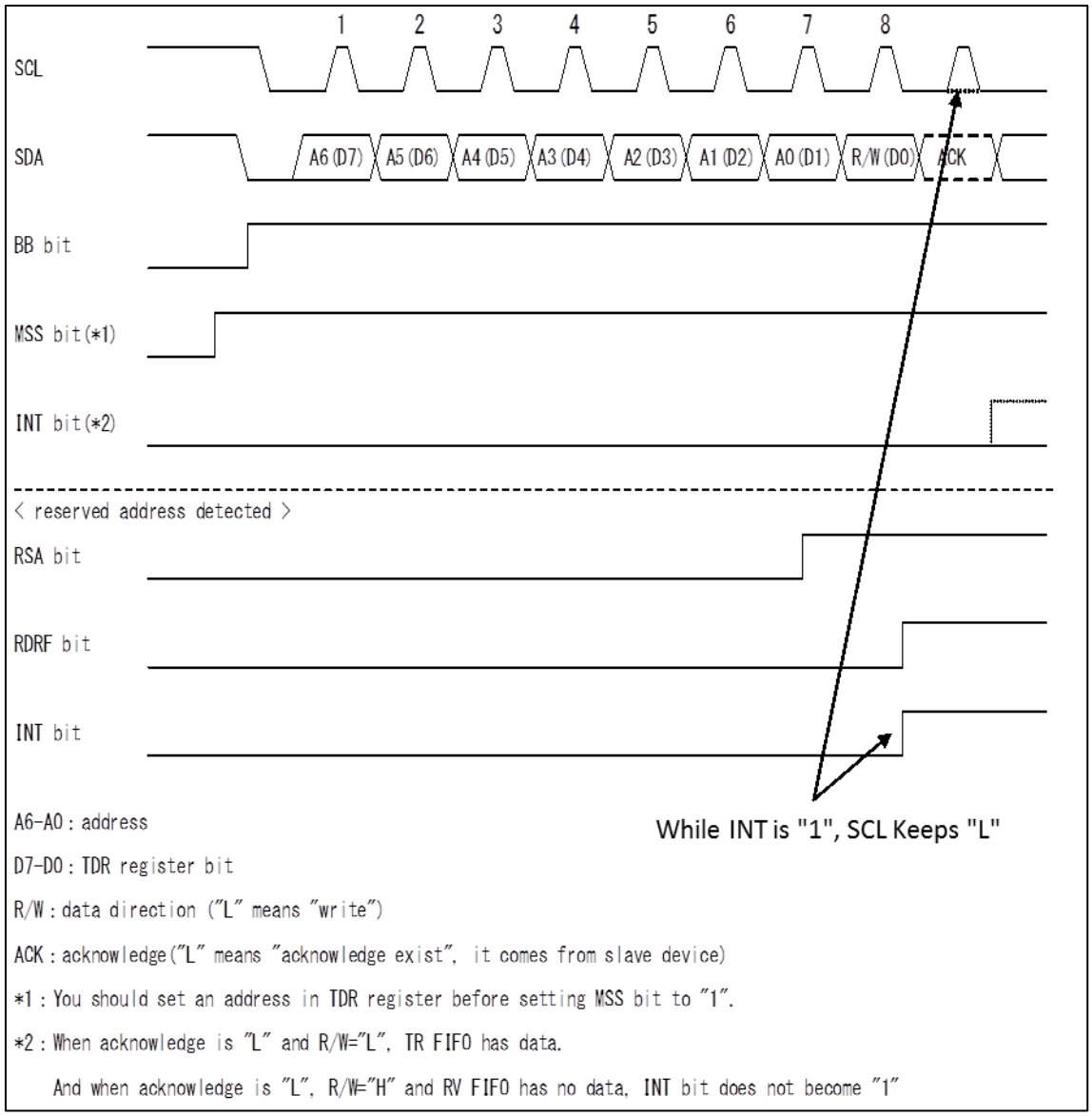


Figure 3-6 Address and Data Direction (When the Transmission and Reception FIFOs Are Enabled)



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		

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 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)

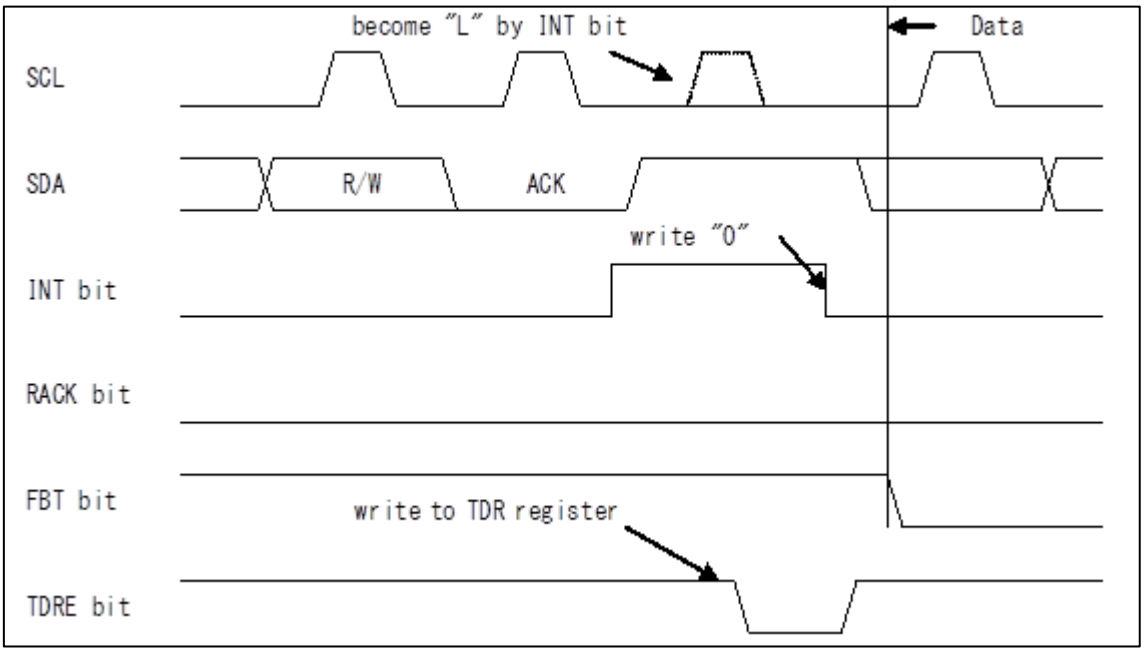
FIFO disabled (transmission and reception FIFOs both disabled)

- 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, and "0" is written to the interrupt flag. This 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)

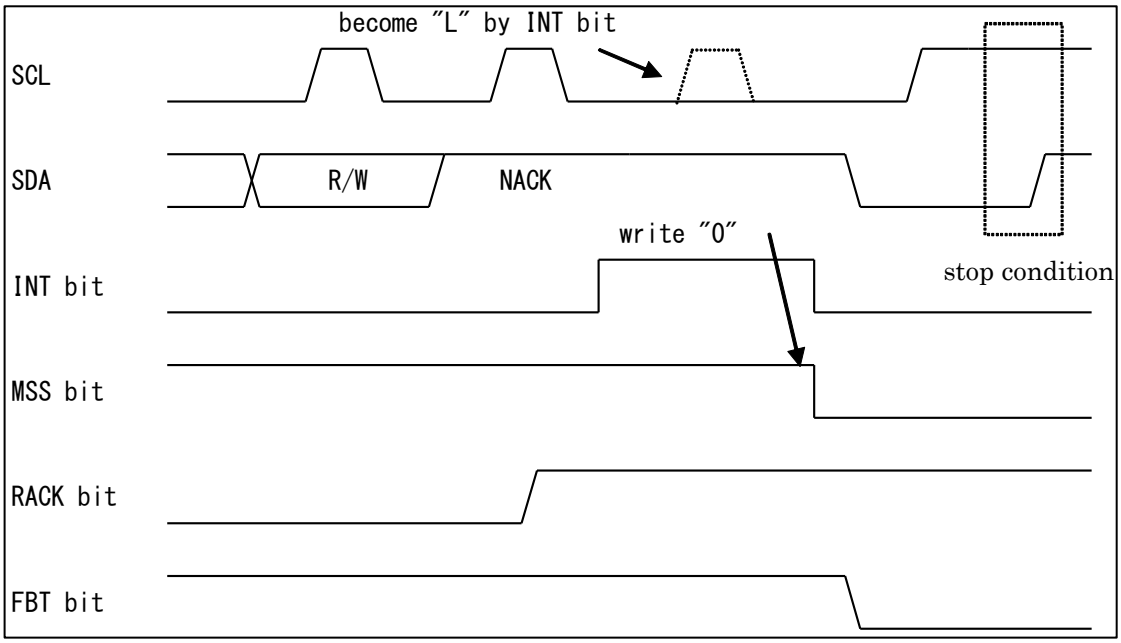


Figure 3-9 Acknowledgment (with FIFO Disabled, IBSR:RSA= 1, and an ACK Response)

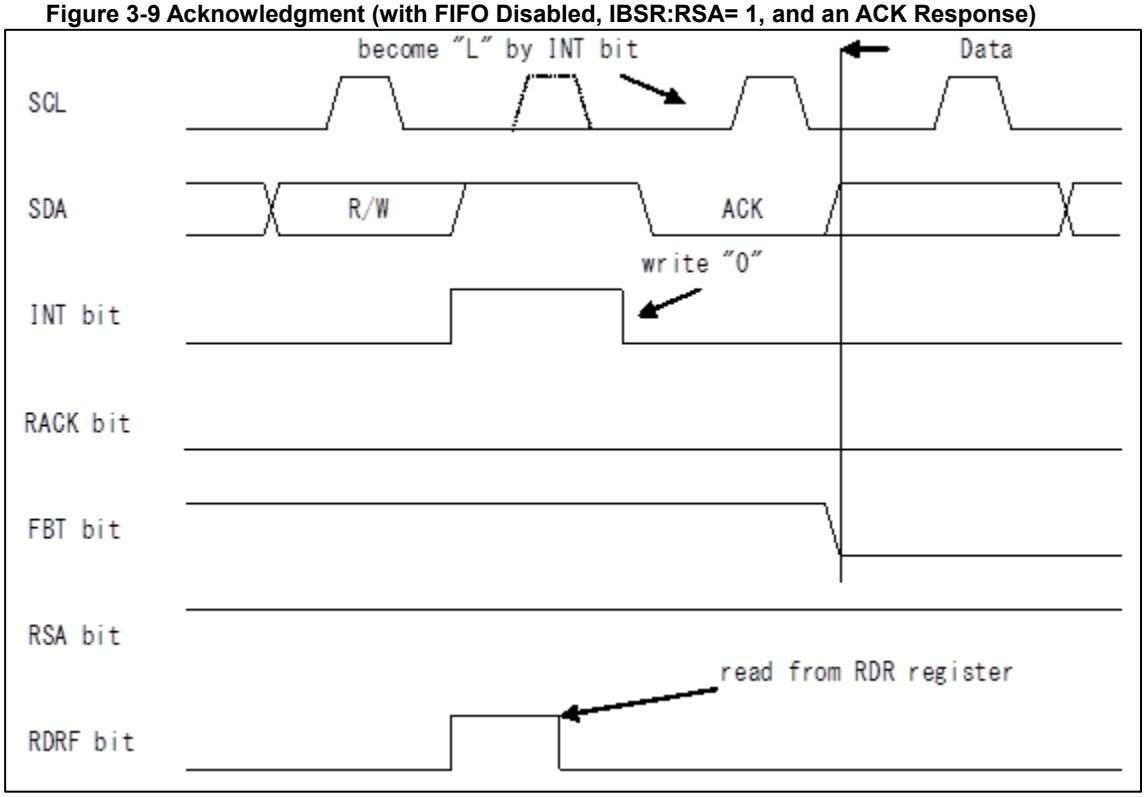


Figure 3-10 Acknowledgment (with FIFO Disabled, IBSR:RSA= 1, and a NACK Response)

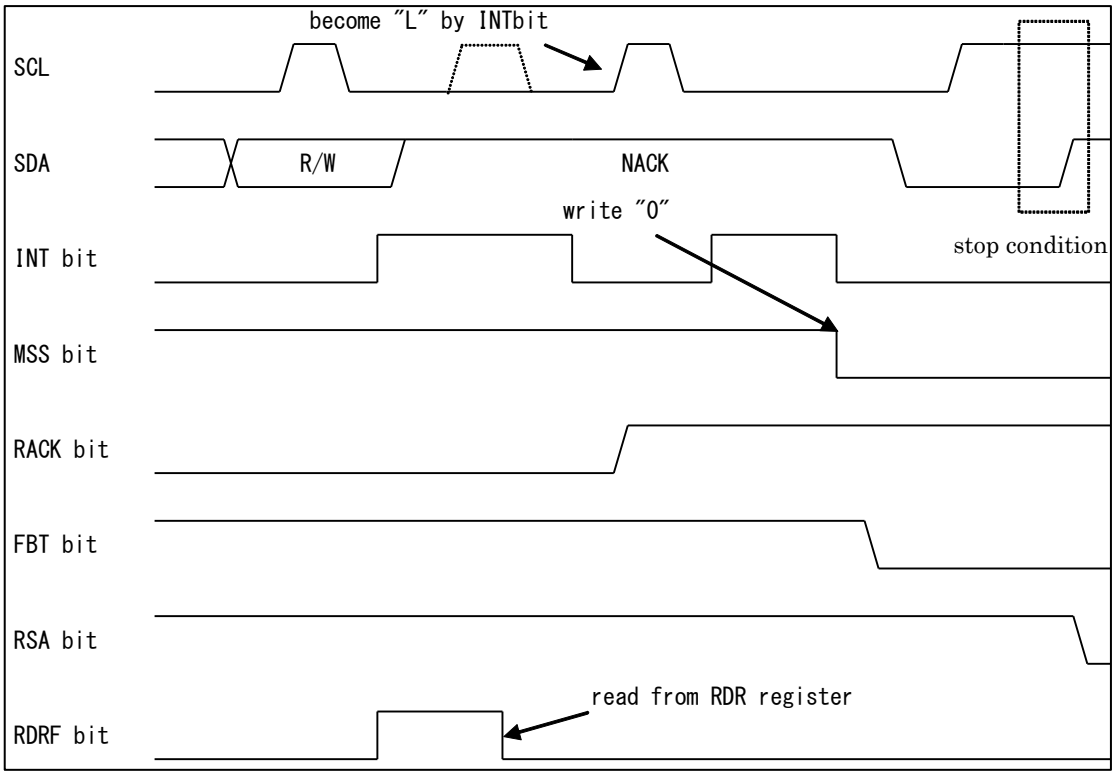
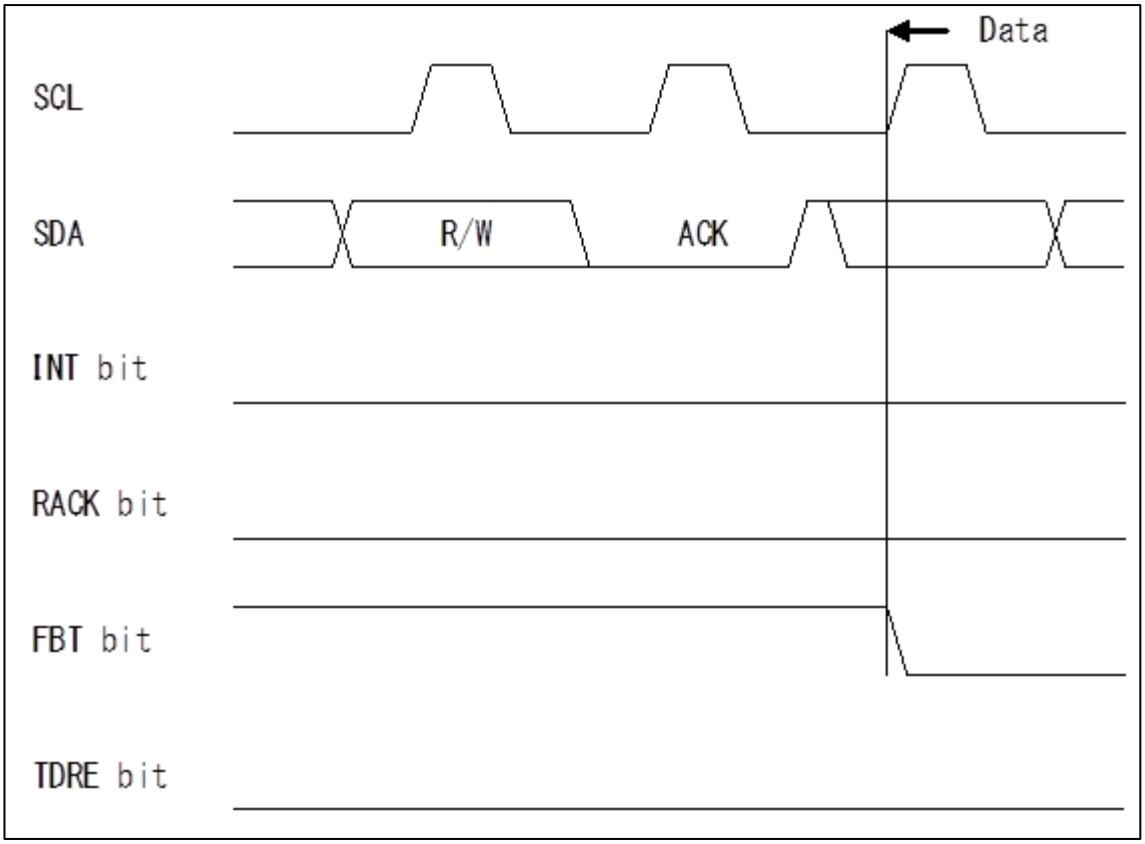


Figure 3-11 Acknowledgment (with FIFO Enabled, the Transmission FIFO with Data, the Reception FIFO without Data, IBSR:RSA= 0, and an ACK Response)



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 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 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 I^2C 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 I^2C 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 I^2C bus.
6. After the transmission of 1 byte, set the interrupt 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 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 I^2C 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 I^2C 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 I^2C bus.
6. 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.
7. Write the data to be transmitted, to the TDR register to release the wait for the I^2C bus.
8. 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 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 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.
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.

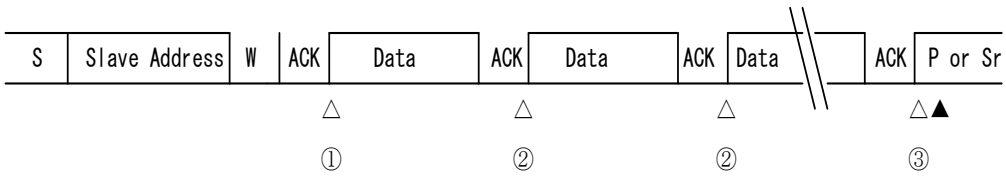
*2 To issue the repeated start condition when DMA mode is enabled (SSR:DMA= 1), the SSR:TBI bit is "1", and the IBCR:INT bit is "0", perform the following procedure.

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

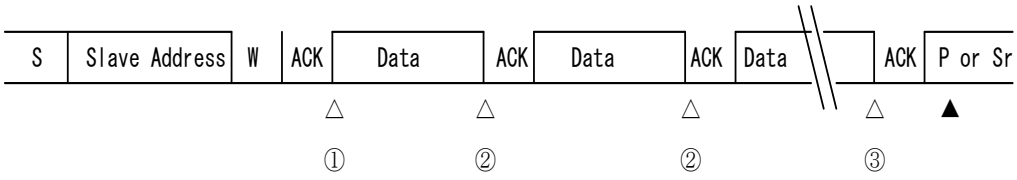
— Write INT = "0" after the transmission data is written to the TDR register

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

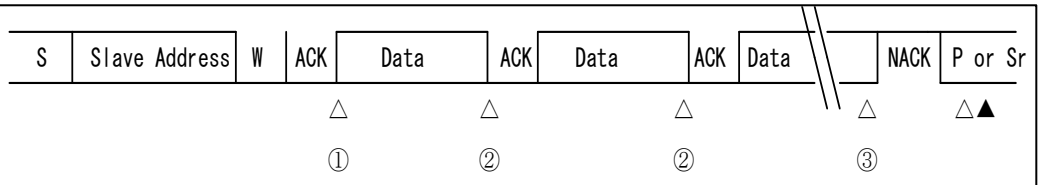
— Write INT = "0" after the transmission data is written to the TDR register

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

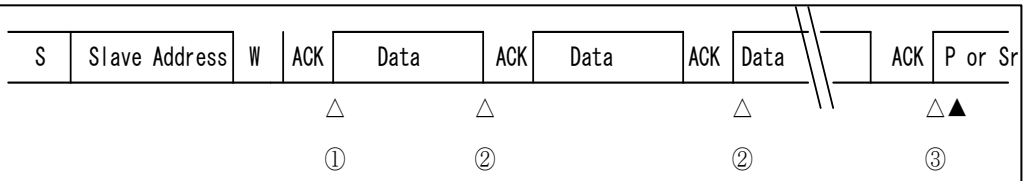
- Write INT="0"after the transmission data is written to the TDR register

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

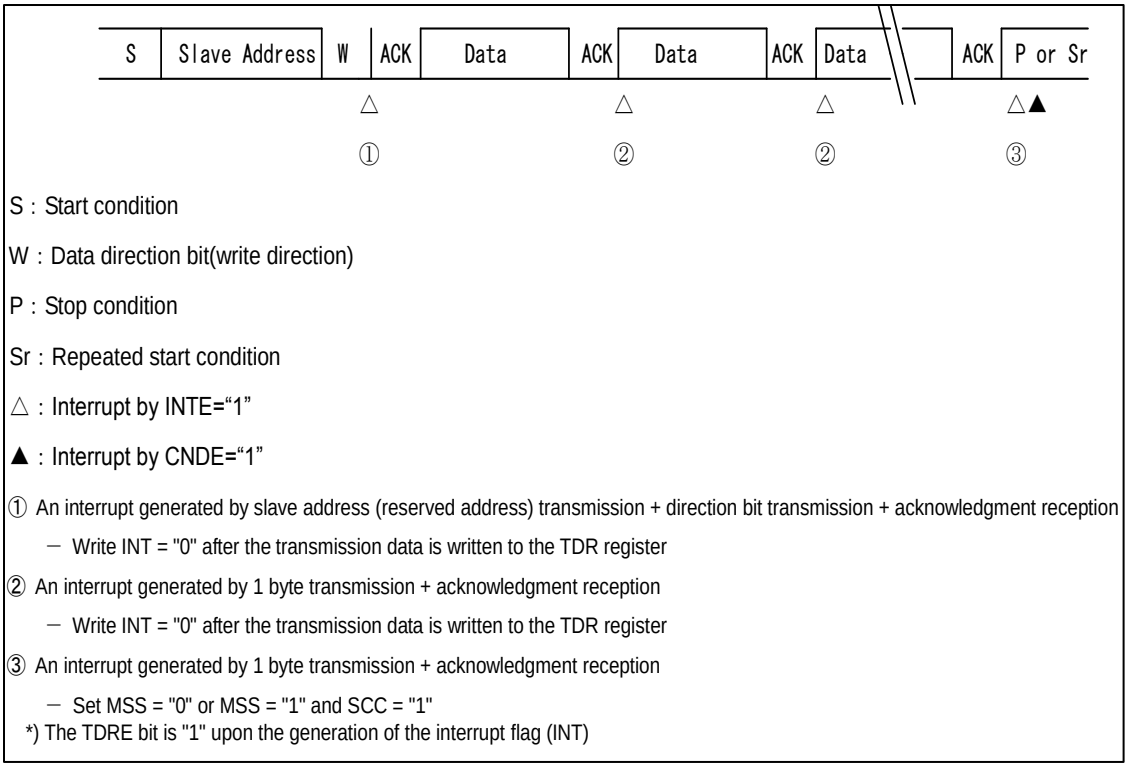
- Write WSEL="0", INT="0" after the transmission data is written to the transmission buffer

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)

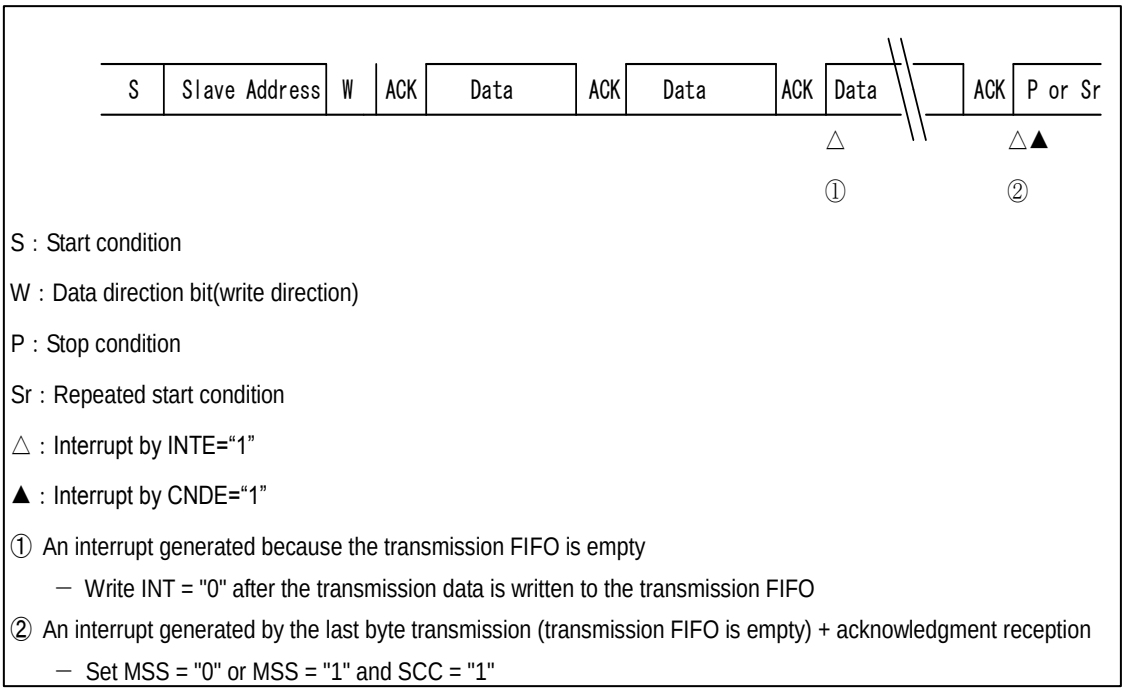


Figure 3-19 Master Transmission Interrupts 8 When FIFO Is Enabled (SSR:DMA= 0, IBCR:WSEL= 1, and IBSR:RSA= 0)

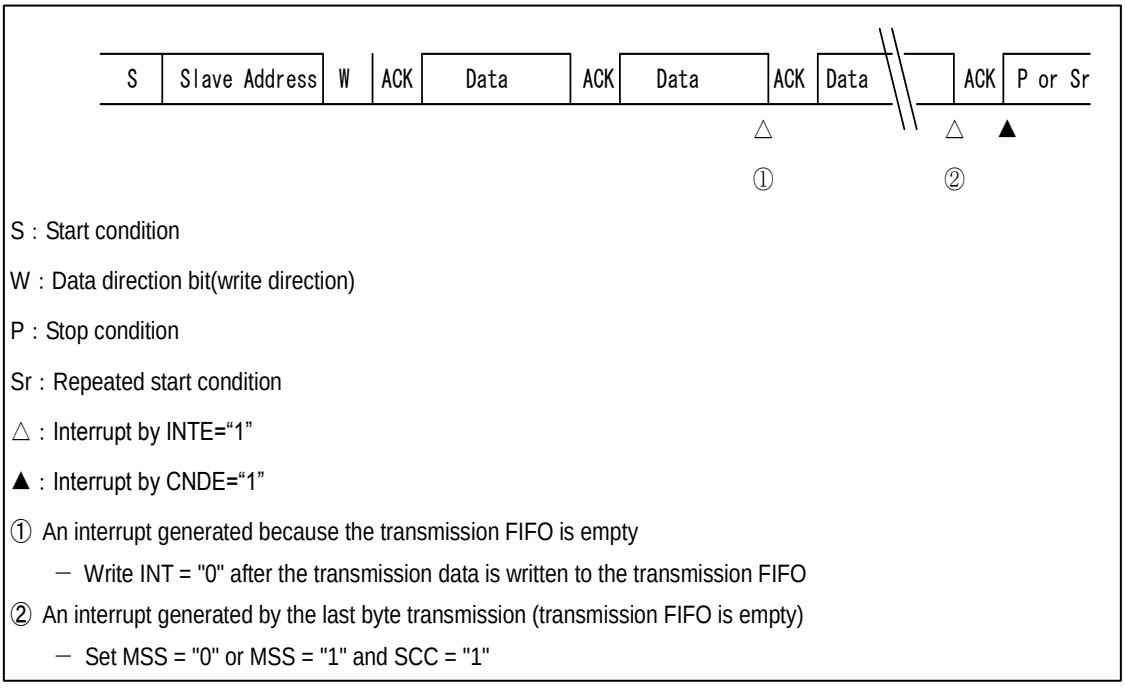
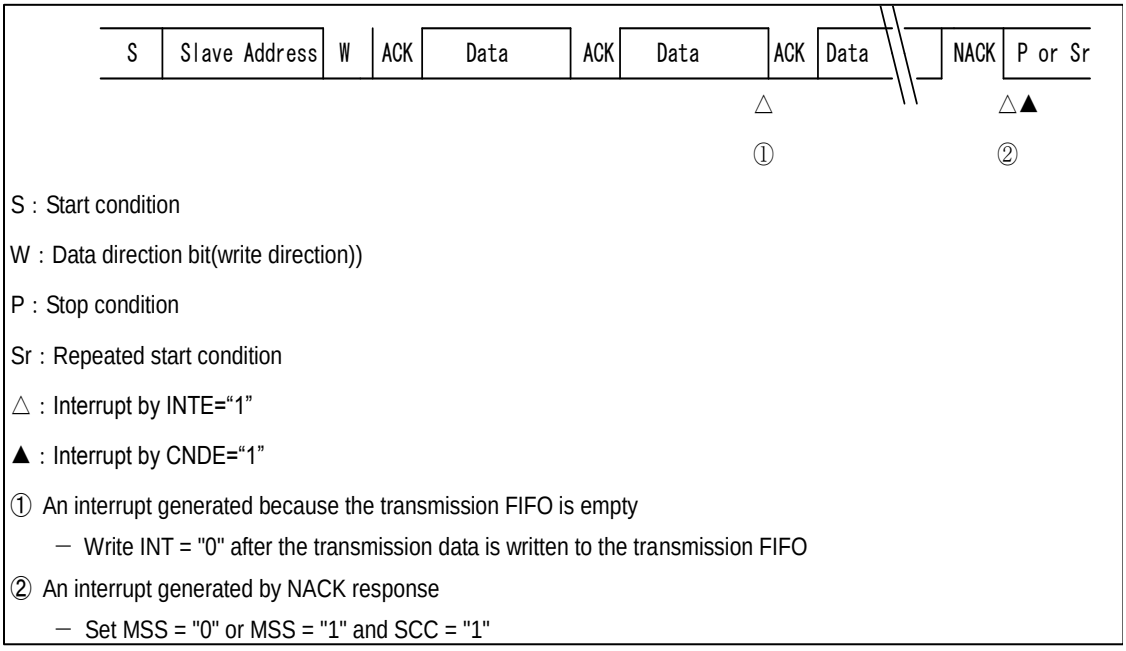


Figure 3-20 Master Transmission Interrupts 9 When FIFO Is Enabled (SSR:DMA= 0, IBCR:WSEL= 1, IBSR:RSA= 0, and NACK Response)



S : Start condition

W : Data direction bit(write direction))

P : Stop condition

Sr : Repeated start condition

△ : Interrupt by INTE="1"

▲ : Interrupt by CNDE="1"

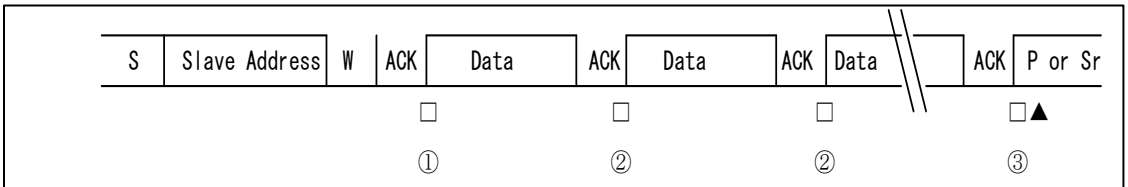
① An interrupt generated because the transmission FIFO is empty

– 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

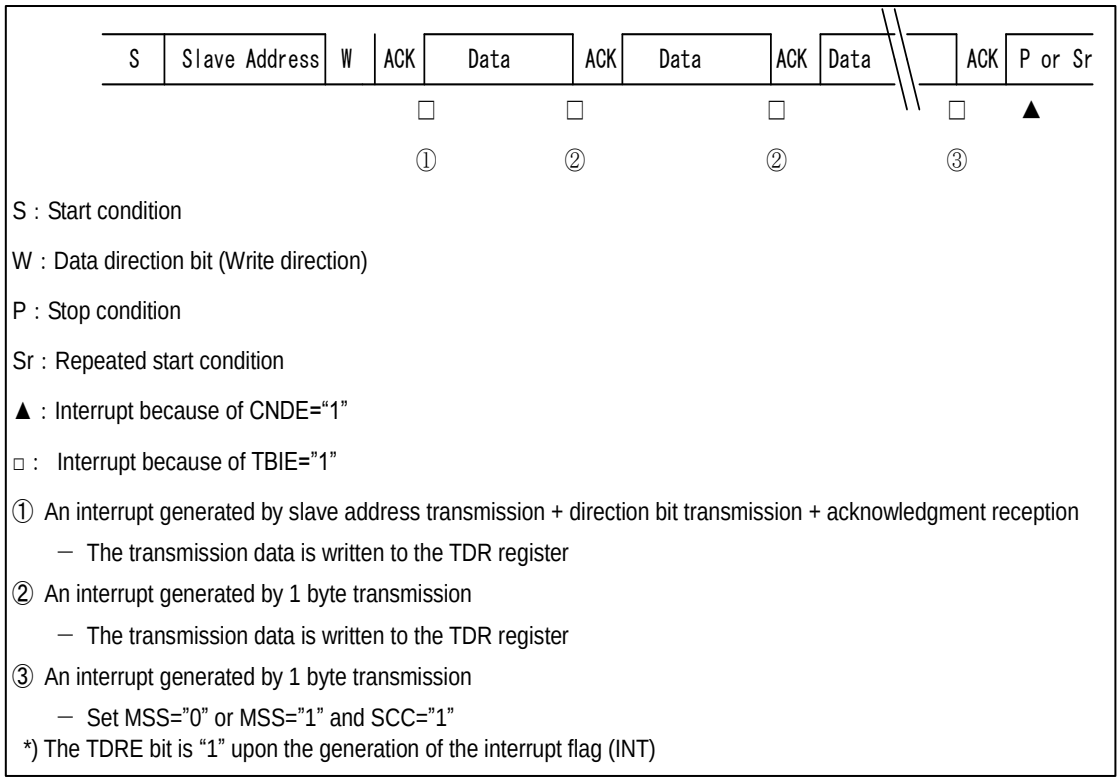
– the transmission data is written to the TDR register

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

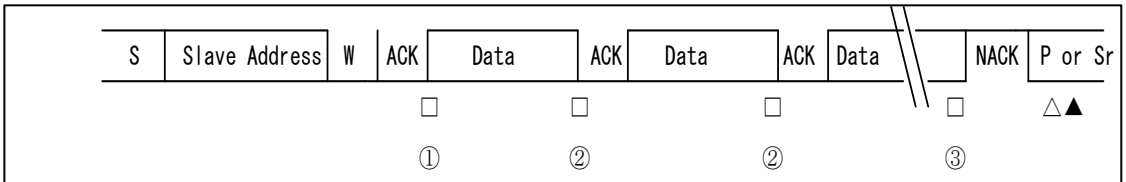
— The transmission data is written to the TDR register

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

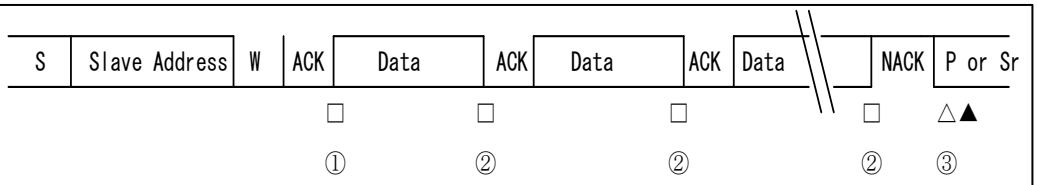
- The transmission data is written to the TDR register

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

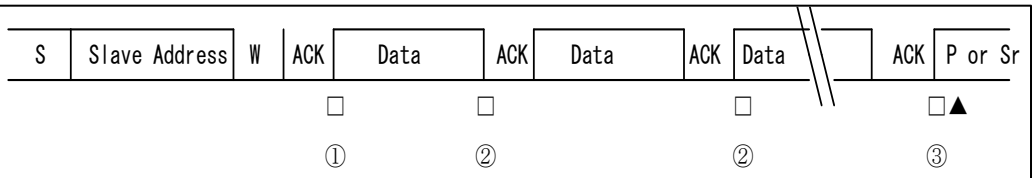
- The transmission data is written to the TDR register

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

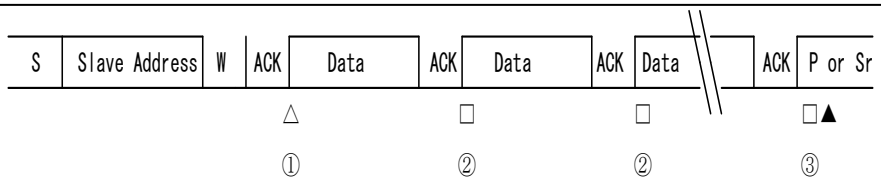
- The transmission data is written to the transmission buffer after setting WSEL="0"

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

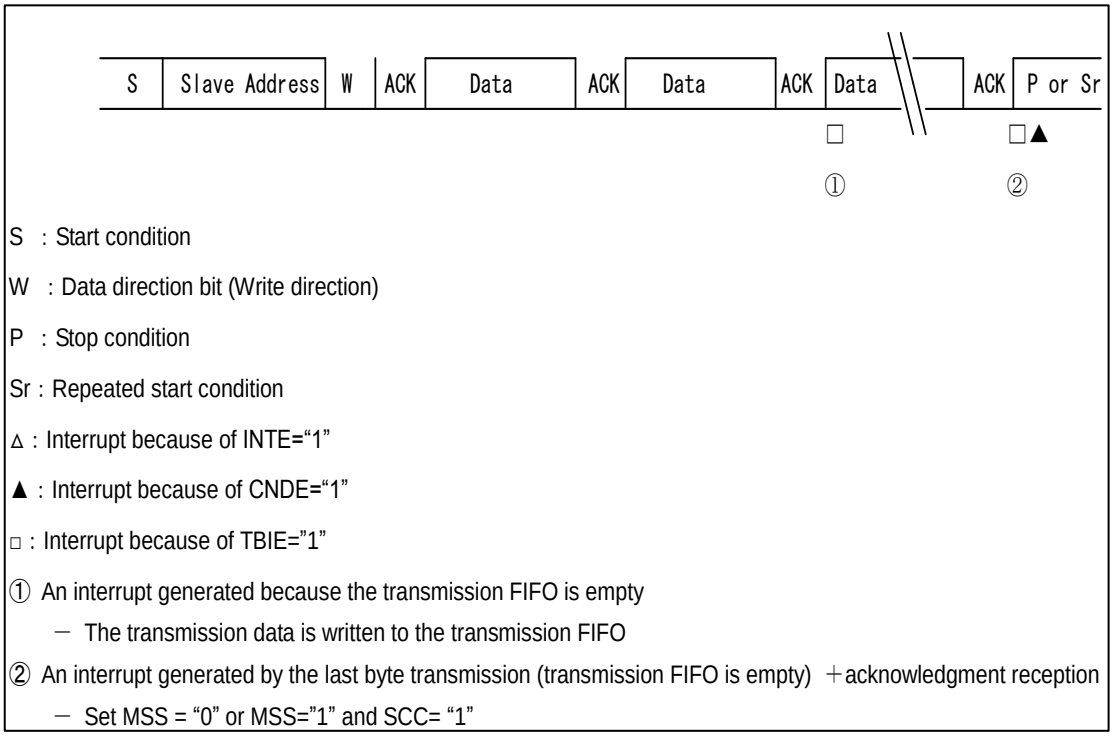
– The transmission data is written to the TDR register

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)



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"

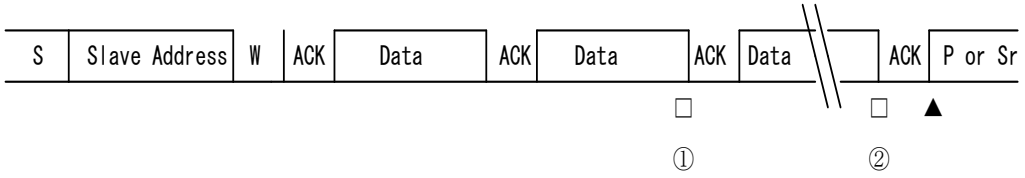
① An interrupt generated because the transmission FIFO is empty

— The transmission data is written to the transmission FIFO

② An interrupt generated by the last byte transmission (transmission FIFO is empty) ＋acknowledgment reception

— Set MSS = "0" or MSS="1" and SCC= "1"

Figure 3-28 Master Transmission Interrupts 17 When FIFO Is Enabled (SSR:DMA= 1, IBCR:WSEL= 1, 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"

□ : Interrupt because of TBIE="1"

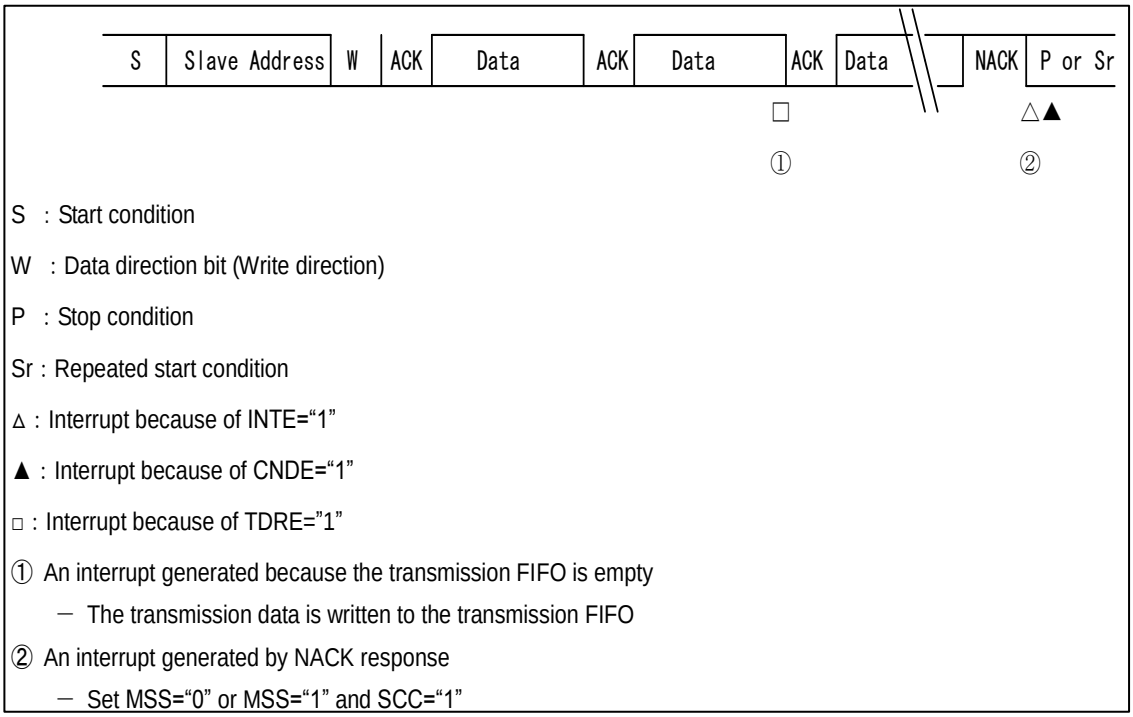
① An interrupt generated because the transmission FIFO is empty

- The transmission data is written to the transmission FIFO

② An interrupt generated by the last byte transmission (transmission FIFO is empty)

- Set MSS="0" or MSS="1" and SCC="1"

Figure 3-29 Master Transmission Interrupts 18 When FIFO Is Enabled (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 TDRE="1"

① An interrupt generated because the transmission FIFO is empty

- 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"

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^{2}C 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^{2}C 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^{2}C 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^{2}C 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^{2}C 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".

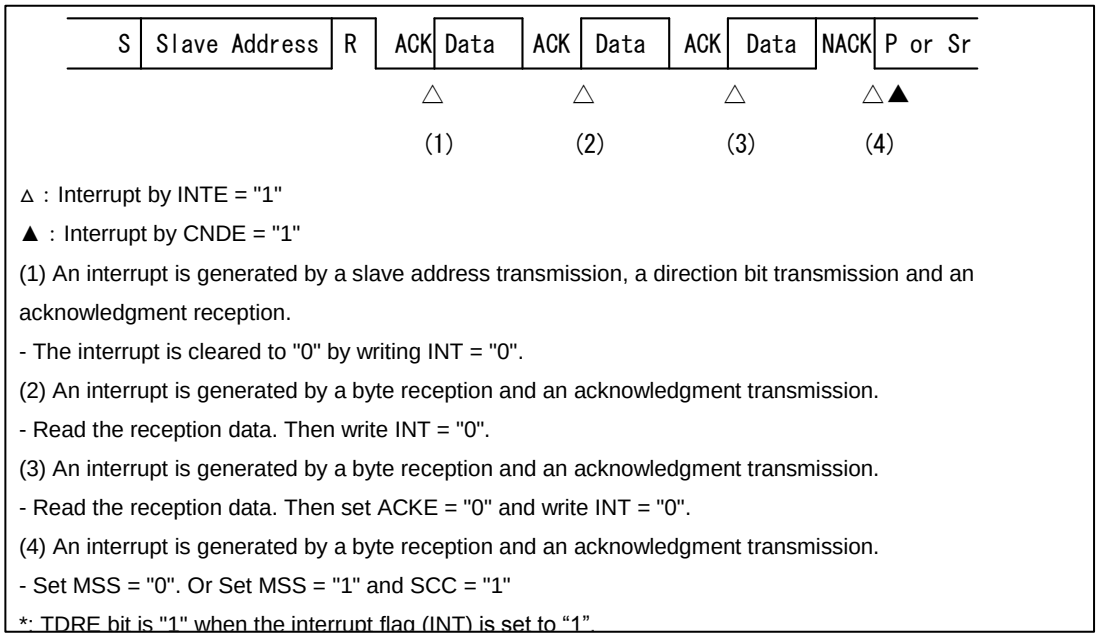
*2 Regardless of the IBCR:WSEL setting, the reception data full flag (SSR:RDRF) is set to "1" immediately after the reception of 1 byte. For the 2nd or subsequent byte, there is the following wait for the I^{2}C bus when the reception data full flag (SSR:RDRF) is set to "1". If IBCR:WSEL = 0,

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.

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.

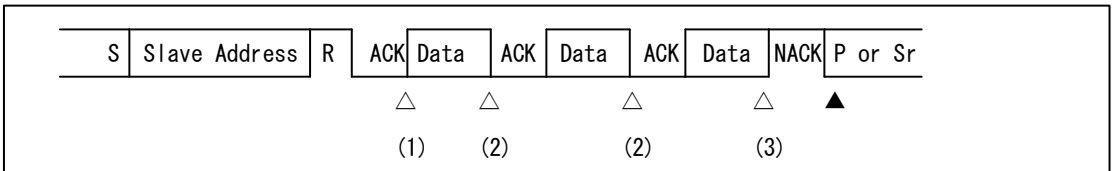
- Read the reception data. Then set ACKE = "0" and write INT = "0".

(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)

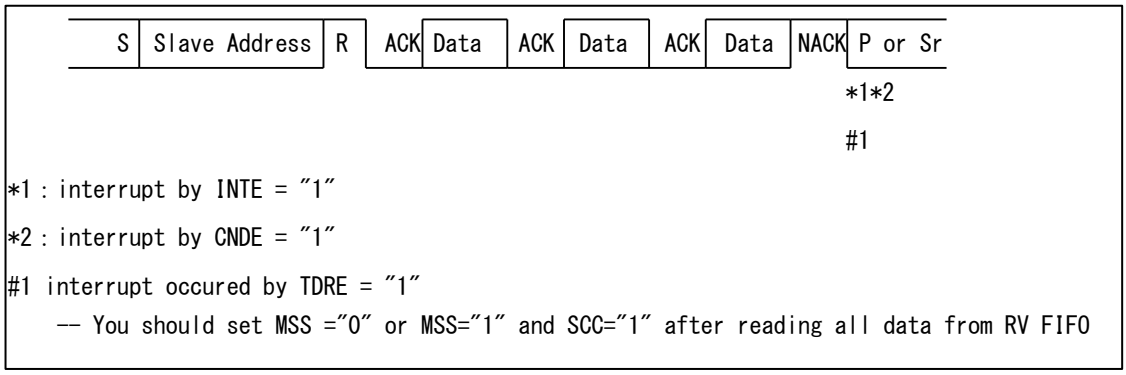


Figure 3-33 Master Reception Interrupts 4 When FIFO Is Enabled (SSR:DMA= 0, IBCR:WSEL= 1, and IBSR:RSA= 0)

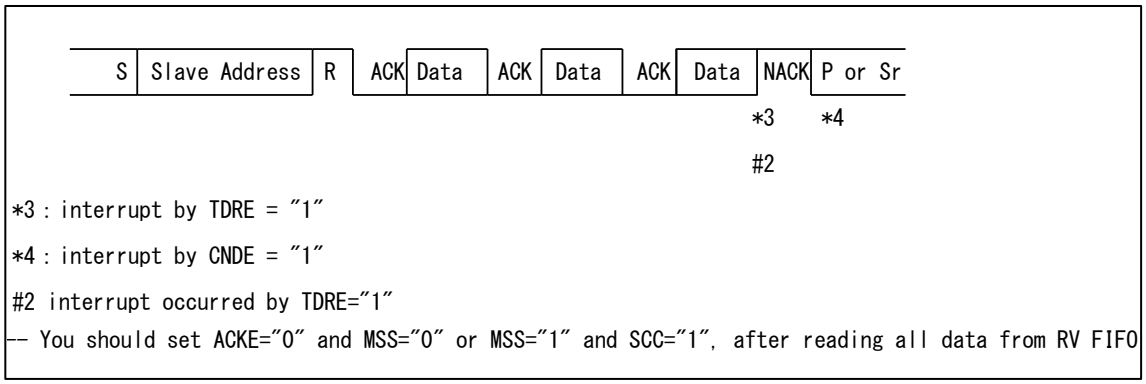


Figure 3-34 Master Reception Interrupts 5 When FIFO Is Disabled (SSR:DMA= 1, IBCR:WSEL= 0, and IBSR:RSA= 0)

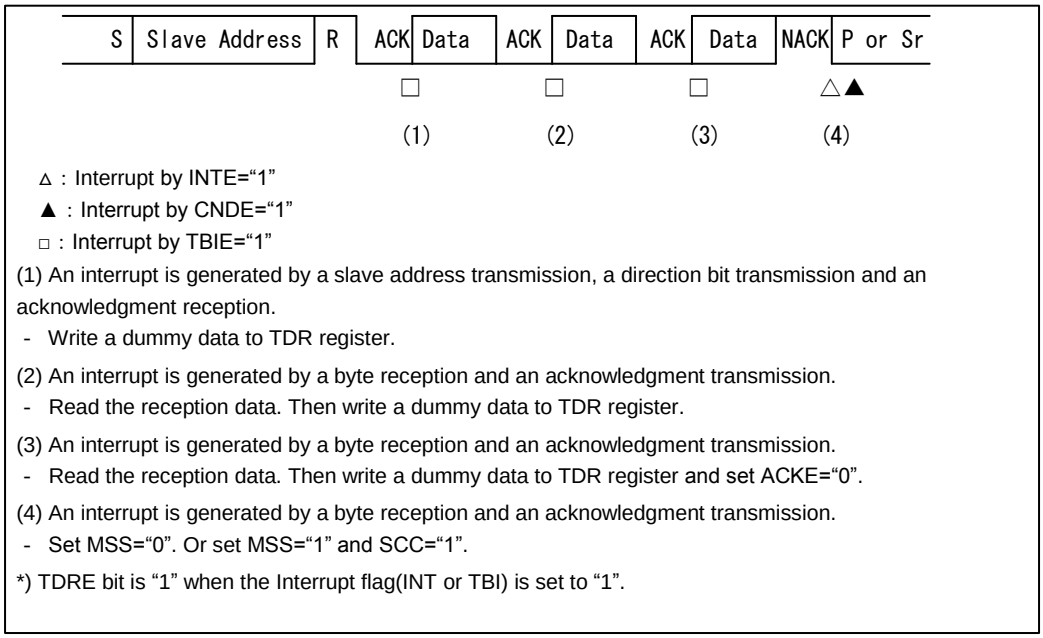


Figure 3-35 Master Reception Interrupts 6 When FIFO Is Disabled (SSR:DMA= 1, IBCR:WSEL= 1, and IBSR:RSA= 0)

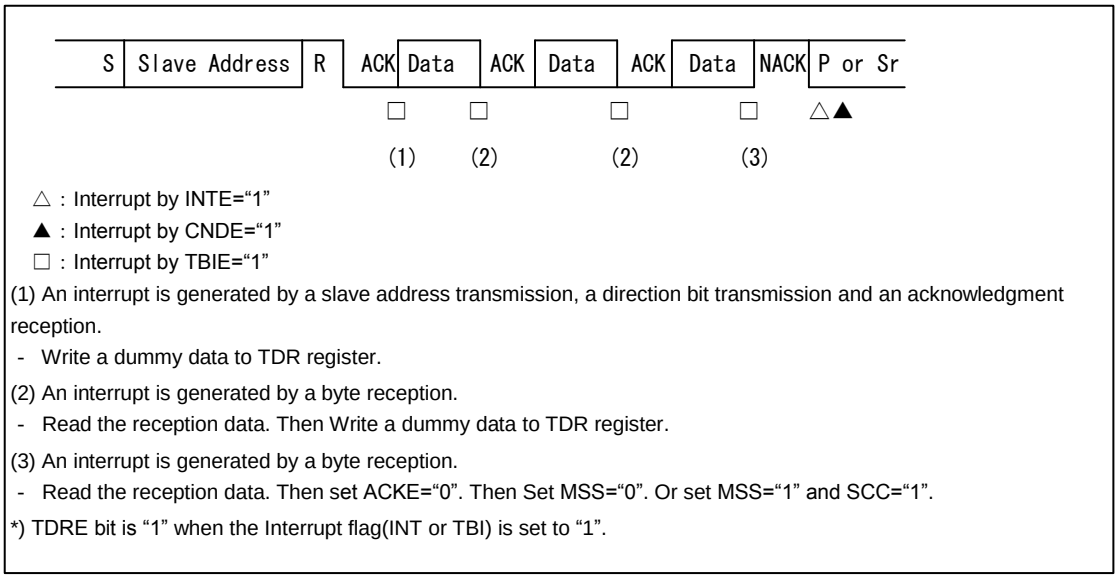


Figure 3-36 Master Reception Interrupts 7 When FIFO Is Enabled (SSR:DMA= 1, IBCR:WSEL= 0, IBCR:ACKE= 0, and IBSR:RSA= 0)

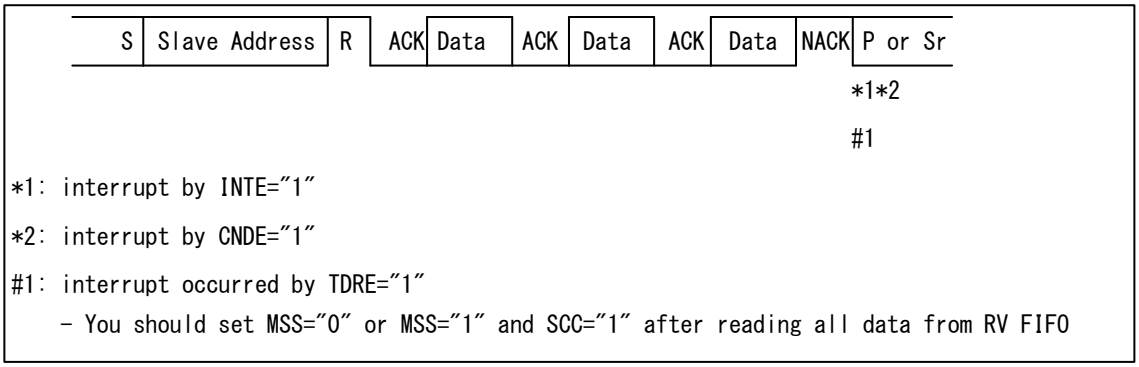
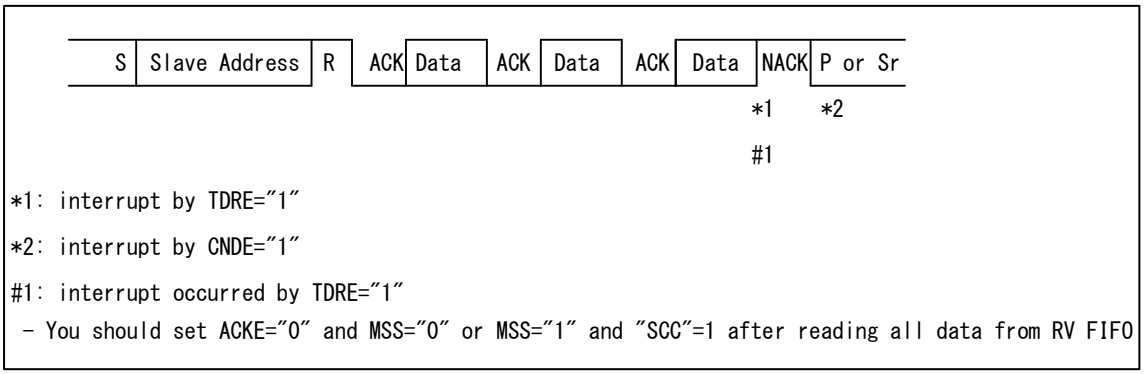


Figure 3-37 Master Reception Interrupts 8 When FIFO Is Enabled (SSR:DMA= 1, IBCR:WSEL= 1, and IBSR:RSA= 0)



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".

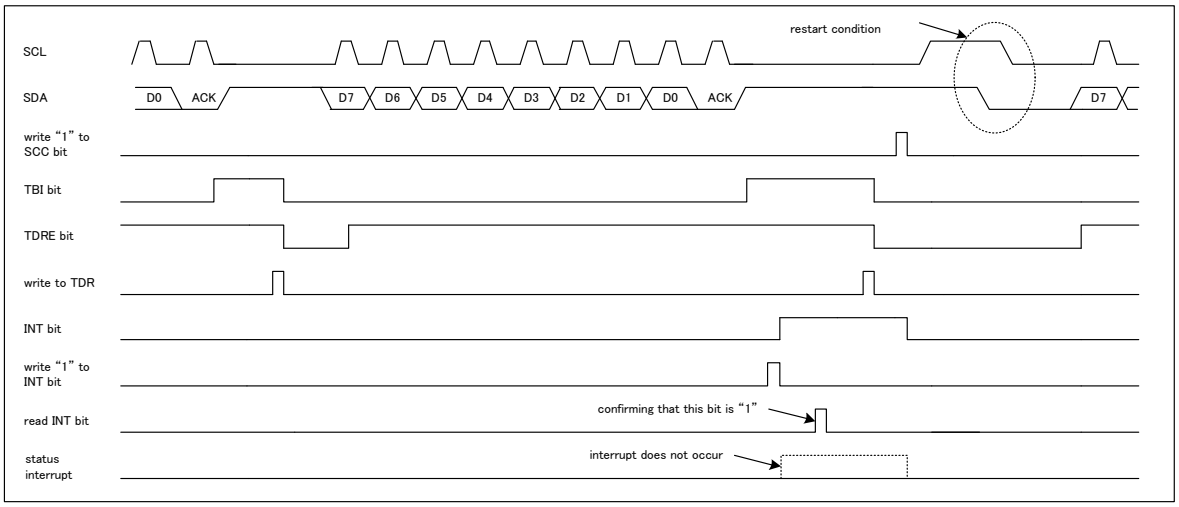
Issuing the Repeated Start Condition When DMA Mode is Enabled (SSM:DMA=1)

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 (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.*
- *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

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.

Table 3-7 Operations Immediately after Acknowledgment Output for a Slave Address

<table>
<tr><th rowspan="2">Transmission FIFO</th><th rowspan="2">Reception FIFO</th><th rowspan="2">Transmission FIFO State</th><th rowspan="2">Reception FIFO State</th><th rowspan="2">Data Direction Bit (R/W)</th><th colspan="2">Operation Immediately after Acknowledgment</th></tr>
<tr><th>Acknowledgment Is ACK</th><th>Acknowledgment Is NACK</th></tr>
<tr><td rowspan="2">Disabled</td><td rowspan="2">Disabled</td><td rowspan="2">-</td><td rowspan="2">-</td><td>0</td><td rowspan="2">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.</td><td rowspan="2">The IBCR:INT bit remains "0", and no wait is performed.</td></tr>
<tr><td>1</td></tr>
<tr><td rowspan="3">Disabled</td><td rowspan="3">Enabled</td><td rowspan="3">-</td><td>Without data</td><td rowspan="2">0</td><td>The IBCR:INT bit remains "0", and no wait is performed.</td><td rowspan="3">The IBCR:INT bit remains "0", and no wait is performed.</td></tr>
<tr><td>With data</td><td>The IBCR:INT bit is set to "1", and a wait is performed.</td></tr>
<tr><td>-</td><td>1</td><td>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.</td></tr>
</table>

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.
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 I^{2}C 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)

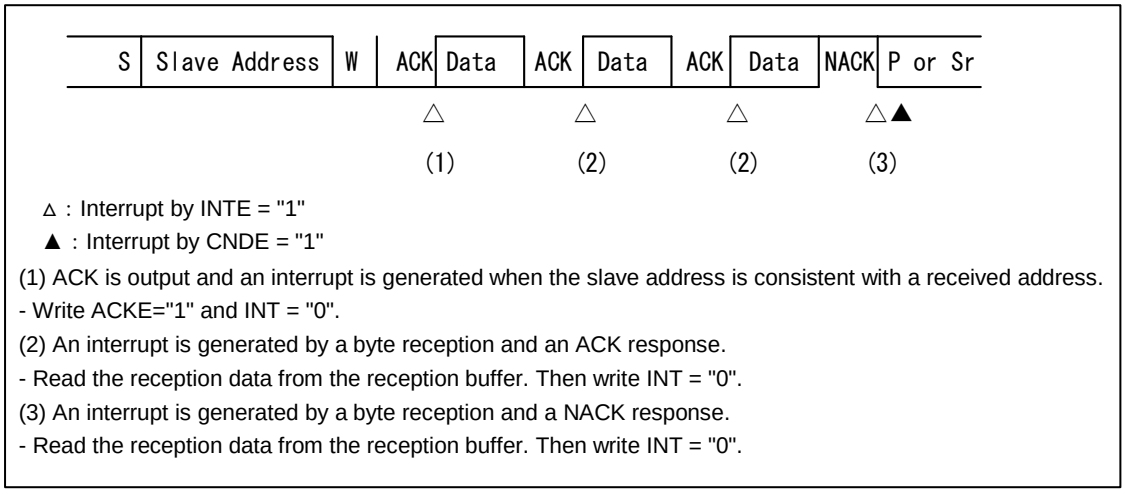
■ When the reception FIFO is disabled

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 3-39 Slave Reception Interrupts 1 When FIFO Is Disabled (SSR:DMA= 0, IBCR:WSEL= 0, and IBSR:RSA= 0)



△ : Interrupt by INTE = "1"

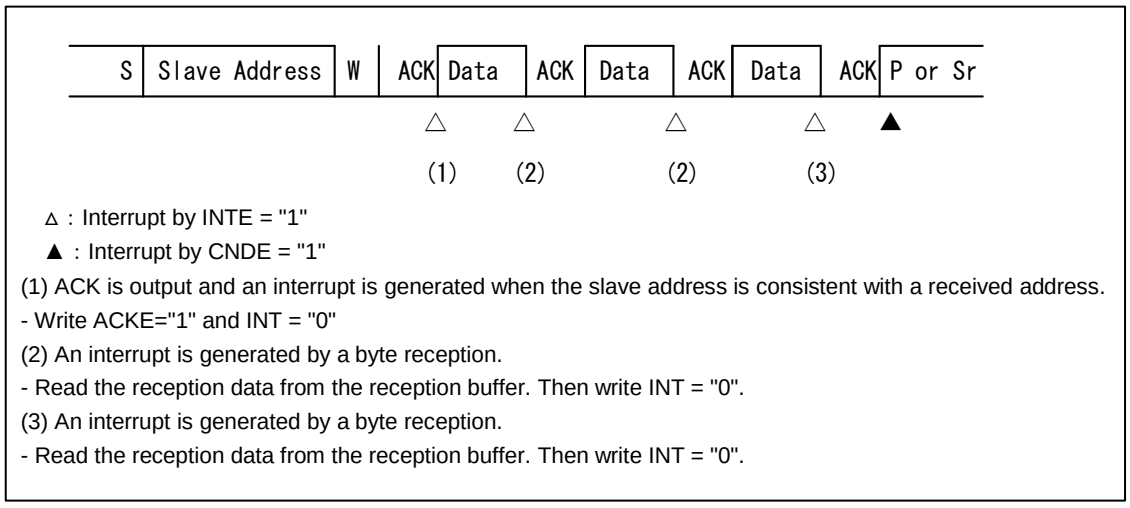
▲ : Interrupt by CNDE = "1"

(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 3-40 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 3-41 Slave Reception Interrupts 3 When FIFO Is Disabled (SSR:DMA= 0, IBCR:WSEL= 1, and IBSR:RSA= 0)

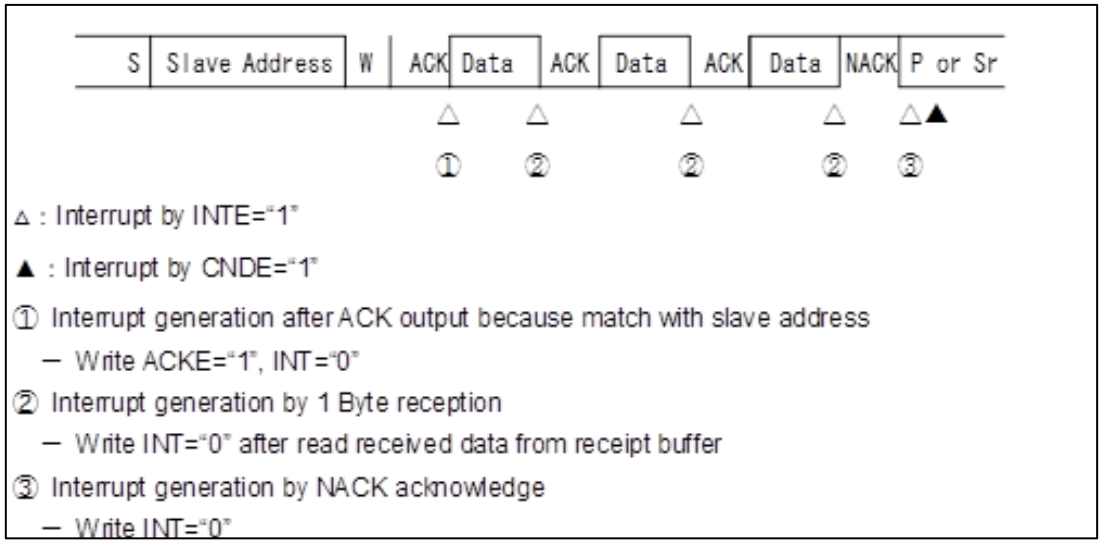


Figure 3-42 Slave Reception Interrupts 4 When the Reception FIFO Is Enabled (SSR:DMA= 0 and IBSR:RSA= 0)

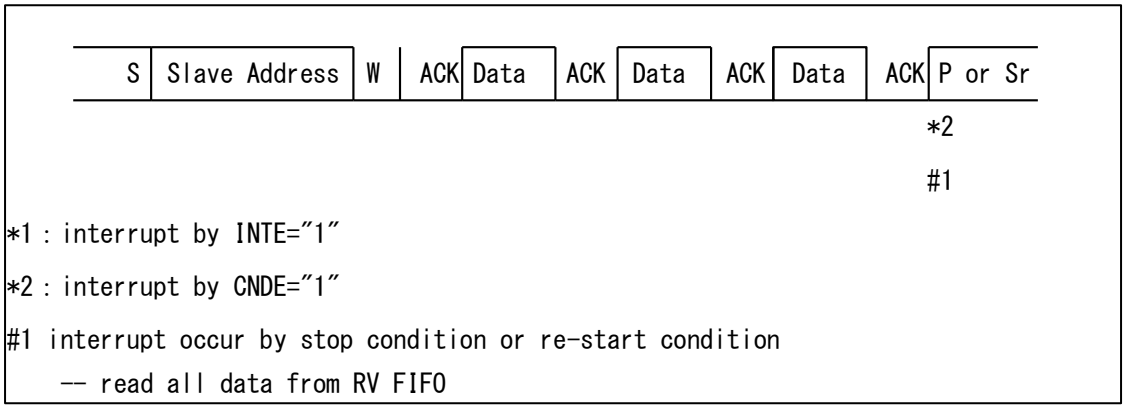


Figure 3-43 Slave Reception Interrupts 5 When the Reception FIFO Is Enabled (SSR:DMA= 0 and IBSR:RSA= 0)

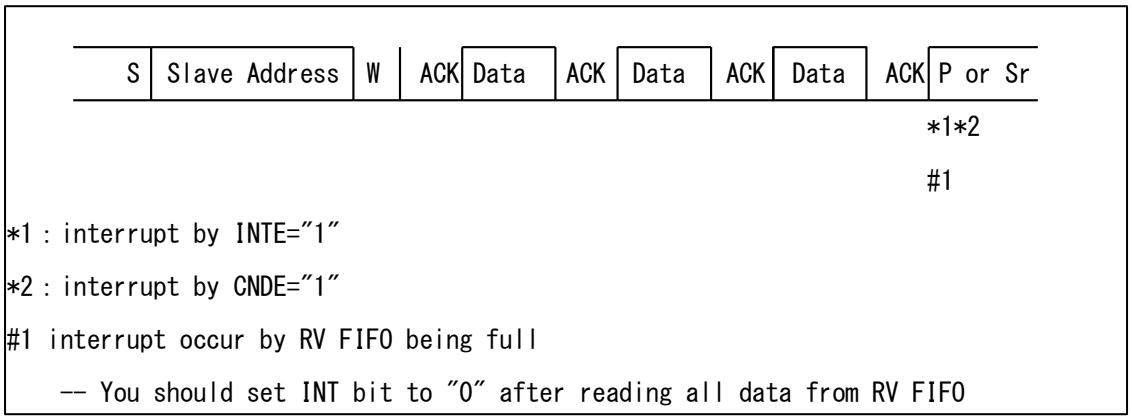
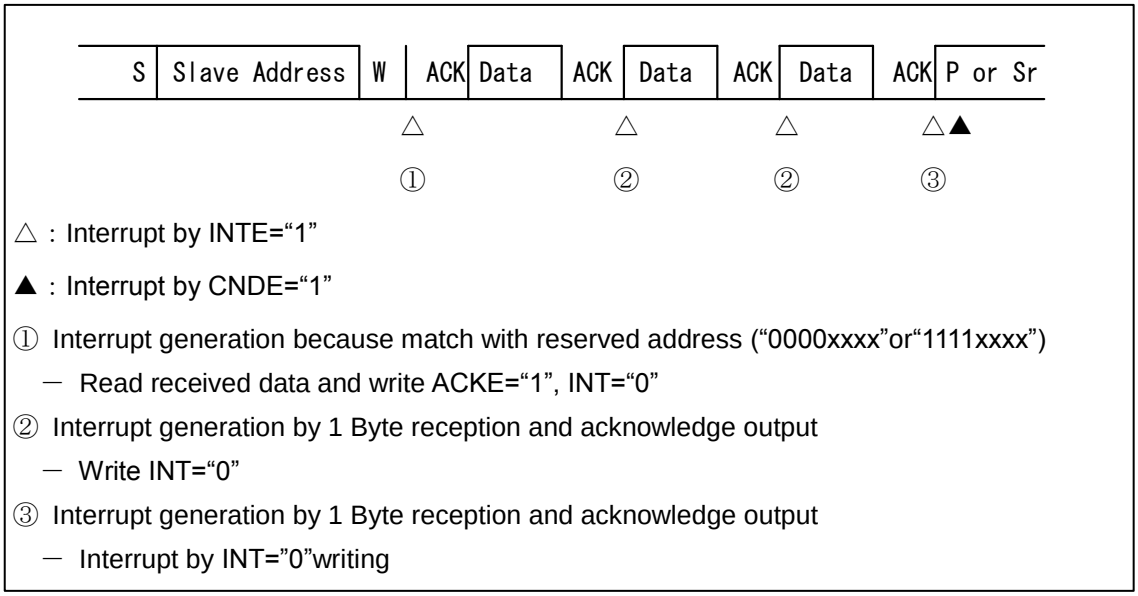


Figure 3-44 Slave Reception Interrupts 6 When FIFO Is Disabled (SSR:DMA= 0, IBCR:WSEL= 0, and IBSR:RSA= 1)



△ : 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"

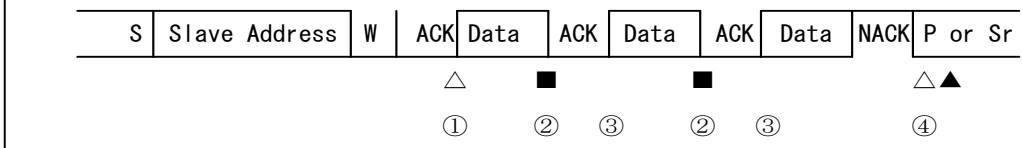
② Interrupt generation by 1 Byte reception and acknowledge output

— Write INT="0"

③ Interrupt generation by 1 Byte reception and acknowledge output

— Interrupt by INT="0"writing

Figure 3-45 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 3-46 Slave Reception Interrupts 8 When FIFO Is Disabled (SSR:DMA= 1, IBCR:WSEL= 1, and IBSR:RSA= 0)

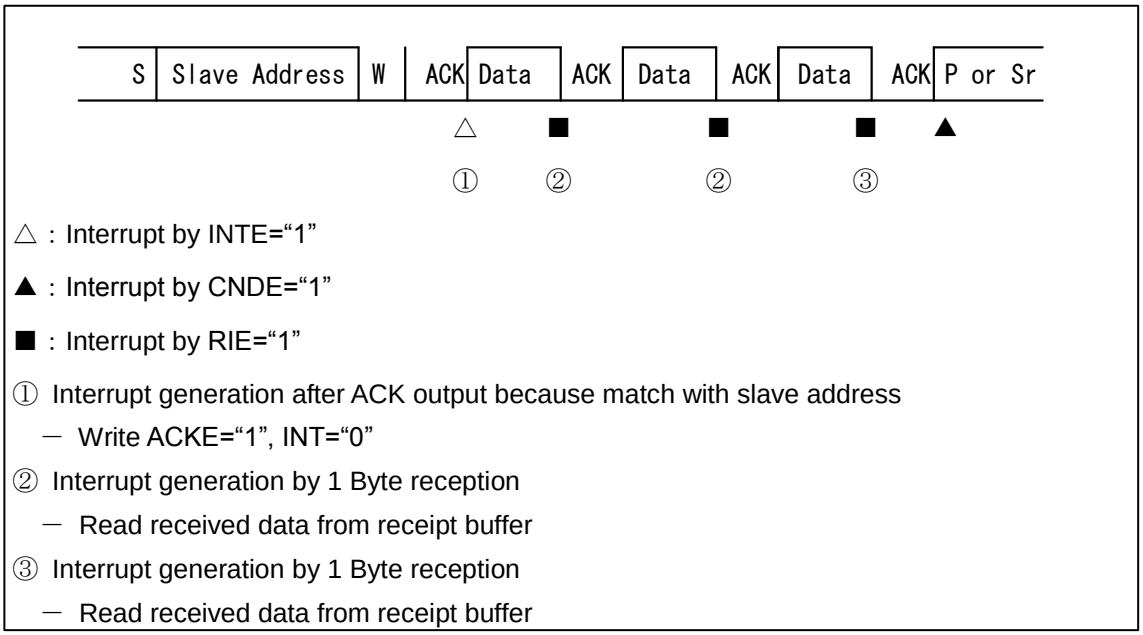


Figure 3-47 Slave Reception Interrupts 9 When FIFO Is Disabled (SSR:DMA= 1, IBCR:WSEL= 1, and IBSR:RSA= 0)

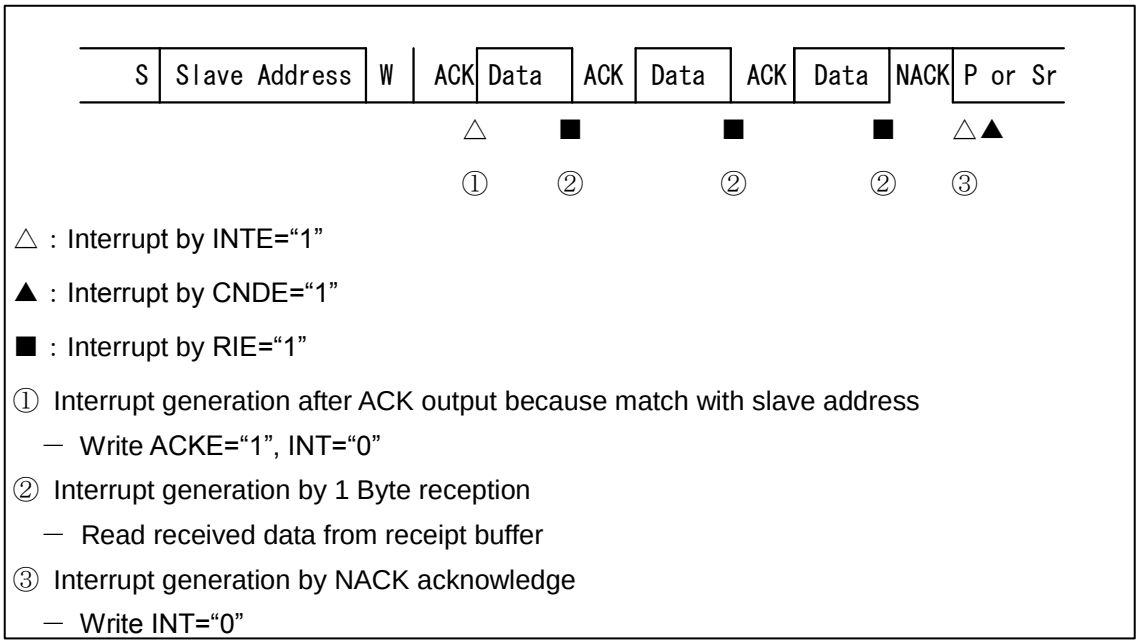


Figure 3-48 Slave Reception Interrupts 10 When the Reception FIFO Is Enabled (SSR:DMA= 1 and IBSR:RSA= 0)

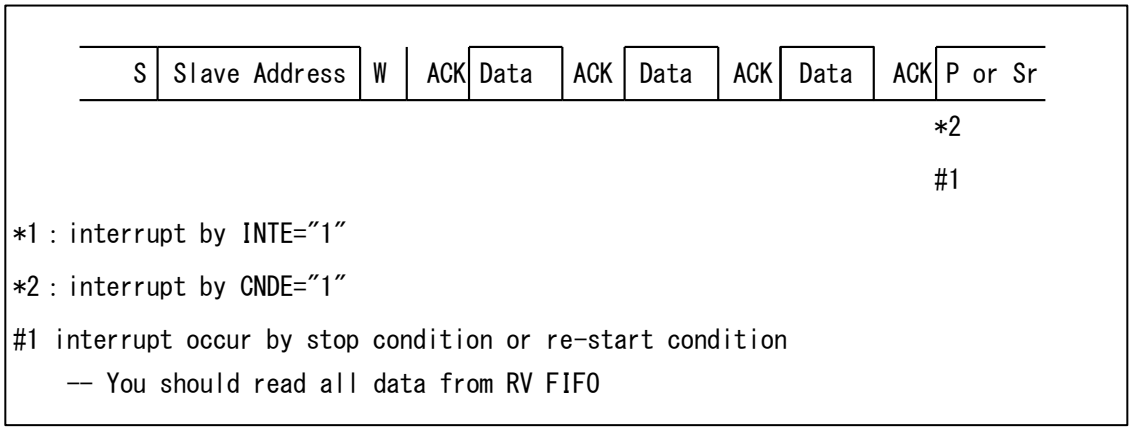


Figure 3-49 Slave Reception Interrupts 11 When the Reception FIFO Is Enabled (SSR:DMA= 1 and IBSR:RSA= 0)

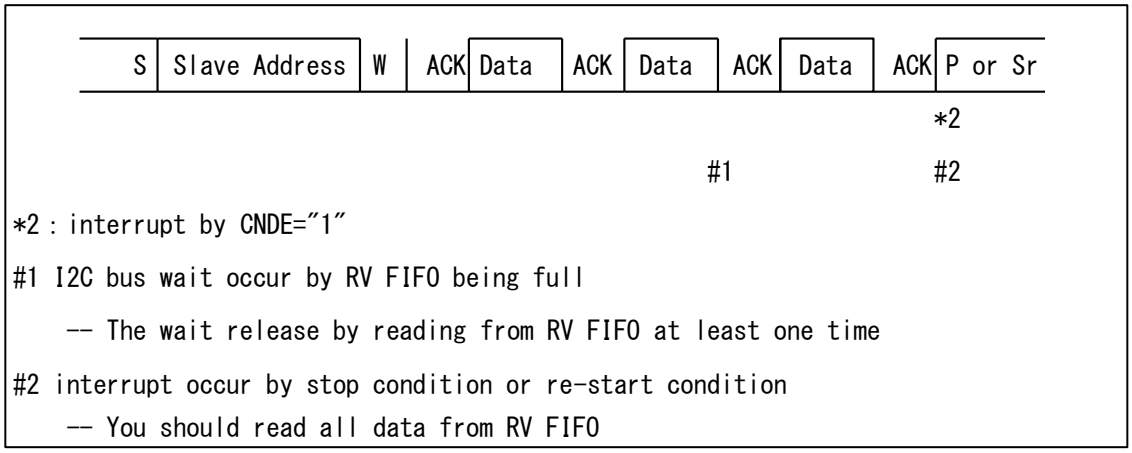
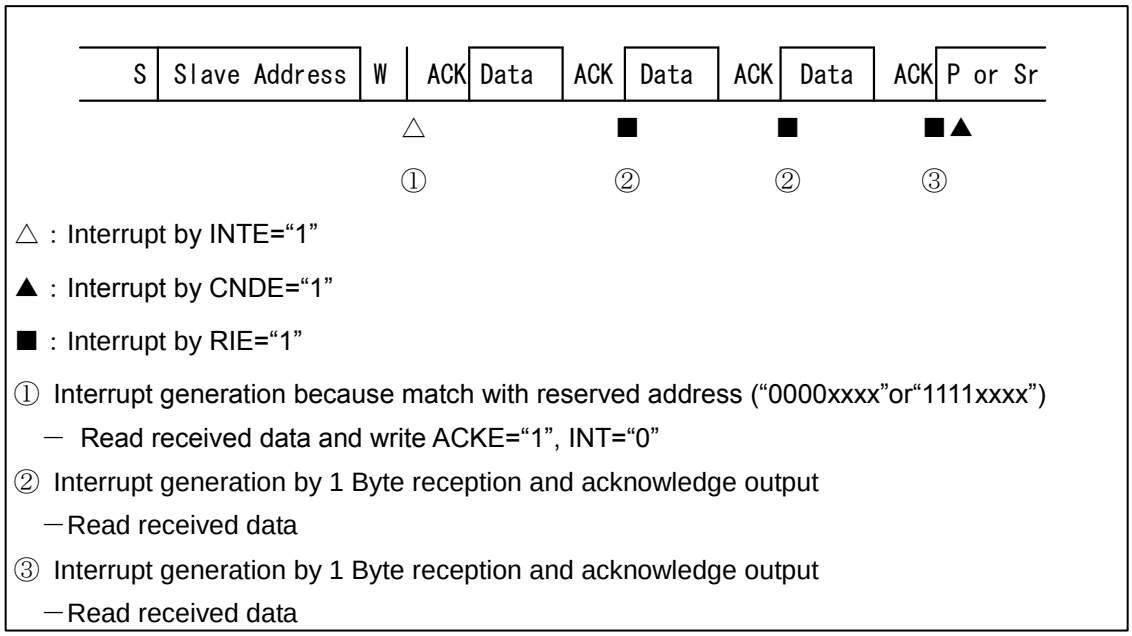


Figure 3-50 Slave Reception Interrupts 12 When FIFO Is Disabled (SSR:DMA= 1, IBCR:WSEL= 0, and IBSR:RSA= 1)



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 the writing of "0" to the IBCR:INT bit.

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

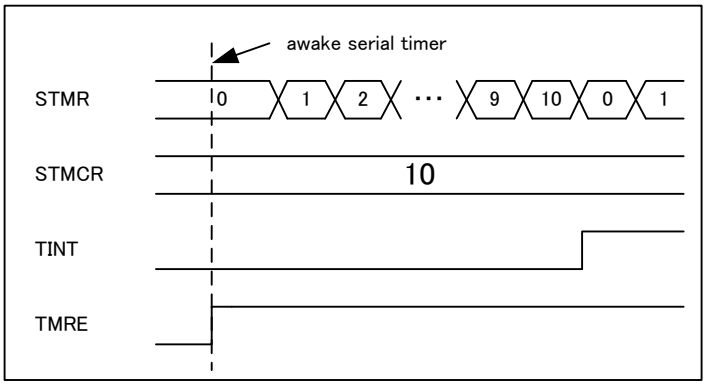
Activating the Serial Timer

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.

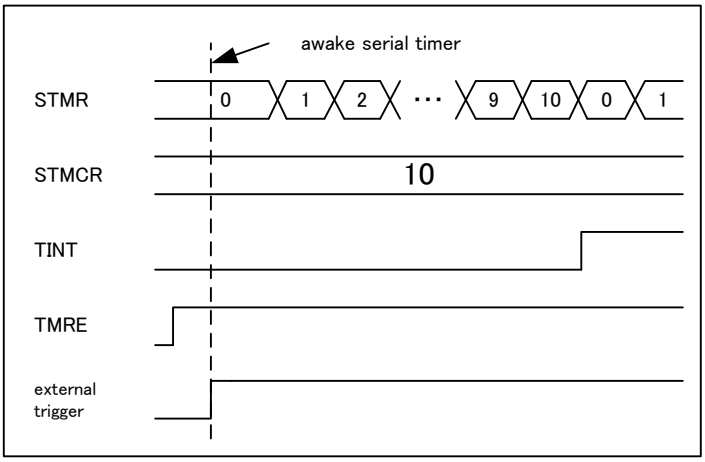
Figure 4-1 Activation by the Serial Timer Enable Bit (STMCR=10, SACSR:TRGE=0)



- 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)

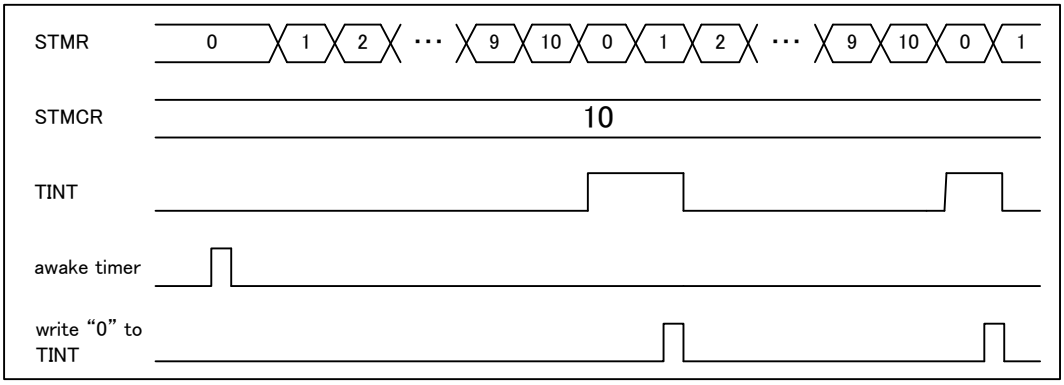


Figure 4-4 Serial Timer Initialization Flowchart

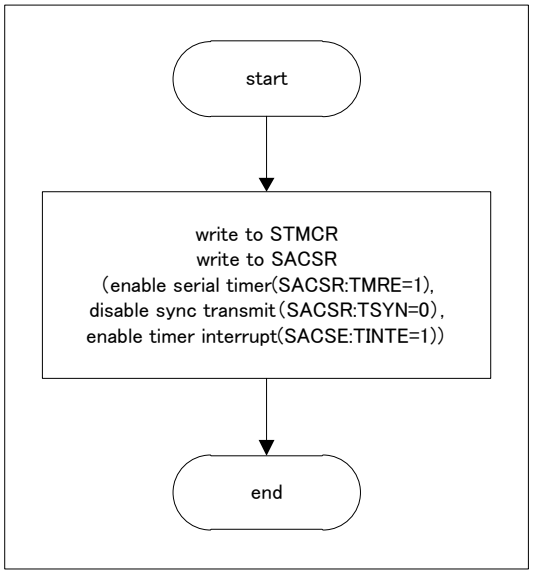
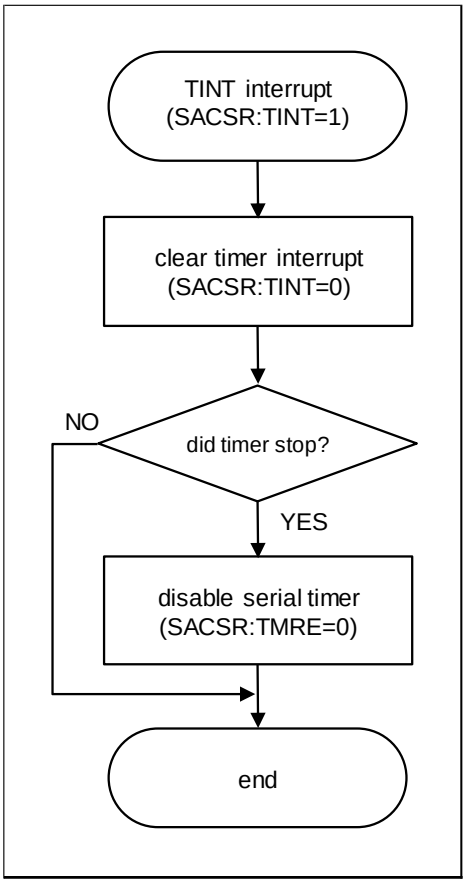


Figure 4-5 Serial Timer Interrupt Processing Flowchart



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 Value	10 MHz Value	16 MHz Value	20 MHz Value	24 MHz Value	32 MHz 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 Value	48 MHz Value	72 MHz Value	80 MHz 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.

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.

The block size is set in the transmission/reception block size register.

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. ACK Mode

This section explains operation in ACK mode for block transfer.

Reception Block Transfer

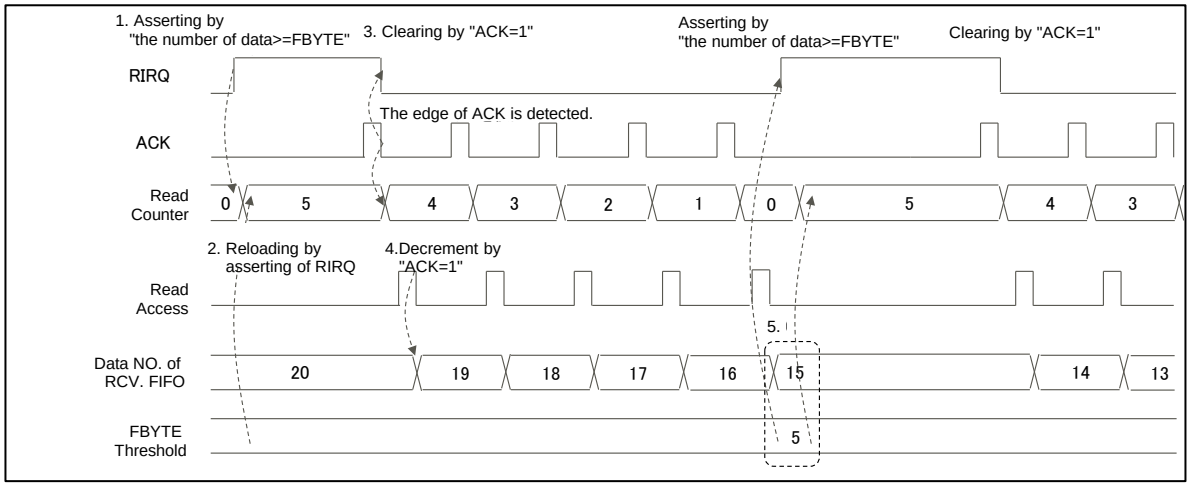
Reception block transfer operates as follows.

A reception block transfer request (RIRQ) is asserted when the data count in the reception FIFO is same or more than 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 reception acknowledgment is input. The read of the reception data decrements the Read counter.

If the data count in the reception FIFO is same or more than 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.

Figure 6-1 Transmission Block Transfer in ACK Mode



Transmission Block Transfer

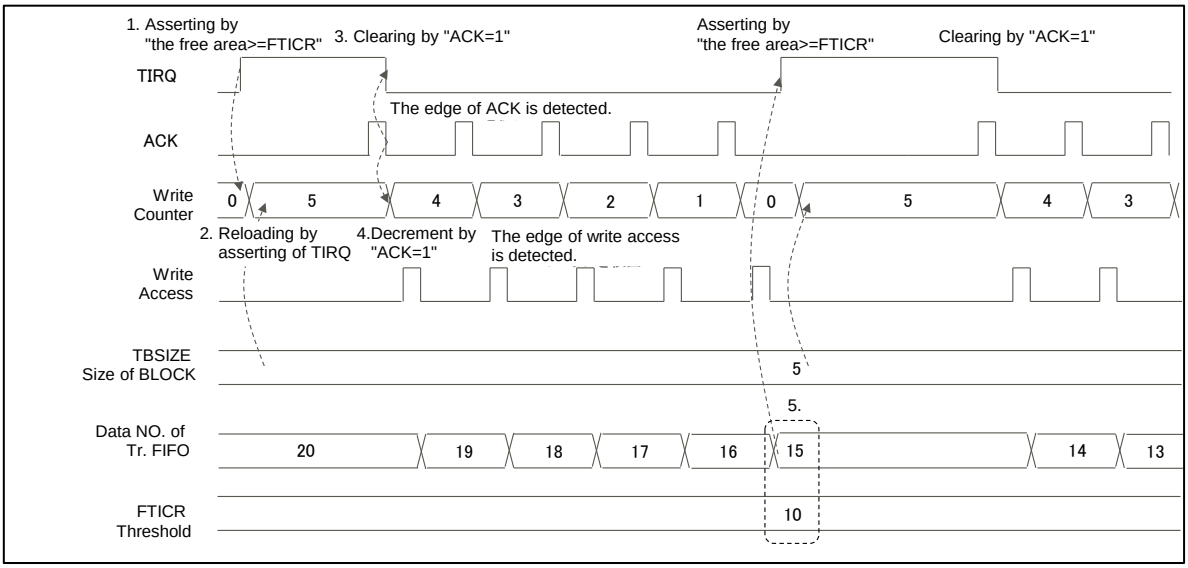
Transmission block transfer operates as follows.

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 a transmission acknowledgment is input. The read of the transmission data 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.

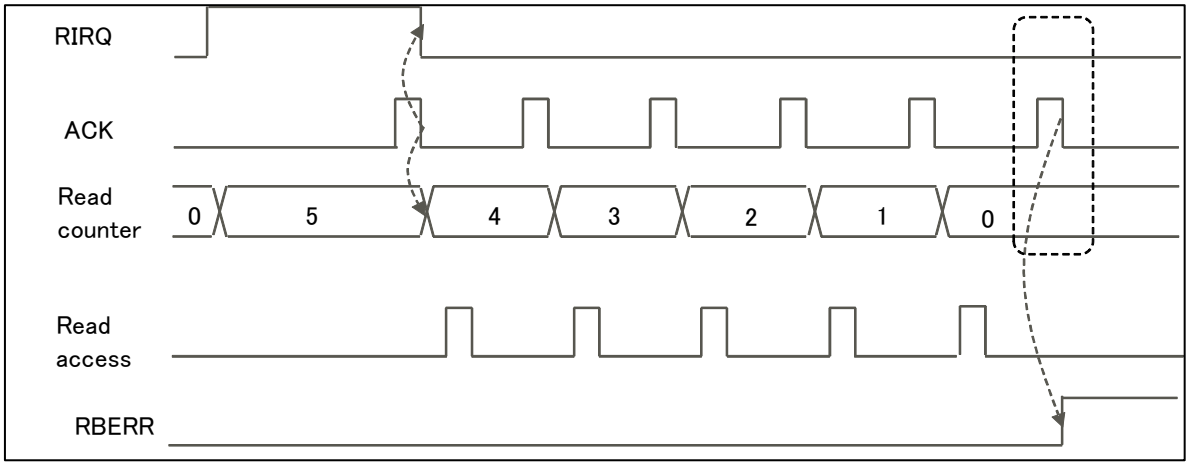
Figure 6-2 Transmission Block Transfer in ACK Mode



Reception Block Transfer Error

Inputting an acknowledgment before asserting 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.

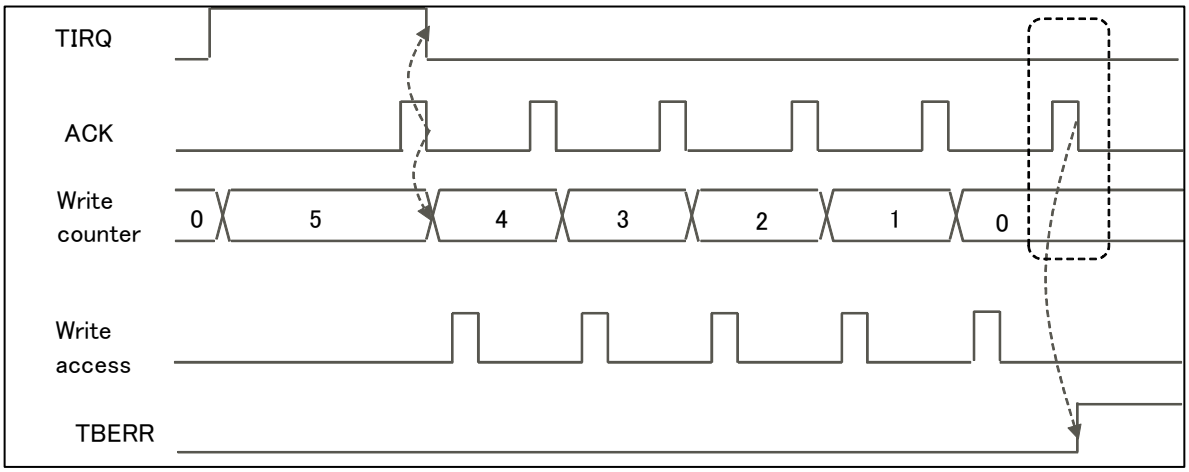
Figure 6-3 Reception Block Transfer Error in ACK Mode



Transmission Block Transfer Error

Inputting an acknowledgment before asserting the next transmission block transfer request (TIRQ) while the Write counter is "0" causes a transmission block transfer error. Then, the FSR:TBERR bit is asserted.

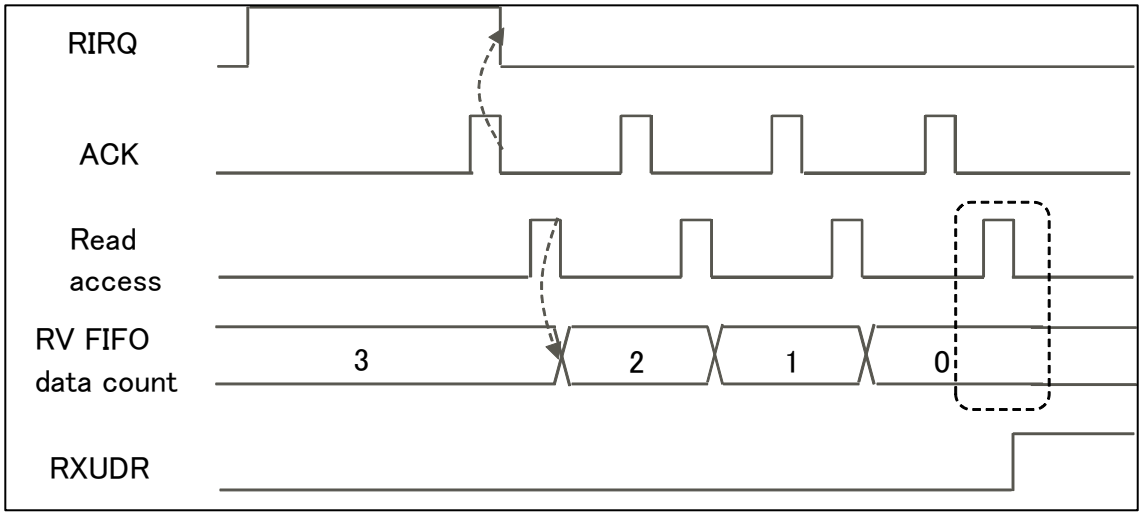
Figure 6-4 Transmission Block Transfer Error in ACK Mode



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.

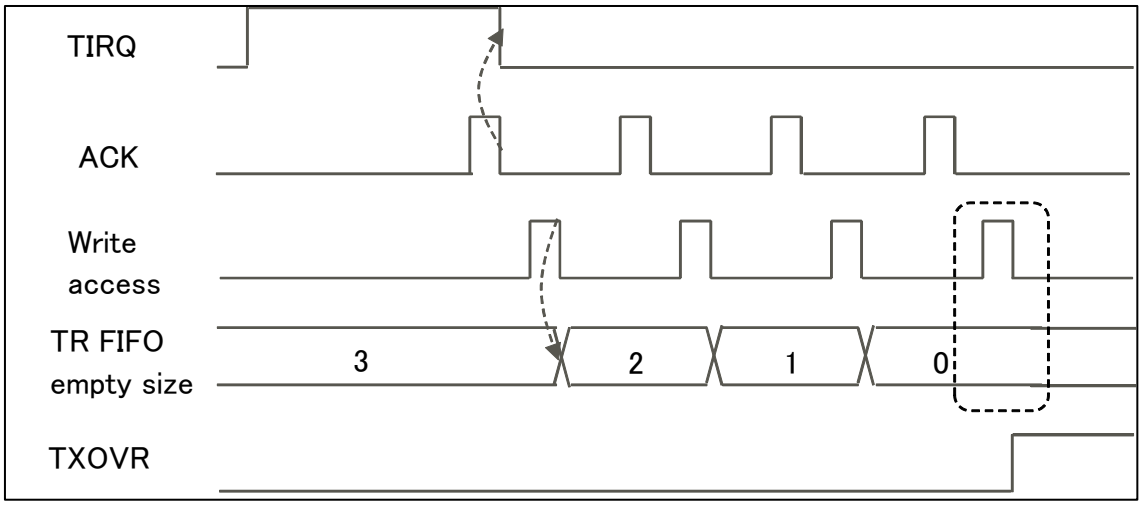
Figure 6-5 Reception FIFO Under Run in ACK Mode



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 ACK Mode



6.2. RW Access Mode

This section explains operation in RW access mode for block transfer.

Reception Block Transfer

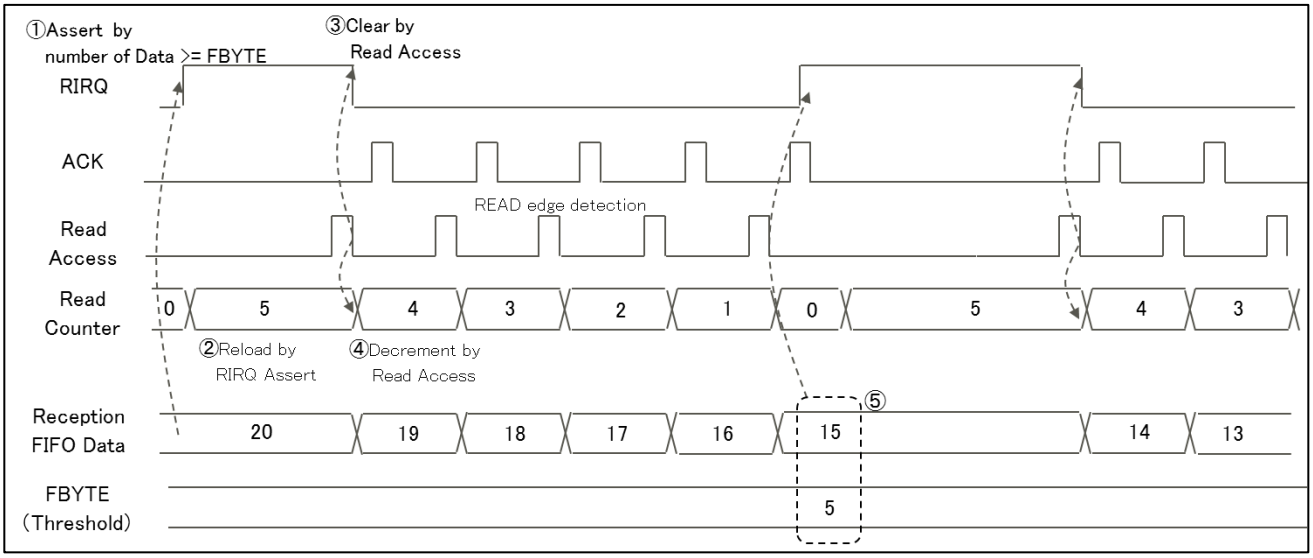
Reception block transfer operates as follows.

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 read access to the reception FIFO occurs. The read access to the reception FIFO 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.

Figure 6-7 Reception Block Transfer in RW Access Mode



Transmission Block Transfer

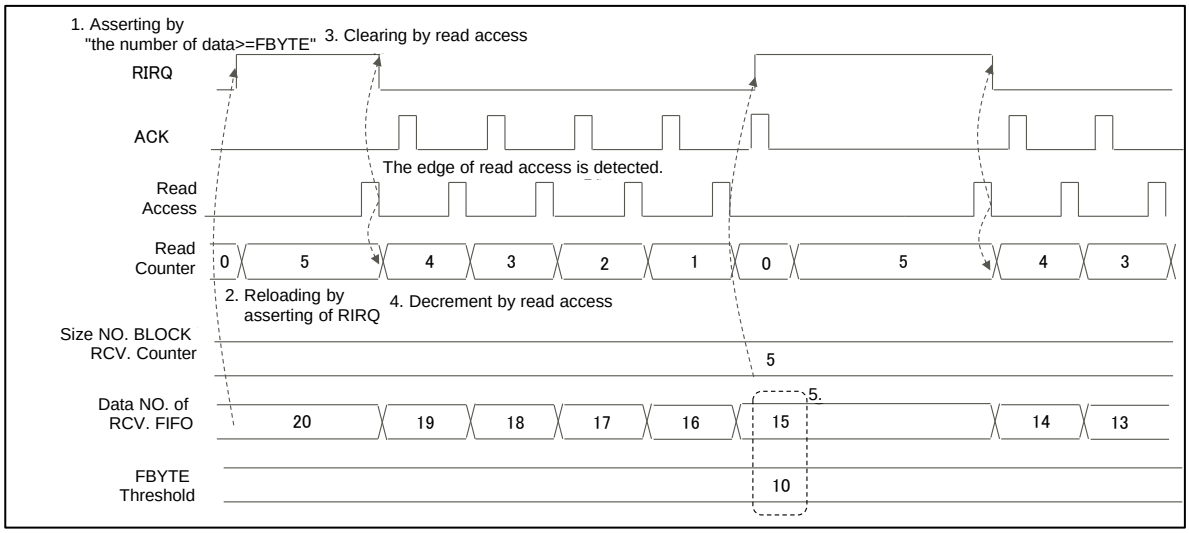
Transmission block transfer operates as follows.

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.

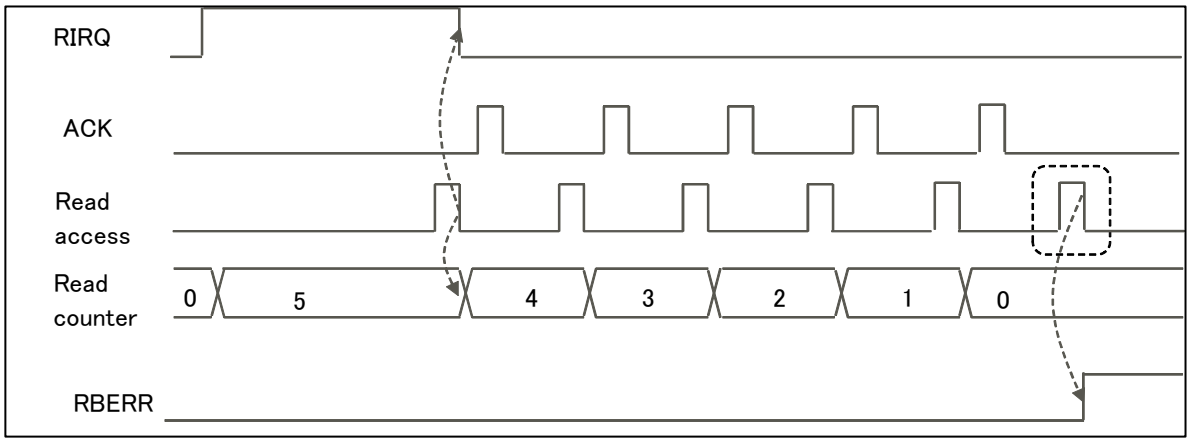
Figure 6-8 Transmission Block Transfer in RW Access Mode



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.

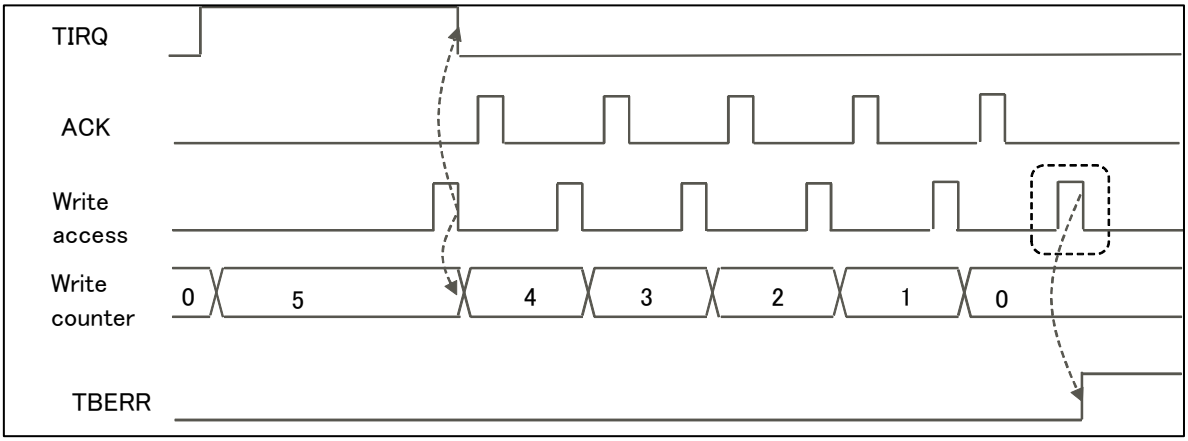
Figure 6-9 Reception Block Transfer Error in RW Access Mode



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 FSR:TBERR bit is asserted.

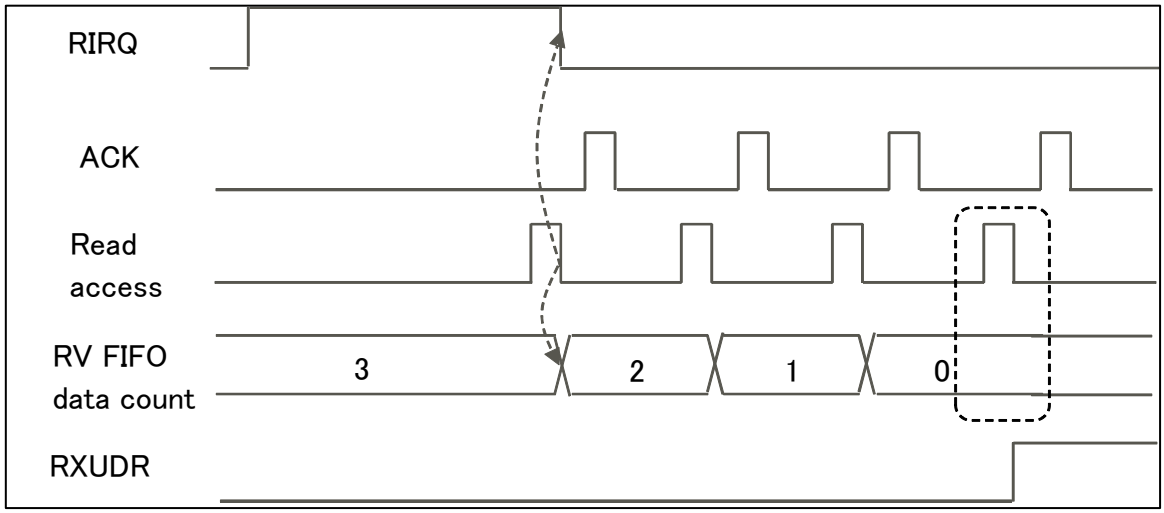
Figure 6-10 Transmission Block Transfer Error in RW Access Mode



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.

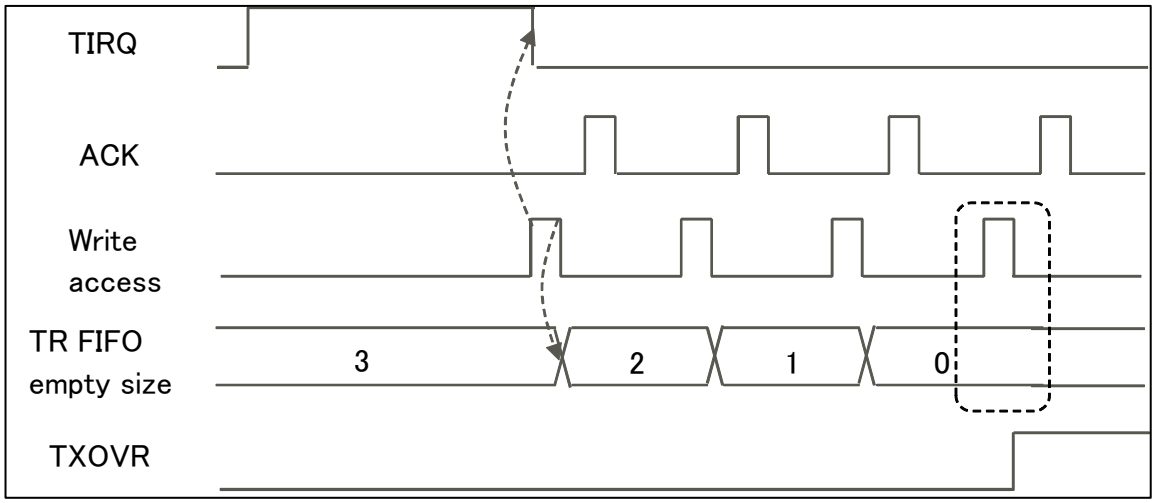
Figure 6-11 Reception FIFO Under Run in RW Access Mode



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 6-12 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

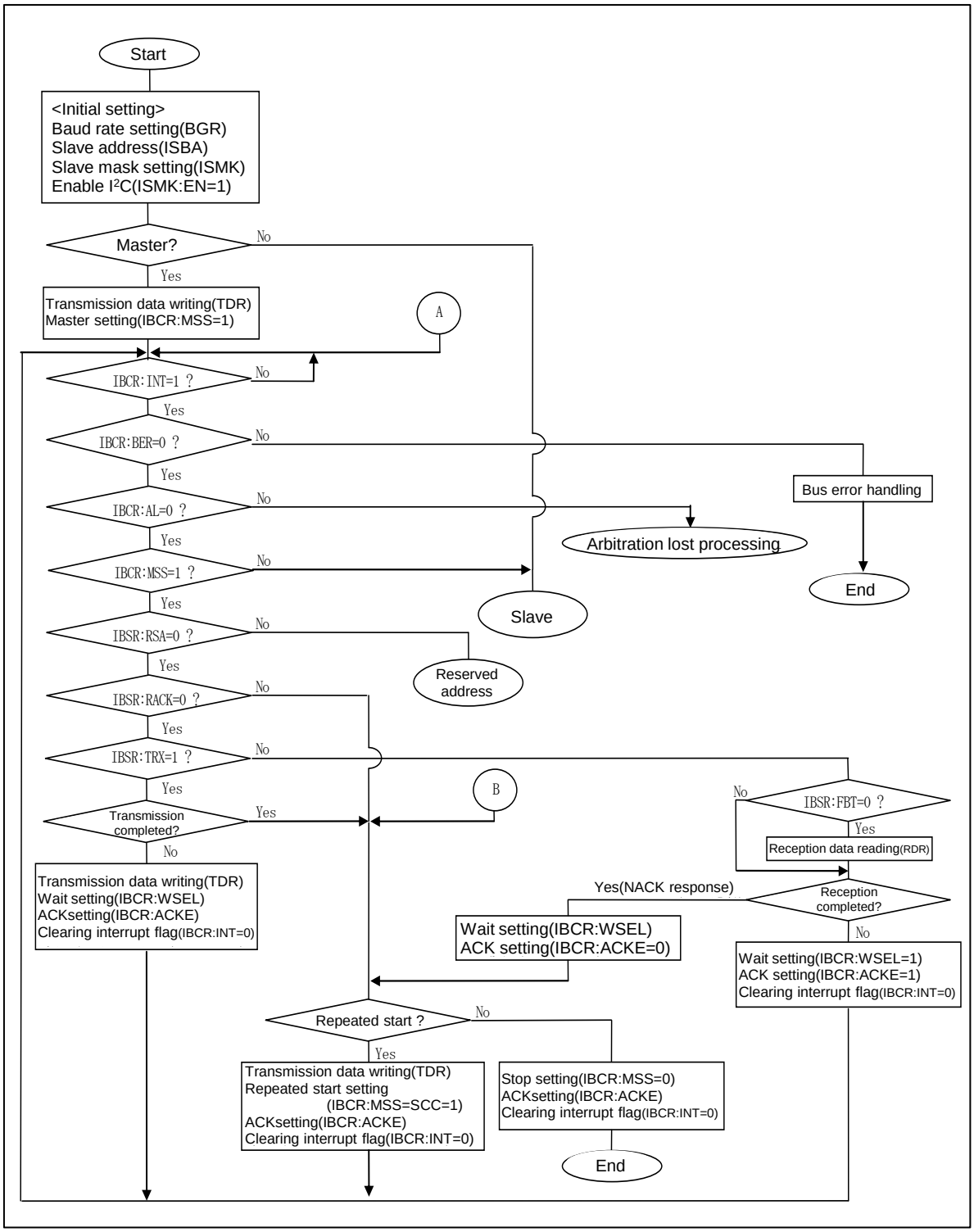


Figure 7-2 I²C Flowchart Example Where DMA Mode Is Disabled (SSR:DMA=0) (FIFO Not Used) 2/3

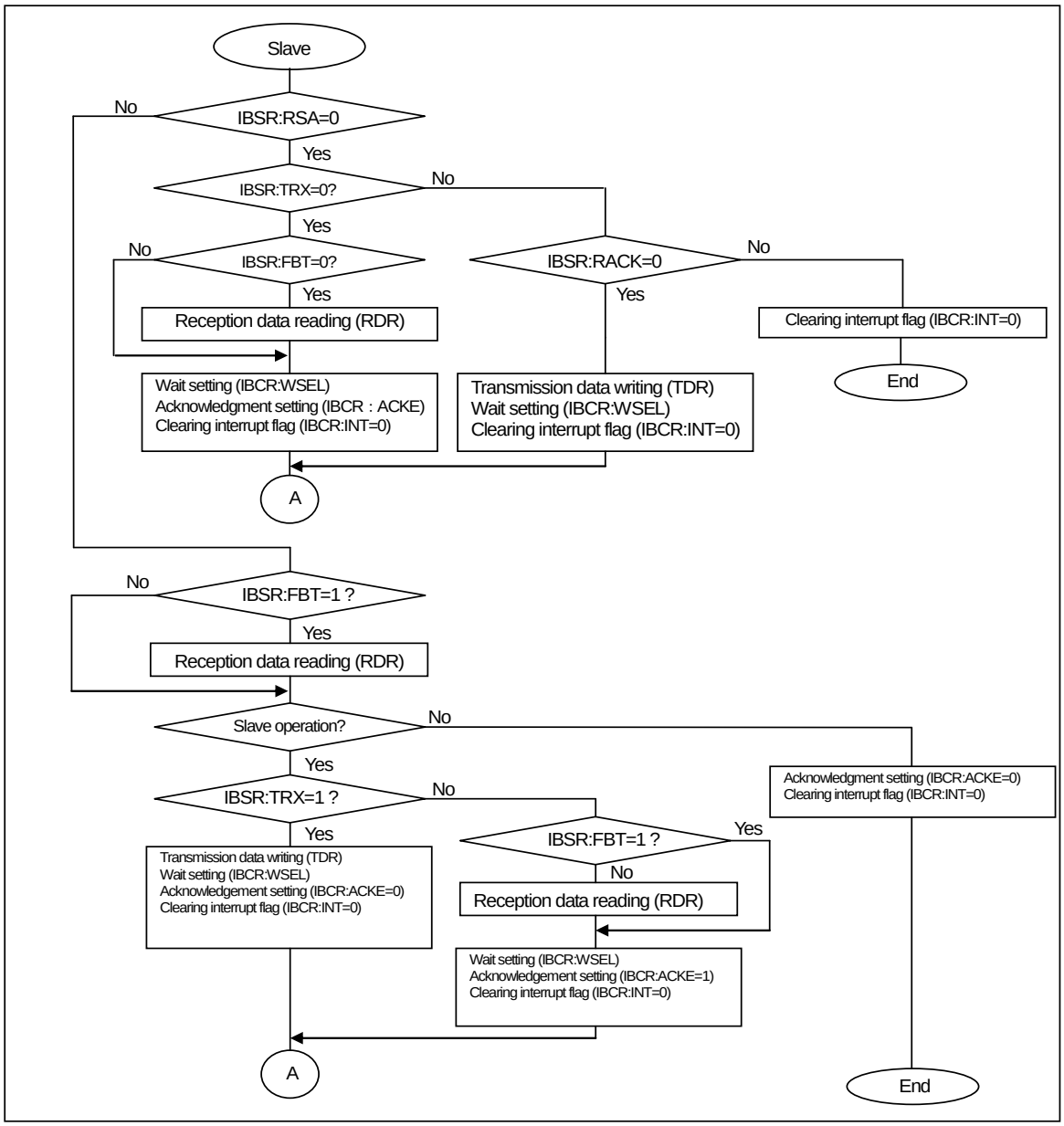
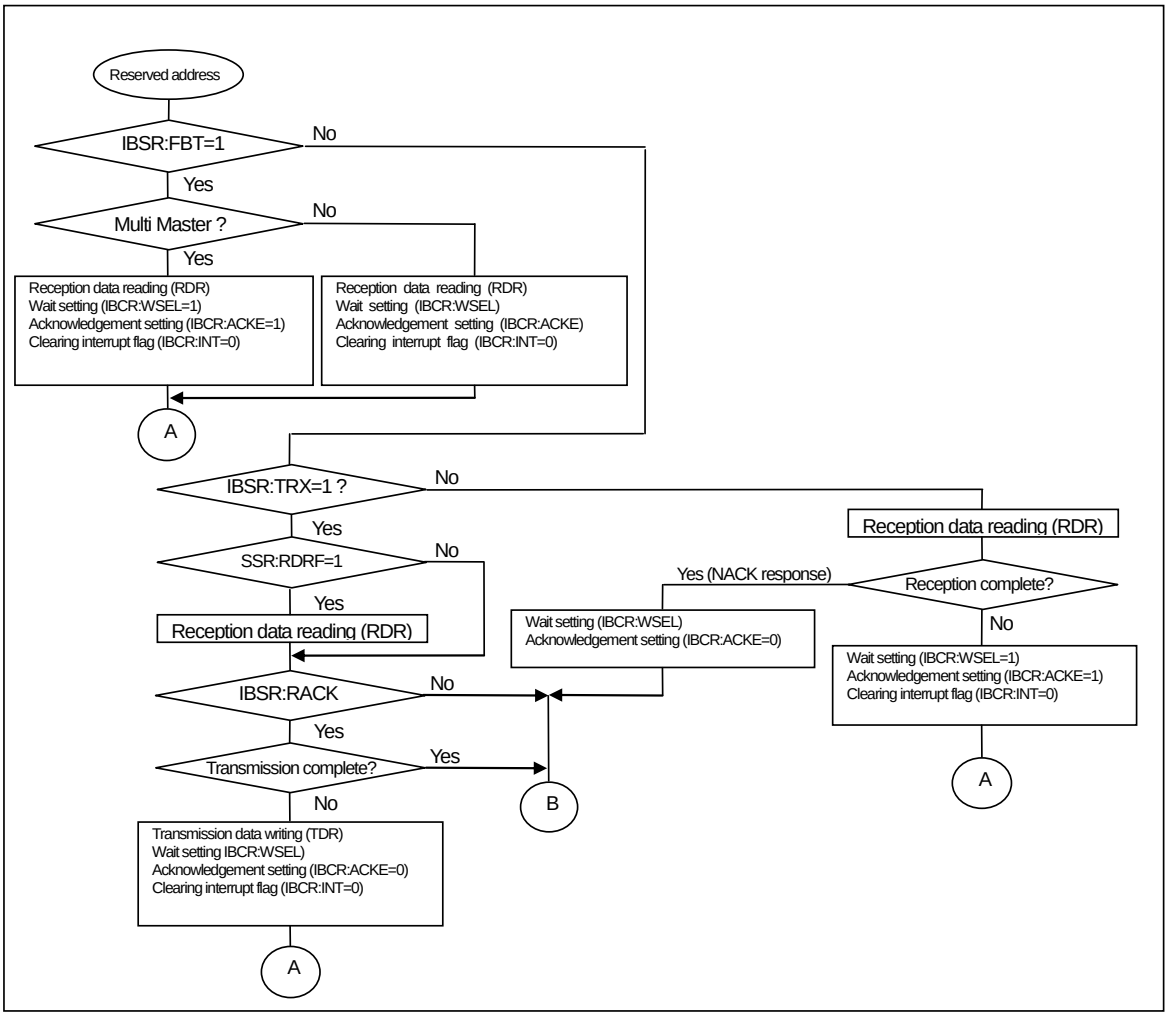


Figure 7-3 I²C Flowchart Example Where DMA Mode Is Disabled (SSR:DMA=0) (FIFO Not Used) 3/3



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

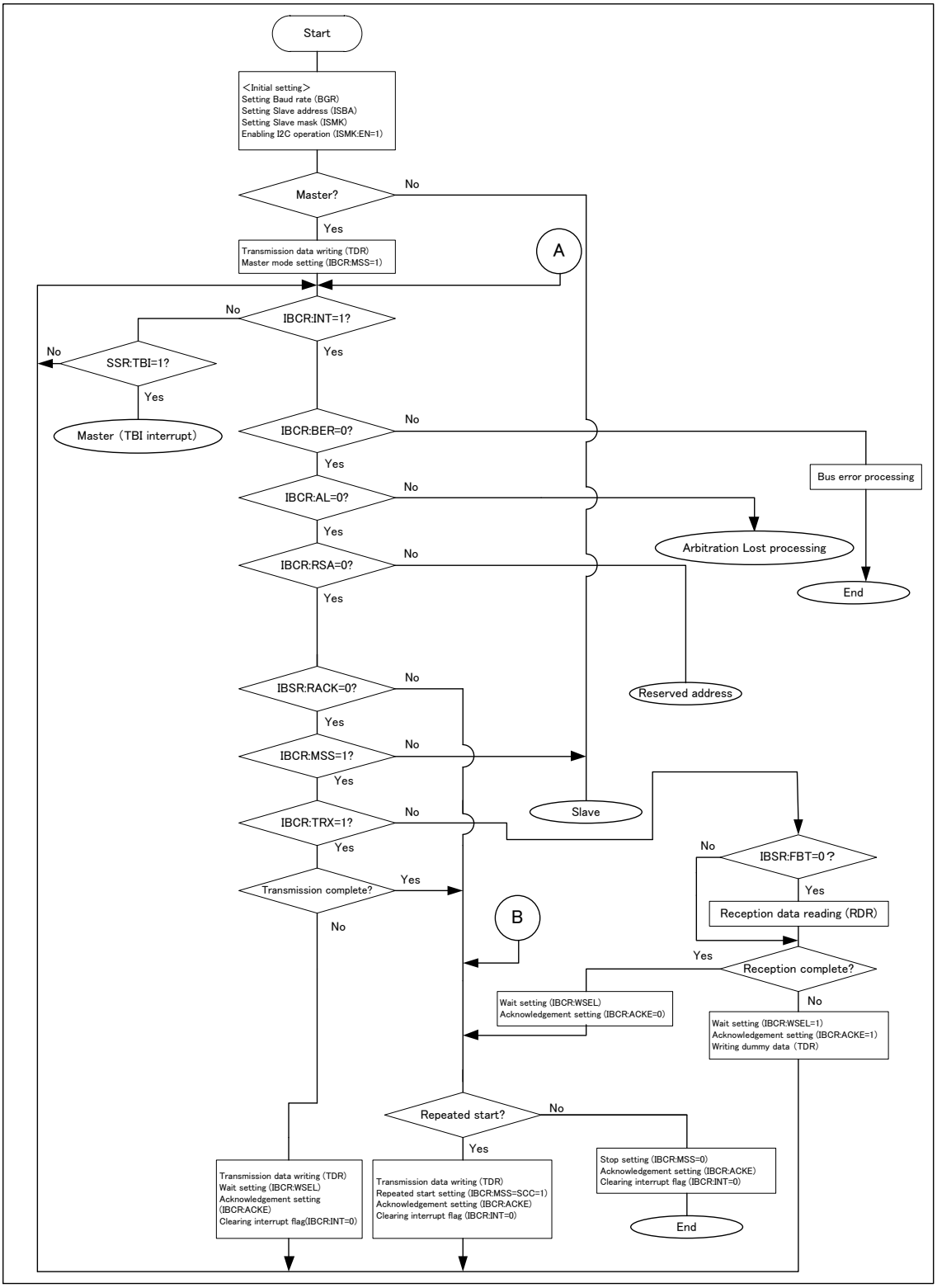


Figure 7-5 I²C Flowchart Example Where DMA Mode Is Enabled (SSR:DMA=1) (FIFO Not Used) 2/5

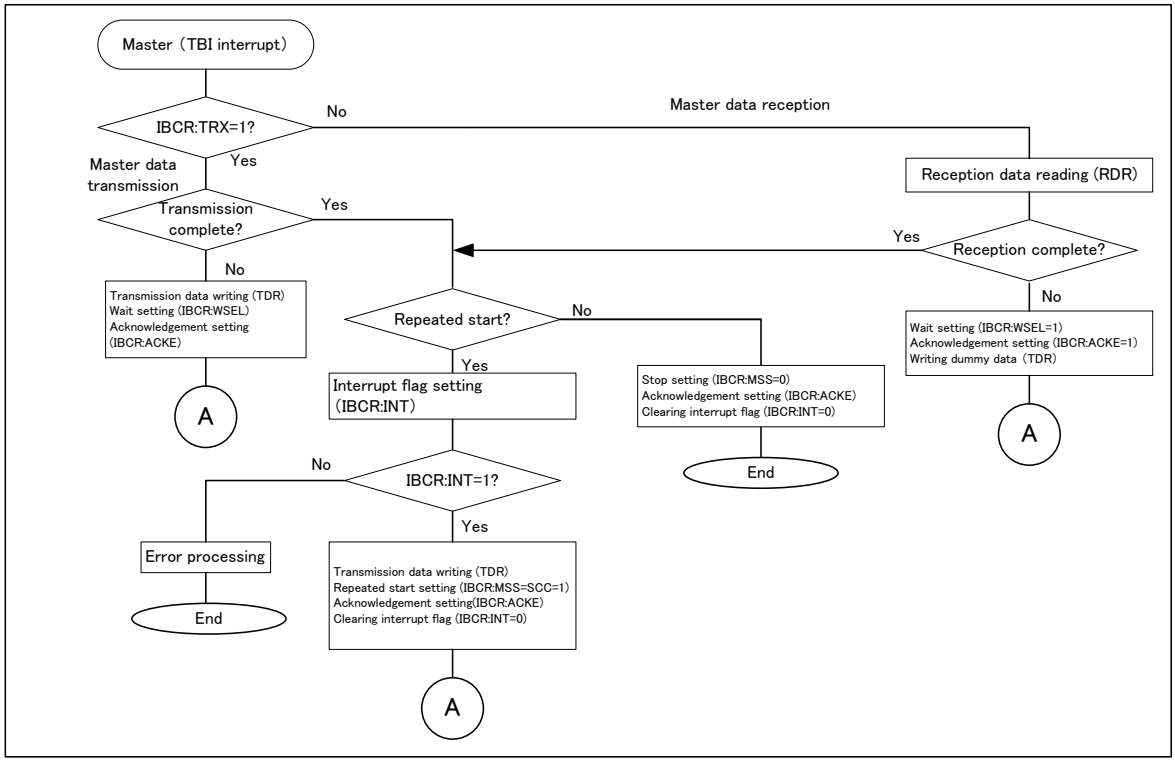


Figure 7-6 I²C Flowchart Example Where DMA Mode Is Enabled (SSR:DMA=1) (FIFO Not Used) 3/5

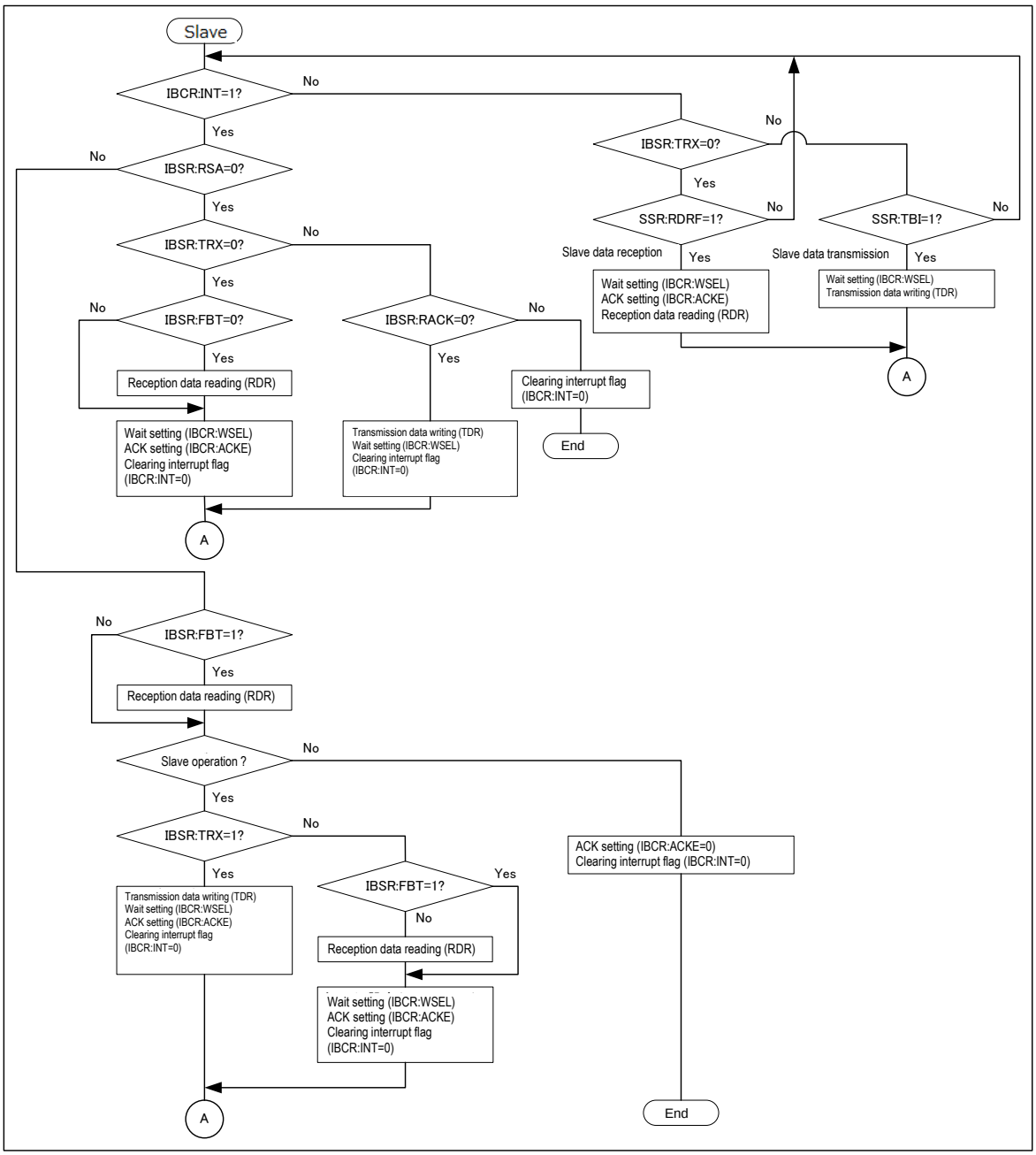


Figure 7-7 I²C Flowchart Example Where DMA Mode Is Enabled (SSR:DMA=1) (FIFO Not Used) 4/5

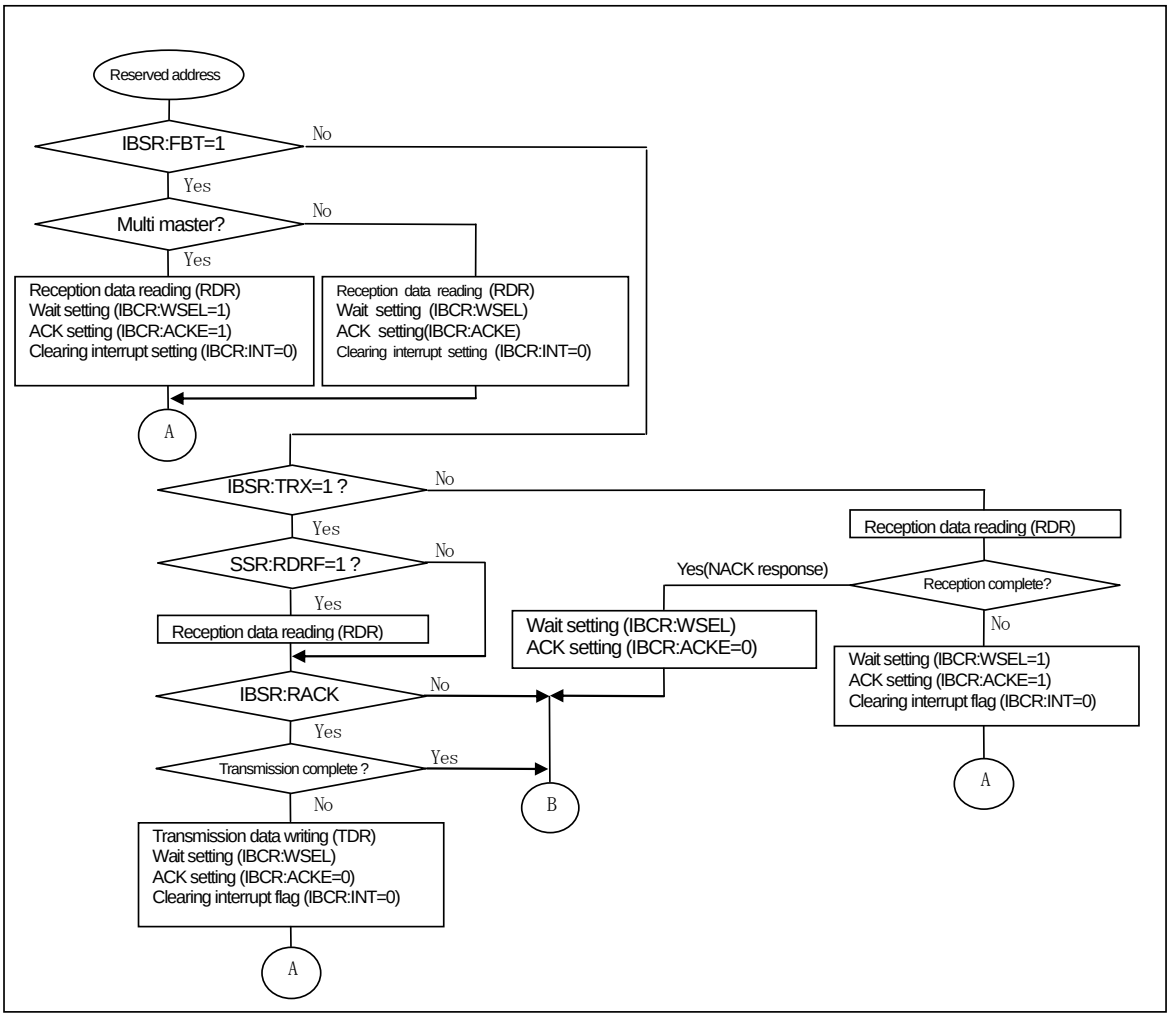
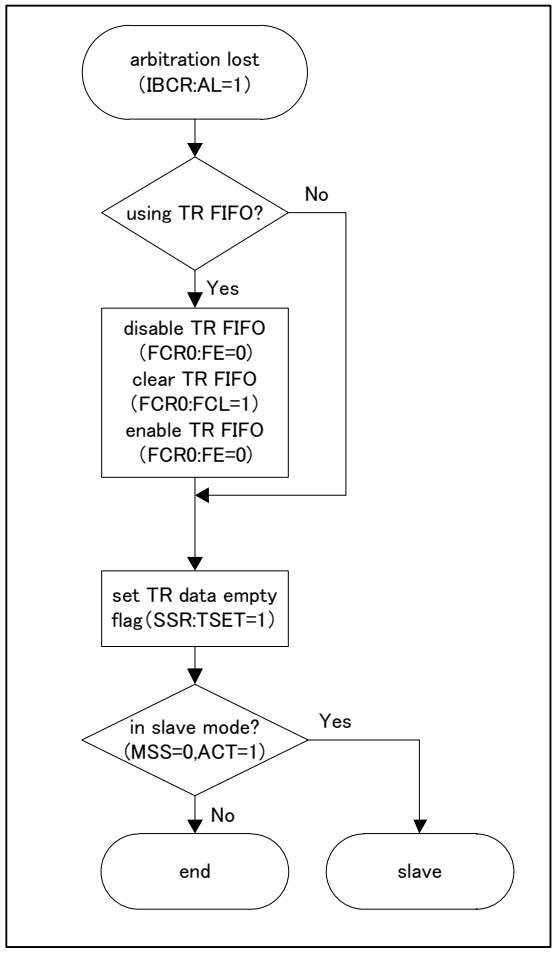


Figure 7-8 Arbitration Lost Processing Flowchart Example 5/5



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
	-	-

*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. I²C 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. I²C 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 I²C 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^{2}C bus.
- When DMA mode is enabled (SSR:DMA= 1), this bit selects whether to wait for the I^{2}C 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^{2}C 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
Protectoin 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.

"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).

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

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.

[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 reading of the reception FIFO empties it during use, this bit is cleared to "0".
- 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 reception is disabled (SCR:RXE= 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.

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 ISMK: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 address bit for which "0" is set in ISMK is excluded from the comparison.

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 8-14 shows the bit configuration of baud rate generator registers 1 and 0 (BGR1, BGR0).

Figure 8-14 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 BGR1/0 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 8-15 shows the bit configuration of FIFO control register 1 (FCR1).

Figure 8-15 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 FCR0:FLST bit 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 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 FRC1C: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:

- *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.*

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 8-16 shows the bit configuration of FIFO control register 0 (FCR0).

Figure 8-16 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

- There is a FIFO reset (writing "1" to FCL).
- "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 FBYTE1 register is set to 0.*

[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.*

[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 8-17 shows the bit configuration of the FIFO byte register (FBYTE).

Figure 8-17 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".
- 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.*
- *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.*
- *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.*

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 8-18 shows the bit configuration of the transmission FIFO interrupt control register (FTICR).

Figure 8-18 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 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.

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. 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).
- 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 8-19 shows the bit configuration of the extended control register (ECR).

Figure 8-19 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 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.

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 8-20 shows the bit configuration of the extended status register (ESR).

Figure 8-20 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:

- *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".*
- *This bit is set only when the reception FIFO is used.*

[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:

- *A software reset (SCR:UPCL= 1) resets this bit to "0".*
- *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.*

[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 (SSR: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:

- *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".*

[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", and transmission DMA transfer is stopped. If transmission interrupts are enabled (SSR: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:

- *A software reset (SCR:UPCL= 1) resets this bit to "0".*
- *If the transmission block transfer enable (ECR:TXBLKEN) is "0", this bit is "0".*

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 8-21 shows the bit configuration of the transmission block size register (TBSIZE).

Figure 8-21 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 8-22 shows the bit configuration of the I²C bus control clear register (IBCRC).

Figure 8-22 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 8-23 shows the bit configuration of the serial mode clear register (SMRC).

Figure 8-23 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 8-24 shows the bit configuration of the serial status clear register (SSRC).

Figure 8-24 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 8-25 shows the bit configuration of the I²C bus status clear register (IBSRC).

Figure 8-25 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 8-26 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. Bit Configuration of FIFO Control Clear Register 1 (FCR1C)

Figure 8-27 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 8-28 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 8-29 shows the bit configuration of the extended status clear register (ESRC).

Figure 8-29 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 8-30 shows the bit configuration of the I²C bus control set register (IBCRS).

Figure 8-30 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 8-31 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 8-32 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 8-33 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
Protectoin 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 8-34 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 8-35 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: 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. Debug Mode
6. Interrupts from the Base Timer
7. Start of DMA Controller (DMAC)
8. Registers of the Base Timer
9. Notes on Using the Base Timer
10. Base Timer Description by Function Mode

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 in 16-bit periods.

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 (PSDR), an ADC Trigger Value Setting Register (ADTR), and a pin controller.

The down counter will be described in 16 bit formats when the 16-bit PWM Timer is 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 3 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 (ADTR) matches the count value of the 16-bit down counter.

The external timer match starting can start a timer, when a match of the value set as the External Timer Compare Data Register (ETCDR) and an external timer value is detected.

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 3 types of external events (rising-edge, falling-edge, and both-edges detection).

The external timer match starting can start a timer, when a match of the value set as the External Timer Compare Data Register (ETCDR) and an external timer value is detected.

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 3 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.

The external timer match starting can start a timer, when a match of the value set as the External Timer Compare Data Register (ETCDR) and an external timer value is detected.

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).

The supported measurement modes:

- "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.

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

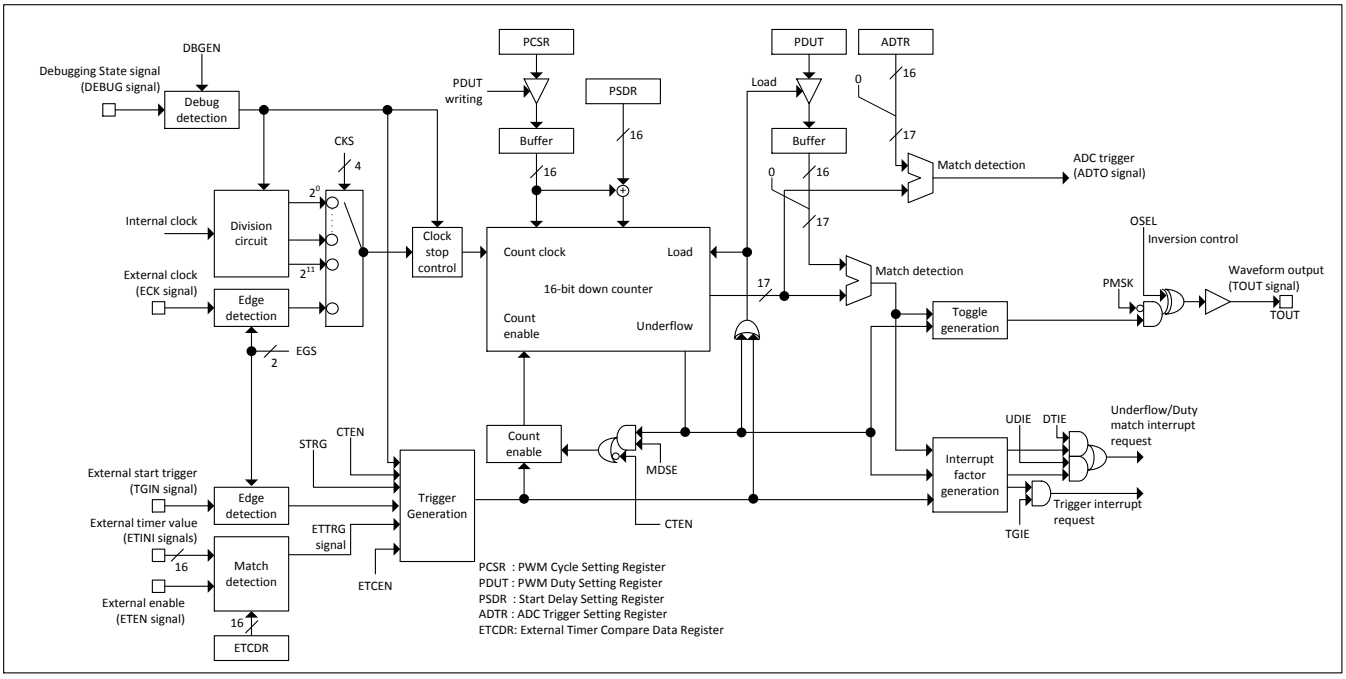


Figure 2-2 Block Diagram of the 16-Bit PPG Timer

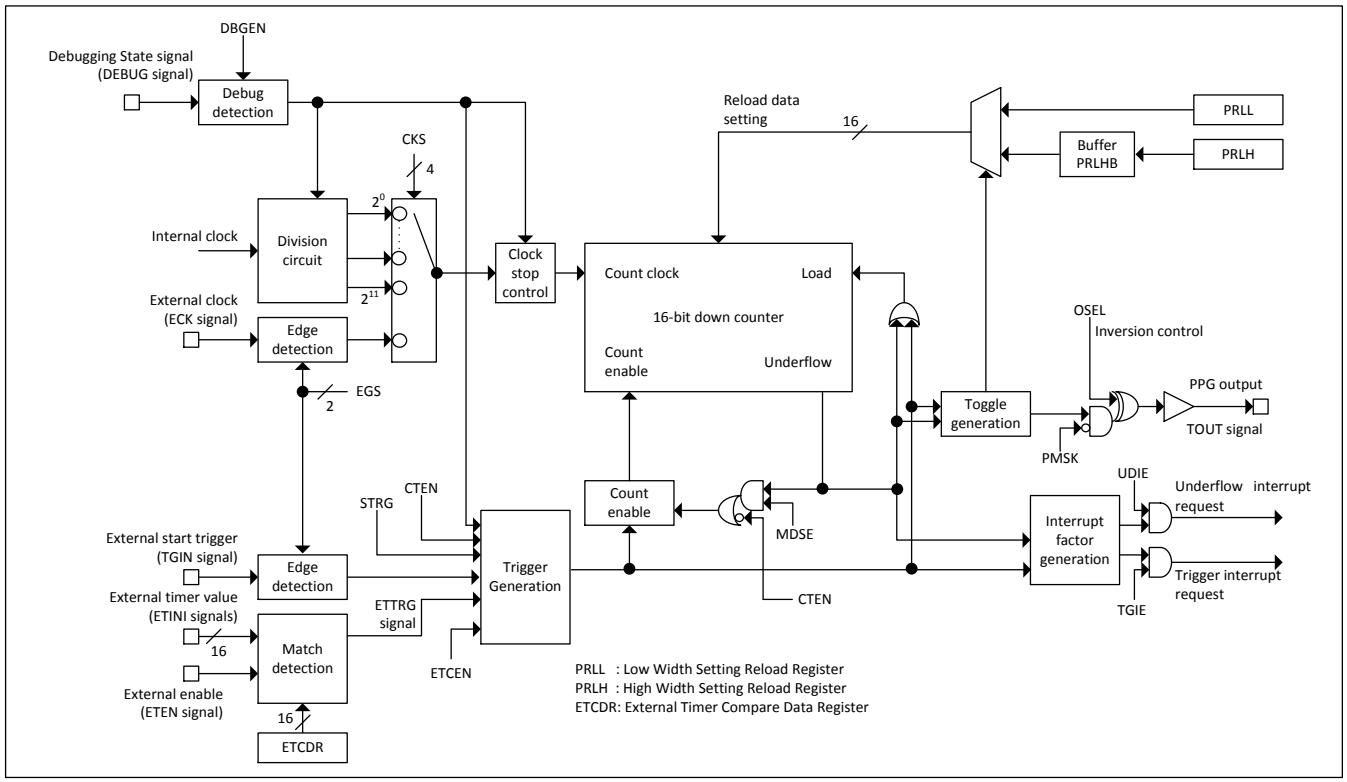


Figure 2-3 Block Diagram of the 16/32-Bit Reload Timer (ch1, ch0)

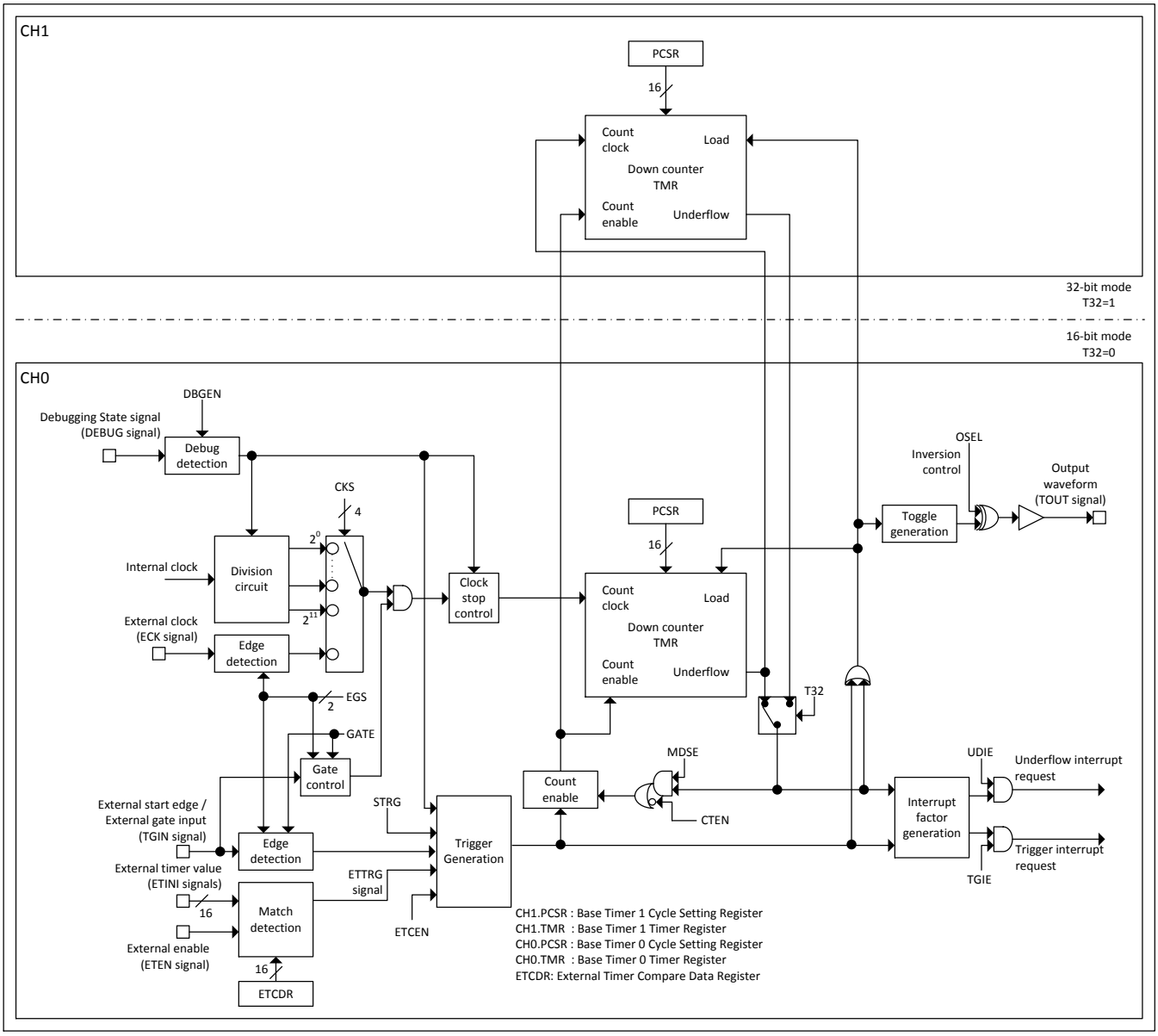
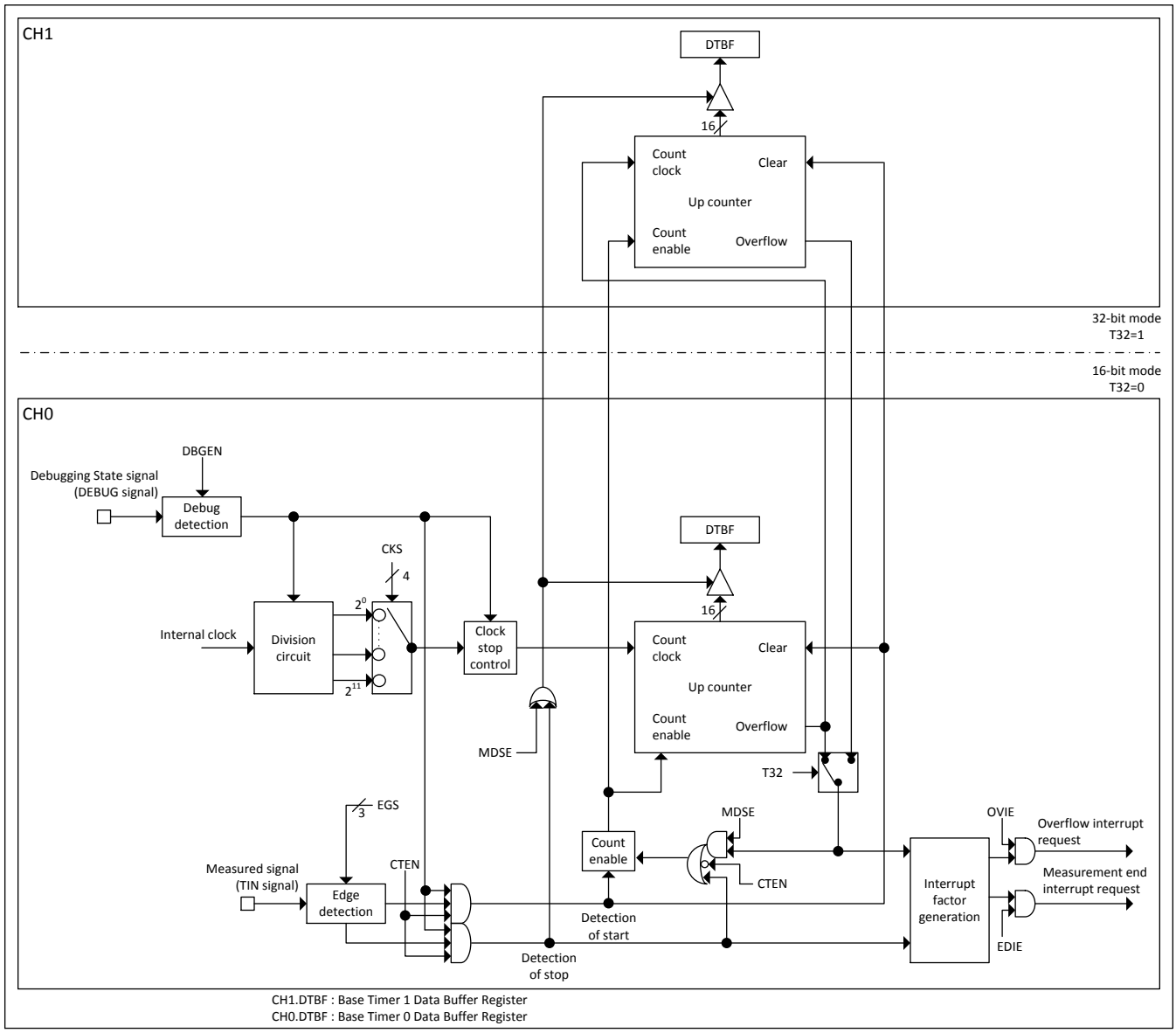


Figure 2-4 Block Diagram of the 16/32-Bit PWC Timer (ch1, ch0)



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. In the beginning the output level is "L". 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 Low Width Setting Reload Register (PRLL) upon trigger activation. In the beginning the output level is "L". The output is inverted to the "H" level when the counter underflows. Subsequently, the counter starts counting down from the set value in the High Width Setting Reload Register (PRLH). 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 defined by the MDSE bit setting - the polarity of the output is toggled each time an underflow is detected, or, the output is set to "H" when counting starts and switches to "L" when an underflow is detected.

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 detection of the measurement start edge, and stops the counter upon the detection of a measurement end edge. The resulting count value will be stored as the measured pulse width in the Data Buffer Register (DTBF).

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 "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 the odd-numbered channel to Reset mode. For the Reload Timer, set the upper 16-bit reload value of the 32 bits in the Cycle Setting Register (PCSR) of the odd-numbered channel. Then, set the lower 16-bit reload value in the Cycle Setting Register (PCSR) 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 while the counter is in a 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.

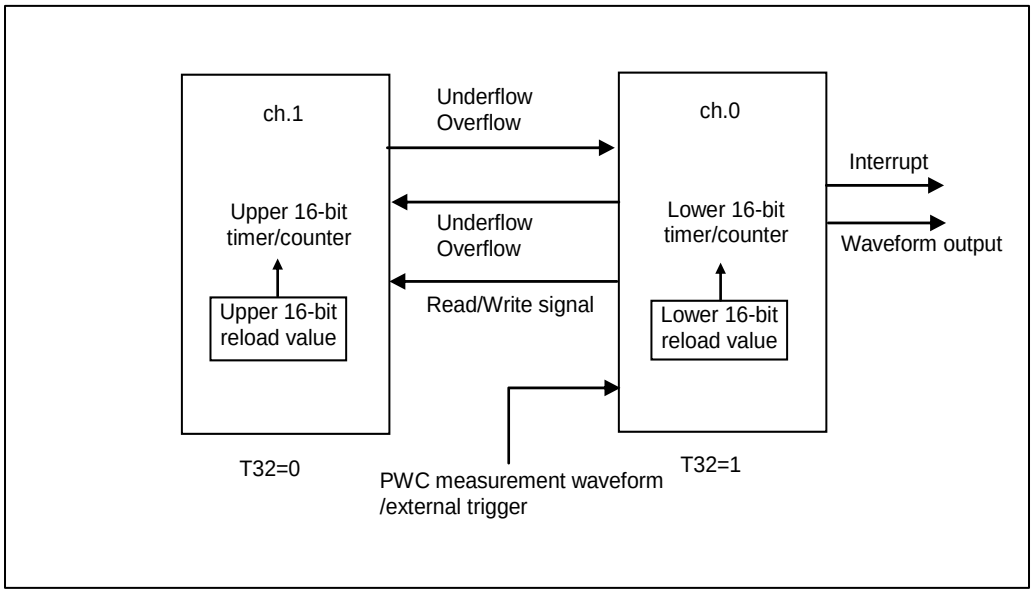
32-Bit Mode Operation

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 (PCSR) at the Reload Timer time) are ignored. The timer start, waveform output, and interrupt signals of the even-numbered channel are valid. (The corresponding signals for the odd-numbered channel will be masked to "L".)

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. Debug Mode

If it goes into a Debug mode, operation of the Base Timer can be stopped and debugged.

If all the following conditions are satisfied, it will go into the Debug mode.

- The debug enable bit (DBGEN) of the Base Timer Debug Register (BT_DEBUG) is "1".
- The debug signal (DEBUG) is active.

It becomes the following operation when it goes into the Debug mode.

- The prescaler stop.
- The timer stop.
- The output waveform stop.
  - It holds the output value at the time of the stop.
- The trigger is ignored in PWM Timer, PPG Timer or Reload timer.
- The measurement edge is ignored in PWC Timer.

It becomes the following operation when it restores from the Debug mode.

- The prescaler is resumed from the stopped value.
- The timer is resumed from the stopped value.
- The output waveform resumes the output from the stopped state.

Note:

- *Register access is possible in the Debug mode.*

6. 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 6-1 shows the interrupt controller bits and interrupt factors of each function.

Table 6-1 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

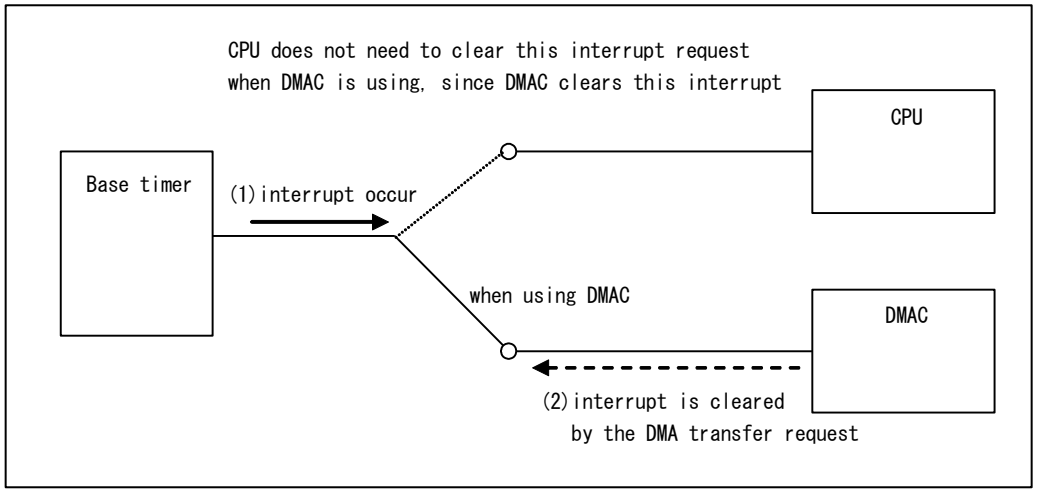
7. Start of DMA Controller (DMAC)

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 7-1 shows an overview of DMAC start with the Base Timer.

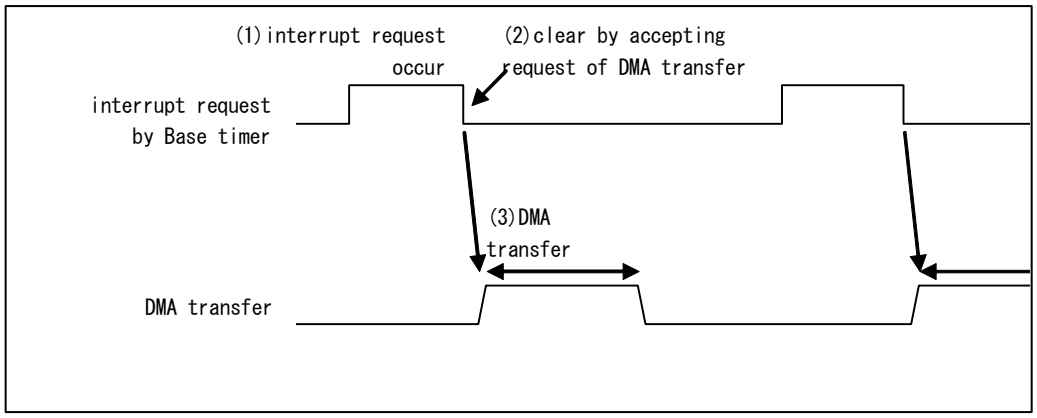
Figure 7-1 Overview of DMAC Start with the Base Timer



Configure the DMAC before starting it with the Base Timer. For details on DMAC settings, see the following chapters: "DMA Controller" and "INTERRUPTS".

Figure 7-2 shows an example of a DMA transfer operation with a Base Timer interrupt request.

Figure 7-2 DMA Transfer Operation Example



8. 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 8-1 List of Registers When the 16-Bit PWM Timer Is Selected

Abbreviated Register Name	Register Name	Reference
TMCR	Timer Control Register	10.1.9
TMCR2	Timer Control Register 2	10.1.9
STC	Status Control Register	10.1.9
STCC	Status Control Clear Register	10.1.9
STCS	Status Control Set Register	10.1.9
PCSR	PWM Cycle Setting Register	10.1.10
PDUT	PWM Duty Setting Register	10.1.11
TMR	Timer Register	10.1.12
PSDR	Start Delay Value Setting Register	10.1.13
ADTR	ADC Trigger Value Setting Register	10.1.14
BT_DEBUG	Base Timer Debug Register	10.1.15
ETCDR	External Timer Compare Data Register	10.1.16

List of Registers when 16-Bit PPG Timer is Selected

Table 8-2 List of Registers When the 16-Bit PPG Timer Is Selected

Abbreviated Register Name	Register Name	Reference
TMCR	Timer Control Register	10.2.7
TMCR2	Timer Control Register 2	10.2.7
STC	Status Control Register	10.2.7
STCC	Status Control Clear Register	10.2.7
STCS	Status Control Set Register	10.2.7
PRLL	Low Width Setting Reload Register	10.2.8
PRLH	High Width Setting Reload Register	10.2.9
TMR	Timer Register	10.2.10
BT_DEBUG	Base Timer Debug Register	10.2.11
ETCDR	External Timer Compare Data Register	10.2.12

List of Registers when Reload Timer is Selected

Table 8-3 List of Registers When the Reload Timer Is Selected

Abbreviated Register Name	Register Name	Reference
TMCR	Timer Control Register	10.3.4
TMCR2	Timer Control Register 2	10.3.4
STC	Status Control Register	10.3.4
STCC	Status Control Clear Register	10.3.4
STCS	Status Control Set Register	10.3.4
PCSR	Cycle Setting Register	10.3.5
TMR	Timer Register	10.3.6
BT_DEBUG	Base Timer Debug Register	10.2.11
ETCDR	External Timer Compare Data Register	10.2.12

List of Registers when PWC Timer is Selected

Table 8-4 List of Registers When the PWC Timer Is Selected

Abbreviated Register Name	Register Name	Reference
TMCR	Timer Control Register	10.4.2
TMCR2	Timer Control Register 2	10.4.2
STC	Status Control Register	10.4.2
STCC	Status Control Clear Register	10.4.2
STCS	Status Control Set Register	10.4.2
DTBF	Data Buffer Register	10.4.3
BT_DEBUG	Base Timer Debug Register	10.2.11

9. 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

- 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.

Bit	Bit Name	Description
[TMCR2 bit0], [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 "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.

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 the Base Timer is configured in one-shot mode and is restarted when count end is detected, the counter will reload the count value and restart.

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 = "100") and the measurement start setting (CTEN="1") are set simultaneously after a system reset assertion or upon leaving Reset mode, the PWC Timer operation may vary due to the state of the immediately preceding measurement signal.
- When operating in continuous measurement mode, if a measurement start edge is detected at the same time that the Base Timer has been restarted, 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 the restart has been performed.
  - 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.

10. Base Timer Description by Function Mode

This section explains each function of the Base Timer.

Functions of the Base Timer

10.1 PWM Timer Function

10.2 PPG Timer Function

10.3 Reload Timer Function

10.4 PWC Timer Function

10.1. 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.

10.1.1 16-Bit PWM Timer Operation
10.1.2 One-Shot Operation
10.1.3 Start Delay Function
10.1.4 ADC Trigger Output Timing
10.1.5 External Timer Match Starting
10.1.6 Interrupt Factors and Timing Chart
10.1.7 Output Waveform
10.1.9 Timer Control Registers (TMCR, TMCR2), Status Control Register (STC), Status Control Clear Register (STCC), and Status Control Set Register (STCS)
10.1.10 PWM Cycle Setting Register (PCSR)
10.1.11 PWM Duty Setting Register (PDUT)
10.1.12 Timer Register (TMR)
10.1.13 Start Delay Value Setting Register (PSDR)
10.1.14 ADC Trigger Value Setting Register (ADTR)
10.1.15 Base Timer Debug Register (BT_DEBUG)
10.1.16 External Timer Compare Data Register (ETCDR)

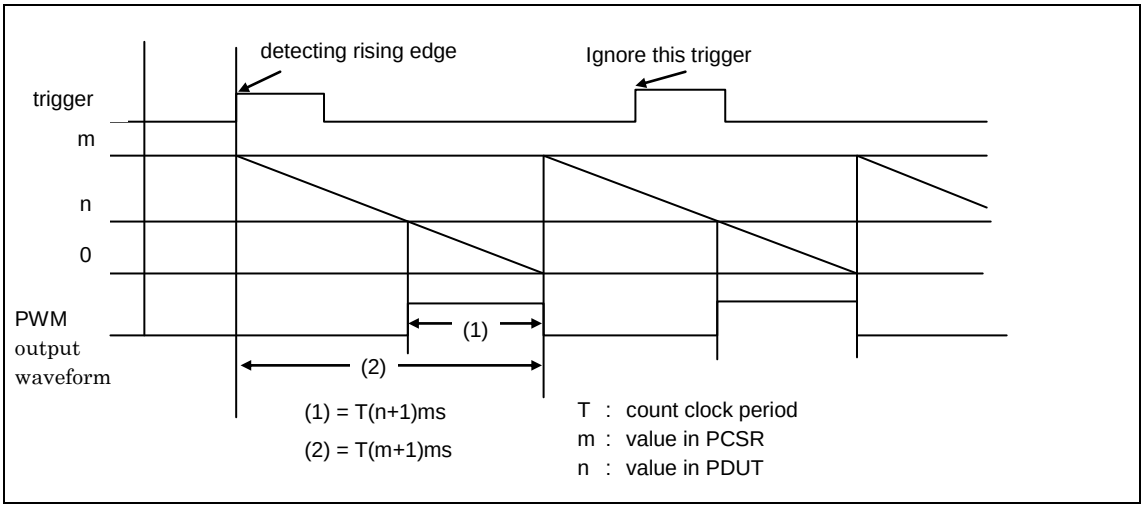
10.1.1. 16-Bit PWM Timer Operation

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

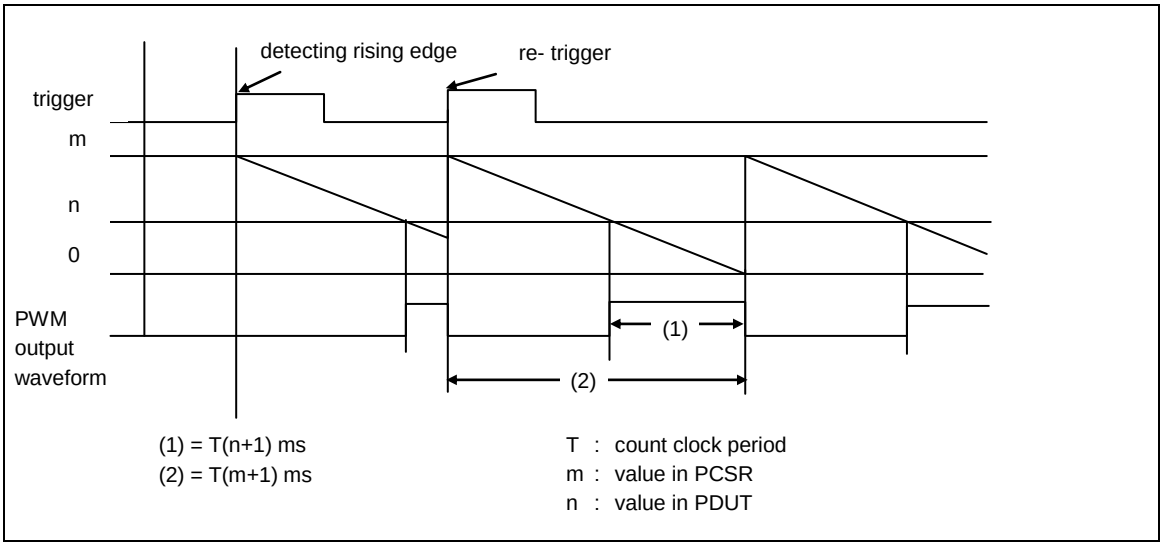
When Restart is Disabled (RTGEN="0")

Figure 10-1 PWM Operation Timing Chart (When Restart Is Disabled)



When Restart is Enabled (RTGEN="1")

Figure 10-2 PWM Operation Timing Chart (When Restart Is Enabled)



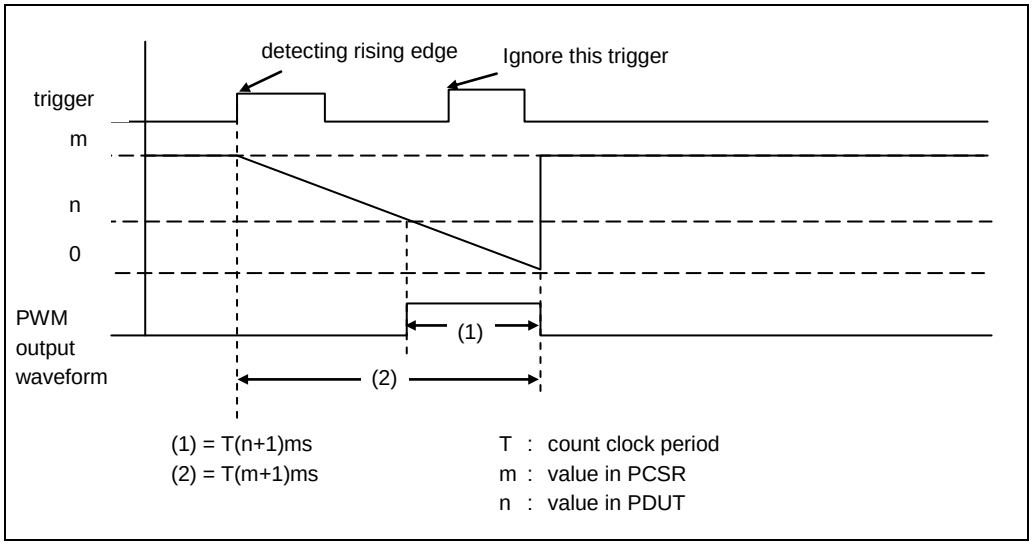
10.1.2. One-Shot Operation

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

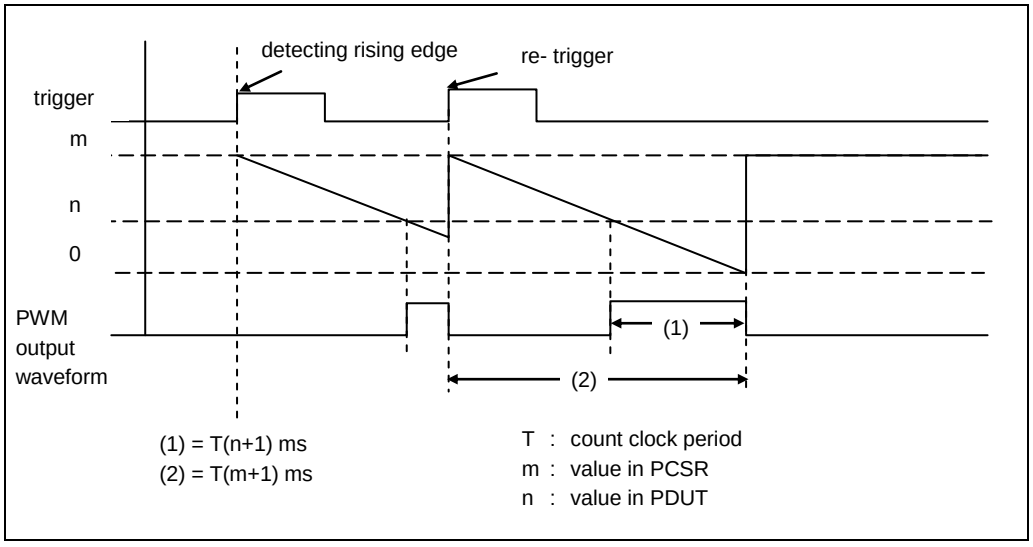
When Restart is Disabled (RTGEN="0")

Figure 10-3 One-Shot Operation Timing Chart (Trigger Restart Disabled)



When Restart is Enabled (RTGEN="1")

Figure 10-4 One-Shot Operation Timing Chart (Trigger Restart Enabled)



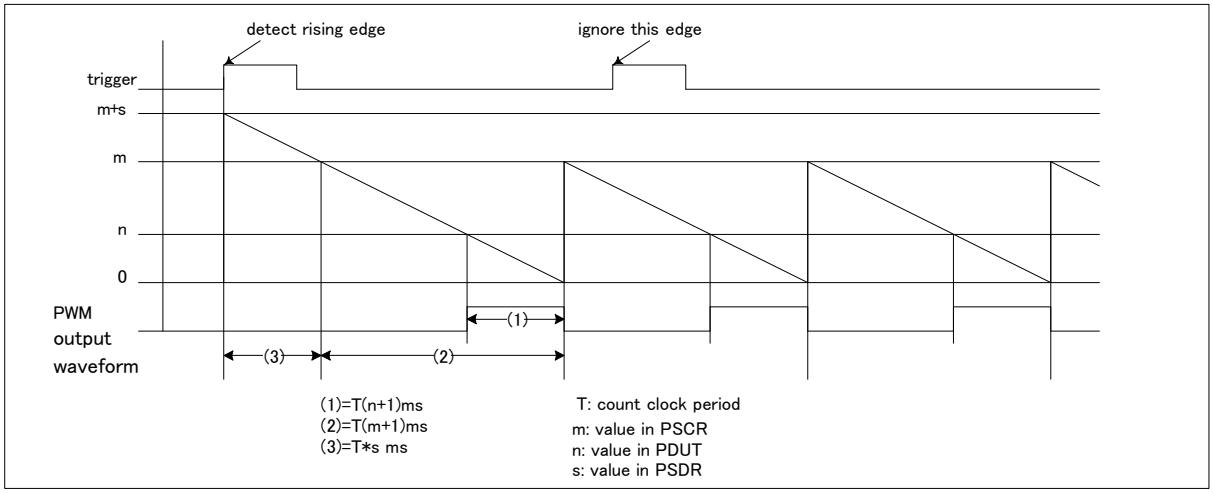
10.1.3. Start Delay Function

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")

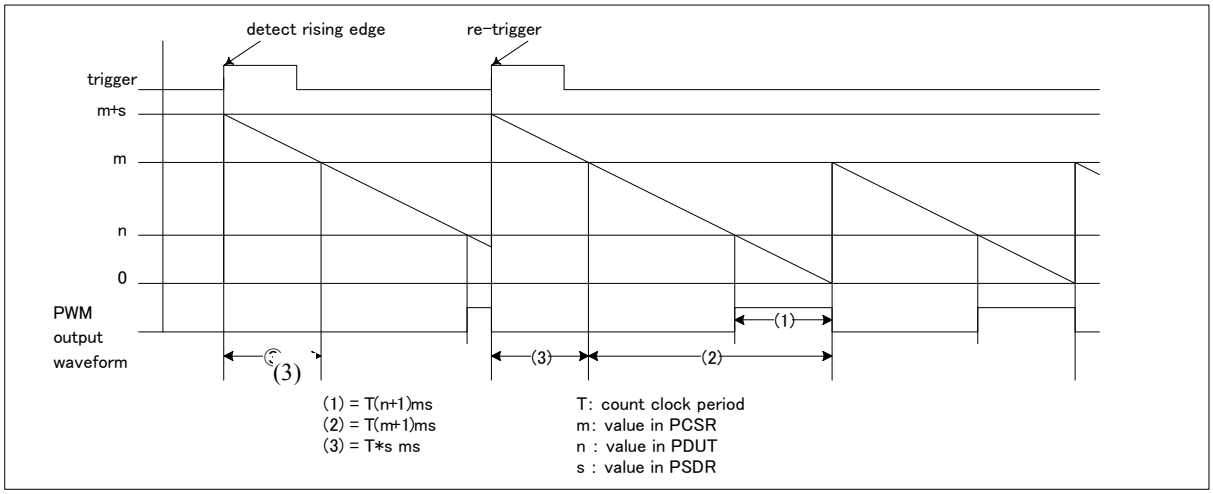
Figure 10-5 Start Delay Operation in Continuous Operation (Trigger Restart Disabled)



No PWM waveform is output during the start delay period ((3) above).

When Restart is Enabled (RTGEN="1")

Figure 10-6 Start Delay Operation in Continuous Operation (Trigger Restart Enabled)

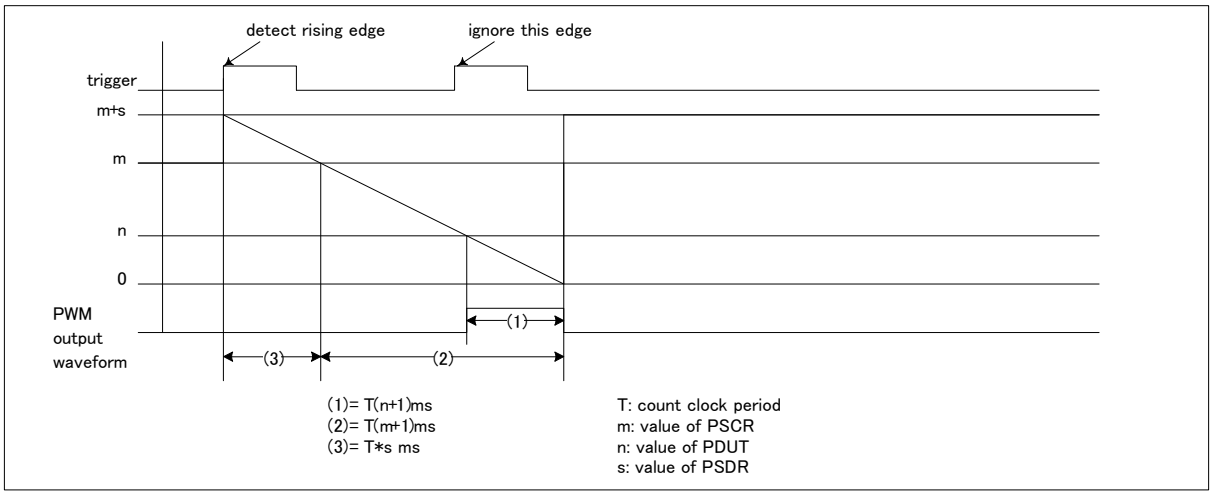


No PWM waveform is output during the start delay period ((3) above).

Start Delay Operation in One-Shot Operation

When Restart is Disabled (RTGEN="0")

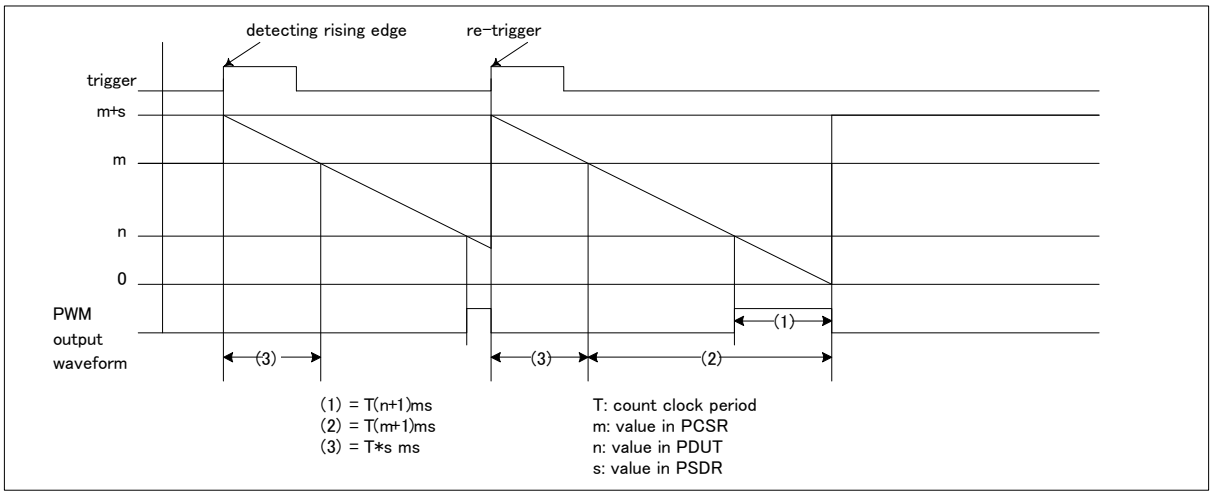
Figure 10-7 Start Delay Operation in One-Shot Operation (Trigger Restart Disabled)



No PWM waveform is output during the start delay period ((3) above).

When Restart is Enabled (RTGEN="1")

Figure 10-8 Start Delay Operation in One-Shot Operation (Trigger Restart Enabled)



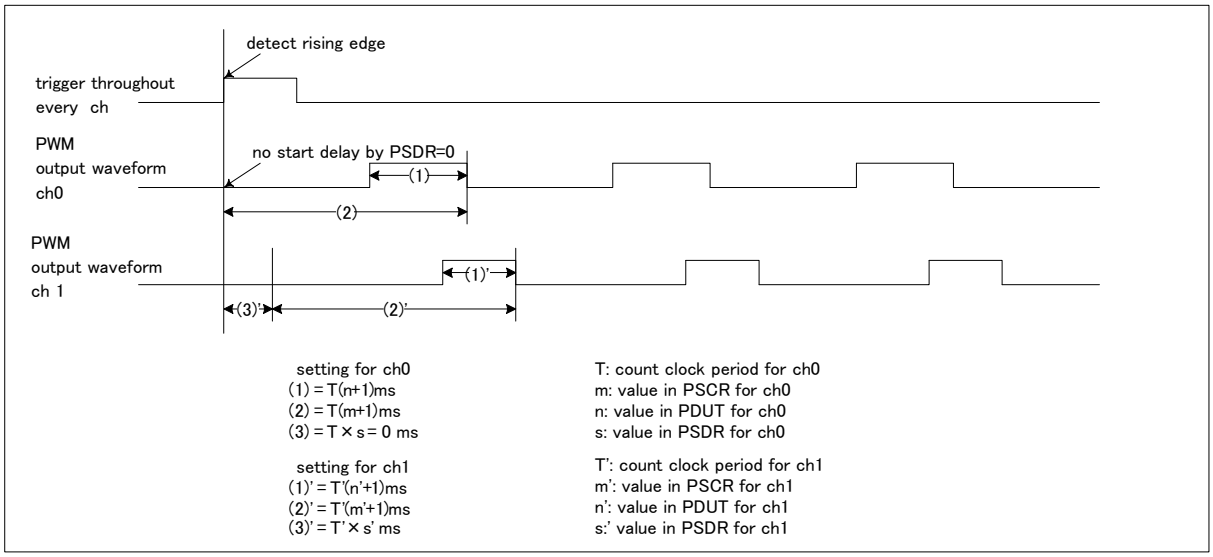
No PWM waveform is output during the start delay period ((3) above).

Start Delay Operation with Multiple Channels interlocked

The trigger can be controlled to interlock several channels in a specific phase relationship.

When Restart is Disabled (RTGEN="0")

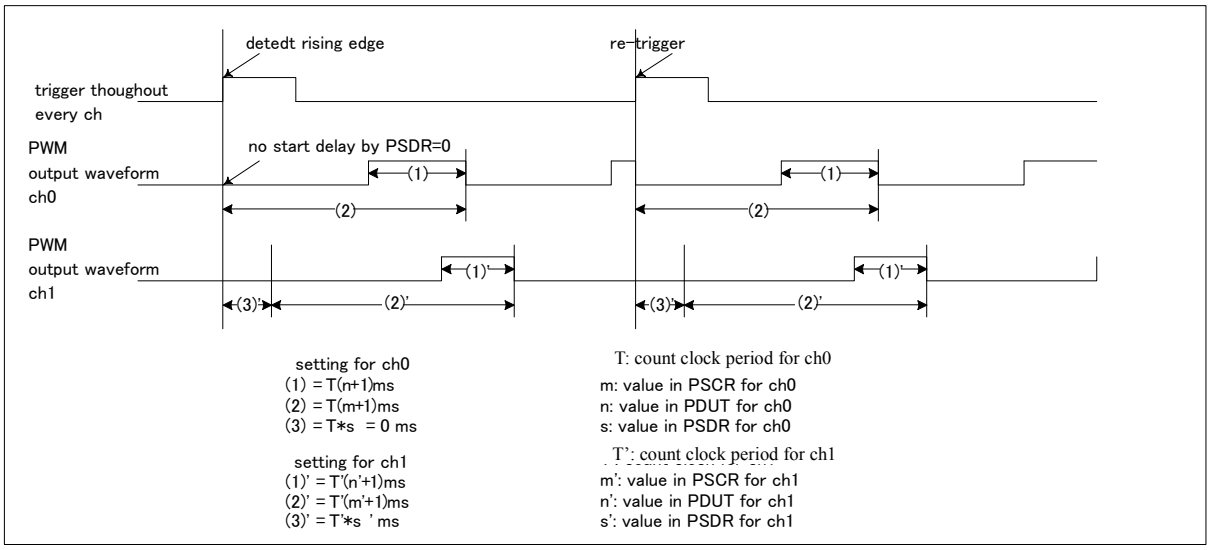
Figure 10-9 Start Delay Operation in Continuous Operation with Multiple Channels Interlocked (Trigger Restart Disabled)



No PWM waveform is output during the start delay period ((3) above).

When Restart is Enabled (RTGEN="1")

Figure 10-10 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).

10.1.4. ADC Trigger Output Timing

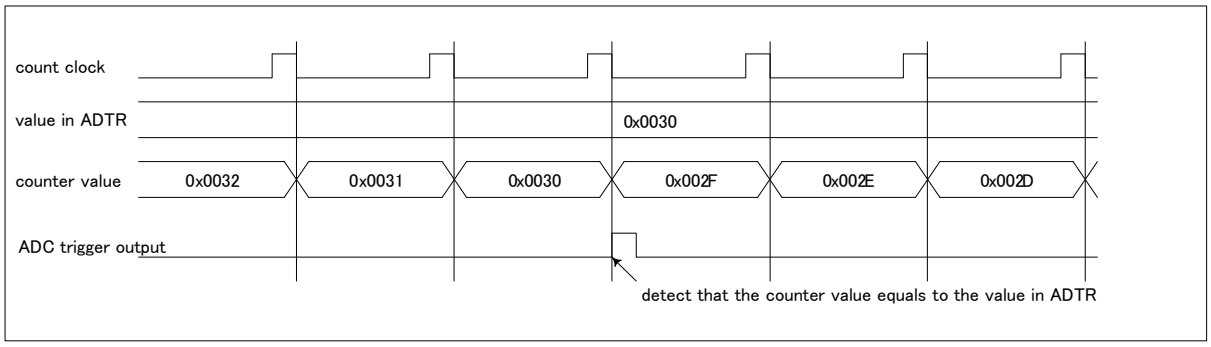
This section explains the ADC trigger output timing.

ADC Trigger Output Timing

ADC trigger is output when the 16-bit down counter is compared with the ADTR register and they match.

Figure 10-11 shows a timing chart of ADC trigger output when ADTR is 0x0030.

Figure 10-11 Timing Chart of ADC Trigger Output



10.1.5. External Timer Match Starting

If the external timer match starting function is used, the 16/17-bit down counter can start in synchronization with an external timer.

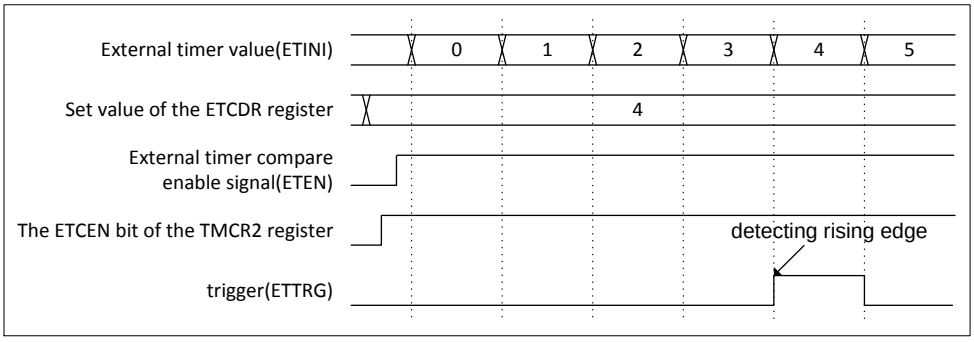
If all the following conditions are satisfied, starting by an external timer is performed.

- The CTEN bit of the TMCR register is "1".
- The ETCEN bit of the TMCR2 register is "1".
- The external timer compare enable signal (ETEN) is active.

When timer is running, the restart by the external timer match starting does not occur. If performing external timer match starting again, perform the following processings.

1. The CTEN bit of the TMCR register is set to "0".
2. The CTEN bit of the TMCR register is set to "1" again.

Figure 10-12 Example of External Timer Match Starting Operation (ETCDR=4)



10.1.6. Interrupt Factors and Timing Chart

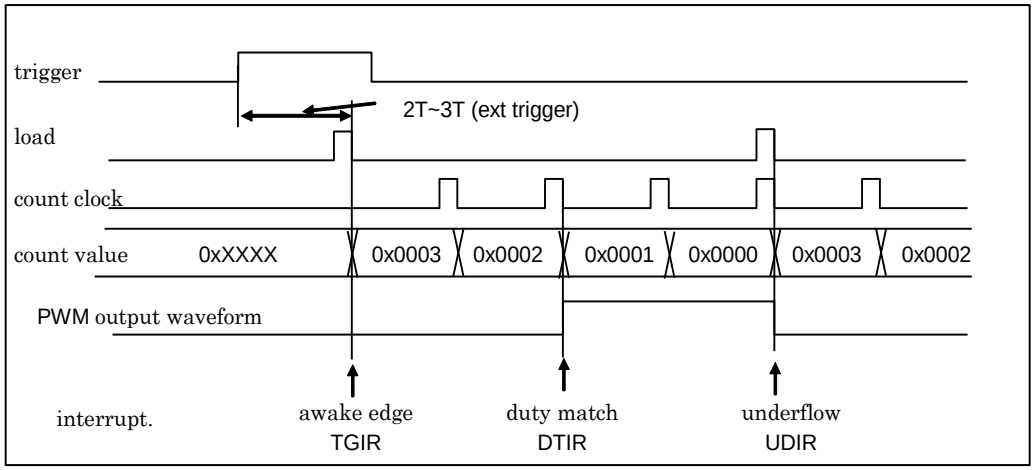
This section shows interrupt factors and a timing chart.

Interrupt Factors and Timing Chart (PWM Output: Normal Polarity)

The time required for a counter value to be loaded following the input of a trigger is, for software triggers T, and for external triggers 2T to 3T (T: internal clock cycle).

Figure 10-13 shows the interrupt factors and a timing chart where the cycle setting value is 3 and the duty value is 1.

Figure 10-13 PWM Timer Interrupt Factors and Timing Chart



10.1.7. Output Waveform

This section shows PWM output.

Output methods for All "L" or All "H" in PWM Output

Figure 10-14 shows an output method where the PWM output is all "L". Figure 10-15 shows an output method where it is all "H".

Figure 10-14 Example Where PWM Output Is All "L" Level（TMCR.OSEL="0"）

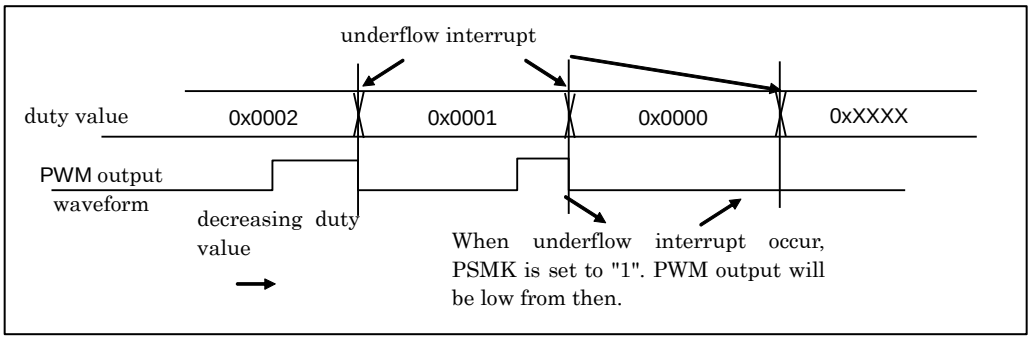
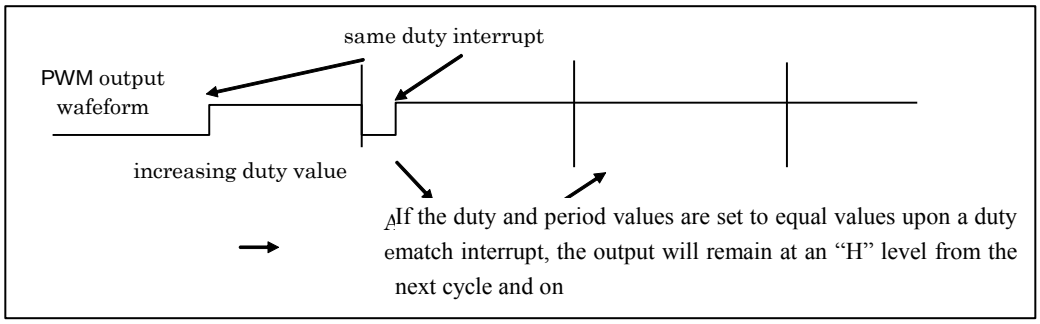


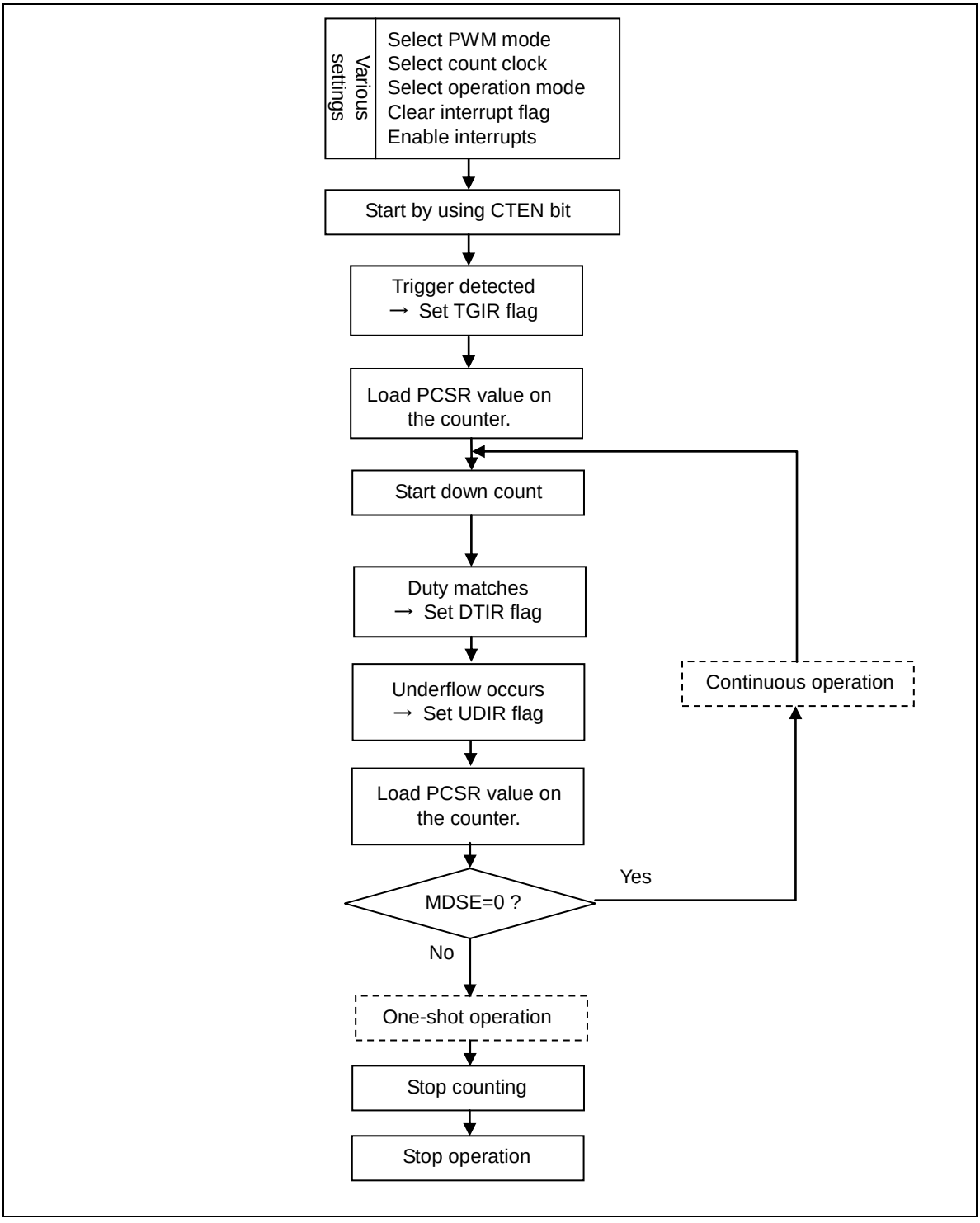
Figure 10-15 Example Where PWM Output Is All "H" Level（TMCR.OSEL="0"）



10.1.8. PWM Timer Operation Flow

This section shows the PWM Timer operation flow.

PWM Timer Operation Flow



10.1.9. 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 "9.Notes on Using the Base Timer".

10.1.9.1 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:bit0,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).

CKS3	CKS2	CKS1	CKS0	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] 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.

EGS1	EGS0	Description
0	0	Trigger input invalid
0	1	Rising edge
1	0	Falling edge
1	1	Both edges

10.1.9.2 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".

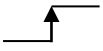
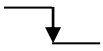
[bit6:4] FMD2 to FMD0: Timer Function Selection Bits

- 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.

FMD2	FMD1	FMD0	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
0	Stop operation.
1	Enable operation.

[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

10.1.9.3 Timer Control Register 2 (TMCR2)

bit	7	6	5	4	3	2	1	0
Field	Reserved						ETCEN	CKS3
R/W Attribute	R0,W0						R/W	R/W
Protection Attribute	-							
Initial Value	000000						0	0

[bit7:2] Reserved: Reserved Bits

The read value is "0".

If writing to this bit, write "0".

[bit1] ETCEN: External Timer Compare Enable Bit

It is the bit which enables match detection of an external timer value and the ETCDR register value.

Bit	Description
0	Match detection disable
1	Match detection enable

[bit0] CKS3: Count Clock Selection Bit

See "Count clock selection bits" in "10.1.9Timer Control Registers (TMCR, TMCR2), Status Control Register (STC), Status Control Clear Register (STCC), and Status Control Set Register (STCS)".

10.1.9.4 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

[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.

10.1.9.5 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

[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.

10.1.9.6 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			

[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".

10.1.10. 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 (TMR) at the start time and at the underflow time.

bit	15	14	13	12	11	10	9	8
Field	PCSR[15:8]							
R/W Attribute	R/W							
Protection Attribute	-							
Initial Value	XXXX XXXX							

bit	7	6	5	4	3	2	1	0
Field	PCSR[7:0]							
R/W Attribute	R/W							
Protection Attribute	-							
Initial Value	XXXX XXXX							

These bits compose the register with a buffer for setting a cycle. There is a transfer to the Timer Register (TMR) at the start time and at the underflow time.

When initializing and rewriting the Cycle Setting Register (PCSR), be sure to write to the duty setting register after writing to the Cycle Setting Register (PCSR).

- 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.

10.1.11. 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	14	13	12	11	10	9	8
Field	PDUT[15:8]							
R/W Attribute	R/W							
Protection Attribute	-							
Initial Value	XXXX XXXX							

bit	7	6	5	4	3	2	1	0
Field	PDUT[7:0]							
R/W Attribute	R/W							
Protection Attribute	-							
Initial Value	XXXX XXXX							

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 (PCSR) 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.

10.1.12. Timer Register (TMR)

The Timer Register (TMR) can read the value of the17-bit down counter.

Bit	31	24
Field	Reserved	
R/W Attribute	R0,W0	
Protection Attribute	-	
Initial Value	0000 0000	

bit	23	17	16
Field	Reserved		TMR[16]
R/W Attribute	R0,W0		R,WX
Protection Attribute	-		
Initial Value	0000000		0

Bit	15	8
Field	TMR[15:8]	
R/W Attribute	R,WX	
Protection Attribute	-	
Initial Value	0000 0000	

Bit	7	0
Field	TMR[7:0]	
R/W Attribute	R,WX	
Protection Attribute	-	
Initial Value	0000 0000	

[bit31:17] Reserved: Reserved Bits

The read value is "0".

[bit16:0] TMR16 to TMR0: Timer Value Data Bits

The Timer Register (TMR) 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.

10.1.13. 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							8
Field	PSDR[15:8]							
R/W Attribute	R/W							
Protection Attribute	-							
Initial Value	0000 0000							

bit	7							0
Field	PSDR[7:0]							
R/W Attribute	R/W							
Protection Attribute	-							
Initial Value	0000 0000							

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").

10.1.14. ADC Trigger Value Setting Register (ADTR)

The ADC Trigger Value Setting Register (ADTR) sets the time for ADC trigger output.

bit	15	14	13	12	11	10	9	8
Field	ADTR[15:8]							
R/W Attribute	R/W							
Protection Attribute	-							
Initial Value	00000 0000							

bit	7	6	5	4	3	2	1	0
Field	ADTR[7:0]							
R/W Attribute	R/W							
Protection Attribute	-							
Initial Value	0000 0000							

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.

10.1.15. Base Timer Debug Register (BT_DEBUG)

The Base Timer Debug Register (BT_DEBUG) performs enable/disable setting of debug.

bit	15	14	13	12	11	10	9	8
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial Value	0000 0000							

bit	7	6	5	4	3	2	1	0
Field	Reserved							DBGEN
R/W Attribute	R0,W0							R/W
Protection Attribute	-							
Initial Value	0000000							0

[bit15:1] Reserved: Reserved Bits

The read value is "0".

If writing to this bit, write "0".

[bit0] DBGEN: Debug Enable Bit

■ It is the bit which permits stopping operation of a Base Timer with the debug signal (DEBUG).

Bit	Description
0	Disable
1	Enable

10.1.16. External Timer Compare Data Register (ETCDR)

The External Timer Compare Data Register (ETCDR) is the register which sets the value which is started by an external timer.

bit	15	8
Field	ETCDR[15:8]	
R/W Attribute	R/W	
Protection Attribute	-	
Initial Value	01111111	

bit	7	0
Field	ETCDR[7:0]	
R/W Attribute	R/W	
Protection Attribute	-	
Initial Value	11111111	

It is a register compared with an external timer signal (ETINI). Refer to "10.1.5 External Timer Match Starting" for details.

- Use 16- or 32-bit data access for the ETCDR register.

10.2. 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.

10.2.1 16-Bit PPG Timer Operation
10.2.2 Continuous Operation
10.2.3 One-Shot Operation
10.2.4 External Timer Match Starting
10.2.5 Interrupt Factors and Timing Chart
10.2.6 PPG Timer Operation Flow
10.2.7 Timer Control Registers (TMCR, TMCR2), Status Control Register (STC), Status Control Clear Register (STCC), and Status Control Set Register (STCS)
10.2.8 Low Width Setting Reload Register (PRLL)
10.2.9 High Width Setting Reload Register (PRLH)
10.2.10 Timer Register (TMR)
10.2.11 Base Timer Debug Register (BT_DEBUG)
10.2.12 External Timer Compare Data Register (ETCDR)

10.2.1. 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

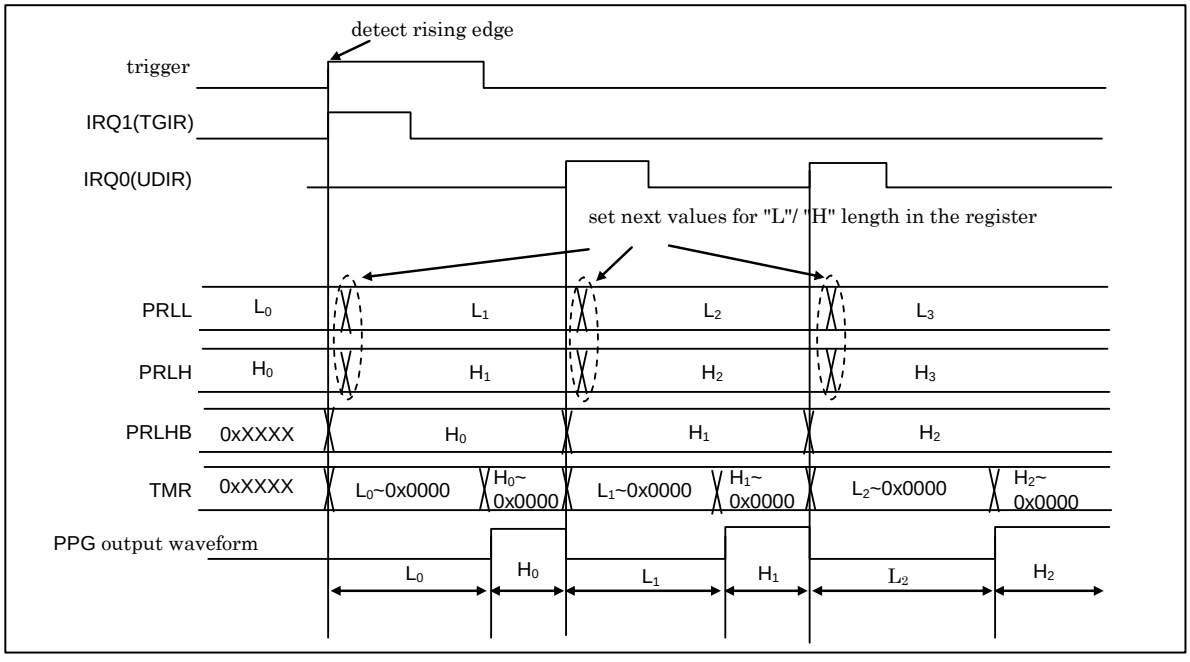
There are two 16-bit reload registers for the "L" width and "H" width settings and one buffer for the "H" width setting (PRLL, PRLH, and PRLHB).

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 the counter counts down each count clock. The detection of an underflow causes reloading of the PRLHB value into the counter, inverts the PPG output waveform and continues counting down each count clock. 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, and also before the next cycle change occurs following an underflow interrupt factor (UDIR) being asserted to "1". 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.



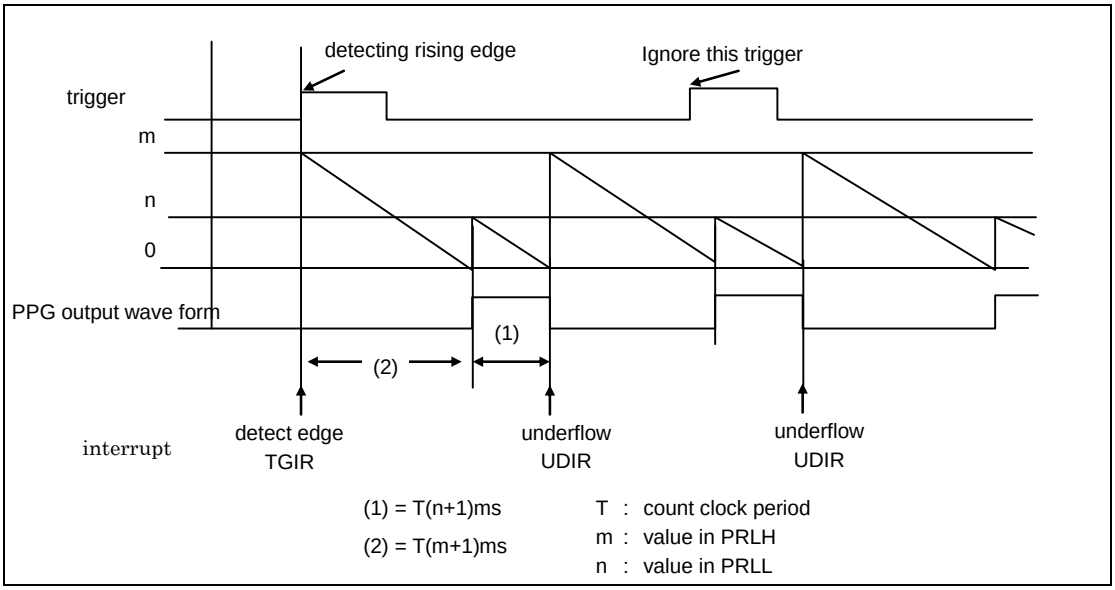
10.2.2. Continuous Operation

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

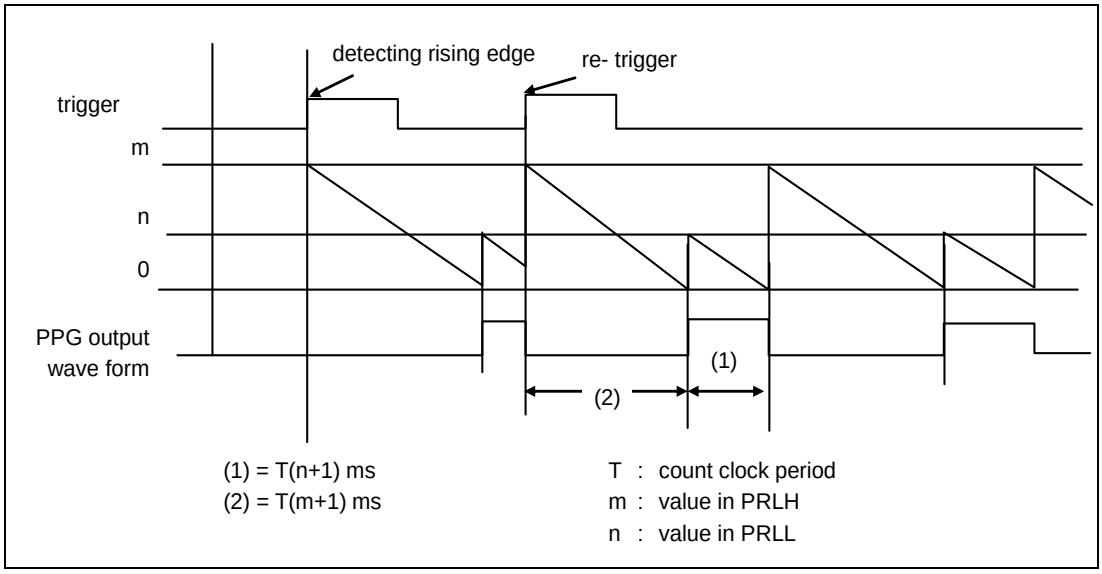
When Restart is Disabled (RTGEN="0")

Figure 10-16 PPG Operation Timing Chart (When Restart Is Disabled)



When Restart is Enabled (RTGEN="1")

Figure 10-17 PPG Operation Timing Chart (When Restart Is Enabled)



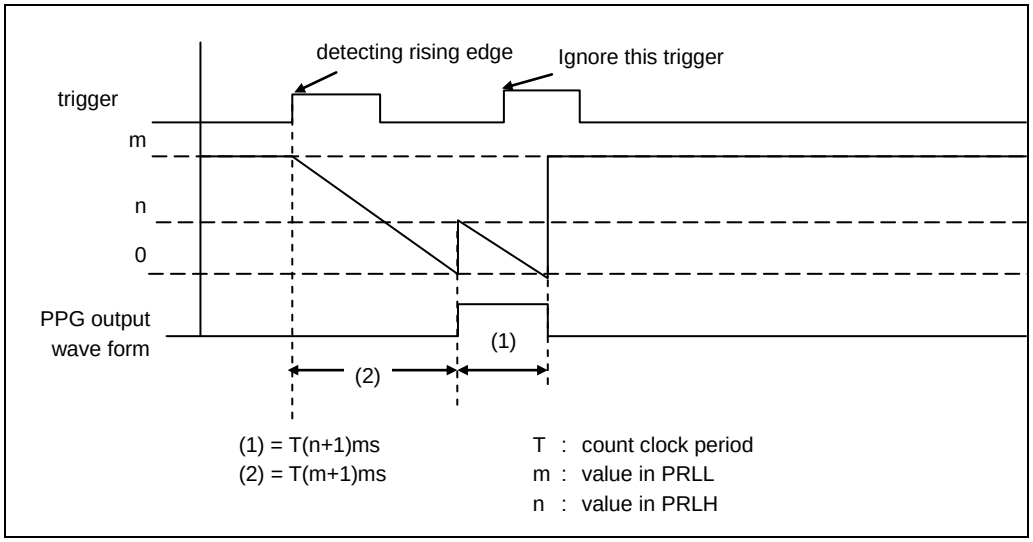
10.2.3. One-Shot Operation

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

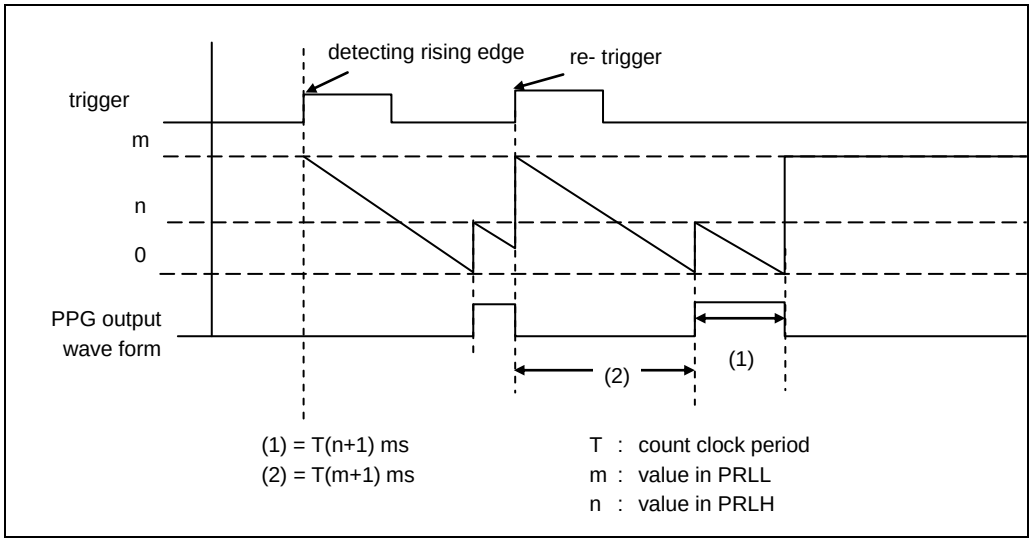
When Restart is Disabled (RTGEN="0")

Figure 10-18 One-Shot Operation Timing Chart (Trigger Restart Disabled)



When Restart is Enabled (RTGEN="1")

Figure 10-19 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, when the value of the reload register is 0x0000, the pulse width is equal to 1 count clock cycle. Similarly, when the value of the reload register is 0xFFFF, the pulse width is equal to 65536 count clock cycles. 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

10.2.4. External Timer Match Starting

If the external timer match starting function is used, the 16/17-bit down counter can start in synchronization with an external timer.

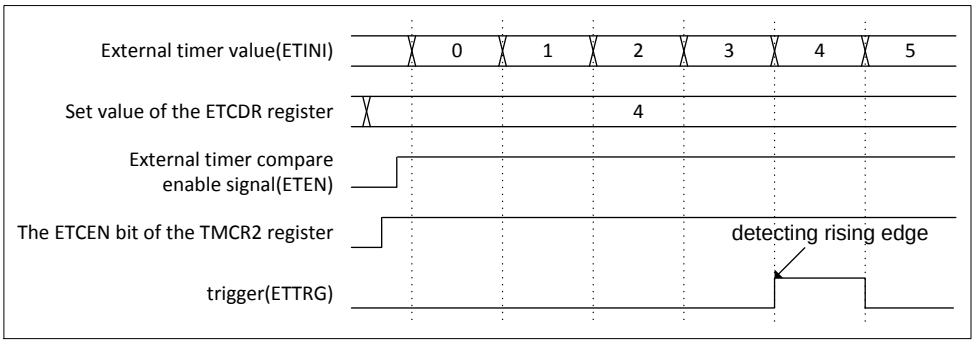
If all the following conditions are satisfied, starting by an external timer is performed.

- The CTEN bit of the TMCR register is "1".
- The ETCEN bit of the TMCR2 register is "1".
- The external timer compare enable signal (ETEN) is active.

When timer is running, the restart by the external timer match starting does not occur. If performing external timer match starting again, perform the following processings.

1. The CTEN bit of the TMCR register is set to "0".
2. The CTEN bit of the TMCR register is set to "1" again.

Figure 10-20 The Example of External Timer Match Starting Operation (ETCDR=4)



10.2.5. Interrupt Factors and Timing Chart

This section shows interrupt factors and a timing chart.

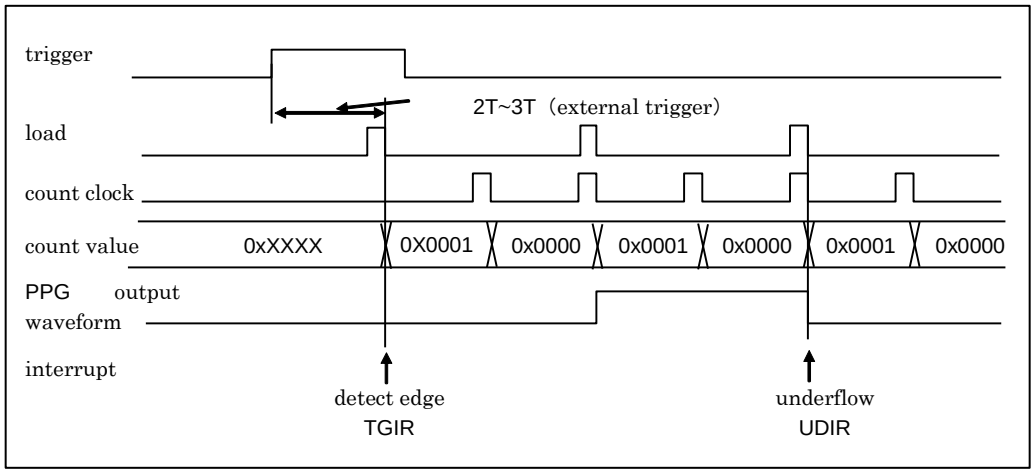
Interrupt Factors and Timing Chart (PPG Output: Normal Polarity)

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 10-21 shows the interrupt factors and a timing chart where the "L" width setting value is 1 and the "H" width setting value is 1.

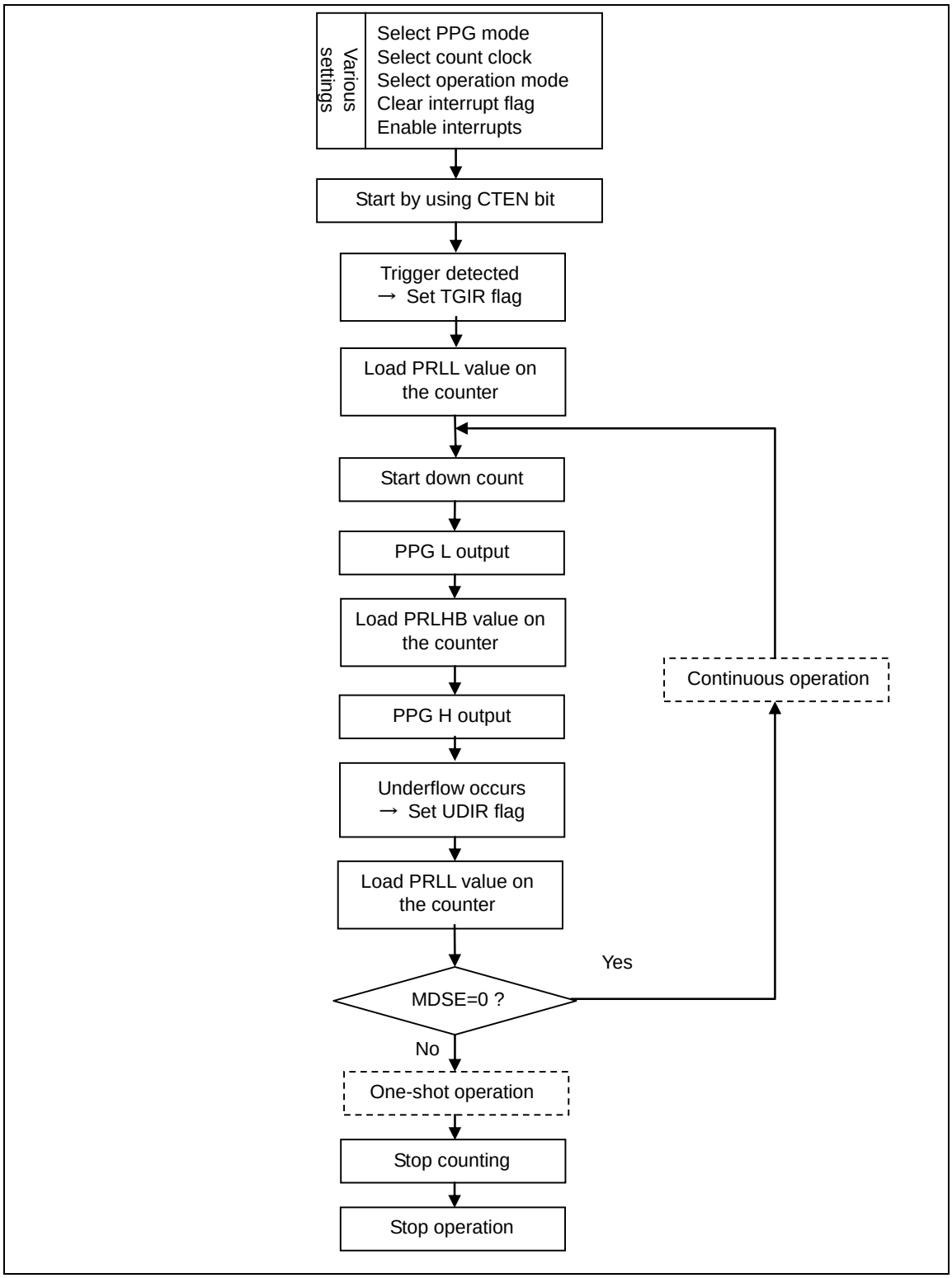
Figure 10-21 PPG Timer Interrupt Factors and Timing Chart



10.2.6. PPG Timer Operation Flow

This section shows the PPG Timer operation flow.

PPG Timer Operation Flow



10.2.7. Timer Control Registers (TMCR, TMCR2), Status Control Register (STC), Status Control Clear Register (STCC), and Status Control Set Register (STCS)

For details on writing to the Status Control Register (STC), see "9. Notes on Using the Base Timer".

10.2.7.1 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:bit0,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.

CKS3	CKS2	CKS1	CKS0	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] 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.

EGS1	EGS0	Description
0	0	Trigger input invalid
0	1	Rising edge
1	0	Falling edge
1	1	Both edges

10.2.7.2 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".

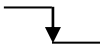
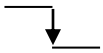
[bit6:4] FMD2 to FMD0: Timer Function Selection Bits

- 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.

FMD2	FMD1	FMD0	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
0	Stop operation.
1	Enable operation.

[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

10.2.7.3 Timer Control Register 2 (TMCR2)

bit	7	6	5	4	3	2	1	0
Field	Reserved						ETCEN	CKS3
R/W Attribute	R0,W0						R/W	R/W
Protection Attribute	-							
Initial Value	000000						0	0

[bit7:2] Reserved: Reserved Bits

The read value is "0".

If writing to this bit, write "0".

[bit1] ETCEN: External Timer Compare Enable Bit

It is the bit which enables match detection of an external timer value and the ETCDR register value.

Bit	Description
0	Match detection disable
1	Match detection enable

[bit0] CKS3: Count Clock Selection Bit

See "Count clock selection bits" in "10.2.7 Timer Control Registers (TMCR, TMCR2), Status Control Register (STC), Status Control Clear Register (STCC), and Status Control Set Register (STCS)".

10.2.7.4 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

[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.

10.2.7.5 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

[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.

10.2.7.6 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			

[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".

10.2.8. Low Width Setting Reload Register (PRLL)

The Low 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 (TMR).

bit	15							8
Field	PRLL[15:8]							
R/W Attribute	R/W							
Protection Attribute	-							
Initial Value	XXXXXXXX							

bit	7							0
Field	PRLL[7:0]							
R/W Attribute	R/W							
Protection Attribute	-							
Initial Value	XXXXXXXX							

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 (TMR).

- 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.

10.2.9. High Width Setting Reload Register (PRLH)

The High 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 (TMR).

bit	15	14	13	12	11	10	9	8
Field	PRLH[15:8]							
R/W Attribute	R/W							
Protection Attribute	-							
Initial Value	XXXXXXXX							

bit	7	6	5	4	3	2	1	0
Field	PRLH[7:0]							
R/W Attribute	R/W							
Protection Attribute	-							
Initial Value	XXXXXXXX							

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 (TMR).

- 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.

10.2.10. Timer Register (TMR)

The Timer Register (TMR) can read the value of the 16-bit down counter.

bit	15	14	13	12	11	10	9	8
Field	TMR[15:8]							
R/W Attribute	R,WX							
Protection Attribute	-							
Initial Value	00000000							

bit	7	6	5	4	3	2	1	0
Field	TMR[7:0]							
R/W Attribute	R,WX							
Protection Attribute	-							
Initial Value	00000000							

The Timer Register (TMR) can read the value of the 16-bit down counter.

- Use 16- or 32-bit data access for the TMR register.

10.2.11. Base Timer Debug Register (BT_DEBUG)

The Base Timer Debug Register (BT_DEBUG) performs enable/disable setting of debug.

bit	15	8
Field	Reserved	
R/W Attribute	R0,W0	
Protection Attribute	-	
Initial Value	00000000	

bit	7	0
Field	Reserved	DBGEN
R/W Attribute	R0,W0	R/W
Protection Attribute	-	
Initial Value	0000000	0

[bit15:1] Reserved: Reserved Bits

The read value is "0".

If writing to this bit, write "0".

[bit0] DBGEN: Debug Enable Bit

■ It is the bit which permits stopping operation of a Base Timer with the debug signal (DEBUG).

Bit	Description
0	Disable
1	Enable

10.2.12. External Timer Compare Data Register (ETCDR)

The External Timer Compare Data Register (ETCDR) is the register which sets the value which is started by an external timer.

bit	15	8
Field	ETCDR[15:8]	
R/W Attribute	R/W	
Protection Attribute	-	
Initial Value	01111111	

bit	7	0
Field	ETCDR[7:0]	
R/W Attribute	R/W	
Protection Attribute	-	
Initial Value	11111111	

It is a register compared with an external timer signal (ETINI). Refer to "10.2.4 External Timer Match Starting" for details.

- Use 16- or 32-bit data access for the ETCDR register.

10.3. 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.

10.3.1 Operations of 16-Bit Reload Timer
10.3.2 External Timer Match Starting
10.3.3 Reload Timer Operation Flow
10.3.4 Timer Control Registers (TMCR, TMCR2), Status Control Register (STC), Status Control Clear Register (STCC), and Status Control Set Register (STCS)
10.3.5 Cycle Setting Register (PCSR)
10.3.6 Timer Register (TMR)
10.3.7 Base Timer Debug Register (BT_DEBUG)
10.3.8 External Timer Compare Data Register (ETCDR)

10.3.1. Operations of 16-Bit Reload Timer

The Reload Timer operates by performing a countdown from the set value in the Cycle Setting Register (PCSR), 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

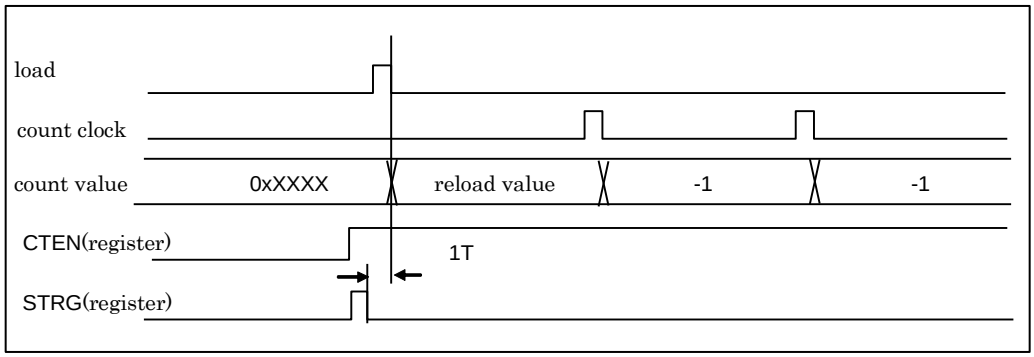
In order to start a count operation and enable counting at the same time, write "1" to both the CTEN bit and STRG bit in the Timer Control Register (TMCR). In a state where the timer has started (CTEN="1"), trigger input with the STRG bit is always enabled regardless of the operation mode.

With count operation enabled, the start of the timer by a software trigger or external trigger results in the Cycle Setting Register (PCSR) 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 (PCSR) data into the counter is 1T (T: Internal clock cycle).

Figure 10-22 shows the start and operation of the counter using a software trigger.

Figure 10-22 Count Operation When the Internal Clock Is Selected



Underflow Operation

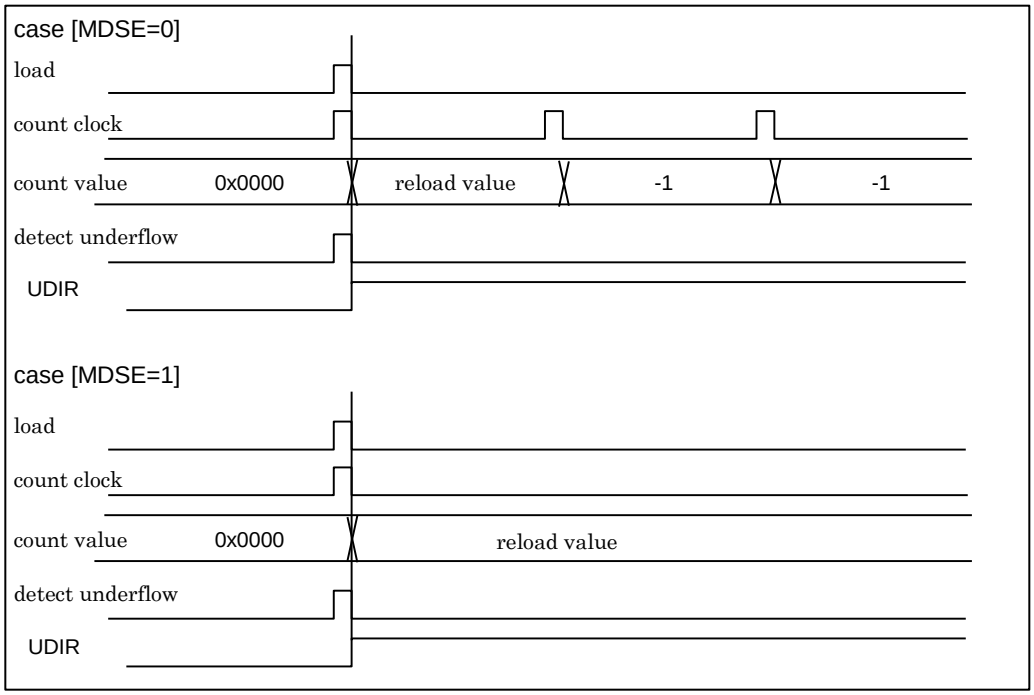
If the counter value changes from 0x0000 to 0xFFFF, an underflow has occurred. Therefore, an underflow occurs at the count of [set value of Cycle Setting Register (PCSR) + 1].

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 10-23 shows an underflow operation timing chart.

Figure 10-23 Underflow Operation Timing Chart

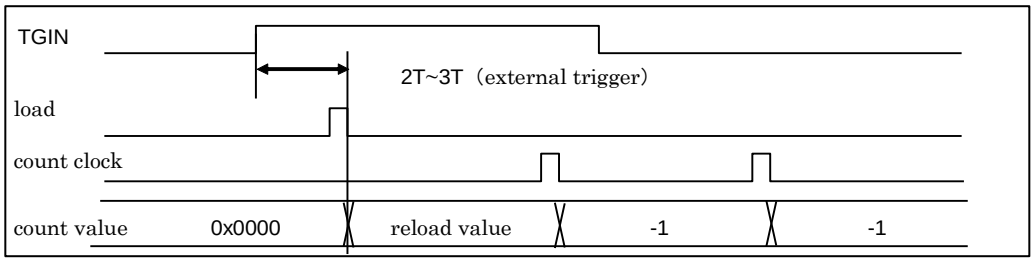


Operation of Input Pin Function

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 (PCSR) 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 10-24 shows trigger input operation where the valid edge specification is a rising edge.

Figure 10-24 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.

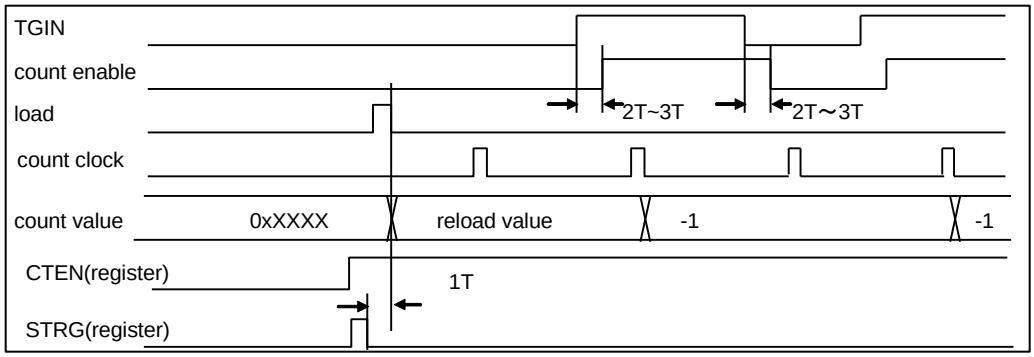
With count operations enabled (TMCR.CTEN="1"), the start of the timer by a software trigger (TMCR.STRG="1") results in the Cycle Setting Register (PCSR) value being loaded into the counter. The count clock counts down only while a valid level is being input to the TGIN pin.

In order to synchronize the signal that is input from the TGIN pin, the time required from the input of a valid level until count enable becomes effective is 2T to 3T at the start and end of the count operation(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 10-25 shows gate function operation where the valid level is "H".

Figure 10-25 Gate Function Operation (TMCR2.GATE="1")

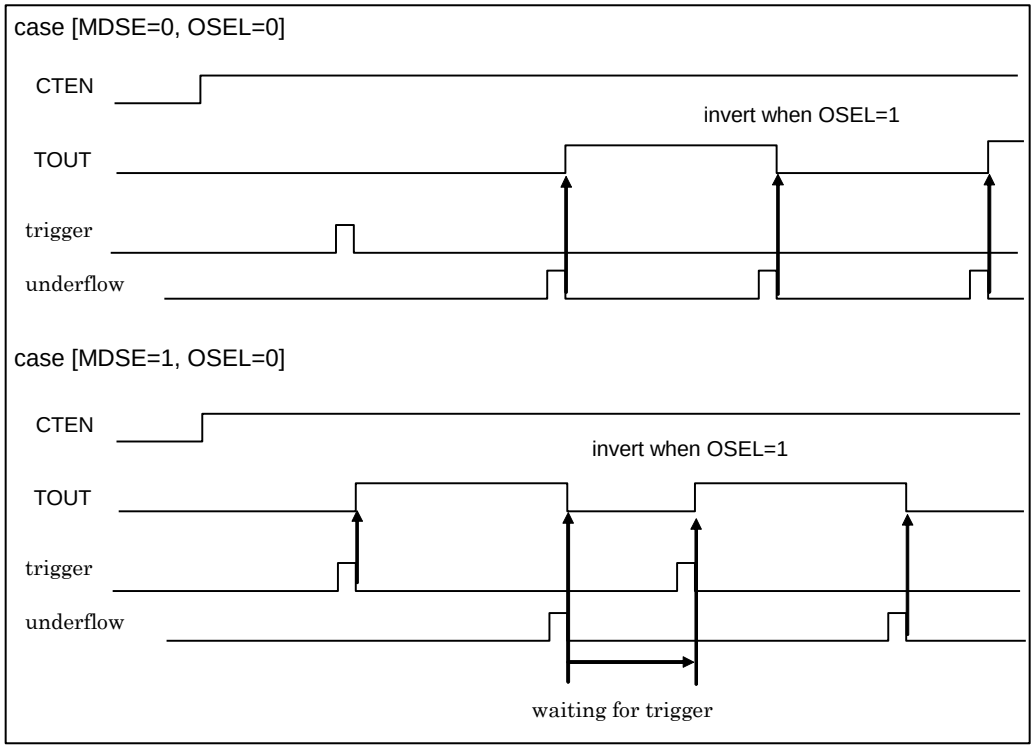


Operation of Output Pin Function

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 10-26 shows an output pin function operation timing chart.

Figure 10-26 Output Pin Function Operation Timing Chart



10.3.2. External Timer Match Starting

If the external timer match starting function is used, the 16/17-bit down counter can start in synchronization with an external timer.

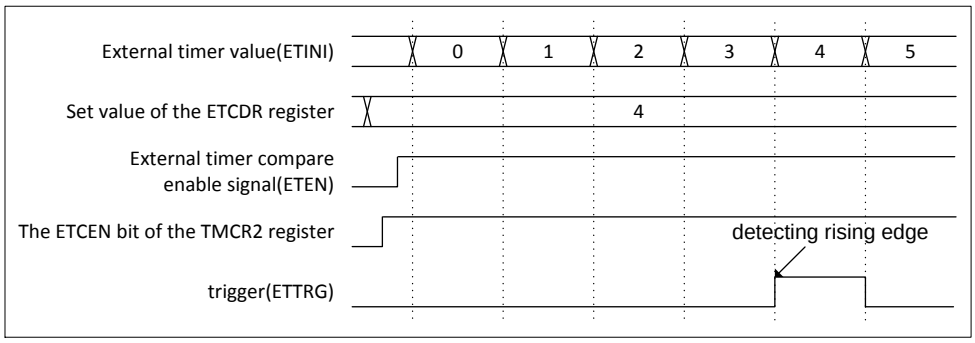
If all the following conditions are satisfied, starting by an external timer is performed.

- The CTEN bit of the TMCR register is "1."
- The ETCEN bit of the TMCR2 register is "1."
- The external timer compare enable signal (ETEN) is active.

When timer is running, the restart by the external timer match starting does not occur. If performing external timer match starting again, perform the following processings.

1. The CTEN bit of the TMCR register is set to "0".
2. The CTEN bit of the TMCR register is set to "1" again.

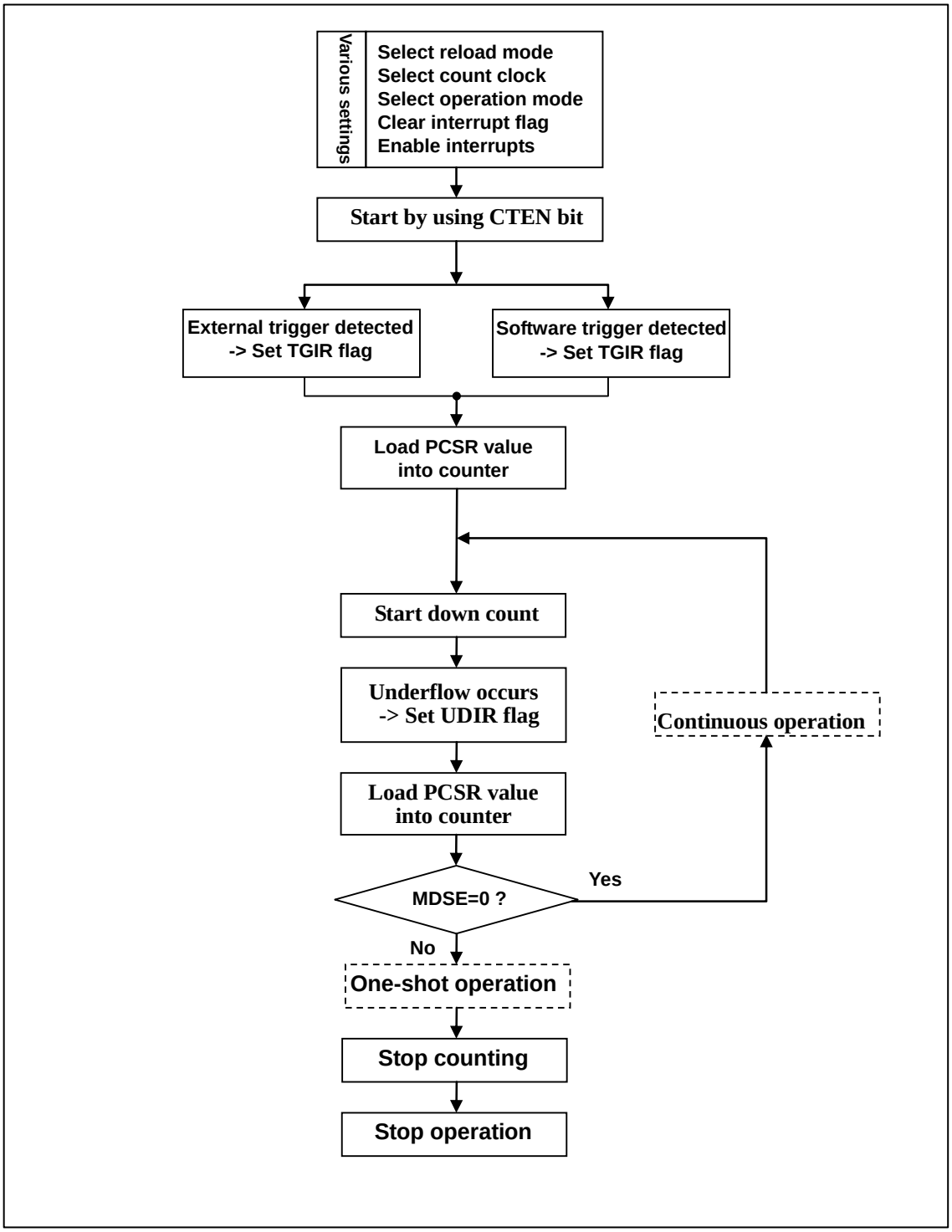
Figure 10-27 The Example of External Timer Match Starting Operation (ETCDR=4)



10.3.3. Reload Timer Operation Flow

This section shows the Reload Timer operation flow.

Reload Timer Operation Flow



10.3.4. 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 timer operation.

For details on writing to the Status Control Register (STC), see "9. Notes on Using the Base Timer".

10.3.4.1 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	0		0	0

[bit15] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

[TMCR2:bit0,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.

CKS3	CKS2	CKS1	CKS0	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] 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.

EGS1	EGS0	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

10.3.4.2 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 "011" 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 "4. 32-Bit Mode Operation")

Bit	Description
0	16-bit timer mode
1	32-bit timer mode

[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.

FMD2	FMD1	FMD0	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 with "L" at count start time
0	1	Toggle output with "H" at 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
0	Stop operation.
1	Enable operation.

[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

10.3.4.3 Timer Control Register 2 (TMCR2)

bit	7	6	5	4	3	2	1	0
Field	GATE	Reserved					ETCEN	CKS3
R/W Attribute	R/W	R0,W0					R/W	R/W
Protection Attribute	-							
Initial Value	0	00000					0	0

[bit7] 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

[bit6:2] Reserved: Reserved Bits

The read value is "0".

If writing to this bit, write "0".

[bit1] ETCEN: External Timer Compare Enable Bit

It is the bit which enables match detection of an external timer value and the ETCDR register value.

Bit	Description
0	Match detection disable
1	Match detection enable

[bit0] CKS3: Count Clock Selection Bit

See "Count clock selection bits" in "10.3.4 Timer Control Registers (TMCR, TMCR2), Status Control Register (STC), Status Control Clear Register (STCC), and Status Control Set Register (STCS)"

10.3.4.4 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

[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.

10.3.4.5 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

[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.

10.3.4.6 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			

[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".

10.3.5. Cycle Setting Register (PCSR)

The Cycle Setting Register (PCSR) is a register that retains the initial count value. In 32-bit mode, if this is an even-numbered channel, this will display the initial value of the lower 16-bit count. If this is an odd-numbered channel, this will display 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	14	13	12	11	10	9	8
Field	PCSR[15:8]							
R/W Attribute	R/W							
Protection Attribute	-							
Initial Value	XXXXXXXX							

bit	7	6	5	4	3	2	1	0
Field	PCSR[7:0]							
R/W Attribute	R/W							
Protection Attribute	-							
Initial Value	XXXXXXXX							

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).

10.3.6. Timer Register (TMR)

The Timer Register (TMR) is a register that can read the count value of a timer. In 32-bit mode, if this is an even-numbered channel, this will display the value of the lower 16-bit count. If this is an odd-numbered channel, this will display 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	14	13	12	11	10	9	8
Field	TMR[15:8]							
R/W Attribute	R/W							
Protection Attribute	-							
Initial Value	XXXXXXXX							

bit	7	6	5	4	3	2	1	0
Field	TMR[7:0]							
R/W Attribute	R/W							
Protection Attribute	-							
Initial Value	XXXXXXXX							

The Timer Register (TMR) 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).

10.3.7. Base Timer Debug Register (BT_DEBUG)

The Base Timer Debug Register (BT_DEBUG) performs enable/disable setting of debug.

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							DBGEN
R/W Attribute	R0,W0							R/W
Protection Attribute	-							
Initial Value	0000000							0

[bit15:1] Reserved: Reserved Bits

The read value is "0".

If writing to this bit, write "0".

[bit0] DBGEN: Debug Enable Bit

■ It is the bit which permits stopping operation of a Base Timer with the debug signal (DEBUG).

Bit	Description
0	Disable
1	Enable

10.3.8. External Timer Compare Data Register (ETCDR)

The External Timer Compare Data Register (ETCDR) is the register which sets the value which is started by an external timer.

bit	15	14	13	12	11	10	9	8
Field	ETCDR[15:8]							
R/W Attribute	R/W							
Protection Attribute	-							
Initial Value	01111111							

bit	7	6	5	4	3	2	1	0
Field	ETCDR[7:0]							
R/W Attribute	R/W							
Protection Attribute	-							
Initial Value	11111111							

It is a register compared with an external timer signal (ETINI). Refer to "10.3.2 External Timer Match Starting" for details.

- Use 16- or 32-bit data access for the ETCDR register.

10.4. 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.

10.4.1 Operations of PWC Timer
10.4.2 Timer Control Registers (TMCR, TMCR2), Status Control Register (STC), Status Control Clear Register (STCC), and Status Control Set Register (STCS)
10.4.3 Data Buffer Register (DTBF)
10.4.4 Base Timer Debug Register (BT_DEBUG)

10.4.1. Operations of PWC Timer

The PWC Timer has a pulse width measurement function. The timer can select 12 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.

10.4.1.1 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 10-28 Pulse Width Measurement Operation (Single Measurement Mode/"H" Width Measurement)

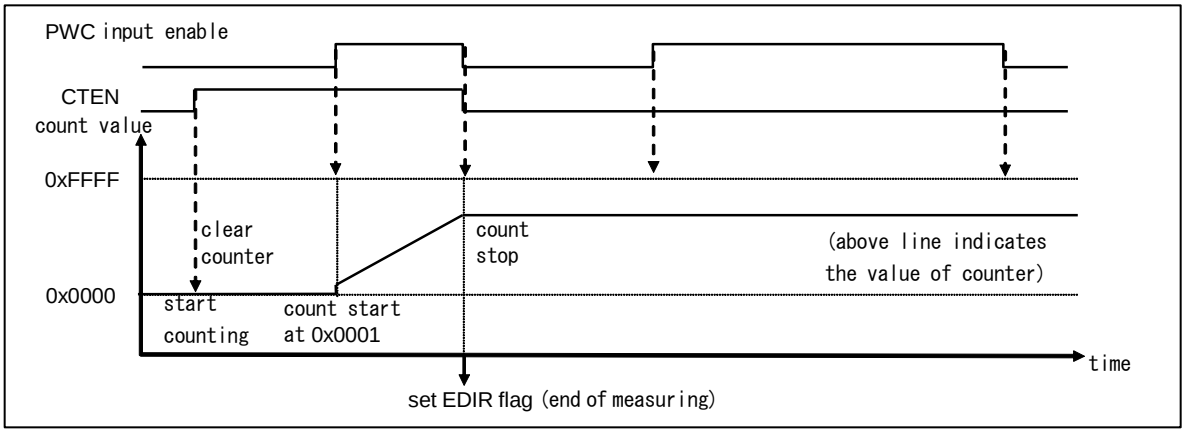
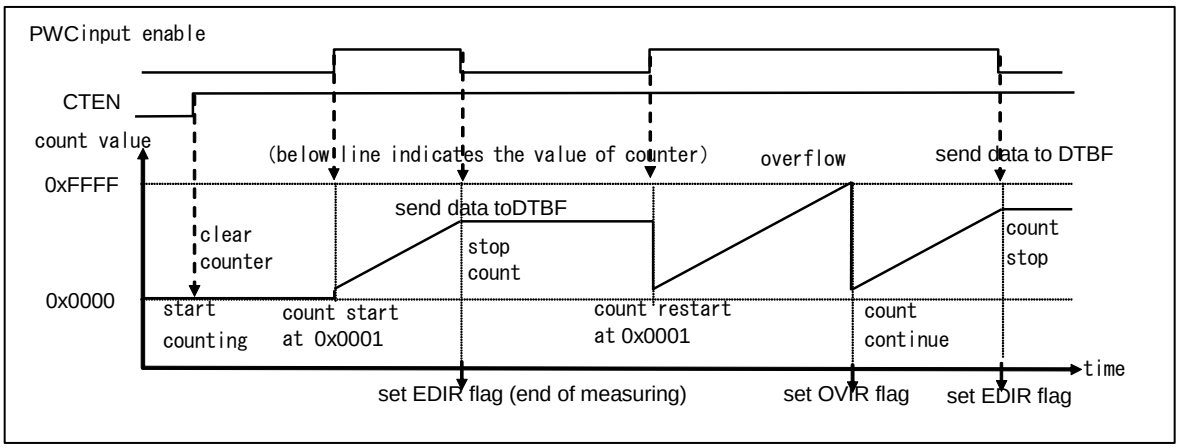


Figure 10-29 Pulse Width Measurement Operation (Continuous Measurement Mode/"H" Width Measurement)



10.4.1.2 Selection of Count Clock

12 types of count clocks can be selected for the counter through the setting of TMCR2 register bit0 (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.

10.4.1.3 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 to falling or falling to 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.

10.4.1.4 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.

10.4.1.5 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.

10.4.1.6 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.

10.4.1.7 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.

10.4.1.8 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

10.4.1.9 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 "10.4.1.3 Selection of Operation Mode") 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:

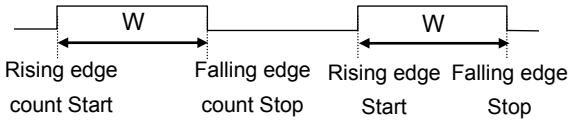
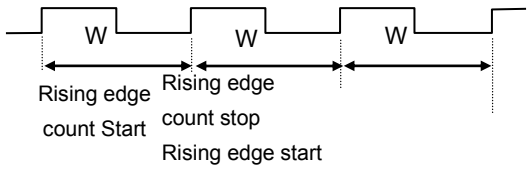
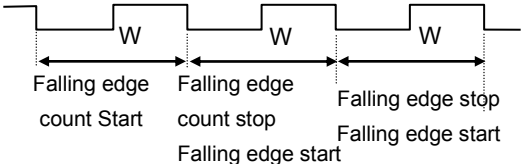
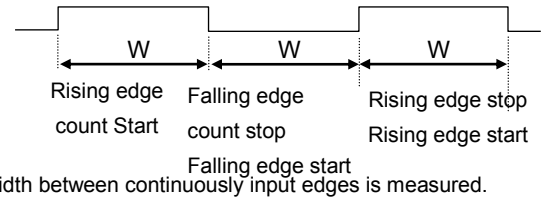
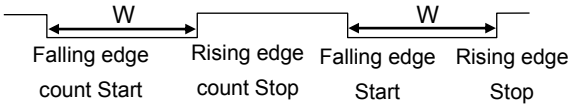
If DTBF is read during operation, the count value being measured is obtained.
If DTBF is read after measurement ends, the measurement result data is obtained.

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 5 types, depending on which part of 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

Regardless of the 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

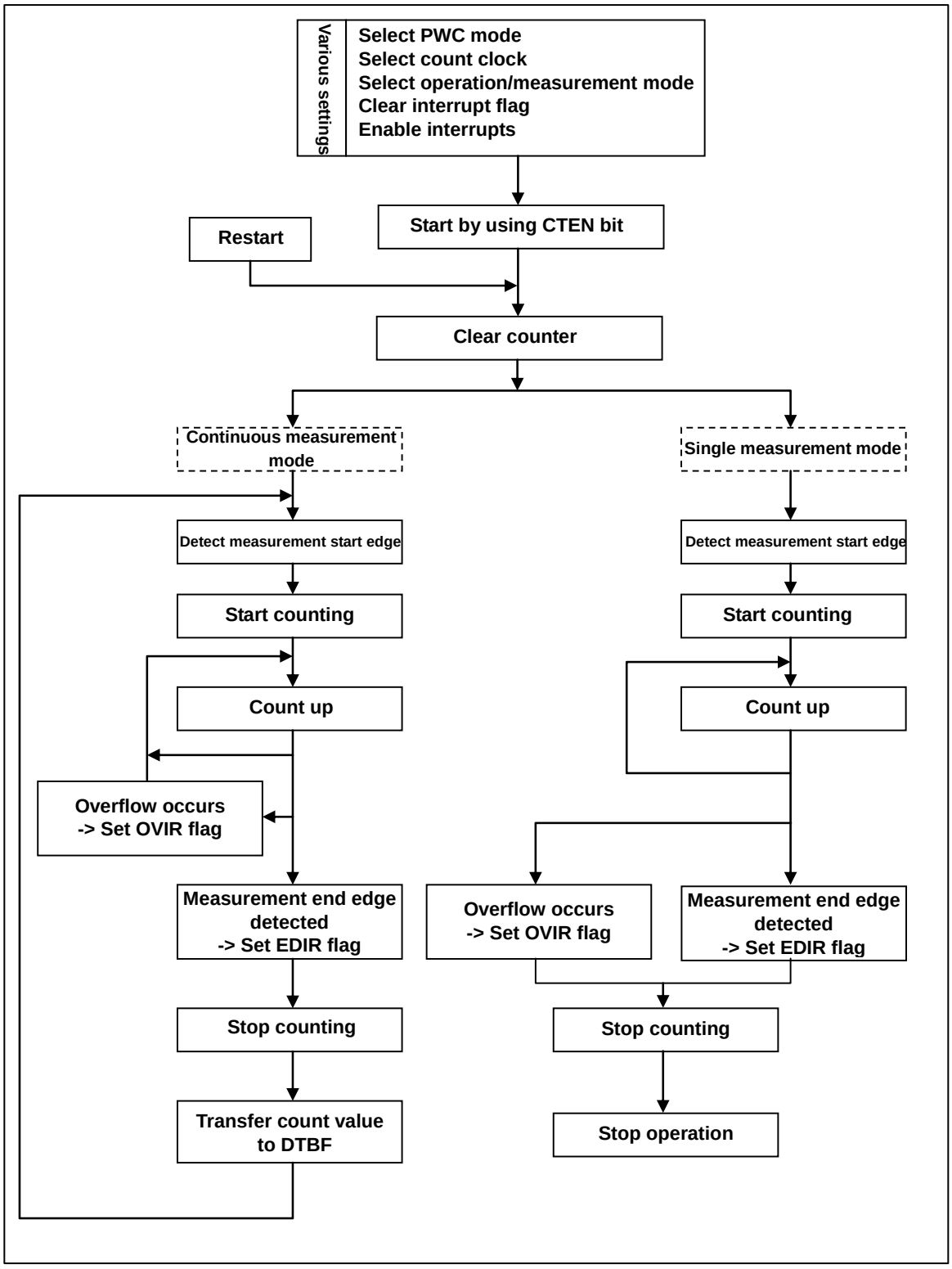
The following two interrupt requests can be generated.

- 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 10-30 Pulse Width Measurement Operation Flow



10.4.2. 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 timer operation.

For details on writing to the Status Control Register (STC), see "9. Notes on Using the Base Timer".

10.4.2.1 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:bit0,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.

CKS3	CKS2	CKS1	CKS0	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] 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.

EGS2	EGS1	EGS0	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	

10.4.2.2 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 "100" 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] 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.

FMD2	FMD1	FMD0	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
0	Stop operation.
1	Enable operation.

[bit0] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

10.4.2.3 Timer Control Register 2 (TMCR2)

bit	7	6	5	4	3	2	1	0
Field	Reserved							CKS3
R/W Attribute	R0,W0							R/W
Protection Attribute	-							
Initial Value	0000000							0

[bit7:1] Reserved: Reserved Bits

The read value is "0".

If writing to this bit, write "0".

[bit0] CKS3: Count Clock Selection Bit

See "Count clock selection bits" in "10.4.2 Timer Control Registers (TMCR, TMCR2), Status Control Register (STC), Status Control Clear Register (STCC), and Status Control Set Register (STCS)"

10.4.2.4 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

[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.

10.4.2.5 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

[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.

10.4.2.6 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			

[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".

10.4.3. 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, if this is an even-numbered channel the lower 16 bits will be read similarly, for odd number channels, the upper 16 bits will be read.

Be sure to read this register with 16- or 32-bit data transfer instructions.

bit	15	14	13	12	11	10	9	8
Field	DTBF[15:8]							
R/W Attribute	R,WX							
Protection Attribute	-							
Initial Value	00000000							

bit	7	6	5	4	3	2	1	0
Field	DTBF[7:0]							
R/W Attribute	R,WX							
Protection Attribute	-							
Initial Value	00000000							

- 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.

10.4.4. Base Timer Debug Register (BT_DEBUG)

The Base Timer Debug Register (BT_DEBUG) performs enable/disable setting of debug.

bit	15							8
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial Value	00000000							

bit	7							0
Field	Reserved							DBGEN
R/W Attribute	R0,W0							R/W
Protection Attribute	-							
Initial Value	0000000							0

[bit15:1] Reserved: Reserved Bits

The read value is "0".

If writing to this bit, write "0".

[bit0] DBGEN: Debug Enable Bit

- It is the bit which permits stopping operation of a Base Timer with the debug signal (DEBUG).

Bit	Description
0	Disable
1	Enable

CHAPTER 39: 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**

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.

The following 7 patterns can be selected for I/O pin connections.

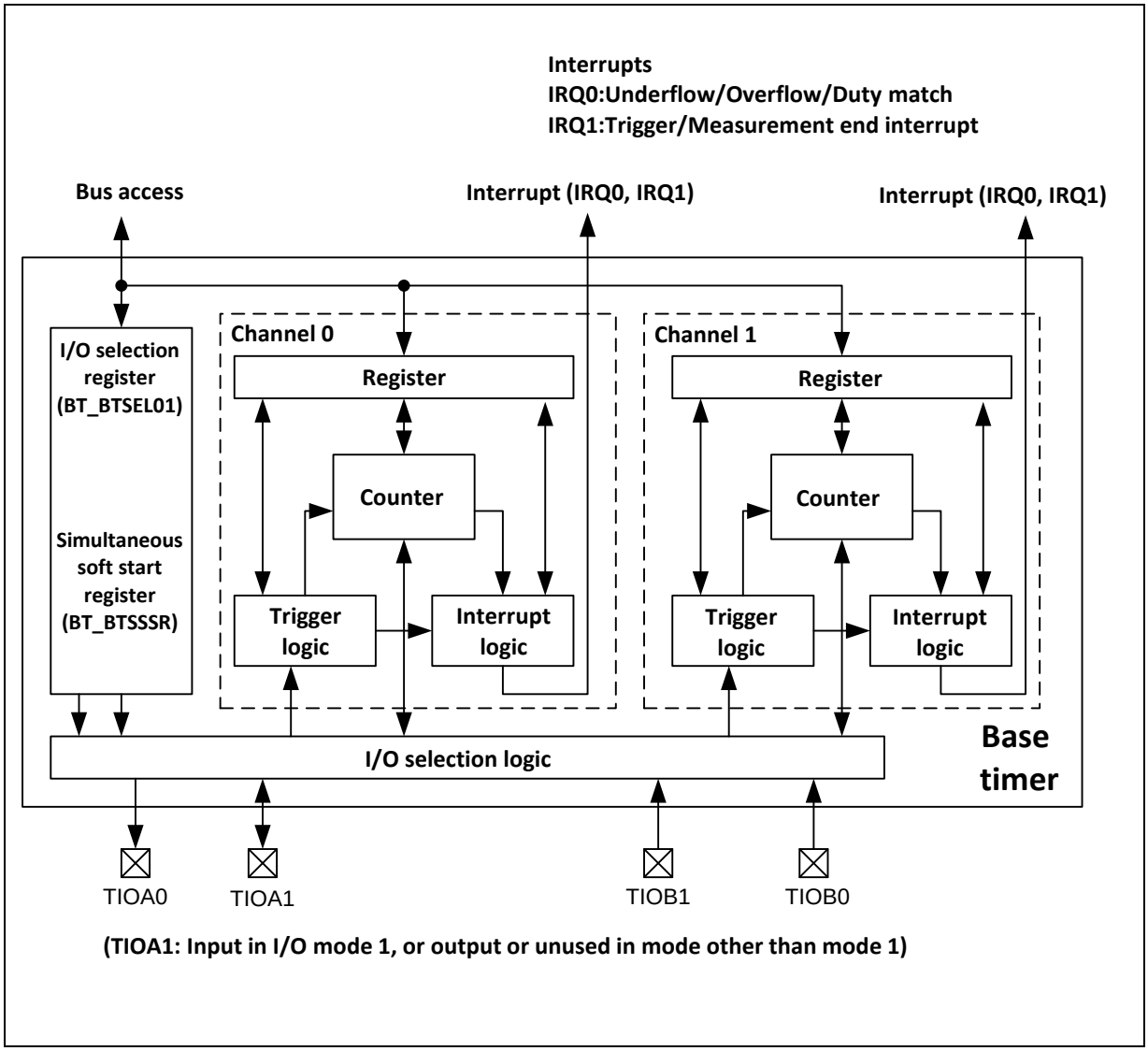
- 16-bit timer standard mode
- 32-bit timer full-function mode
- PPG trigger 2-channel sharing mode
- Timer start/stop mode
- Simultaneous soft start mode
- Timer start/stop and simultaneous soft start mode
- Timer start mode

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

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

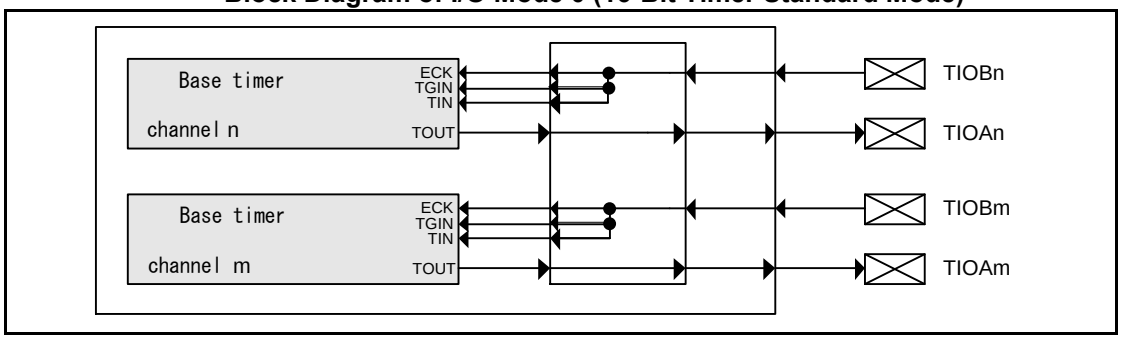
This section explains base timer I/O assignment.

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.

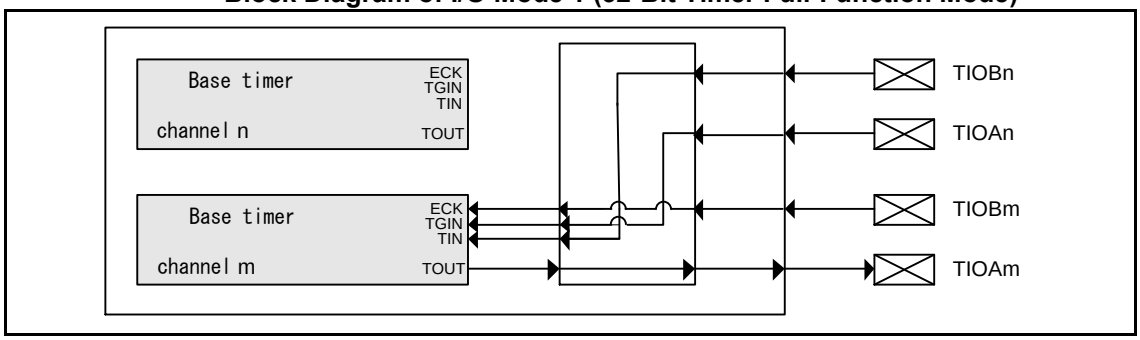
- I/O mode 0: 16-bit timer standard mode
  In this mode, the base timer in 1 channel operates individually and separately from the others.
- I/O mode 1: 32-bit timer full-function mode
  The signals of the even-numbered channels of the base timer are individually assigned to external pins in operation 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.

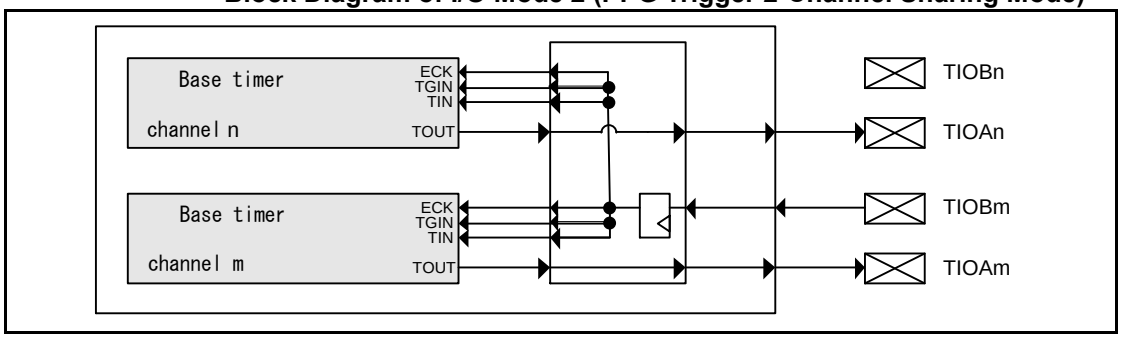
Block Diagram of I/O Mode 0 (16-Bit Timer Standard Mode)



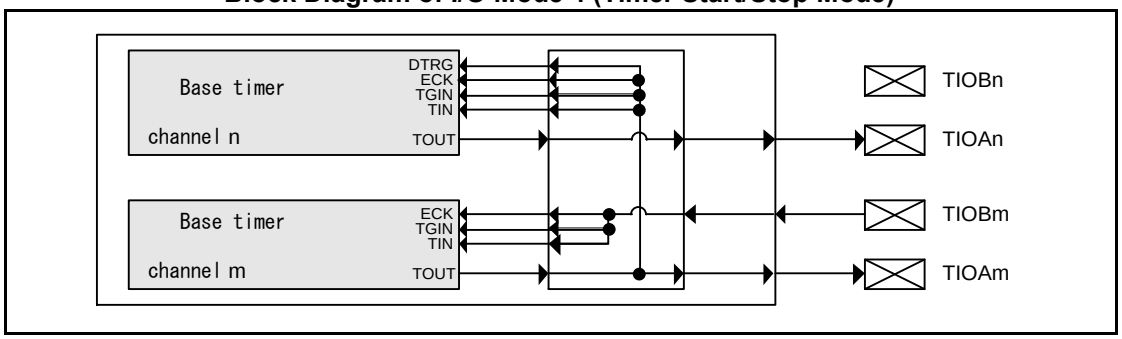
Block Diagram of I/O Mode 1 (32-Bit Timer Full-Function Mode)



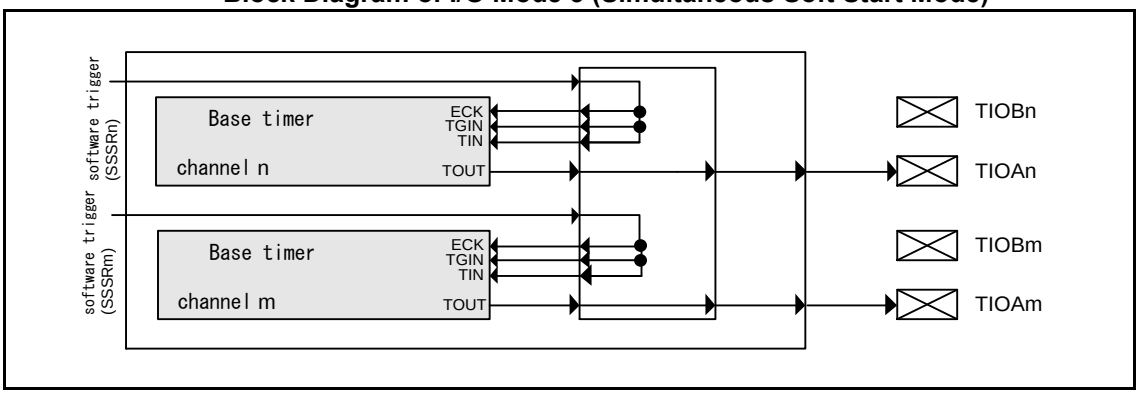
Block Diagram of I/O Mode 2 (PPG Trigger 2-Channel Sharing Mode)



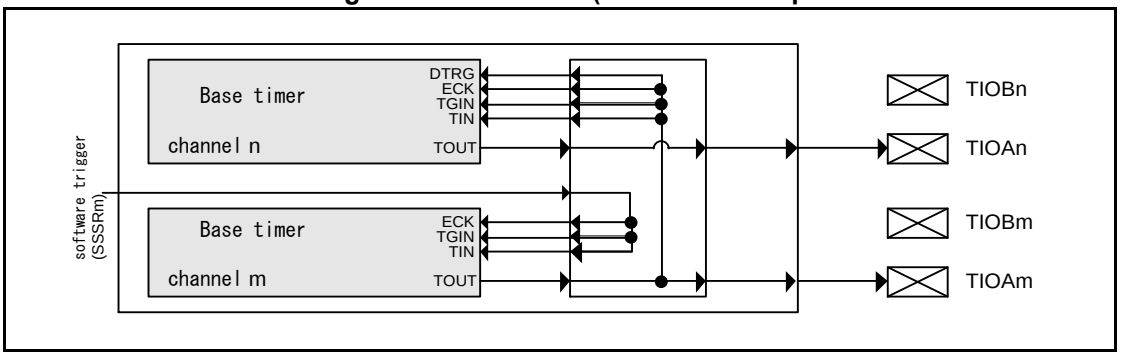
Block Diagram of I/O Mode 4 (Timer Start/Stop Mode)



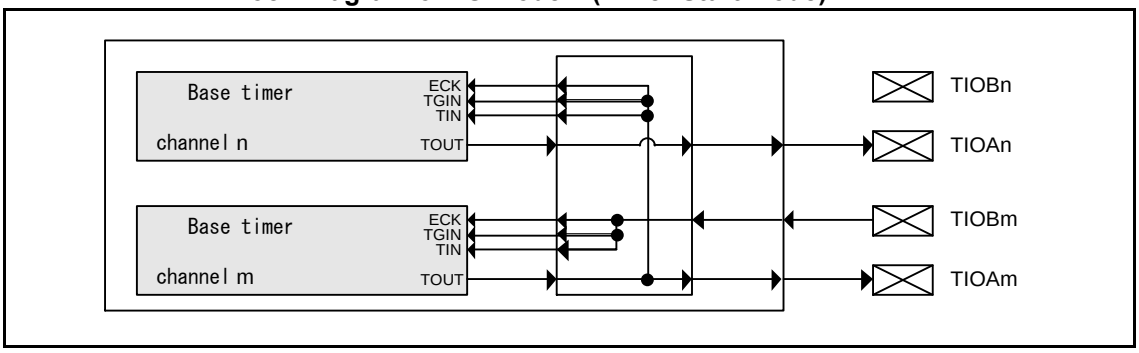
Block Diagram of I/O Mode 5 (Simultaneous Soft Start Mode)



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)	4.2
BT_BTSEL1415	I/O selection register (channel 14, 15)	4.2
BT_BTSEL1617	I/O selection register (channel 16, 17)	4.2
BT_BTSEL1819	I/O selection register (channel 18, 19)	4.2
BT_BTSEL2021	I/O selection register (channel 20, 21)	4.2
BT_BTSEL2223	I/O selection register (channel 22, 23)	4.2
BT_BTSEL2425	I/O selection register (channel 24, 25)	4.3
BT_BTSEL2627	I/O selection register (channel 26, 27)	4.3
BT_BTSEL2829	I/O selection register (channel 28, 29)	4.3
BT_BTSSSR0	Simultaneous soft start register(channel0～11)	4.4
BT_BTSSSR12	Simultaneous soft start register(channel12～23)	4.5
BT_BTSSSR24	Simultaneous soft start register(channel24～29)	4.6

4.1. I/O Selection Registers (BT_BTSEL01, BT_BTSEL23, BT_BTSEL45, BT_BTSEL67, BT_BTSEL_89, BT_BTSEL1011)

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				BTSEL01,23,45,67,89,1011			
R/W Attribute	R1,WX				R/W			
Protection Attribute	-							
Initial Value	1111				0000			

[bit31:4] Reserved: Reserved Bits

[bit3:0] BTSEL01,23,45,67,89,1011[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.

bit3:0				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:

- *You cannot initialize this register by setting reset mode (TMCR:FMD2 to 0="0b000"). After setting reset mode, rewrite this register.*

4.2. I/O Selection Registers (BT_BTSEL1213, BT_BTSEL1415, BT_BTSEL1617, BT_BTSEL1819, BT_BTSEL_2021, BT_BTSEL2223)

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				BTSEL1213,1415,1617,1819,2021,2223			
R/W Attribute	R1,WX				R/W			
Protection Attribute	-							
Initial Value	1111				0000			

[bit31:4] Reserved: Reserved Bits

[bit3:0] BTSEL1213,1415,1617,1819,2021,2223[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.

bit3:0				Description
0	0	0	0	Setting prohibited
0	0	0	1	Setting prohibited
0	0	1	0	Setting prohibited
0	0	1	1	Setting prohibited
0	1	0	0	Setting prohibited
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	Setting prohibited
1	x	x	x	Setting prohibited

Note:

- *You cannot initialize this register by setting reset mode (TMCR:FMD2 to 0="0b000"). After setting reset mode, rewrite this register.*

4.3. I/O Selection Registers (BT_BTSEL2425, BT_BTSEL2627, BT_BTSEL2829)

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				BTSEL2425,2627,2829			
R/W Attribute	R1,WX				R/W			
Protection Attribute	-							
Initial Value	1111				0000			

[bit31:4] Reserved: Reserved Bits

[bit3:0] BTSEL2425,2627,2829[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.

bit3:0				Description
0	0	0	0	Setting prohibited
0	0	0	1	Setting prohibited
0	0	1	0	Setting prohibited
0	0	1	1	Setting prohibited
0	1	0	0	Setting prohibited
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	Setting prohibited
1	x	x	x	Setting prohibited

Note:

- *You cannot initialize this register by setting reset mode (TMCR:FMD2 to 0="0b000"). After setting reset mode, rewrite this register.*

4.4. Simultaneous Soft Start Register (BT_BTSSSR0)

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] 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."

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.

Bit [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.

4.5. Simultaneous Soft Start Register (BT_BTSSSR12)

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				SSSR23	SSSR22	SSSR21	SSSR20
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	SSSR19	SSSR18	SSSR17	SSSR16	SSSR15	SSSR14	SSSR13	SSSR12
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] SSSR23to SSSR12: 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" is invalid. Up to 12 channels with channel numbers 12 to 23 can be started simultaneously.

Bit [X]	Description
0	No operation
1	Assigns the "1" pulse to input and starts the corresponding channel.

[X] is a value from 0 to 11. (X+12) represents the channel number of the base timer.

4.6. Simultaneous Soft Start Register (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	30	29	28	27	26	25	24	23	22	21	20	19	18	17	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							
R/W Attribute	R1, WX							
Protection Attribute	-							
Initial Value	11111111_11111111							

Bit	7	6	5	4	3	2	1	0
Field	Reserved		SSSR29	SSSR28	SSSR27	SSSR26	SSSR25	SSSR24
R/W Attribute	R1, WX		R1,W	R1,W	R1,W	R1,W	R1,W	R1,W
Protection Attribute	-							
Initial Value	11		1	1	1	1	1	1

[bit31:12] Reserved: Reserved Bits

[bit11:0] SSSR29 to SSSR24: 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" is invalid. Up to 6 channels with channel numbers 24 to 29 can be started simultaneously.

Bit [X]	Description
0	No operation
1	Assigns the "1" pulse to input and starts the corresponding channel.

[X] is a value from 0 to 5. (X+24) represents the channel number of the base timer.

CHAPTER 40: Base Timer Simultaneous Operation

This chapter explains the base timers simultaneous Soft Start .

1. Overview
2. Configuration
3. Explanation of Simultaneous Soft Start Operation
4. Overview of the 16-bit Global Timer
5. Explanation of the 16-bit Global Timer Operation
6. Registers of the 16-bit Global Timer
7. Precautions for Using This Device

1. Overview

This section provides an overview of the Base Timer simultaneous soft start.

There are the undermentioned two operational modes in the soft start of the Base timer simultaneous.

- The operation by the write to simultaneous soft start registers.
  (Refer to CHAPTER 39 Base Timer I/O Selection Function)
- The operation which uses an external timer
  (Refer to CHAPTER 38 Base Timer 10.1.6 External Timer Match Starting)

This Chapter describes the operation which uses an external timer.
As an external Timer, it uses Global Timer described in this Chapter.

2. Configuration

This section explains the configuration of the Base Timer simultaneous start which uses a Global Timer

From the Global Timer, the undermentioned signals are outputted and it is inputted into each Base Timer.

- Global timer value

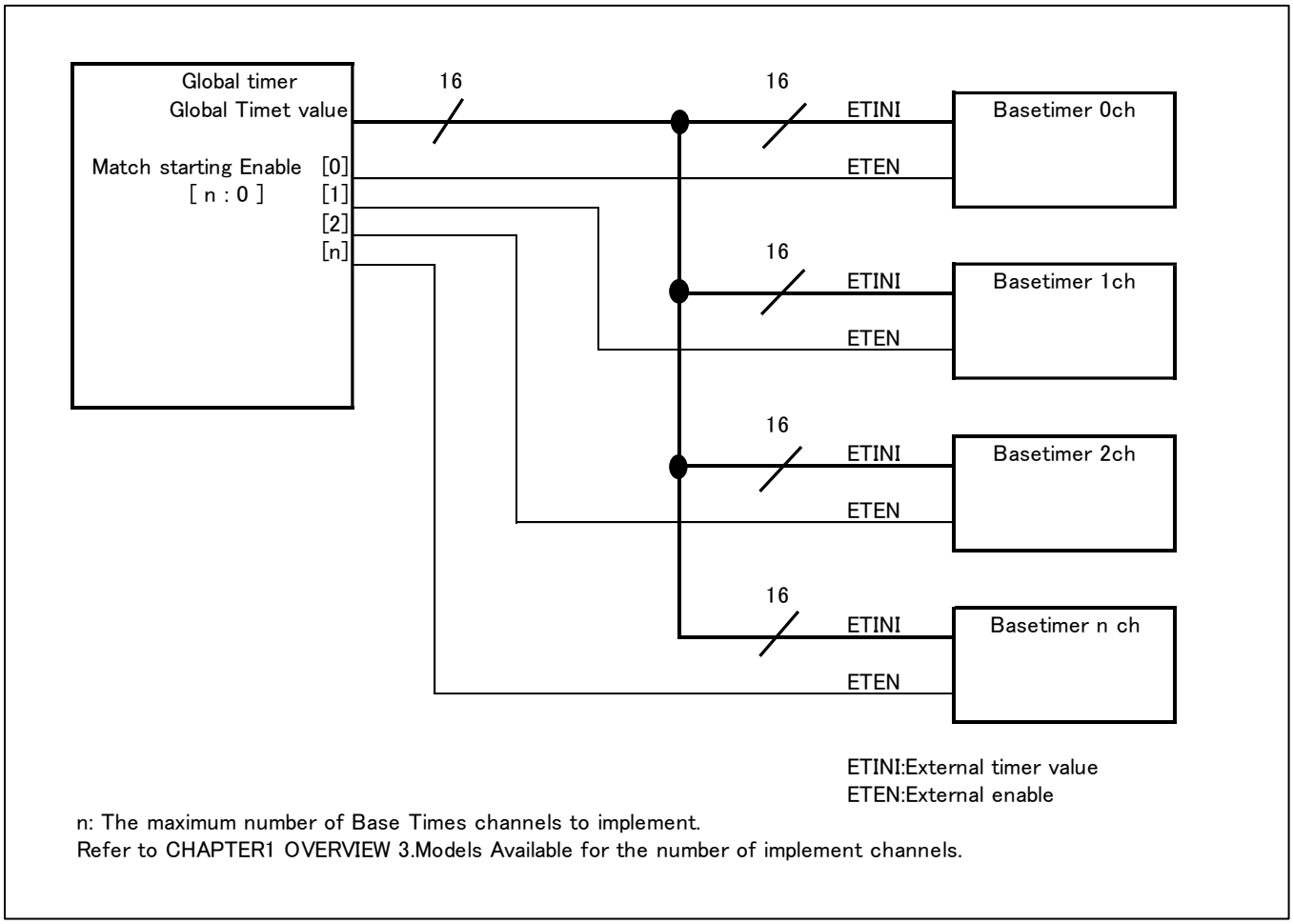
  It is a 16-bit timer counter value in the Global Timer.

- Match starting Enable

  It outputs, the Global Timer match enabling signal to each base timer channel.

.

Figure 2-1 Block Diagram



n: The maximum number of Base Times channels to implement.
Refer to CHAPTER1 OVERVIEW 3.Models Available for the number of implement channels.

3. Explanation of Simultaneous Soft Start Operation

This section explains the operation of Base Timer Simultaneous Soft Start.

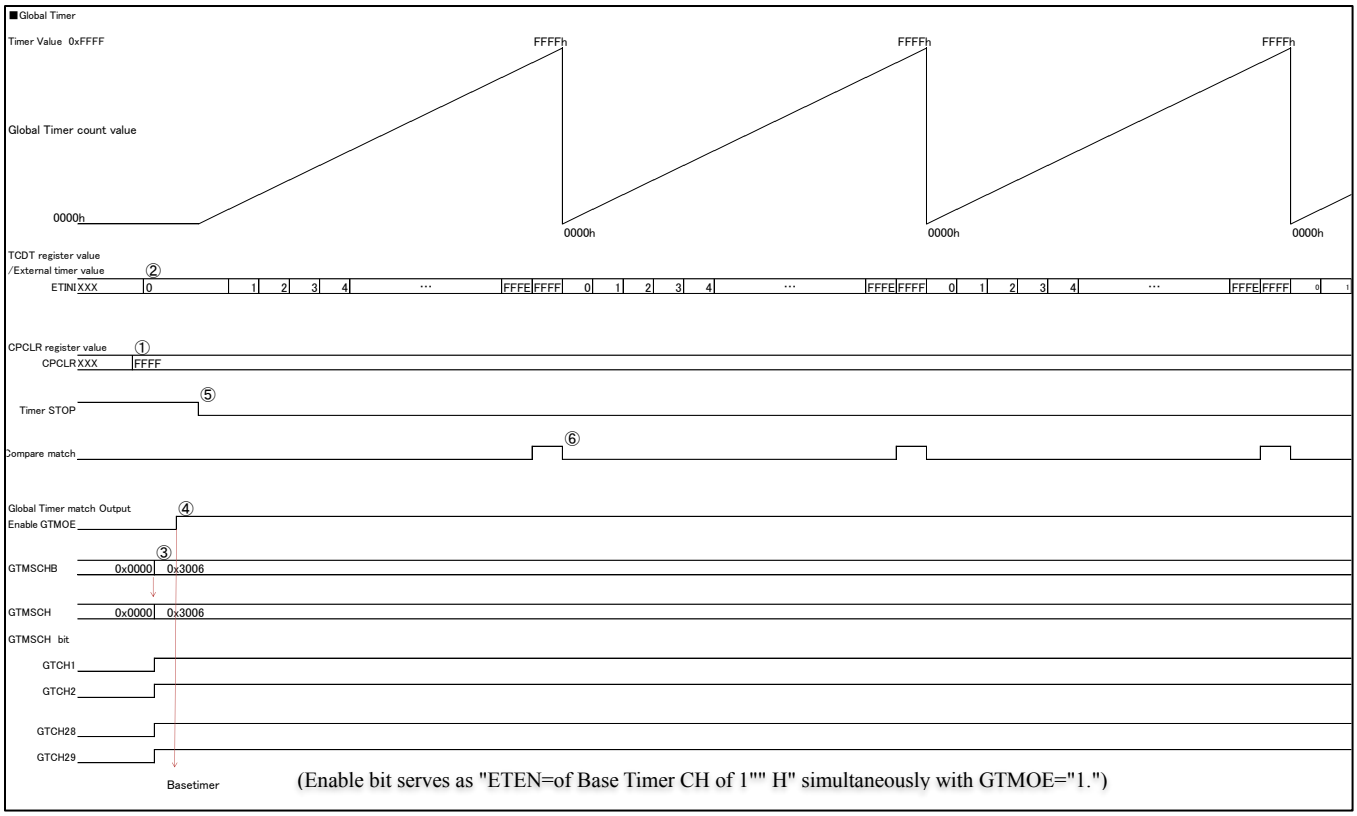
The Procedure in which the simultaneous soft start of the Base Timer was simplified below is shown.

1. It performs an output set of a waveform set of the Base Timer, etc. of operation
2. It performs a set of the Global Timer of operation.
3. Initialization of Global Timer.
4. Set of the Base Timer channel to use an external timer match startup.
5. Global Timer Start.
6. A match of the Global Timer value and the value of external timer compare data in the Base Timer will start the Base Timer.

3.1. Global Time and Base Timers Operation Waveform

The waveform of operation which controls STOP bit in Timer State Control Register (TCCS) starts, and starts Global Timer, and each Base Timers starts is shown below.

Figure 3-1 Global Timer Waveform (It Sets a STOP Bit As "0" From "1", And Starts Global Timer.)



① Global Timer various sets.

It writes an any value in the Global Clear Register (CPCLR), and sets up the Global Timer period.（ex:0xFFFF）

It writes any value in GT_CLK [3:0], and sets up Global Timer counter clock frequency.

MODE set by GT_MSCTR.MODE bit.

(In the above-mentioned waveform, MODE Bit is "0" and is buffer disable.)

② It writes "0x0000" in the Global Timer Data Register (GT_TCDT), and clears the Global Timer value.

③ It writes an arbitrary value in Global Timer Match Channel Buffer Register (GT_MSCHB).

(It writes "1" in Global Timer Match Starting Channel Buffer Bits (GTCH) of the Base Timer channel to use an external timer match start.)

Since a timer is stopping, the Global Timer Match Channel Buffer Register (GT_MSCHB) set point is immediately transmitted to Global Timer Match Starting Channel register (GT_MSCH).

For output disable (GTMOE="0"), All External Timer Match Enabling Signals (ETEN) are "L."

④ After a Global Timer Match Channel Buffer Register (GT_MSCHB) set, it sets it as GTMOE bit ="1" (Enable).

According to the Global Timer Match Starting Channel register (GT_MSCH) set point, the External Timer Match Enable Signal (ETEN) of each Base Timer Channel changes to "H".

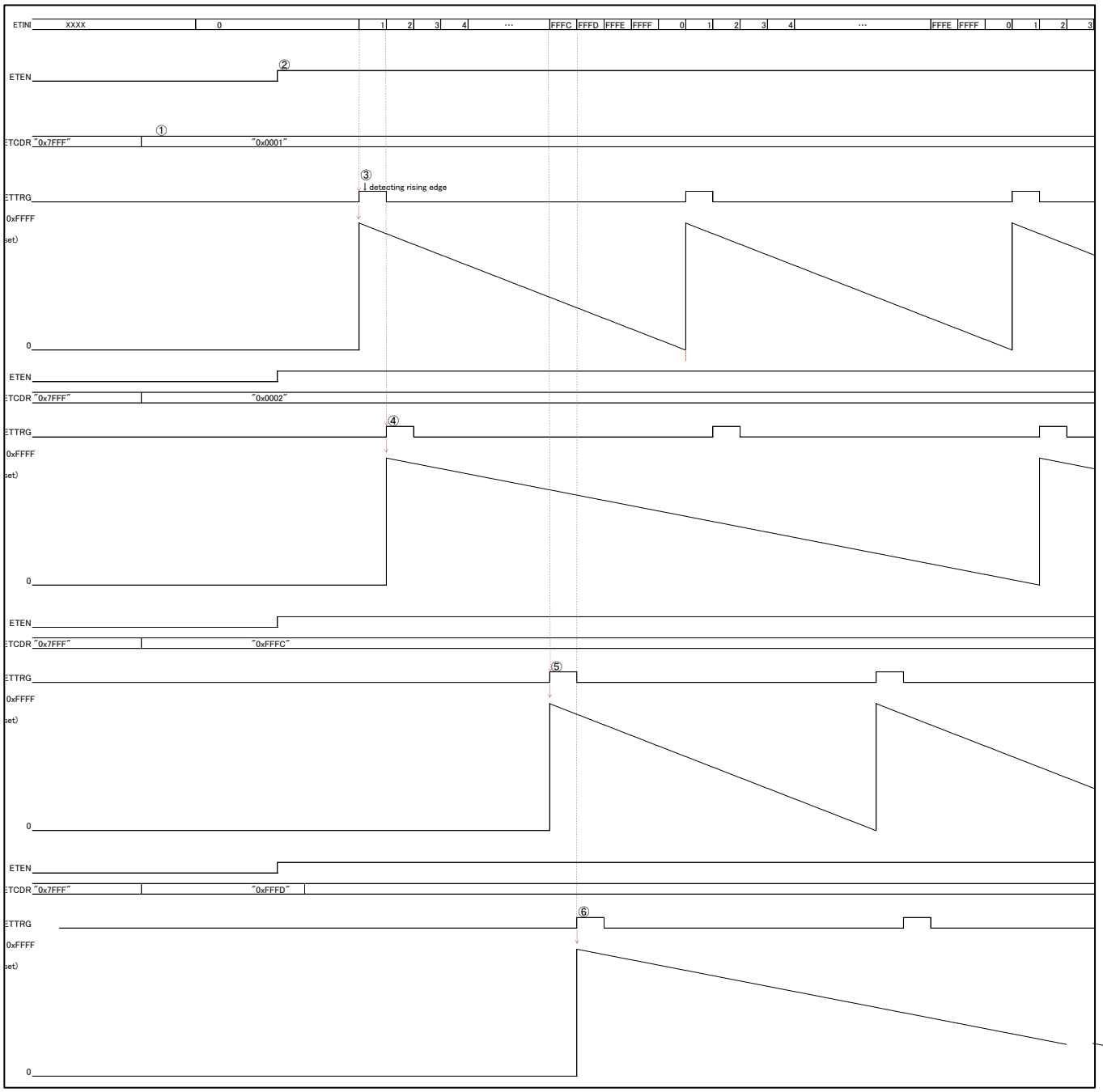
⑤ It writes "0" in STOP bit of Global Timer State Control Register(GT_TCCS).

Global Timer start. (It does not stop Global Timer in start once.)

⑥ When a match the Global Timer Compare Clear Register (GT_CPCLR) value and Global Timer value, it is a comparison clear flag generate.

In accordance with the setup value, the Base Timer operates after the Global Timer start.

Figure 3-2 Base Timer Waveform



① Base timer various sets. (For details, refer to CHAPTER38 Base Timer)

Note:

- *Please set ETCDR of the channel which uses an external timer match start as a larger value than "0x0000".*

② Simultaneously with GTMOE="1", the ETEN signal of Base Timer CH by which the start set was carried out. "H" It becomes.

③ When a match the External Time Comparison Data Register(ETCDR) value and a Global Timer value , ETTRG signal occurs and Base Timer start.(CH1)

④ When a match the External Time Comparison Data Register(ETCDR) value and a Global Timer value , ETTRG signal occurs and Base Timer start.(CH2)

⑤ When a match the External Time Comparison Data Register(ETCDR) value and a Global Timer value , ETTRG signal occurs and Base Timer start.(CH28)

⑥ When a match the External Time Comparison Data Register(ETCDR) value and a Global Timer value , ETTRG signal occurs and Base Timer start.(CH29)

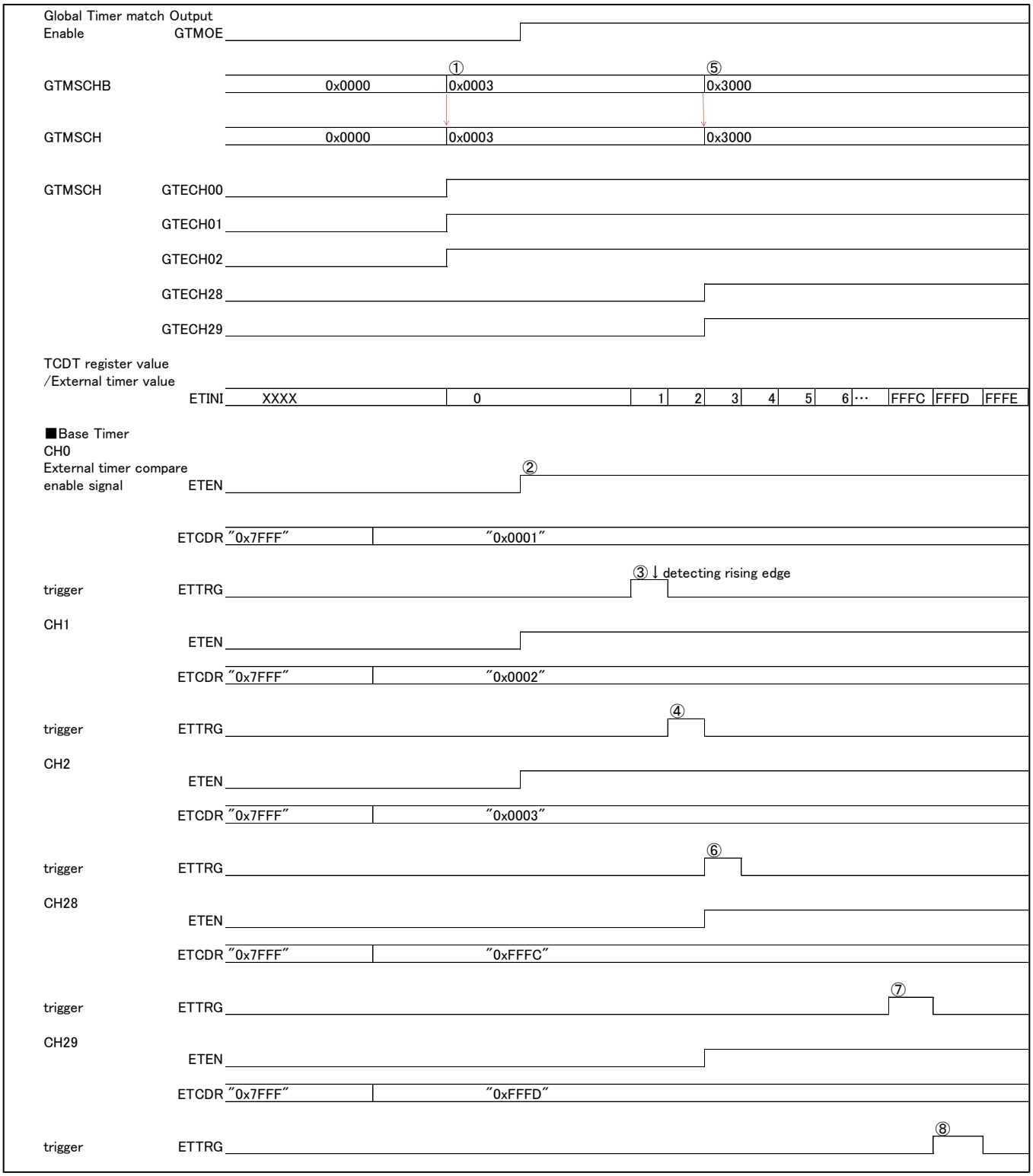
By adding the channel which carries out a Simultaneous Soft Start to GT_MSCHB Register, and carrying out an enable set, it can carry out the additional operation of Base Timers also after the Global Timer operation.

In that case, Base Timers which want to run an additional simultaneous soft start before needs to be set up.

3.2. Global Timer Match Starting Channel Buffer Operation by MODE Set

The MODE value in Global Timer Match Starting Control register, the update time of the Global Timer Match Starting Channel register value has a difference.

Figure 3-3 The Waveform in the Case of Adding Base Timers CH of the Global Timer Match Start Enable. (MODE=0)



① It is a start set from Base Timers CH0 to CH3.

A data is immediately updated by GTMSCH for MODE=0, without carrying out a buffer.

② Simultaneously with GTMOE="1", the ETEN signal of Base Timer CH by which the start set was carried out. "H" It becomes.

③ When a match the External Time Comparison Data Register(ETCDR) value and the Global Timer value , ETTRG signal occurs and Base Timer starts.(CH0)

④ When a match the External Time Comparison Data Register(ETCDR) value and the Global Timer value , ETTRG signal occurs and Base Timer starts.(CH1)

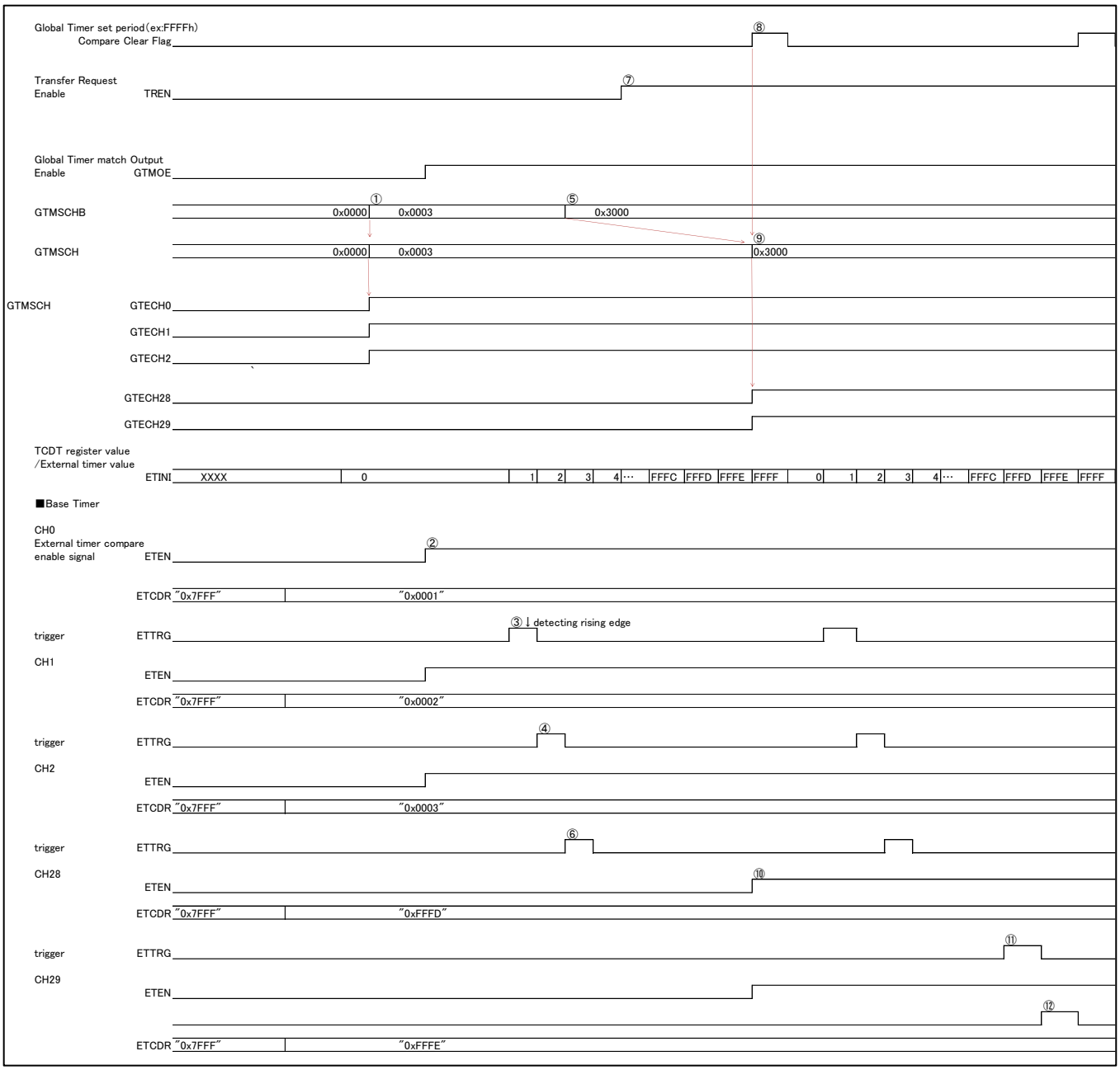
⑤ It is a start set from Base Timers CH28 to CH29.

⑥ When a match the External Time Comparison Data Register(ETCDR) value and the Global Timer value , ETTRG signal occurs and Base Timer starts.(CH2)

⑦ When a match the External Time Comparison Data Register(ETCDR) value and the Global Timer value , ETTRG signal occurs and Base Timer starts.(CH28)

⑧ When a match the External Time Comparison Data Register(ETCDR) value and the Global Timer value , ETTRG signal occurs and Base Timer starts.(CH29)

Figure 3-4 The Waveform in the Case of Adding Base Timers CH of the Global Timer Match Start Enable. (MODE=1)



① It is a start set from Base Timers CH0 to CH3.

A data is immediately updated by GTMSCH for MODE=0, without carrying out a buffer.

② Simultaneously with GTMOE="1", the ETEN signal of Base Timer CH by which the start set was carried out. "H" It becomes.

③ When a match the External Time Comparison Data Register(ETCDR) value and the Global Timer value , ETTRG signal occurs and Base Timer starts.(CH0)

④ When a match the External Time Comparison Data Register(ETCDR) value and the Global Timer value , ETTRG signal occurs and Base Timer starts.(CH1)

⑤ It is a start set from Base Timers CH28 to CH29.

It is not updated until the period comparison match flag of Global Timer is outputted for buffer enable mode.

⑥ When a match the External Time Comparison Data Register(ETCDR) value and the Global Timer value , ETTRG signal occurs and Base Timer starts.(CH2)

⑦ Enable the transfer request of Register data.

⑧ By the signal outputted for every period of setup Global Timer.

⑨ Global Timer Match Starting Channel register (GT_MSCH) is updated.

⑩ Simultaneously with GTMOE="1", the ETEN signal of Base Timer CH by which the start set was carried out. "H" It becomes.

⑪ When a match the External Time Comparison Data Register(ETCDR) value and the Global Timer value , ETTRG signal occurs and Base Timer starts.(CH28)

⑫ When a match the External Time Comparison Data Register(ETCDR) value and the Global Timer value , ETTRG signal occurs and Base Timer starts.(CH29)

4. Overview of the 16-bit Global Timer

The 16-bit Global Timer supports the 16-bit up count mode.

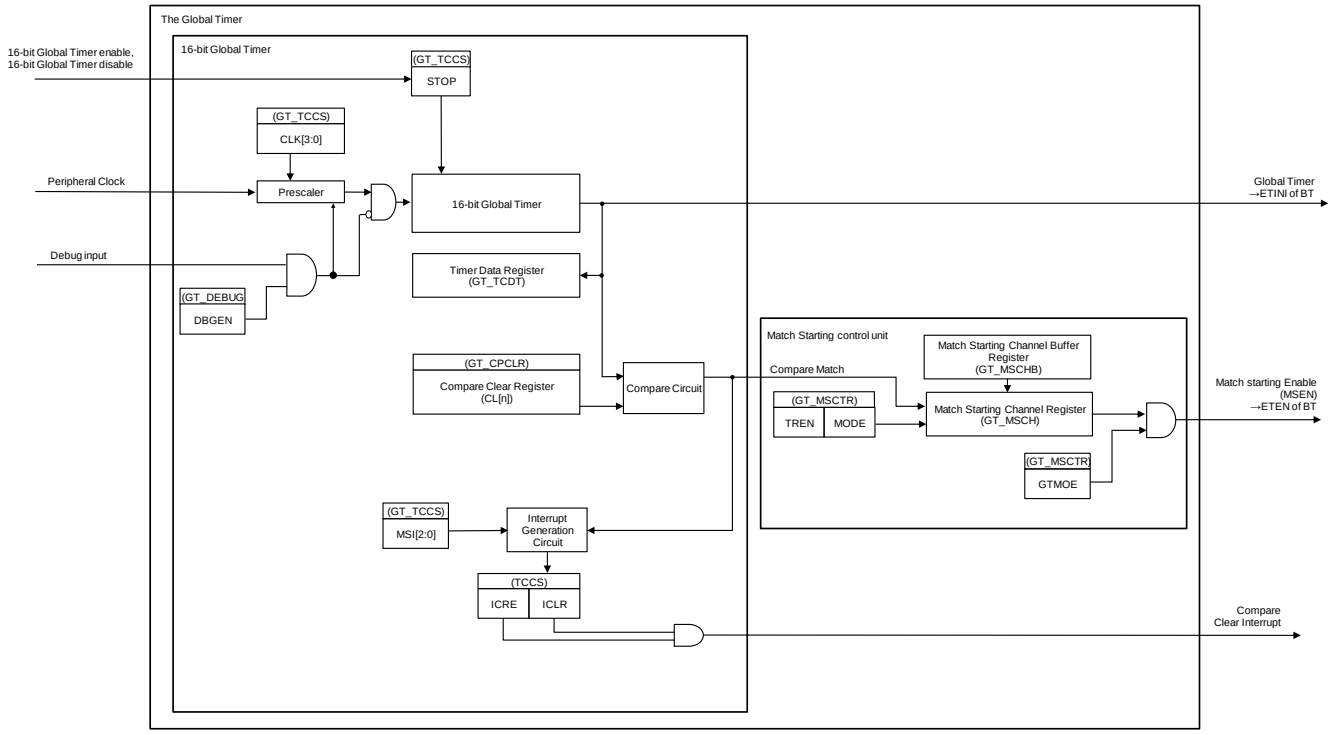
Functions of the 16-bit Global Timer

- The 16-bit Global Timer is composed of the 16-bit up counter, control register, 16-bit compare clear register, and prescaler.
- The 12 types of counter operation clocks (φ, φ/2, φ/4, φ/8, φ/16, φ/32, φ/64, φ/128, φ/256, φ/512, φ/1024, and φ/2048) (φ: peripheral clock: CLK_LCP0A) can be selected.
- A compare clear interrupt is generated when the values of the compare clear register and the 16-bit counter are compared and their values match.
- The compare clear register has a buffer register (the data written to this buffer register is transferred to the compare clear register). When the 16-bit counter is stopped, the data is transferred as soon as data is written to the buffer. When the 16-bit counter is operating, data is transferred from the buffer upon detecting the count value "0000H".
- The count value is reset to "0000H" when a hardware reset occurs; software clears the timer, or the value of the compare clear register and the count value match.

4.1. Configuration Diagram of the 16-bit Global Timer

Figure **4-1** is a configuration diagram of the 16-bit Global Timer.

Figure 4-1 **Configuration Diagram of 16-bit Global Timer**



5. Explanation of the 16-bit Global Timer Operation

This section provides a summary of the operation of the 16-bit Global Timer.

Operation of the 16-bit Global Timer

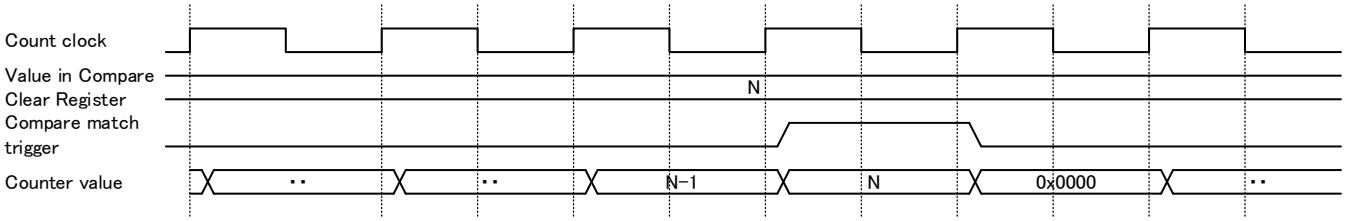
The 16-bit Global Timer starts counting from the value set in the Global Timer Data Register (GT_TCDT) after the timer is set to enabled (GT_TCCS:STOP).

Counter clear

The count value of the Global Timer is cleared to 0 when any of the following conditions is met.

- When the count value matches the value of the Global Timer Compare Clear Register (GT_CPCLR) in the up count.
- When "0000H" is written to the Global Timer Data Register (GT_TCDT) while the Global Timer is stopped (STOP: bit6=1 of the Global Timer State Control Register (GT_TCCS))
- When the hardware is reset. Upon reset, the counter is cleared immediately.

Figure 5-1 Clear Timing of 16-bit Global Timer



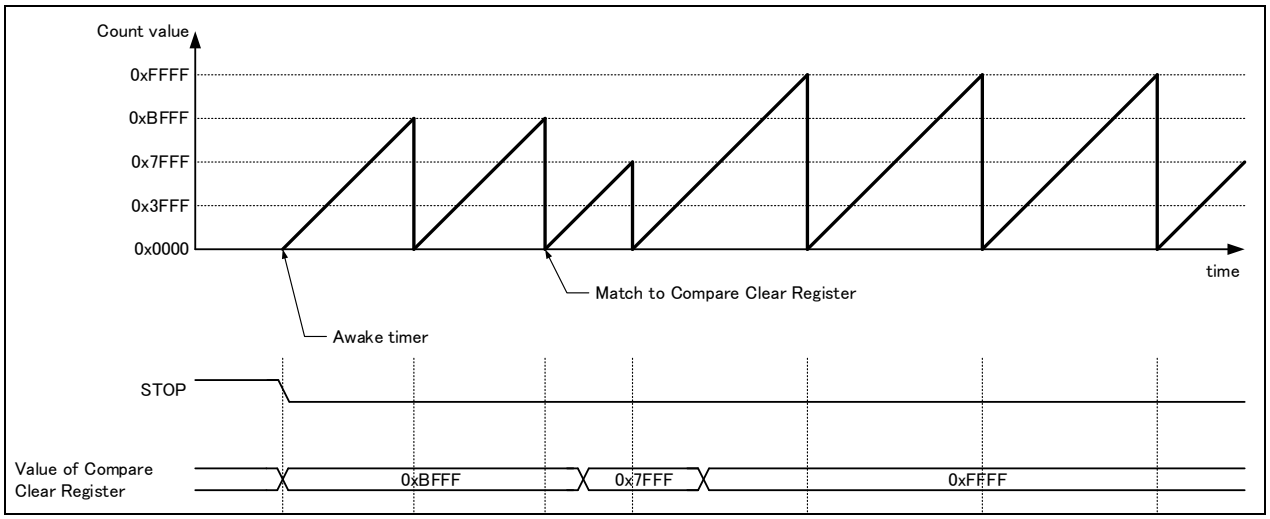
Note:

- *Count clock is the count operation clocks.*

Compare clear

The data can be directly written to the Global Timer Compare Clear Register (GT_CPCLR).

Figure 5-2 **Up Count Operation**

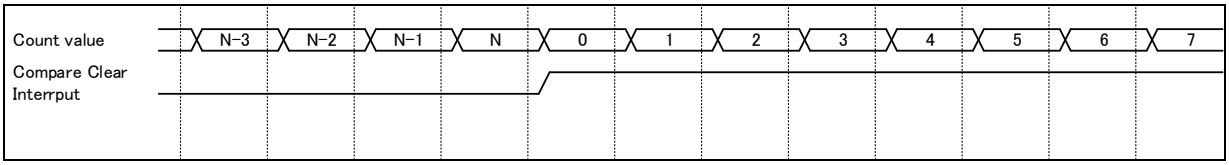


Timer interrupt

The 16-bit Global Timer can generate the following interrupts.

- Compare clear interrupt

Figure 5-3 Interrupt Generated in Up Count



Interrupt mask function

The compare clear interrupt can be masked.

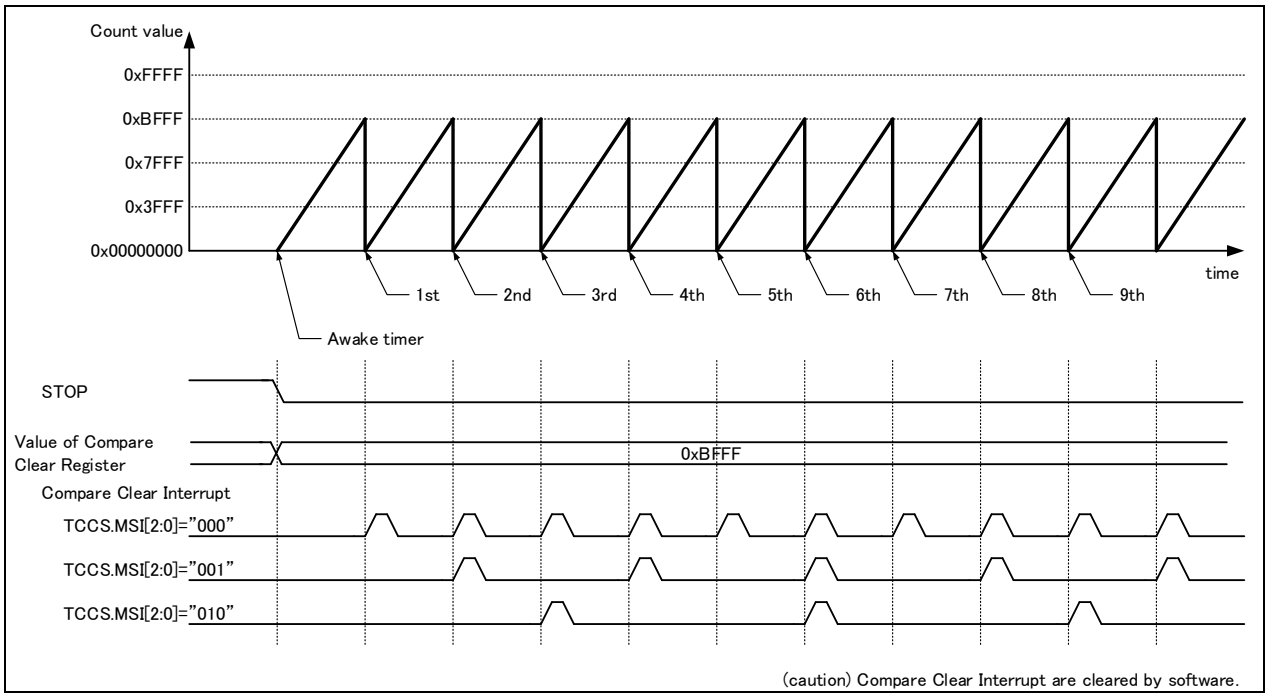
The following describes the case when interrupt is masked.

- The interrupt request flag can be masked by setting the interrupt mask selection bits (MSI2 to GT_MSI0: bit12 to bit10) of the Global Timer State Control Register (GT_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). The interrupt request flag is not masked when the mask count value (MSI2 to MSI0: bit12 to bit10) becomes "000B".

Note:

- *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 Global Timer stopped.*

Figure 5-4 **Compare Clear Interrupts to Be Masked in Up Count**



5.1. Interrupts of the 16-bit Global Timer

The 16-bit Global Timer has one type of interrupt. : compare interrupt

Global Timer interrupt

Table 2-1 shows the interrupt control bits and the interrupt factor of the Global Timer.

Table 5-1 Interrupt Control Bits and Interrupt Factor of Global Timer

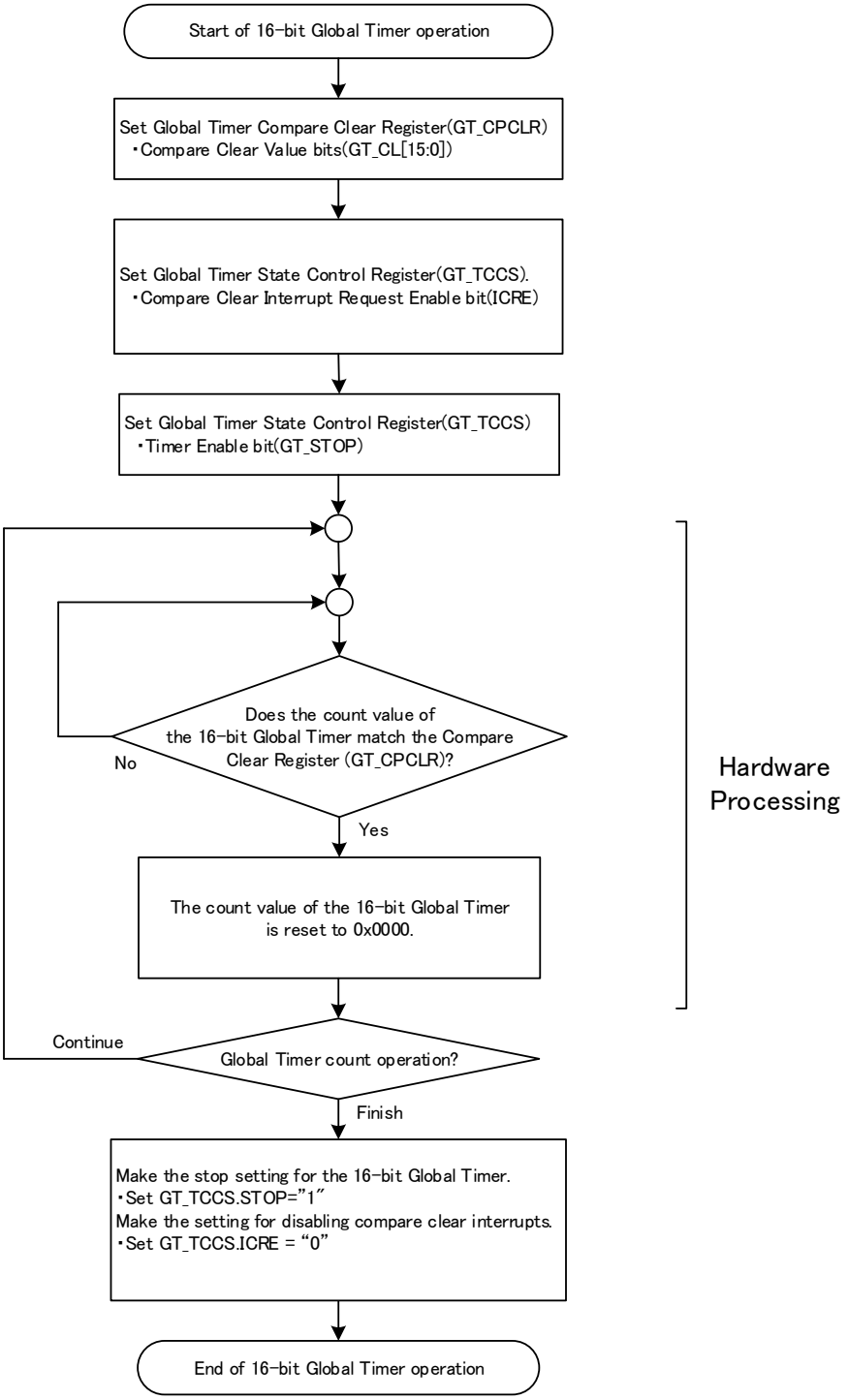
	Global Timer
	Compare Clear
Interrupt request flag bit	Compare clear interrupt flag bit (ICLR:bit9) of the Global Timer State Control Register (GT_TCCS)
Interrupt request enable bit	Compare clear interrupt request enable bit (ICRE:bit8) of the Global Timer State Control Register (GT_TCCS)
Interrupt Factor	The value of the Global Timer matches the value of the Global Timer Compare Clear Register (GT_CPCLR).

"1" is set to the compare clear interrupt flag (ICLR: bit9) of the Global Timer State Control Register (GT_TCCS) when the value of the Global Timer matches the value of the Global Timer Compare Clear Register (GT_CPCLR). When the interrupt request is set to enabled (ICRE: bit8=1 of the Global Timer State Control Register (GT_TCCS)) in this state, then the interrupt request signal (MIRQ) becomes "H".

5.2. Setting Procedure Example of the 16-bit Global Timer

This section provides a setting procedure example of the Global Timer.

Figure 5-5 Setting Procedure Example of Global Timer Operation



5.3. DEBUG MODE

If it goes into a debug mode, operation of the 16-bit Global Timer can be stopped and debugged.

If all the following conditions are satisfied, it will go into the debug mode.

- The Debug Enable bit (DBGEN) in the Global Timer Debug Register (GT_DEBUG) is "1."
- The debug input (Debug input) is active.

It becomes the following operation when it goes into the debug mode.

- The prescaler stop. The state of prescaler is kept.
- The 16-bit Global Timer stop. Timer value and timer output value are kept.

It becomes the following operation when it restores from the debug mode.

- The prescaler is resumed from the stopped value.
- The 16-bit Global Timer is resumed from the stopped value.

Note:

- *Register access is possible in the debug mode.*

6. Registers of the 16-bit Global Timer

This section provides the register list of the 16-bit Global Timer.

Table 6-1 List of Register Mirror Area

Address of Register Mirror Area	Mirrored Peripheral	Mirror Area Behavior	PPU
B484_9040 to B484_907C	Global Timer	Accessing this region results to the access to Global Timer register area.(B484_9000 to B484903C)	This area is covered by PPU #300 as well as the mirrored peripheral.
B484_9080 to B484_90BC			
B484_90C0 to B484_90FC			

Registers of the 16-bit Global Timer

Table 6-2 Register List of 16-bit Global Timer

Abbreviated Register Name	Register Name	Reference
GT_CPCLR	Global Timer Compare Clear Register	6.1
GT_TCDT	Global Timer Data Register	6.2
GT_TCCS	Global Timer State Control Register	6.3
GT_TCCSC	Global Timer State Control Clear Register	6.4
GT_TCCSS	Global Timer State Control Set Register	6.5
GT_DEBUG	Global Timer Debug Register	6.6
GT_MSCTR	Global Timer Match Starting Control Register	6.7
GT_MSCHB	Global Timer Match Starting Channel Buffer Register	6.8
GT_MSCH	Global Timer Match Starting Channel Register	6.9

Register bit locations of the 16-bit Global Timer

Table 6-3 Register Bit Locations of the 16-bit Global 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
GT_CPCLR	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
	CL15	CL14	CL13	CL12	CL11	CL10	CL09	CL08
	CL07	CL06	CL05	CL04	CL03	CL02	CL01	CL00
GT_TCDT	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
	T15	T14	T13	T12	T11	T10	T09	T08
	T07	T06	T05	T04	T03	T02	T01	T00
GT_TCCS	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
	Reserved	Reserved	Reserved	MSI2	MSI1	MSI0	ICLR	ICRE
	Reserved	STOP	Reserved	Reserved	CLK3	CLK2	CLK1	CLK0
GT_TCCSC	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	ICLRC	ICREC
	Reserved	STOPC	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
GT_TCCSS	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	ICRES
	Reserved	STOPS	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
GT_DEBUG	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	DBGEN
GT_MSCTR	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
	Reserved	Reserved	Reserved	Reserved	Reserved	TREN	MODE	GTMOE
GT_MSCHB	Reserved	Reserved	GTCH29	GTCH28	GTCH27	GTCH26	GTCH25	GTCH24
	GTCH23	GTCH22	GTCH21	GTCH20	GTCH19	GTCH18	GTCH17	GTCH16
	GTCH15	GTCH14	GTCH13	GTCH12	GTCH11	GTCH10	GTCH9	GTCH8
	GTCH7	GTCH6	GTCH5	GTCH4	GTCH3	GTCH2	GTCH1	GTCH0
GT_MSCH	Reserved	Reserved	GTCHR29	GTCHR28	GTCHR27	GTCHR26	GTCHR25	GTCHR24
	GTCHR23	GTCHR22	GTCHR21	GTCHR20	GTCHR19	GTCHR18	GTCHR17	GTCHR16
	GTCHR15	GTCHR14	GTCHR13	GTCHR12	GTCHR11	GTCHR10	GTCHR9	GTCHR8
	GTCHR7	GTCHR6	GTCHR5	GTCHR4	GTCHR3	GTCHR2	GTCHR1	GTCHR0

6.1. Global Timer Compare Clear Register (GT_CPCLR)

Compare clear register (GT_CPCLR)

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	CL15	CL14	CL13	CL12	CL11	CL10	CL09	CL08
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	CL07	CL06	CL05	CL04	CL03	CL02	CL01	CL00
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

[bit31:16] Reserved bits

"1" is always read from this bit.

Writing data to these bits has no effect on the operation.

[bit15:0] CL15 to CL00: Compare clear value bits

The compare clear register (GT_CPCLR) is used to compare the count value of the Global Timer.

The count value of the Global Timer is reset to "0000H" when the value of this register matches the count value of the Global Timer.

Notes:

- *Do not set “0x0000” in the Global Timer Compare Clear Register (GT_CPCLR).*
- *For access to this register, use half word access instructions.*

6.2. Global Timer Data Register (GT_TCDT)

The timer data register (GT_TCDT) reads the count value of the Global Timer. This register can also be used to set the count value of the Global 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	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

[bit31:16] Reserved bits

"1" is always read from this bit.

Writing data to these bits has no effect on the operation.

[bit15:0] T15 to T00: Global Timer data Value Bits

The timer data register (GT_TCDT) is used to read the count value of the Global Timer.

The count value can be set by writing a value to this register. However, a value needs to be written while the Global Timer is stopped (STOP: bit 6=1 of the timer state control register (GT_TCCS)).

The counter is cleared to the count value "0x0000" as soon as any of the following factors occurs.

- Hardware reset
- The value of the compare clear register (GT_CPCLR) matches.

Note:

- *For access to this register, use half word access instructions.*

6.3. Global Timer State Control Register (GT_TCCS)

The timer state control register (GT_TCCS) is a register that is used to control the operation of the Global Timer.

For details on writing to this register, see "7.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	Reserved			MSI2	MSI1	MSI0	ICLR	ICRE
R/W Attribute	R0/WX			R,W	R,W	R,W	R	R/W
Protection Attribute	-			-	-	-	-	-
Initial Value	000			0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	Reserved	STOP	Reserved		CLK3	CLK2	CLK1	CLK0
R/W Attribute	R0/WX	R/W	R0/WX		R/W	R/W	R/W	R/W
Protection Attribute	-	-	-		-	-	-	-
Initial Value	0	1	00		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.

[bit15:13] Reserved Bits

"0" is always read from this bit.

Writing data to these bits has no effect on the operation.

[bit12:10] MSI2 to MSI0: Interrupt Mask Selection Bits

- These bits are used to set the number of times to mask the compare clear interrupt flag.
- When these bits are set to "000B", the interrupt flags are not masked.

Bit			Description
MSI2	MSI1	MSI0	
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 Global Timer stopped.*

[bit9] ICLR: Compare Clear Interrupt Flag Bit

This bit is set to "1" when the value of the Global Timer Compare Clear Registers (GT_CPCLR) and the value of the Global 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 Global Timer State Control Clear Register (GT_TCCSC).

Bit	Description
0	No compare clear match
1	Compare clear match.

Notes:

- *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.*

[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 Global Timer State Control Clear Register (GT_TCCSC).
This bit is set to "1" by writing "1" to the ICRES bit of the Global Timer State Control Set Register (GT_TCCSS).

Bit	Description
0	Disable interrupt requests.
1	Enable interrupt requests.

[bit7] Reserved Bits

"0" is always read from this bit.
Writing data to these bits has no effect on the operation.

[bit6] STOP: Timer Enable Bit

This bit is used to stop/start counting of the Global Timer.
When this bit is set to "0".
The Global Timer starts counting.
When this bit is set to "1".
The Global Timer stops counting.

This bit is cleared to "0" by writing "1" to the STOPC bit of the Global Timer State Control Clear Register (GT_TCCSC).
This bit is set to "1" by writing "1" to the STOPS bit of the Global Timer State Control Set Register (GT_TCCSS).

Bit	Description
0	Enable counting (start counting).
1	Disable counting (stop counting).

Notes:

- *After counting of the Global Timer is stopped , the following initialization setting is necessary.*
  1. Initializes of the Global Timer count value.

     It writes "0000_0000h" to the Global Timer Data Register(GT_TCDT)[15:0].
  2. Disable of an ETEN(external timer compare enabling signal) signal.

     It sets Global Timer Match Output Enable bit(GTMODE) in Global Timer Match Starting Control Register(GT_MSCTR) to “0”

[bit5:4] Reserved Bits

"0" is always read from this bit..

Writing data to these bits has no effect on the operation.

[bit3:0] CLK3 to CLK0: Clock Frequency Selection Bits

These bits are used to select the count clock frequency of the Global Timer.

The clock frequency is changed immediately upon setting these bits.

Bit				Description		
CLK3	CLK2	CLK1	CLK0	Count Clock	ϕ = 36 MHz	ϕ = 24 MHz
0	0	0	0	Φ	27.8 ns	41.7 ns
0	0	0	1	Φ/2	55.6 ns	83.3 ns
0	0	1	0	Φ/4	111.1 ns	166.7 ns
0	0	1	1	Φ/8	222.2 ns	333.3 ns
0	1	0	0	Φ/16	444.4 ns	666.7 ns
0	1	0	1	Φ/32	888.9 ns	1.33 us
0	1	1	0	Φ/64	1.8 us	2.67 us
0	1	1	1	Φ/128	3.6 us	5.33 us
1	0	0	0	Φ/256	7.1 us	10.7 us
1	0	0	1	Φ/512	14.2 us	21.4 us
1	0	1	0	Φ/1024	28.4 us	42.8 us
1	0	1	1	Φ/2048	56.8 us	85.6 us
Other settings are prohibited.				-	-	-

ϕ : Peripheral clock: CLK_LCP0A

6.4. Global Timer State Control Clear Register (GT_TCCSC)

The Global Timer State Control Clear Register (GT_TCCSC) is a register that is used to clear bits of the Global Timer State Control Register (GT_TCCS).

For details on writing to this register, see "7.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	Reserved						ICLRC	ICREC
R/W Attribute	R0,WX						R0,W	R0,W
Protection Attribute	-						-	-
Initial Value	000000						0	0

bit	7	6	5	4	3	2	1	0
Field	Reserved	STOPC	Reserved					
R/W Attribute	R0,WX	R0,W	R0,WX					
Protection Attribute	-	-	-					
Initial Value	0	0	00000					

[bit31:16] Reserved Bits

"1" is always read from this bit.

Writing data to these bits has no effect on the operation.

[bit15:10] Reserved Bits

"0" is always read from this bit.

Writing data to these bits has no effect on the operation.

[bit9] ICLRC: ICLR Clear Bit

When "1" is written to this bit, the compare clear interrupt flag bit (ICLR) of the Global Timer State Control Register (GT_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 Global Timer State Control Register (GT_TCCS).
1	Clear the compare clear interrupt flag bit (ICLR) of the Global Timer State Control Register (GT_TCCS).

[bit8] ICREC: ICRE Clear Bit

When "1" is written to this bit, the compare clear interrupt request enable bit (ICRE) of the Global Timer State Control Register (GT_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 Global Timer State Control Register (GT_TCCS).
1	Clear the compare clear interrupt request enable bit (ICRE) of the Global Timer State Control Register (GT_TCCS).

[bit7] Reserved Bits

"0" is always read from this bit.
Writing data to these bits has no effect on the operation.

[bit6] STOPC: STOP Clear Bit

When "1" is written to this bit, the timer enable bit (STOP) of the Global Timer State Control Register (GT_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 Global Timer State Control Register (GT_TCCS).
1	Clear the timer enable bit (STOP) of the Global Timer State Control Register (GT_TCCS).

[bit5:0] Reserved Bits

"0" is always read from this bit. Always write "0" to this bit.

Writing data to these bits has no effect on the operation.

6.5. Global Timer State Control Set Register (GT_TCCSS)

The Global Timer State Control Set Register (GT_TCCSC) is a register that is used to set bits of the Global Timer State Control Register (GT_TCCS).

For details on writing to this register, see "7.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	Reserved							ICRES
R/W Attribute	R0,WX							R0,W
Protection Attribute	-							-
Initial Value	0000000							0

bit	7	6	5	4	3	2	1	0
Field	Reserved	STOPS	Reserved					
R/W Attribute	R0,WX	R0,W	R0,WX					
Protection Attribute	-	-	-					
Initial Value	0	0	000000					

[bit31:16] Reserved Bits

"1" is always read from this bit.

Writing data to these bits has no effect on the operation.

[bit15:9] Reserved Bits

"0" is always read from this bit.

Writing data to these bits has no effect on the operation.

[bit8] ICRES: ICRE Set Bit

When "1" is written to this bit, the compare clear interrupt request enable bit (ICRE) of the Global Timer State Control Register (GT_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 Global Timer State Control Register (GT_TCCS).
1	Set the compare clear interrupt request enable bit (ICRE) of the Global Timer State Control Register (GT_TCCS).

[bit7] Reserved Bits

"0" is always read from this bit.
Writing data to these bits has no effect on the operation.

[bit6] STOPS: STOP Set Bit

When "1" is written to this bit, the timer enable bit (STOP) of the Global Timer State Control Register (GT_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 Global Timer State Control Register (GT_TCCS).
1	Set the timer enable bit (STOP) of the Global Timer State Control Register (GT_TCCS).

[bit5:0] Reserved Bits

"0" is always read from this bit.
Writing data to these bits has no effect on the operation.

6.6. Global Timer Debug Register (GT_DEBUG)

The Global Timer Debug Register (GT_DEBUG) performs enable/disable setting of debug.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved							
ACCESS_TYPE	R1,WX							
PROT_TYPE	-							
INITIAL_VALUE	11111111							

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved							
ACCESS_TYPE	R1,WX							
PROT_TYPE	-							
INITIAL_VALUE	11111111							

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved							
ACCESS_TYPE	R0,W0							
PROT_TYPE	-							
INITIAL_VALUE	00000000							

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved							DBGEN
ACCESS_TYPE	R0,W0							R/W
PROT_TYPE	-							
INITIAL_VALUE	0000000							0

[bit31:16] Reserved Bits

The read value is "1".

If writing to this bit, write "1".

[bit15:1] Reserved Bits

The read value is "0".

If writing to this bit, write "0".

[bit0] DBGEN: Debug Enable Bit

It is the bit which permits stopping operation of a 16-bit Global Timer with the debug input (Debug input).

Bit	Description
0	Disable
1	Enable

6.7. Global Timer Match Starting Control Register (GT_MSCTR)

This Global Timer Match Start Control Register controls the Global Timer Match Start Enable Signal which it inputs into the Base 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							
R/W Attribute	R1,WX							
Protection Attribute	-							
Initial Value	11111111							

bit	7	6	5	4	3	2	1	0
Field	Reserved					TREN	MODE	GTMOE
R/W Attribute	R0,WX					R/W	R/W	R/W
Protection Attribute	-							
Initial Value	00000					0	0	0

[bit31:8] Reserved Bits

"1" is always read from this bit.

Writing data to these bits has no effect on the operation.

[bit7:3] Reserved Bits

"0" is always read from this bit.

Writing data to these bits has no effect on the operation.

[bit2] TREN: Transfer Request Enable Bit

It uses this bit in order to enable a transmission request of the Global Timer Match Starting Channel Buffer Register (GT_MSCHB) value.

When this bit is "0", even if the count value of the Global Timer is match with a comparison clear register (GT_CPCLR), the Global Timer Match Starting Channel Buffer Register (GT_MSCHB) value is not transmitted.

When this bit is "1", even if the count value of the Global Timer is match with a comparison clear register (GT_CPCLR), the Global Timer Match Starting Channel Buffer Register (GT_MSCHB) value is transmitted to the Global Timer Match Starting Channel Register (GT_MSCH).

Bit	Description
0	Disable the transfer request of Register data
1	Enable the transfer request of Register data

Note:

- *When changing the Global Timer Match Starting Channel Buffer Register (GT_MSCHB) set, it sets this bit to "0."*

[bit1] MODE: Mode Select Bit

It uses this bit in order to enable the Global Timer Match Starting Channel Buffer Register (GT_MSCHB). The Global Timer match starting enabling signal of each Base Timer channel is controlled by this bit setup as follows.

- When the is MODE ="0"
  - The Global Timer Match Starting Channel Buffer Register (GT_MSCHB) is invalid.
  - After writing "1" in GTMOE bit, it becomes external timer comparison enabling signal to the Base Timer channel set as 1 (ETEN)" H" by a GT_MSCH register at once.
- When the is MODE ="1"
  - The Global Timer Match Starting Channel Buffer Register (GT_MSCHB) is valid.
  - The data which was written in the Global Timer Match Start Channel Buffer Register (GT_MSCHB) and was held, After writing "1" in TREN bit, if the Global Timer counting value and the Global Time Compare Clear Register (GT_CPCLR) matches, it is transmitted to the Global Timer Match Starting Channel Register (GT_MSCH), the External Timer Compare Enable Signal (ETEN) to the Base Timer Channel set as “1” in the Global Timer Match Start Channel Register (GT_MSCH) becomes "H".

Bit	Description
0	Buffer Disable
1	Buffer Enable

[bit0] GTMOE: Global Timer Match Output Enable Bit

It uses this as an output Enable of the External Timer Compare Enable Signal (ETEN) of Base Timers.

When this bit is "0", the External Timer Compare Enable Signal (ETEN) to the channel of all Base Timers is set to "L.".

- It outputs "H" as an external timer compare enabling signal (ETEN) at the time of GTCH="1."
- It outputs "L" as an external timer compare enabling signal (ETEN) at the time of GTCH="0.".

Bit	Description
0	Disable the output of all External Timer Compare Enable Signals.
1	Enable the output of all External Timer Compare Enable Signals.

6.8. Global Timer Match Starting Channel Buffer Register (GT_MSCHB)

The Global Timer Match Startup Channel Buffer Register (GT_MSCHB) is a buffer register of the Global Timer Match Startup Channel Register (GT_MSCH)

There is the undermentioned two timing by which this register value is transmitted to the Global Timer Match Startup Channel Register (GT_MSCH).

- The Global Timer is stopping (STOP ="1" of a Global Timer Status Control Register (GT_TCCS)), or it is at the disabled (MODE ="0" of Global Timer Match Starting Control Register (GT_MSCTR)) time about the STOP = "1" buffer function of the Global Timer Status Control Register (GT_TCCS), the value of a Global Timer Match Start Channel Buffer Register (GT_MSCHB) is transmitted to a Global Timer Match Start Channel Register (GT_MSCH) at once.
- The buffer function is at the enable (MODE ="1" of Global Timer Match Starting Control Register (GT_MSCTR)) time, it sets a transmission request of a register value as enable (TREN bit ="1" of Global Timer Match Starting Control Register (GT_MSCTR)), and when the count value of the Global Timer matches the value of the Global Timer Compare Clear Register(GT_CPCLR), the value of a Global Timer Match Start Channel Buffer Register (GT_MSCHB) is transmitted to the Global Timer Match Start Channel Register (GT_MSCH).

Global Timer Match Starting Channel Buffer registers (GT_MSCHB)

bit	31	30	29	28	27	26	25	24
Field	Reserved		GTCH29	GTCH28	GTCH27	GTCH26	GTCH25	GTCH24
R/W Attribute	R0/WX		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	GTCH23	GTCH22	GTCH21	GTCH20	GTCH19	GTCH18	GTCH17	GTCH16
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	GTCH15	GTCH14	GTCH13	GTCH12	GTCH11	GTCH10	GTCH9	GTCH8
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	GTCH7	GTCH6	GTCH5	GTCH4	GTCH3	GTCH2	GTCH1	GTCH0
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

"0" is always read from this bit.

Writing data to these bits has no effect on the operation.

[bit29:0] GTCH29～0 : Global Timer Match Starting Channel Buffer Bits

Bit	Description
0	Global Timer Match Starting Disabled
1	Global Timer Match Starting Enabled

Notes:

- *At the time of the Global Timer stop, it is after the Global Timer Match Start Channel Buffer Register (GT_MSCHB) set, please permit the External Timer Compare Enabling-Signal Output (GTMOE ="1" of Global Timer Match Starting Control Register (GT_MSCTR)).*
- *At the case of TREN = "1" (transmission request Clearance) of Global Timer Match Starting Control register (GT_MSCTR), please do not change the Global Timer Match Start Channel Buffer-Register (GT_MSCHB) set.*
- *For access to this register, use word access instructions.*

6.9. Global Timer Match Starting Channel Register (GT_MSCH)

This register displays the value transmitted from the Global Timer Match Starting Channel Buffer Register (GT_MSCHB) register.

bit	31	30	29	28	27	26	25	24
Field	Reserved		GTCHR29	GTCHR28	GTCHR27	GTCHR26	GTCHR25	GTCHR24
R/W Attribute	R0,WX		R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-		-	-	-	-	-	-
Initial Value	00		0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	GTCHR23	GTCHR22	GTCHR21	GTCHR20	GTCHR19	GTCHR18	GTCHR17	GTCHR16
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	GTCHR15	GTCHR14	GTCHR13	GTCHR12	GTCHR11	GTCHR10	GTCHR9	GTCHR8
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	GTCHR7	GTCHR6	GTCHR5	GTCHR4	GTCHR3	GTCHR2	GTCHR1	GTCHR0
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

[bit31:30] Reserved Bits
"0" is always read from this bit.

[bit29:0] GTCHR29～0 : Global Timer Match Starting Channel Bits

This bit becomes external timer compare enabling-signal to the Base Timer channel which the Global Timer match start is permitted and corresponds when it is 1" (ETEN)" H".

Bit	Description
0	Global Timer Match Starting Disabled
1	Global Timer Match Starting Enabled

Note:

- *For access to this register, use word access instructions.*

7. Precautions for Using This Device

The following shows the notes when using the 16-bit Global Timer.

Notes to observe when accessing a register

When accessing the compare clear register (GT_CPCLR)

Use half word access instructions to the Global Timer Compare Clear Register (GT_CPCLR)

When accessing the timer state control registers (GT_TCCS)

- This register supports writing from the bit-band alias area. For the bit-band alias area, see "CHAPTER: 55 Bit-Band Unit."
- To clear a specified bit of this register, clear the bit by writing "1" to the applicable bit of the Global Timer State Control Clear Register (GT_TCCSC).
- To set a specified bit of this register, set the bit by writing "1" to the applicable bit of the Global Timer State Control Set Register (GT_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 MSI2 to MSI0.
- 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 Global Timer stopped.

When accessing the Global Timer Match Starting Channel Buffer Register (GT_MSCHB)

Use word access instructions to the Global Timer Match Starting Channel Buffer Register (GT_MSCHB)

When accessing the Global Timer Match Starting Channel Register (GT_MSCH)

Use word access instructions to the Global Timer Match Starting Channel Register (GT_MSCH)

Notes when operating the Global Timer

When setting using a program

- The compare clear flag is not set if the timer starts counting when the value of the Global Timer Compare Clear Register (GT_CPCLR) matches the count value.
- Set any values other than 0 to the Global Timer Compare Clear Register (GT_CPCLR). Note that the following operations occur if 0 is set.
  - When the count value is updated to 0 and fixed to 0. Then, the compare clear flag are set at every count clock.

CHAPTER 41: 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. **Block Diagram of the 32-bit Free-run Timer**
3. **Operation of the 32-bit Free-run Timer**
4. **Registers of the 32-bit Free-run Timer**
5. **Precautions for Using This Device**

FRT-T00P00R02L00-E1-GBB00

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.

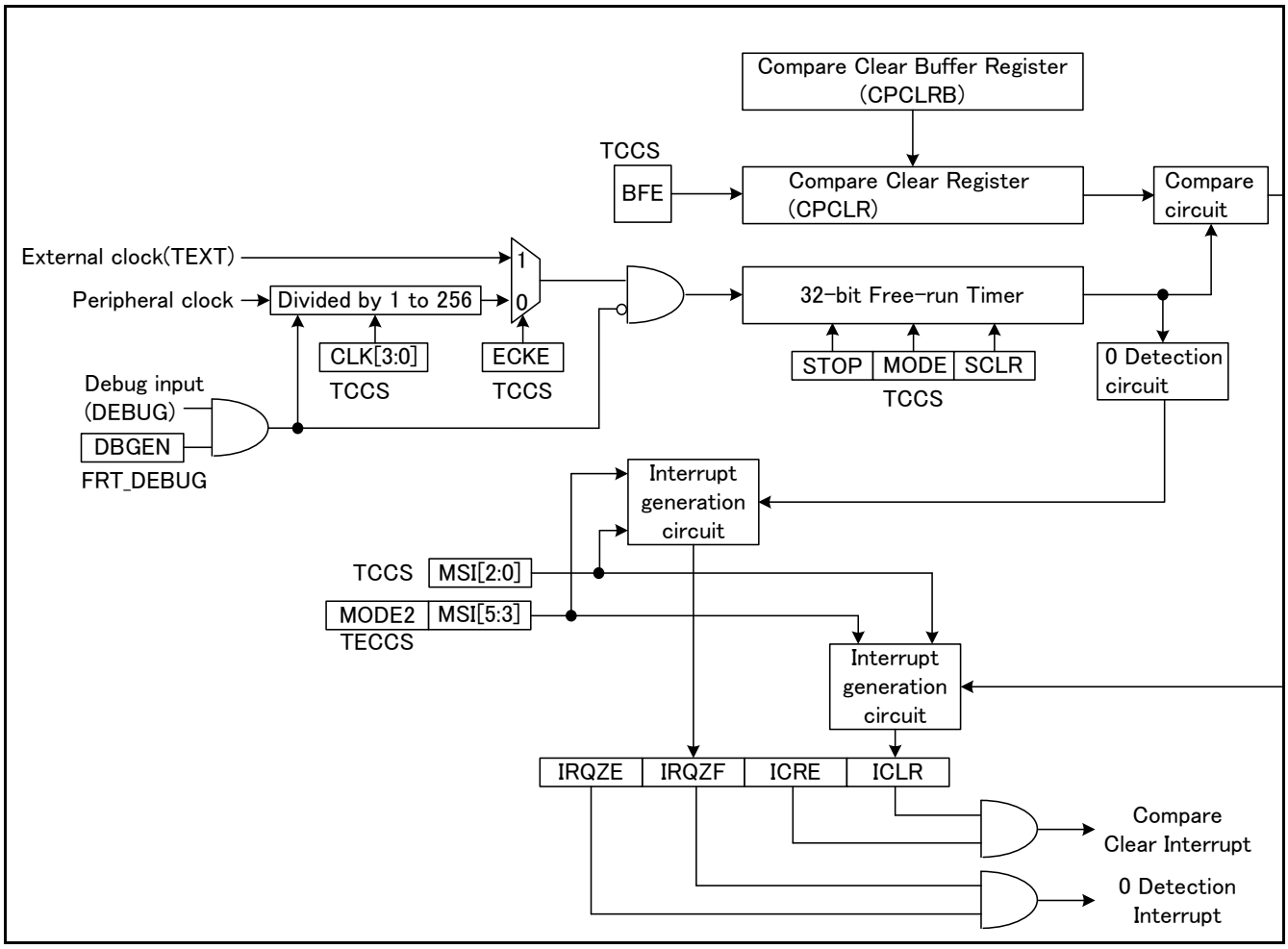
Functions of the 32-Bit Free-Run Timer

- The 32-bit Free-run Timer is composed of the 32-bit up/down counter, Timer State Control Register, 32-bit Compare Clear Register, 32-bit Compare Clear Buffer Register, and Prescaler.
- The 9 types of count 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, 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 Free-run Timer can be used as a clock count of the 32-bit Output Compare and the 32-bit Input Capture.

2. Block Diagram of the 32-Bit Free-Run Timer

This section shows a block diagram of 32-bit Free-run Timer

Figure 2-1 Block Diagram of 32-Bit Free-Run Timer



3. Operation of the 32-Bit Free-Run Timer

This section describes 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 (the Timer Enable bit(STOP) in the Timer State Control Register (TCCS)). When the 32-bit Output Compare and the 32-bit Input Capture are connected to the 32-bit 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.

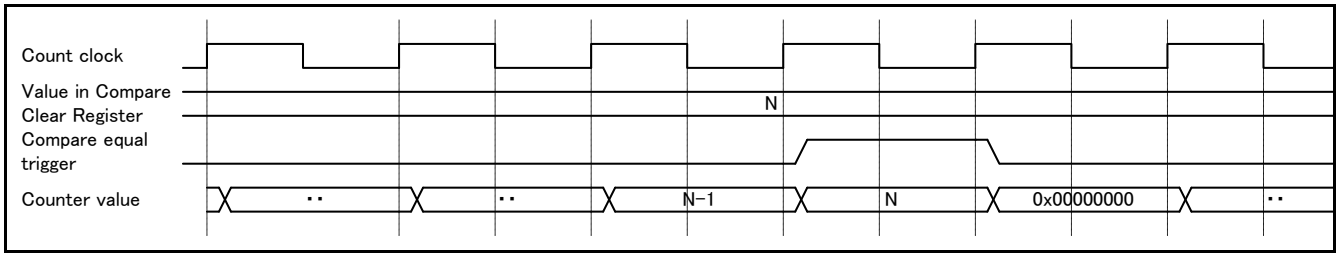
Counter Clear

The count value of the 32-bit Free-run Timer is cleared to 0x00000000 when any of the following conditions is met.

- When the count value matches the value in the Compare Clear Register (CPCLR) in the up count mode (Timer Count Mode bit(MODE) in the Timer State Control Register (TCCS) ="0")
- When "1" is written to the Timer Clear bit (SCLR) in the Timer State Control Register (TCCS) while the 32-bit Free-run Timer is operating (Timer Enable bit(STOP) in the Timer State Control Register (TCCS) ="0")
- When 0x00000000 is written to the Timer Data Register (TCDT) while the 32-bit Free-run Timer is stopped (the Timer Enable bit(STOP) in the Timer State Control Register (TCCS) ="1")
- When the hardware is reset. Upon reset, the counter is cleared immediately.

When "1" is written to the Timer Clear bit (SCLR) in the Timer State Control Register (TCCS) or when the count value matches the value in the Compare Clear Register (CPCLR), 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 32-bit Free-run Timer is not cleared if "1" is written to the Timer Clear bit (SCLR) in the Timer State Control Register (TCCS) while the timer is stopped.*

Timer Mode

Either of the following modes can be selected for the 32-bit Free-run Timer.

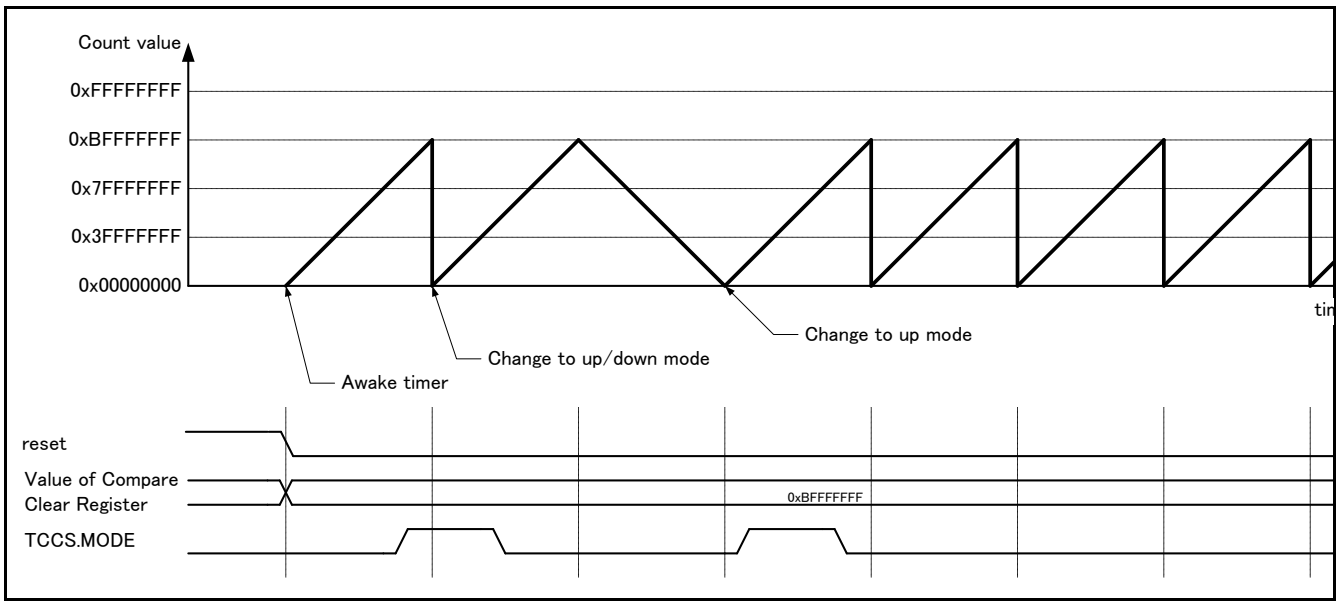
- Up count mode (the Timer Count Mode bit(MODE) in the Timer State Control Register (TCCS) ="0")
- Up/down count mode (the Timer Count Mode bit(MODE) in the Timer State Control Register (TCCS) ="1")

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 in 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 in 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) in 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 (the Compare Clear Buffer Enable bit(BFE) in the Timer State Control Register (TCCS) ="1"), the data written to the Compare Clear Buffer Register (CPCLRB) is transferred to the Compare Clear Register (CPCLR) upon detecting the count value 0x00000000. When the buffer function is disabled (the Compare Clear Buffer Enable bit(BFE) in the Timer State Control Register (TCCS) ="0"), 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 (the Compare Clear Buffer Enable Bit(BFE) of Timer State Control Register (TCCS) ="0")

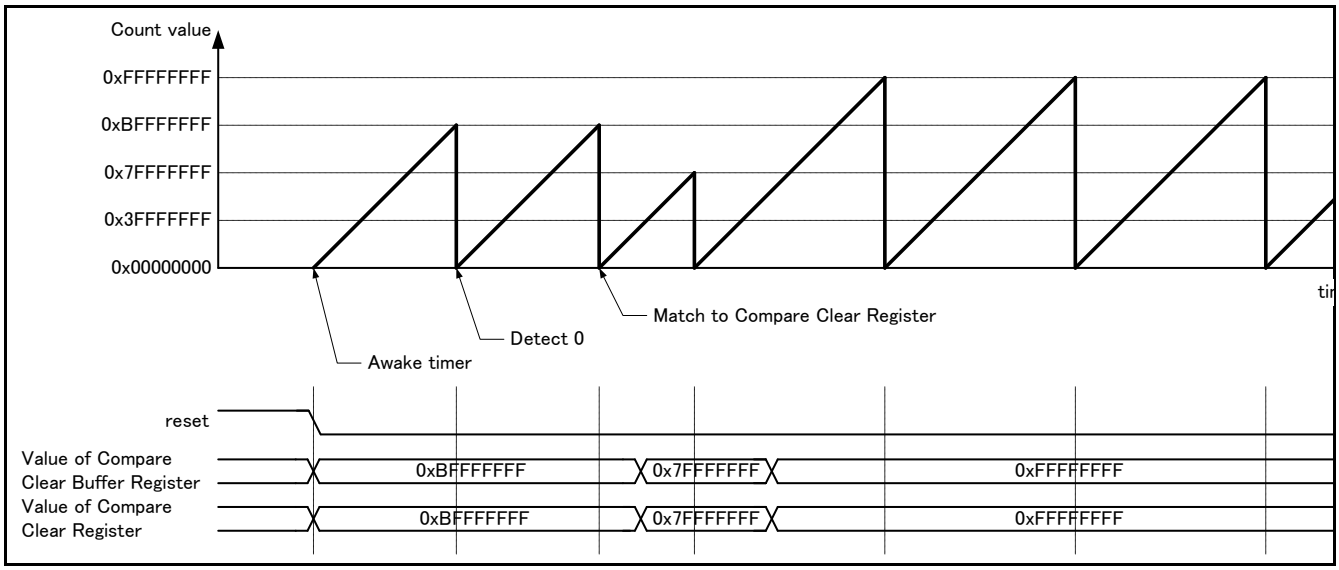


Figure 3-4 Up Count Mode Operation When Compare Clear Buffer Is Enabled (the Compare Clear Buffer Enable Bit(BFE) of Timer State Control Register (TCCS) ="1")

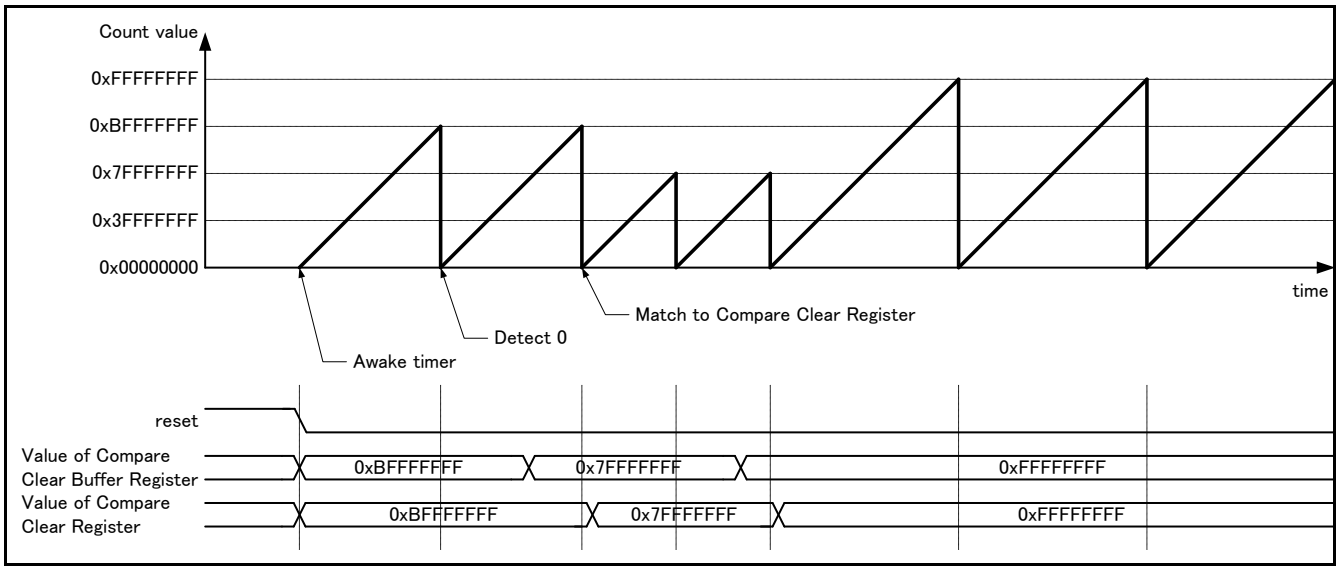
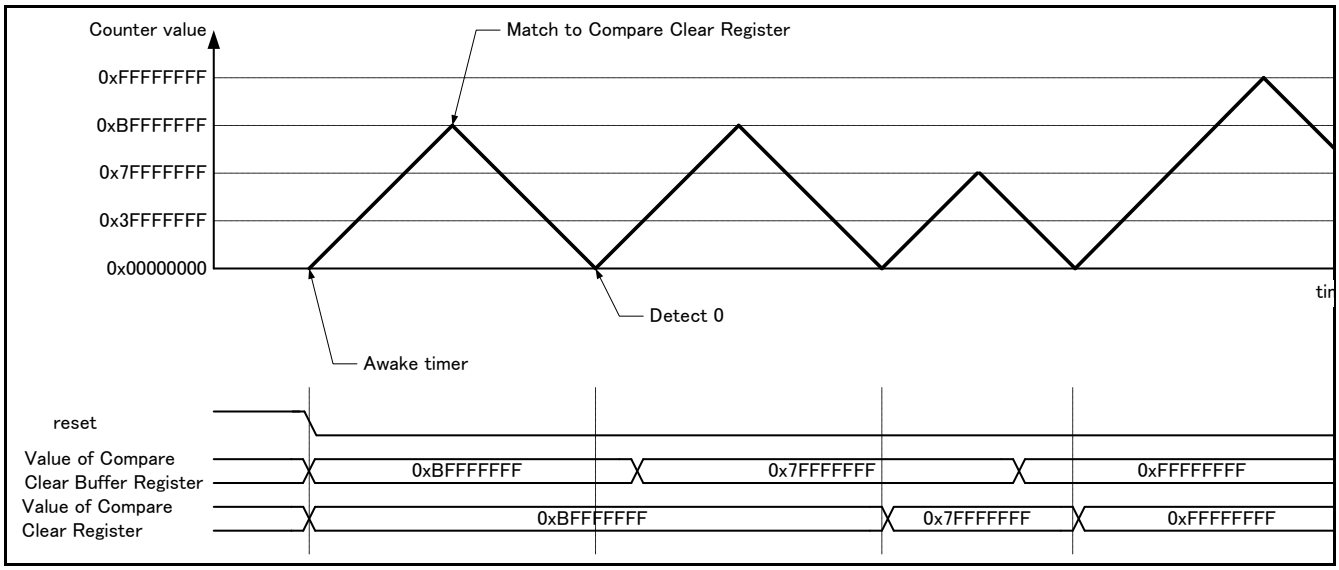


Figure 3-5 Up/Down Count Mode Operation When Compare Clear Buffer Is Enabled (the Compare Clear Buffer Enable Bit(BFE) of Timer State Control Register (TCCS) ="1")



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 in the Compare Clear Register (CPCLR). The 0 detection interrupt is generated when the count value reaches 0x00000000.

For Zero Detection and Compare Clear Detection, detection begins when counting is enabled on the 32-bit Free-run Timer and the count value is updated from the current value to the next value.

Note:

- *The 0 detection interrupt is not generated when the timer is cleared (the Timer Clear bit(SCLR) in the Timer State Control Register (TCCS) ="1").*

Figure 3-6 Interrupt Generated in Up Count Mode (the Timer Count Mode Bit(MODE) in the Timer State Control Register (TCCS) ="0")

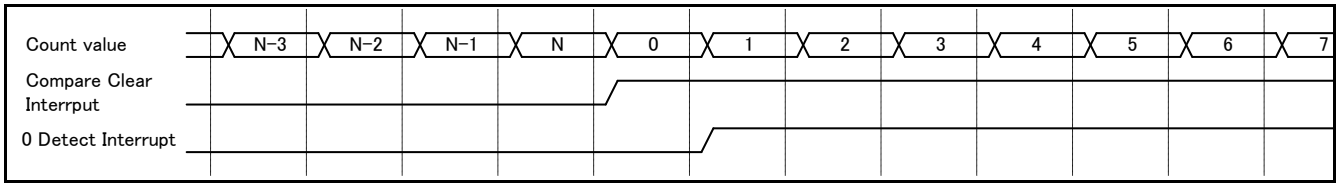
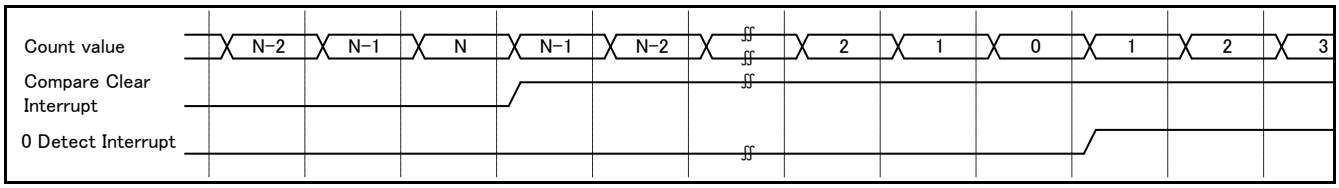


Figure 3-7 Interrupt Generated in Up/Down Count Mode (the Timer Count Mode Bit(MODE) in the Timer State Control Register (TCCS) ="1")



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 (MSI[2:0]) in the Timer State Control Register (TCCS). The Interrupt Mask Selection bits (MSI[2:0]) in the Timer State Control Register (TCCS) are the 3-bit reload down register that reloads the value when the mask count value reaches "000".
  The mask count value can also be loaded by directly writing data to the Interrupt Mask Selection bits (MSI[2:0]) in the Timer State Control Register (TCCS). The number of mask counts is the value set in the Interrupt Mask Selection bits (MSI[2:0]) in the Timer State Control Register (TCCS). An interrupt request flag is not masked when the mask count value (setting in the Interrupt Mask Selection bits(MSI[2:0]) in the Timer State Control Register (TCCS)) becomes "000".
- The mask control of an interrupt request varies depending on the count mode (the Timer Count Mode bit(MODE) in the Timer State Control Register (TCCS)). In the up count mode (Timer Count Mode bit(MODE) in the Timer State Control Register (TCCS) ="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 (the Timer Count Mode bit(MODE) in the Timer State Control Register (TCCS) ="1"), only the 0 Detection Interrupt Flag(IRQZF) in the Timer State Control Register (TCCS) can be masked.

The following describes the case when both interrupt requests are masked.

- Both interrupts can be masked only when the 32-bit Free-run Timer is in the up/down count mode (the Timer Count Mode bit(MODE) in the Timer State Control Register (TCCS) ="1") and when the Timer Extended State Control Register (TECCS) is set to the Interrupt Mask Mode bit 2(MODE2) in the Timer Extended State Control Register (TECCS) ="1" and the Timer State Control Register (TCCS) is set to the Timer Count Mode bit(MODE) in the Timer State Control Register (TCCS) ="1".
- Interrupt Mask Selection bits(MSI[2:0]) in the Timer State Control Register (TCCS) are used to mask the 0 detection interrupt. Compare Clear Interrupt Mask Selection bits(MSI[5:3]) in the Timer Extended State Control Register (TECCS) are used to mask the Compare Clear Interrupt.

Note:

- *The 0 detection interrupt is not generated when the timer is cleared (the Timer Clear bit(SCLR) in the Timer State Control Register (TCCS) ="1").*
- *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

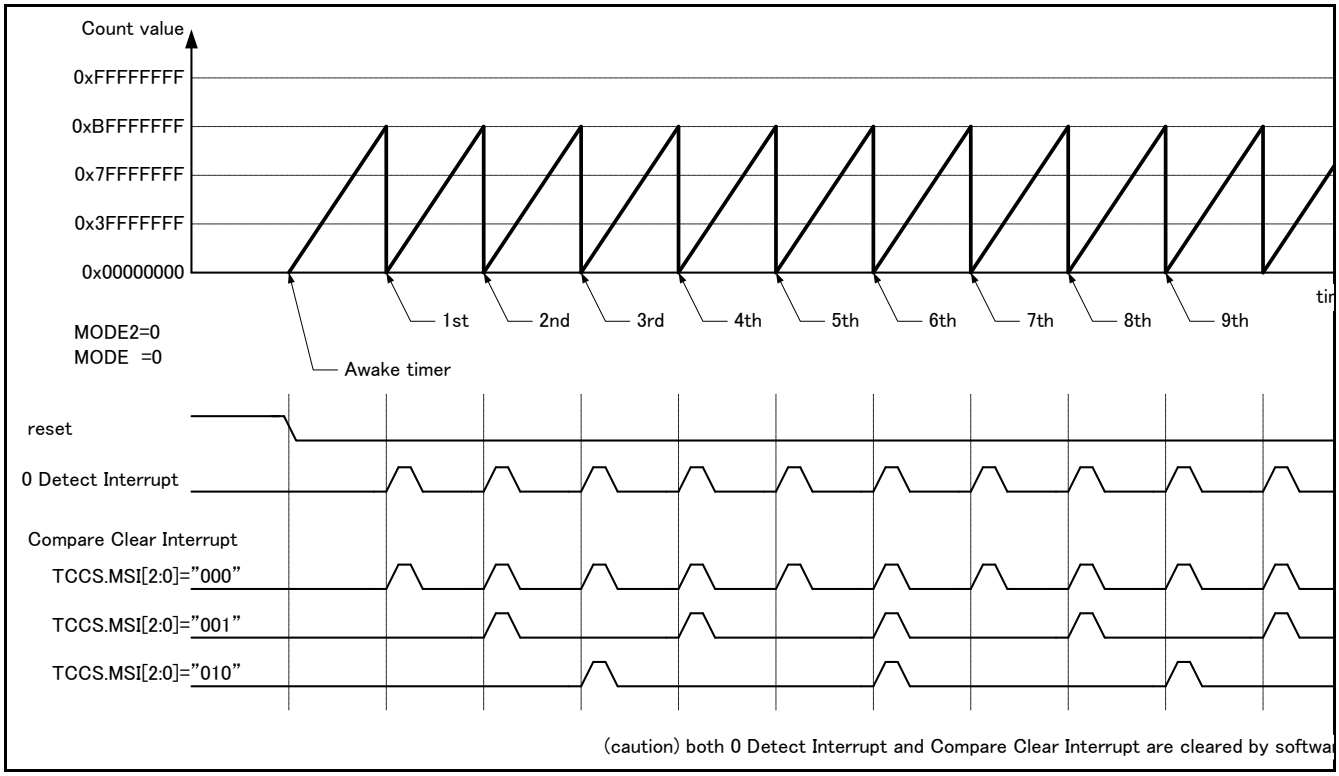


Figure 3-9 0 Detection Interrupts to Be Masked in Up/Down Count Mode

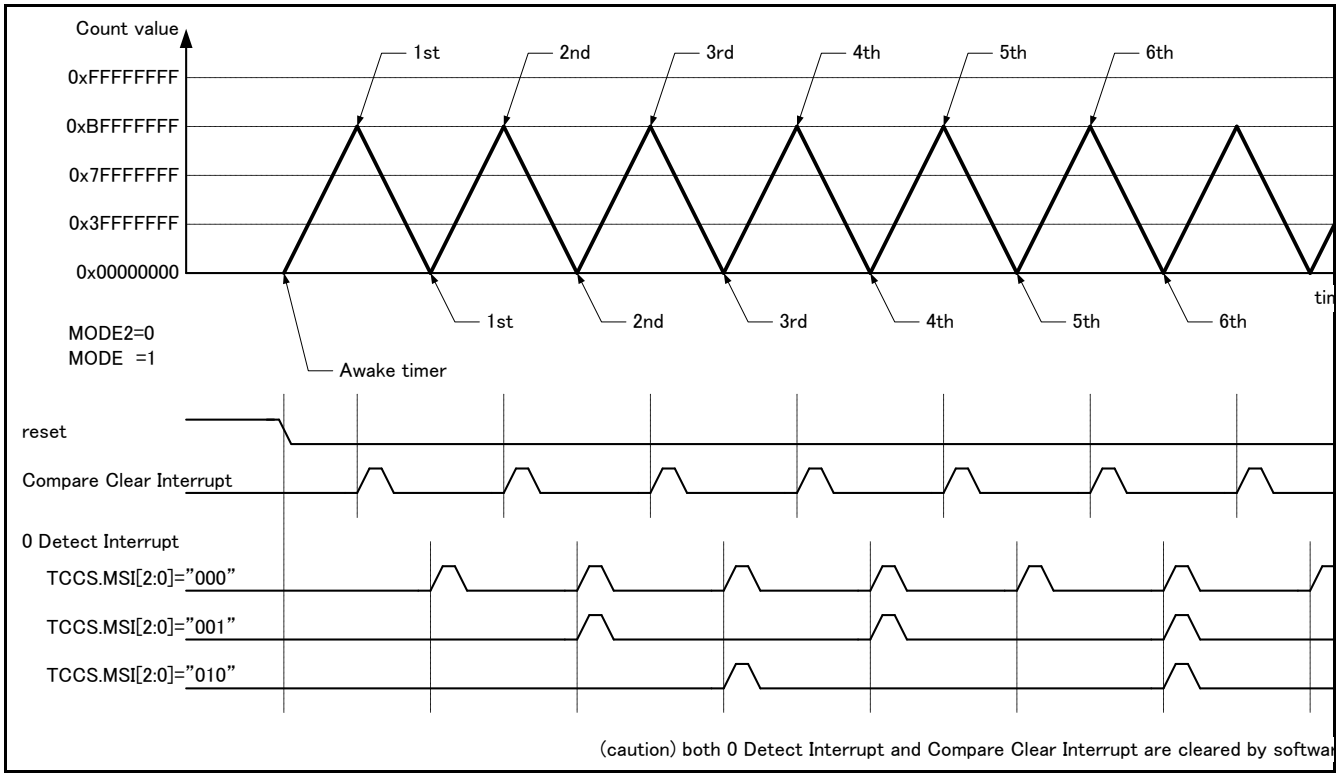
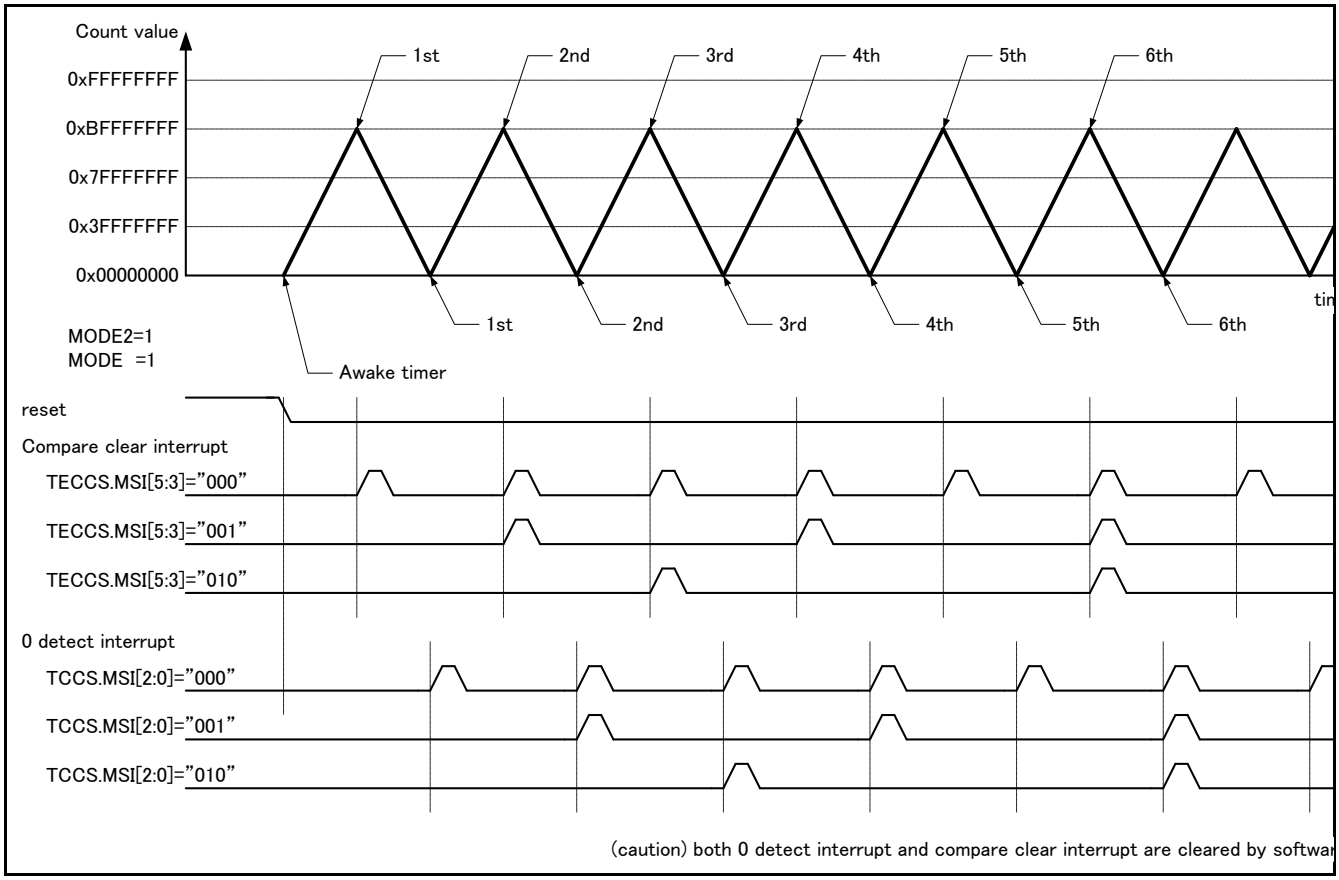


Figure 3-10 0 Detection Interrupts and Compare Clear Interrupts to Be Masked in Up/Down Count Mode

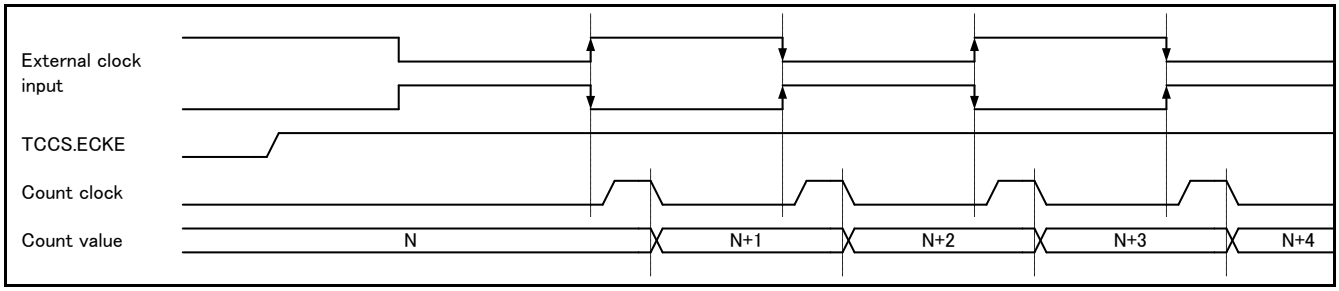


Selected External Count Clock

The 32-bit Free-run Timer counts based on an input clock (peripheral clock or external clock).

If the external clock mode (the Clock Selection bit(ECKE) in the Timer State Control Register (TCCS) ="1") is selected, the 32-bit 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 32-bit Free-run Timer counts at both edges. When the initial value of the external input is "L", the 32-bit Free-run Timer starts counting at a falling edge of the external input. After that, the 32-bit Free-run Timer counts up at both edges.

Figure 3-11 Count Timing of 32-Bit 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 Clear Interrupt and 0 Detection Interrupt.

32-Bit Free-Run Timer Interrupt

Table 3-1shows the interrupt control bits and the interrupt factor of the 32-bit Free-run Timer.

Table 3-1 Interrupt Control Bits and Interrupt Factor of 32-Bit Free-Run Timer

	32-Bit Free-Run Timer	
	Compare Clear	**0 Detection**
Interrupt request flag	Compare Clear Interrupt Flag (ICLR) in the Timer State Control Register (TCCS)	0 Detection Interrupt Flag (IRQZF) in the Timer State Control Register (TCCS)
Interrupt request enable bit	Compare Clear Interrupt request enable bit (ICRE) in the Timer State Control Register (TCCS)	0 detection interrupt request enable bit (IRQZE) in the Timer State Control Register (TCCS)
Interrupt Factor	The value of the 32-bit Free-run Timer matches the value of the Compare Clear Register (CPCLR).	The 32-bit Free-run Timer value becomes 0x00000000.

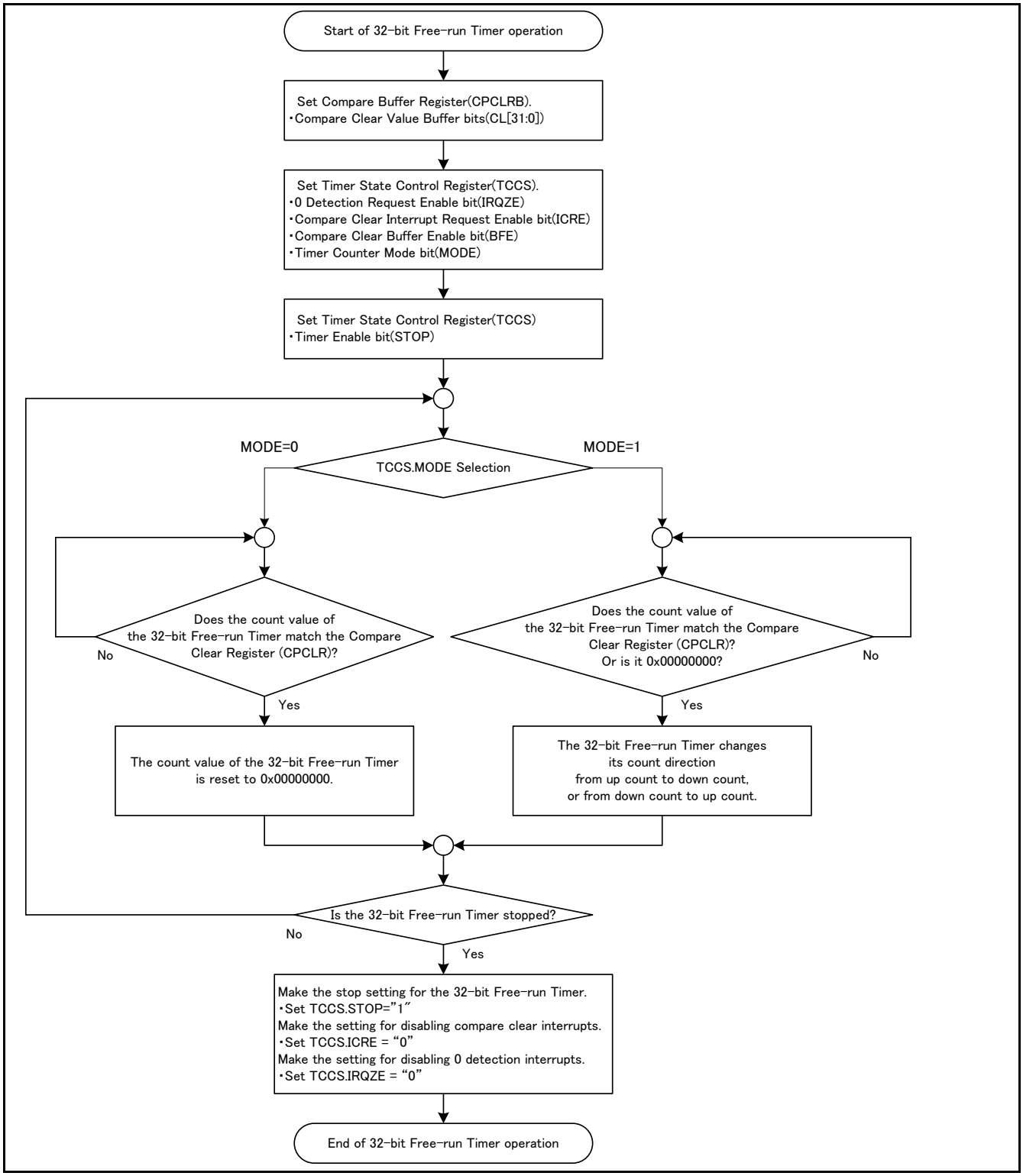
"1" is set to the Compare Clear Interrupt Flag (ICLR) in the Timer State Control Register (TCCS) when the value of the 32-bit Free-run Timer matches the value of the Compare Clear Register (CPCLR). When the interrupt request is set to enabled (the Compare Clear Interrupt Request Enable bit(ICRE) in the Timer State Control Register (TCCS) ="1") in this state, then the interrupt request is generated..

"1" is set to the 0 Detection Interrupt Flag (IRQZF) in the Timer State Control Register (TCCS) when the value of the 32-bit Free-run Timer becomes 0x00000000. When the interrupt request is set to enabled (the 0 Detection Request Enable bit(IRQZE) in the Timer State Control Register (TCCS) ="1") in this state, then the interrupt request is generated..

3.2. Setting Procedure Example of the 32-Bit Free-Run Timer

This section provides a setting procedure example of the 32-bit Free-run Timer.

Figure 3-12 Setting Procedure Example of 32-Bit Free-Run Timer Operation



3.3. Debug Mode

If it goes into a debug mode, operation of the 32-bit Free-run Timer can be stopped and debugged.

If all the following conditions are satisfied, it will go into the debug mode.

- The Debug Enable bit (DBGEN) in the Free-run Timer Debug Register (FRT_DEBUG) is "1."
- The debug input (DEBUG) is active.

It becomes the following operation when it goes into the debug mode.

- The prescaler stop. The state of prescaler is kept.
- The 32-bit Free-run Timer stop. Timer value and timer output value are kept.

It becomes the following operation when it restores from the debug mode.

- The prescaler is resumed from the stopped value.
- The 32-bit Free-run Timer is resumed from the stopped value.

Note:

- *Register access is possible in the debug mode.*

4. Registers of the 32-Bit Free-Run Timer

This section provides the register list of the 32-bit Free-run Timer.

Registers of the 32-Bit Free-Run Timer

Table 4-1 Register List of 32-Bit Free-Run Timer

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
FRT_DEBUG	Free-run Timer Debug Register	4.7

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	CL[31:24]							
R/W Attribute	W							
Protection Attribute	-							
Initial Value	11111111							

bit	23	22	21	20	19	18	17	16
Field	CL[23:16]							
R/W Attribute	W							
Protection Attribute	-							
Initial Value	11111111							

bit	15	14	13	12	11	10	9	8
Field	CL[15:8]							
R/W Attribute	W							
Protection Attribute	-							
Initial Value	11111111							

bit	7	6	5	4	3	2	1	0
Field	CL[7:0]							
R/W Attribute	W							
Protection Attribute	-							
Initial Value	11111111							

[bit31:0] CL[31:0]: 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 (the Compare Clear Buffer Enable bit(BFE) in the Timer State Control Register (TCCS) ="0") or the 32-bit 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 (the Compare Clear Buffer Enable bit(BFE) in the Timer State Control Register (TCCS) ="1") and 0 is detected as the count value of the 32-bit Free-run Timer.

Notes:

- *Do not set 0x00000000 in the Compare Clear Buffer Register (CPCLRB).*
- *For access to this register, use 32-bit access instructions.*

Compare Clear Register (CPCLR)

bit	31	30	29	28	27	26	25	24
Field	CL[31:24]							
R/W Attribute	R							
Protection Attribute	-							
Initial Value	11111111							

bit	23	22	21	20	19	18	17	16
Field	CL[23:16]							
R/W Attribute	R							
Protection Attribute	-							
Initial Value	11111111							

bit	15	14	13	12	11	10	9	8
Field	CL[15:8]							
R/W Attribute	R							
Protection Attribute	-							
Initial Value	11111111							

bit	7	6	5	4	3	2	1	0
Field	CL[7:0]							
R/W Attribute	R							
Protection Attribute	-							
Initial Value	11111111							

[bit31:0] CL[31:0]: Compare Clear Value Bits

The Compare Clear Register (CPCLR) is used to compare the count value of the 32-bit Free-run Timer.

In the up count mode (the Timer Count Mode bit(MODE) in the Timer State Control Register (TCCS) ="0"), the count value of the 32-bit Free-run Timer is reset to 0x00000000 when the value of this register matches the count value of the 32-bit Free-run Timer.

In the up/down count mode (the Timer Count Mode bit(MODE) in the Timer State Control Register (TCCS) ="1"), the 32-bit 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 32-bit Free-run Timer. The counting direction is changed from down count to up count when 0 is detected.

Notes:

- *For access to this register, use 32-bit access instructions.*

4.2. Timer Data Register (TCDT)

The Timer Data Register (TCDT) reads the count value of the 32-bit Free-run Timer. This register can also be used to set the count value of the 32-bit Free-run Timer.

bit	31	30	29	28	27	26	25	24
Field	T[31:24]							
R/W Attribute	R,W							
Protection Attribute	-							
Initial Value	00000000							

bit	23	22	21	20	19	18	17	16
Field	T[23:16]							
R/W Attribute	R,W							
Protection Attribute	-							
Initial Value	00000000							

bit	15	14	13	12	11	10	9	8
Field	T[15:8]							
R/W Attribute	R,W							
Protection Attribute	-							
Initial Value	00000000							

bit	7	6	5	4	3	2	1	0
Field	T[7:0]							
R/W Attribute	R,W							
Protection Attribute	-							
Initial Value	00000000							

[bit31:0] T[31:0]: Timer Data Value Bits

The Timer Data Register (TCDT) is used to read the count value of the 32-bit 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 32-bit Free-run Timer is stopped (the Timer Enable bit(STOP) in the Timer State Control Register (TCCS) ="1").

The counter is cleared to the count value 0x00000000 as soon as any of the following factors occurs.

- Hardware reset. A counter is cleared immediately after being hardware reset.
- The Timer Clear bit (SCLR) in the Timer State Control Register (TCCS) is set to "1" while the 32-bit Free-run Timer is operating (the Timer Enable bit(STOP) in the Timer State Control Register (TCCS) ="0").
- The value of the Compare Clear Register (CPCLR) matches the timer count value in the up count mode (the Timer Count Mode bit(MODE) in the Timer State Control Register (TCCS) ="0").
- The Timer Data Register (TCDT) is set to 0x00000000 while the 32-bit Free-run Timer is stopped (the Timer Enable bit(STOP) in the Timer State Control Register (TCCS) ="1").

If the Timer Clear bit (SCLR) in the Timer State Control Register (TCCS) is set to "1" or the value of the Compare Clear Register (CPCLR) matches the timer count value, the counter is cleared in synchronization with the count timing.

Notes:

- *The 32-bit Free-run Timer is not cleared to 0x00000000 even if the Timer Clear bit (SCLR) in the Timer State Control Register (TCCS) is set to "1" while the 32-bit Free-run Timer is stopped (the Timer Enable bit(STOP) in the Timer State Control Register (TCCS) ="1").*
- *For access to this register, use 32-bit 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 32-bit 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	RX,WX							
Protection Attribute	-							
Initial Value	XXXXXXXX							

bit	23	22	21	20	19	18	17	16
Field	Reserved							
R/W Attribute	RX,WX							
Protection Attribute	-							
Initial Value	XXXXXXXX							

bit	15	14	13	12	11	10	9	8
Field	ECKE	IRQZF	IRQZE	MSI[2:0]			ICLR	ICRE
R/W Attribute	R/W	R,WX	R/W	R,W			R,WX	R/W
Protection Attribute	-							
Initial Value	0	0	0	000			0	0

bit	7	6	5	4	3	2	1	0
Field	BFE	STOP	MODE	SCLR	CLK[3:0]			
R/W Attribute	R/W	R/W	R/W	R0,W	R/W			
Protection Attribute	-							
Initial Value	0	1	0	0	0000			

[bit31:16] Reserved Bits

Reading these bits returns an undefined value.

Writing data to these bits has no effect on the operation.

[bit15] ECKE: Clock Selection Bit

This bit is used to select a bus clock or an external clock as the count clock of the 32-bit Free-run Timer.

When this bit is set to "0", a bus clock is selected.

To select the count clock frequency, the Clock Frequency Selection bits (CLK[3:0]) must also be selected.

When this bit is set to "1", an external clock is selected and the clock is input from the external clock input (TEXT).

This bit is cleared to "0" by writing "1" to the Clock Selection Clear bit(ECKEC) in the Timer State Control Clear Register (TCCSC).
This bit is set to "1" by writing "1" to the Clock Selection Set bit(ECKES) in the Timer State Control Set Register (TCCSS).

Bit	Description
0	Bus clock
1	External clock

Note:

- *The count clock is changed immediately when this bit is changed. Therefore, change this bit when the 32-bit Output Compare and the 32-bit Input Capture are stopped if the 32-bit Free-run Timer is connected to them.*

[bit14] IRQZF: 0 Detection Interrupt Flag

This flag is set to "1" when the count value of the 32-bit Free-run Timer is 0x00000000.

This flag is a read-only bit. Writing data to this flag has no effect on the operation.

This flag is cleared by writing "1" to the 0 Detection Interrupt Flag Clear(IRQZFC) in the Timer State Control Clear Register (TCCSC).

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 Timer Clear bit(SCLR)) while the 32-bit Free-run Timer is operating (the Timer Enable bit (STOP)="0").*
- *In the up/down count mode (the Timer Count Mode bit(MODE) ="1"), this flag is set to "1" when an interrupt that is set by the Interrupt Mask Selection bits (MSI[2:0]) occurs. This flag is not set to "1" when no interrupt occurs.*
- *In the up count mode (the Timer Count Mode bit(MODE) ="0"), this flag is set to "1" every time 0 is detected regardless of the value of the Interrupt Mask Selection bits (MSI[2:0]).*

[bit13] IRQZE: 0 Detection Request Enable Bit

When this bit is set to "1" and the 0 Detection Interrupt Flag (IRQZF) is set to "1", an interrupt request is generated.

This bit is cleared to "0" by writing "1" to the 0 Detection Request Enable Clear bit(IRQZEC) in the Timer State Control Clear Register (TCCSC).

This bit is set to "1" by writing "1" to the 0 Detection Request Enable Set bit (IRQZES) in the Timer State Control Set Register (TCCSS).

Bit	Description
0	Disable interrupt requests.
1	Enable interrupt requests.

[bit12:10] MSI[2:0]: Interrupt Mask Selection Bits

For the Interrupt Mask Mode bit 2(MODE2) in the Timer Extended State Control Register (TECCS) ="0"

- In the up count mode (the Timer Count Mode bit(MODE)="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 (the Timer Count Mode bit(MODE) ="1"), these bits are used to set the number of times to mask the 0 detection interrupt flag.
- When these bits are set to "000", the interrupt flags are not masked.

For the Interrupt Mask Mode bit 2(MODE2) in the Timer Extended State Control Register (TECCS) ="1"

- In the up/down count mode (the Timer Count Mode bit(MODE) ="1"), these bits are used to set the number of times to mask the 0 detection interrupt flag.
- In the up count mode (the Timer Count Mode bit(MODE) ="0"), it is prohibited to use these bits to make this setting.

Bit	Description
MSI[2:0]	
000	Generate an interrupt flag at the first match occurrence.
001	Generate an interrupt flag at the second match occurrence.
010	Generate an interrupt flag at the third match occurrence.
011	Generate an interrupt flag at the fourth match occurrence.
100	Generate an interrupt flag at the fifth match occurrence.
101	Generate an interrupt flag at the sixth match occurrence.
110	Generate an interrupt flag at the seventh match occurrence.
111	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

This flag is set to "1" when the value of the Compare Clear Register (CPCLR) and the value of the 32-bit Free-run Timer match.

This flag is a read-only bit. Writing data to these bits has no effect on the operation.

This flag is cleared by writing "1" to the Compare Clear Interrupt Flag Clear(ICLRC) in the Timer State Control Clear Register (TCCSC).

Bit	Description
0	No compare clear match
1	Compare clear match

Notes:

- *In the up count mode (the Timer Count Mode bit(MODE) ="0"), this flag is set to "1" when the interrupt flag set by the Interrupt Mask Selection bits (MSI[2:0]) occurs. This flag is not set to "1" when no interrupt occurs.*
- *In the up/down count mode (the Timer Count Mode bit(MODE) ="1"), regardless of the value of the Interrupt Mask Selection bits (MSI[2:0]), this flag 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 (ICLR) is set to "1", an interrupt request is generated.

This bit is cleared to "0" by writing "1" to the Compare Clear Interrupt Request Enable Clear bit(ICREC) in the Timer State Control Clear Register (TCCSC).
This bit is set to "1" by writing "1" to the Compare Clear Interrupt Request Enable Set bit(ICRES) in the Timer State Control Set Register (TCCSS).

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 32-bit Free-run Timer.

This bit is cleared to "0" by writing "1" to the Compare Clear Buffer Enable Clear bit(BFEC) in the Timer State Control Clear Register (TCCSC).
This bit is set to "1" by writing "1" to the Compare Clear Buffer Enable Set bit(BFES) in the Timer State Control Set Register (TCCSS).

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 32-bit Free-run Timer.

When this bit is set to "0":

The 32-bit Free-run Timer starts counting.

When this bit is set to "1":

The 32-bit Free-run Timer stops counting.

This bit is cleared to "0" by writing "1" to the Timer Enable Clear bit(STOPC) in the Timer State Control Clear Register (TCCSC).
This bit is set to "1" by writing "1" to the Timer Enable Set bit(STOPS) in the Timer State Control Set Register (TCCSS).

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 32-bit 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 0x00000000. 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 0x00000000, the count direction changes to up count again.

A value can be written to this bit regardless of whether the timer is operating (the Timer Enable bit (STOP) ="0") or stopped (the Timer Enable bit(STOP) ="1"). When the timer is operating, the value written to this bit is stored in a buffer. After that, when the timer value becomes 0x00000000, the count mode is set based on the value stored in the buffer.

This bit is cleared to "0" by writing "1" to the Timer Count Mode Clear bit(MODEC) in the Timer State Control Clear Register (TCCSC).
This bit is set to "1" by writing "1" to the Timer Count Mode Set bit(MODES) in the Timer State Control Set Register (TCCSS).

Bit	Description
0	Up count mode
1	Up/down count mode

[bit4] SCLR: Timer Clear Bit

This bit is used to initialize the 32-bit Free-run Timer.

Initializing the value of the 32-bit Free-run Timer:

The 32-bit Free-run Timer is initialized to 0x00000000 at the next count clock when "1" is written to this bit while the 32-bit Free-run Timer is operating (the Timer Enable bit (STOP) ="0").

The 32-bit Free-run Timer is not initialized if "1" is written to this bit when the 32-bit Free-run Timer is stopped (the Timer Enable bit (STOP) ="1").

Initializing the count direction of the 32-bit Free-run Timer:

If this bit is set to "1" while the 32-bit Free-run Timer is operating (the Timer Enable bit (STOP) = "0"), the 32-bit Free-run Timer is initialized and then starts counting up.

If this bit is set to "1" while the 32-bit Free-run Timer is stopped (the Timer Enable bit (STOP) = "1") and then the 32-bit Free-run Timer is restarted (the Timer Enable bit (STOP) = "0"), the 32-bit Free-run Timer always counts up when starting operation.

Suppose the 32-bit Free-run Timer stops operating (the Timer Enable bit (STOP) = "1") while counting down and this bit is set to "1". After that, even if the 32-bit Free-run Timer resumes operation (the Timer Enable bit (STOP) = "1"), the 32-bit Free-run Timer counts up when starting operation.

This bit is set to "1" by writing "1" to the Timer Clear Set bit(SCLRS) in the Timer State Control Set Register (TCCSS).

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 Timer Clear bit(SCLR) before updating the timer count after "1" is written to the Timer Clear bit(SCLR) 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 32-bit Free-run Timer.

The clock frequency is changed immediately upon setting these bits.

CLK[3:0]	Description					
	Count Clock	ϕ = 40 MHz	ϕ = 20 MHz	ϕ = 10 MHz	ϕ = 5 MHz	ϕ = 2.5 MHz
0000	ϕ	25 ns	50 ns	100 ns	200 ns	400 ns
0001	ϕ/2	50 ns	100 ns	200 ns	400 ns	800 ns
0010	ϕ/4	100 ns	200 ns	400 ns	800 ns	1.6 us
0011	ϕ/8	200 ns	400 ns	800 ns	1.6 us	3.2 us
0100	ϕ/16	400 ns	800 ns	1.6 us	3.2 us	6.4 us
0101	ϕ/32	800 ns	1.6 us	3.2 us	6.4 us	12.8 us
0110	ϕ/64	1.6 us	3.2 us	6.4 us	12.8 us	25.6 us
0111	ϕ/128	3.2 us	6.4 us	12.8 us	25.6 us	51.2 us
1000	ϕ/256	6.4 us	12.8 us	25.6 us	51.2 us	102.4 us
Other settings are prohibited.	-	-	-	-	-	-

ϕ: Bus 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 32-bit Free-run Timer.

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	RX,WX							
Protection Attribute	-							
Initial Value	XXXXXXXX							

bit	23	22	21	20	19	18	17	16
Field	Reserved							
R/W Attribute	RX,WX							
Protection Attribute	-							
Initial Value	XXXXXXXX							

bit	15	14	13	12	11	10	9	8
Field	Reserved				MODE2	MSI[5:3]		
R/W Attribute	RX,WX				R/W	R,W		
Protection Attribute	-							
Initial Value	XXXX				0	000		

bit	7	6	5	4	3	2	1	0
Field	Reserved							
R/W Attribute	RX,WX							
Protection Attribute	-							
Initial Value	XXXXXXXX							

[bit31:12] Reserved: Reserved Bits

Reading these bits returns an undefined value.

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 32-bit Free-run Timer is in the up/down count mode (the Timer Count Mode bit(MODE) in the Timer State Control Register (TCCS) ="1").

If "1" is written to this bit when the 32-bit Free-run Timer is in the up/down count mode (the Timer Count Mode bit(MODE) in the Timer State Control Register (TCCS) ="1"), the value set to the Compare Clear Interrupt Mask Selection bits (MSI[5:3]) of this register is enabled and the Compare Clear Interrupt Flag (ICLR) in the Timer State Control Register (TCCS) is masked the specified number of times. The value set to the Interrupt Mask Selection bits (MSI[2:0]) in the Timer State Control Register (TCCS) becomes

valid as the number of times to mask the 0 Detection Interrupt Flag (IRQZF) in the Timer State Control Register (TCCS).

Bit		Description
MODE2	MODE*	
0	0	The setting values of Compare Clear Interrupt Mask Selection bits(MSI[5:3]) are invalid.
0	1	The setting values of Compare Clear Interrupt Mask Selection bits(MSI[5:3]) are invalid.
1	0	The setting is prohibited (the operation is not guaranteed).
1	1	The setting values of Compare Clear Interrupt Mask Selection bits(MSI[5:3]) are valid.

* Timer Count Mode bit (MODE) in the Timer State Control Register (TCCS)

Note:

- *The operation when "1" is written to this bit is not guaranteed when the 32-bit 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 (MODE2) is "1" and the 32-bit Free-run Timer is in the up/down count mode (the Timer Count Mode bit(MODE) in the Timer State Control Register (TCCS) ="1"). These bits are used to set the number of times to mask the Compare Clear Interrupt Flag (ICLR) in the Timer State Control Register (TCCS).

The number of times to mask the 0 Detection Interrupt Flag (IRQZF) in the Timer State Control Register (TCCS) is set by Interrupt Mask Selection bits(MSI[2:0]) in the Timer State Control Register (TCCS).

The Compare Clear Interrupt Flag (ICLR) in the Timer State Control Register (TCCS) is not masked when these bits are set to "000".

Bit	Description
MSI[5:3]	
000	Generate an interrupt flag at the first match occurrence.
001	Generate an interrupt flag at the second match occurrence.
010	Generate an interrupt flag at the third match occurrence.
011	Generate an interrupt flag at the fourth match occurrence.
100	Generate an interrupt flag at the fifth match occurrence.
101	Generate an interrupt flag at the sixth match occurrence.
110	Generate an interrupt flag at the seventh match occurrence.
111	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 32-bit Free-run Timer is operating (the Timer Enable bit (STOP) in the Timer State Control Register (TCCS) ="0"), the value written to the mask register is reloaded to the counter only when the mask counter becomes "000".*
- *When the 32-bit Free-run Timer is stopped (the Timer Enable bit (STOP) in the Timer State Control Register (TCCS) ="1"), the value written to the mask register is immediately reloaded to the mask counter.*

[bit7:0] Reserved: Reserved Bits

Reading these bits returns an undefined value.

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	0			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

Reading these bits returns "0".

Always write "0" to these bits.

[bit15] ECKEC: Clock Selection Clear Bit

It is the bit which sets the Clock Selection bit (ECKE) in the Timer State Control Register (TCCS) to "0".

Bit	Description	
	Writing	Reading
0	No effect.	Always read "0".
1	Set the Clock Selection bit (ECKE) in the Timer State Control Register (TCCS) to "0".	

[bit14] IRQZFC: 0 Detection Interrupt Flag Clear

It is the bit which sets the 0 Detection Interrupt Flag (IRQZF) in the Timer State Control Register (TCCS) to "0".

Bit	Description	
	Writing	Reading
0	No effect.	Always read "0".
1	Set the 0 Detection Interrupt Flag (IRQZF) in the Timer State Control Register (TCCS) to "0".	

[bit13] IRQZEC: 0 Detection Request Enable Clear Bit

It is the bit which sets the 0 Detection Request Enable bit (IRQZE) in the Timer State Control Register (TCCS) to "0".

Bit	Description	
	Writing	Reading
0	No effect.	Always read "0".
1	Set the 0 Detection Request Enable bit (IRQZE) in the Timer State Control Register (TCCS) to "0".	

[bit12:10] Reserved: Reserved Bits

Reading these bits returns "0".

Always write "0" to these bits.

[bit9] ICLRC: Compare Clear Interrupt Flag Clear

It is the bit which sets the Compare Clear Interrupt Flag (ICLR) in the Timer State Control Register (TCCS) to "0".

Bit	Description	
	Writing	Reading
0	No effect.	Always read "0".
1	Set the Compare Clear Interrupt Flag (ICLR) in the Timer State Control Register (TCCS) to "0".	

[bit8] ICREC: Compare Clear Interrupt Request Enable Clear Bit

It is the bit which sets the Compare Clear Interrupt Request Enable bit (ICRE) in the Timer State Control Register (TCCS) to "0".

Bit	Description	
	Writing	Reading
0	No effect.	Always read "0".
1	Set the Compare Clear Interrupt Request Enable bit (ICRE) in the Timer State Control Register (TCCS) to "0".	

[bit7] BFEC: Compare Clear Buffer Enable Clear Bit

It is the bit which sets the Compare Clear Buffer Enable bit (BFE) in the Timer State Control Register (TCCS) to "0".

Bit	Description	
	Writing	Reading
0	No effect.	Always read "0".
1	Set the Compare Clear Buffer Enable bit (BFE) in the Timer State Control Register (TCCS) to "0".	

[bit6] STOPC: Timer Enable Clear Bit

It is the bit which sets the Timer Enable bit (STOP) in the Timer State Control Register (TCCS) to "0".

Bit	Description	
	Writing	Reading
0	No effect.	Always read "0".
1	Set the Timer Enable bit (STOP) in the Timer State Control Register (TCCS) to "0".	

[bit5] MODEC: Timer Count Mode Clear Bit

It is the bit which sets the Timer Count Mode bit (MODE) in the Timer State Control Register (TCCS) to "0".

Bit	Description	
	Writing	Reading
0	No effect.	Always read "0".
1	Set the Timer Count Mode bit (MODE) in the Timer State Control Register (TCCS) to "0".	

[bit4:0] Reserved: Reserved Bits

Reading these bits returns "0".

Always write "0" to these bits.

4.6. Timer State Control Set Register (TCCSS)

The Timer State Control Set Register (TCCSS) 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,WX			
Protection Attribute	-							
Initial Value	0	0	0	0	0000			

[bit31:16] Reserved: Reserved Bits

Reading these bits returns "0".

Always write "0" to these bits.

[bit15] ECKES: Clock Selection Set Bit

It is the bit which sets the Clock Selection bit (ECKE) in the Timer State Control Register (TCCS) to "1".

Bit	Description	
	Writing	Reading
0	No effect.	Always read "0".
1	Set the Clock Selection bit (ECKE) in the Timer State Control Register (TCCS) to "1".	

[bit14] Reserved: Reserved Bit

Reading this bit returns "0".

Always write "0" to this bit.

[bit13] IRQZES: 0 Detection Request Enable Set Bit

It is the bit which sets the 0 Detection Request Enable bit (IRQZE) in the Timer State Control Register (TCCS) to "1".

Bit	Description	
	Writing	Reading
0	No effect.	Always read "0".
1	Set the 0 Detection Request Enable bit (IRQZE) in the Timer State Control Register (TCCS) to "1".	

[bit12:9] Reserved: Reserved Bits

Reading these bits returns "0".

Always write "0" to these bits.

[bit8] ICRES: Compare Clear Interrupt Request Enable Set Bit

It is the bit which sets the Compare Clear Interrupt Request Enable bit (ICRE) in the Timer State Control Register (TCCS) to "1".

Bit	Description	
	Writing	Reading
0	No effect.	Always read "0".
1	Set the Compare Clear Interrupt Request Enable bit (ICRE) in the Timer State Control Register (TCCS) to "1".	

[bit7] BFES: Compare Clear Buffer Enable Set Bit

It is the bit which sets the Compare Clear Buffer Enable bit (BFE) in the Timer State Control Register (TCCS) to "1".

Bit	Description	
	Writing	Reading
0	No effect.	Always read "0".
1	Set the Compare Clear Buffer Enable bit (BFE) in the Timer State Control Register (TCCS) to "1".	

[bit6] STOPS: Timer Enable Set Bit

It is the bit which sets the Timer Enable bit (STOP) in the Timer State Control Register (TCCS) to "1".

Bit	Description	
	Writing	Reading
0	No effect.	Always read "0".
1	Set the Timer Enable bit (STOP) in the Timer State Control Register (TCCS) to "1".	

[bit5] MODES: Timer Count Mode Set Bit

It is the bit which sets the Timer Count Mode bit (MODE) in the Timer State Control Register (TCCS) to "1".

Bit	Description	
	Writing	Reading
0	No effect.	Always read "0".
1	Set the Timer Count Mode bit (MODE) in the Timer State Control Register (TCCS) to "1".	

[bit4] SCLRS: Timer Clear Set Bit

It is the bit which sets the Timer Clear bit (SCLR) in the Timer State Control Register (TCCS) to "1".

Bit	Description	
	Writing	Reading
0	No effect.	Always read "0".
1	Set the Timer Clear bit (SCLR) in the Timer State Control Register (TCCS) to "1".	

[bit3:0] Reserved: Reserved Bits

Reading these bits returns "0".

Always write "0" to these bits.

4.7. Free-Run Timer Debug Register(FRT_DEBUG)

The Free-run Timer Debug Register (FRT_DEBUG) performs enable/disable setting of debug.

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							DBGEN
R/W Attribute	R0,W0							R/W
Protection Attribute	-							
Initial Value	0000000							0

[bit15:1] Reserved: Reserved Bits

The read value is "0".

If writing to this bit, write "0".

[bit0] DBGEN: Debug Enable Bit

It is the bit which permits stopping operation of a 32-bit Free-run Timer with the debug input (DEBUG).

Bit	Description
0	Disable
1	Enable

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 32-bit access instructions to the Compare Clear Register (CPCLR) and the Compare Clear Buffer Register (CPCLRB).

When Accessing the Timer State Control Register (TCCS)

- 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 Compare Clear Interrupt Mask Selection bits(MSI[5:3]) in the Timer Extended State Control Register (TECCS).

Notes when Operating the 32-Bit Free-Run Timer

When Setting Using a Program

- When the hardware is reset, the count value becomes 0x00000000, but the 0 Detection Interrupt Flag (IRQZF) in the Timer State Control Register (TCCS) is not set.
- Because the timer mode bit (the Timer Count Mode bit(MODE) in the Timer State Control Register (TCCS)) has a buffer, a specified timer mode is enabled after 0 is detected.
- The Timer Clear bit (SCLR) in the Timer State Control Register (TCCS) ="1" 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 0x00000000 to the Compare Clear Register (CPCLR). Note that the following operations occur if 0x00000000 is set.
  - When the timer mode bit (the Timer Count Mode bit(MODE) in the Timer State Control Register (TCCS)) is in the up count mode (the Timer Count Mode bit(MODE) in the Timer State Control Register (TCCS) ="0"), the count value is updated to 0x00000000 and fixed to 0x00000000. Then, the 0 Detection Interrupt Flag (IRQZF) in the Timer State Control Register (TCCS) and the Compare Clear Interrupt Flag (ICLR) in the Timer State Control Register (TCCS) are set at every count clock.
  - When the Timer Count Mode bit (MODE) in the Timer State Control Register (TCCS) is in the up/down count mode (Timer Count Mode bit(MODE) in the Timer State Control Register (TCCS) ="1"), the count value is counted up from 0x00000000 to 0xFFFFFFFF, and this up count operation is repeated. The 0 Detection Interrupt Flag (IRQZF) in the Timer State Control Register (TCCS) and the Compare Clear Interrupt Flag (ICLR) in the Timer State Control Register (TCCS) are set to "1" when the count value becomes 0x00000000.

CHAPTER 42: 32-Bit Input Capture

This chapter describes the functions of the 32-bit input capture.

1. Overview of the 32-Bit Input Capture
2. Explanation of 32-Bit Input Capture Operation
3. Registers of the 32-Bit Input Capture
4. Precautions for Using This Device

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.

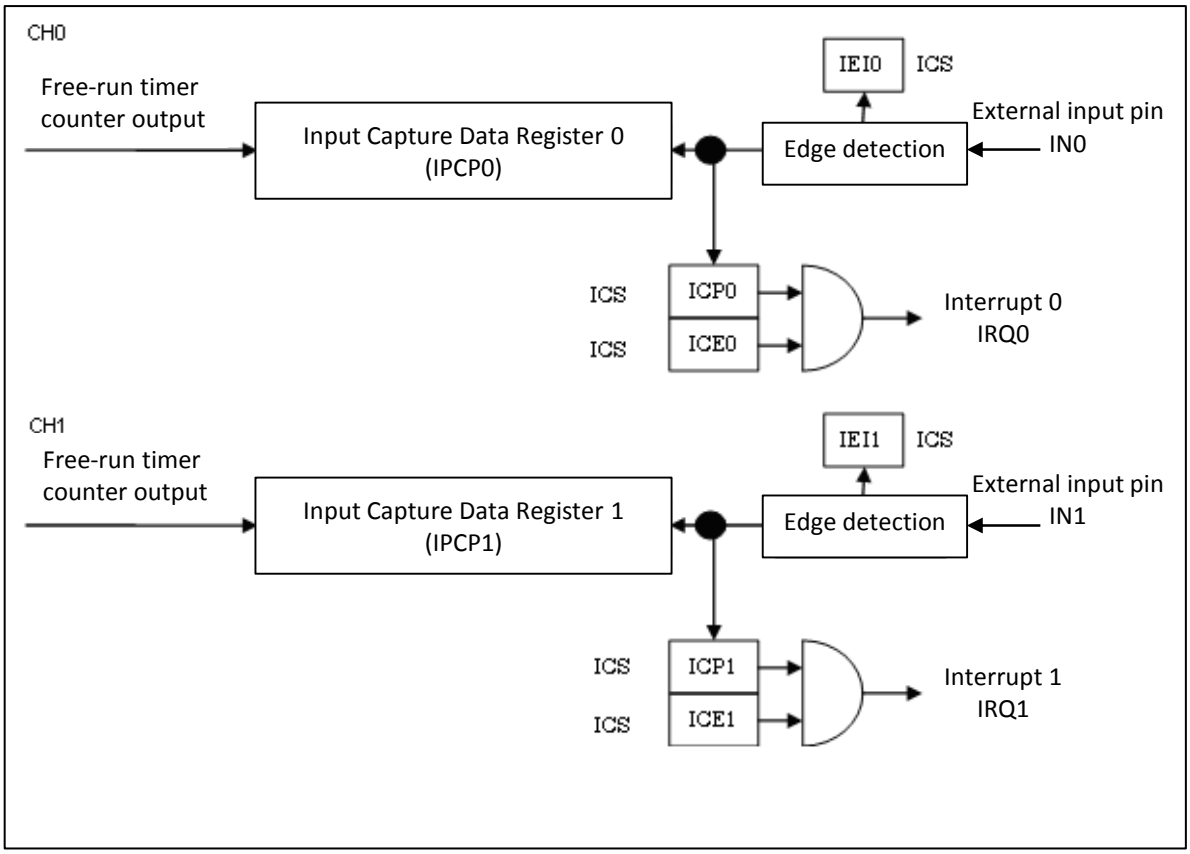
Functions of the 32-Bit Input Capture

- 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.

Configuration Diagram of the 32-Bit Input Capture

Figure 1-1 is a configuration diagram of the 32-bit input capture.

Figure 1-1 Configuration Diagram of 32-Bit Input Capture



2. Explanation of 32-Bit Input Capture Operation

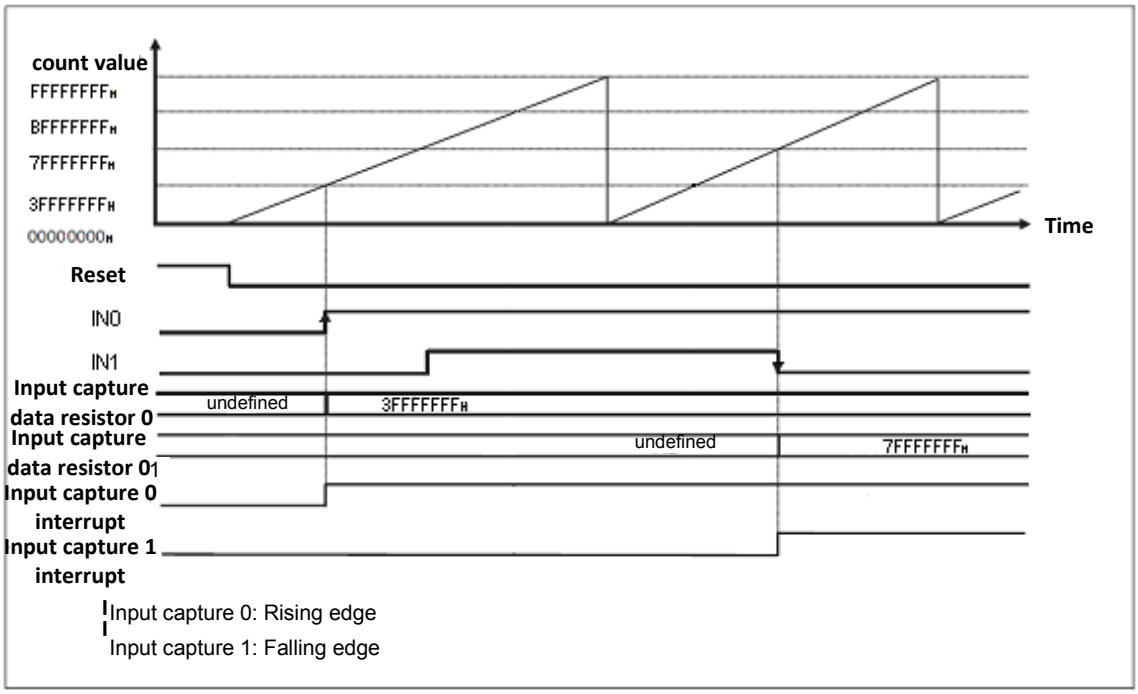
This section provides an overview of 32-bit input capture operation.

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.

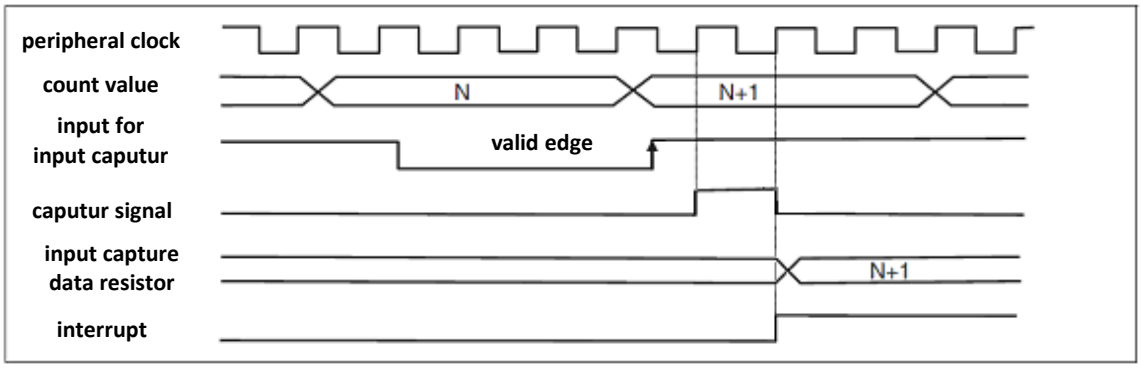
Input Capture Operation

Figure 2-1 Example of Input Capture Timing



Input Capture Input Timing

Figure 2-2 Example of Input Capture Timing for Input Signals



2.1. 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 2-1 Interrupt Control Bits and Interrupt Factor of Input Capture 1, 0

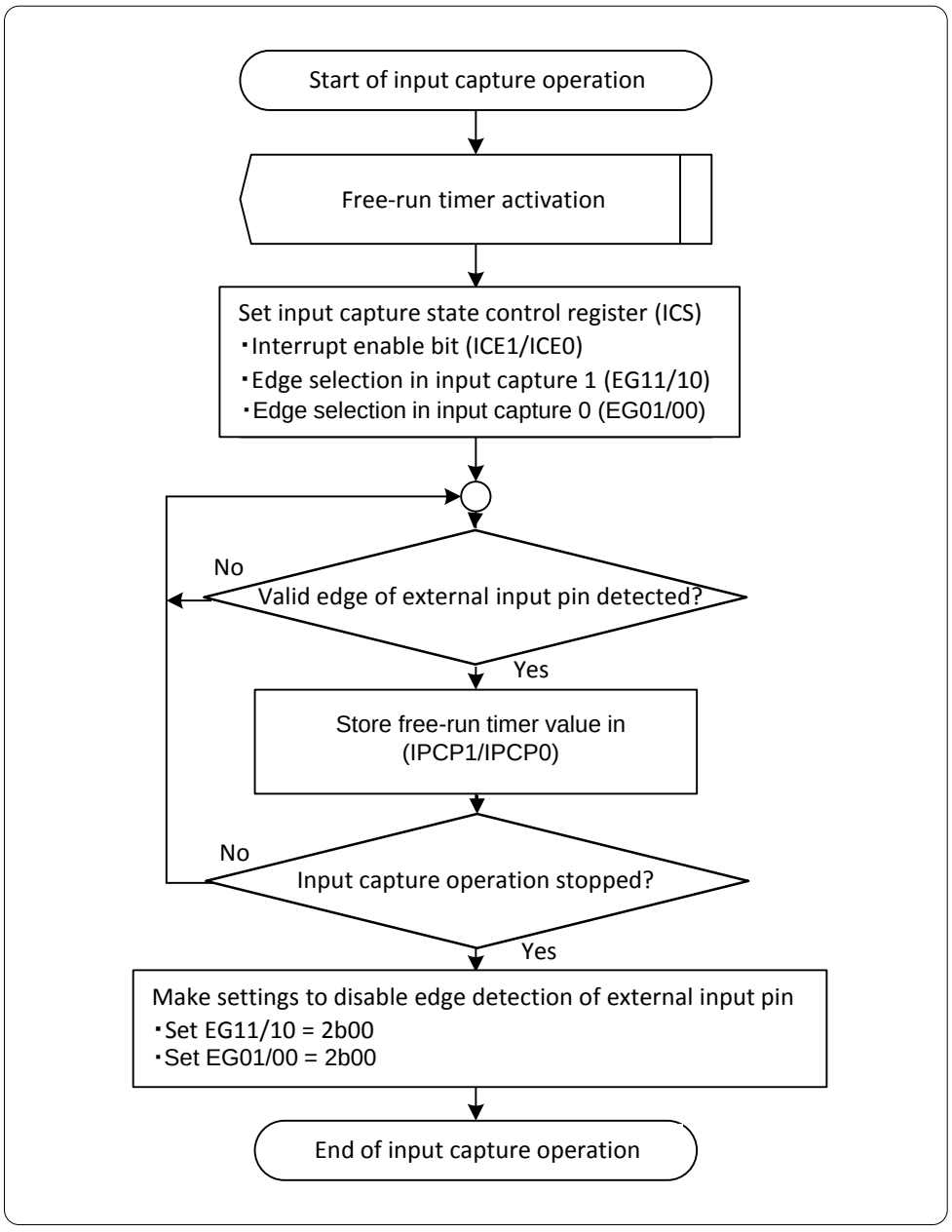
	Input Capture 1, 0
Interrupt Request Flag Bit	The interrupt request flag bits ICP1, 0 (bit7, bit6) in the input capture state control register (ICS)
Interrupt Request Enable Bit	The interrupt request enable bits ICE1, 0 (bit5, bit4) in the input capture state control register (ICS)
Interrupt Factor	A 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.

2.2. Setting Procedure Example

This section provides a setting procedure example of the input capture.

Figure 2-3 Procedure Example of Setting Input Capture Operation



3. Registers of the 32-Bit Input Capture

This section provides the register list of the 32-bit input capture.

Register List of the 32-Bit Input Capture

Table 3-1 Register List of the 32-Bit Input Capture

Abbreviated Register Name	Register Name	Reference
IPCP0/IPCP1	Input capture data registers 0, 1	3.1
ICS	Input capture state control register	3.2
ICSC	Input capture state control clear register	3.3
ICSS	Input capture state control set register	3.4

Register Bit Locations of 32-Bit Input Capture

Table 3-2 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	CP 25	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			

3.1. 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.*

3.2. 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 "4. 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.

3.3. 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.

3.4. 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.

4. 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 43: 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-T00P00R01L03-E1-GBB01

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.

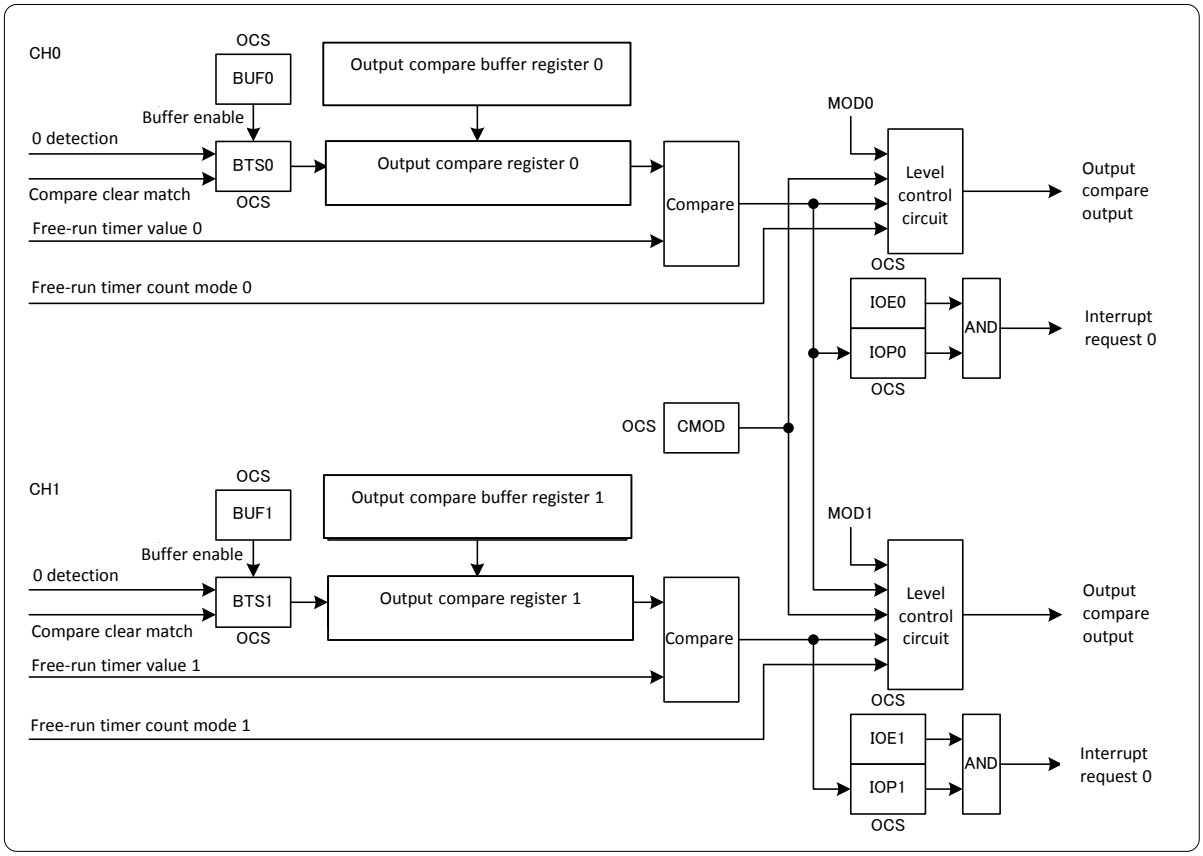
Functions of the 32-Bit Output Compare

- 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.

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.)

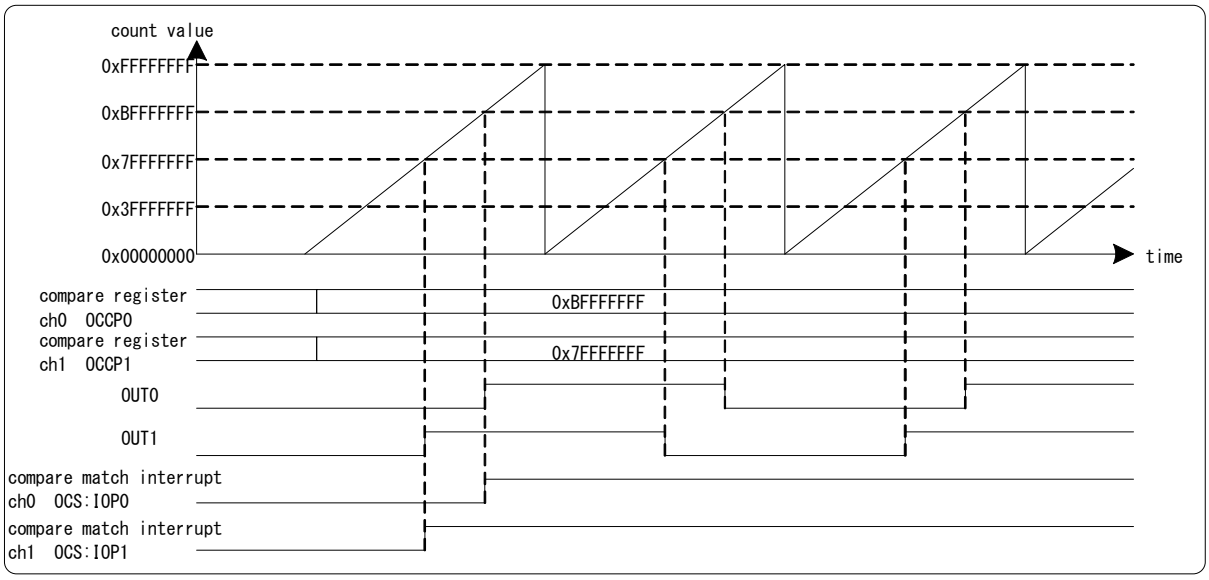
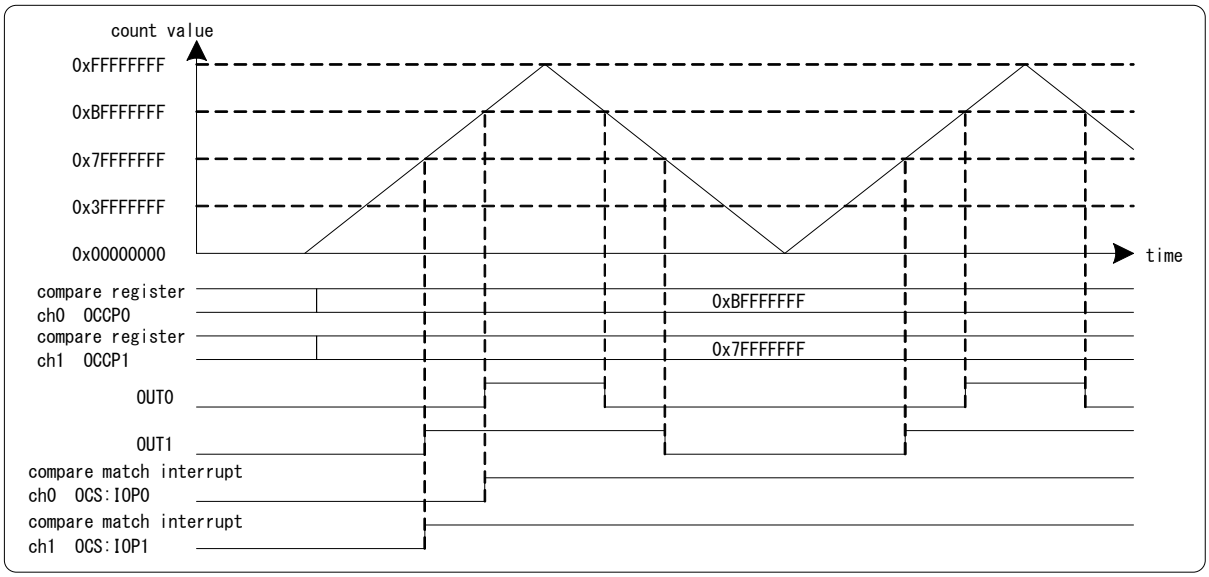


Figure 3-2 Output Waveform Example for When Compare Registers 0 and 1 are Separately Used with Initial Output Value = 0 (Free-Run Timer is in Up-Down Count Mode.)



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.)

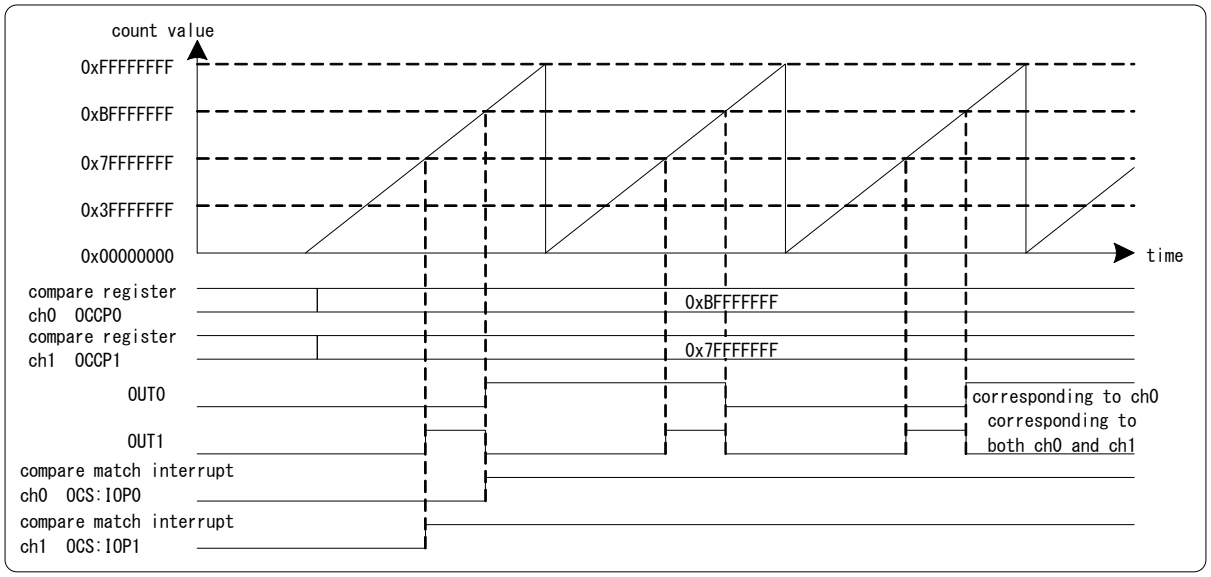
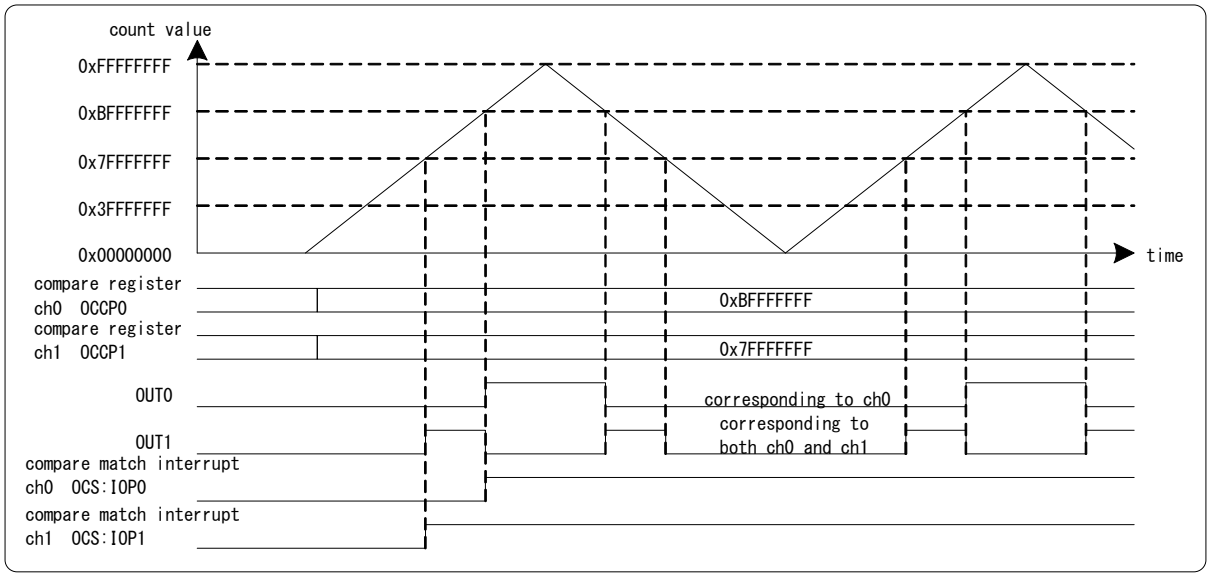
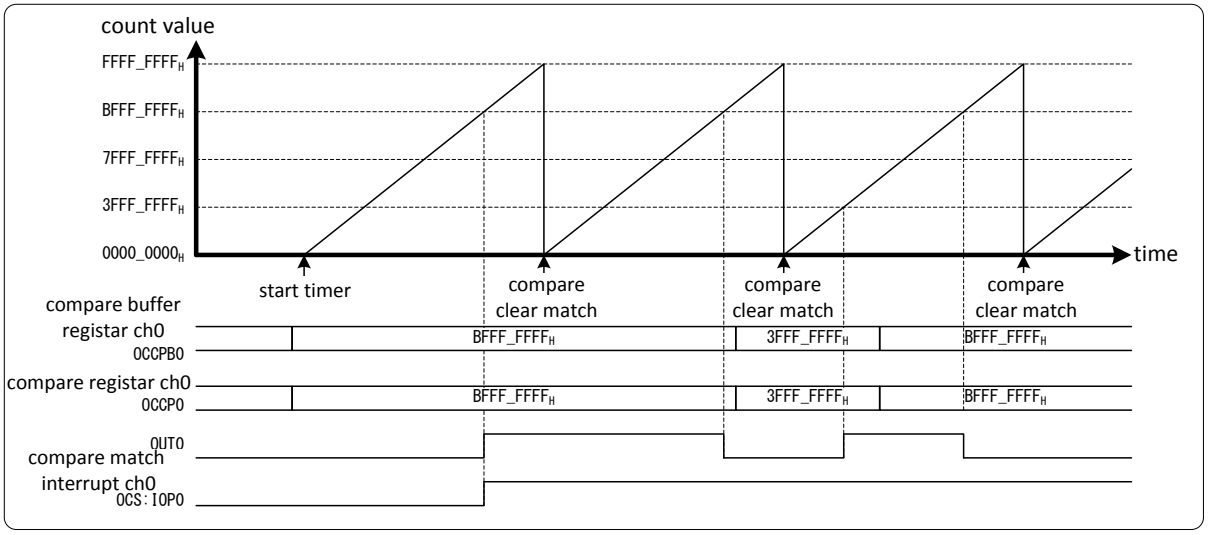


Figure 3-4 Output Waveform Example for When Compare Registers 0 and 1 are Used as a Pair with Initial Output Value = 0 (Free-Run Timer is in Up-Down Count Mode.)



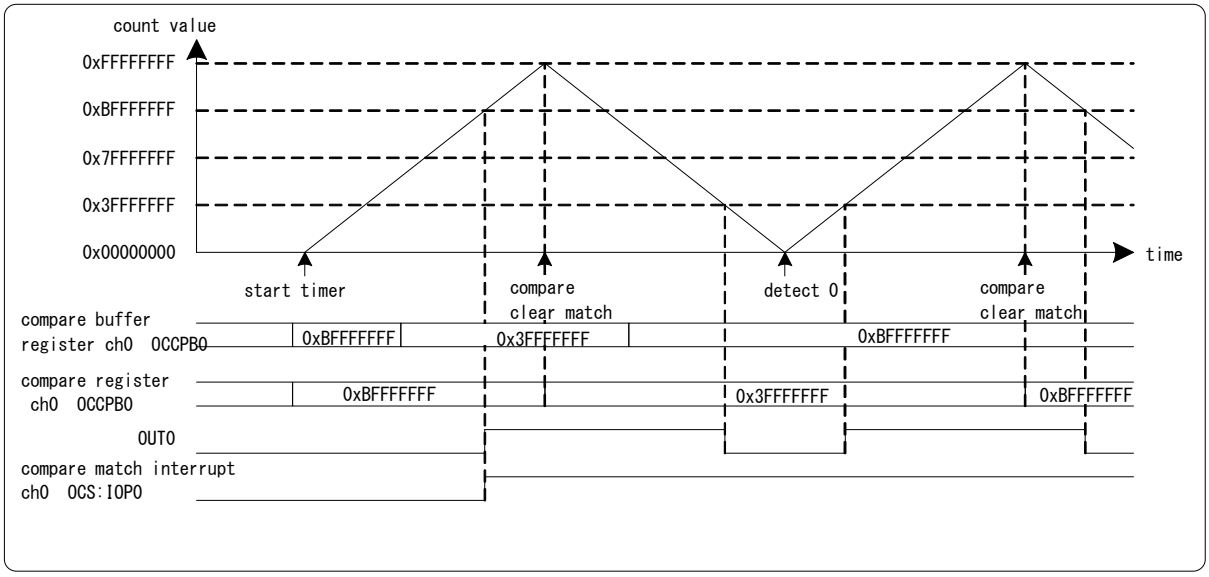
Output Waveform When the Compare Buffer is Disabled

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.)

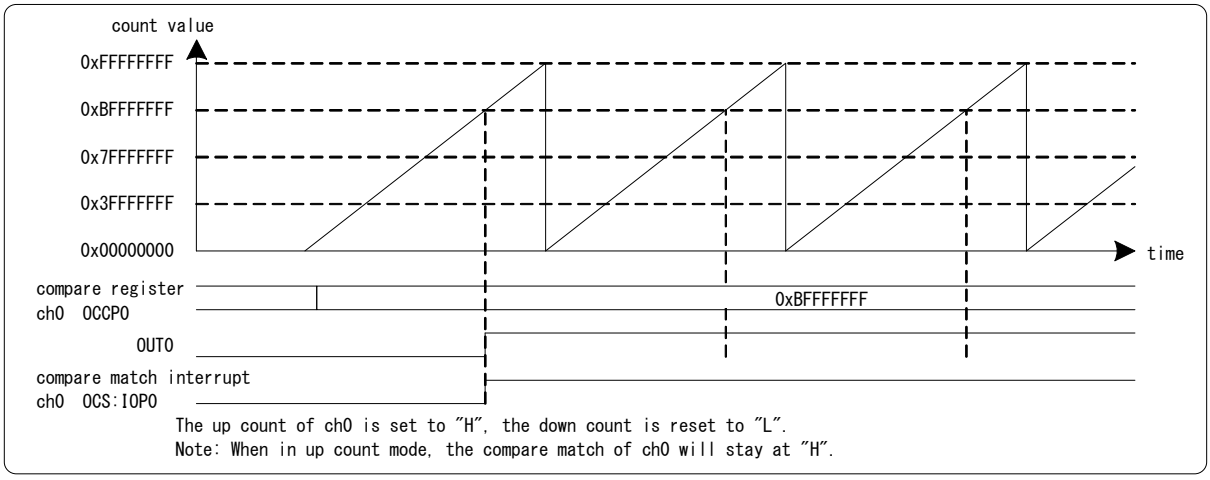
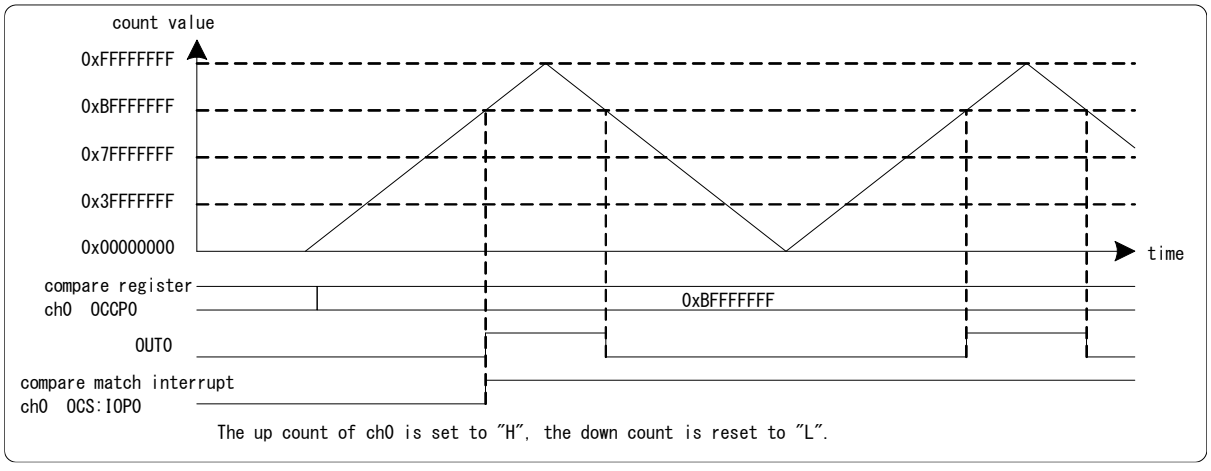


Figure 3-8 Output Waveform for When the Output Pin is Set/Reset at Occurrence of Output Compare Match (Free-Run Timer is in Up-Down Count Mode.)



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

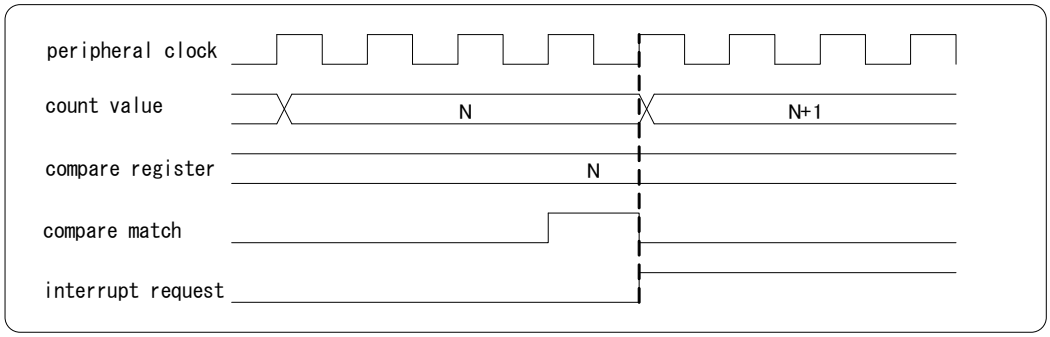
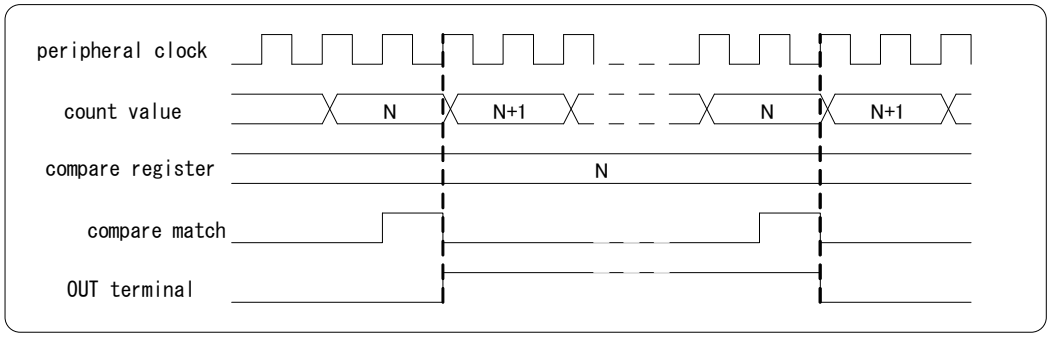


Figure 3-10 Timing of Change in OUT Pin Output



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
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

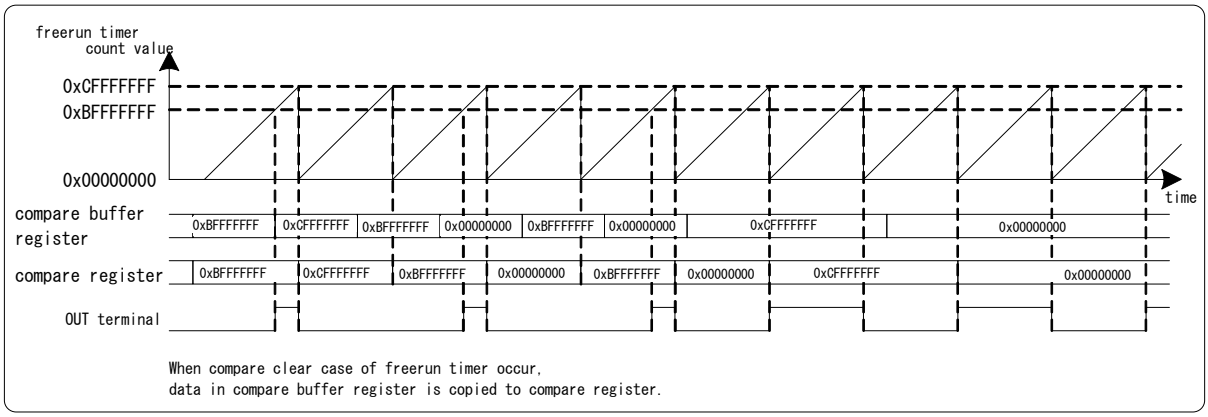
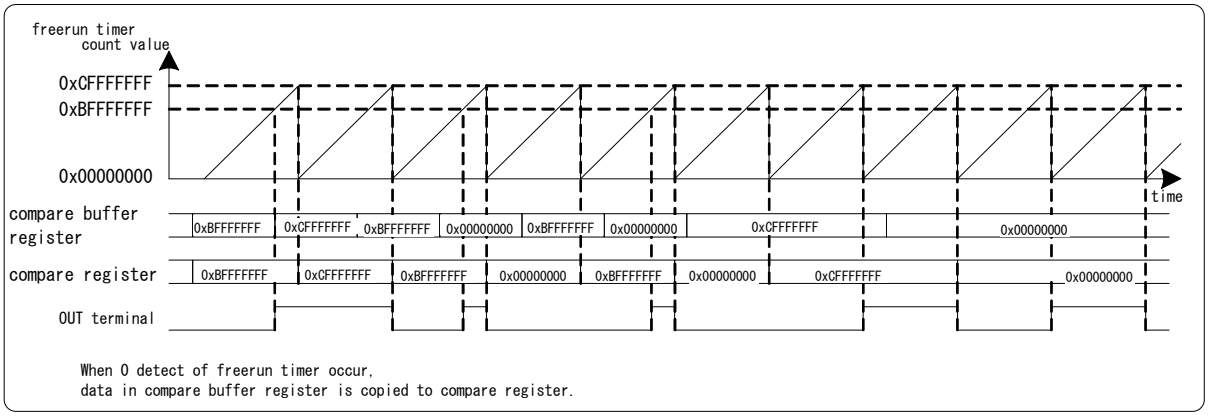


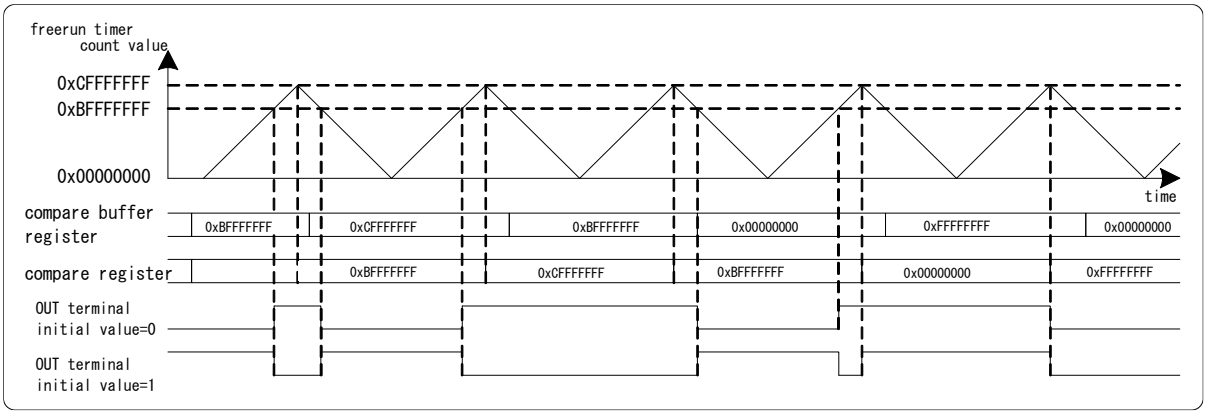
Figure 3-12 For Transfer When 0 is Detected



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)



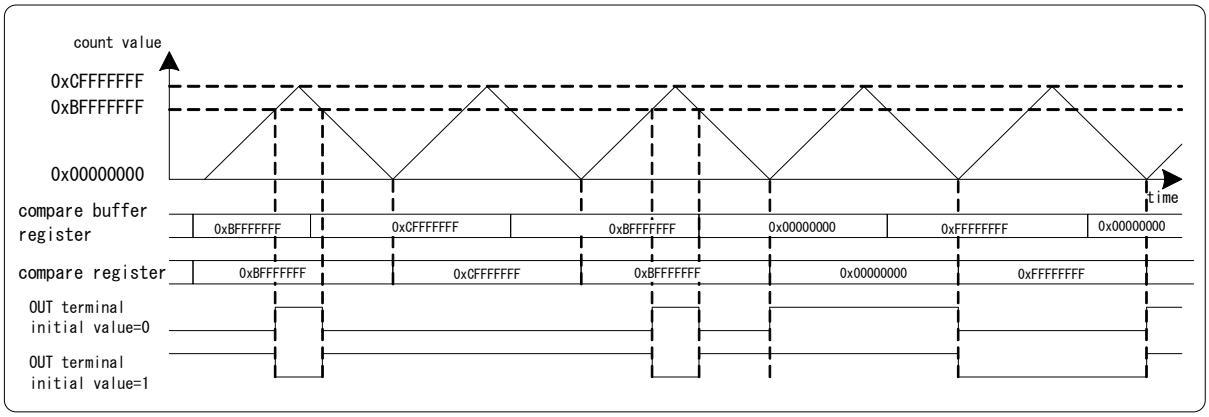
Notes:

- *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)



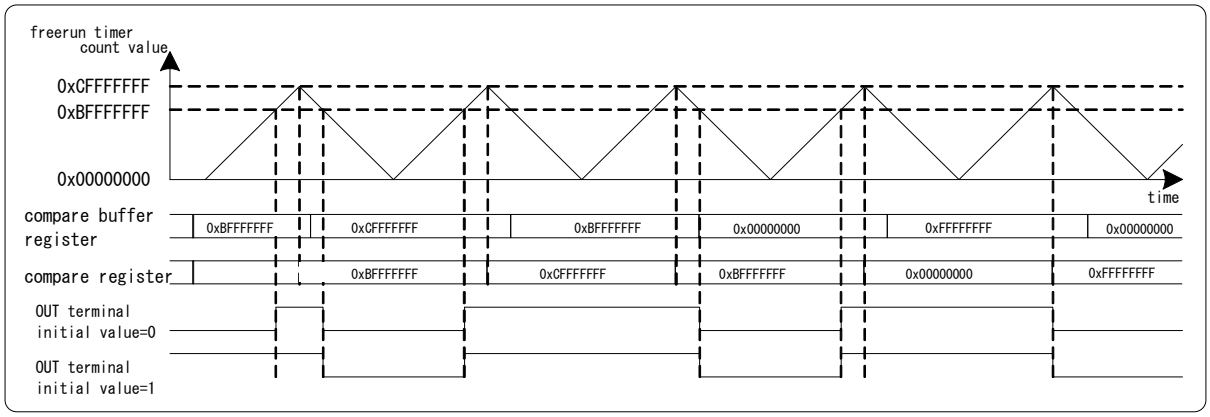
Notes:

- *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)



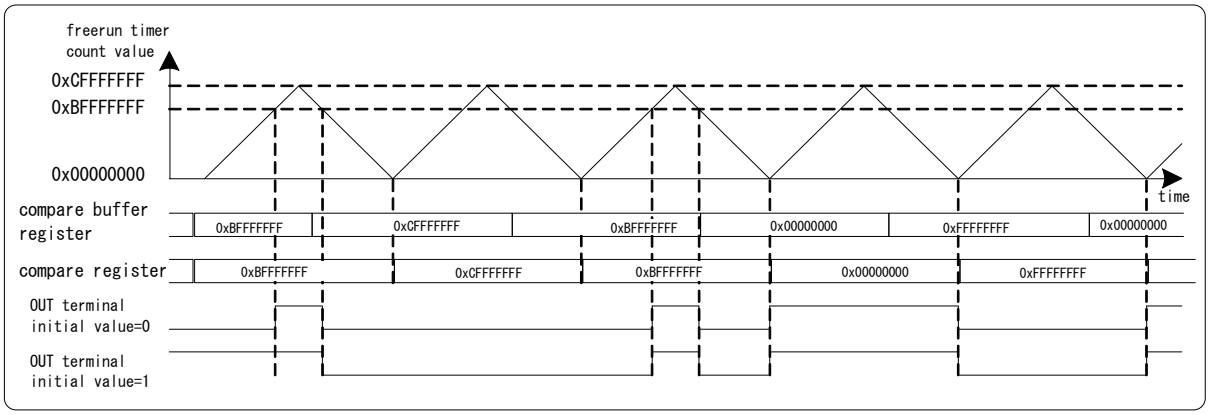
Notes:

- *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)



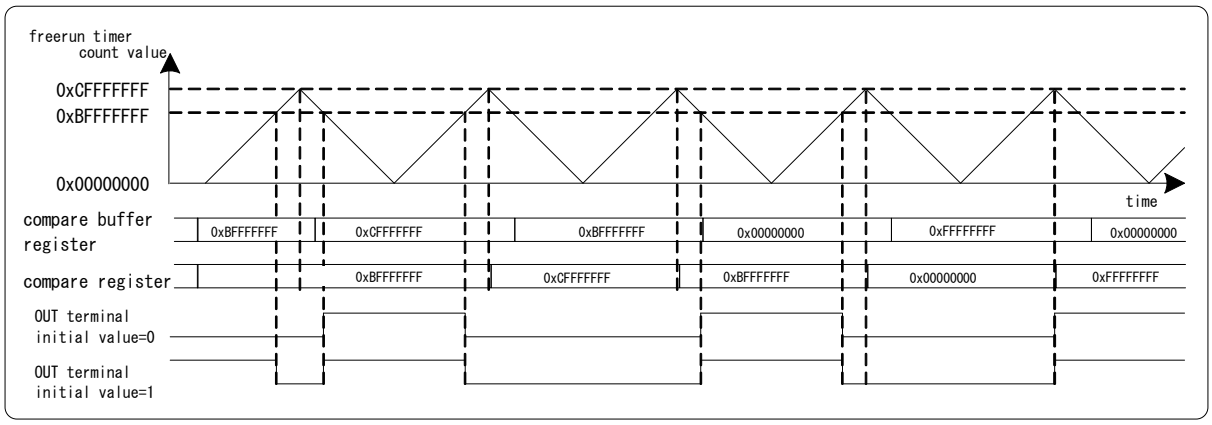
Notes:

- *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)



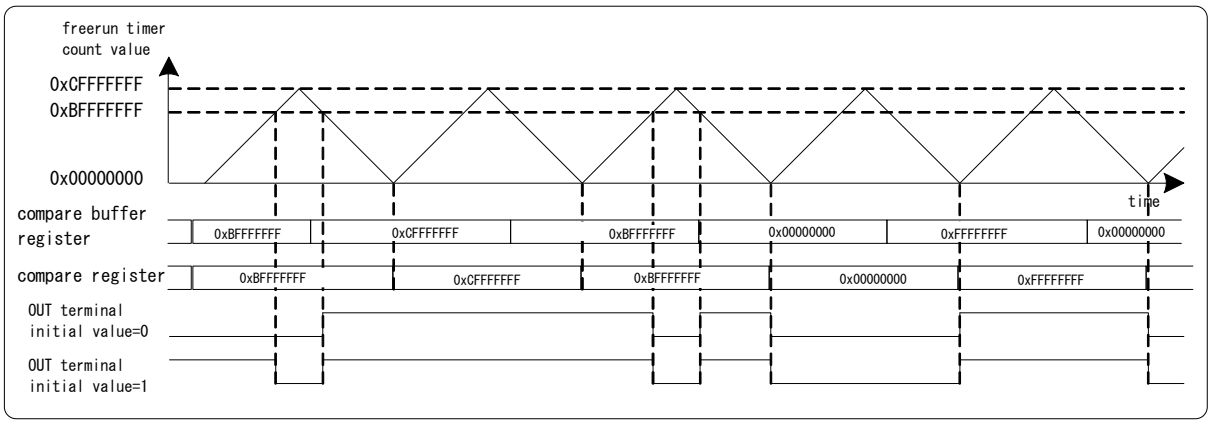
Notes:

- *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)

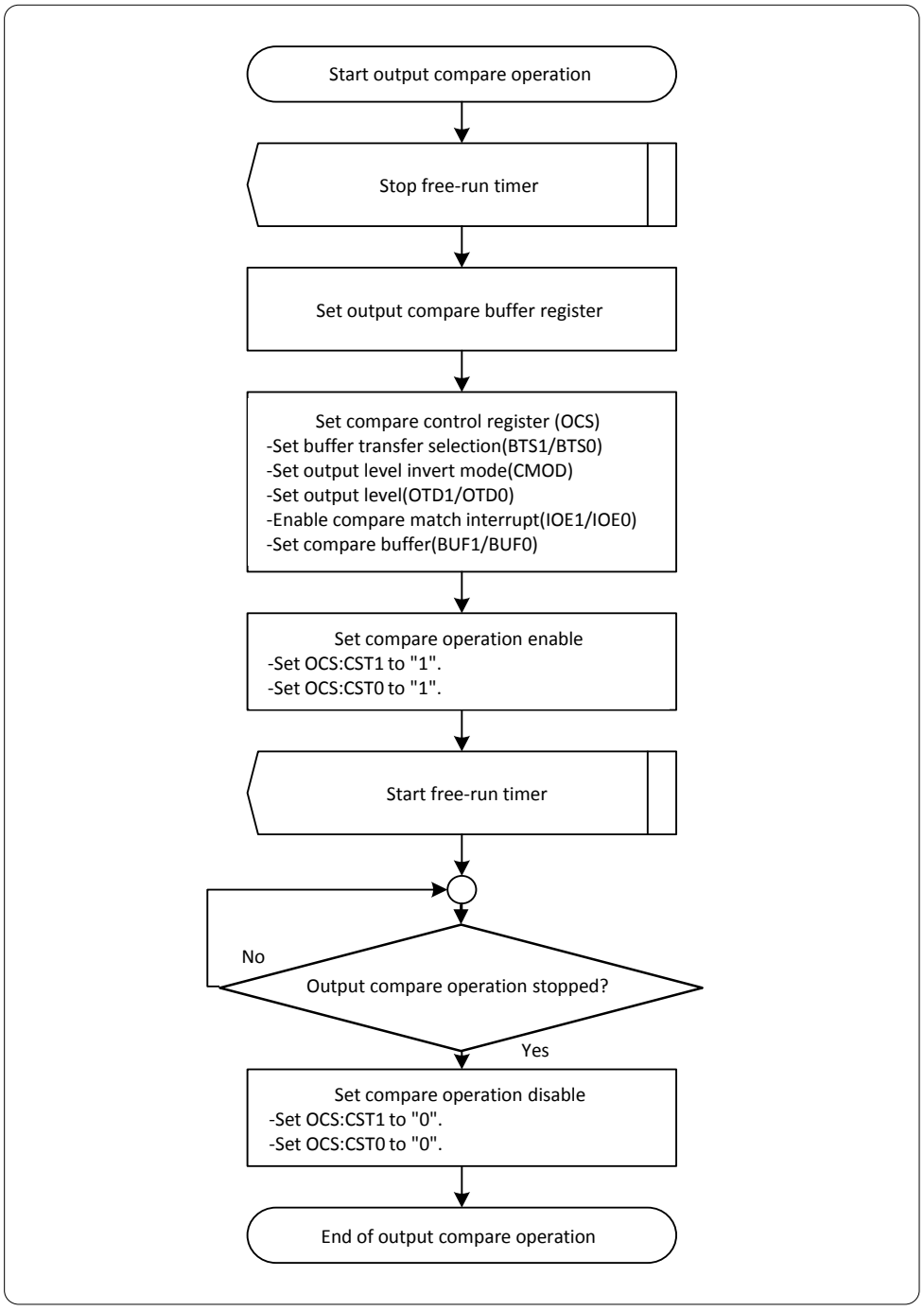


Notes:

- *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 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.
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.

Address of Register Mirror Area	Mirrored Peripheral	Mirror Area Behavior	PPU
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.14 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 BST1S 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." [x]
- 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 44: 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-T01P00R01L01-E1-GBM00

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

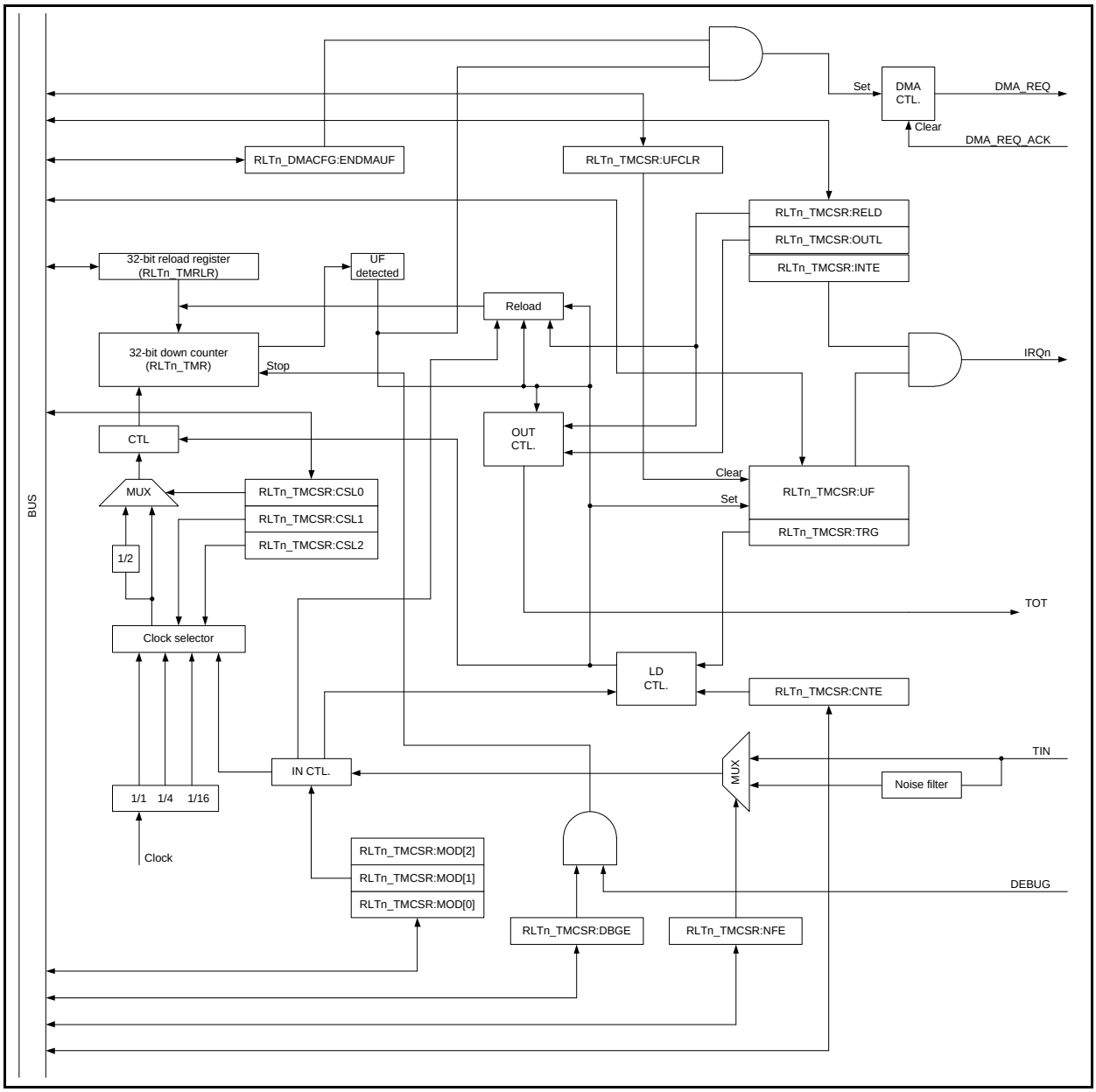
DMA and interrupts

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



Note:

- *In this implementation, RTLn_TMCBR:DBGE is reserved to 0.*

3. Operation of the 32-bit Reload Timer

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

3.1. Internal Clock and External Event Counter Operations of 32-bit Reload Timer

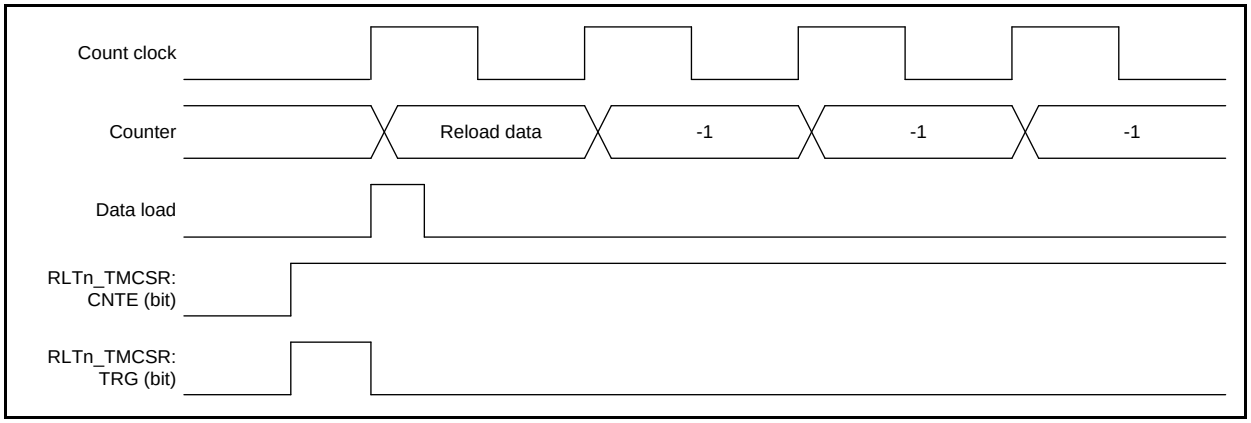
In internal clock mode, the peripheral clock with different divider settings can be selected as the clock source for operating the Reload Timer. The external input TIN can be selected as either a trigger input, or as a gate input by a register setting.
In event counter mode, the TIN is used as an external 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



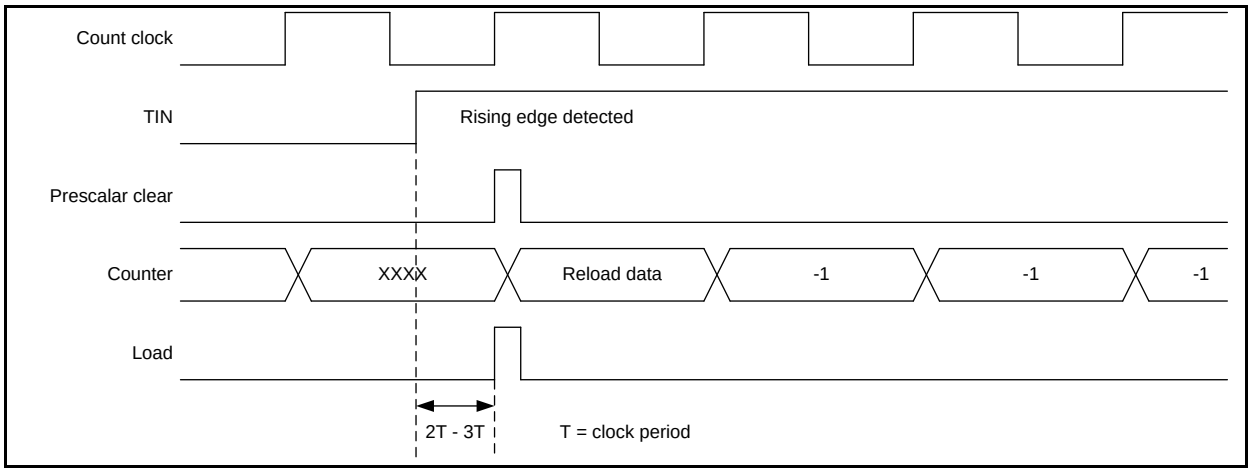
Input functions of 32-bit Reload Timer (in internal clock mode)

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.

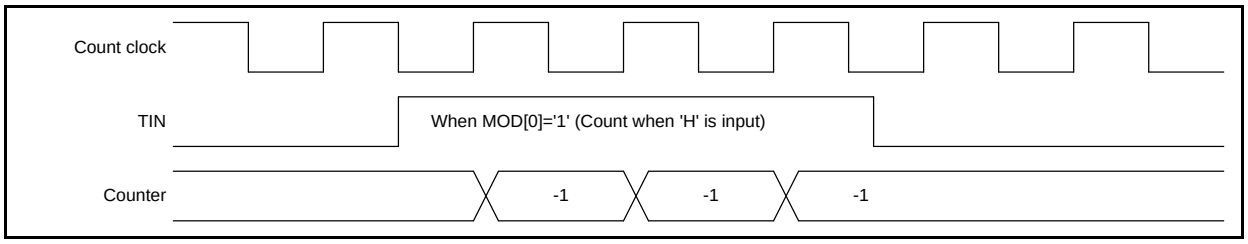
Figure 3-2 Trigger Input Operation of 32-bit Reload Timer



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

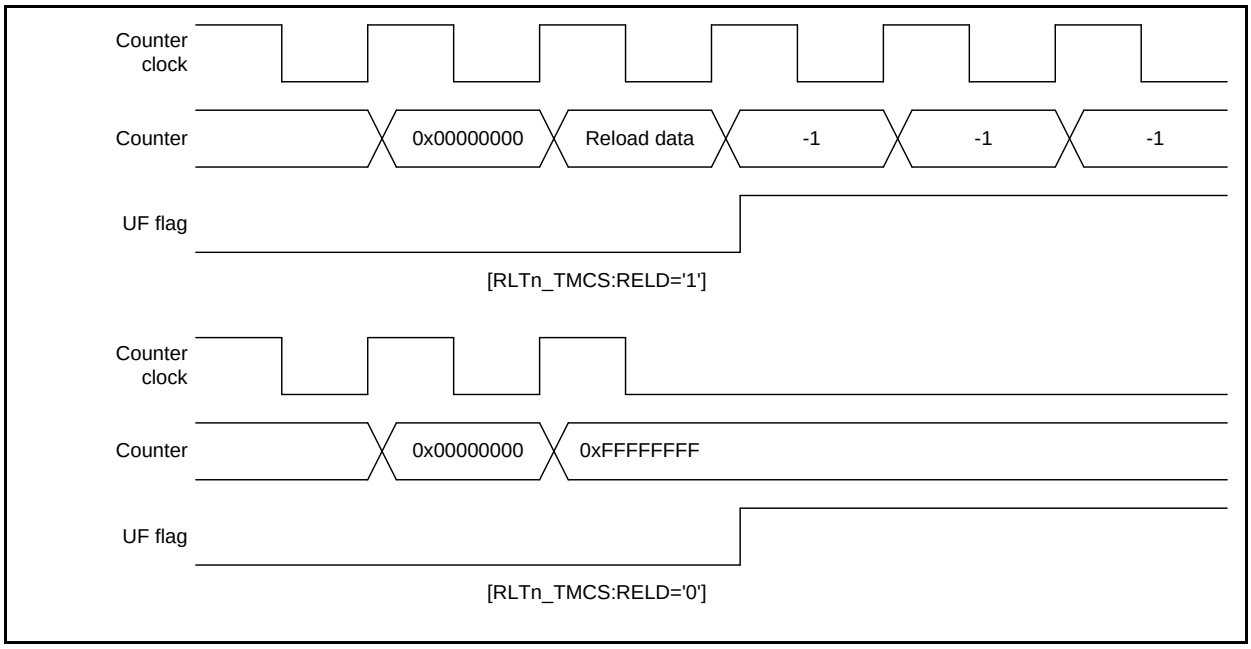
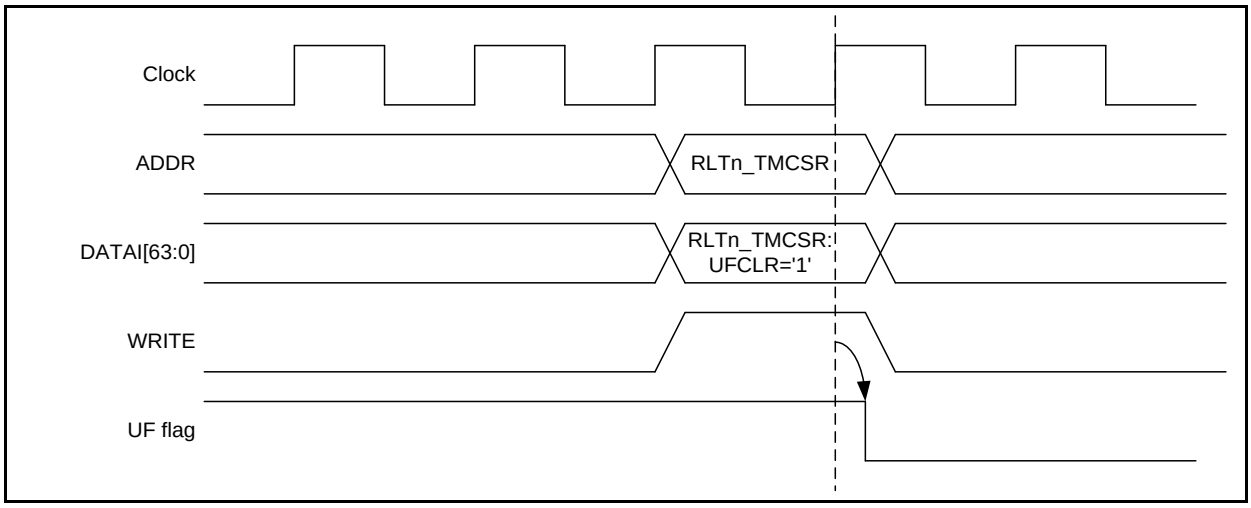


Figure 3-5 Clearing of Underflow bit



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

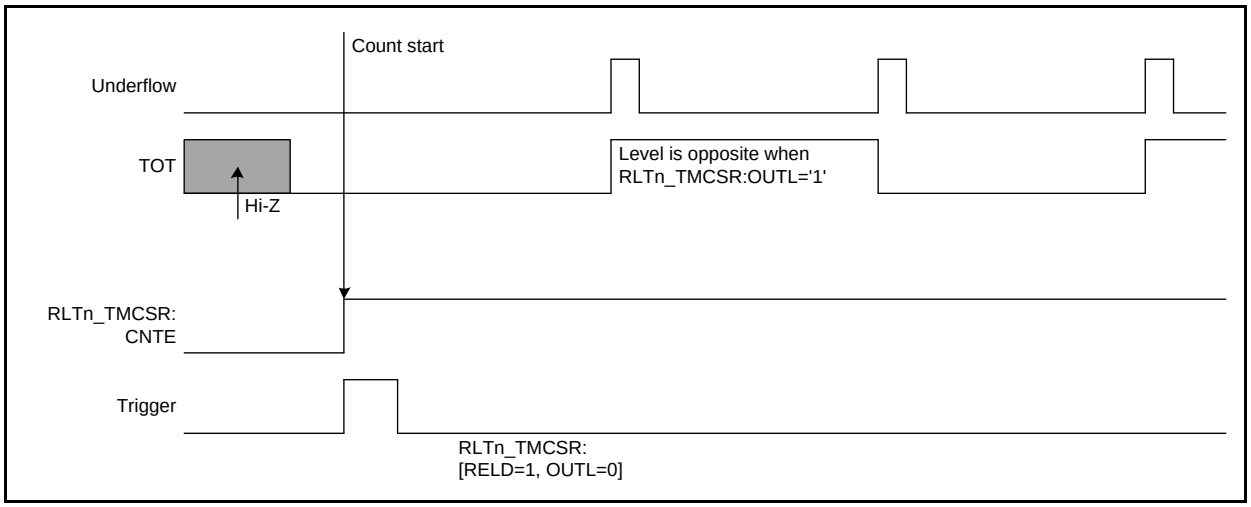
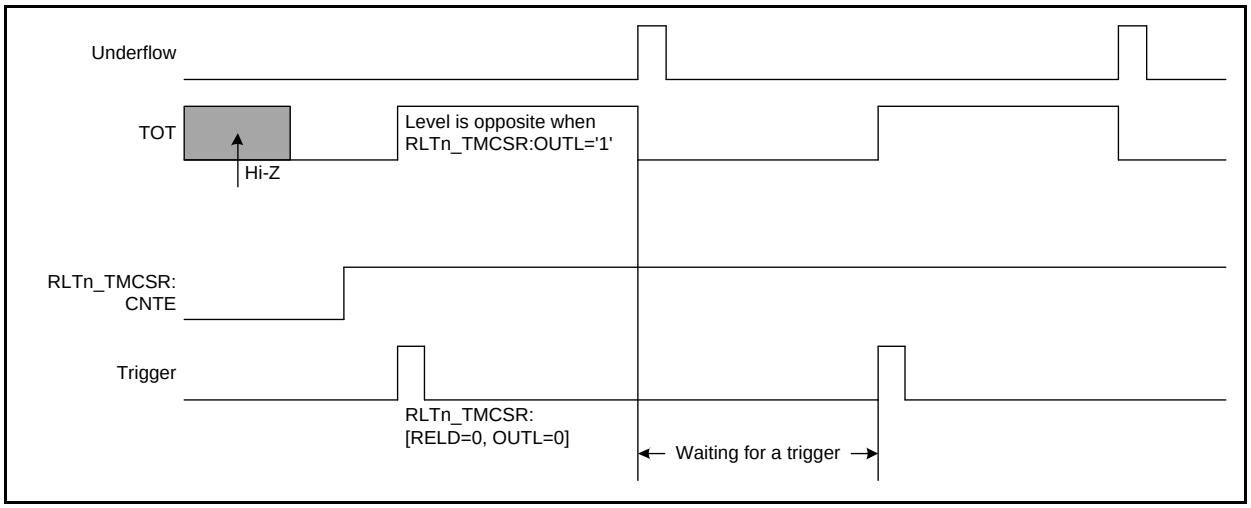


Figure 3-7 Output Signal function of 32-bit Reload Timer in One-Shot Mode



3.4. Counter Operation state

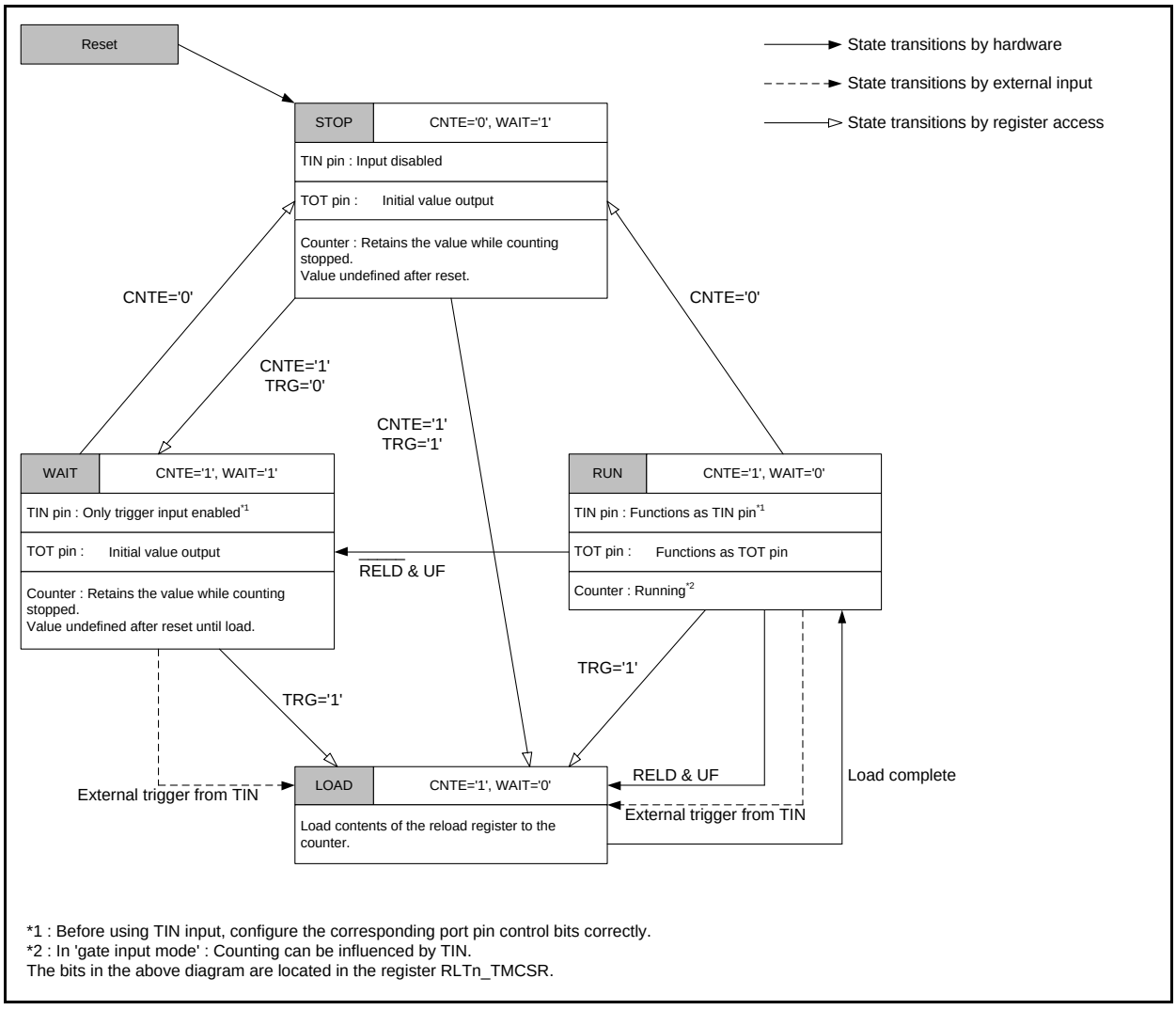
The counter state is determined by RLTn_TMCSR:CNTE bit in the Timer Control Status Register and the internal WAIT signal.

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.

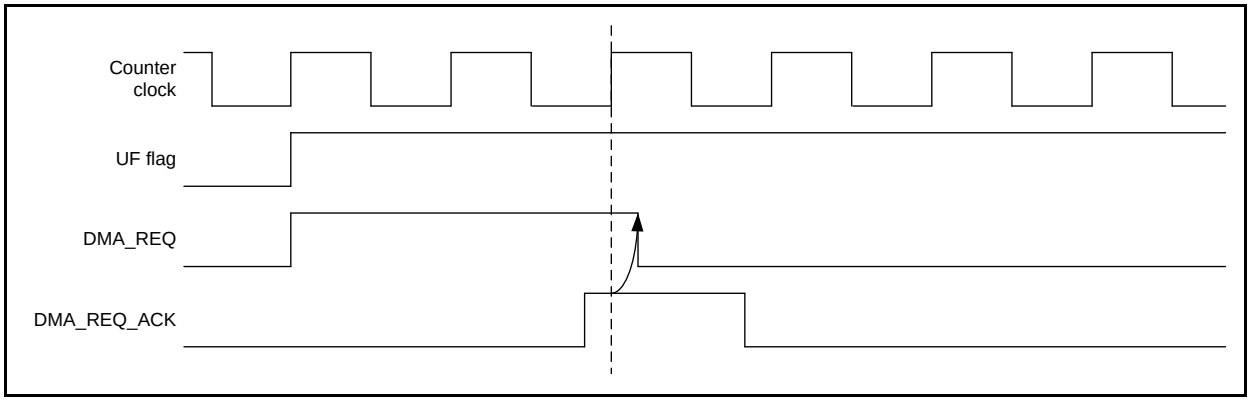
Enabling DMA support

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.

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	Reserved	Reserved	OUTL	RELD	INTE	Reserved	Reserved	Reserved
ACCESS_TYPE	R/W0	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 4-1 and Table 4-2 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 4-3 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 4-3 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 4-3 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] Reserved

Always write '0' to this bit.

[bit6] Reserved

[bit5] OUTL : Output Level

This Output Level (OUTL) bit sets the output level for the TOT.
Refer to Table 4-4.

[bit4] RELD : Reload

This Reload (RELD) bit enables reload operations.
Refer to Table 4-4.

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 4-1 RLTn_TMCSR:MOD[2:0] bit Settings for Internal Clock Mode (RLTn_TMCSR:CSL1 / RLTn_TMCSR:CSL2 = '00', '01', or '10')

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 4-2 RLTn_TMCSR:MOD[2:0] bit Settings for Event Counter Mode (RLTn_TMCSR:CSL1 / RLTn_TMCSR:CSL2 = '11')

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 4-3 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 4-4 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 45: 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-T01P00R01L02-E1-GBM01

1. Overview

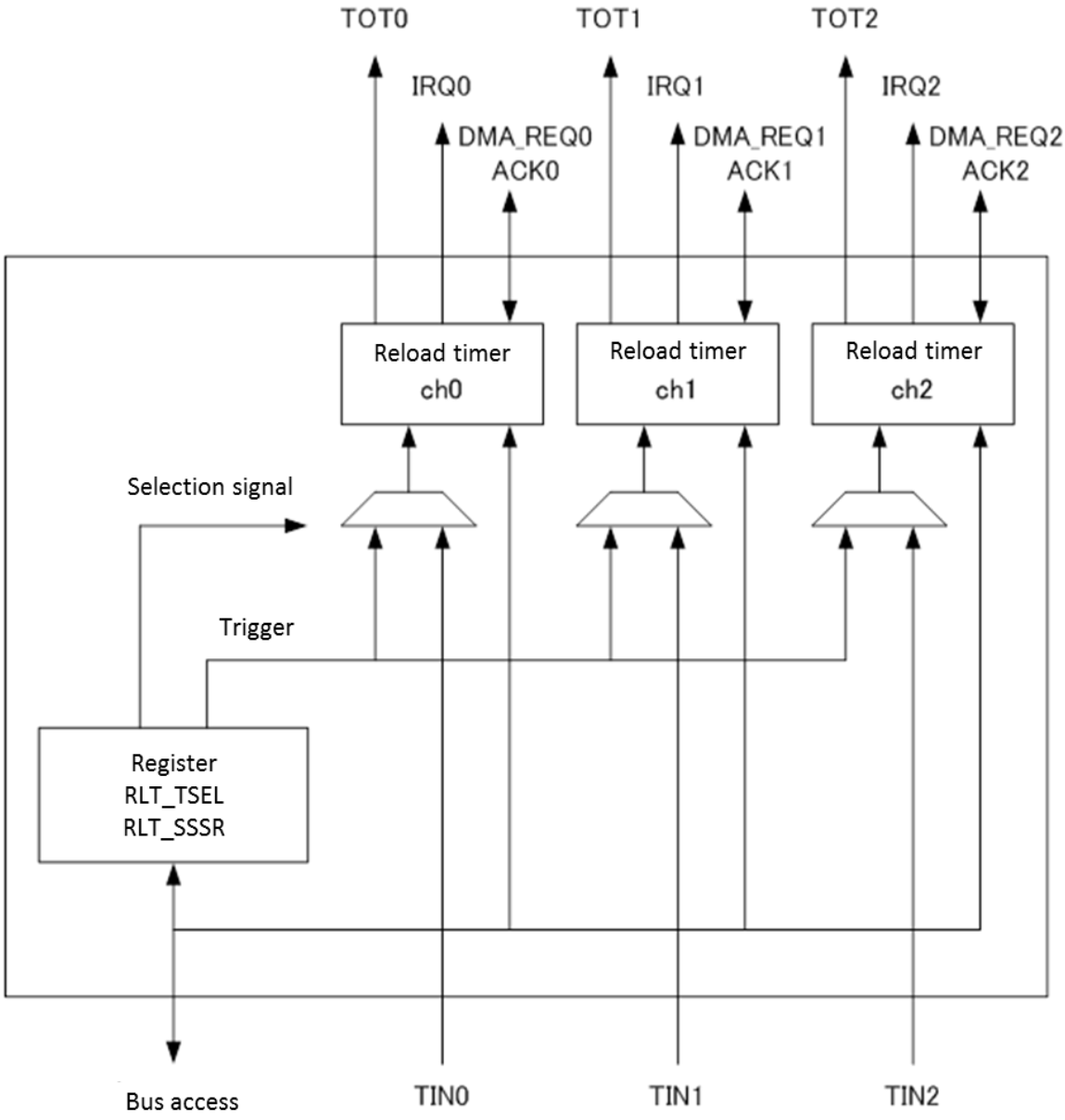
This section provides the overview of the reload timer simultaneous soft start.

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 RLT_TSEL register enables selections of the external trigger and software trigger for each channel.

Multiple channel simultaneous start

A write access to the RLT_SSSR 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 disenable 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 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 4-2 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 46: Resource-linked Operation

This chapter explains operation where the free-run timer is linked with input capture or output compare and cascade input of reload timer.

1. Overview
2. Configuration with Free-run Timer
3. Configuration of Free-run Timer Selector
4. Output Compare Control Configuration
5. Cascade Input of Reload Timer
6. Registers
7. Operation to Select Free-run Timer Output
8. Output Compare Control Register Settings
9. Cascade Input of Reload Timer Settings
10. Precautions on Operation Specifications

TLO-T01P00R01L01-E1-GBM00

1. Overview

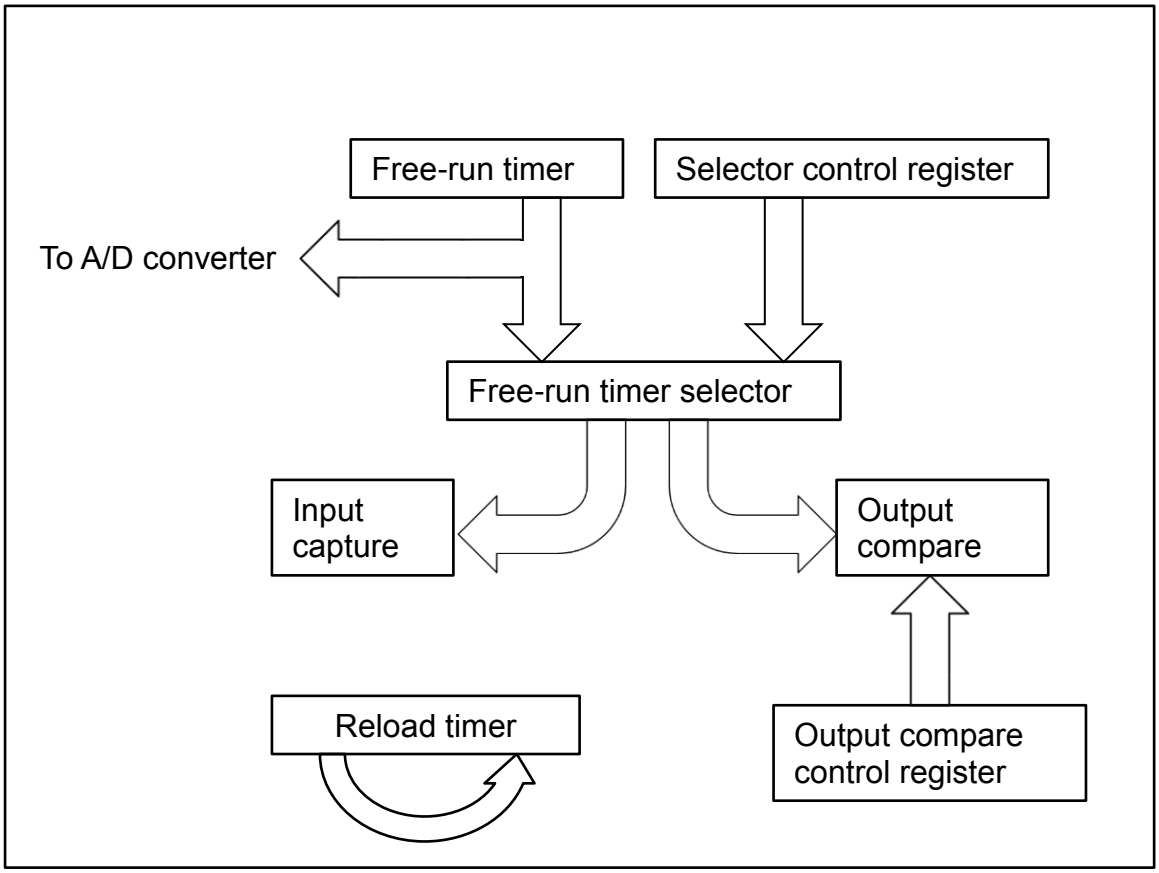
This section explains the configuration of a free-run timer used in combination with input capture, output compare, and a cascade input of reload timer.

This chapter explains the operation setting when a free-run timer is used in combination with input capture and output compare.

In addition, this chapter explains the output level invert mode function setting of the output compare, and cascade input of reload timer settings.

The following figure shows the configuration of the free-run timer, input capture, output compare, each control register, and the cascade input of reload timer.

Figure 1-1 Schematic Configuration Diagram



For the operation to connect to the A/D converter, see the "12-bit A/D Converter" specification.

2. Configuration with Free-Run Timer

This section explains the connection between a free-run timer and other peripherals.

The 32-bit free-run timer can be used in combination with an input capture and output compare.

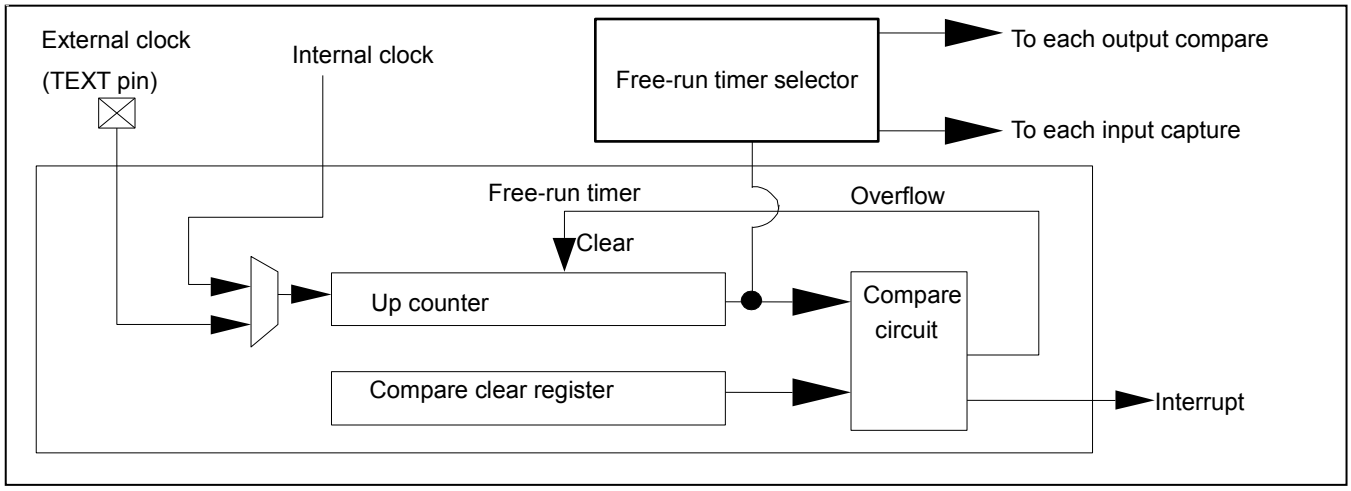
Function of the 32-bit free-run timer

The output value of the free-run timer counter can be used as a clock count of the input capture and the output compare.

Function of the free-run timer output selection selector

The output selection selector allows each free-run timer output to be selected for each channel of a 32-bit input capture and 32-bit output compare.

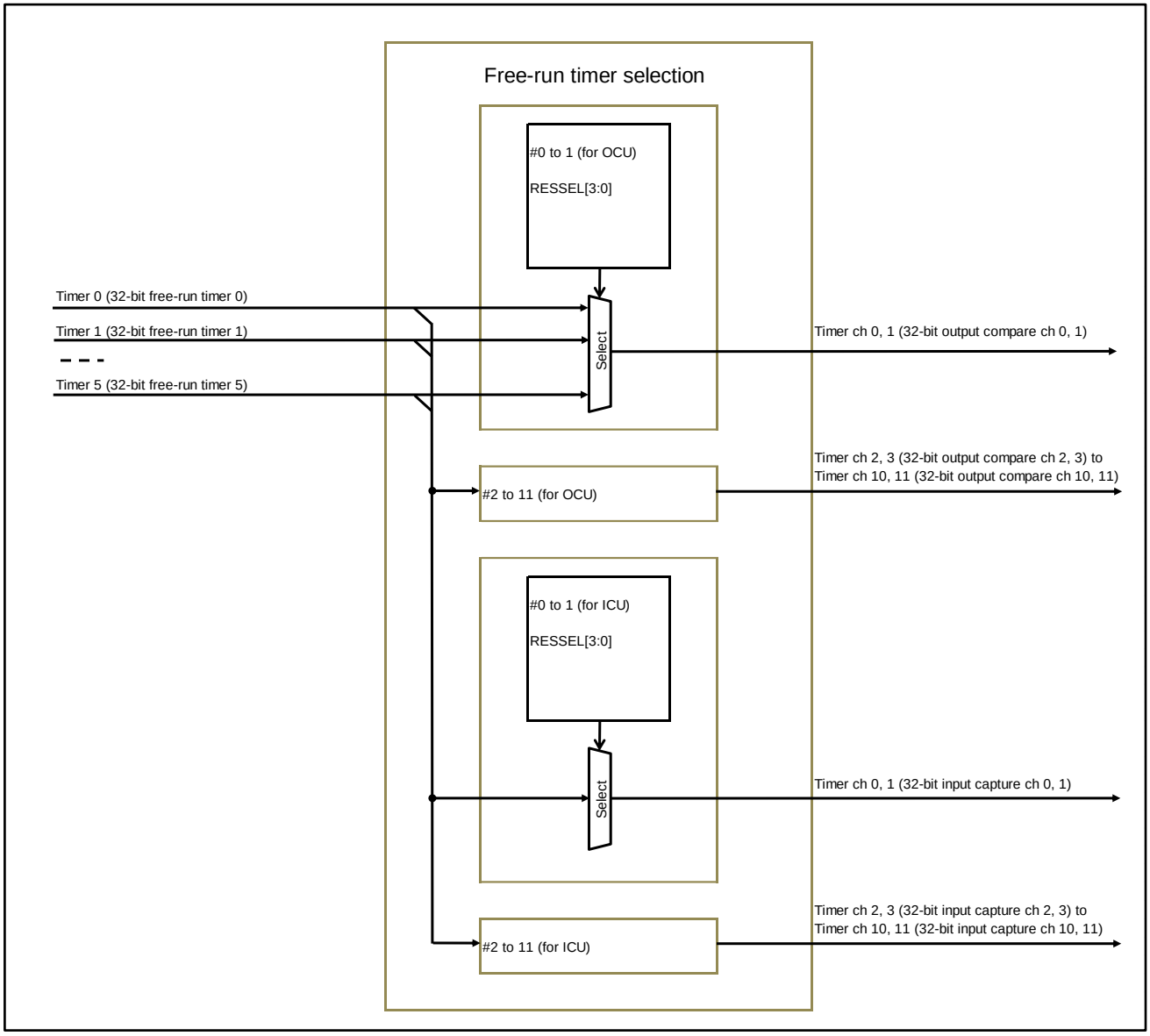
Figure 2-1 Block Diagram (Outline of Free-Run Timer for 1 Channel)



3. Configuration of Free-Run Timer Selector

This section explains a configuration diagram of the free-run timer selector.

Figure 3-1 Configuration Diagram of Free-Run Timer Selector



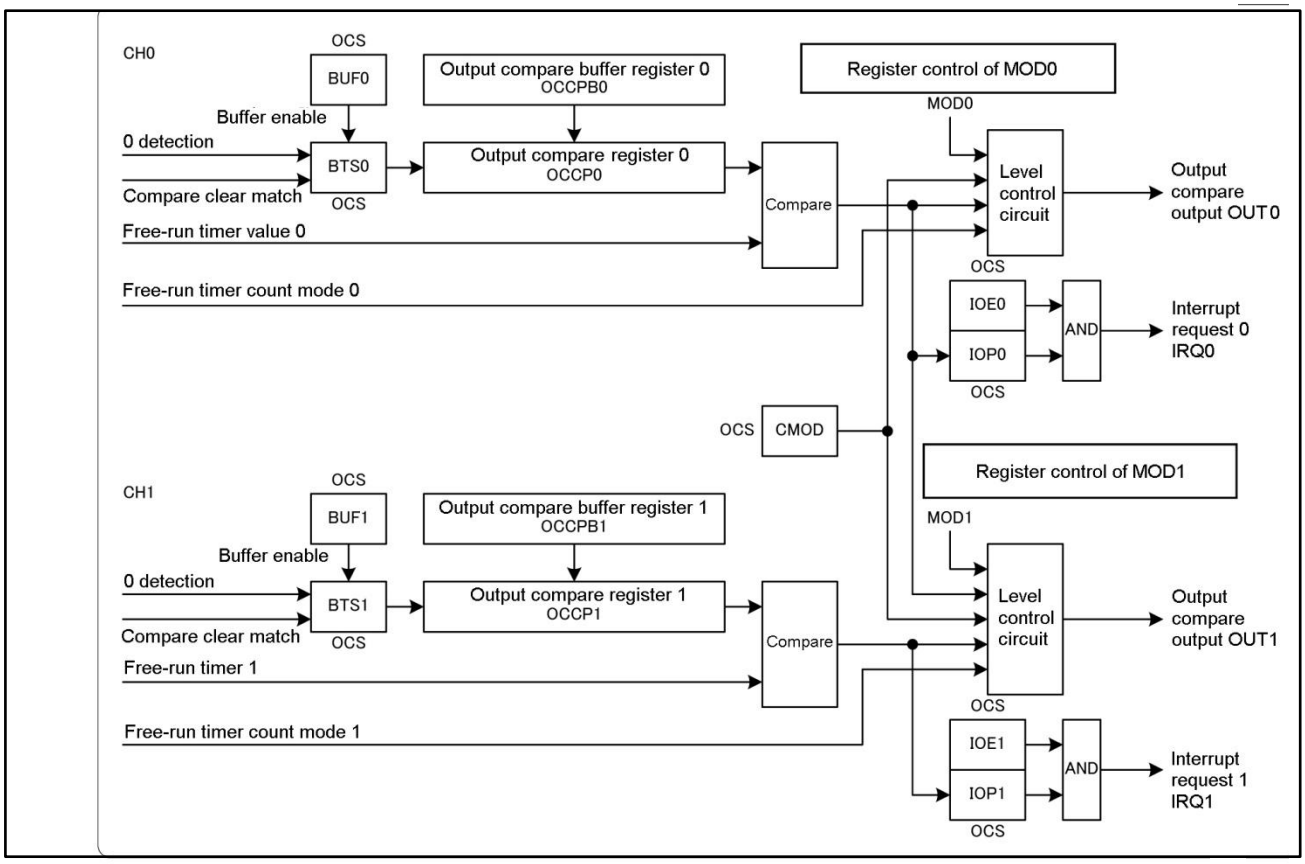
4. Output Compare Control Configuration

This section explains control related to the output compare setting.

To set the output compare, MOD0 and MOD1 of each channel are set in a register.

For details on the setting contents, see the output compare specifications.

Figure 4-1 Configuration Diagram Showing Control of the Output Level Invert Mode of an Output Compare



Note:

- *In the output compare specifications, MODn is described as an output pin. However, this model controls MODn in an internal register.*

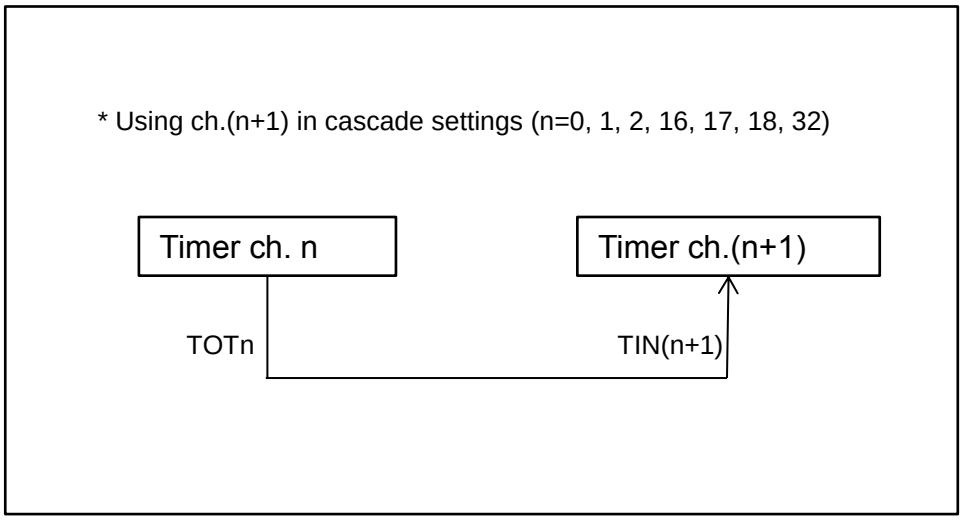
5. Cascade Input of Reload Timer

This section explains cascade input of reload timer.

When you select cascade input, you can use the timer's ch.n output (TOTn) for the input for ch.n+1 (TIN (n+1)).

For example, you can use the timer's ch.0 output (TOT0) for the input for ch.1 (TIN1), ch.1 output (TOT1) for the input for ch.2 (TIN2), and ch.2 output (TOT2) for the input for ch.3 (TIN3).

Figure 5-1 Timer Input/Output in Cascade Input Configuration



6. Registers

This section explains the registers that are used to control the free-run timer selector, output compare, and the cascade input of reload timer setting registers.

The free-run timer selector is set in the following registers.

Table 6-1 Free-Run Timer Selector Control Register (for ICU)

Register	Register Function
RIC_RESIN 108	FRT selection function register for ICU ch. 0, 1
RIC_RESIN 111	FRT selection function register for ICU ch. 2, 3
RIC_RESIN 114	FRT selection function register for ICU ch. 4, 5
RIC_RESIN 117	FRT selection function register for ICU ch. 6, 7
RIC_RESIN 120	FRT selection function register for ICU ch. 8, 9
RIC_RESIN 123	FRT selection function register for ICU ch. 10, 11

Table 6-2 Free-Run Timer Selector Control Register (for OCU)

Register	Register Function
RIC_RESIN 090	FRT selection function register for OCU ch. 0, 1
RIC_RESIN 093	FRT selection function register for OCU ch. 2, 3
RIC_RESIN 096	FRT selection function register for OCU ch. 4, 5
RIC_RESIN 099	FRT selection function register for OCU ch. 6, 7
RIC_RESIN 102	FRT selection function register for OCU ch. 8, 9
RIC_RESIN 105	FRT selection function register for OCU ch. 10, 11

Note:

- *For details on the registers, see "4.12 Resource Input Setting Register (RIC_RESINn)" in "I/O PORT."*

Control of the output level invert mode function setting of an output compare is set in the following registers.

Table 6-3 Output Compare Output Level Invert Mode Function Setting Control Registers (for OCU)

Register	Register Function
RIC_RESIN 091	OCU ch. 0 MDO0 setting
RIC_RESIN 092	OCU ch. 1 MDO1 setting
RIC_RESIN 094	OCU ch. 2 MDO0 setting
RIC_RESIN 095	OCU ch. 3 MDO0 setting
RIC_RESIN 097	OCU ch. 4 MDO0 setting
RIC_RESIN 098	OCU ch. 5 MDO0 setting
RIC_RESIN 100	OCU ch. 6 MDO0 setting
RIC_RESIN 101	OCU ch. 7 MDO0 setting
RIC_RESIN 103	OCU ch. 8 MDO0 setting
RIC_RESIN 104	OCU ch. 9 MDO0 setting
RIC_RESIN 106	OCU ch. 10 MDO0 setting
RIC_RESIN 107	OCU ch. 11 MDO0 setting

Note:

- *For details on the registers, see "4.12 Resource Input Setting Register (RIC_RESINn)" in "I/O PORT."*

Cascade input of reload timer Settings are set in the following registers.

Table 6-4 Cascade Input of Reload Timer Setting Registers

Register	Register Function
RIC_RESIN 207	RLT ch. 0, ch. 1 cascade input setting
RIC_RESIN 208	RLT ch. 1, ch. 2 cascade input setting
RIC_RESIN 209	RLT ch. 2, ch. 3 cascade input setting
RIC_RESIN 211	RLT ch. 16, ch. 17 cascade input setting
RIC_RESIN 212	RLT ch. 17, ch. 18 cascade input setting
RIC_RESIN 213	RLT ch. 18, ch. 19 cascade input setting
RIC_RESIN 215	RLT ch. 32, ch. 33 cascade input setting

Note:

- *For details on the registers, see "4.12 Resource Input Setting Register (RIC_RESINn)" in "I/O PORT."*

7. Operation to Select Free-run Timer Output

This section explains the free-run timer selector selection setting.

The free-run timer output to each input capture and output compare is selected in the resource selection bits ([bit3:0] RESSEL) in the resource input setting register (RIC_RESINn).

7.1. Selection Setting with Input Capture

By setting the resource input setting register of an address assigned to each input capture, a free-run timer can be selected for each channel.

Table 7-1 Free-Run Timer and Input Capture Connection Setting

Reg Add	Resource	RESSEL Data[3:0]						
		0000	0001	0010	0011	0100	0101	Data Value Other Than Those Listed at Left
RIC_RESIN 108	ICU0, ICU1	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	0 is input.
RIC_RESIN 111	ICU2, ICU3	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	0 is input.
RIC_RESIN 114	ICU4, ICU5	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	0 is input.
RIC_RESIN 117	ICU6, ICU7	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	0 is input.
RIC_RESIN 120	ICU8, ICU9	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	0 is input.
RIC_RESIN 123	ICU10, ICU11	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	0 is input.

The following shows the detailed settings of the resource selection bits.

[bit3:0] RESSEL: Resource selection bits

These bits select an input from each FRT to the ICU [n, n+1] ch. (n = 0, 2, 4, 6, 8, 10)

Bit3:0				Description
0	0	0	0	FRT0 output is input to the ICU [n, n+1] ch.
0	0	0	1	FRT1 output is input to the ICU [n, n+1] ch.
0	0	1	0	FRT2 output is input to the ICU [n, n+1] ch.
0	0	1	1	FRT3 output is input to the ICU [n, n+1] ch.
0	1	0	0	FRT4 output is input to the ICU [n, n+1] ch.
0	1	0	1	FRT5 output is input to the ICU [n, n+1] ch.
Setting values other than those listed above				ALL "0's" are input to the ICU [n, n+1] ch.

Notes:

- *The ICU that is connected to each FRT is connected per each 2 ch.*
- *In the table above, if a resource selection setting value is "0", then the FRT0 is connected to ICU ch. 0 and ch. 1.*

7.2. Selection Setting with Output Compare

By setting the resource input setting register of an address assigned to each output compare, a free-run timer can be selected for each channel.

Table 7-2 Free-Run Timer and Output Compare Connection Setting Table

Reg	Resource	RESSEL Data[3:0]						
		0000	0001	0010	0011	0100	0101	Data Value Other Than Those Listed at Left
RIC_RESIN 090	OCU0, OCU1	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	0 is input.
RIC_RESIN 093	OCU2, OCU3	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	0 is input.
RIC_RESIN 096	OCU4, OCU5	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	0 is input.
RIC_RESIN 099	OCU6, OCU7	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	0 is input.
RIC_RESIN 102	OCU8, OCU9	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	0 is input.
RIC_RESIN 105	OCU10, OCU11	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	0 is input.

The following shows the detailed settings of the resource selection bits.

[bit3:0] RESSEL: Resource selection bits

These bits select an input from each FRT to the OCU [n, n+1] ch. (n = 0, 2, 4, 6, 8, 10)

Bit3:0				Description
0	0	0	0	FRT0 output is input to the OCU [n, n+1] ch.
0	0	0	1	FRT5 output is input to the OCU [n, n+1] ch.
0	0	1	0	FRT4 output is input to the OCU [n, n+1] ch.
0	0	1	1	FRT3 output is input to the OCU [n, n+1] ch.
0	1	0	0	FRT2 output is input to the OCU [n, n+1] ch.
0	1	0	1	FRT1 output is input to the OCU [n, n+1] ch.
Setting values other than those listed above				ALL "0's" are input to the OCU [n, n+1] ch.

Notes:

- *The OCU that is connected to each FRT is connected per each 2 ch.*
- *In the table above, if a resource selection setting value is "0", then the FRT0 is connected to OCU ch. 0 and ch.1.*

8. Output Compare Control Register Settings

This section explains the output operation setting of an output compare.

In the output level invert mode function setting of the compare control register (OCS) settings, the MODn (n = 0, 1) signal control is set in the following registers. For details on operation, see "5.2 Compare Control Register (OCS)" in the Chapter 32-BIT OUTPUT COMPARE. The following shows the register addresses and settings of each output compare.

Table 8-1 Output Compare MODn (n = 0, 1) Setting Table

Reg	MOD (n)	Resource	RESSEL Data[3:0]		
			0000	0001	Data Value Other Than Those Listed at Left
RIC_RESIN 091	0	OCU ch.0	"0" is set.	"1" is set.	"0" is set.
RIC_RESIN 092	1	OCU ch.1	"0" is set.	"1" is set.	"0" is set.
RIC_RESIN 094	0	OCU ch.2	"0" is set.	"1" is set.	"0" is set.
RIC_RESIN 095	1	OCU ch.3	"0" is set.	"1" is set.	"0" is set.
RIC_RESIN 097	0	OCU ch.4	"0" is set.	"1" is set.	"0" is set.
RIC_RESIN 098	1	OCU ch.5	"0" is set.	"1" is set.	"0" is set.
RIC_RESIN 100	0	OCU ch.6	"0" is set.	"1" is set.	"0" is set.
RIC_RESIN 101	1	OCU ch.7	"0" is set.	"1" is set.	"0" is set.
RIC_RESIN 103	0	OCU ch.8	"0" is set.	"1" is set.	"0" is set.
RIC_RESIN 104	1	OCU ch.9	"0" is set.	"1" is set.	"0" is set.
RIC_RESIN 106	0	OCU ch.10	"0" is set.	"1" is set.	"0" is set.
RIC_RESIN 107	1	OCU ch.11	"0" is set.	"1" is set.	"0" is set.

The following shows the detailed settings of the resource selection bits.

OCU [n].ch OCU channel (n = 0, 2, 4, 6, 8, 10)

[bit3:0] RESSEL: Resource selection bits These bits Set MOD0.

Bit3:0				Description
0	0	0	0	MOD0 is set to "0".
0	0	0	1	MOD0 is set to "1".
Setting values other than those listed above				MOD0 is set to "0".

OCU [n+1] ch OCU channel (n = 0, 2, 4, 6, 8, 10)

[bit3:0] RESSEL: Resource selection bits These bits Set MOD1.

Bit3:0				Description
0	0	0	0	MOD0 is set to "0".
0	0	0	1	MOD0 is set to "1".
Setting values other than those listed above				MOD0 is set to "0".

9. Cascade Input of Reload Timer Settings

This section explains cascade input of reload timer settings.

The cascade input of reload timer is selected in the resource selection bits ([bit3:0] RESSEL) in the resource input setting register (RIC_RESINn).

By setting the resource input setting register of an address assigned to each reload timer, a cascade input can be selected for each channel.

Table 9-1 Cascade Input of Reload Timer Setting Table

Reg	Resource	RESSEL Data[3:0]		
		0000	0001	Data Value Other Than Those Listed at Left
RIC_RESIN 207	RLT1(TIN1)	External pin	RLT0(TOT0)	0 is input.
RIC_RESIN 208	RLT2(TIN2)	External pin	RLT1(TOT1)	0 is input.
RIC_RESIN 209	RLT3(TIN3)	External pin	RLT2(TOT2)	0 is input.
RIC_RESIN 211	RLT17(TIN17)	External pin	RLT16(TOT16)	0 is input.
RIC_RESIN 212	RLT18(TIN18)	External pin	RLT17(TOT17)	0 is input.
RIC_RESIN 213	RLT19(TIN19)	External pin	RLT18(TOT18)	0 is input.
RIC_RESIN 215	RLT33(TIN33)	External pin	RLT32(TOT32)	0 is input.

The following shows the detailed settings of the resource selection bits.

[bit3:0] RESSEL: Resource selection bits

These bits select an input from each RLT[n] to the RLT [n+1] ch. (n = 0, 1, 2, 16, 17, 18, 32)

Bit3:0				Description
0	0	0	0	External pin input is input to the TIN of RLT.
0	0	0	1	TOT of [RLT] output is input to the TIN of RLT [n+1].
Setting values other than those listed above				"0" is input to the TIN of RLT.

10. Precautions on Operation Specifications

This section explains precautions on operation.

10.1. Precautions on Using the Free-Run Timer Selector

This section explains the precautions on the output control operation of the free-run timer selector.

Be sure to perform the selection setting while the free-run timer is stopped.

10.2. Precautions on Using OCU

This section explains the precautions on the output operation setting of the output compare.

Be sure to perform the selection setting while the output compare is stopped.

10.3. Precautions on Using RLT

This section explains the precautions on the cascade input setting of reload timer.

Be sure to perform the selection setting while the reload timer is stopped.

CHAPTER 47: Quad Position & Revolution Counter

This chapter explains the functions and operations of the Quad Position & Revolution Counter (QPRC).

1. Overview
2. Configuration
3. Operations
4. Registers

QPRC-T01P00R01L00-E1-GBM00

1. Overview

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

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.

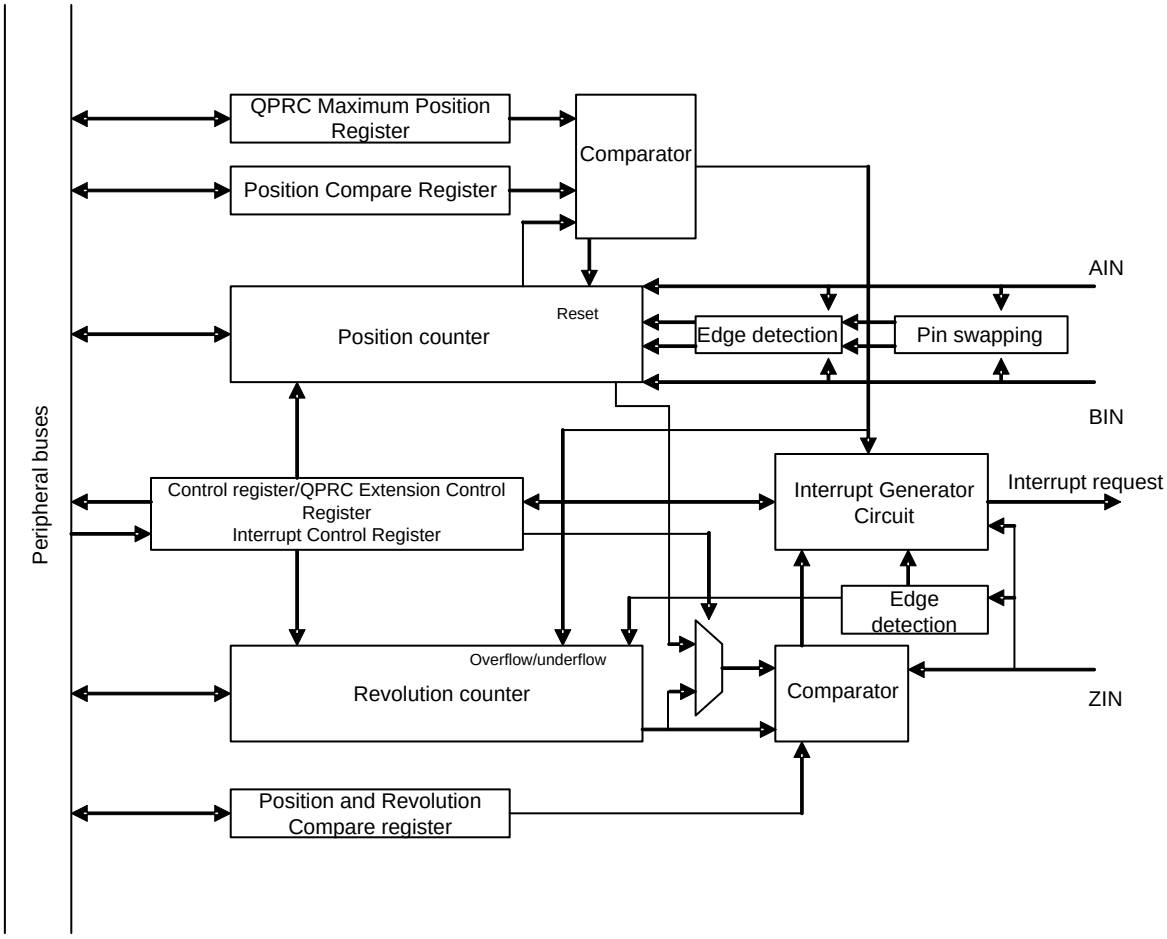
The following useful functions are provided for counting

- 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

2. Configuration

The following shows the configuration of Quad Position & Revolution Counter.

Figure 2-1 Block Diagram of Quad Position & Revolution Counter



3. Operations

This section explains the operation of Quad Position & Revolution Counter.

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 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

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.

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]="00"	Position counter disable	Position counter disable
Up/down counting PC_Mode1: QCR:PCM[1:0]="01"	AIN Active edge	BIN Active edge
Phase difference count PC_Mode2:QCR:PCM[1:0]="10"	AIN Active edge or high/low level	High/low level or BIN active edge
Counting with direction PC_Mode3:QCR:PCM[1:0]="11"	High/low level	BIN Active edge

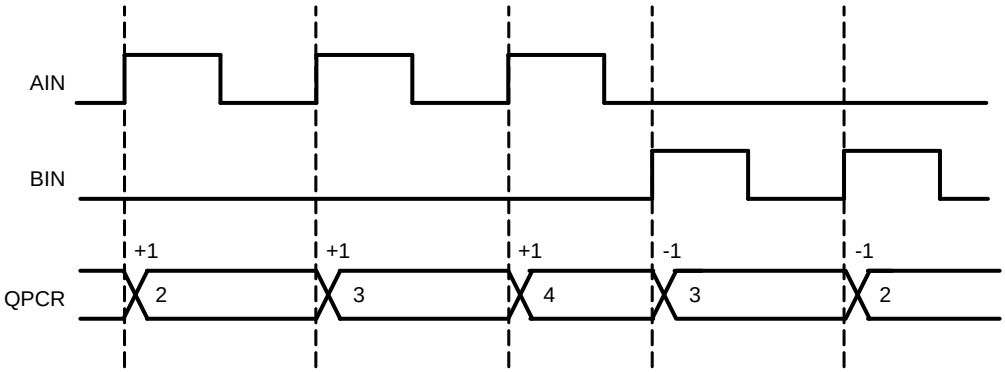
Note:

- *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]="01" or "10" or "11") or by the BIN Detection Edge Select bits (QCR:BES[1:0]="01" or "10" or "11").*

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]="10", QCR:BES[1:0]="10", 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	AES[1:0] Setting	BES[1:0] Setting
1-time frequency multiplication mode	01	00
	10	00
	00	01
	00	10
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]="00", QCR:BES[1:0]="11", QCR:SWAP="0")

Edge Detection Pin	Detection Edge	Level Check Pin	Input Level	Counting Direction	Figure 3-2 Timing
BIN	Rising edge	AIN	High	Up	(1)
	Rising edge		Low	Down	(2)
	Falling edge		High	Down	(3)
	Falling edge		Low	Up	(4)

Figure 3-2 Operation in 2-Time Frequency Multiplication Mode (QCR:AES[1:0]="00", QCR:BES[1:0]="11", QCR:SWAP="0")

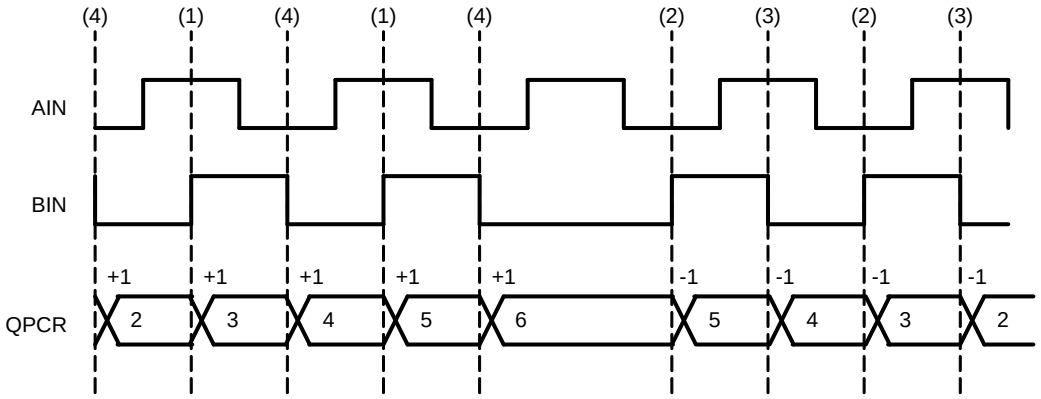
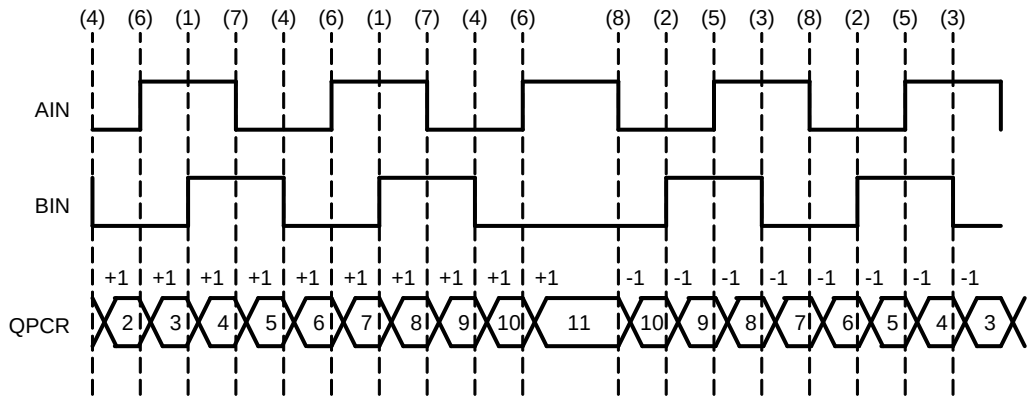


Table 3-4 Counting in 4-Time Frequency Multiplication Mode (QCR:AES[1:0]="11", QCR:BES[1:0]="11")

Edge Detection Pin	Detection Edge	Level Check Pin	Input Level	Counting Direction	Figure 3-3 Timing
BIN	Rising edge	AIN	High	Up	(1)
	Rising edge		Low	Down	(2)
	Falling edge		High	Down	(3)
	Falling edge		Low	Up	(4)
AIN	Rising edge	BIN	High	Down	(5)
	Rising edge		Low	Up	(6)
	Falling edge		High	Up	(7)
	Falling edge		Low	Down	(8)

Figure 3-3 Operation in 4-Time Frequency Multiplication Mode (QCR:AES[1:0]="11", QCR:BES[1:0]="11", QCR:SWAP="0")



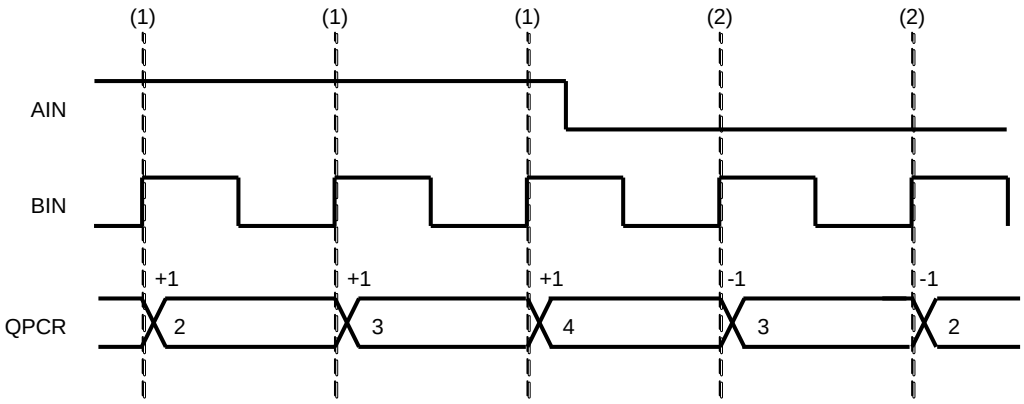
PC_Mode3: Count Mode with Direction

- 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	High	Up	(1)
	Active edge		Low	Down	(2)

Figure 3-4 Operation in the direction Control Counting Mode (QCR:AES[1:0]="00", QCR:BES[1:0]="10", QCR:SWAP="0")



Operation of revolution counter

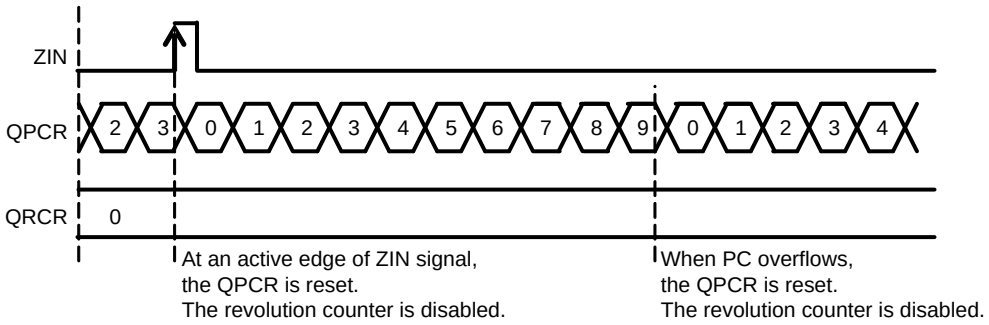
When the revolution counter receives an input from the ZIN pin (having the counter clear function) or an output of position counter (underflow or overflow), it is counted up or down. A rising edge, a falling edge, or both can be set as the active edge of ZIN signal.

The counting conditions of revolution counter depend on the selected mode as follows.

RC_Mode0 (QCR:RCM[1:0]="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.

Figure 3-5 RC_Mode0 operation (QPRC Maximum Position Register QMPR=9, QCR:CGSC="0")



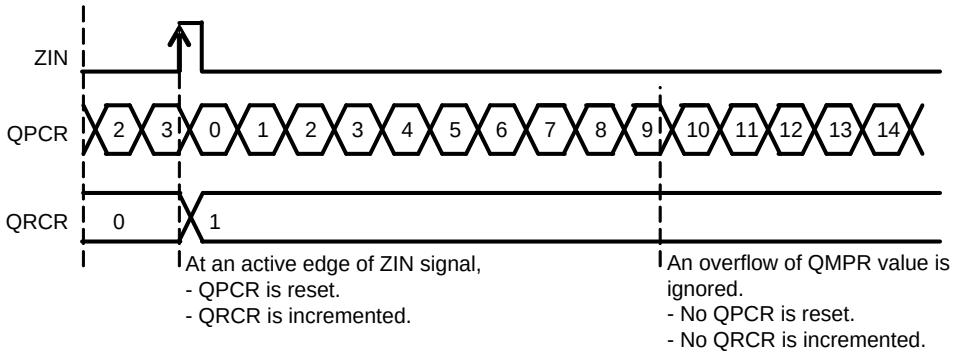
QPCR: QPRC Position Count Register

QRCR: QPRC Revolution Count Register

RC_Mode1 (QCR:RCM[1:0]="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).
- 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 logical 1.

Figure 3-6 RC_Mode1 operation (QPRC Maximum Position Register QMPR=9, QCR:CGSC="0")



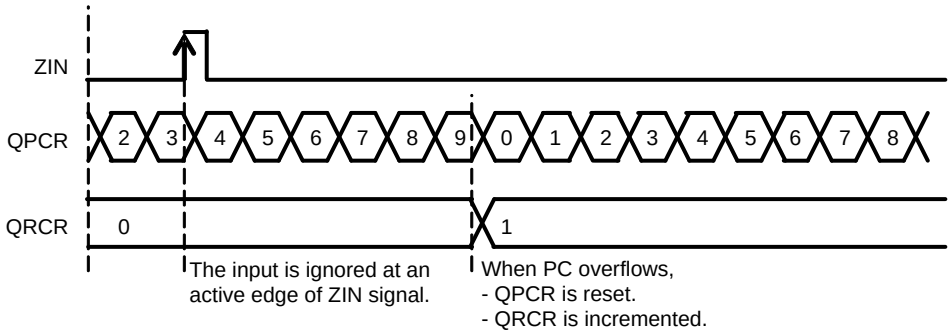
Notes:

- *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]="10"):

- 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.

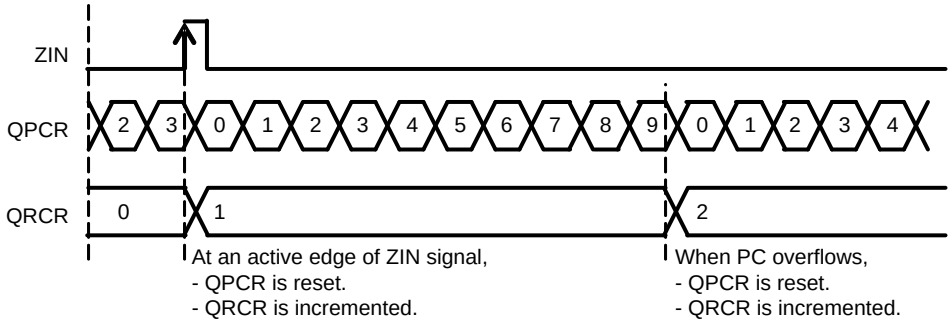
Figure 3-7 RC_Mode2 operation (QPRC Maximum Position Register QMPR=9)



RC_Mode3 (QCR:RCM[1:0]="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.
- 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).*

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).

Quad Position & Revolution Counter interrupts

The following table defines the conditions where an interrupt request of Quad Position & Revolution Counter can generate.

Table 3-6 Generation conditions of Quad 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"	QICR:CDCF is set to "0".
Zero index interrupt request	QICR: ZIIF="1"	QICR: OUZIE="1"	QICR:ZIIF is set to "0".
Overflow interrupt request	QICR: OFDF="1"		QICR:OFDF is set to "0".
Underflow interrupt request	QICR: UFDF="1"		QICR:UFDF is set to "0".
PC and RC match interrupt request	QICR: QPRCMF="1"	QICR: QPRCMIE="1"	QICR.QPRCMF is set to "0".
PC match interrupt request	QICR: QPCMF="1"	QICR: QPCMIE="1"	QICR:QPCMF is set to "0".
PC match and RC match interrupt request	QICR: QPCNRCMF="1"	QICR: QPCNRCMIE="1"	QICR:QPCNRCMF is set to "0".
Outrange interrupt request	QECR: ORNGF="1"	QICR: ORNGIE="1"	QECR:QRNGF is set to "0".

QICR: QPRC Interrupt Control Register

QECR: QPRC Extension Control Register

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]="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").

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.

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 (QICRL:UFDF) is set to logical 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.*

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).

The position counter reset mask function is executed in the following sequence.

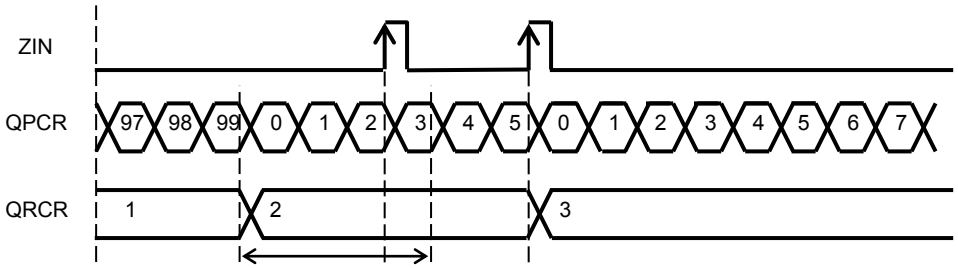
1. When an active event of ZIN signal, an overflow of position counter, or an underflow of position counter are detected, a value being set by the position counter reset mask bits (QCR:PCRM[1:0]) is set to the mask counter (*1).
2. When the position counter is counted up or down in the same counting 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]="11").

Example 1:
An active edge of ZIN signal is ignored for four (4) counts(QPRC=0-3) 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]="10", QCR:CGSC="0")

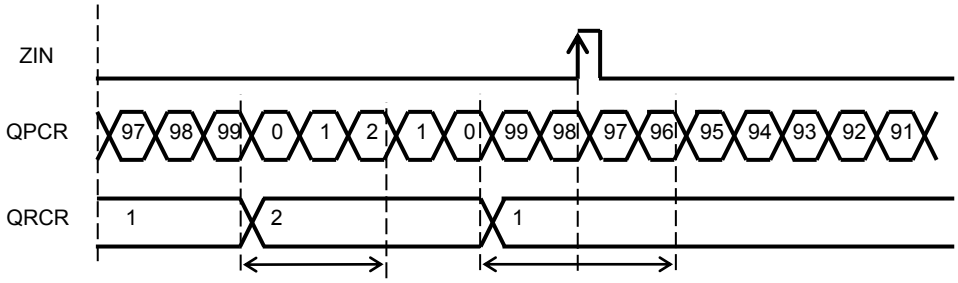


The reset events are ignored for 4 counts after PC overflow

At an active edge of ZIN signal,
- QPCR is reset.
- QRCRis incremented..

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")



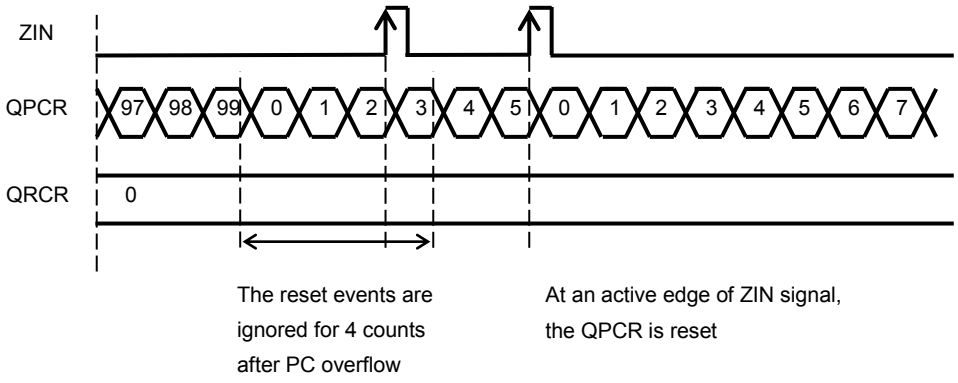
The reset events are ignored after PC overflow. The Mask function is canceled as PC counting was inverted.

The reset events are ignored for 4 counts after PC underflow.

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")



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".*

4. Registers

This section explains the configuration and functions of the registers used for the Quad Position & Revolution Counter (QPRC).

List of Quad Position & Revolution Counter Registers

Abbreviation	Register Name	See
QPCR	Quad Position & Revolution Counter 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. Quad Position & Revolution Counter Position Count Register (QPCR)

The Quad Position & Revolution Counter 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:

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").
- 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.

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.

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:

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").

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

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:

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:

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	RCM1	RCM0	PCM1	PCM0
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] 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]="00").*

[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]="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.

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] 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".

bit3	Bit2	Description
0	0	Disables the revolution counter (RC_Mode0).
0	1	The revolution counter is counted up or down only with a ZIN active edge (RC_Mode1).
1	0	The revolution counter is counted up or down only when overflow or underflow is detected in the position counter that matches QMPR (RC_Mode2).
1	1	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] PCM1, PCM0: Position Counter Mode Bits

These bits are used to select the count mode of the position counter.

Bit1	Bit0	Description
0	0	Disables the position counter (PC_Mode0) to stop it.
0	1	Up-down count mode (PC_Mode1) Increments the value with an AIN active edge and decrements it with a BIN active edge.
1	0	Phase difference count mode (PC_Mode2) Counts up if AIN is leading BIN and down if BIN is leading AIN.
1	1	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]="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

High-Order Bytes of QPRC Control Register (QCRH)

bit	15	14	13	12	11	10	9	8
Field	CGE1	CGE0	BES1	BES0	AES1	AES0	PCRM1	PCRM0
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

[bit15:14] 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").

Bit15	Bit14	ZIN Used for Counter Clear Function (CGSC="0")	ZIN Used for Gate Function (CGSC="1")
0	0	Disables edge detection.	Disables level detection.
0	1	Detects a falling edge.	Detects level "L".
1	0	Detects a rising edge.	Detects level "H".
1	1	Detects a rising or falling edge.	Disables level detection.

[bit13:12] BES1, BES0: BIN Detection Edge Selection Bits

These bits are used to select the detection edge of the BIN external pin.

Bit13	Bit12	Description
0	0	Disables edge detection.
0	1	Detects a falling edge.
1	0	Detects a rising edge.
1	1	Detects rising and falling edges.

[bit11:10] AES1, AES0: AIN Detection Edge Selection Bits

These bits are used to select the detection edge of the AIN external pin.

Bit11	Bit10	Description
0	0	Disables edge detection.
0	1	Detects a falling edge.
1	0	Detects a rising edge.
1	1	Detects rising and falling edges.

[bit9:8] 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.

Bit9	Bit8	Description
0	0	No reset mask
0	1	The position counter reset or a revolution counter count-up or -down events are ignored until the position counter changes twice.
1	0	The position counter reset or a revolution counter count-up or -down events are ignored until the position counter changes four times.
1	1	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]="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).*
- *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 Bit

[bit2] 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").

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 only clear 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 only clear 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]="10").*

[bit6] OFDF: Overflow Interrupt Request Flag Bit

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.

"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 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.

"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 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").

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 only clear 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]="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").

- *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").

[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 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").

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 only clear 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]="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").

[bit0] 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").

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	R0,W0	R,W	R/W	R,WX	R,WX	R,W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

[bit15:14] Reserved

[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 only clear 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]="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").
  - 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").
  - 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 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").

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]="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

[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 only clear 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]="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".*

[bit8] 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").

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:

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").

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-T00P00R01L03-E1-GBB00

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

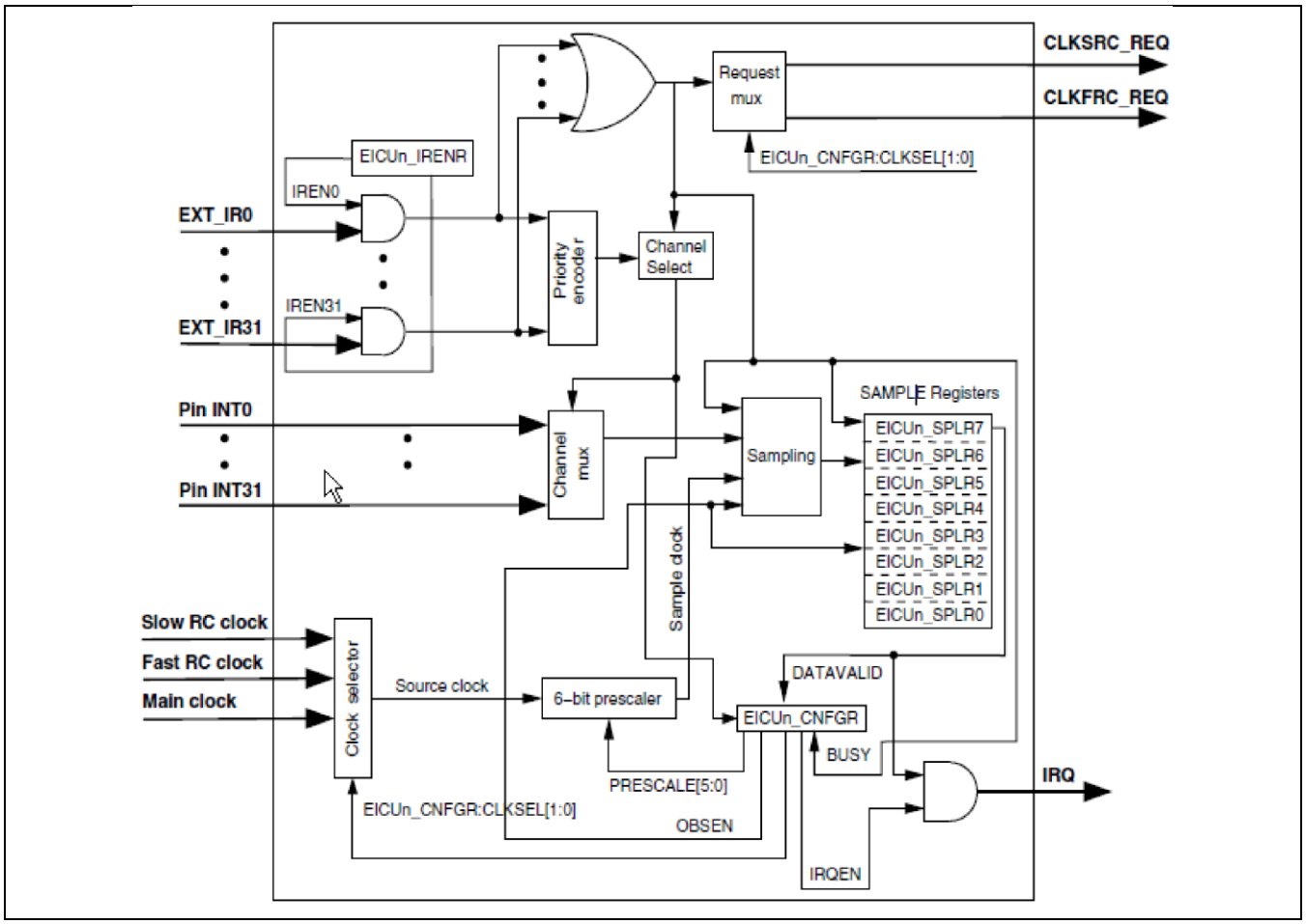
This module does not generate any DMA requests.

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 RC clock. If the EICUn_CNFGR:CLKSEL[1:0] is programmed for Main clock, it is assumed that the clocks are already available.

Note:

- *During power saving modes (which disables the Main clock), only the RC 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 RC clock/Fast RC 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 RC clock must be started.*

Clock Settings

This block selects between the following three source clocks based on EICUn_CNFGR:CLKSEL[1:0] programming:

- Slow RC clock
- Fast RC clock and
- Main 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.

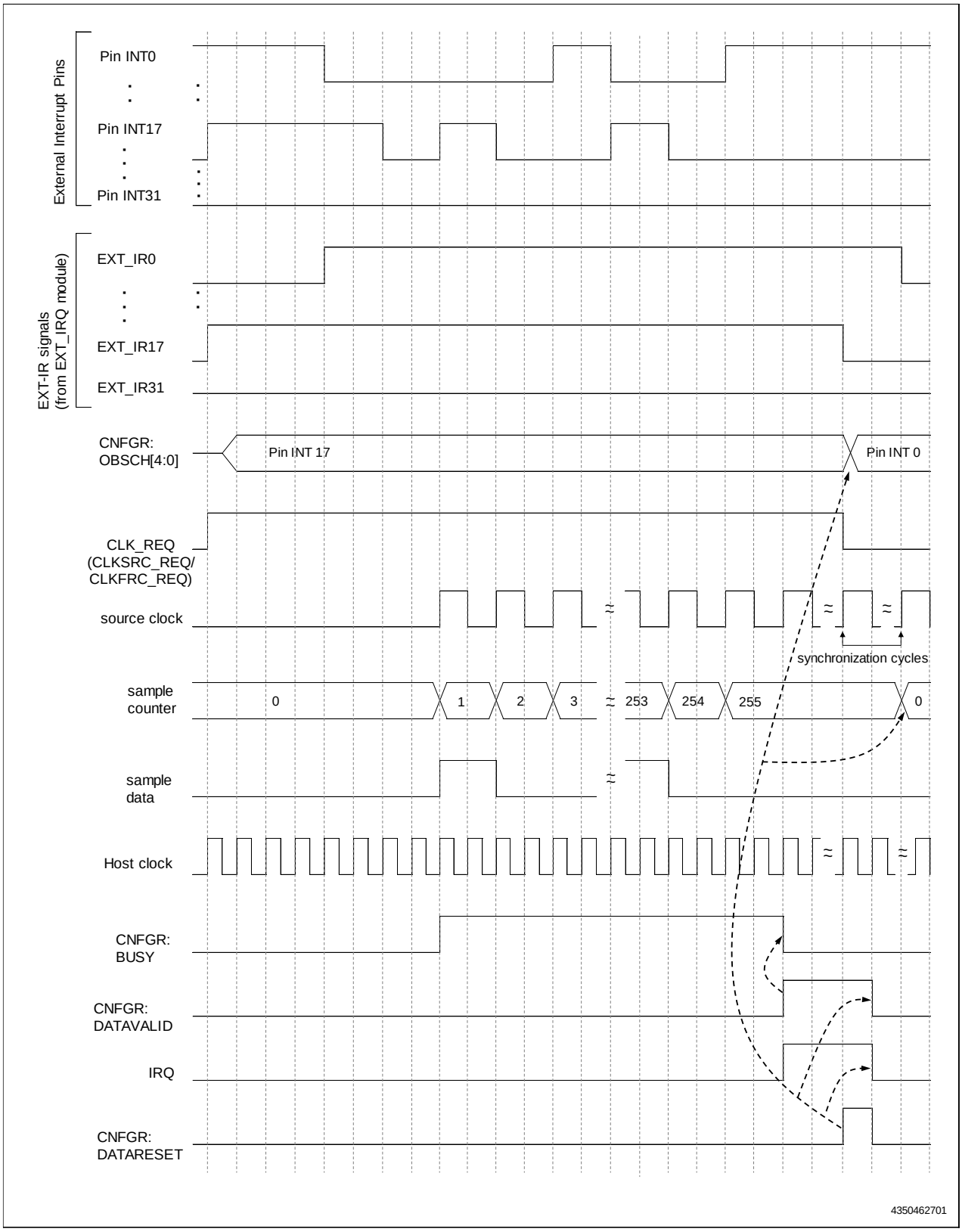
Example Operation of EICU

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 RC/Fast RC 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

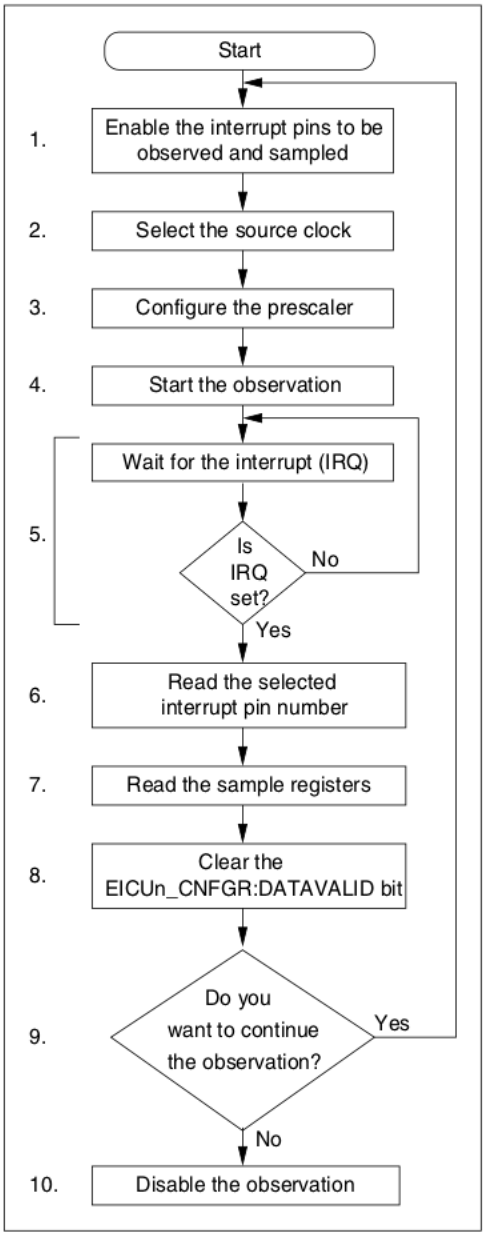


4. Setting Procedure Examples

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 RC clock, Fast RC clock, Main clock. Note that at reset the Slow RC 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 RC clock or Fast RC clock. After the arrival of the source clock (refer to the datasheet for the time required for the Slow RC and Fast RC 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 RC clock is operable (refer to the datasheet for time required for Slow RC and Fast RC 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 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

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	00000						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": Reserved

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. **Operations**
3. **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

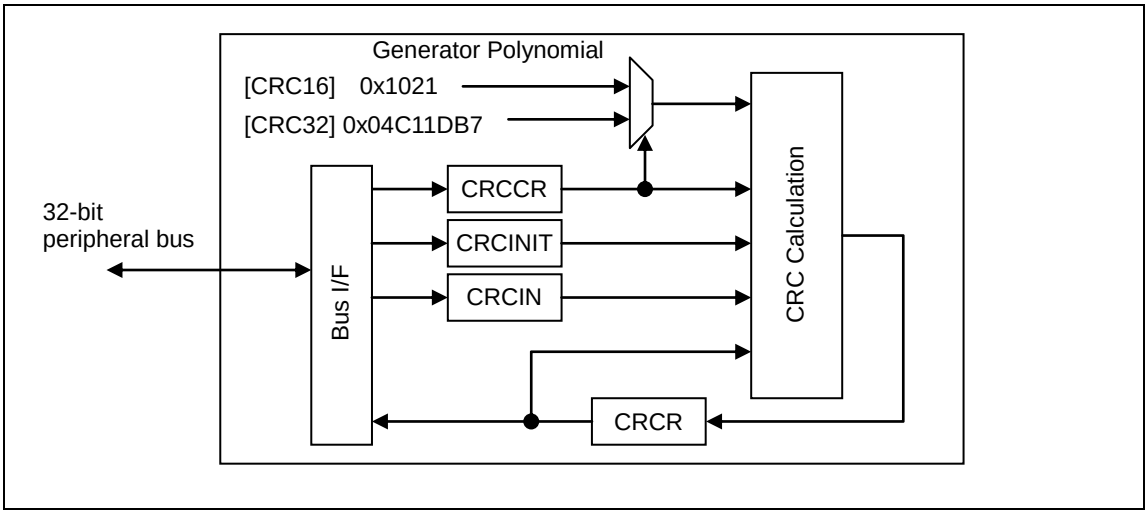
This module enables the calculation in both CCITT CRC16 and IEEE-802.3 CRC32. In this module, the generator polynomial is fixed to the numeric values for those two modes; therefore, the CRC value based on other generator polynomials cannot be calculated.

- CCITT CRC16 generator polynomial: 0x1021
- IEEE-802.3 CRC32 generator polynomial: 0x04C11DB7

CRC Configuration

Figure 1-1 shows the CRC configuration.

Figure 1-1 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.

2. 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)

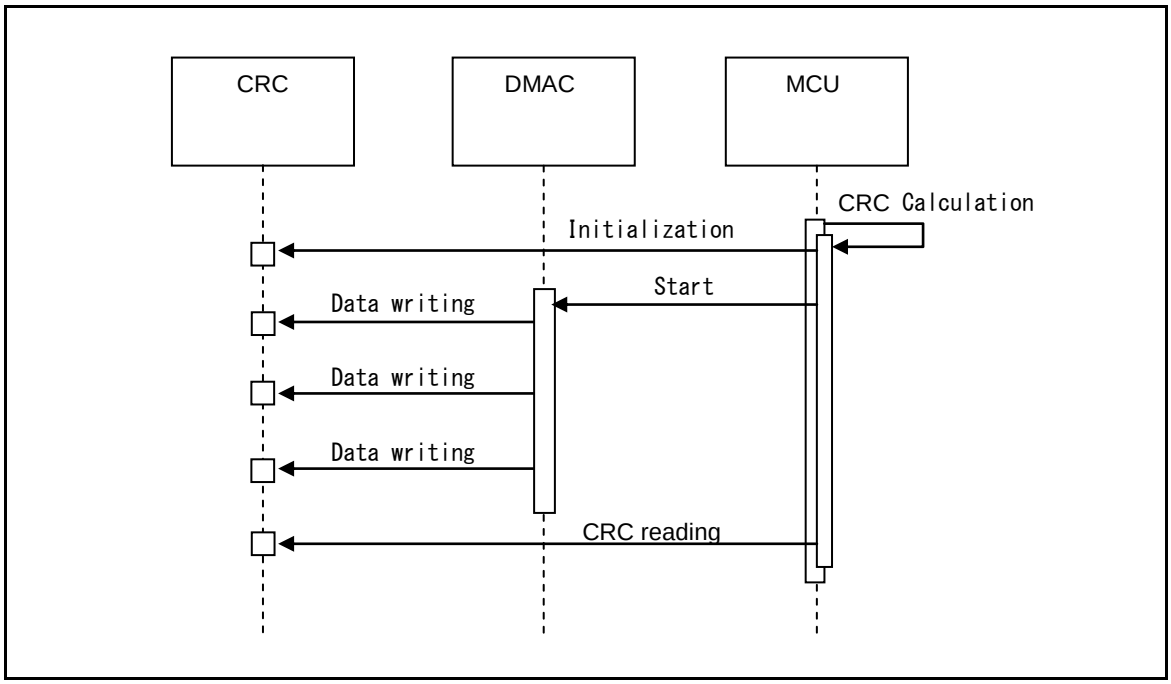
Note:

- *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).

2.1. Calculation Sequence

Figure 2-1 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 2-1 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).

2.2. Use Examples

Figure 2-2 to Figure 2-5 show CRC use examples.

Use Example 1 CRC16, Byte Input Fixed

Figure 2-2 Use Example 1 (CRC16, Byte Input Fixed, Core Byte Order: Big Endian)

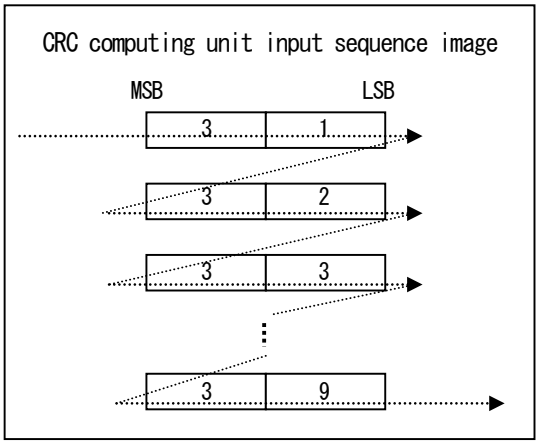
```
//*****************************************
//  CRC16 (CRC ITU-T)
//      polynomial:      0x1021
//      initial value:   0xFFFF
//      CRCCR.CRC32:     0 //CRC16
//      CRCCR.LTLEND:    0 //big endian
//      CRCCR.LSBFST:    0 //MSB First
//      CRCCR.CRCLTE:    0 //CRC big endian
//      CRCCR.CRCLSF:    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 2-1 shows the CRC result byte order, CRCR (CRC Register) output position, and read address in CRC16 mode.

Table 2-1 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

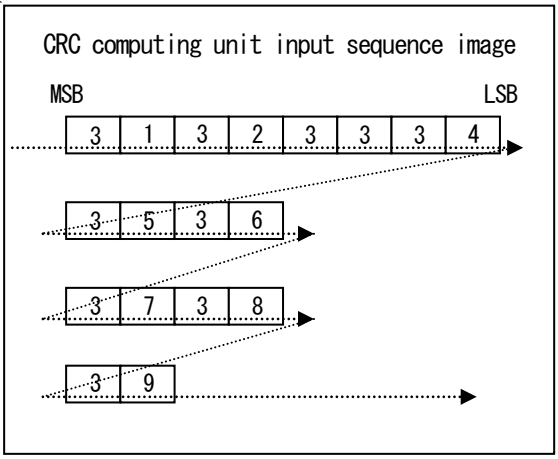
Use Example 2 CRC16, Different Input Bit Widths Mixed

Figure 2-3 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
```

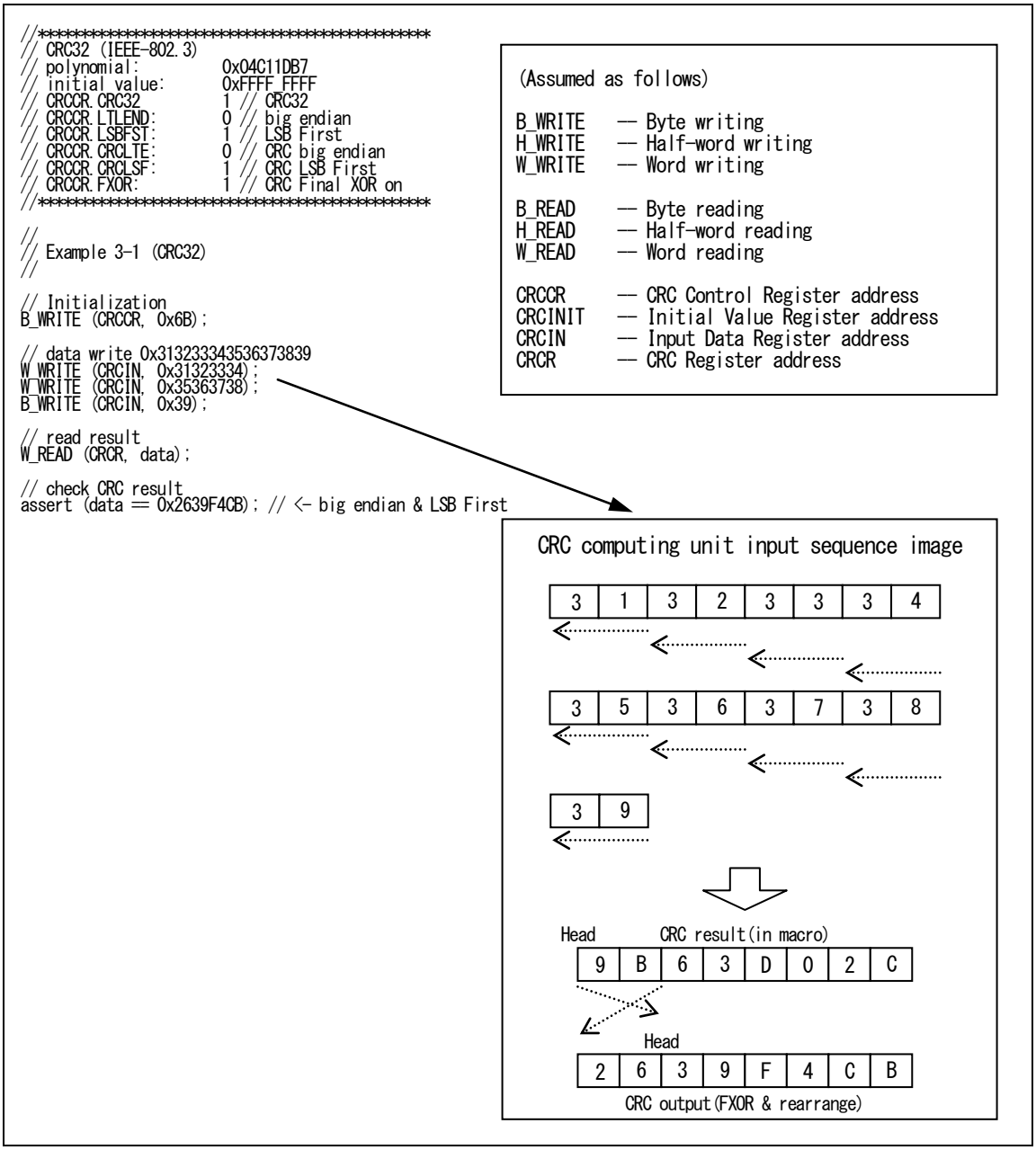


If the byte or bit order setting is correct and the bit input sequence to the CRC computing unit is the same, the writing width can be specified arbitrarily.

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

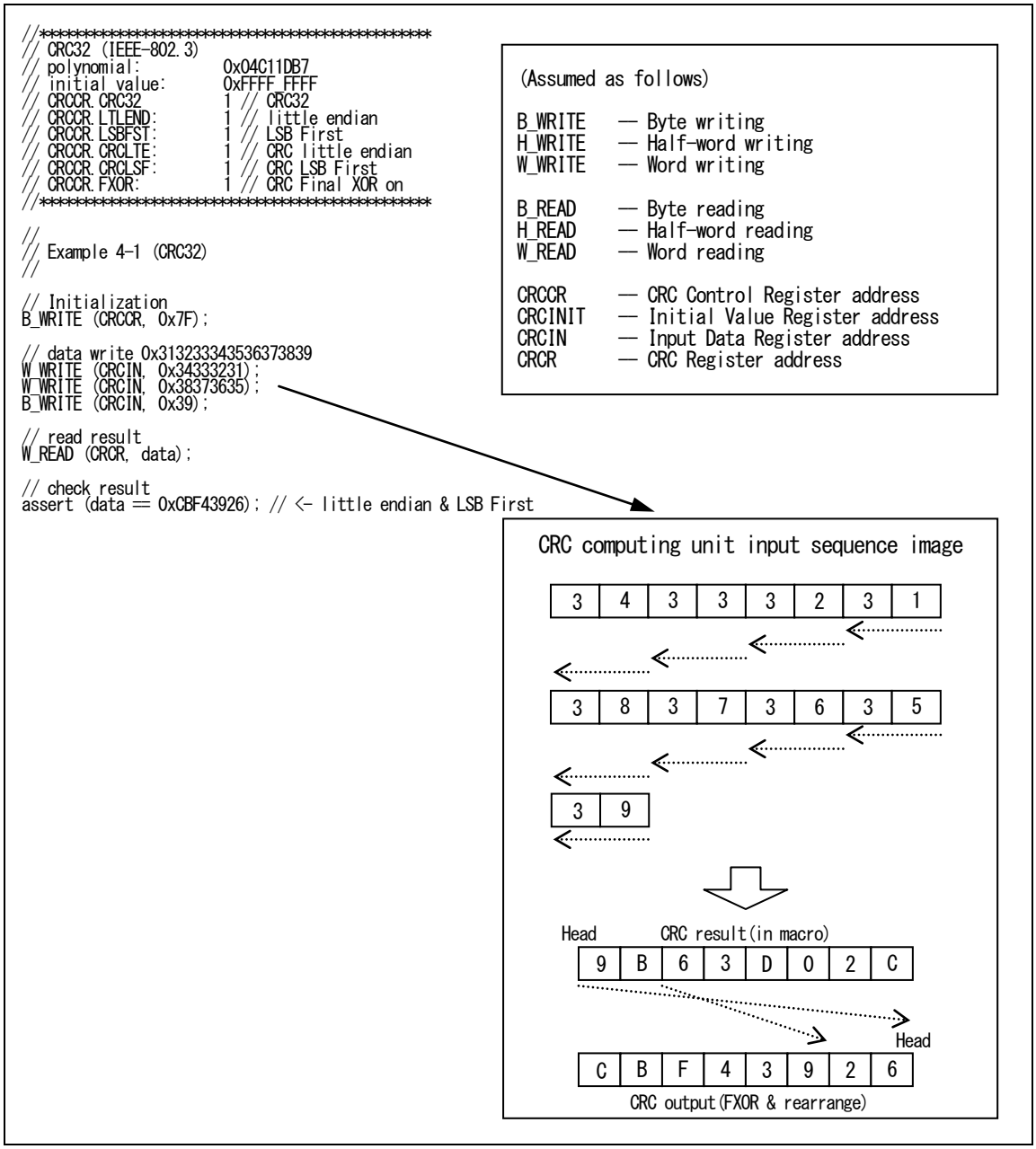
Figure 2-4 Use Example 3 (CRC32, Byte Order: Big Endian)



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 2-5 Use Example 4 (CRC32, Byte Order: Little 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 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).

3. Registers

This section provides a list of CRC registers.

Table 3-1 List of CRC Register List

Abbreviation	Register Name	Reference
CRCCR	CRC Control Register	3.1
CRCINIT	Initial Value Register	3.2
CRCIN	Input Data Register	3.3
CRCR	CRC Register	3.4

3.1. 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	

3.2. 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.

3.3. 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

3.4. 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-T00P00R01L06-E1-GBB01

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 5 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 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

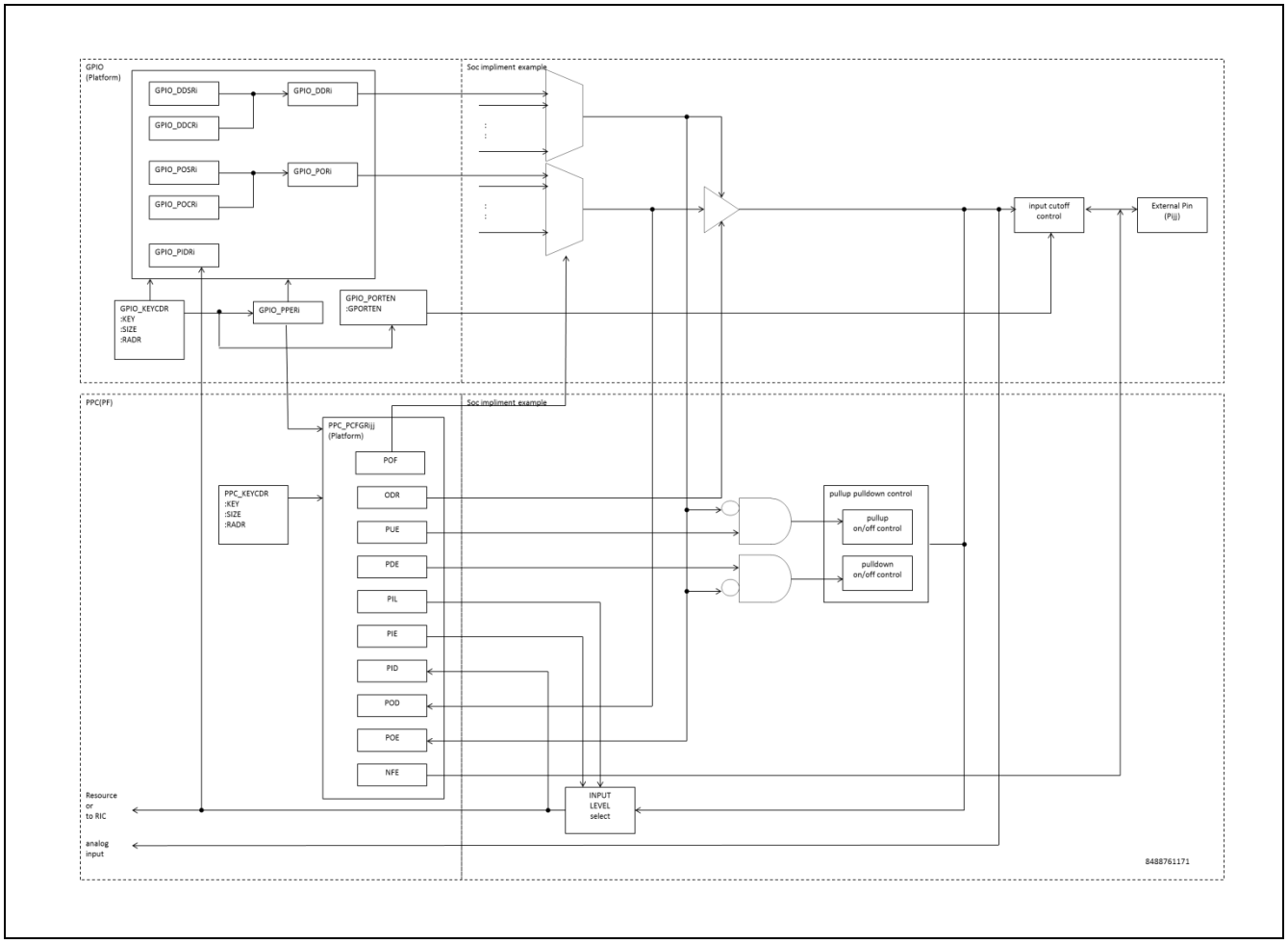
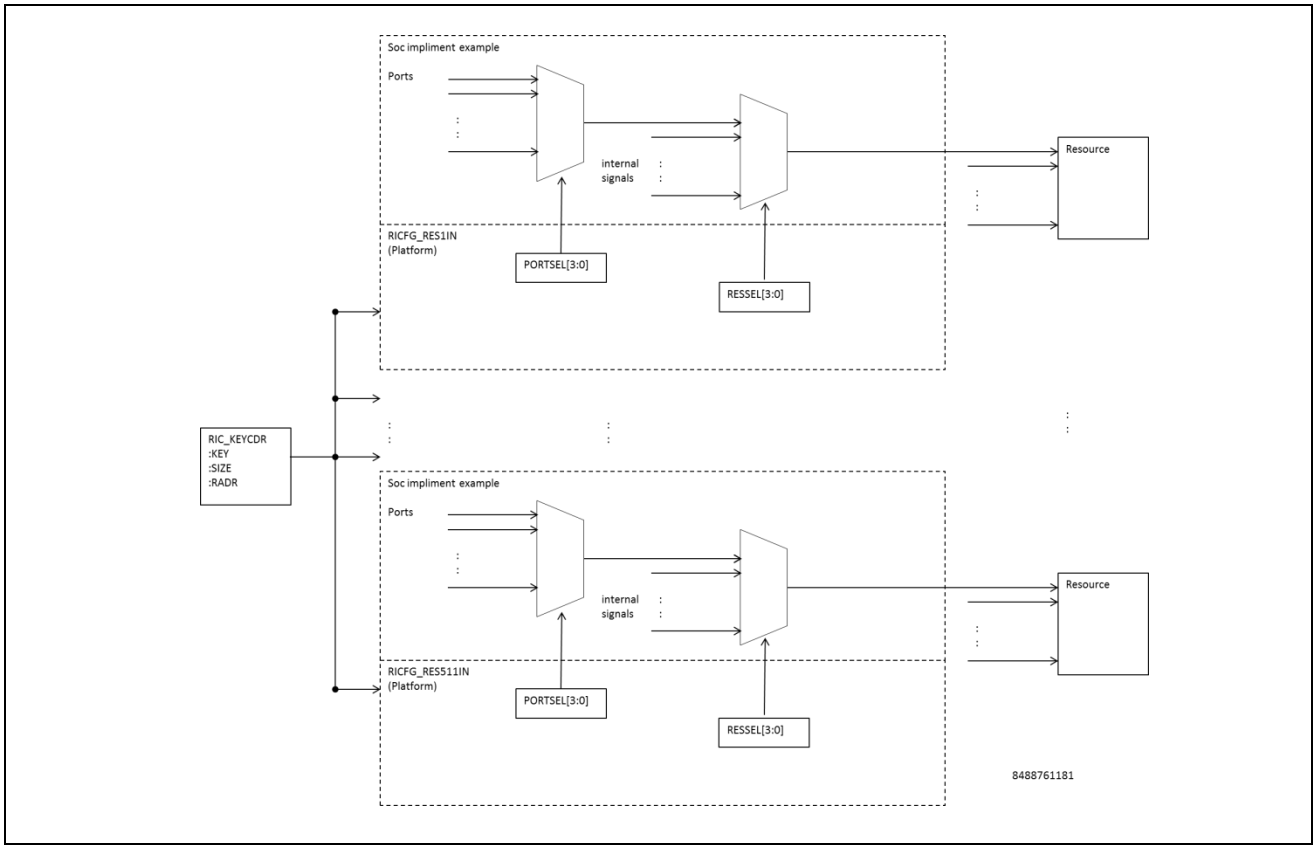


Figure 2-2 Configuration Diagram of RIC

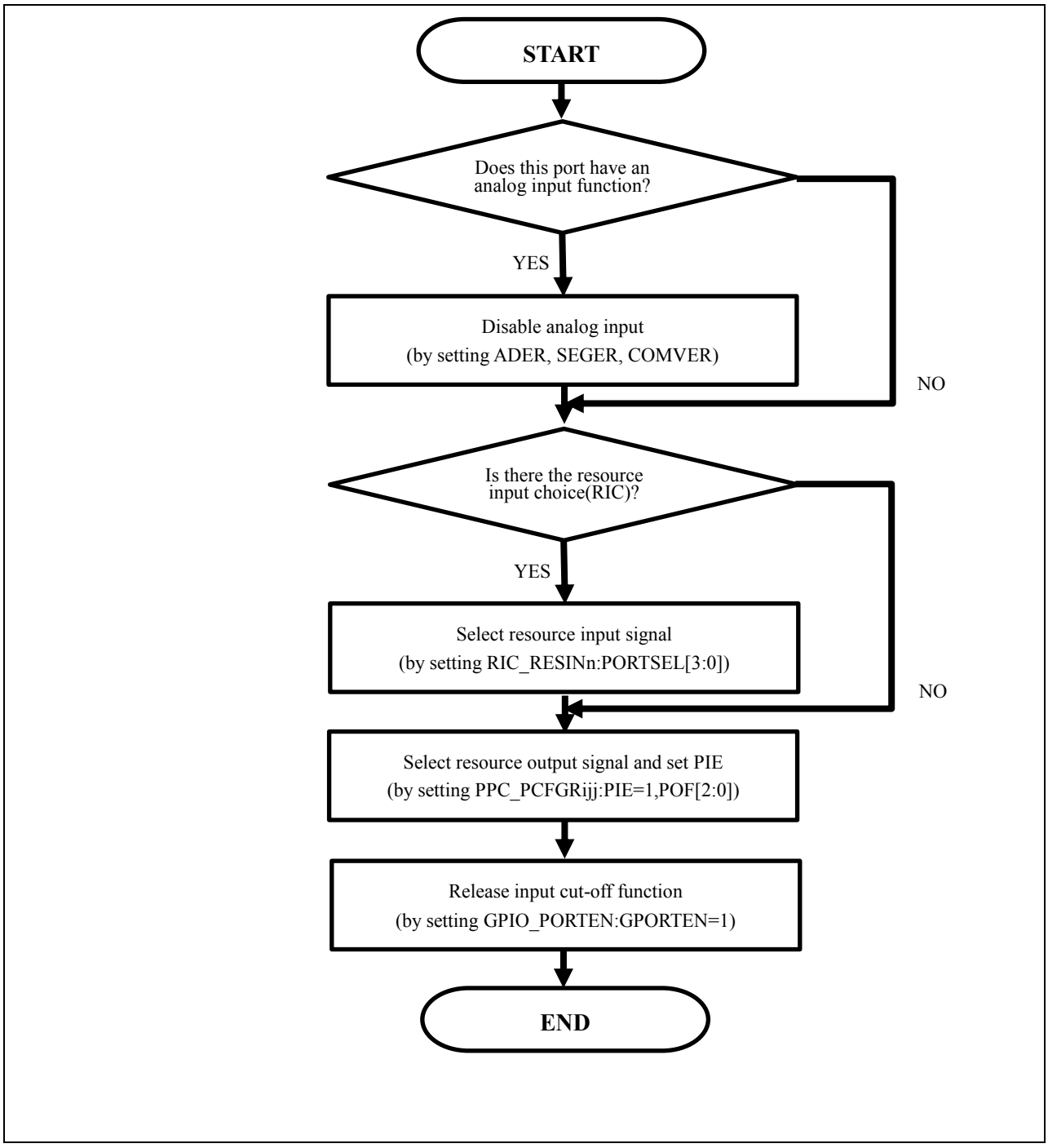


3. Setting Procedure Examples

This section describes the setting procedure examples of I/O port.

Assignment of Resource I/O Pin (both Directions)

Figure 3-1 Setting Procedure

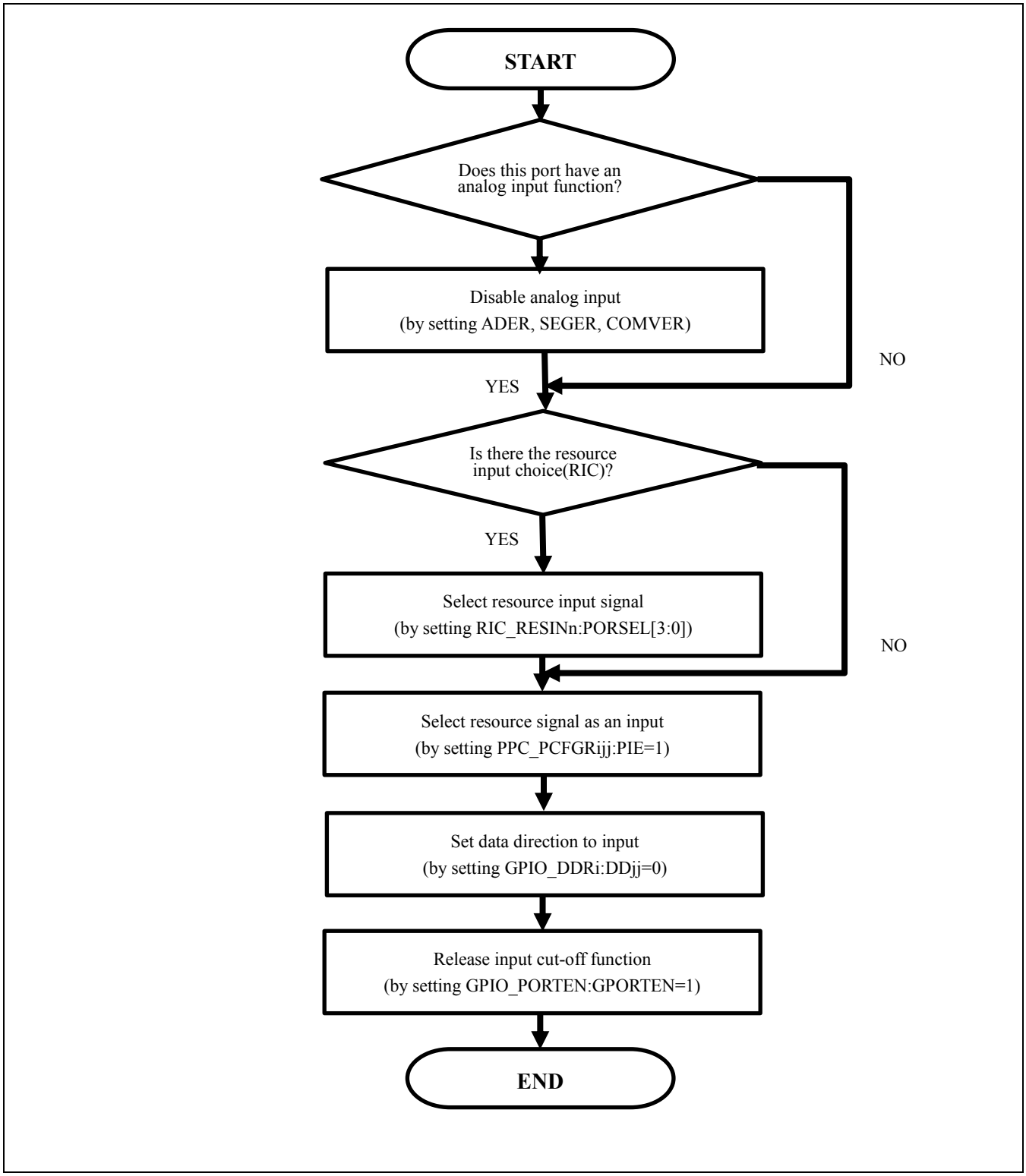


Notes:

- *The following registers are the applicable key code registers.*
  - *Analog input control register (ADER)*
  - *Segment output register (SEGER), Common output V pin control register (COMVER)*
  - *Resource input setting register (RIC_RESINn)*
  - *Port setting register (PPC_PCFGRijj)*
  - *Data direction register (GPIO_DDRi)*
  - *Port enable register (GPIO_PORTEN)*
- *For details on the analog I/O disable setting, see "12-bit A/D Converter" and "LCD CONTROLER".*

Assignment of Resource Input Pin (Selecting the Resource Input of an External Pin)

Figure 3-2 Setting Procedure

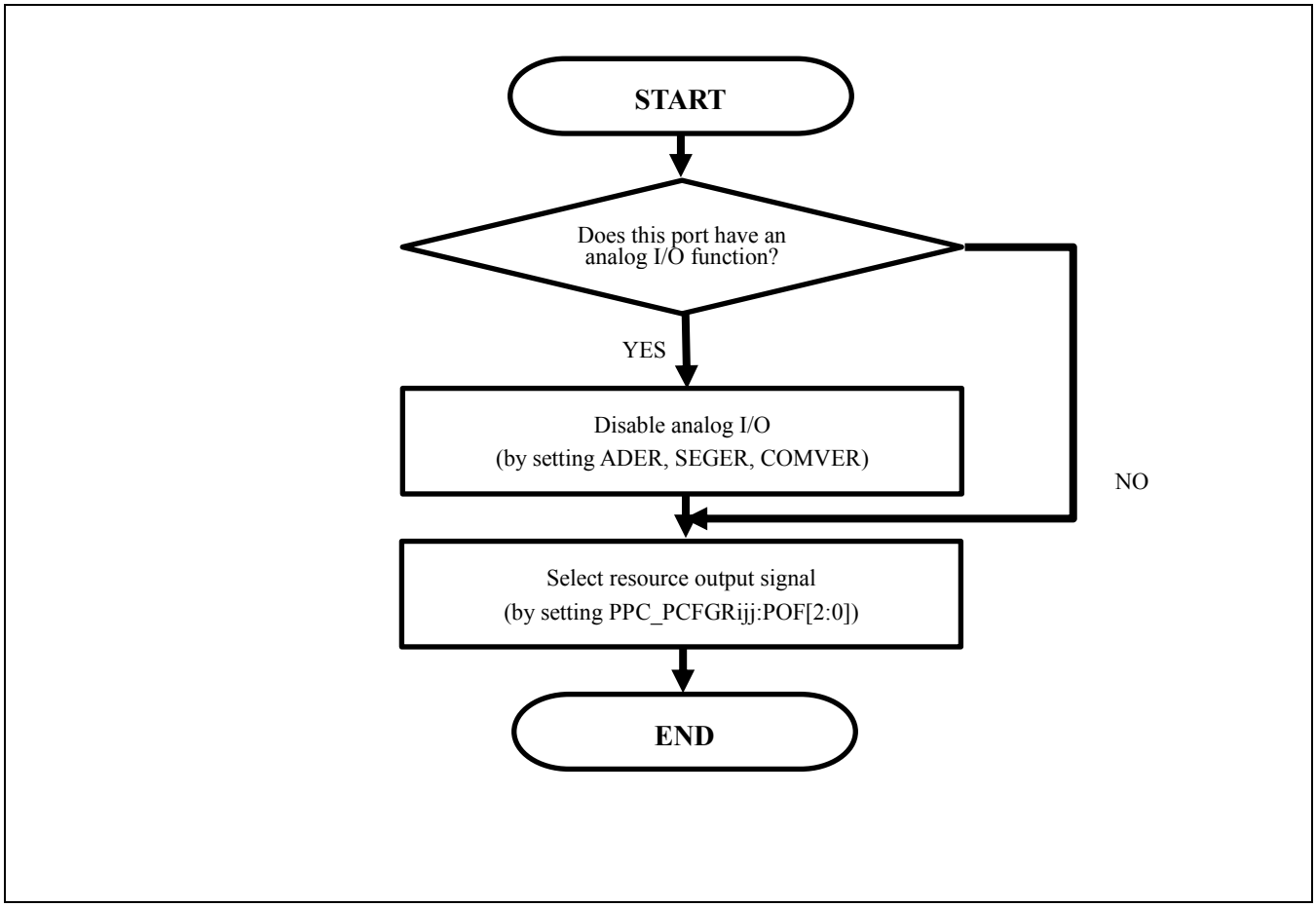


Notes:

- *The following registers are the applicable key code registers.*
  - *Analog input control register (ADER)*
  - *Segment output register (SEGER), Common output V pin control register (COMVER)*
  - *Resource input setting register (RIC_RESINn)*
  - *Port setting register (PPC_PCFGRijj)*
  - *Data direction register (GPIO_DDRi)*
  - *Port enable register (GPIO_PORTEN)*
- *For details on the analog I/O disable setting, see "12-bit A/D Converter" and "LCD CONTROLER".*

Assignment of Resource Output Pin

Figure 3-3 Setting Procedure

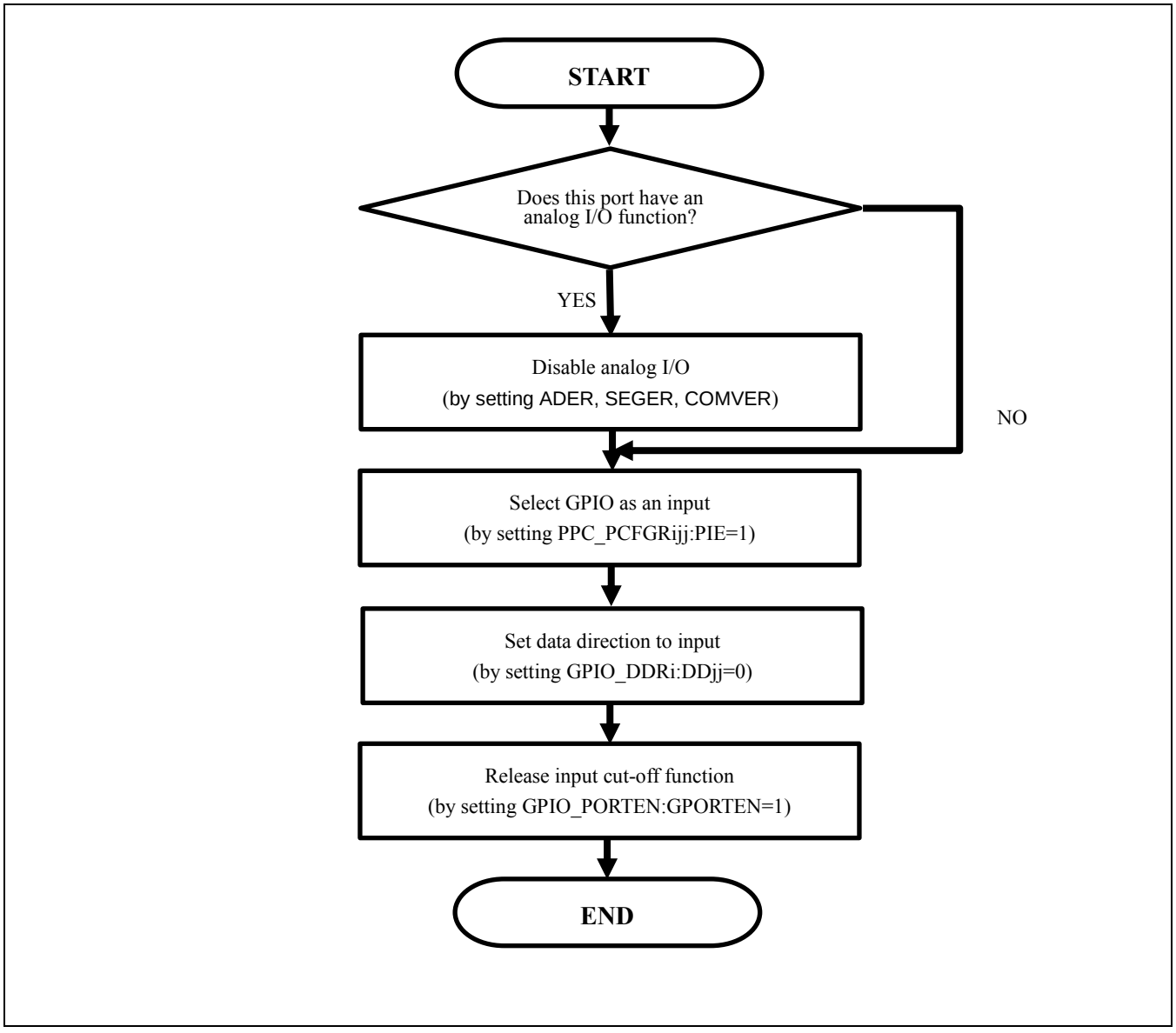


Notes:

- *The following registers are the applicable key code registers.*
  - *Analog input control register (ADER)*
  - *Segment output register (SEGER), Common output V pin control register (COMVER)*
  - *Port setting register (PPC_PCFGRijj)*
- *For details on the analog I/O disable setting, see "12-bit A/D Converter" and "LCD CONTROLER".*

Assignment of Port Function (Input)

Figure 3-4 Setting Procedure

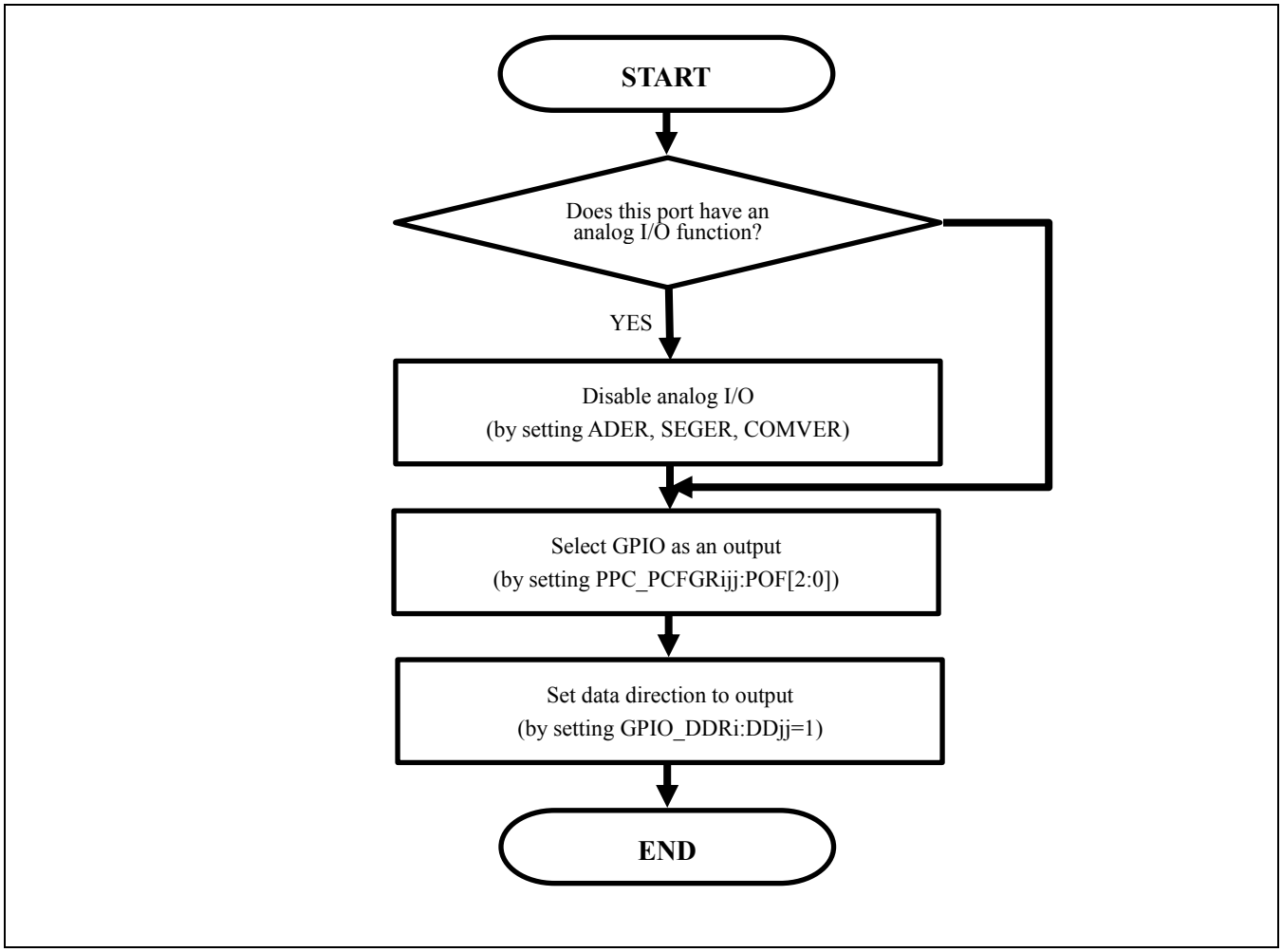


Notes:

- *The following registers are the applicable key code registers.*
  - *Analog input control register (ADER)*
  - *Segment output register (SEGER), Common output V pin control register (COMVER)*
  - *Port setting register (PPC_PCFGRijj)*
  - *Data direction register (GPIO_DDRi)*
  - *Port enable register (GPIO_PORTEN)*
- *For details on the analog I/O disable setting, see "12-bit A/D Converter" and "LCD CONTROLER".*

Assignment of Port Function (Output)

Figure 3-5 Setting Procedure



Notes:

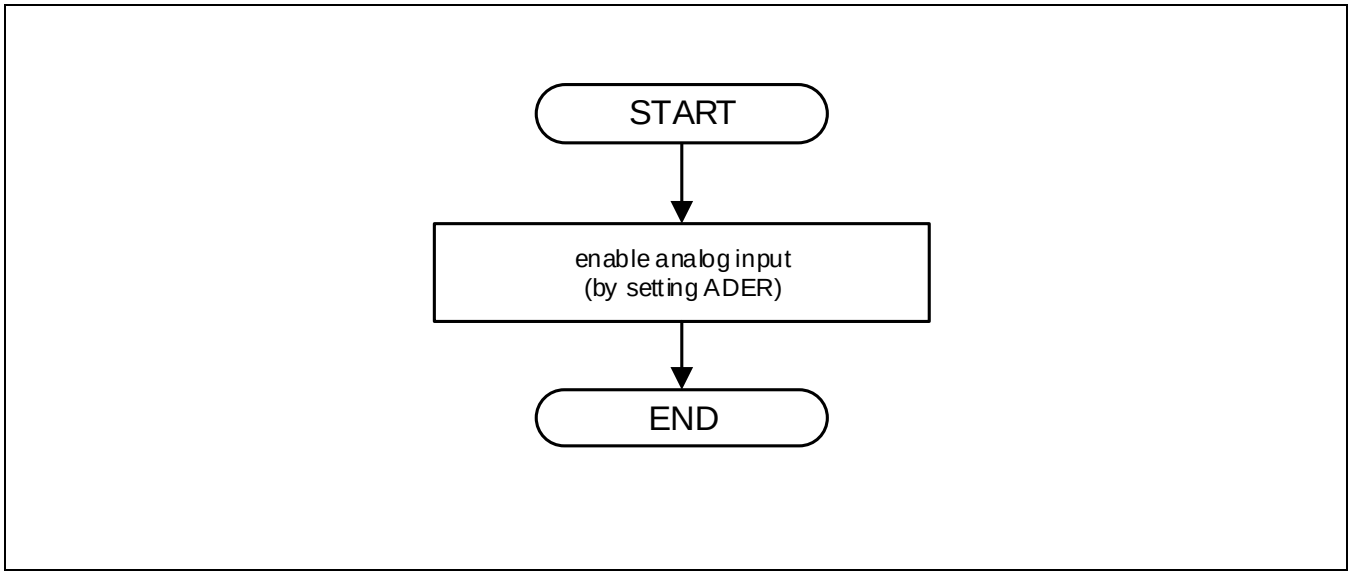
- *The following registers are the applicable key code registers.*
  - *Analog input control register (ADER)*
  - *Segment output register (SEGER), Common output V pin control register (COMVER)*
  - *Port setting register (PPC_PCFGRijj)*
  - *Data direction register (GPIO_DDRi)*

- *When JTAG terminals (TRST/TDO/TDI) and P321 are used as the port output, please set PPC_PCFGRijj:POF[2:0] to 001_B.*
  - *Please refer to "APPENDIX: VARIOUS SETTING OF I/O PORTS" to set function assigned to each terminal.*
  - *When they change JTAG terminal and P321 to a port, Pull-up/Pull-down become invalid.*

- *For details on the analog I/O disable setting, see "12-bit A/D Converter" and "LCD CONTROLER".*

Assignment of AD Converter Input

Figure 3-6 Setting Procedure

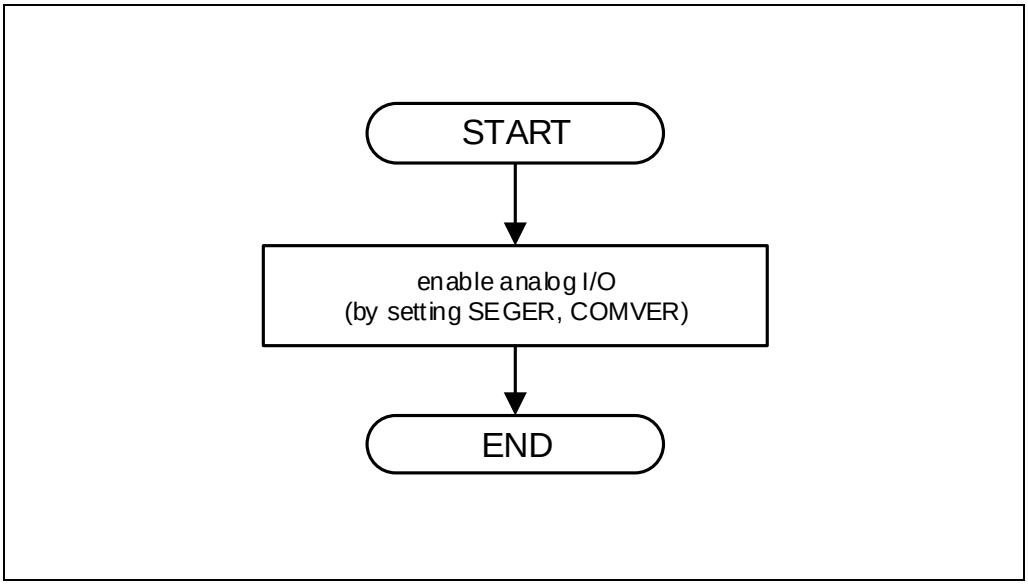


Notes:

- *The following registers are the applicable key code registers.*
  - *Analog input control register (ADER)*
- *For details on the analog input enable setting, see "12-bit A/D Converter".*

Assignment of LCD controller I/O pin

Figure 3-7 Setting Procedure

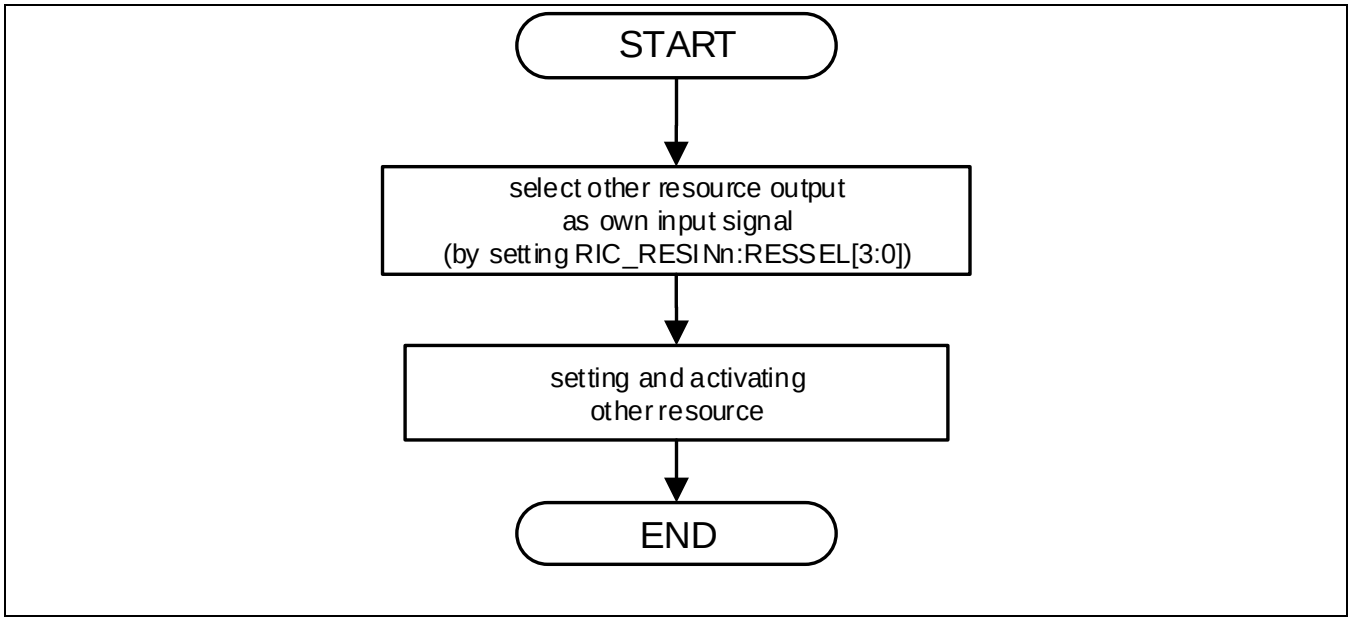


Notes:

- *The following registers are the applicable key code registers.*
  - *Segment output register (SEGER), Common output V pin control register (COMVER).*
- *For details on the analog I/O disable setting, see "LCD CONTROLER".*

Resource Input Selection (Selecting Other Resource Output)

Figure 3-8 Setting Procedure



Note:

- *For details on resource input selection, see "RESOURCE INPUT SELECTION of I/O PORT VARIOUS SETTINGS "*

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 4)

jj: GPIO channel number (jj = 00 to 31)

n: Number of selectable resource inputs (n = 0 to 367)

4.1. Data Direction Register (GPIO_DDRi) (i = 0 to 4)

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 31t	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.*
- *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 4)

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 4)

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 to31	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 4)

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	-	
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".

4.5. Port Output Set Register (GPIO_POSRi) (i = 0 to 4)

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	-	
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".

4.6. Port Output Clear Register (GPIO_POCRi) (i = 0 to 4)

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	-	
Initial Value	00000000_00000000_00000000_00000000	

[bit31:0] POC[n]: Port Output Clear Bit (n = 0 to 31)

POC[n] n=0 to 31t	Description
0	No effect
1	Clear the port output data bit (GPIO_PODRi: POD31 to POD0) to "0".

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	It is controlled by PIE bit whether the input buffer enabled or not.

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 4)

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 to31	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.

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 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 RADA) are accessed while writing the key code 0b00, 0b01, 0b10, and 0b11.*

4.10. Port Setting Register (PPC_PCFGRijj) (i = 0 to 4, 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 (GPIO_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 " INPUT LEVEL SETTING of I/O PORT VARIOUS SETTINGS."

[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 " OUTPUT DRIVE CAPACITY SETTING of I/O PORT VARIOUS SETTINGS."

[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.

Function to be used	PPC_PCFGR: NFE	EICxx_NFER: NFEx	RLTn_TMCSR: NFE	RIC_RESINyyy: RESSELz (yyy=350 to 367)
GPIO	1	0	0	0
Free-run timer(TEXT)	1	0	0	0
Input capture(IN)	1	0	0	0
Base timer(TIOA, TIOB)	1	0	0	0
Quad counter(AIN, BIN, CIN)	1	0	0	0
External interrupt(INT)	0	1	0	0
Reload timer(TIN)	0	0	1	0
I^2C INTERFACE	0	0	0	1

For details of pin assignment, see "List of Pin Functions" in "CHAPTER: OVERVIEW".
For details of EICxx_NFER:NFEx register, see "CHAPTER: EXTERNAL INTERRUPT".
For details of RLTn_TMCSR:NFE register, see "CHAPTER: RELOAD TIMER".
For details of RIC_RESINyyy:RESSELz register, see "APPENDIX: VARIOUS SETTINGS OF I/O PORTS".

[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 "OUTPUT RESOURCE SELECTION of I/O PORT VARIOUS SETTINGS."

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 RADA) are accessed while writing the key code 0b00, 0b01, 0b10, and 0b11.*

4.12. Resource Input Setting Register (RIC_RESINn) (n = 0 to 367)

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 "RESOURCE INPUT SELECTION of I/O PORT VARIOUS SETTINGS."

[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 "RESOURCE INPUT SELECTION of I/O PORT VARIOUS SETTINGS."

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 RADA) are accessed while writing the key code 0b00, 0b01, 0b10, and 0b11.*

5. Precautions for Using This Device

This section explains precautions for using I/O port.

5.1 External Pin setting priority

5.2 Notes on switching the I/O port function

5.1. External Pin setting priority

External pin setting priority is explained below.

The valid peripheral is determined based on the following priorities.
(1) A/D converter input, LCD controller I/O
(2) Other peripherals

Functions other than A/D converter input and LCD I/O is set by port output function selection bit (PPC_PCFGRijj.POF). Therefore, there is no priority.

When using a pin with the A/D converter function to perform a different function, set the relevant bit of the A/D converter analog input enable register (ADER) to "Analog input disable" in advance. For information on the setting method, see "CHAPTER: 12-bit A/D CONVERTER".
Similarly, when using a pin with the LCD controller function to perform a different function, set the relevant bit of the LCD controller segment output register and common output V pin control register (SEGER, COMVER) to "Analog input and output disable" in advance. For information on the setting method, see "CHAPTER: LCD CONTROLLER.
If analog input or analog output is enabled, inputs from ports and from peripheral functions are fixed at "0" and outputs are fixed at Hi-Z regardless of the port function register (PPC_PCFGRijj.POF) and extended port function register (RIC_RESINn) settings.

5.2. Notes on switching the I/O port function

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.

When switching noise filtering on or off, please execute it while resource input function is stopped.

When I/O port is switched from LCD analog output enable to LCD analog output disable, the pin state switches according to each port setting register. If setting after switching is output, a conflict of analog output and digital output may occur. Therefore, please set the port direction to input before setting analog output disable.

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-T00P00R01L02-E1-GBB01

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.

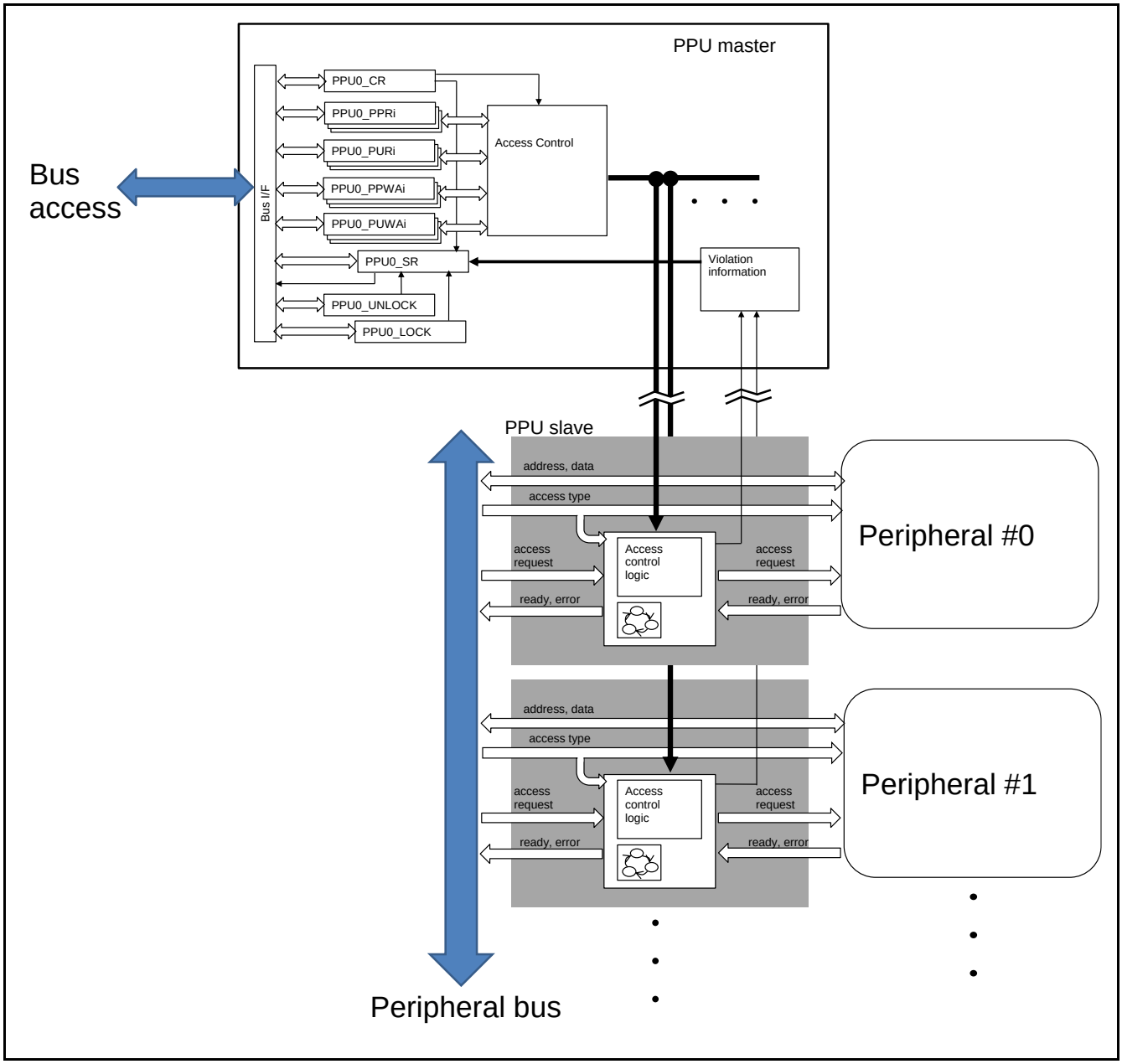
Refer OVERVIEW chapter to confirm which bus master is affected by the PPU.

Refer 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

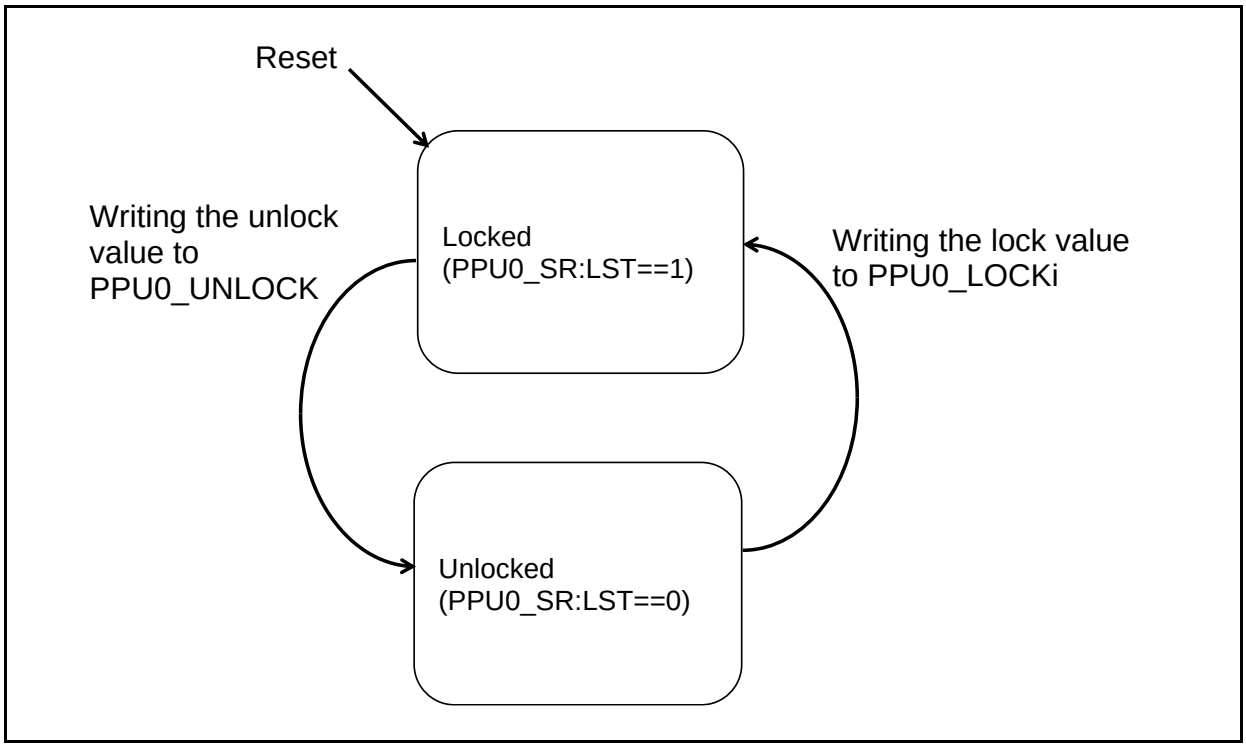
In this section explains access protection scheme to PPU registers.

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～0x07F	Reserved	Dummy value
3	0x080～0x0F4	PPU0_PPRi	Set value
4	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～0x07F	Reserved	Dummy value
3	0x080～0x0F4	PPU0_PPRi	Set value
4	0x0F8	PPU0_LOCK0	Dummy value（Excluding　0x30004000)
5	0x0FC	Reserved	Dummy value
6	0x100～0x174	PPU0_PURi	Set value
7	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～0x07F	Reserved	Dummy value
3	0x080～0x0F4	PPU0_PPRi	Set value
4	0x0F8	PPU0_LOCK0	Dummy value（Excluding　0x30004000)
5	0x0FC	Reserved	Dummy value
6	0x100～0x174	PPU0_PURi	Set value
7	0x178	PPU0_LOCK1	Dummy value（Excluding　0x30004000)
8	0x17C	Reserved	Dummy value
9	0x180～0x1F4	PPU0_PPWAi	Set value
10	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～0x07F	Reserved	Dummy value
3	0x080～0x0F4	PPU0_PPRi	Set value
4	0x0F8	PPU0_LOCK0	Dummy value（Excluding　0x30004000)
5	0x0FC	Reserved	Dummy value
6	0x100～0x174	PPU0_PURi	Set value
7	0x178	PPU0_LOCK1	Dummy value（Excluding　0x30004000)
8	0x17C	Reserved	Dummy value
9	0x180～0x1F4	PPU0_PPWAi	Set value
10	0x1F8	PPU0_LOCK2	Dummy value（Excluding　0x30004000)
11	0x1FC	Reserved	Dummy value
12	0x200～0x274	PPU0_PUWAi	Set value
13	0x278	PPU0_LOCK3	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

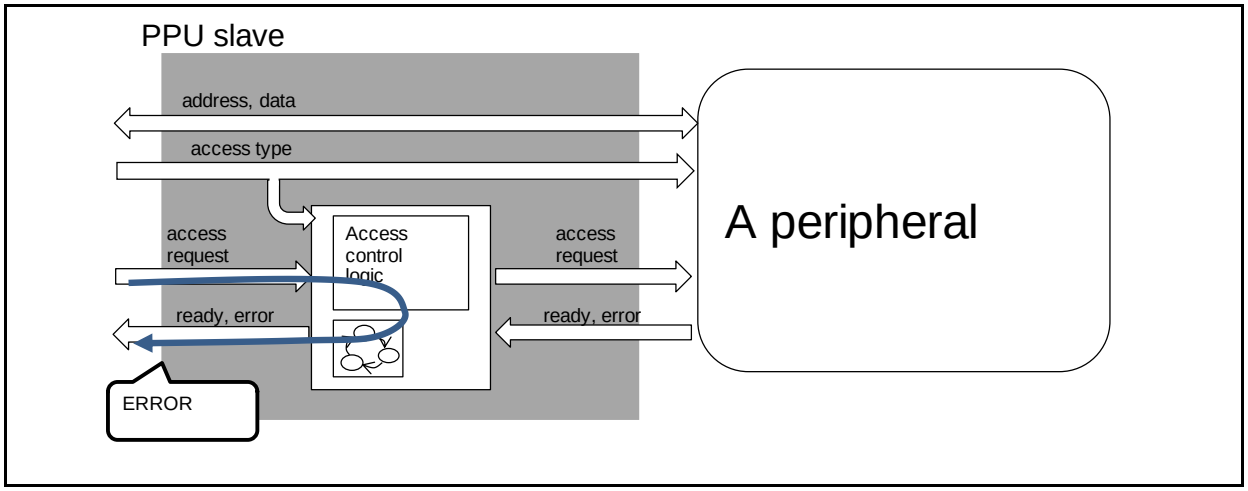
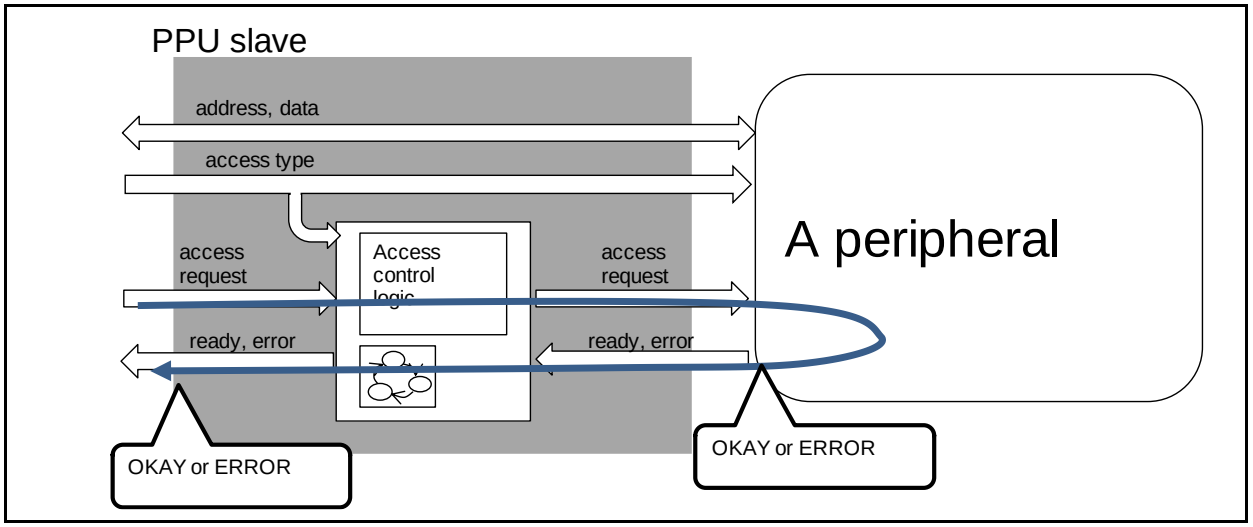


Figure 3-3 Access Operation when the Access is Permitted



4. Registers

This section describes the registers of PPU

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	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	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 29] 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.

[bit 23] 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.

Note:

- *The access from the CPU to the corresponding peripheral, please go after writing to the MODE bit is completed.*

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

[bit 23] 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.

[bit 22] VP : Violation Privileged Level Bit

This register contains information for privilege level of the first violated access. If the value is set once, this register is locked and does not accept information for privilege level of the violated access after that. 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.

[bit 21] VW : Violation Write Access Bit

This register contains information for access type (read or write) of the first violated access. If the value is set once, this register is locked and does not accept information for access type of the violated access after that. 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.

[bit 20:16] VL[4:0] : Violation Location

This register contains information for access destination of the first violated access. If the value is set once, this register is locked and does not accept further information for destination of the violated access after that. The bits are cleared by writing '1' to PPU0_CR:VCLR.

Bit	Description
00000	Reserved
00001	Violated access was done to a register in MCU Config Group.
00010	Reserved
00011	Violated access was done to a register in Memory Config Group.
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

[bit 0] 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
Memory Config Group	(Higher)
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.*

[bit 31:0] 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 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

[bit x] PPR[x] : PPU Privileged Mode Read Attribute Bit (x=31 to 0)

Read attribute for Privileged mode access of the peripheral #(i*32+x).

Bit	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.*

4.5. 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

[bit x] PUR[x] : PPU User Mode Read Attribute Bit (x=31 to 0)

Read attribute for user mode access of the peripheral #(i*32+x).

Bit	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.*

4.6. 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

[bit x] 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).

Bit	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.*

4.7. 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

[bit x] 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).

Bit	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.*

4.8. 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.*

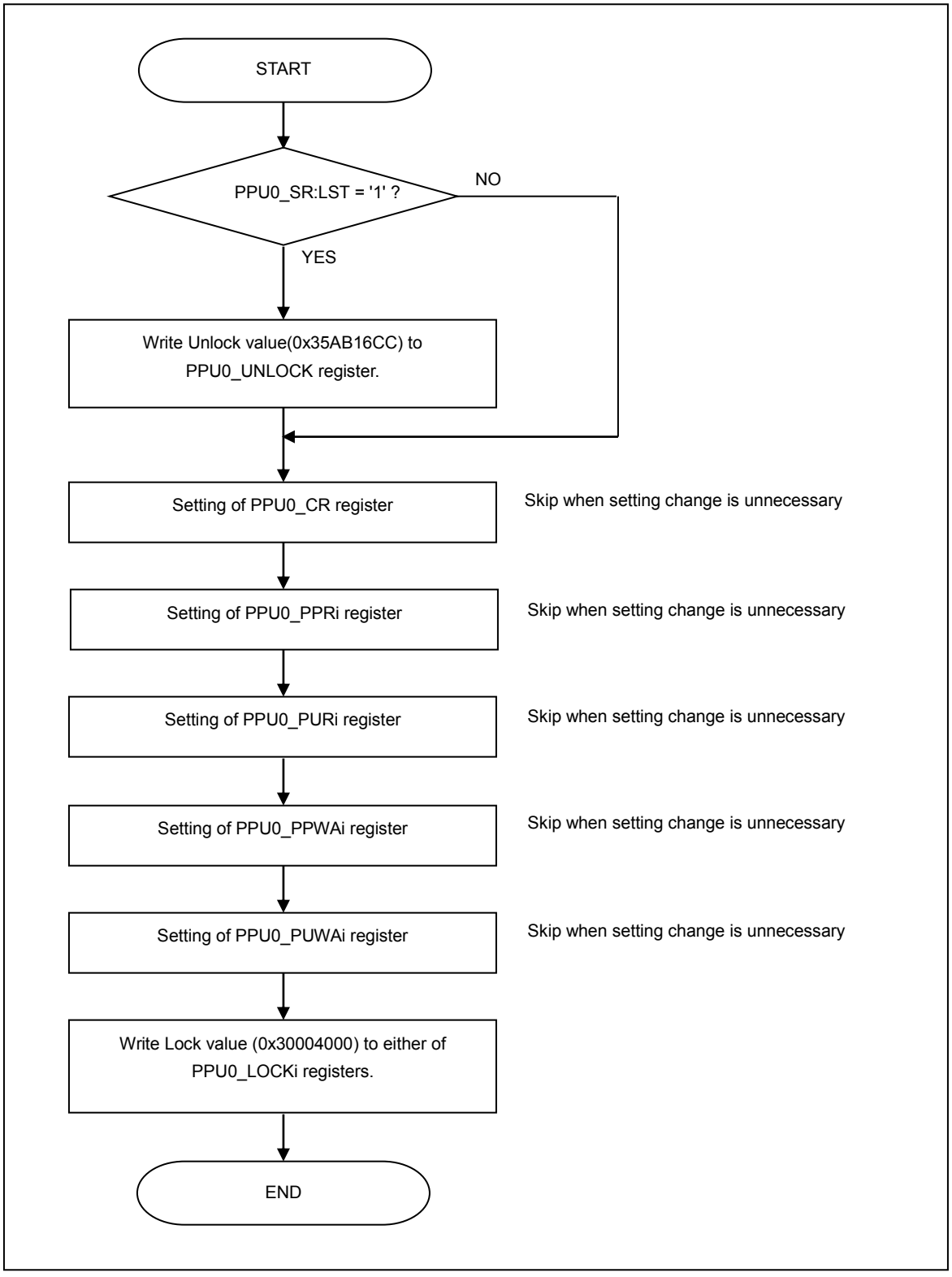
[bit 31:0] L : Lock

By writing 0x30004000, the PPU is locked and PPU0_SR:LST is set to '1'. Writing any other values has no effect.

5. Setup Steps of PPU Register

Setup steps of the PPU register are shown below.

Figure 5-1 Setup Steps of PPU Register



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-T01P00R01L01-E1-GBM00

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
  - 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

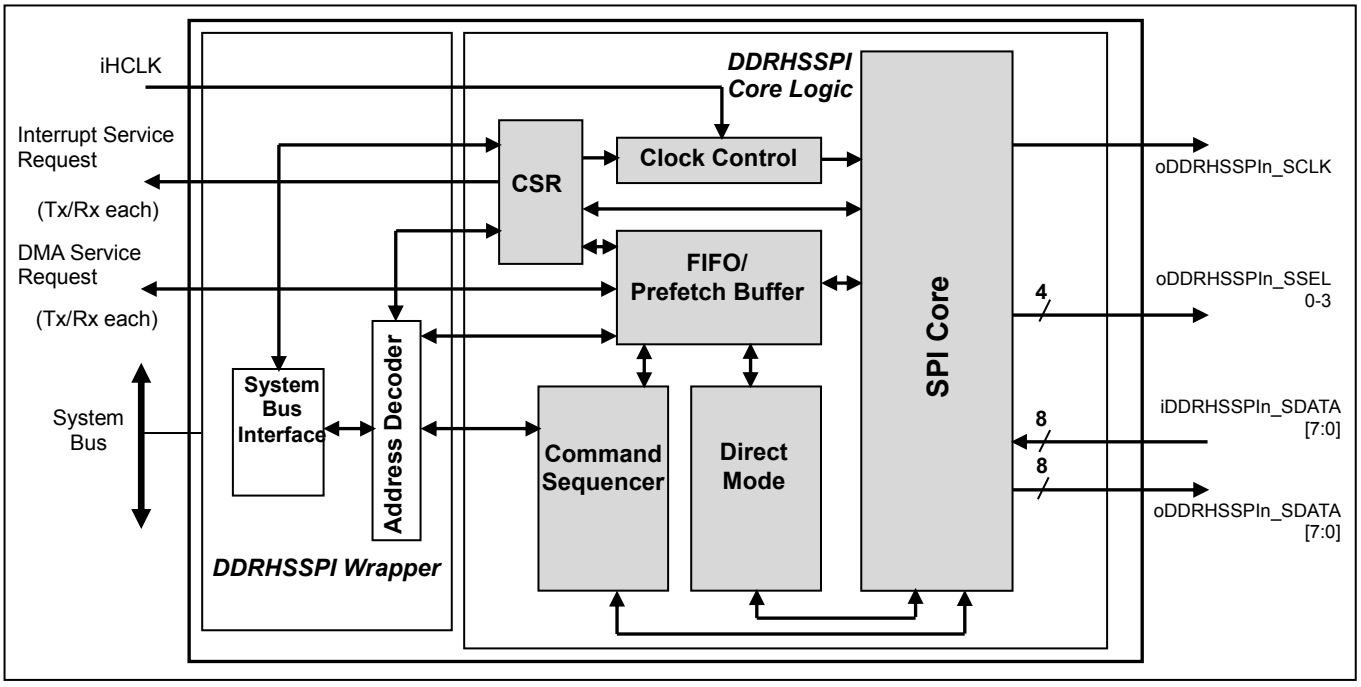
2. Configuration

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

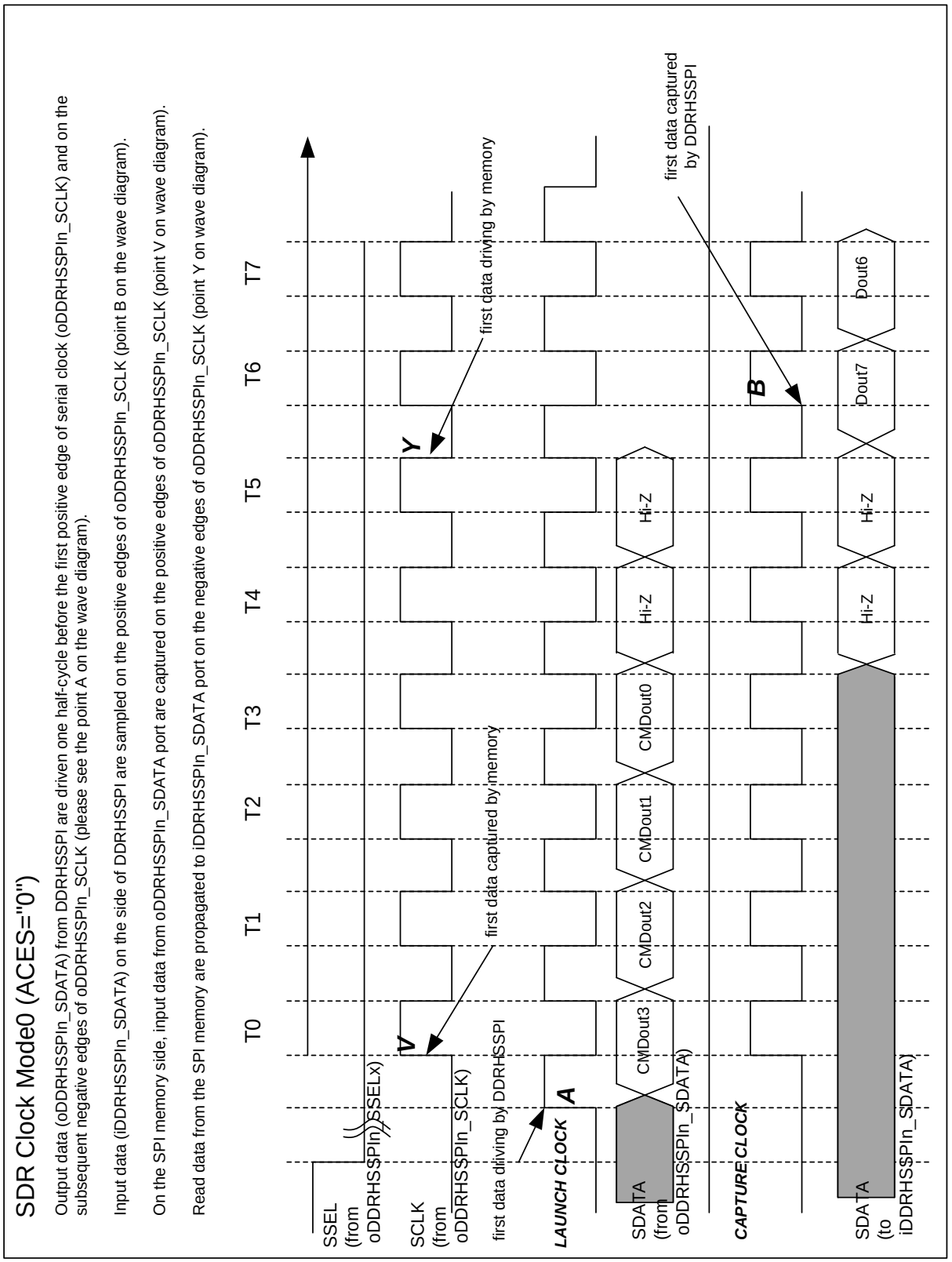


Figure 3-2 SDR Clock Mode 4

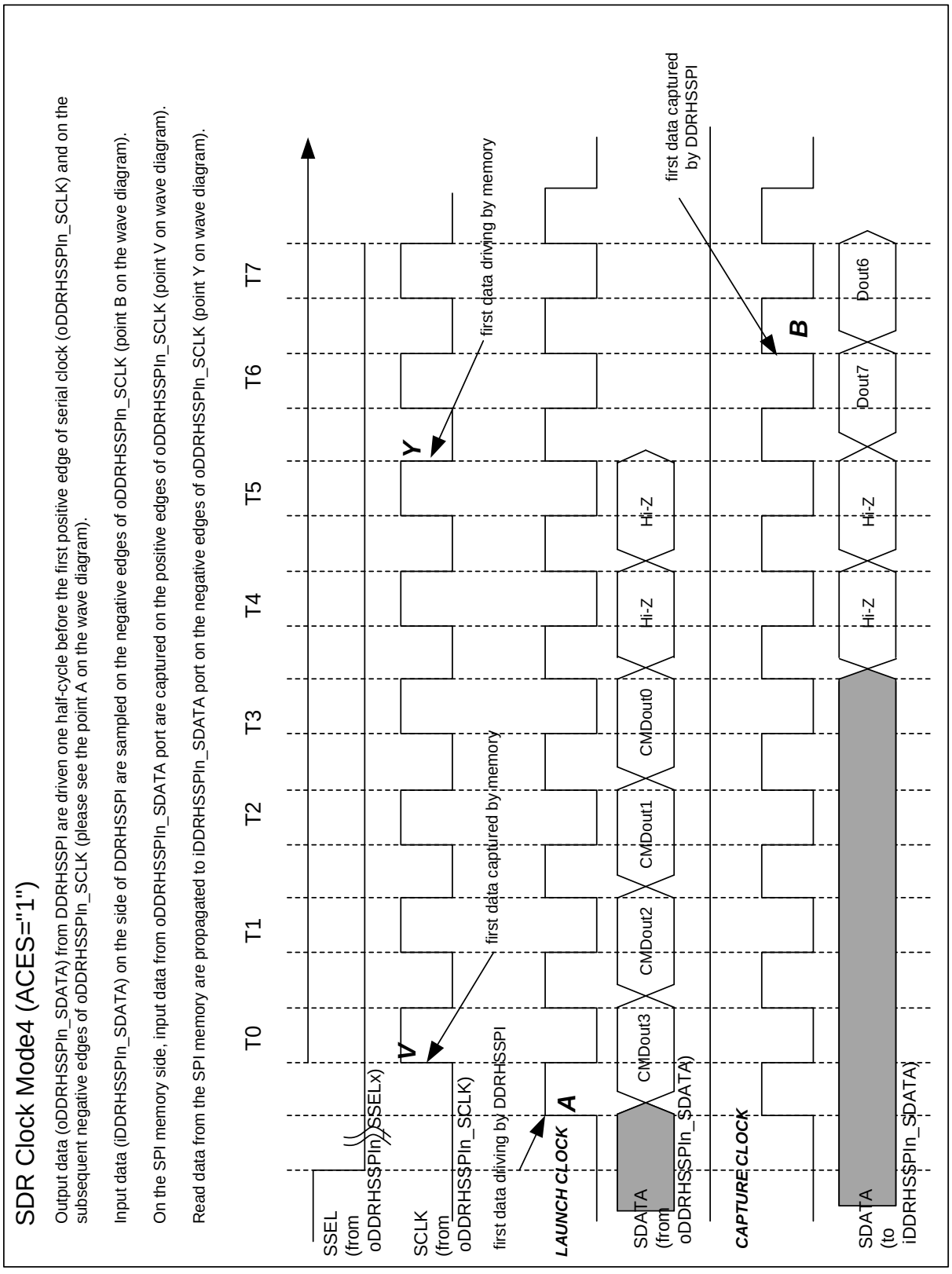


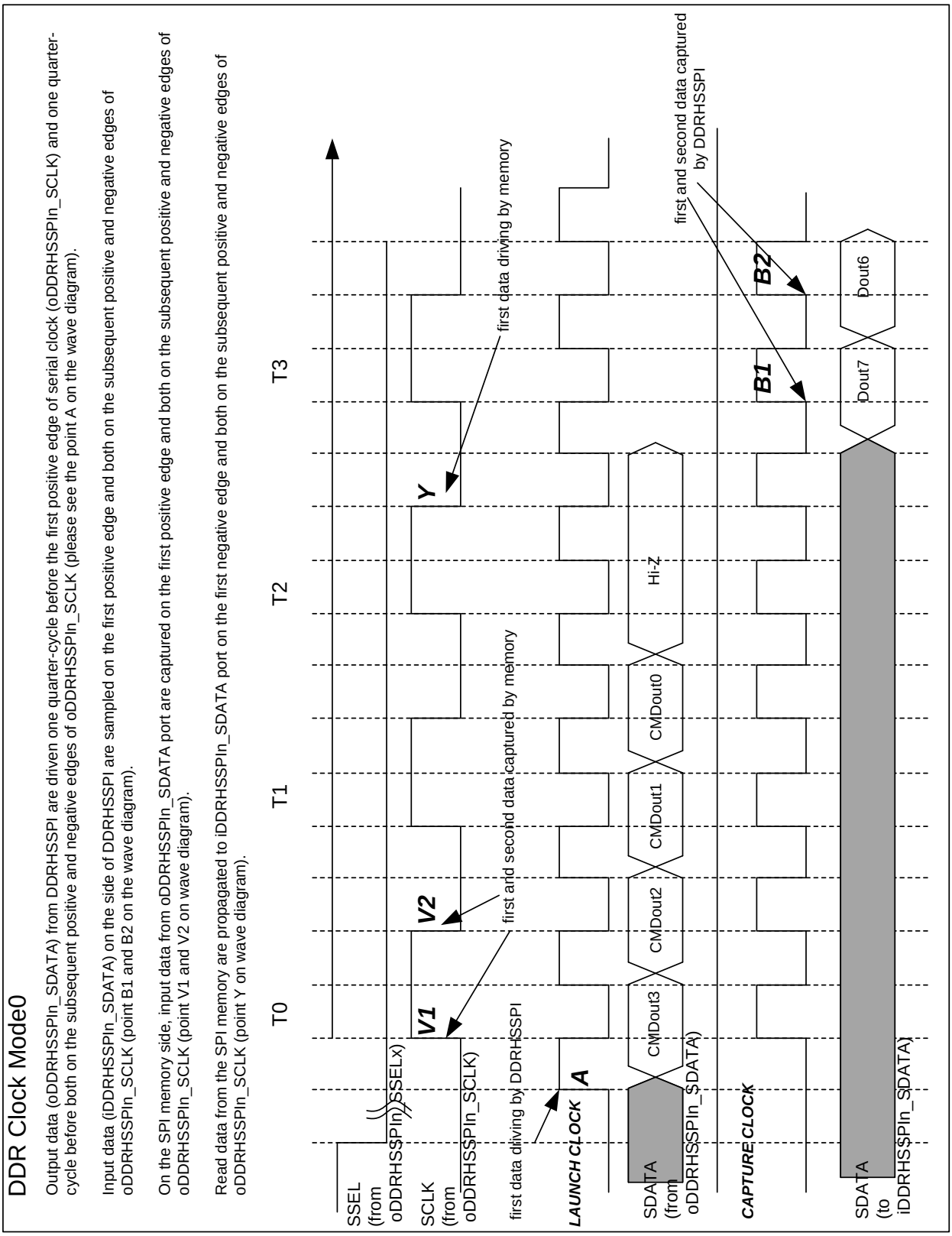
Table 3-2 Clock Modes in DDR Mode

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.

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.

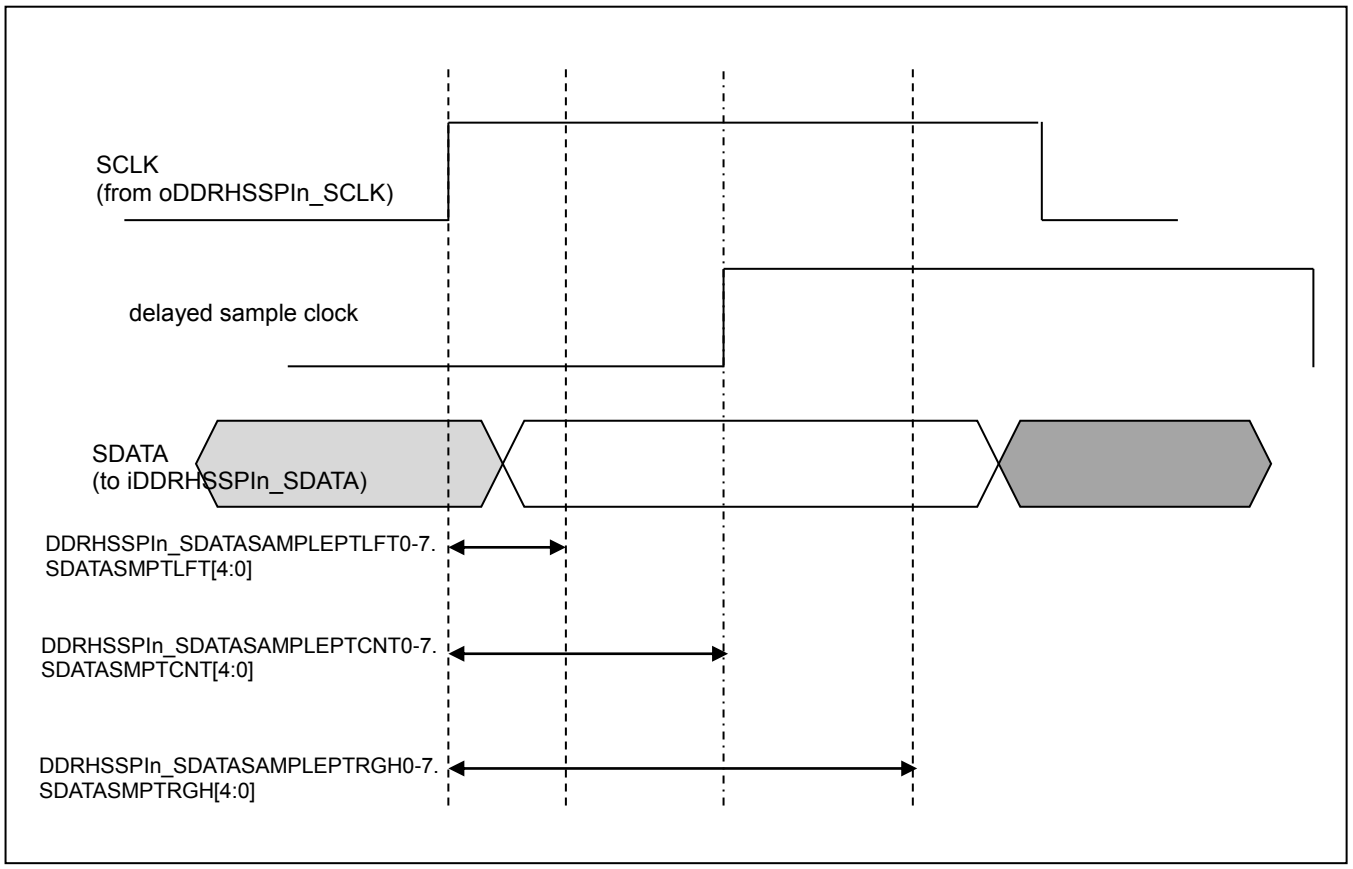
The resolution for the fine-tuning of the sampling point is fixed and it is equivalent to one buffer delay. Please consult physical implementation library datasheet for exact delay inserted with each buffer.

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.

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.

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.

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.

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

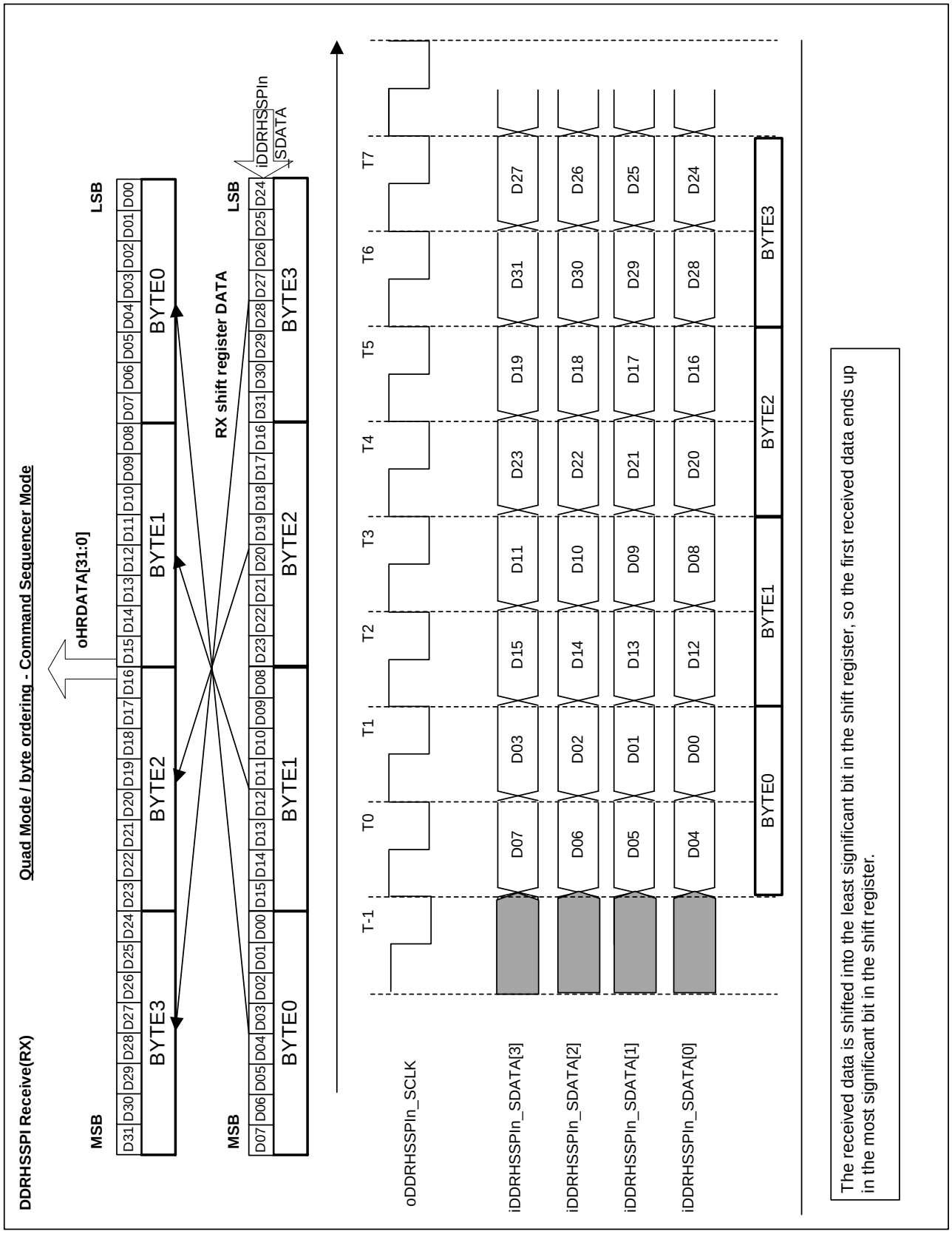
Most Significant Bit in the Shift Register is transmitted first and the first received data is shifted into the Least Significant Bit in the Shift Register. i.e. the Shift Register is shifted left. The read or write accesses to the data-registers always have Least Significant Bit of the data in bit 0.

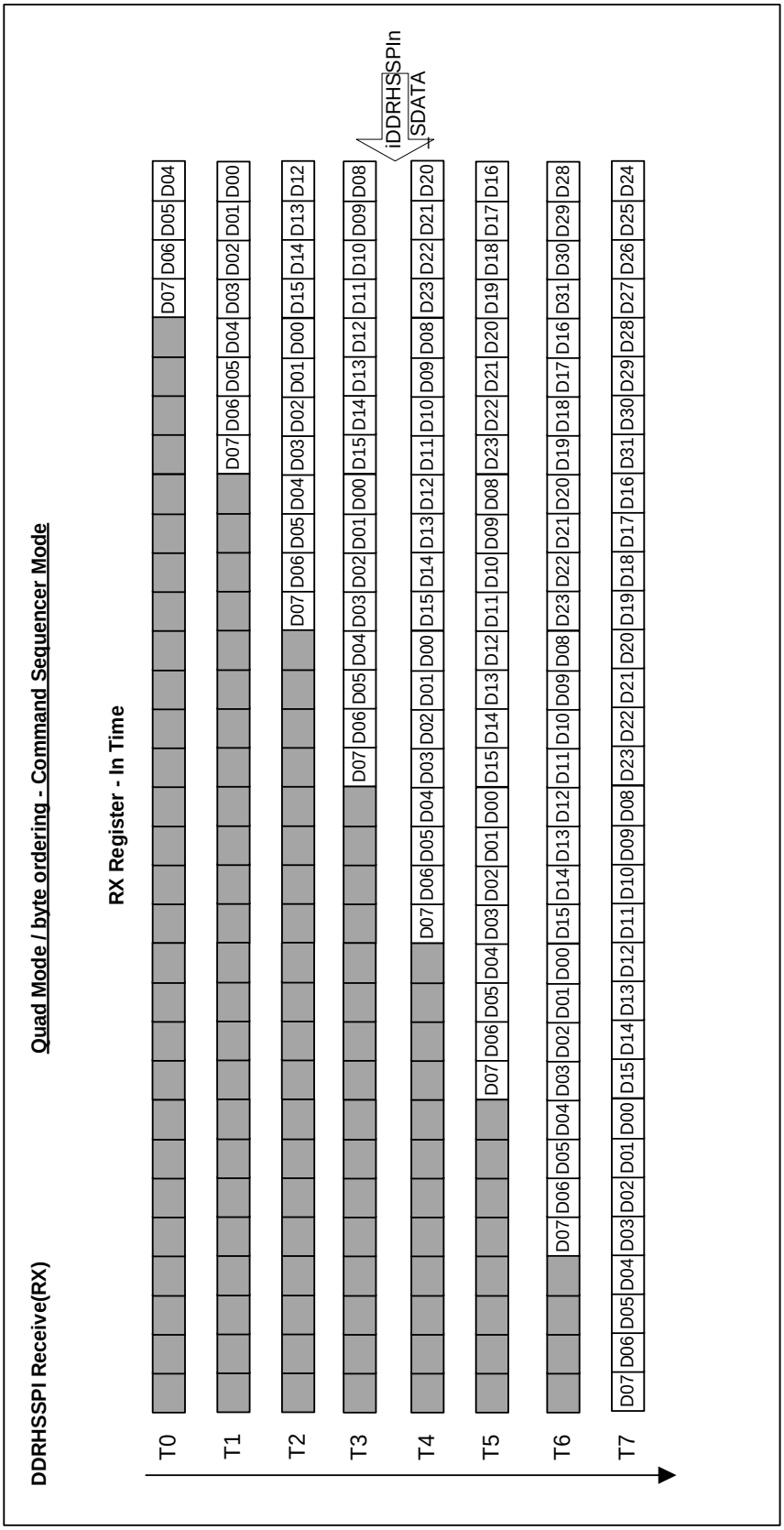
Figure 4-7 depicts the direction in which the data in the Shift Register is shifted to/from the serial data lines, under Quad Mode. The waveforms assume DDRHSSPIn_DMFIFOCFG.FWIDTH = "00". The Figures depict that the transmit data is loaded into the Shift Register from the TX-FIFO. However, the source of transmit data can be other registers, such as DDRHSSPIn_RDCSDC0-11.RDCSDATA.

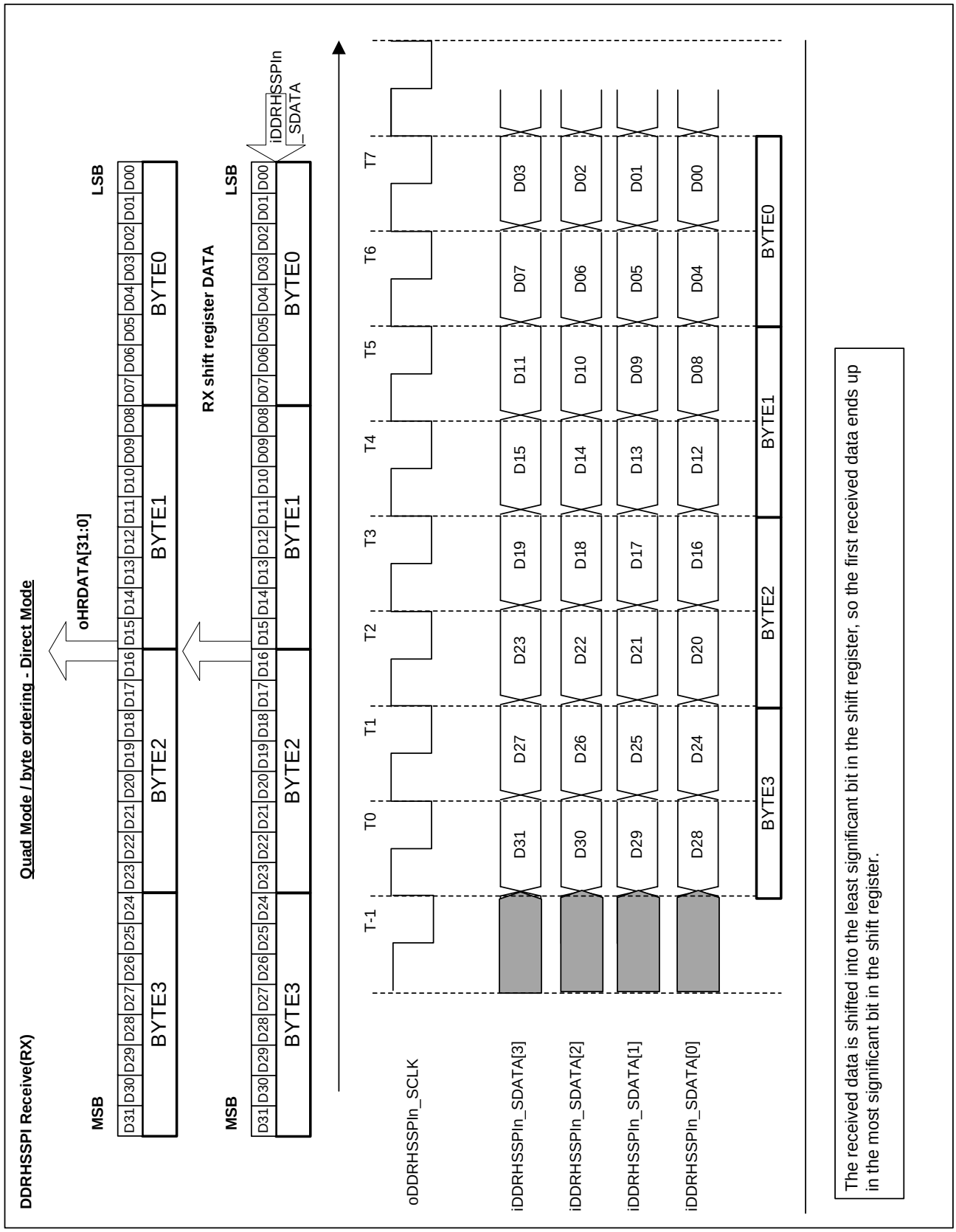
When DDRHSSPIn_MID.MID is 0x00000100, DDRHSSPI is available for Dual Legacy Mode and Dual Quad Mode. Figure 3-6 and Figure 3-7 depict the direction in which the data in the Shift Register is shifted to/from the serial data lines, under Dual Legacy Mode and Dual Quad Mode.

In Command Sequencer Mode, byte ordering is always LSB first.

In Direct Mode, byte ordering is always MSB first.


DDRHSSPI Receive(RX)
Quad Mode / byte ordering - Command Sequencer Mode
oHRDATA[31:0]
MSB
LSB
BYTE3
BYTE2
BYTE1
BYTE0
RX shift register DATA
iDDRHSSPIn SDATA
oDDRHSSPIn_SCLK
iDDRHSSPIn_SDATA[3]
iDDRHSSPIn_SDATA[2]
iDDRHSSPIn_SDATA[1]
iDDRHSSPIn_SDATA[0]
T-1
T0
T1
T2
T3
T4
T5
T6
T7
The received data is shifted into the least significant bit in the shift register, so the first received data ends up in the most significant bit in the shift register.


DDRHSSPI Receive(RX)
Quad Mode / byte ordering - Command Sequencer Mode
RX Register - In Time
iDDRHSSPIn SDATA
T0
T1
T2
T3
T4
T5
T6
T7


DDRHSSPI Receive(RX)
Quad Mode / byte ordering - Direct Mode
oHRDATA[31:0]
MSB
LSB
BYTE3
BYTE2
BYTE1
BYTE0
RX shift register DATA
iDDRHSSPIn_SDATA
oDDRHSSPIn_SCLK
iDDRHSSPIn_SDATA[3]
iDDRHSSPIn_SDATA[2]
iDDRHSSPIn_SDATA[1]
iDDRHSSPIn_SDATA[0]
T-1
T0
T1
T2
T3
T4
T5
T6
T7
The received data is shifted into the least significant bit in the shift register, so the first received data ends up in the most significant bit in the shift register.

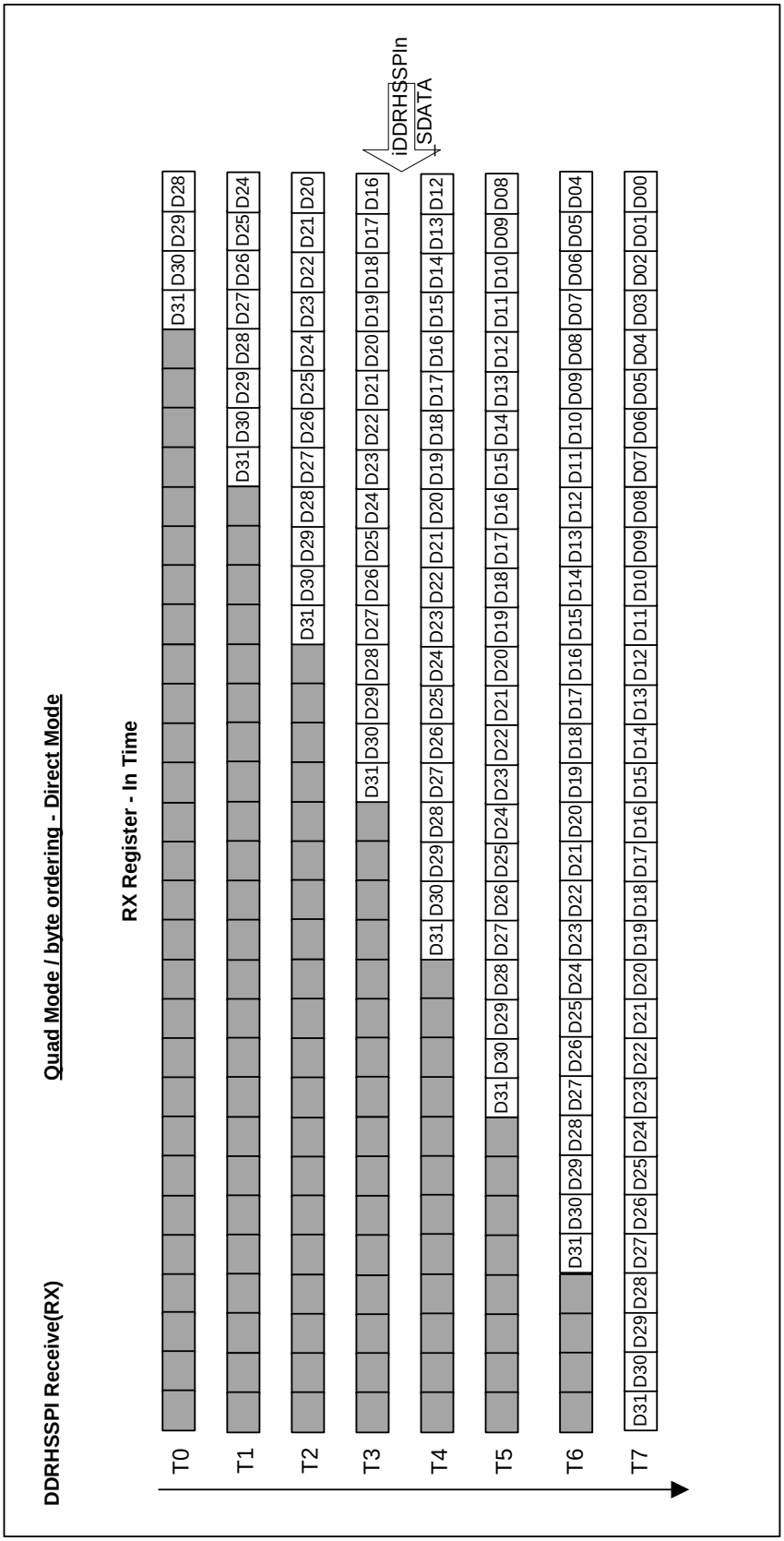

DDRHSSPI Receive(RX)
Quad Mode / byte ordering - Direct Mode
RX Register - In Time
iDDRHSSPIn
SDATA
T0
T1
T2
T3
T4
T5
T6
T7

Figure 3-5 Shift Direction (Legacy Mode and Quad Mode)
(Assumptions: DDRHSSPIn_DMFIFOCFG.FWIDTH="00")

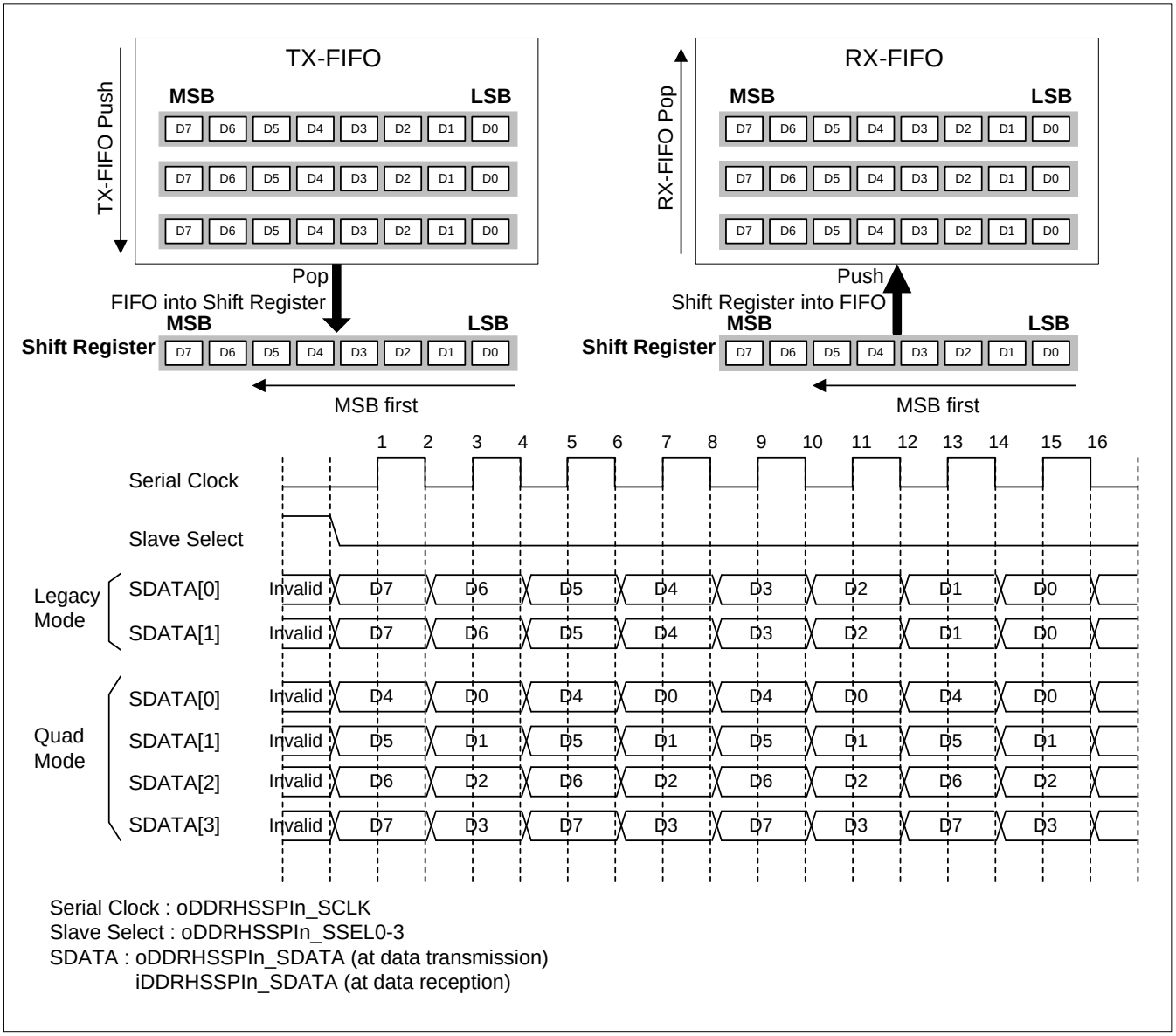


Figure 3-6 Shift Direction (Dual Legacy Mode and Dual Quad Mode in Direct Mode)
(Assumptions: DDRHSSPIn_DMFIFOCFG.FWIDTH="00")

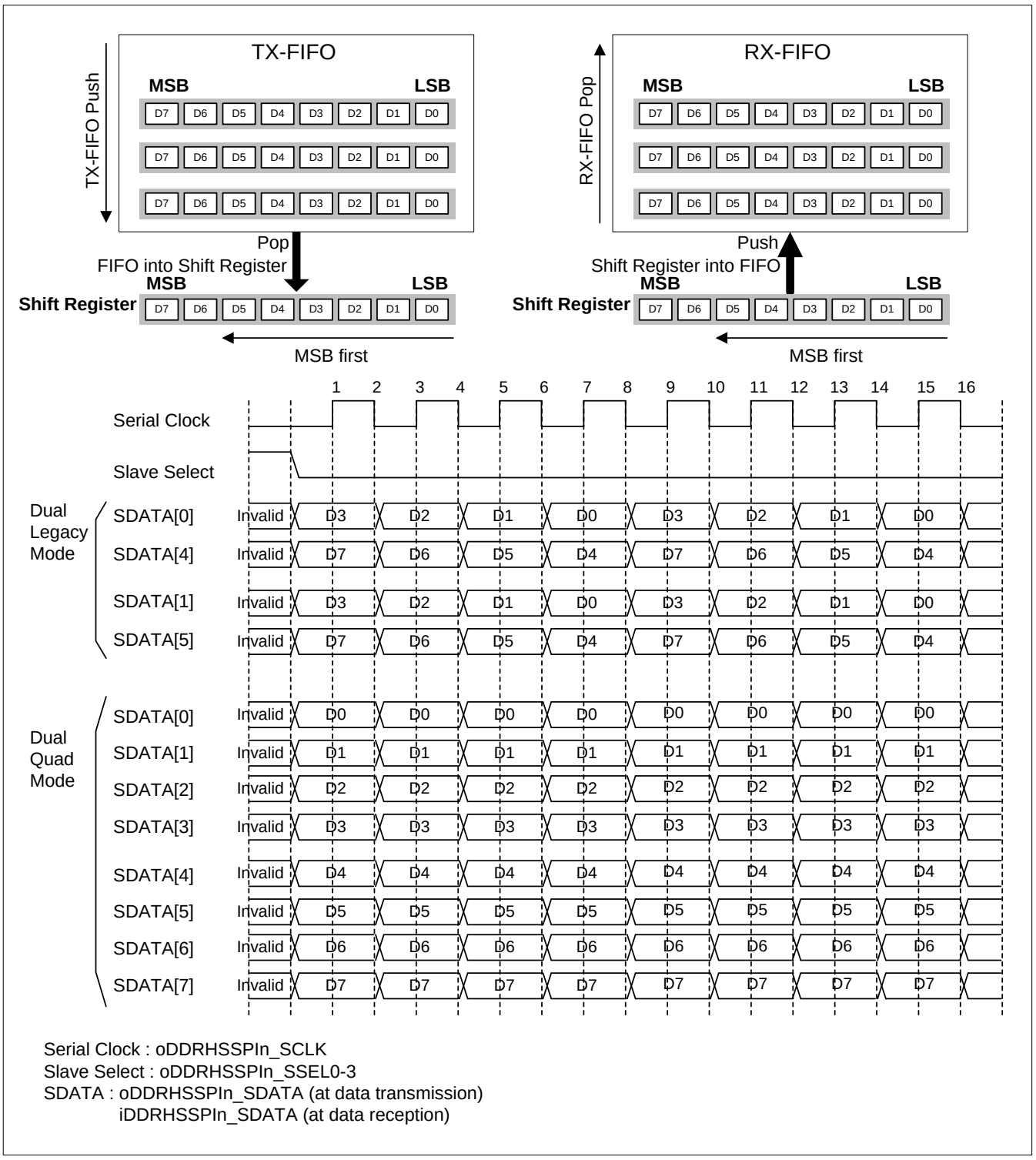
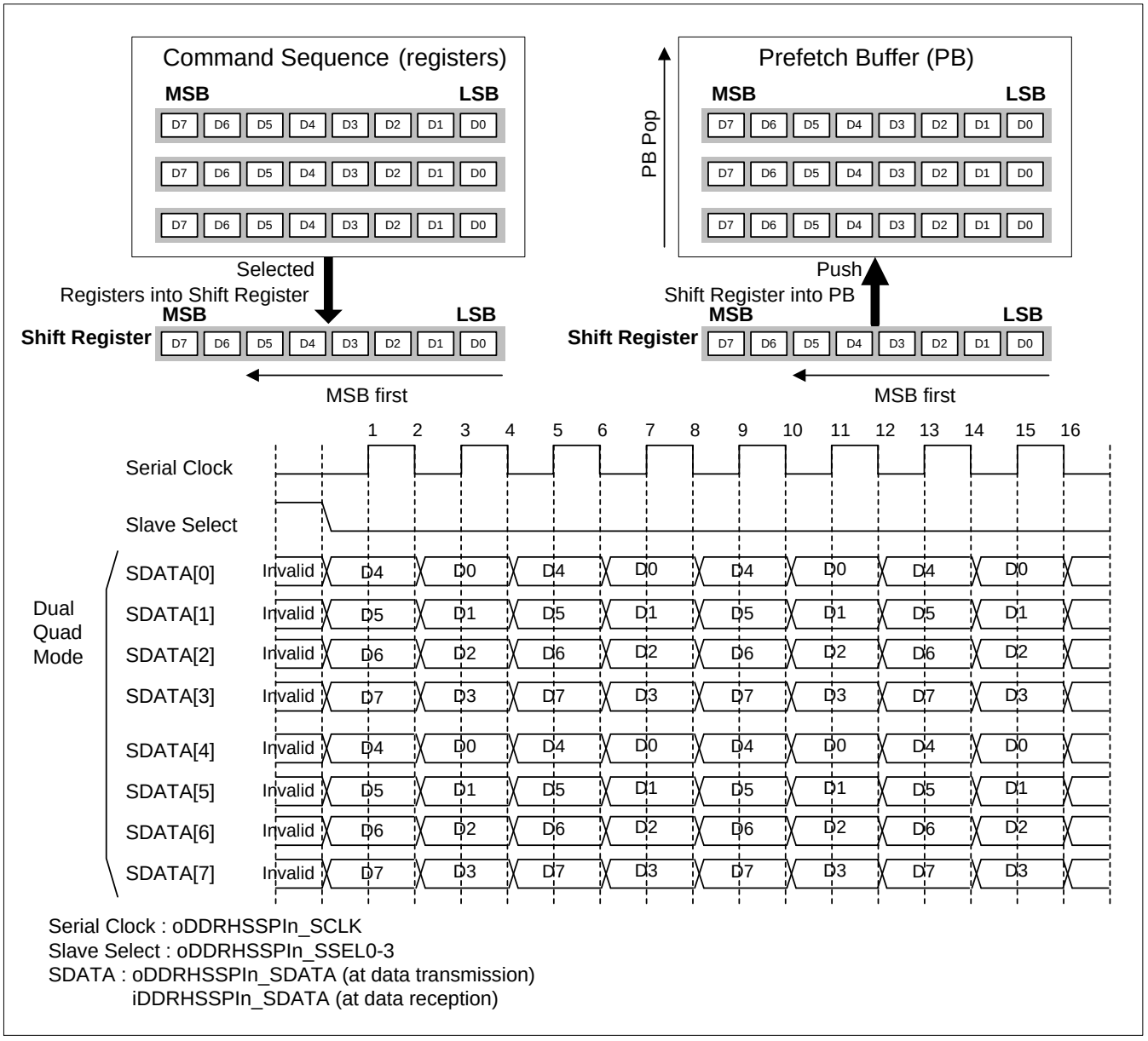


Figure 3-7 Shift Direction (Dual Quad Mode in Command Sequencer Mode)
(Assumptions: DDRHSSPIn_DMFIFOCFG.FWIDTH="00")



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, 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, 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 doesn't 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.

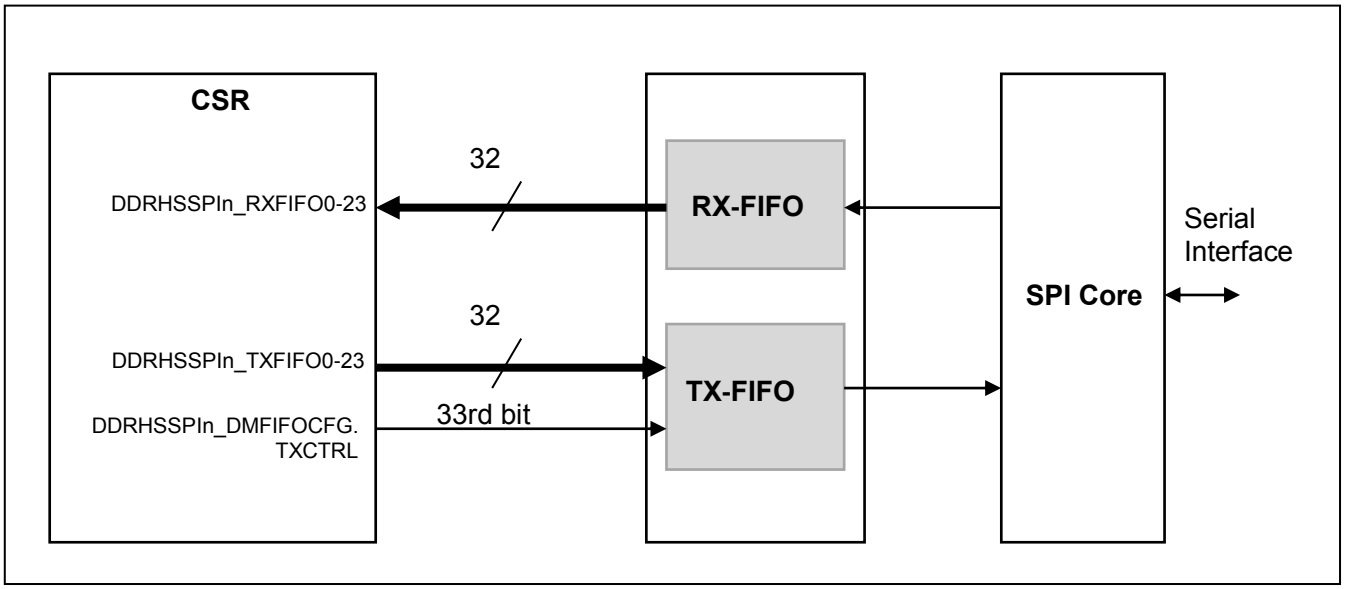
The Shift Register in the SPI Core is 32-bit wide. When the width of the FIFO is changed in the DDRHSSPIn_DMFIFOCFG.FWIDTH, the usable width of the Shift Register also changes accordingly.

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 "11", 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 "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]. (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. **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.

DDRHSSPIn_DMFIFOCFG.TXCTRL	Bit 12 of TX-FIFO Data (TXDATA[12])	Bit 11 of TX-FIFO Data (TXDATA[11])	Description
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: "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). 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.

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]: "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): − 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="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". (Continued)

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	**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 – 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".

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 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.

(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				
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				
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				
Available	Available	"00" (Legacy)	"00" (Legacy)	SDATA[1]	SDATA[0]
N/A	N/A	"00" (Legacy)	"10" (Quad)	Not allowed	Not allowed
Available	Available	"10" (Quad)	"00" (Legacy)	SDATA[1]	SDATA[0]
Available	Available	"10" (Quad)	"10" (Quad)	Not allowed	SDATA[3:0]
N/A	Available	"11" (Dual Quad)	"00" (Dual Legacy)	SDATA[1], SDATA[5]	SDATA[0], SDATA[4]
N/A	Available	"11" (Dual Quad)	"10" (Quad)	Not allowed	Not allowed
N/A	Available	"11" (Dual Quad)	"11" (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 "11" (32-bit wide mode), then only 32-bit access to the FIFO is allowed.

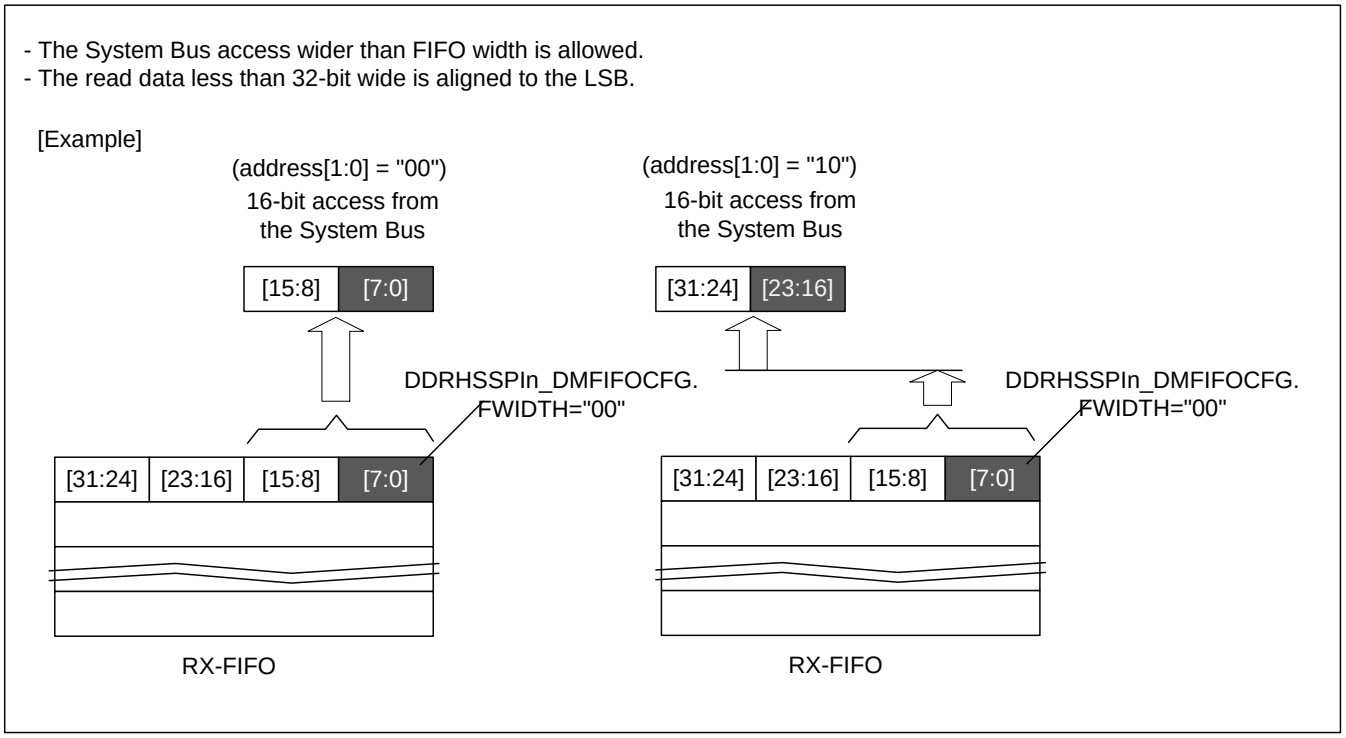
If DDRHSSPIn_DMFIFOCFG.FWIDTH is set to "01" (16-bit wide mode), then 16-bit and 32-bit accesses to the FIFO are allowed. A 32-bit access covers a whole word line for one FIFO entry. A 16-bit access with any alignment covers lower 16 bits of one FIFO entry.

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.

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.

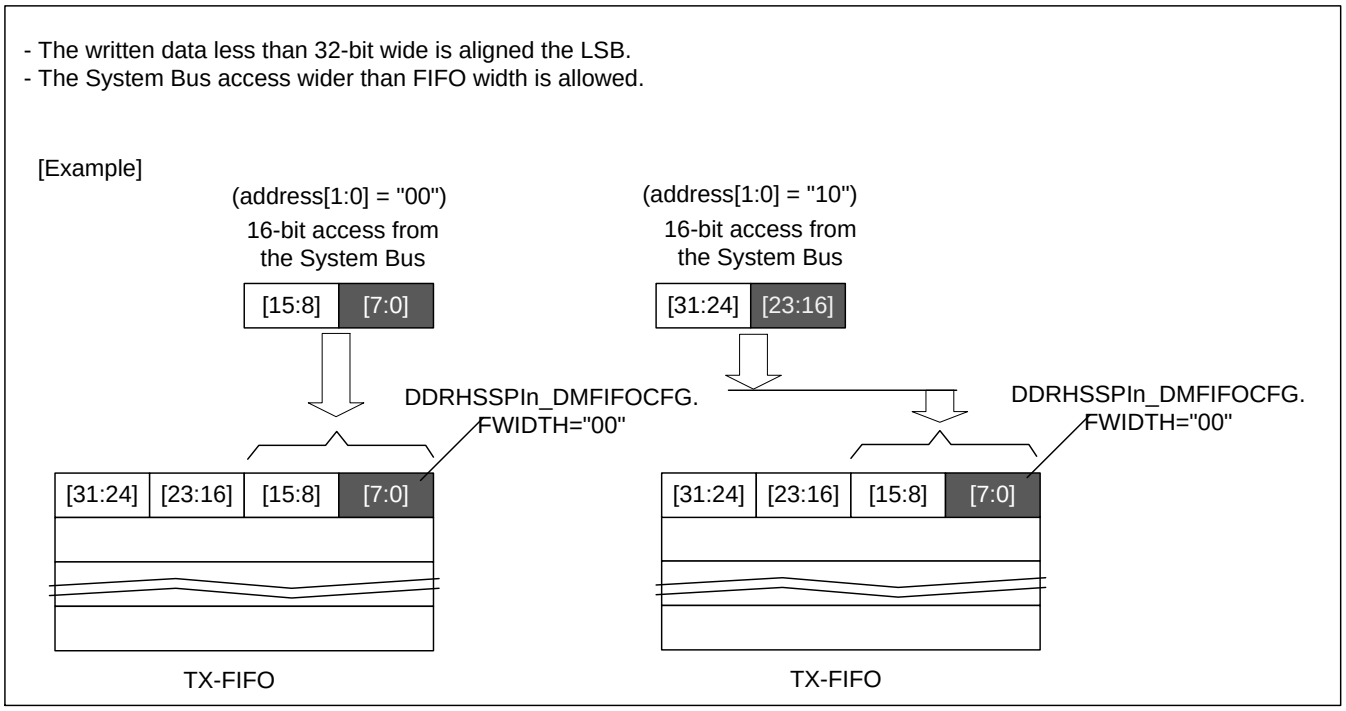
When the System Bus access is wider than RX-FIFO width (configured by DDRHSSPIn_DMFIFOCFG. FWIDTH), the width of read data follows FWIDTH value, and the read data is aligned to the LSB. Figure 4-2 shows this alignment and a notice of the System Bus address.

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. (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 "00" (8-bit wide mode) is not allowed.

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 to the system through 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.

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) 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. 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 set to "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, whenever 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]="00"

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
- The mode of operation is switched from Direct Mode to Command Sequencer Mode

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:

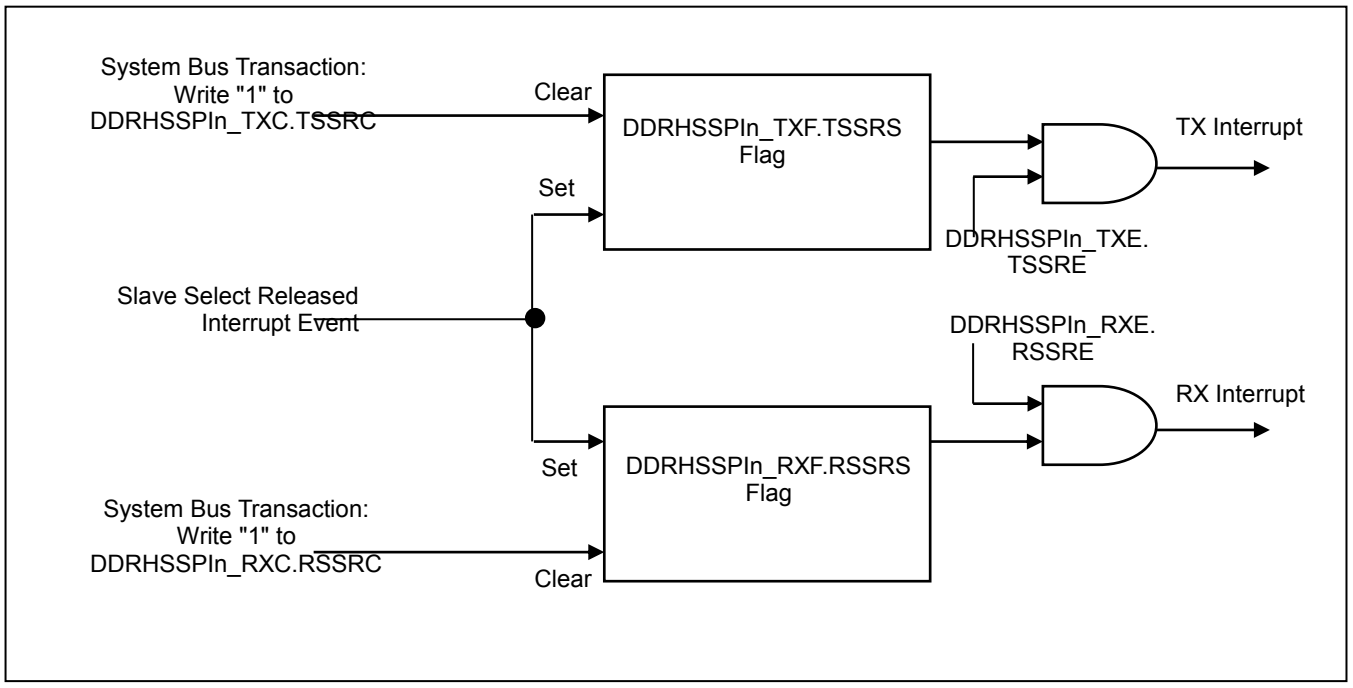
- The DMA Write Channel Service Request has been acknowledged by the DMA controller, OR
- DDRHSSPIn_DMAEN.TXDMAEN="0", OR
- DDRHSSPIn_MCTRL.MES="0", OR
- The mode of operation is switched from Direct Mode to Command Sequencer Mode

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 separate Interrupt-Clear and Interrupt-Enable bits. The interrupt flags are routed onto separate 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.
  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.
  If Byte Counter Mode is used for controlling the transfer length, then the DDRHSSPIn_DMBCS Register will be loaded with the value in DDRHSSPIn_DMBCC Register immediately when 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, it is necessary to avoid the halting of serial transfer. 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, during 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 the 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.

Note:

- *When in TX-and-RX Mode, please avoid halting the serial transfer.*

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", that omits the instruction bit field and starts with address field from 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 that 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. 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.*

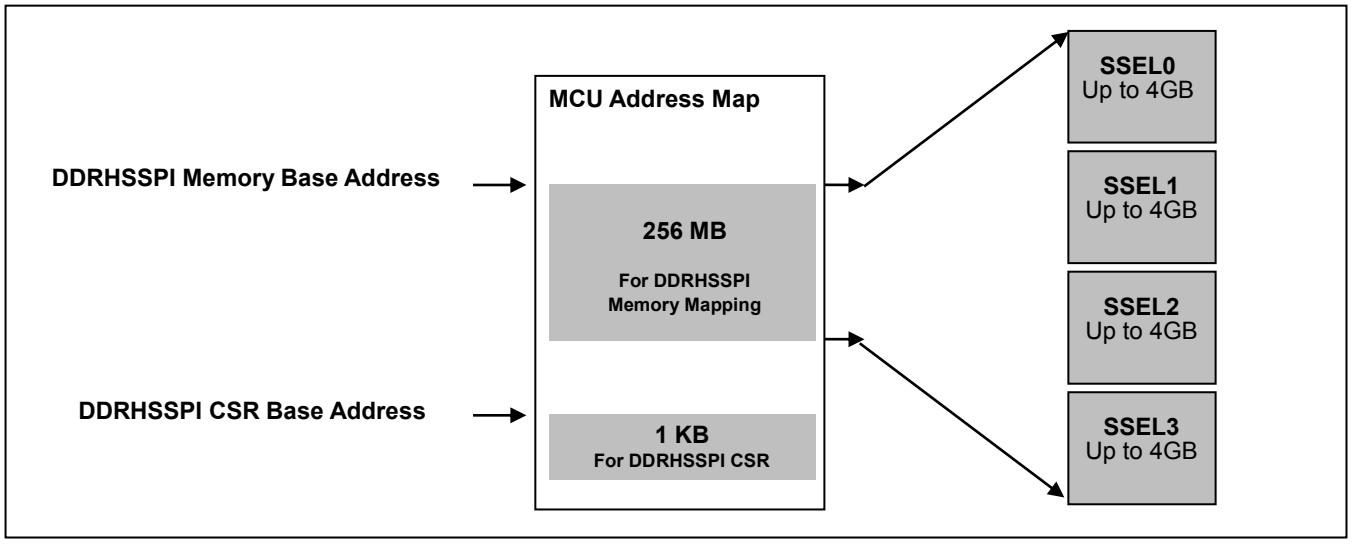
Memory Mapping

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 the few 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 is virtually mapped to 16GB 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 Address 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.

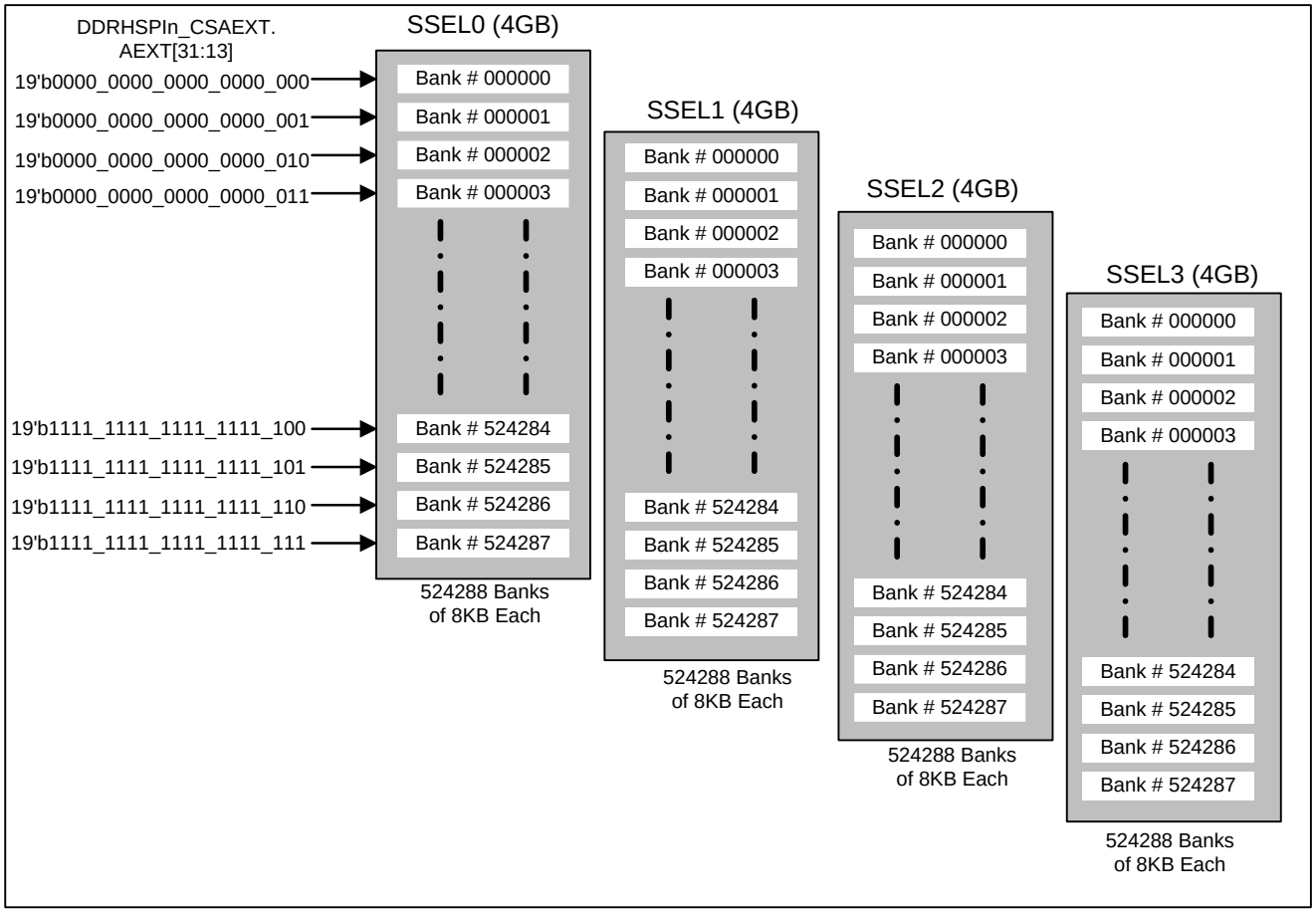
Even if the address of the System Bus access is in the valid range, if the target Slave Select is not set valid, the DDRHSSPIn_FAULTF.UMAFS flag is set as well. For example, if DDRHSSPIn_CSCFG.SSEL0EN is "0", any memory access corresponding to the Slave Select 0 ends up asserting an error of DDRHSSPIn_FAULTF.UMAFS = "1".

Generation of 32-bit Memory Address

The Address Extension Mechanism allows the 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.

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".

Figure 4-6 Addressing 4GB Serial Flash Memories on Each Slave Select, Through Different Banks (DDRHSSPIn_CSCFG.MSEL="0000")



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.

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.

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.

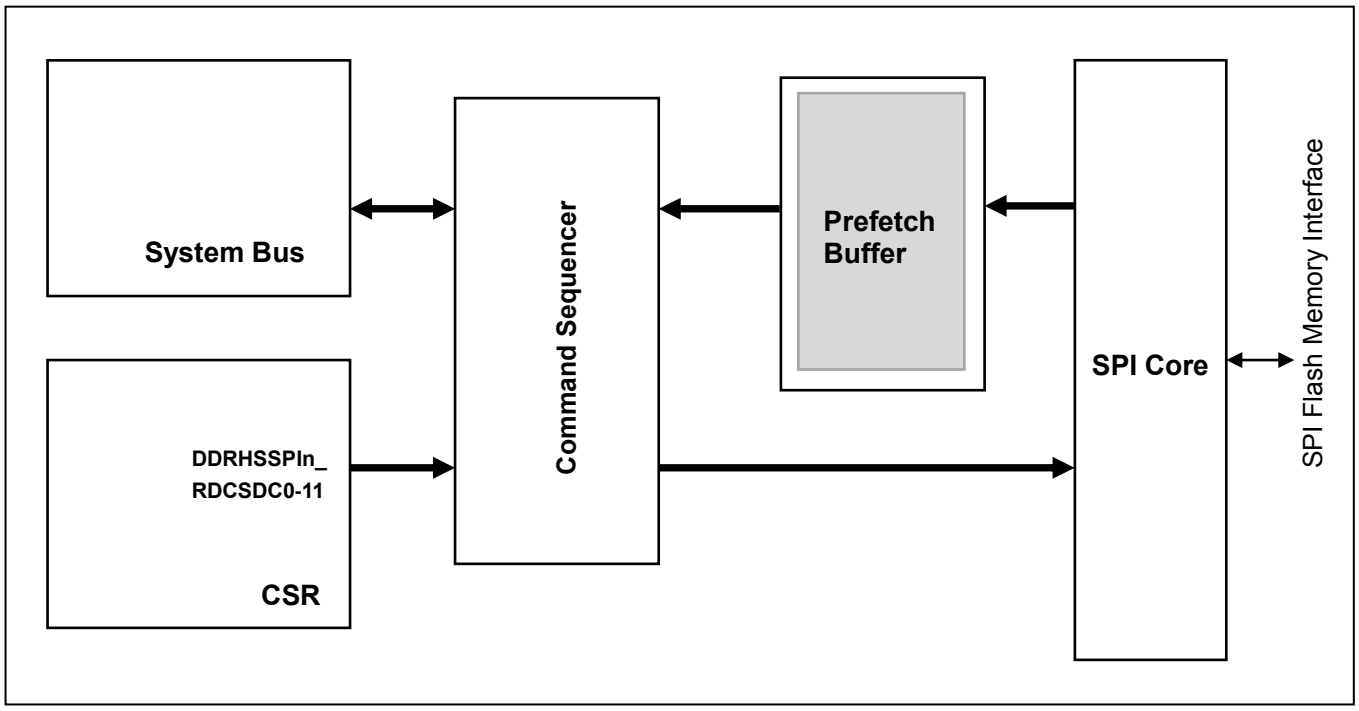
When a byte or halfword is read at a random (non-sequential) address, the Command Sequence shall be generated at the aligned word address which contains this byte or halfword. In any access scenario where the required data can be read from the top most position of the Prefetch Buffer, there will be no 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:

- 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 maximum throughput, 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 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.

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:

- 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).

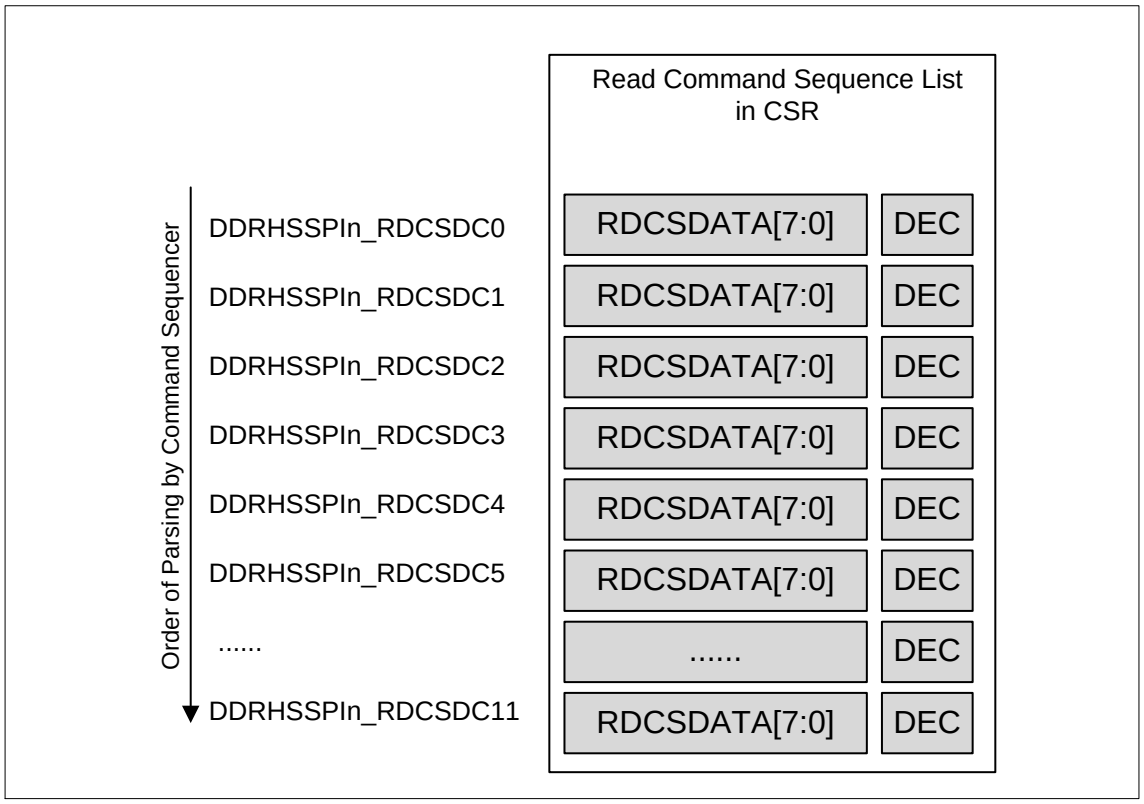
Configuration of Command Sequencer Mode in the CSR

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	0000_0000	Transmit address bits [7:0] of the Serial Flash Memory to be accessed
1	0000_0001	Transmit address bits [15:8] of the Serial Flash Memory to be accessed
1	0000_0010	Transmit address bits [23:16] of the Serial Flash Memory to be accessed
1	0000_0011	Transmit address bits [31:24] of the Serial Flash Memory to be accessed
1	XXXX_X100	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.
1	0000_0111	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	0000_0002	1	address bits [23:16] of the Serial Flash Memory
DDRHSSPIn_RDCSDC1	0000_0001	1	address bits [15:8] of the Serial Flash Memory
DDRHSSPIn_RDCSDC2	0000_0000	1	address bits [7:0] of the Serial Flash Memory
DDRHSSPIn_RDCSDC3	byte data	0	Mode code (*1)
DDRHSSPIn_RDCSDC4	XXXX_X100	1	Dummy cycles Bits at "X" are for the value to control the length of dummy cycles.
DDRHSSPIn_RDCSDC5	0000_0111	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	0000_0003	1	address bits [31:24] of the Serial Flash Memory
DDRHSSPIn_RDCSDC1	0000_0002	1	address bits [23:16] of the Serial Flash Memory
DDRHSSPIn_RDCSDC2	0000_0001	1	address bits [15:8] of the Serial Flash Memory
DDRHSSPIn_RDCSDC3	0000_0000	1	address bits [7:0] of the Serial Flash Memory
DDRHSSPIn_RDCSDC4	byte data	0	Mode code (*1)
DDRHSSPIn_RDCSDC5	XXXX_X100	1	Dummy cycles Bits at "X" are for the value to control the length of dummy cycles.
DDRHSSPIn_RDCSDC6	0000_0111	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 "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.
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.
20. Set the appropriate value to DDRHSSPIn_CSITIME.ITIME.
21. Set the appropriate value to DDRHSSPIn_CSAEXT.AEXT.
22. Write appropriate values to DDRHSSPIn_RDCSDC0-11 Registers according to Table 4-7 or Table 4-8.
23. 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.
24. 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.
25. Operate steps from 1 to 17 in Direct Mode to get out of current Continuous Read Command Sequence. In this operation, please write a different code from the step 10. About the exact code, please refer to the data sheet of the Serial Flash Memory about the mode code.
26. 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 accesses are 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 is full.

In case of new System Bus transfers for the Serial Flash Memory reads (in Command Sequencer Mode), two scenarios are possible as below:

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, 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 doesn't 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

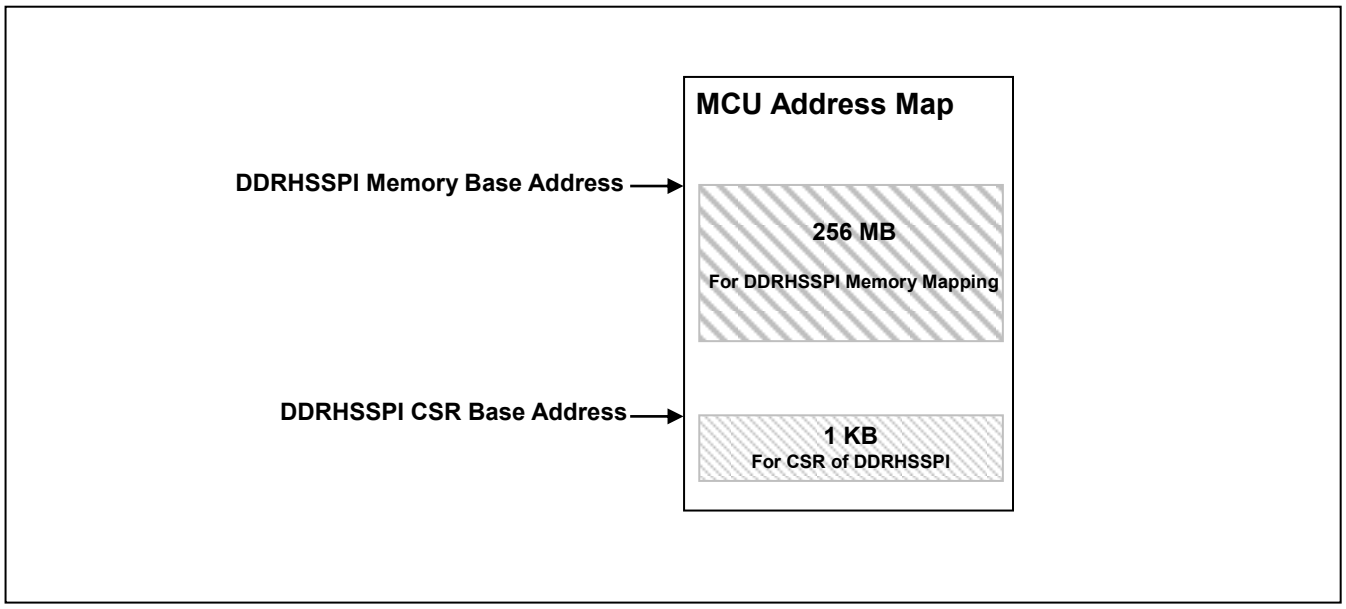
The DDRHSSPI is allocated 256MB of MCU's address space for memory mapping of the external Serial Flash Memories, using Command Sequencer Mode. An additional 1KB of MCU's address space is reserved for mapping the internal Configuration and Status Registers (i.e. CSRs) of the DDRHSSPI.

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



- 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 space.

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.

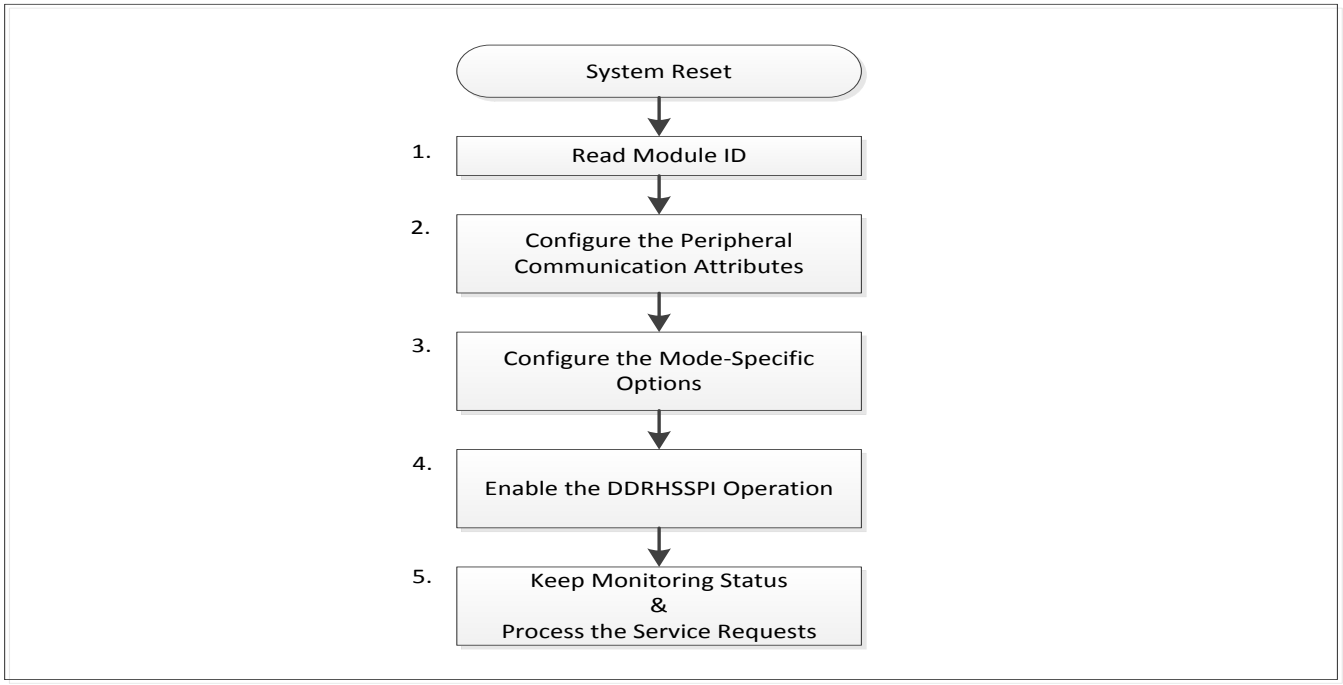
Guidelines on typical use cases of DDRHSSPI

- Any serial-transaction related parameters and control bits (DDRHSSPIn_DMCFG, DDRHSSPIn_DMBCC, DDRHSSPIn_DMBCS, DDRHSSPIn_DMTRP, DDRHSSPIn_DMPSEL and DDRHSSPIn_DMFIFOCFG) should not be performed 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.
- 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.
- 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 under run. In case the FIFOs get overrun or under run, 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, 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 remainder data remains in the RX Shift Register 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 remainder data from the RX Shift Register is pushed into the RX-FIFO before DDRHSSPI releases the Slave Select line. The remainder data is pushed into the RX-FIFO irrespective of whether the RX Shift Register is filled completely or partially. Thus, in Direct Mode, remainder 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.

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

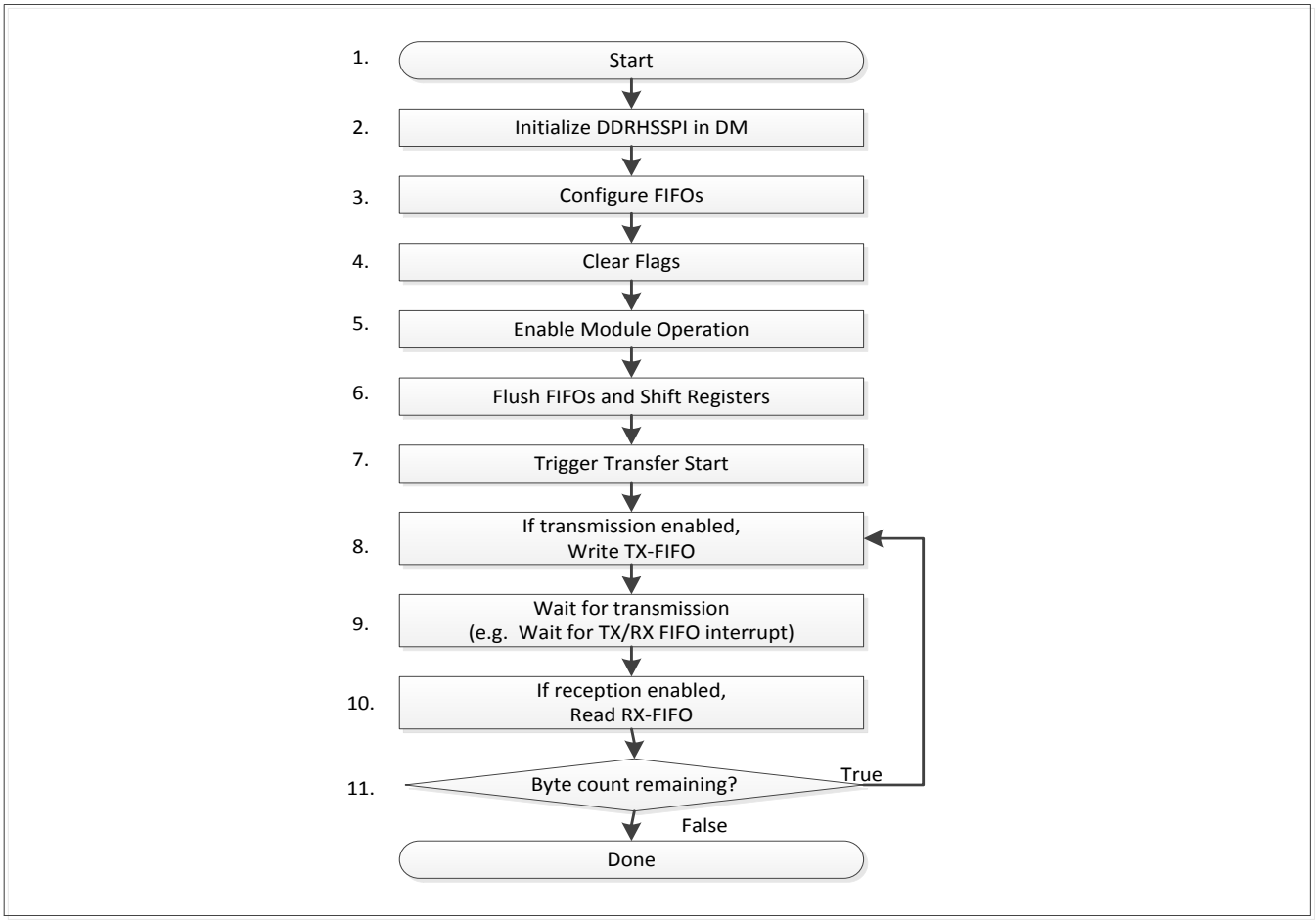


1. After the System Reset, the software shall detect the Module ID number of DDRHSSPI, by reading the DDRHSSPIn_MID Register. This would help it in identifying the attributes and capabilities supported by the DDRHSSPI.
2. The next step is to configure the Attributes related to the Peripheral Communication with the Serial Flash Memory(Memories) connected with DDRHSSPI. DDRHSSPI can be interfaced with up to 4 Serial Flash Memories. 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. 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, through 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).
4. Only after all module-specific configurations are programmed, the DDRHSSPI shall be enabled (by setting the DDRHSSPIn_MCTRL.MEN to "1").
5. Once the DDRHSSPI is enabled, its normal working 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 service those requests, in order to ensure the normal working of DDRHSSPI.

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, DDRHSSPIn_PCC1, DDRHSSPIn_PCC2 and DDRHSSPIn_PCC3 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 both for transmission and reception) in the DDRHSSPIn_DMTRP.TRP. DDRHSSPI loads the DDRHSSPIn_DMBCC.BCC with the number of bytes to be serially transferred.
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 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. 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 data to be transmitted into the TX-FIFO via DDRHSSPIn_TXFIFO0-23 Register address. Before writing to the DDRHSSPIn_TXFIFO0-23 Register, modify the value of the DDRHSSPIn_DMFIFOCFG. TXCTRL bit appropriately. Generally (i.e. when the data being written 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 DMA Controller.
10. If reception is enabled in DDRHSSPIn_DMTRP Register, 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 resetting DDRHSSPIn_MCTRL.MEN bit to "0"). The software can check the status bit DDRHSSPIn_TXF.TSSRS, to see if the current transfer has finished.

Using the DDRHSSPI in Command Sequencer Mode of Operation

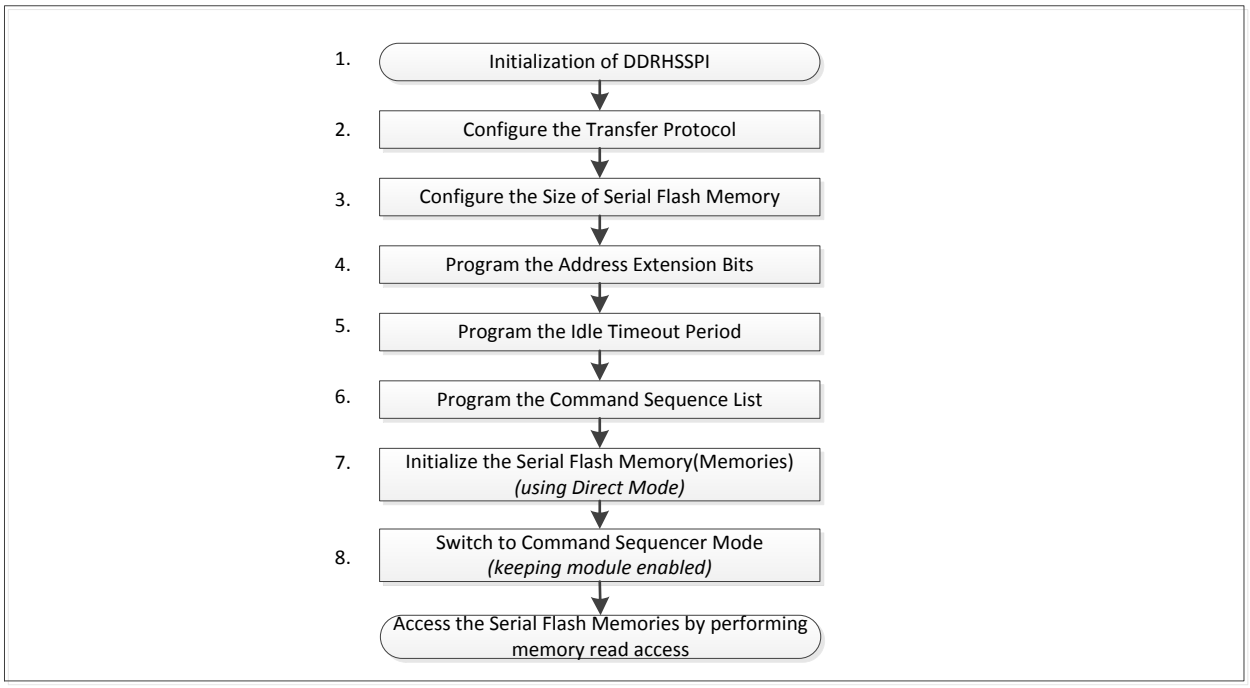
- Using the Memory Mapped Memories

  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 same family, but with 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 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. This is because a Serial Flash Memory which has 24-bit addressing cannot be used with a bit-stuffed 32-bit address - its address phase 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". The Serial Flash Memory can be programmed in the Continuous Read Command Sequence, using the Direct Mode operation of DDRHSSPI.
  - When Command Sequencer Mode is used, internal Prefetch Buffer will be used. Since in Command Sequencer Mode only memory reads are supported there is only Prefetch Buffer available in Command Sequencer Mode. This buffer is 48 location deep, 32 bit each.
  - 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 there is address miss with next System Bus transfer then Prefetch Buffer flush is performed.
  - 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.
- 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

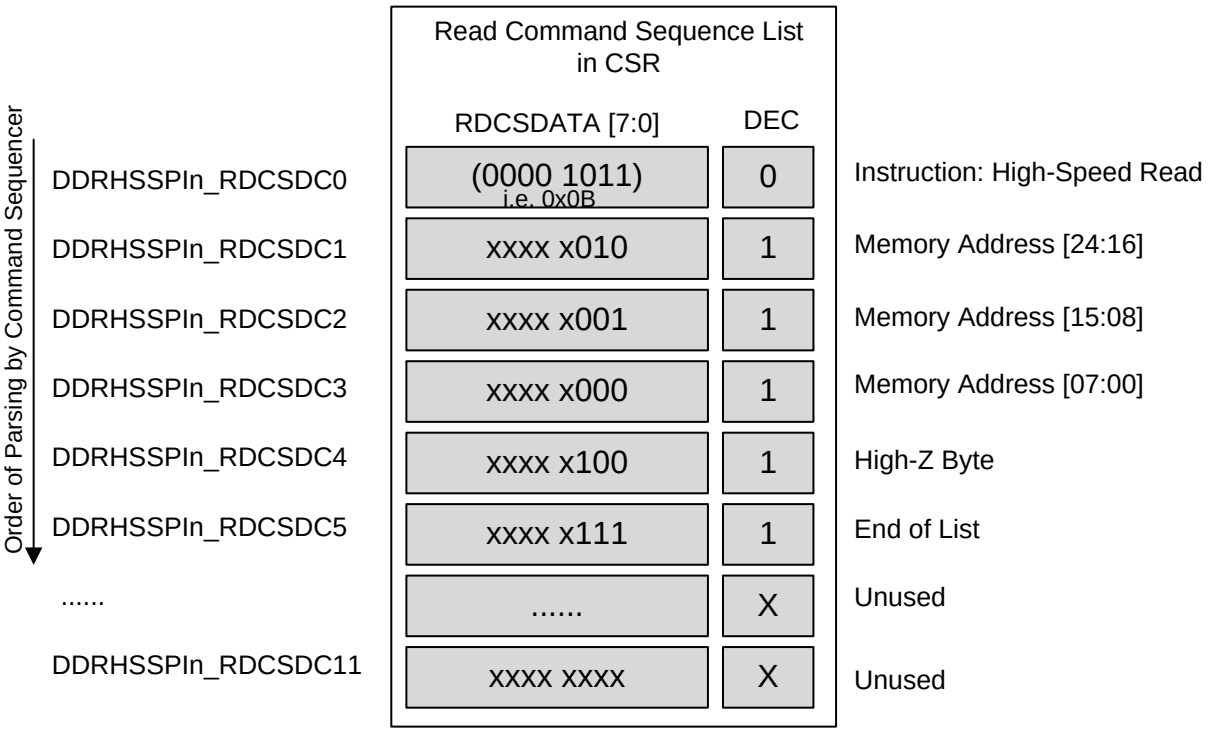


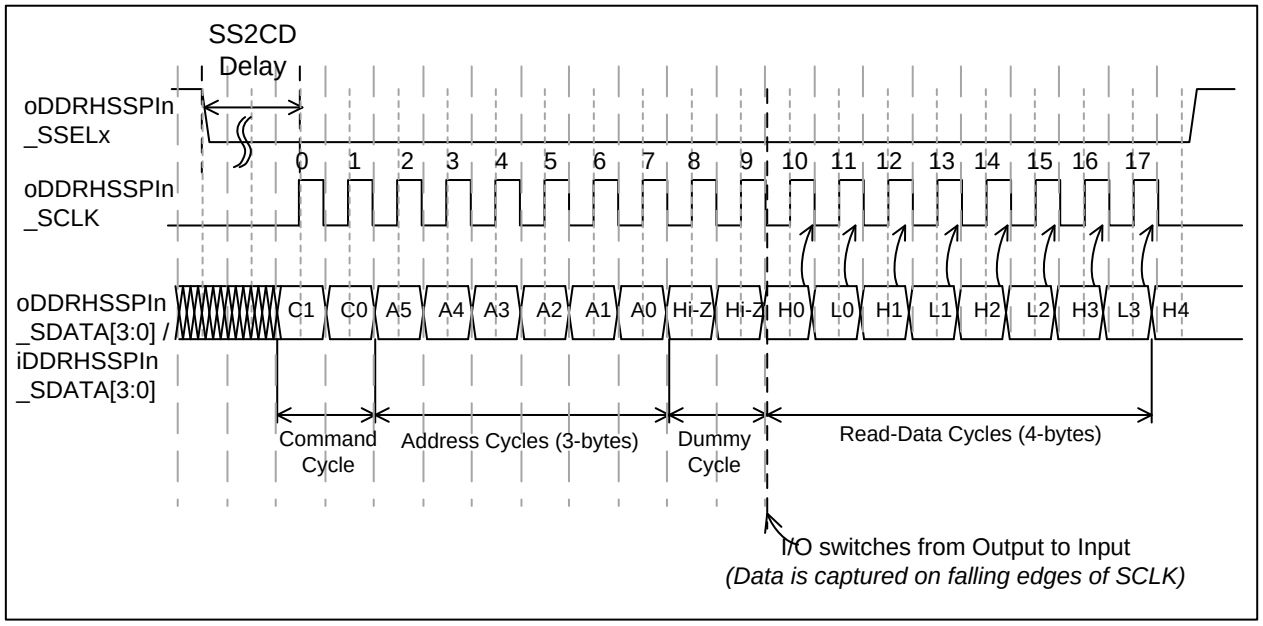
1. After the System Reset, the software shall initialize the DDRHSSPI by setting the Peripheral Communication related attributes in the DDRHSSPIn_PCC0-3 Registers. It is very important that these attributes shall be the same as being used by the Serial Flash Memory with which DDRHSSPI is interfaced. When Serial Flash Memories are to be memory-mapped using Command Sequencer Mode, all Serial Flash Memories shall be of 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_DMFIFOCFG.DDRM bit.
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 4.2 for 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 4.2 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 for the Serial Flash Memory, 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. 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.

- Timing Diagram for Command Sequencer

  Figure 4-13 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 4-13 shows also the corresponding timing diagram for a Read Command Sequence, for Mode 4.

Figure 4-13 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 doesn't mean the actual example of the SPI transaction for Serial Flash Memory.*

4.4.1. Notice On Document Designations

Cypress Inc. 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.

4.4.2. 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.

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
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	5.20
DDRHSSPIn_TXFIFO22	DDRHSSPI TX-FIFO Register 22	
DDRHSSPIn_TXFIFO23	DDRHSSPI TX-FIFO Register 23	
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	

Abbreviation	Register name	See
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	
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

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	R0/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.

[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.

SSELDEASRT[4:0]	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.

CDRS[3:0]	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 + 0.5) 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.

[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 doesn't 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 isn't 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 isn't 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 isn't 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) aren't 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 isn't 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 isn't 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.

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 doesn't check the DLP data.
1	DDRHSSPIn_DLP.DLP doesn't 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 doesn't 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 isn't 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 isn't less than or equal to the configured RX-FIFO threshold value.
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 isn't 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 isn't 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 isn't 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 doesn't detect the DLP error.
1	DDRHSSPI detects the DLP error.

The DLP error fault occurs if the DDRHSSPI receives 8 bit DLP value that isn't 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 doesn't 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 doesn't 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 doesn't 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.

PSEL[1:0]	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.

TRP[3:0]	Description
0000	TX-and-RX in Legacy Mode is configured.
0011	This value is available only when DDRHSSPIn_MID.MID is 0x00000100. 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. 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 don't 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 don't 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.

FWIDTH[1:0]	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="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.

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="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.

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="00"), 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] = "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.
  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	R0,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0 or 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 R1,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0 or 1

[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.

MID[31:0]	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 "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".
0x00000100	This version supports the following features. − Octal protocol (Dual Quad and Dual Legacy) − Sample Point Control function − DLP function

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.

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	SDATASMP TCNT[5]	SDATASMP TCNT[4]	SDATASMP TCNT[3]	SDATASMP TCNT[2]	SDATASMP TCNT[1]	SDATASMP TCNT[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.

5.26. DDRHSSPI SDATA Left Clock Sample Point Registers (DDRHSSPIn_SDATASAMPLEPTLFT0-7)

This feature is available only when DDRHSSPIn_MID.MID is 0x00000100.

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	SDATASMP TLFT[5]	SDATASMP TLFT[4]	SDATASMP TLFT[3]	SDATASMP TLFT[2]	SDATASMP TLFT[1]	SDATASMP TLFT[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.

5.27. DDRHSSPI SDATA Right Clock Sample Point Registers (DDRHSSPIn_SDATASAMPLEPTRGH0-7)

This feature is available only when DDRHSSPIn_MID.MID is 0x00000100.

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.

5.28. DDRHSSPI Data Learning Pattern Register (DDRHSSPIn_DLP)

This feature is available only when DDRHSSPIn_MID.MID is 0x00000100.

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.

5.29. DDRHSSPI Data Learning Pattern Sample Status Register (DDRHSSPIn_DLPSAMPLESTATUS)

This feature is available only when DDRHSSPIn_MID.MID is 0x00000100.

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	DLPSMP LST7R	DLPSMP LST7C	DLPSMP LST7L	Reserved	DLPSMP LST6R	DLPSMP LST6C	DLPSMP LST6L
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	DLPSMP LST5R	DLPSMP LST5C	DLPSMP LST5L	Reserved	DLPSMP LST4R	DLPSMP LST4C	DLPSMP LST4L
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	DLPSMP LST3R	DLPSMP LST3C	DLPSMP LST3L	Reserved	DLPSMP LST2R	DLPSMP LST2C	DLPSMP LST2L
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	DLPSMP LST1R	DLPSMP LST1C	DLPSMP LST1L	Reserved	DLPSMP LST0R	DLPSMP LST0C	DLPSMP LST0L
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

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.

MBM[1:0]	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. 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 don't 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: External Bus Interface

This chapter explains the functions and operation of the External Bus Interface (EBI).

1. **Outline of External Bus Interface**
2. **External Bus Interface Configuration registers**
3. **Operation of External Bus Interface**
4. **External memory connection to External Bus Interface**
5. **Access waveform examples for External Bus Interface**

EBI_NONSD-T01P00R01L00-E1-GBM01

1. Outline of External Bus Interface

This section describes the features and the block diagram of the External Bus Interface.

External Bus Interface is a generic memory controller, which can access external memories like SRAM/FLASH. It can support access to eight SRAM/FLASH memory.

Features of External Bus Interface

- Supports 8/16/32-bit width SRAM/FLASH
- Supports bi-endian access for SRAM/FLASH
- 8 chip selects for SRAM
- Device can be accessed during NAND FLASH accessing (no need for exclusive access control)
- Supports NOR FLASH page accesses

Abbreviations

This section lists the terms and abbreviations used in this chapter.

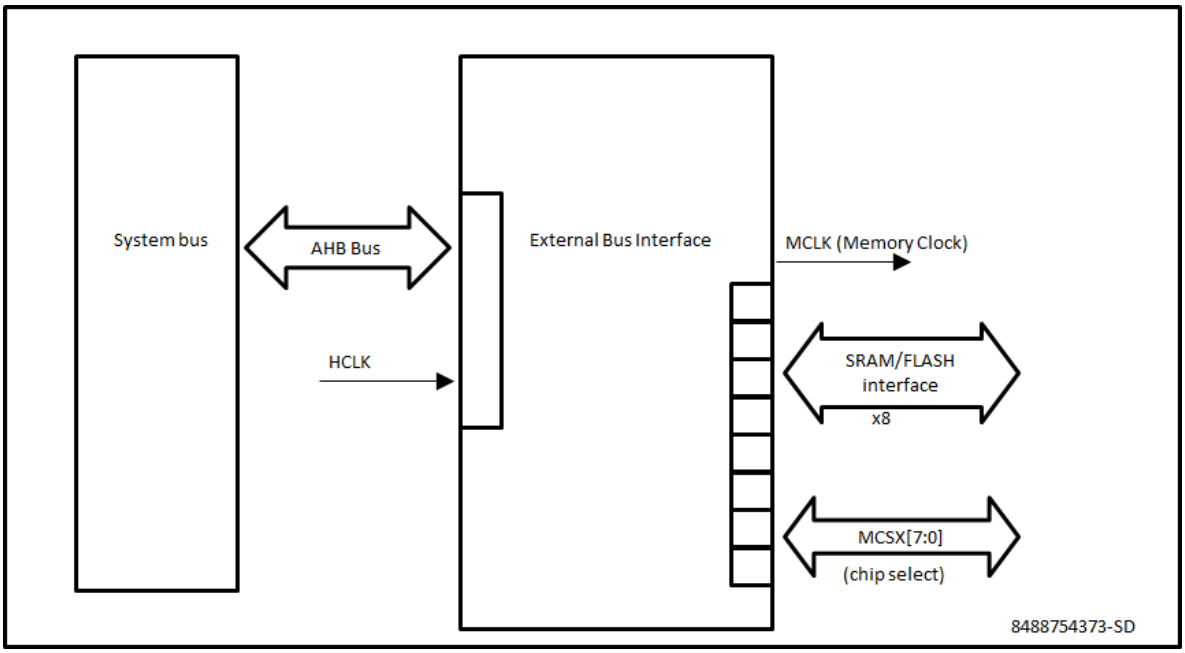
Table 1-1 Terms and Abbreviations

Term	Meaning
BA	Base Address
EBI	External Bus Interface
SRAM	Static Random Access Memory
IF	Interface

Block Diagram of External Bus Interface

The Figure 1-1 shows the block diagram of the External Bus Interface.

Figure 1-1 Block Diagram of External Bus Interface



1.1. Pad Interface

Table 1-2 External I/F pins of EBI module external interface details.

Table 1-2 External I/F Pins of EBI Module

Name	IO	Clock Domain	Active	Default (Steady State)	Description
MAD[24:0]	O	MCLK	N/A	N/A	Address output.
MDQM[3:0]	O	MCLK	L	HHHH	Byte mask signal for SRAM.
MWEX	O	MCLK	L	H	Write enable signal for SRAM.
MOEX	O	MCLK	L	H	Read enable for SRAM.
MCLK	O	HCLK	clock	clock	Memory clock. Output of HCLK.
MCSX[7:0]	O	MCLK	L	HHHHHHHHH	Chip select. SRAM/FLASH: MCSX[7:0]
MNALE	O	MCLK	H	L	ALE signal to control NAND Flash.
MNCLE	O	MCLK	H	L	CLE signal to control NAND Flash.
MNWEX	O	MCLK	L	H	Write enable signal to control NAND Flash
MNREX	O	MCLK	L	H	Read enable signal to control NAND Flash
MRDY	I	MCLK	N/A	N/A	Memory Ready.
MRDatb[31:0]	I	MCLK	N/A	N/A	Read Data.
MWDatb[31:0]	O	MCLK	N/A	N/A	Write Data.

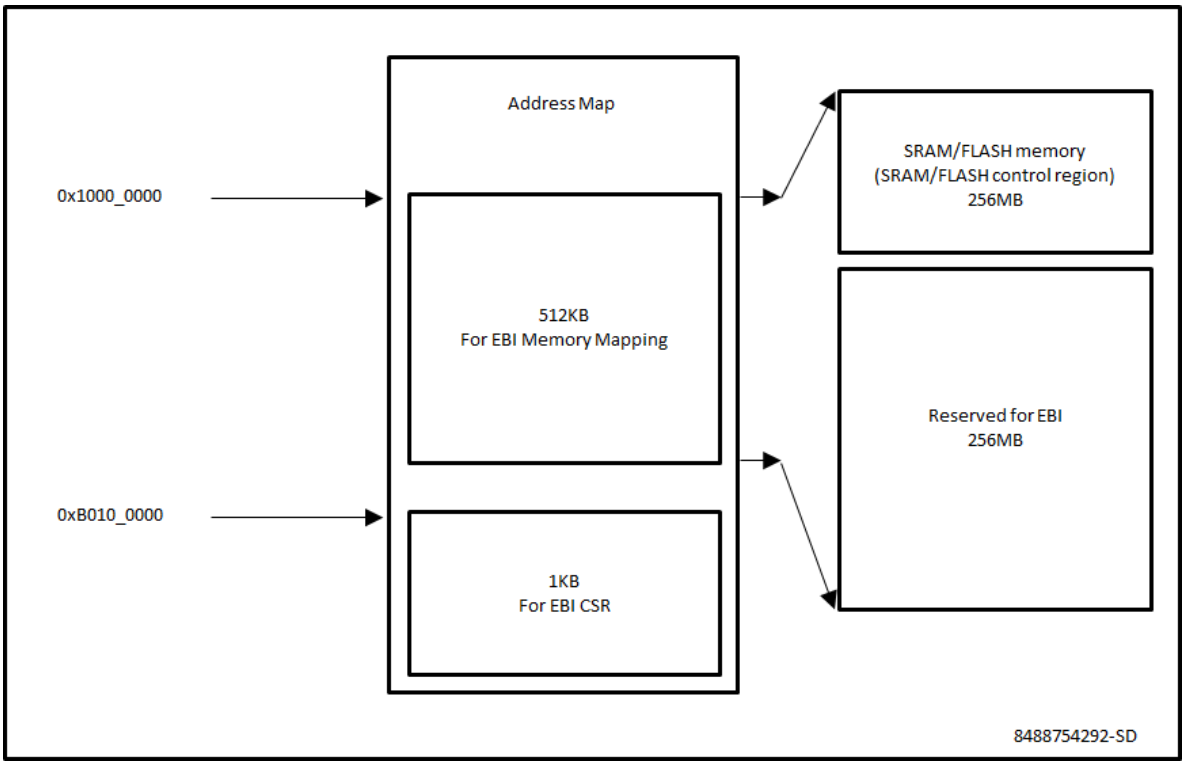
Note:

- *MCLK - Memory clock*
- *HCLK - Operation clock.*

1.2. External Bus Interface and Memory Regions

The External Bus Interface is configured by the setting register region. SRAM/Flash can be accessed via the SRAM/Flash control region.

Figure 1-2 EBI Memory Region (Configuration Register and Control Region)



Note:

- *For details on EBI Memory regions refer to the device datasheet.*

2. External Bus Interface Configuration Registers

This section lists all the External Bus Interface registers and explains their function in detail.

EBI configuration registers can be written with 32-bit accesses only. Write accesses with a size smaller than 32-bit return an error response. EBI configuration registers can be accessed only with correct PPU access settings.

Registers of EBI

- Unlock Register (EBI_UNLOCK)
- Lock Status Register (EBI_LSTSR)
- SRAM/FLASH Mode Control Register (EBI_SFMR0~7)
- SRAM/FLASH Access Configuration Register (EBI_SFACCR0~7)
- SRAM/FLASH Address Control Register (EBI_SFADDCR0~7)
- Error Register (EBI_ERRR)

Figure 2-1 Memory Layout of External Bus Interface Registers

Offset	+3	+2	+1	+0
0x00000000	EBI_UNLOCK 00000000 00000000 00000000 00000000			
0x00000004	EBI_LSTSR 00000000 00000000 00000000 00000001			
0x00000008	EBI_SFMR0 00000000 00000000 00000000 00000000			
0x0000000C	EBI_SFMR1 00000000 00000000 00000000 00000000			
0x00000010	EBI_SFMR2 00000000 00000000 00000000 00000000			
0x00000014	EBI_SFMR3 00000000 00000000 00000000 00000000			
0x00000018	EBI_SFMR4 00000000 00000000 00000000 00000000			
0x0000001C	EBI_SFMR5 00000000 00000000 00000000 00000000			
0x00000020	EBI_SFMR6 00000000 00000000 00000000 00000000			
0x00000024	EBI_SFMR7 00000000 00000000 00000000 00000000			
0x00000028	EBI_SFACCR0 00000101 01011111 11110000 00001111			
0x0000002C	EBI_SFACCR1 00000101 01011111 11110000 00001111			
0x00000030	EBI_SFACCR2 00000101 01011111 11110000 00001111			
0x00000034	EBI_SFACCR3 00000101 01011111 11110000 00001111			
0x00000038	EBI_SFACCR4 00000101 01011111 11110000 00001111			

Offset	+3	+2	+1	+0
0x0000003C	EBI_SFACCR5 00000101 01011111 11110000 00001111			
0x00000040	EBI_SFACCR6 00000101 01011111 11110000 00001111			
0x00000044	EBI_SFACCR7 00000101 01011111 11110000 00001111			
0x00000048	EBI_SFADDCR0 00000000 00001111 00000000 00000000			
0x0000004C	EBI_SFADDCR1 00000000 00001111 00000000 00010000			
0x00000050	EBI_SFADDCR2 00000000 00001111 00000000 00100000			
0x00000054	EBI_SFADDCR3 00000000 00001111 00000000 00110000			
0x00000058	EBI_SFADDCR4 00000000 00001111 00000000 01000000			
0x0000005C	EBI_SFADDCR5 00000000 00001111 00000000 01010000			
0x00000060	EBI_SFADDCR6 00000000 00001111 00000000 01100000			
0x00000064	EBI_SFADDCR7 00000000 00001111 00000000 01110000			
0x00000068	Reserved 00000000 00000000 00010011 00000000			
0x0000006C	Reserved 00000000 00000000 00000000 00101000			
0x00000070	Reserved 00000000 00000000 00000000 00000000			
0x00000074	Reserved 00000000 01000010 00010001 01000001			
0x00000078	Reserved 00000000 00000000 00000000 00000000			
0x0000007C	EBI_ERRR 00000000 00000000 00000000 00000000			

Notes:

- *When EBI_LSTSR.LOCKSTATUS bit is '1', write access to reserved address (Offset 0x00000068 - 0x00000078) causes protection error.*
- *For reserved addresses (Offset 0x00000068 - 0x00000078), it is not allowed to write the value other than initial value.*

  i.e. It is not allowed to:

  - write value other than 0x00001300 to reserved address (Offset 0x00000068)
  - write value other than 0x00000028 to reserved address (Offset 0x0000006C)

- write value other than 0x00000000 to reserved address (Offset 0x00000070)
- write value other than 0x00421141 to reserved address (Offset 0x00000074)
- write value other than 0x00000000 to reserved address (Offset 0x00000078)

2.1. Unlock Register (EBI_UNLOCK)

This register decides the write accessibility of the configuration registers of External Bus Interface module. It has to be written with the specific unlock value (0xEB14_10CE) to reset the EBI_LSTSR:LOCKSTATUS bit, which allows write to the configuration registers. After completion of configuration the EBI_UNLOCK register should be written with specific lock (0x10CE_0EB1) value, which sets the EBI_LSTSR:LOCKSTATUS bit and restricts the write access to the configuration register.

Unlock Register (EBI_UNLOCK)

Figure 2-2 Unlock Register (EBI_UNLOCK)

Bit	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit 24
Field	UNLOCK [31]	UNLOCK [30]	UNLOCK [29]	UNLOCK [28]	UNLOCK [27]	UNLOCK [26]	UNLOCK [25]	UNLOCK [24]
R/W attribute	R0,W							
Protection attribute	-							
Initial value	0	0	0	0	0	0	0	0

Bit	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit 16
Field	UNLOCK [23]	UNLOCK [22]	UNLOCK [21]	UNLOCK [20]	UNLOCK [19]	UNLOCK [18]	UNLOCK [17]	UNLOCK [16]
R/W attribute	R0,W							
Protection attribute	-							
Initial value	0	0	0	0	0	0	0	0

Bit	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit 8
Field	UNLOCK [15]	UNLOCK [14]	UNLOCK [13]	UNLOCK [12]	UNLOCK [11]	UNLOCK [10]	UNLOCK [9]	UNLOCK [8]
R/W attribute	R0,W							
Protection attribute	-							
Initial value	0	0	0	0	0	0	0	0

Bit	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit 0
Field	UNLOCK [7]	UNLOCK [6]	UNLOCK [5]	UNLOCK [4]	UNLOCK [3]	UNLOCK [2]	UNLOCK [1]	UNLOCK [0]
R/W attribute	R0,W							
Protection attribute	-							
Initial value	0	0	0	0	0	0	0	0

[bit31:0] UNLOCK Bits

This Unlock Register protects the EBI registers from being modified accidentally by software. The configuration and exception vector registers cannot be written until this register has been written with unlock (0xEB14_10CE) value. To lock EBI registers again, software must write lock (0x10CE_0EB1) value. Illegal access to configuration registers or writing value other than lock/unlock value to this register causes protection error. The list of registers in EBI, which are protected by this register are:

EBI_SFMR0~7

EBI_SFACCR0~7

EBI_SFADDCR0~7

EBI_ERRR

Read to this register always returns '0'.

2.2. Lock Status Register (EBI_LSTSR)

This register provides the status of External Bus Interface lock feature.

Lock Status Register (EBI_LSTSR)

Figure 2-3 **Lock Status Register (EBI_LSTSR)**

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

Bit	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit 16
Field	Reserved							
R/W attribute	R0,WX							
Protection attribute	-							
Initial value	0	0	0	0	0	0	0	0

Bit	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit 8
Field	Reserved							
R/W attribute	R0,WX							
Protection attribute	-							
Initial value	0	0	0	0	0	0	0	0

Bit	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit 0
Field	Reserved							LOCKSTATUS
R/W attribute	R0,WX							R,WX
Protection attribute	-							
Initial value	0	0	0	0	0	0	0	1

[bit31:8] Reserved Bits

[bit7:1] Reserved Bits

[bit0] LOCKSTATUS Bit

This bit provides locked/unlocked status of External Bus Interface configuration registers. By default this bit is '1', writing the correct unlock value to EBI_UNLOCK:UNLOCK register resets this bit to '0'. Writing the correct lock value to EBI_UNLOCK:UNLOCK register sets this bit to '1'.

Bit	Description
0	Configuration registers are unlocked for write access
1	Configuration registers are locked for write access

2.3. SRAM/FLASH Mode Control Register (EBI_SFMR0~7)

This register provides the mode configuration for SRAM/FLASH. EBI_SFMRn:RDY, EBI_SFMRn:PAGE and EBI_SFMRn:NAND should be set mutual exclusive. Each of these 8 registers are identical in function. Only one register (i.e. EBI_SFMR0) is explained here.

SRAM/FLASH Mode Control Register (EBI_SFMR0)

Figure 2-4 SRAM/FLASH Mode Control Register (EBI_SFMR0)

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

Bit	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit 16
Field	Reserved							
R/W attribute	R0,WX							
Protection attribute	WS							
Initial value	0	0	0	0	0	0	0	0

Bit	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit 8
Field	Reserved							
R/W attribute	R0,WX							
Protection attribute	WS							
Initial value	0	0	0	0	0	0	0	0

Bit	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit 0
Field	ENDIANESS	RDY	PAGE	NAND	WEOFF	RBMON	WDTH[1]	WDTH[0]
R/W attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Protection attribute	WS							
Initial value	0	0	0	0	0	0	0	0

[bit31:8] Reserved Bits

[bit7] ENDIANESS Bit

These bits provide endianess control for 8 SRAM/FLASH memory locations.

Bit	Description
0	SRAM/FLASH memory bank 0 is in little endian mode for access
1	SRAM/FLASH memory bank 0 is in big endian mode for access

[bit6] RDY Bit

Set this bit to '1' to have a handshake with a low-speed peripheral that uses MRDY signal. At least, four or more cycles (MCLK) must be inserted between the falling edge of MRDY and the falling edge of the read/write strobe signal. Set this bit to '0' when a device such as an SRAM memory is not using the ready signal is accessed.

Bit	Description
0	Ready mode OFF
1	Ready mode ON

Note:

- *The behavior of EBI is undefined if EBI_SFMR0:NAND, EBI_SFMR0:PAGE and RDY are not set mutually exclusive.*

[bit5] PAGE Bit

This bit controls the NOR FLASH page access mode. In the NOR FLASH page access mode, the first address cycle is issued according to the setting of the first read address cycle (EBI_SFACCR0:FRADC). After that, accesses are continuously executed until it reaches 16 bytes boundary according to the setting of the read access cycle (EBI_SFACCR0:RACC). When this mode is selected, set EBI_SFMR0:RBMON to '0' and the read address cycle (EBI_SFACCR0:RADC) to '0'.

Bit	Description
0	Ready mode OFF
1	Ready mode ON

Note:

- *The behavior of EBI is undefined if EBI_SFMR0:NAND, EBI_SFMR0:PAGE and RDY are not set mutually exclusive.*

[bit4] NAND Bit

This bit controls the mode to connect NAND FLASH. To enable the access to NAND FLASH, this bit must be set to '1'. In NAND FLASH mode, the corresponding MCSX is fixed to '0', and NAND FLASH dedicated pin is used in accesses after this. Reset this bit to '0' when NAND is not used, and the MCSX is fixed to '1', and the NAND FLASH is kept in a power saving state.

Bit	Description
0	NAND FLASH mode OFF
1	NAND FLASH mode ON

Note:

- *The behavior of EBI is undefined if EBI_SFMR0:PAGE, EBI_SFMR0:RDY and NAND are not set mutually exclusive.*

[bit3] WEOFF Bit

Disables write enable signal (MWEX) operation. Saves current consumption by prohibiting unnecessary operation of MWEX when using the byte mask signal (MDQM) as the write enable of the device. When disabled, MWEX is fixed to '1'.

bit	Description
0	Enable
1	Disable

[bit2] RBMON Bit

Enables byte mask signal (MDQM) at read access. Controls unnecessary data outputs from devices on which byte mask is possible, saves power consumed by the total accessing.

Bit	Description
0	Enable
1	Disable

[bit1:0] WDTH Bits

Data bit width of a connected device is specified.

Bit	Description
00	8 bits
01	16 bits
10	32 bits
11	Reserved

2.4. SRAM/FLASH Access Configuration Register (EBI_SFACCR0~7)

This register provides the configuration bits to control the access timing for SRAM/ FLASH memory. Each of these 8 registers are identical in function. Only one register (i.e. EBI_SFACCR0) is explained here.

SRAM/FLASH Access Configuration Register (EBI_SFACCR0)

Figure 2-5 **SRAM/FLASH Access Configuration Register (EBI_SFACCR0)**

Bit	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit 24
Field	WIDLC[3]	WIDLC[2]	WIDLC[1]	WIDLC[0]	WWEC[3]	WWEC[2]	WWEC[1]	WWEC[0]
R/W attribute	R/W				R/W			
Protection attribute	WS							
Initial value	0	0	0	0	0	1	0	1

Bit	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit 16
Field	WADC[3]	WADC[2]	WADC[1]	WADC[0]	WACC[3]	WACC[2]	WACC[1]	WACC[0]
R/W attribute	R/W				R/W			
Protection attribute	WS							
Initial value	0	1	0	1	1	1	1	1

Bit	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit 8
Field	RIDLC[3]	RIDLC[2]	RIDLC[1]	RIDLC[0]	FRADC[3]	FRADC[2]	FRADC[1]	FRADC[0]
R/W attribute	R/W				R/W			
Protection attribute	WS							
Initial value	1	1	1	1	0	0	0	0

Bit	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit 0
Field	RADC[3]	RADC[2]	RADC[1]	RADC[0]	RACC[3]	RACC[2]	RACC[1]	RACC[0]
R/W attribute	R/W				R/W			
Protection attribute	WS							
Initial value	0	0	0	0	1	1	1	1

[bit31:28] WIDLC[3:0] Bits

These bits set the number of idle cycles after write accessing. Configure 0x2 or greater value, when EBI_SFMR0:RDY bit is set to '1'.

Bit31:28	Description
0000	1 cycle
0001	2 cycles
...	...
1111	16 cycles

Note:

- *When a low-speed device with ready function is used, WIDLC is one cycle less for subsequent accesses of a splitted access.*

[bit27:24] WWEC[3:0] Bits

These bits set the number of cycles for write enable assertion. The setting of these bits also affects the MDQM (byte mask signal).

Bit27:24	Description
0000	1 cycle
0001	2 cycles
...	...
1110	15 cycles
1111	Reserved

[bit23:20] WADC[3:0] Bits

These bits set the number of setup cycles of the write address. The address is output for the cycles set in these bits, but the write enable is not asserted during such cycles.

Bit23:20	Description
0000	1 cycle
0001	2 cycles
...	...
1110	15 cycles
1111	Reserved

[bit19:16] WACC[3:0] Bits

These bits set the number of cycles necessary for write access. The address is not changed during the cycles specified in these bits.

Bit19:16	Description
0000	Reserved
0001	Reserved
0010	3 cycles
...	...
1111	16 cycles

Note:

- *The value must be greater than the sum of the address setup cycle and the write enable cycle.*

[bit15:12] RIDLC [3:0] Bits

These bits set the number of idle cycles after the read access. The bits are used to avoid data collision caused by a write access immediately after the read access. Configure 0xF when a low-speed device with ready function is used.

Bit15:12	Description
0000	1 cycle
0001	2 cycles
...	...
1111	16 cycles

Note:

- *When a low-speed device with ready function is used RIDLC is one cycle less for subsequent accesses of a splitted access.*

[bit11:8] FRADC [3:0] Bits

These bits are used exclusively for the setting of NOR FLASH that supports the page mode access. The bits set the initial latency of the address in the read access of the FLASH. The address is held during the specified cycles only in the first accessing. After that, accesses are performed according to the number of cycles set in EBI_SFACCR0:RACC. In the page mode access, MCSX and MOEX are asserted at the same time. When any value other than '0' is set in these bits, configure 0x0 in the EBI_SFACCR0:RADC (read address setup cycle).

Bit11:8	Description
0000	0 cycle
0001	1 cycles
...	...
1111	15 cycles

[bit7:4] RADC [3:0] Bits

These bits set the number of setup cycles of the read address. In the read address setup cycles, MCSX and the address is asserted, but MOEX is not asserted. When 0x0 is selected, MOEX and MCSX are asserted at the same time. The value specified here must be within the number of read access cycles (EBI_SFACCR0:RADC less than EBI_SFACCR0:RACC). Configure 0xA (10) to use a low-speed device with ready function. Set this bit to '0' when the NOR FLASH page access mode is used.

Bit7:4	Description
0000	0 cycle
0001	1 cycles
...	...
1111	15 cycles

[bit3:0] RACC [3:0] Bits

These bits set necessary cycles for read access. The address is not changed during the cycles specified in these bits, and data is fetched in the final cycle. Configure 0xF to use a low-speed device with ready function.

Bit3:0	Description
0000	1 cycle
0001	2 cycles
...	...
1111	16 cycles

2.5. SRAM/FLASH Address Control Register (EBI_SFADDCR0~7)

This register provides the address and address mask to select specific memory bank. Each of these 8 registers are identical in function. Only one register (i.e. EBI_SFADDCR0) is explained here.

SRAM/FLASH Address Control Register (EBI_SFADDCR0)

Figure 2-6 **SRAM/FLASH Address Control Register (EBI_SFADDCR0)**

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

Bit	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit 16
Field	MASK[7]	MASK[6]	MASK[5]	MASK[4]	MASK[3]	MASK[2]	MASK[1]	MASK[0]
R/W attribute	R/W							
Protection attribute	WS							
Initial value	0	0	0	0	1	1	1	1

Bit	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit 8
Field	Reserved							
R/W attribute	R0,WX							
Protection attribute	WS							
Initial value	0	0	0	0	0	0	0	0

Bit	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit 0
Field	ADDR[7]	ADDR[6]	ADDR[5]	ADDR[4]	ADDR[3]	ADDR[2]	ADDR[1]	ADDR[0]
R/W attribute	R/W							
Protection attribute	WS							
Initial value	0	0	0	0	0	0	0	0

Initial value of,
EBI_SFADDCR1 : 0x000F0010
EBI_SFADDCR2 : 0x000F0020
EBI_SFADDCR3 : 0x000F0030
EBI_SFADDCR4 : 0x000F0040
EBI_SFADDCR5 : 0x000F0050
EBI_SFADDCR6 : 0x000F0060
EBI_SFADDCR7 : 0x000F0070

[bit31:24] Reserved Bits

[bit23:16] MASK [7:0] Bits

These bits set the mask value of the value set in EBI_SFADDCR0:ADDR. The External Bus Interface masks the EBI_SFADDCR0:ADDR and internal bus address (masked when '1' is set) according to the specified mask, and compares them. When they match, it accesses MCSX signal.

[bit15:8] Reserved Bits

[bit7:0] ADDR [7:0] Bits

These bits specify addresses to set the corresponding MCSX region. The addresses are those in the fixed area of the 256 MB assigned to the SRAM/FLASH IF. Value equivalent to part of the addresses [27:20] must be defined.

MASK bit Example:

When they match, it accesses MCSX signal. [22:16] mask addresses [26:20] respectively. Setting for 64 MB and 128 MB is not permitted, because the bit width of MAD(external pin) is 25 bits.

Example:

EBI_SFADDCR:ADDR = 00001000 (b)

EBI_SFADDCR:MASK = 00000011 (b)

When to be selected

Internal bus address (address of the external I/F) MAD = 0x10900000

Masking

EBI_SFADDCR:ADDR & (!MASK) = 00001000 (b)

EBI_SFADDCR:AD [27:20] & (!MASK) = 00001000 (b) .. Match. The device is selected.

When not to be selected

Internal bus address (address of the external I/F) MAD = 0x10C00000

Masking

EBI_SFADDCR:ADDR & (!MASK) = 00001000 (b)

EBI_SFADDCR:AD [27:20] & (!MASK) = 00001100 (b) .. Not match. The device is not selected.

The masking selects the size of the region. In the example, 0x10800000~0x10BFFFFF (4 MB) are selected. The bits for which '1' is specified in masking are lost in the mask processing. The bits are invalidated even they have been set in EBI_SFADDCR:ADDR. If the LSB in the example is '1' (EBI_SFADDCR:ADDR=0x09), the same address field is selected since it is invalidated in the masking. The following shows relations between mask settings and the sizes of address fields.

0000000 (b) -> 1 MB	0001111 (b) -> 16 MB
0000001 (b) -> 2 MB	0011111 (b) -> 32 MB
0000011 (b) -> 4 MB	0111111 (b) -> 64 MB (Reserved)
0000111 (b) -> 8 MB	1111111 (b) -> 128 MB (Reserved)

Notes:

- *Each address field must not overlap.*
- *Refer to the device data sheet for the actual number of external pins of MAD.*

2.6. Error Register (EBI_ERRR)

This register provides the Error status for SRAM/FLASH access.

Error Register (EBI_ERRR)

Figure 2-7 Error Register (EBI_ERRR)

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

Bit	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit 16
Field	Reserved							
R/W attribute	R0,WX							
Protection attribute	WS							
Initial value	0	0	0	0	0	0	0	0

Bit	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
Field	Reserved							
R/W attribute	R0,WX							
Protection attribute	WS							
Initial value	0	0	0	0	0	0	0	0

Bit	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit 0
Field	Reserved						Reserved	SFER
R/W attribute	R0,WX						R,W1	R,W
Protection attribute	WS							
Initial value	0	0	0	0	0	0	0	0

[bit31:8] Reserved Bits

[bit7:2] Reserved Bits

[bit1] Reserved Bits

[bit0] SFER Bit

Indicates an access has been made to the region that is not mapped during an access to the SRAM/FLASH. In this case the External Bus Interface returns an error response to the internal bus. Writing '1' to this bit clears the SRAM/FLASH error flag. Writing '0' to this register does not have effect.

Bit	Description
0	No error
1	Error

3. Operation of External Bus Interface

The External Bus Interface can control SRAM and FLASH. This chapter describes the operation of the External Bus Interface in detail.

3.1. SRAM/FLASH Controller

The SRAM/FLASH controller has an address space of 256 MB. Each address can be set optionally (With the bit width of the external output addresses taken into account, the maximum size of an actual address is 32 MB). A different timing and endianess can be specified for each chip select. NAND FLASH and NOR FLASH can be connected. NOR FLASH is accessed by usual SRAM accessing. Dedicated pins are provided for NAND FLASH. With this, access to the device during a NAND access, sharing a data pin is enabled (see Figure 4-5). When the access is made using a bit width greater than the targeted one, the access is converted into a continuous access where only addresses are changed with the MCSX being kept to ‘0’. For example, when a 32-bit read access is performed to an 8-bit width device from the internal bus and the device is not using READY signal to control the data access, the address transits 0->1->2->3 with the MCSX being kept ‘0’, and data are sequentially fetched out from D[7:0] according to transit timings (see Section 5). Refer Section ‘Low-speed device interface function’ for the device which use READY signal to control the data access. Data read from device to the internal bus depend on the configured endianess. When the access is made using a bit width narrower than the targeted one (e.g. byte access to a 16-bit width target), the byte access will be executed by being controlled by MDQM (byte mask) signal if it is in write operation (The SRAM/FLASH controller outputs only necessary data). Concerning a device without input mask, the MDQM signal can be used as write enable. In read, if the target device has output mask signal, unnecessary outputs from the device can be avoided by letting the device output only the data of necessary bytes using MDQM control. As a result, power dissipation is reduced.

3.1.1. Low-Speed Device Interface Function

External Bus Interface (EBI) can be connected to the peripheral that has general purpose low- speed interface by connecting MRDY pin of EBI to peripheral ready (READY) signal. The MRDY pins should be in ‘wait’ at ‘0’ and in ‘ready’ at ‘1’. Setting the MRDY to ‘0’ must be synchronous with the falling edge of MCSX signal. MRDY is synchronized inside EBI with operating clock, and it becomes valid internally after max two cycles. In an access exceeding the external data bus width (e.g. word (32 bits) access to an 8-bit device), accesses will be generated in succession ‘read’ -> ‘read’, ‘write’ -> ‘write’ until it covers all the excessive bits. In this case ,the MCSX will be high between the 2 continuous read or write. The high time for write is defined by the EBI_SFACCRn:WIDLC and for the read it is defined by EBI_SFACCRn:RIDLC. For devices not using the READY function, such as SRAM memory, etc, the EBI_SFMRn:RDY bit of the corresponding chip select must be OFF. If the MRDY signal is at ‘1’ from the beginning of access, the access will be executed in the same manner as usual SRAM access. Operations cannot be guaranteed in case the MRDY falls (or becomes high pulse) in the course of access cycle.

Note:

- *The function is not applicable to RDY/BUSY signal of FLASH memory.*

3.2. NAND FLASH access

Access to NAND FLASH requires different process from usual SRAM access. The NAND FLASH has (512+16) bytes of internal registers (8-bit/16-bit NANDs have (1024+32) bytes). Basic data accesses are performed in these units. In read, several µs to tens of µs are needed for loading data to internal registers, in data write to memory cells, hundreds of µs are required, in block deletion, several ms. The chip select signal must be kept '0' in data load to the internal registers. Therefore, the read/write signals to the NAND cannot be controlled. The External Bus Interface, which has a discrete enable signal for NAND, executes accesses with the chip select signal connected to the NAND FLASH being held at '0'. (It can be reset to '1'). With this, accessing to other chips that share the data line becomes possible. For user side, accesses made to the NAND FLASH are I/O accesses:

1. Write access to +0x2000 is converted to address issue (MNALE is asserted) to the NAND FLASH.
2. Write access to +0x1000 is converted to command issue (MNCLE is asserted) to the NAND FLASH.
3. Write/read access to +0x0000 is converted to data access (MNALE and MNCLE are not asserted) to the NAND FLASH: based on the base addresses in the region set to NAND mode.

Settings of access timings are the same as those used in SRAM access. MNCLE is output in the same timing as the address output of the access. MNALE is kept being asserted after the address issue until the write accessing to +0x3000 or any other write accessing (data or command) than address issue. This is because the NAND FLASH is incapable of de asserting the MNALE in multiple write accesses made for the address issue. In the access to +0x3000, only assertion of MNALE is generated and no access is made. The following charts show steps of the NAND FLASH accesses.

Figure 3-1 NAND Read access

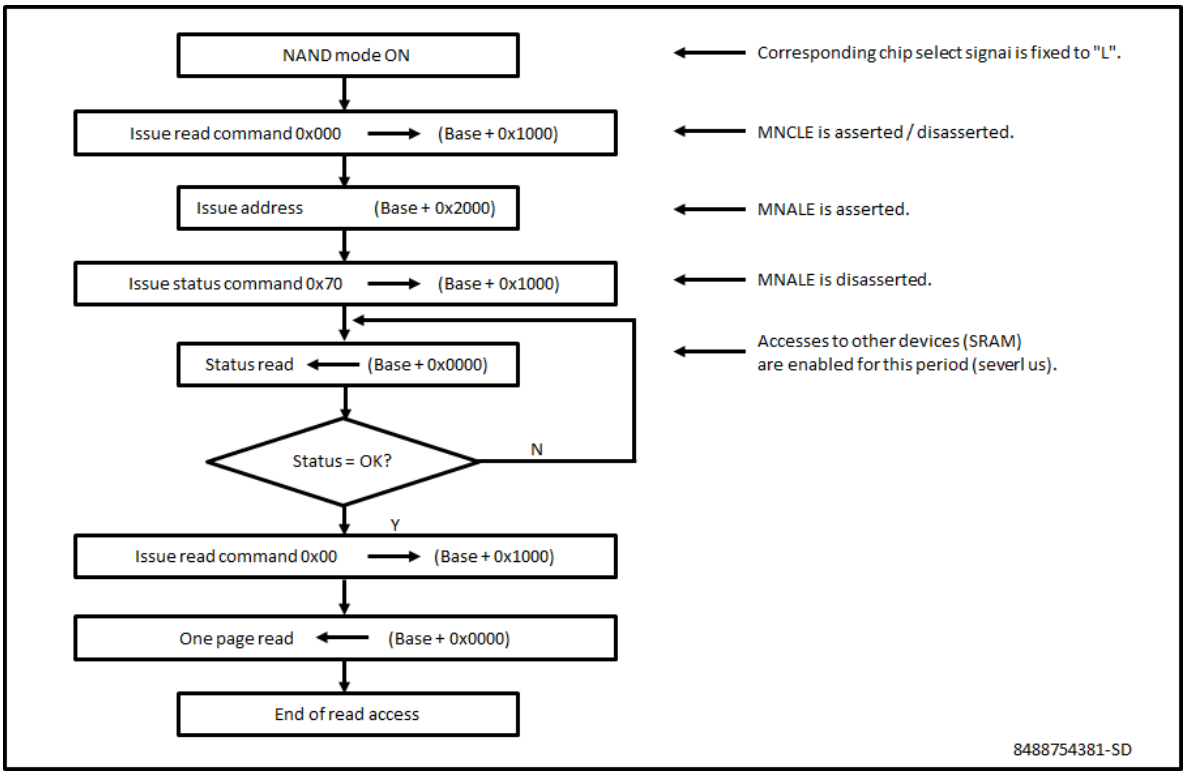


Figure 3-2 NAND FLASH Write access (Auto Program)

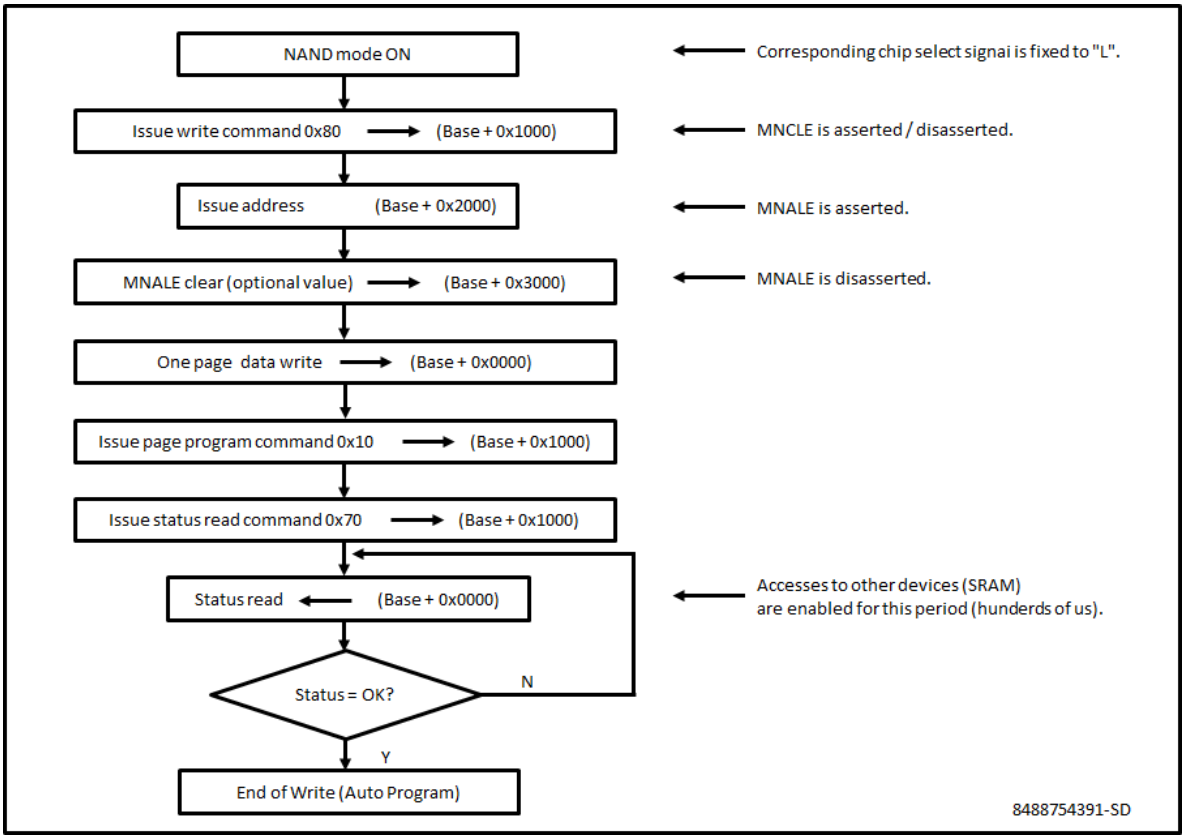
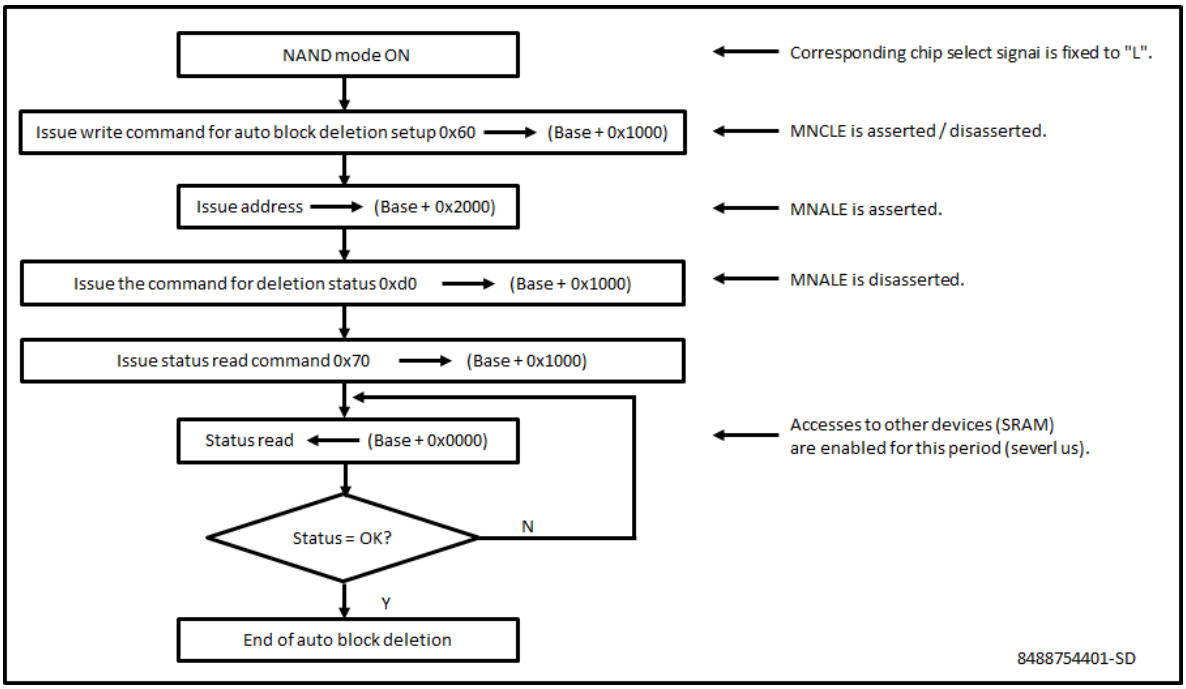


Figure 3-3 NAND FLASH Auto Block Deletion access



As shown in the charts, accesses to other memory devices are enabled even in phases where the NAND FLASH accesses have not been completed. Since also DMA can be used for data read and data write, the processor is capable of making accesses to the NAND with minimum operation.

3.3. Endianess and Enabled byte Lanes

The External Bus Interface each SRAM/FLASH bank supports bi-endian. The table shows correspondence between the endianess and byte lanes.

Table 3-1 Endianess access Table

Endian	Access Size	Target width	Internal bus address	Enabled byte Lane	Corresponding AHB Data	MDQM [3:0]	MAD [1:0]
Little	Word	8-bit	00	M*Datb[7:0]	H*DATA[7:0]	1110	00
					H*DATA[15:8]		01
					H*DATA[23:16]		10
					H*DATA[31:24]		11
		16-bit	00	M*Datb[15:0]	H*DATA[15:0]	1100	00
				M*Datb[15:0]	H*DATA[31:16]		10
		32-bit	00	M*Datb[31:0]	H*DATA[31:0]	0000	00
	Half word	8-bit	00	M*Datb[7:0]	H*DATA[7:0]	1110	00
					H*DATA[15:8]		01
			10	M*Datb[7:0]	H*DATA[23:16]	1110	10
					H*DATA[31:24]		11
		16-bit	00	M*Datb[15:0]	H*DATA[15:0]	1100	00
			10	M*Datb[15:0]	H*DATA[31:16]	1100	10
		32-bit	00	M*Datb[15:0]	H*DATA[15:0]	1100	00
			10	M*Datb[31:16]	H*DATA[31:16]	0011	00
	Byte	8-bit	00	M*Datb[7:0]	H*DATA[7:0]	1110	00
			01	M*Datb[7:0]	H*DATA[15:8]	1110	01
			10	M*Datb[7:0]	H*DATA[23:16]	1110	10
			11	M*Datb[7:0]	H*DATA[31:24]	1110	11
		16-bit	00	M*Datb[7:0]	H*DATA[7:0]	1110	00
			01	M*Datb[15:8]	H*DATA[15:8]	1101	00
			10	M*Datb[7:0]	H*DATA[23:16]	1110	10
			11	M*Datb[15:8]	H*DATA[31:24]	1101	10
		32-bit	00	M*Datb[7:0]	H*DATA[7:0]	1110	00
			01	M*Datb[15:8]	H*DATA[15:8]	1101	00
			10	M*Datb[23:16]	H*DATA[23:16]	1011	00
			11	M*Datb[31:24]	H*DATA[31:24]	0111	00

Endian	Access Size	Target width	Internal bus address	Enabled byte Lane	Corresponding AHB Data	MDQM [3:0]	MAD [1:0]
Big	Word	8-bit	00	M*Datb[7:0]	H*DATA[31:24]	1110	00
					H*DATA[23:16]		01
					H*DATA[15:8]		10
					H*DATA[7:0]		11
		16-bit	00	M*Datb[15:0]	H*DATA[31:16]	1100	00
				M*Datb[15:0]	H*DATA[15:0]		10
		32-bit	00	M*Datb[31:0]	H*DATA[31:0]	0000	00
	Half word	8-bit	00	M*Datb[7:0]	H*DATA[15:8]	1110	00
					H*DATA[7:0]		01
			10	M*Datb[7:0]	H*DATA[31:24]	1110	10
					H*DATA[23:16]		11
		16-bit	00	M*Datb[15:0]	H*DATA[15:0]	1100	00
			10	M*Datb[15:0]	H*DATA[31:16]	1100	10
		32-bit	00	M*Datb[31:16]	H*DATA[15:0]	0011	00
			10	M*Datb[15:0]	H*DATA[31:16]	1100	00
	Byte	8-bit	00	M*Datb[7:0]	H*DATA[7:0]	1110	00
			01	M*Datb[7:0]	H*DATA[15:8]	1110	01
			10	M*Datb[7:0]	H*DATA[23:16]	1110	10
			11	M*Datb[7:0]	H*DATA[31:24]	1110	11
		16-bit	00	M*Datb[15:8]	H*DATA[7:0]	1101	00
			01	M*Datb[7:0]	H*DATA[15:8]	1110	00
			10	M*Datb[15:8]	H*DATA[23:16]	1101	10
			11	M*Datb[7:0]	H*DATA[31:24]	1110	10
		32-bit	00	M*Datb[31:24]	H*DATA[7:0]	0111	00
			01	M*Datb[23:16]	H*DATA[15:8]	1011	00
			10	M*Datb[15:8]	H*DATA[23:16]	1101	00
			11	M*Datb[7:0]	H*DATA[31:24]	1110	00

M*Datb: Read or write data on the external bus.

H*DATA: Read or write data on the internal bus.

Example:

When a byte access is executed to a 32-bit width target, the enabled byte lane varies according to the order of the internal bus addresses shown in the table. Since the data width of the internal bus (32 bits) and the width of the target agree, the data in the internal bus is output to the corresponding byte lane. When a byte access is made to a 16-bit width target, only 2 bytes become enabled according to the endian as shown in the table. Data in the internal bus are distributed to 16 bits according to the endian, and provides the required input and output.

Notes:

- *Using external SRAM on the EBI Interface.*

It is recommended to set the EBI memory region to "Device" or "Strongly Ordered" (by setting the memory range) using the MPU of the Cortex R5 CPU to ensure that memory accesses occur in the same order and have the sizes as the program code imply.

However, if the memory region needs to be set to "Normal" with "Cachable WBWA" (WBWA = Write Back, Write Allocate), it is important to remember that thin setting only works in "little endian" mode on the EBI.

4. External memory Connection to External Bus Interface

This section is examples showing various external memory connection to External Bus Interface.

Notes:

- *Memory Interface names corresponds to their functional pin names which is similar to the given one.*
- *Inputs/outputs in the above figure which end with an "X", for example "MOEX", indicate active low inputs/outputs.*

Figure 4-1 8-Bit SRAM with External Bus Interface

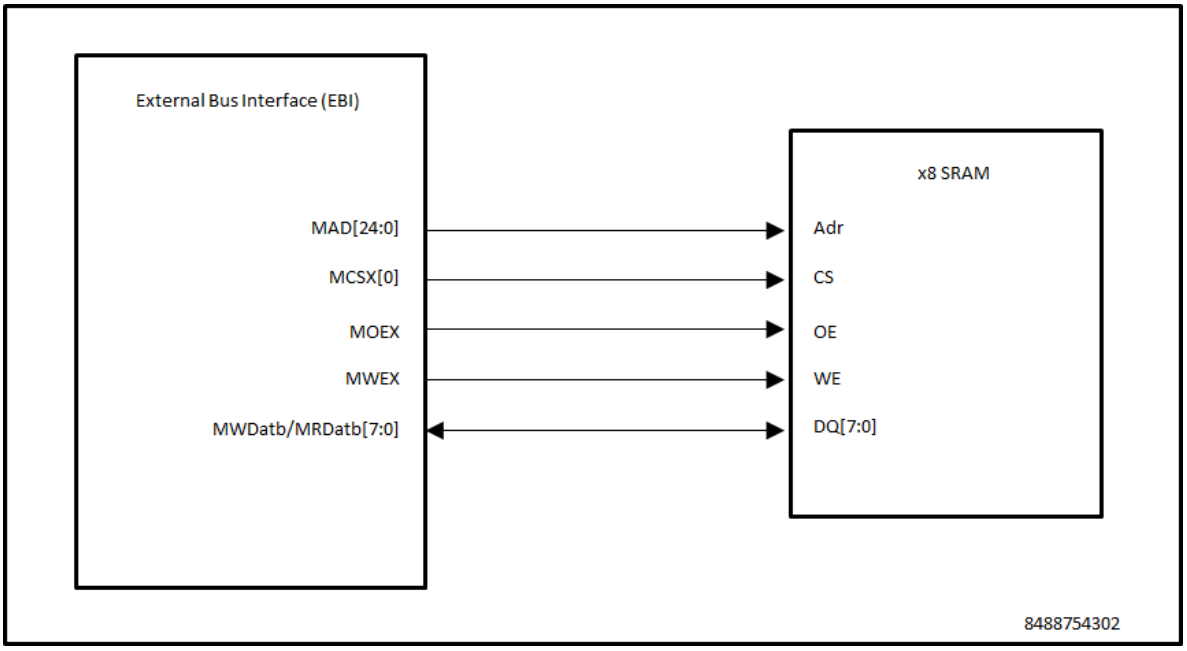


Figure 4-2 Two 8-Bit SRAM with External Bus Interface

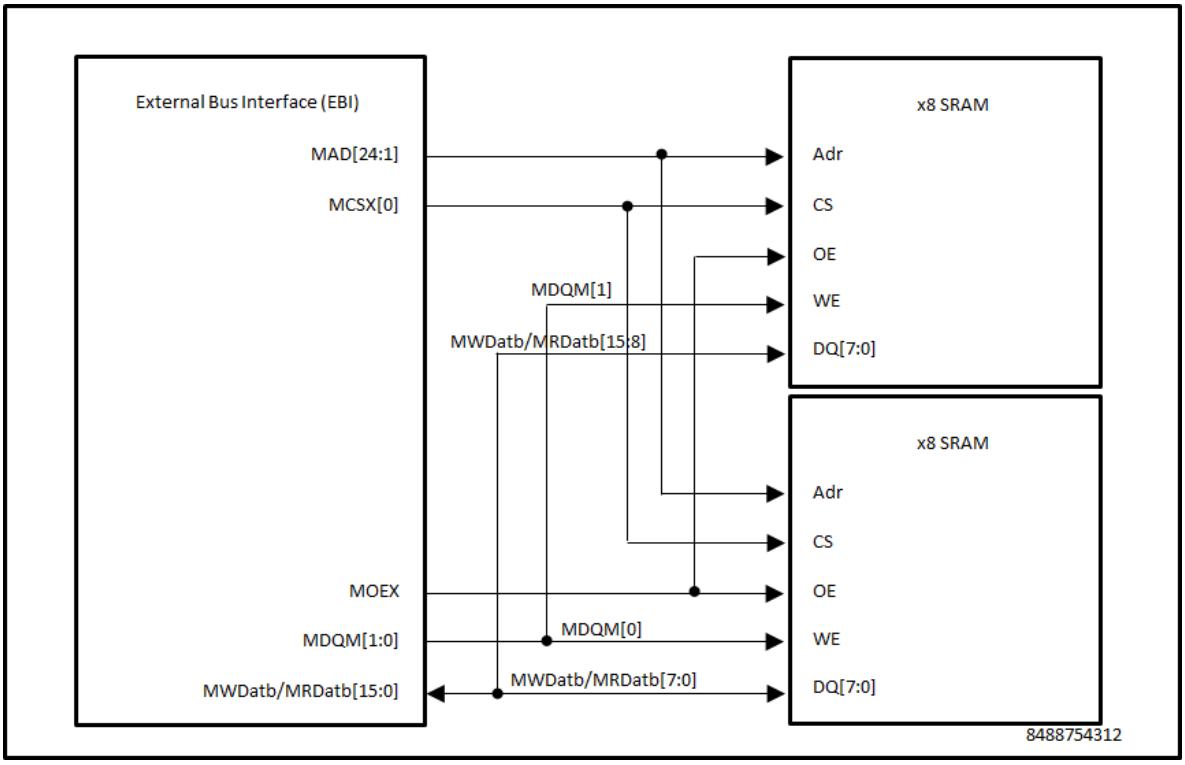


Figure 4-3 16-Bit SRAM with External Bus Interface

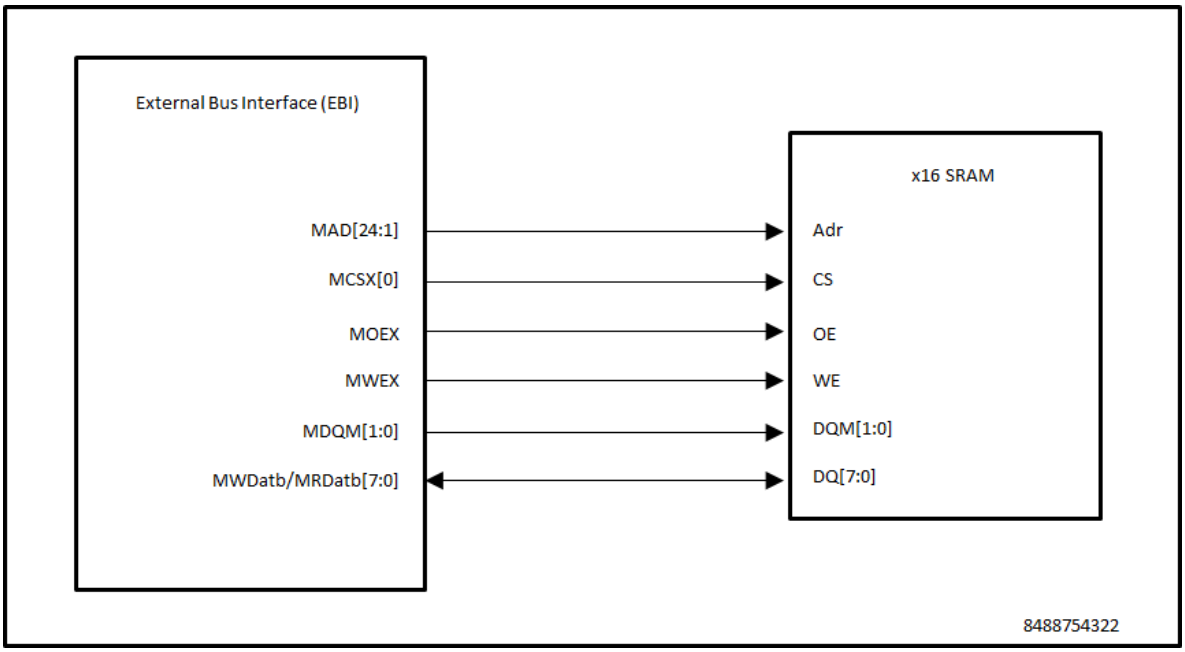


Figure 4-4 Two 16-Bit SRAM with External Bus Interface

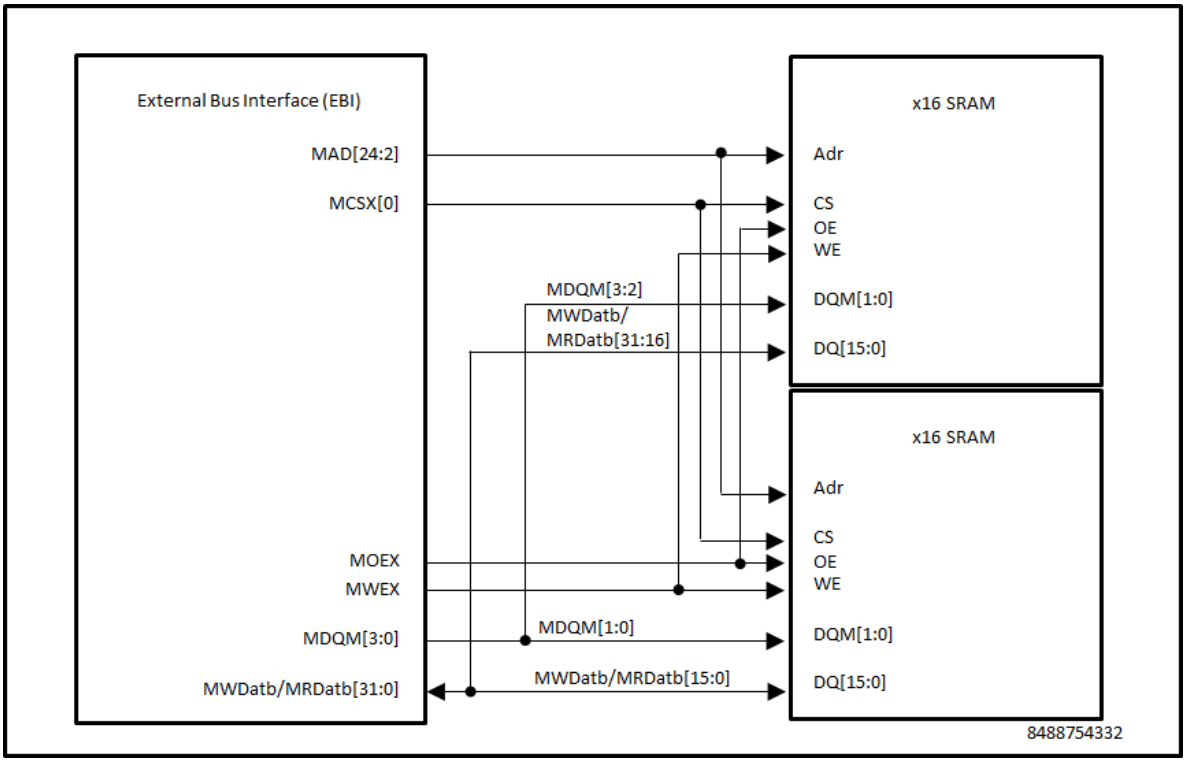
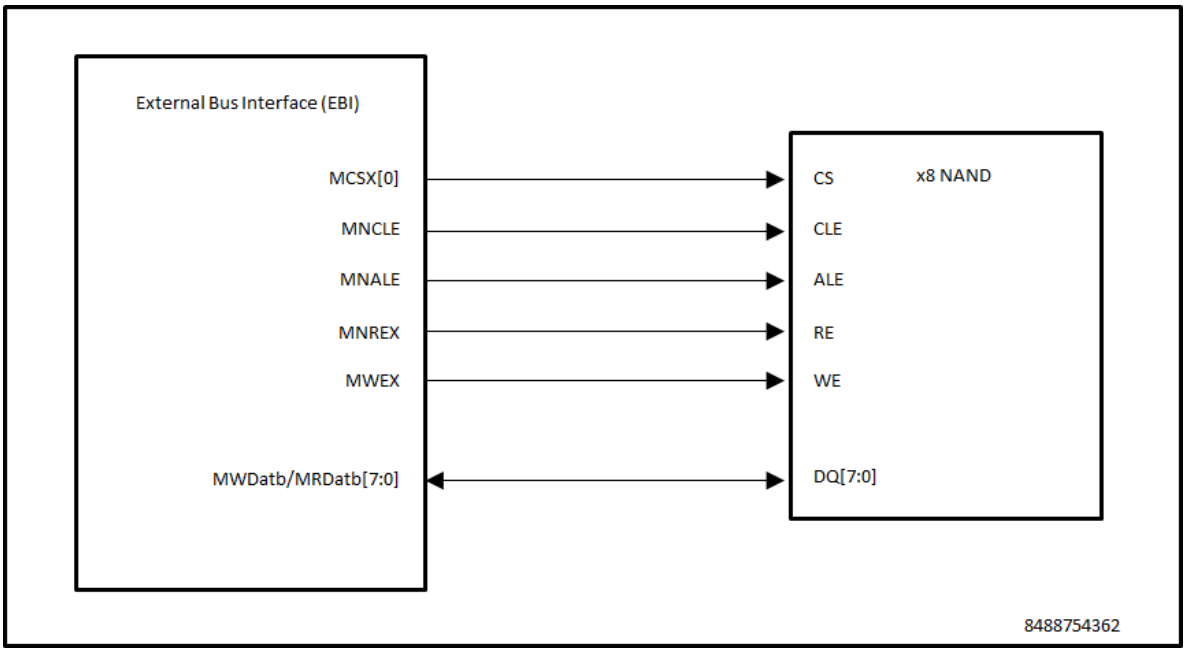


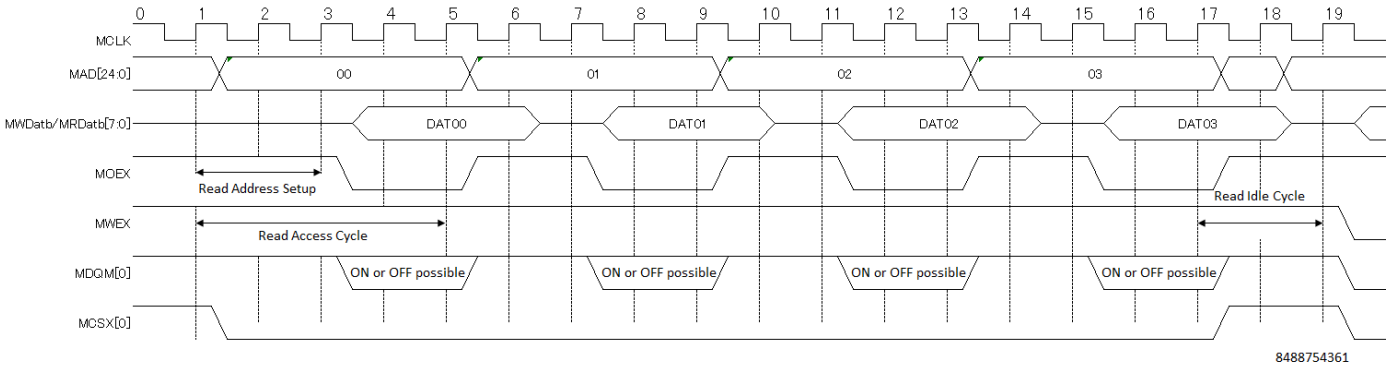
Figure 4-5 8-Bit NAND with External Bus Interface IF



5. Access Waveform Examples for External Bus Interface

This section is examples showing access to various external memory.

Figure 5-1 Word Read access to 8-Bit width SRAM

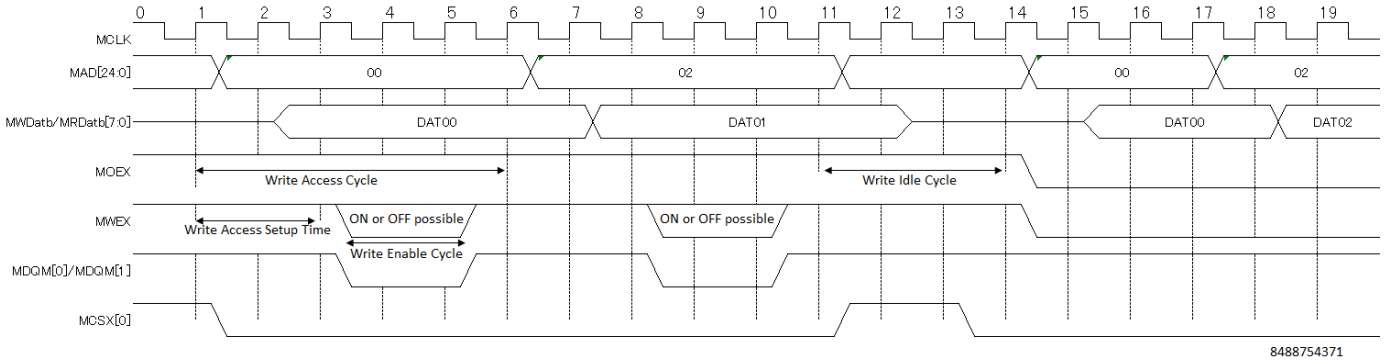


RACC=3 [4-cycle]

RADC=2 [2-cycle]

RIDLC=1 [2-cycle]

Figure 5-2 Word Write/Read Continuous access to 16-Bit width SRAM



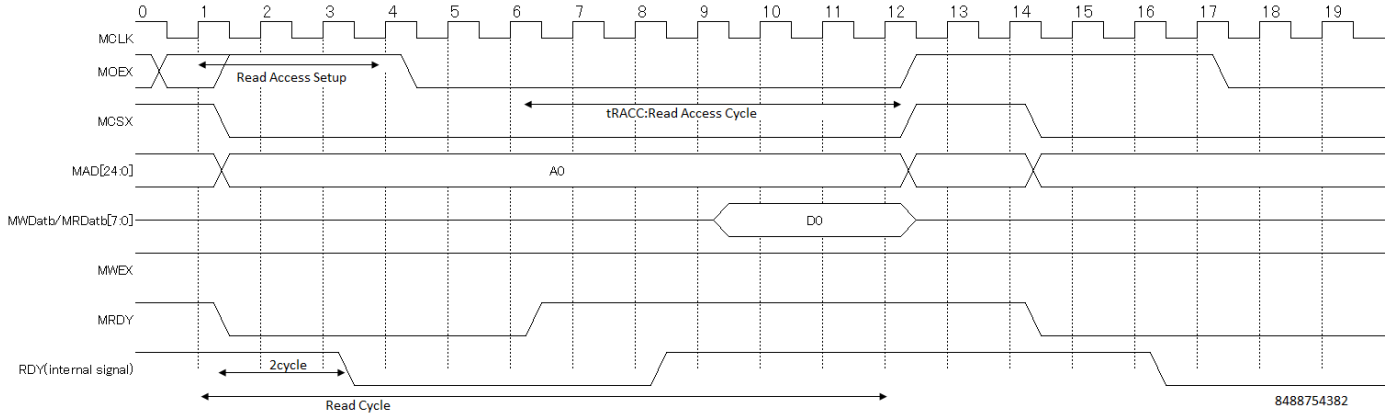
WACC=4 [5-cycle]

WADC=1 [2-cycle]

WWEC=1 [2-cycle]

WIDLC=2 [3-cycle]

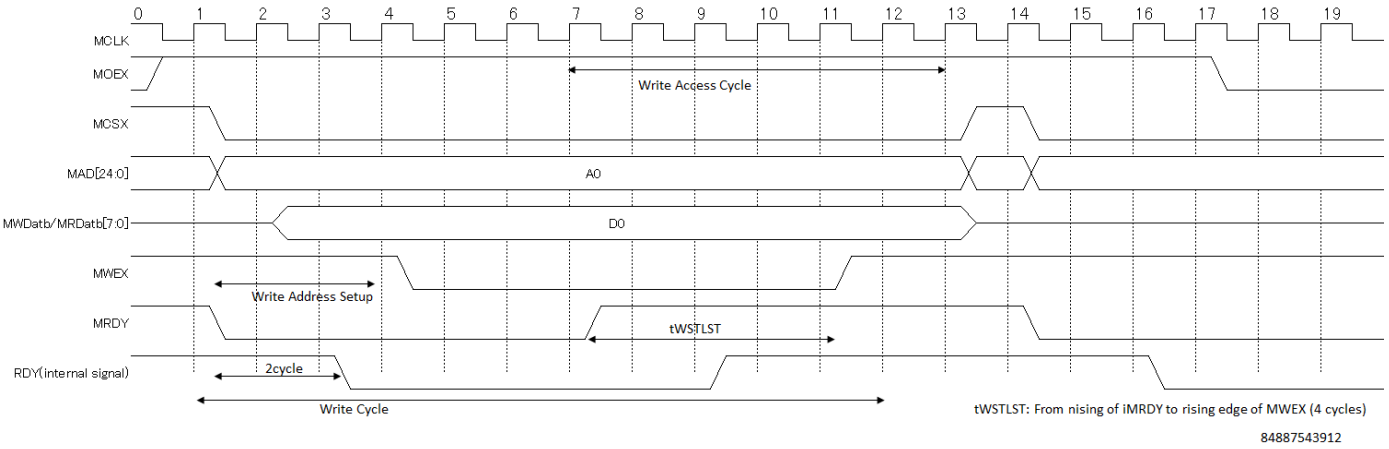
Figure 5-3 Read to Low-Speed Device



RACC=5 [6-cycle]

RADC=3 [3-cycle]

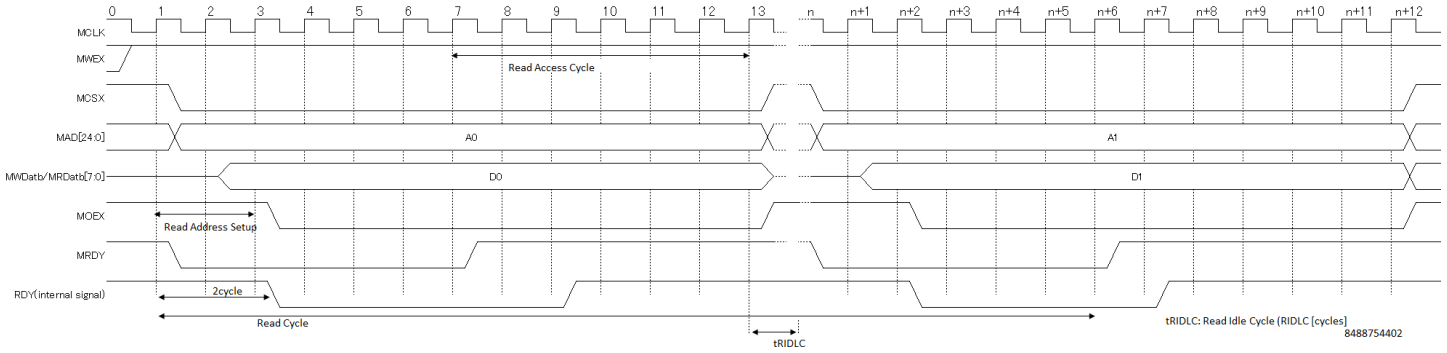
Figure 5-4 Write to Low-Speed Device



WACC=5 [6-cycle]

WADC=2 [3-cycle]

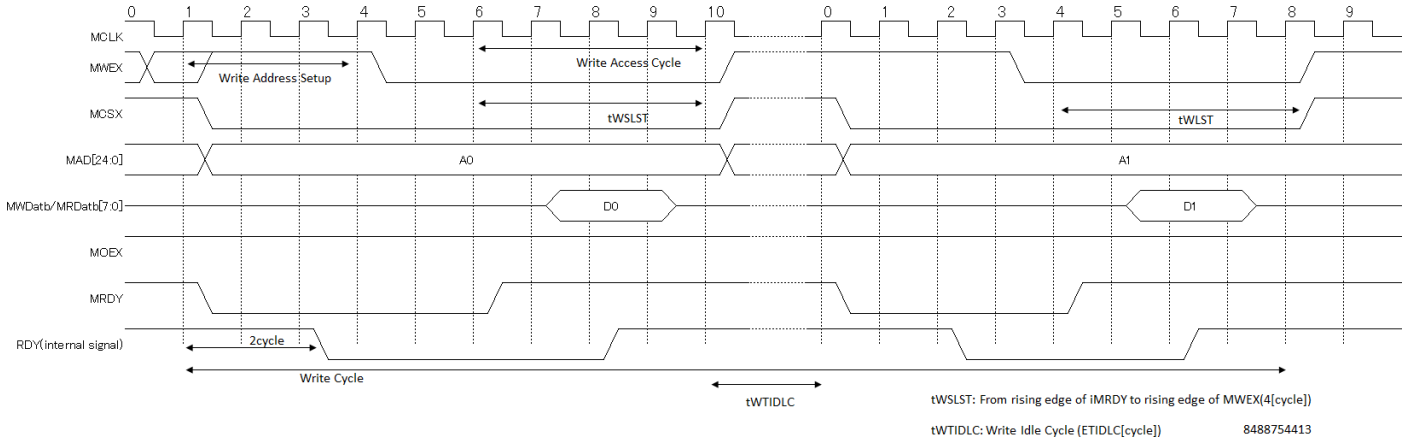
Figure 5-5 Read to Low-Speed Device with Split Transfer on Memory Interface



RACC=5 [6-cycle]

RADC=2 [2-cycle]

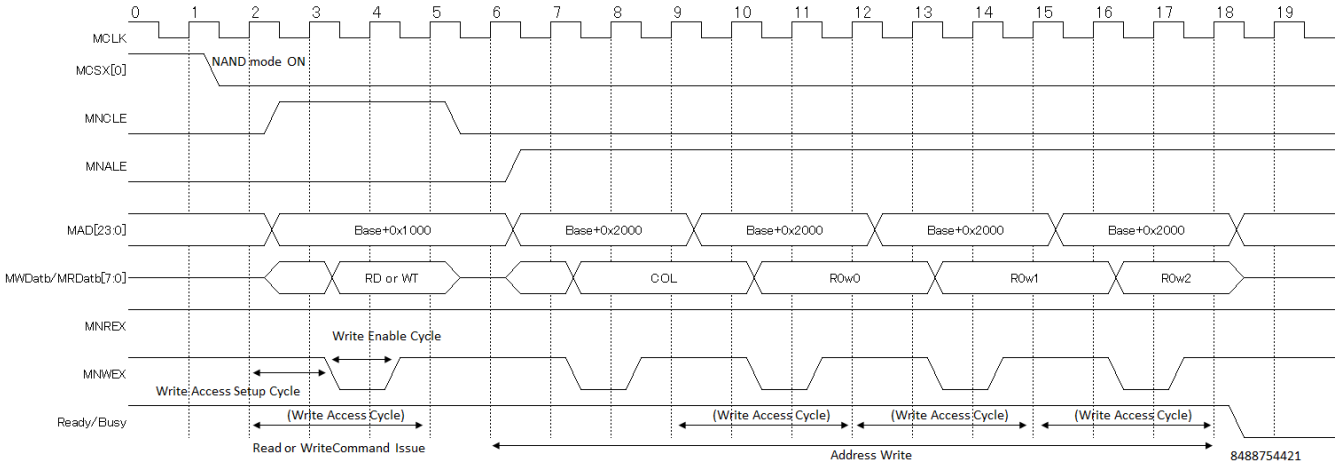
Figure 5-6 Write to Low-Speed Device with Split Transfer on Memory Interface



WACC=3 [4-cycle]

WADC=2 [3-cycle]

Figure 5-7 8-Bit NAND Read or Write command Issue

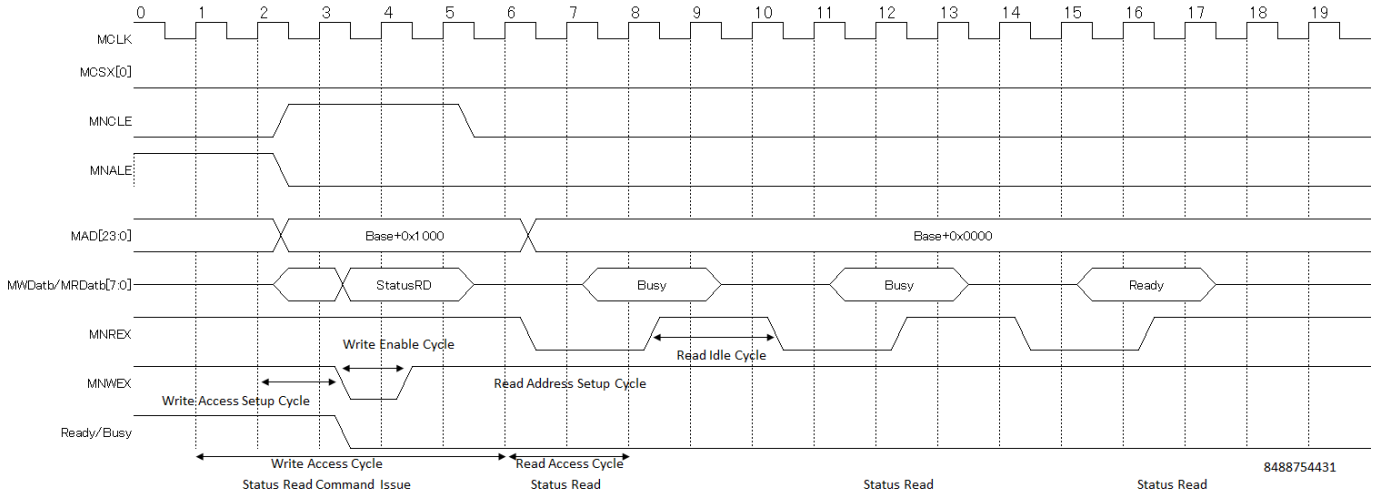


WACC=2 [2-cycle]

WADC=0 [1-cycle]

WWEC=0 [1-cycle]

Figure 5-8 8-Bit NAND Status Read



WACC=2 [2-cycle]

WADC=0 [1-cycle]

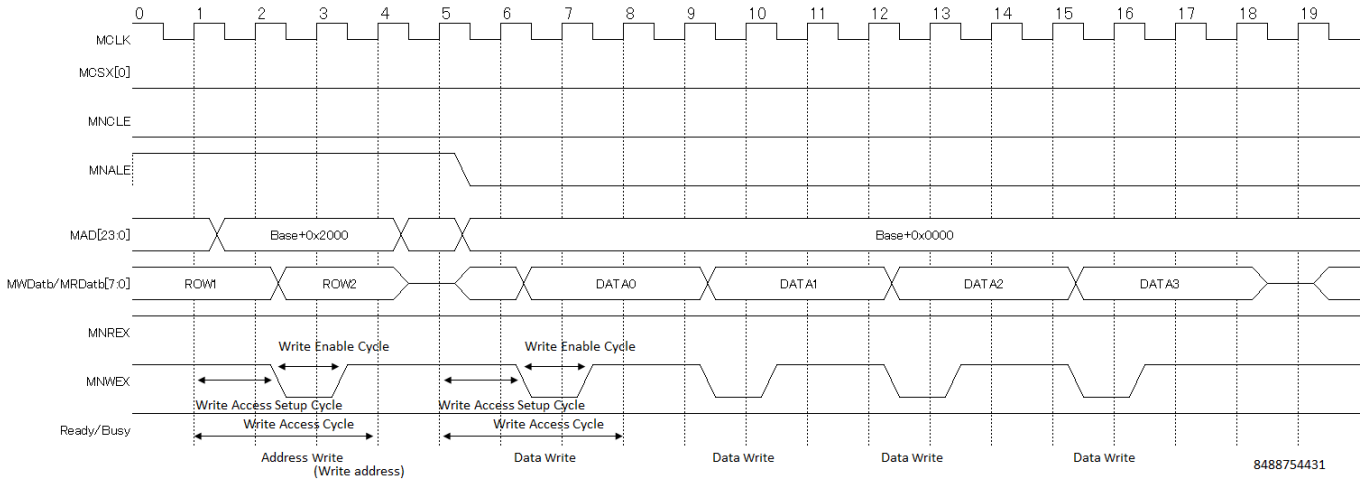
WWEC=0 [1-cycle]

RACC=1 [2-cycle]

RADC=0 [0-cycle]

RIDLC=2 [2-cycle]

Figure 5-9 8-Bit NAND Data Write

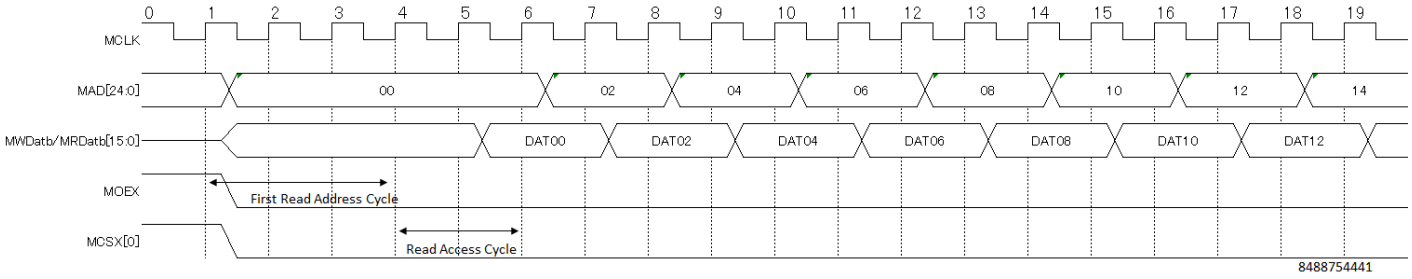


WACC=2 [2-cycle]

WADC=0 [1-cycle]

WWEC=0 [1-cycle]

Figure 5-10 16-Bit NOR Page Read



FRADC=3 [3-cycle]

RACC=1 [2-cycle]

Refer to the Device Datasheet for details on AC timings of EBI.

CHAPTER 54: System SRAM Module (SRAM_IF)

This chapter explains the functions and operations of System SRAM module (SRAM_IF).

1. "Outline of SRAM_IF"
2. "Operation of SRAM_IF"
3. "SRAM_IF Register Set"
4. "How to Use SRAM_IF"

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
ECU	Error Collection Unit
AXI	Advanced eXtensible Interface

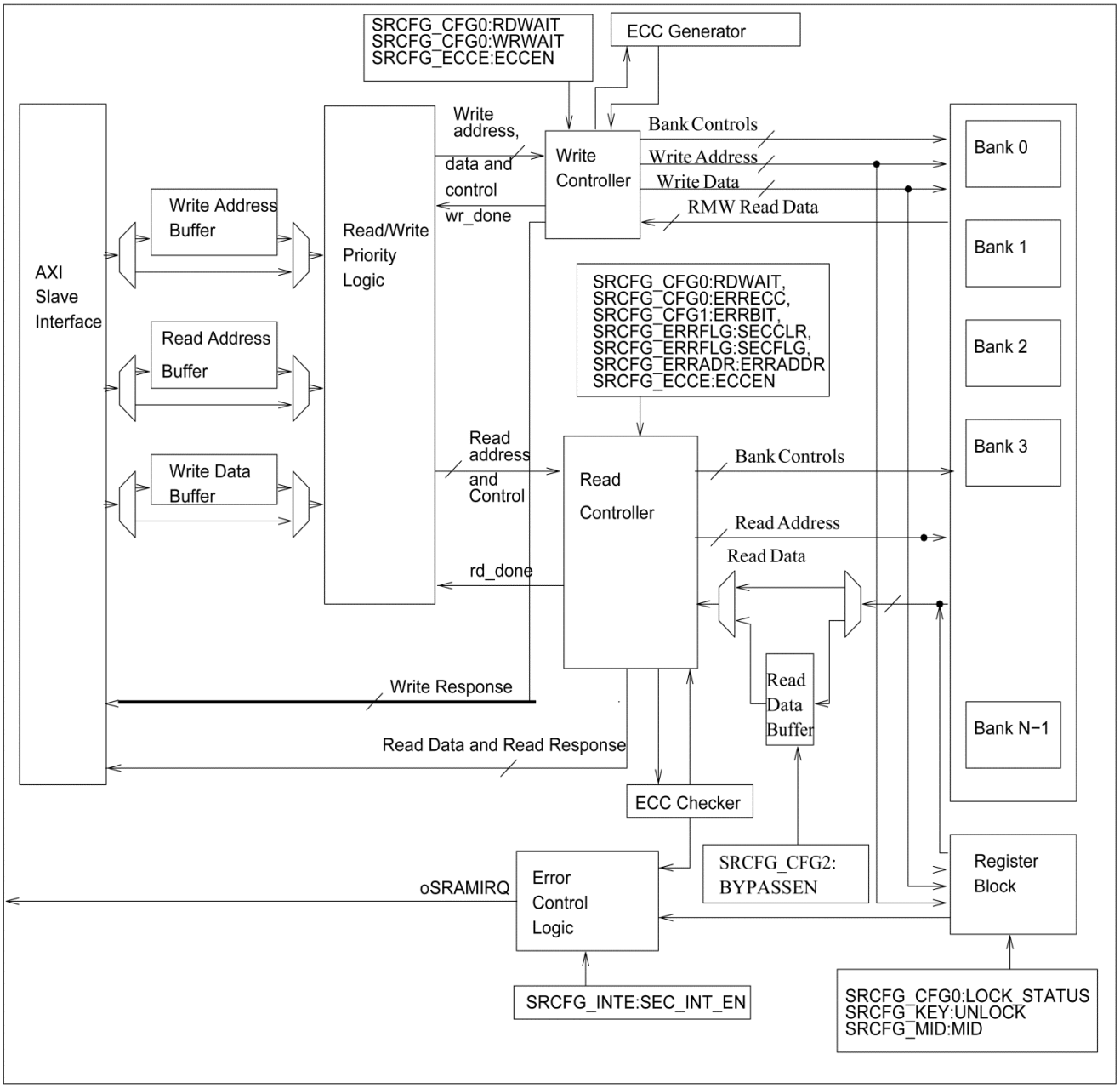
2. Operation of SRAM_IF

This chapter describes the operation of SRAM_IF.

Block Diagram

The Figure 2-1 shows the block diagram of SRAM_IF.

Figure 2-1 SRAM_IF Block Diagram



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 are 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 4-1 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:SECFLAG 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.

3. SRAM_IF Register Set

All registers in SRAM_IF are explained in this section.

SRAM_IF Registers

3.1. 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

Table 3-1 SRCFG_CFG0 Register Bit Description

Bit Position	Bit Name	Description
[31:26]	Reserved	Reserved
[25:24]	RDWAIT[1:0]	Read Data Wait State Value (RDWAIT[1:0]) These 2 bits are used to configure the number of data wait states for SRAM read transaction. 00: 0 wait State 01: 1 wait State 10: 2 wait States 11: 3 wait States (default)
[23:18]	Reserved	Reserved

Bit Position	Bit Name	Description
[17:16]	WRWAIT[1:0]	Write Data Wait State Value (WRWAIT[1:0]) These 2 bits are used to configure the number of data wait states for SRAM Write transaction. 00: 0 wait State 01: 1 wait State 10: 2 wait States 11: 3 wait States (default)
[15:9]	Reserved	Reserved
8	LOCK_STATUS	SRAM_IF Lock Status (LOCK_STATUS) This bit provides locked or unlocked status of SRAM_IF. 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)
7	Reserved	Reserved
[6:0]	ERRECC[6:0]	ECC ERRBIT Value (ERRECC[6:0]) 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).*

3.2. 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

Table 3-2 SRCFG_CFG1 Register Bit Description

Bit Position	Bit Name	Description
[31:0]	ERRBIT[31:0]	ERRBIT Value (ERRBIT[31:0]) 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).*

3.3. 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

Table 3-3 SRCFG_CFG2 Register Bit Description

Bit Position	Bit Name	Description
[31:8]	Reserved	Reserved
[7:1]	Reserved	Reserved
0	BYPASSEN	RDB Bypass Disable or Enable (BYPASSEN) RDB Bypass will be enabled in case this bit is 1. 0: RDB Bypass path is disabled (default) 1: RDB Bypass path is enabled

Note:

- *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).*

3.4. 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

Table 3-4 SRCFG_KEY Register Bit Description

Bit Position	Bit Name	Description
[31:0]	UNLOCK[31:0]	SRAM_IF Unlock or Lock Key (UNLOCK[31:0]) 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.

Note:

- *The "Unlock/Lock" Key values are specified as followings.*
  SRAM_IF UNLOCK_VALUE = 0x5ECC551F
  SRAM_IF LOCK_VALUE = 0x551FB10C

3.5. 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

Table 3-5 SRCFG_ERRFLG Register Bit Description

Bit Position	Bit Name	Description
[31:16]	Reserved	Reserved
[15:9]	Reserved	Reserved
8	SECCLR	Single-Bit Error Flag Clear (SECCLR) If this bit is 1, it clears the flag SRCFG_ERRFLG:SECFLG. This bit always returns 0 on read.
[7:1]	Reserved	Reserved
0	SECFLG	Single-Bit Error Detection Flag (SECFLG) 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.*

3.6. 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

Table 3-6 SRCFG_INTE Register Bit Description

Bit Position	Bit Name	Description
[31:8]	Reserved	Reserved
[7:1]	Reserved	Reserved
0	SEC_INT_EN	Single-Bit Error Interrupt Enable bit (SEC_INT_EN) 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)

3.7. 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

Table 3-7 SRCFG_ECCE Register Bit Description

Bit Position	Bit Name	Description
[31:8]	Reserved	Reserved
[7:1]	Reserved	Reserved
0	ECCEN	ECCEN Value (ECCEN) 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. 0: ECC logic is disabled 1: ECC logic is enabled (default)

3.8. 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	R,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Table 3-8 SRCFG_ERRADR Register Bit Description

Bit Position	Bit Name	Description
[31:0]	ERRADDR[31:0]	ECC Error Address (ERRADDR[31:0]) 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.

3.9. 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

Table 3-9 SRCFG_MID Register Bit Description

Bit Position	Bit Name	Description
[31:0]	MID[31:0]	Module ID (MID[31:0]) 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.

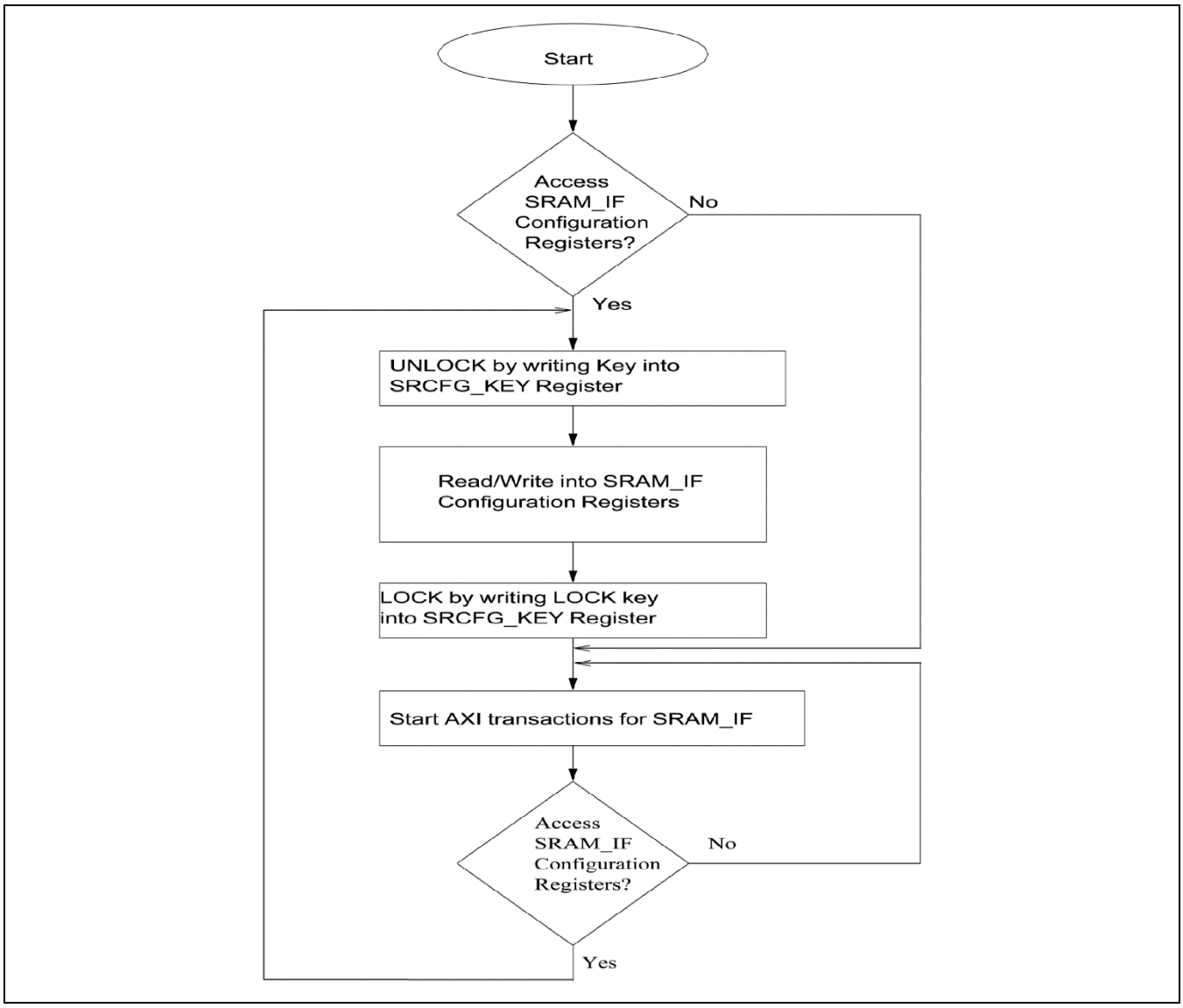
4. How to Use SRAM_IF

The following flowchart shows the sequence of steps to be followed to configure the SRAM_IF Registers.

Typical SRAM_IF Configuration Flow

Figure 4-1 shows the steps to write to SRAM_IF Configuration Registers.

Figure 4-1 Configuration Flowchart



CHAPTER 55: Bit-Band Unit

This chapter provides an overview and notes on bit-band unit

.

1. **Overview**
2. **Explanation of Operation**

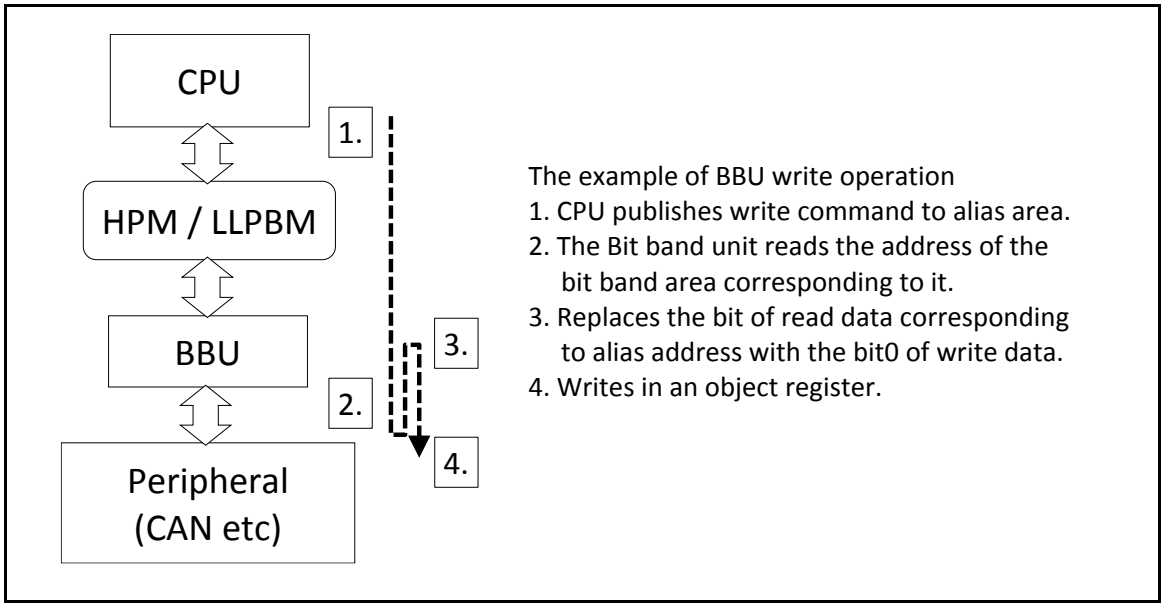
BBU-T00P00R01L02-E1-GBB00

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

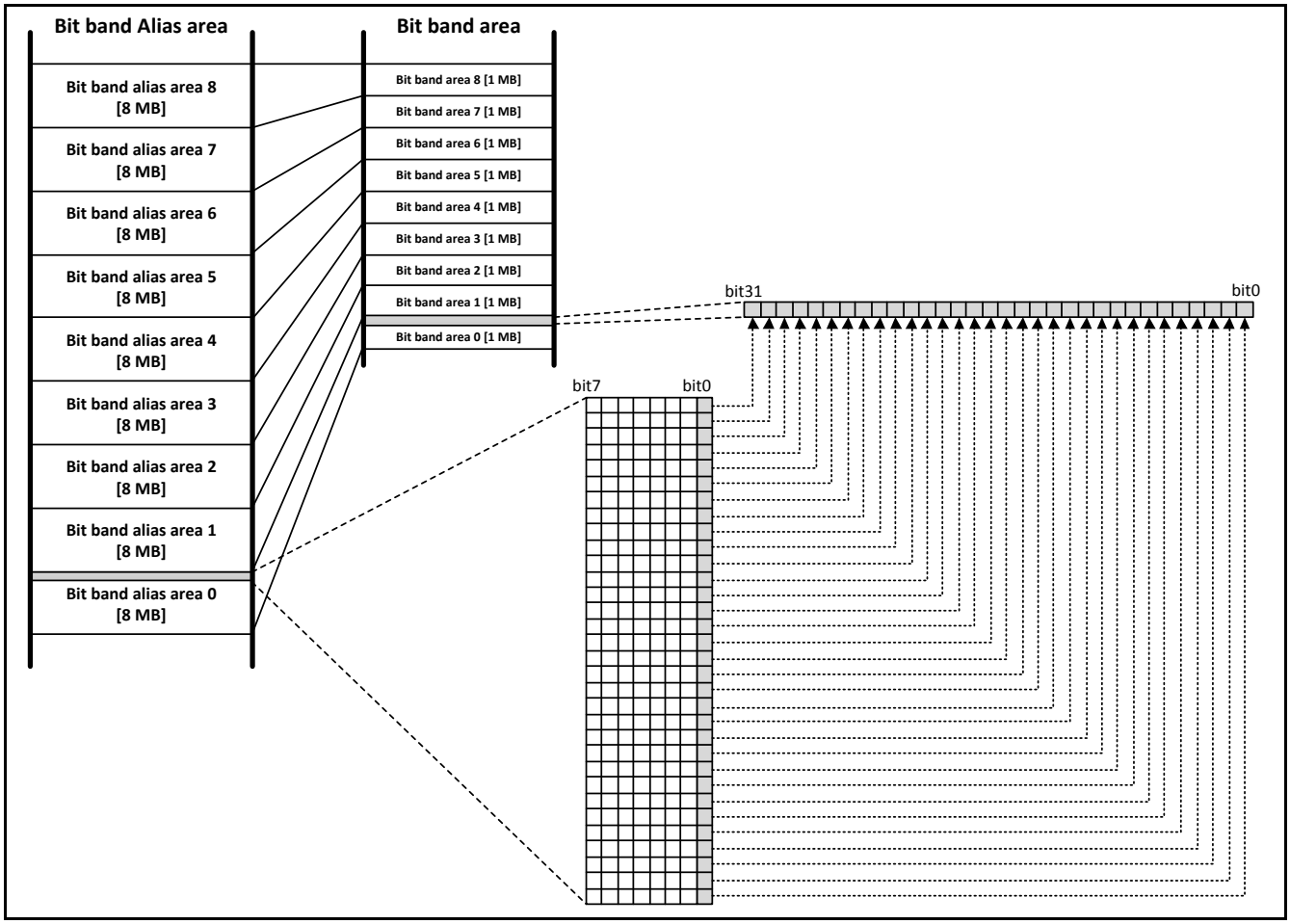


Table 2-1 Correspondence Between Bit-Band Alias Area and Bit-Band Area

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 56: 12-Bit A/D Converter

This chapter explains the 12-bit A/D converter.

1. **Overview**
2. **Features**
3. **Configuration**
4. **Registers**
5. **Explanation of Operation**
6. Notes

ADC12-T00P00R01L01-E1-GBB02

1. Overview

This section provides an overview of the 12-bit A/D converter.

The 12-bit A/D converter has the function of converting analog input voltage into a 12-bit digital value by using RC successive approximation conversion system. It performs A/D conversion upon input of an A/D activation trigger. If another A/D activation trigger is input during A/D conversion, the A/D conversion is restarted. It also supports a forcible stop function, which is started by the A/D conversion cancellation input signal.

It also implements a failure diagnosis function. This function performs A/D conversion while blocking the input from ANx (x: integer number), which is an A/D converter analog input pin, and keeping the internal input voltage level of the A/D converter at the AVRH or AVRL level.
Through the use of this function, you can determine whether there has been a failure in the A/D converter circuit.

2. Features

This section explains the features of the 12-bit A/D converter.

2.1. A/D Activation Compare Function

2.2. A/D Activation Arbitration Functions

2.3. 12-Bit A/D Converter Control Functions

2.1. A/D Activation Compare Function

This section explains the A/D activation compare function.

Analog Input Control

- This A/D converter includes 2 converter units, and can enable or disable each of up to 64 channels of analog input.

Activation Channel

- A/D activation request control and storing of A/D-converted data are performed for each activation channel.
- A/D activation channels correspond to individual units of the 12-bit A/D converter. The correspondence is as follows.
  - Activation channels 0 to 31 : 12-bit A/D converter unit 0
  - Activation channels 32 to 63 : 12-bit A/D converter unit 1
- Each activation channel consists of the following registers.
  - Compare buffer register/compare register
  - A/D activation trigger control status register
  - A/D data register
  - A/D activation trigger extended control register
  - Range comparison control status register
  - Range comparison threshold excess flag register
  - Range comparison flag register
  - Activation channel conversion count setting register
  - Activation channel conversion count completion flag register
  - A/D activation trigger control status clear register
  - Range comparison status register
  - Range comparison flag clear register

A/D activation Request

- An A/D activation request can be selected from among the following.
  - Activation by software
  - Activation by the base timer
    - Reload timer output
    - PWM timer compare match
  - Free-run timer compare match
- In an activation channel, reactivation is impossible during A/D conversion (activation request).
- For the function of the reload timer, see "Reload Timer Function" in "CHAPTER: BASE TIMER."
- For the function of the PWM timer, see "PWM Timer Function" in "CHAPTER: BASE TIMER."
- For software activation, reload timer activation, and PWM timer activation, an arbitrary activation channel can be selected.
- The correspondence between the base timers and 12-bit A/D converter units is as follows.
- By setting register, can be connected to all of Base timers.
- The Base timer to be used is selected by the RESSEL bit in the resource input setting register (RIC_RESIN), and is input to each activation channel. For details of RIC_RESIN, see "Resource Input Setting Register" in "CHAPTER: I/O PORTS."

In the configuration register initial value will be set below.

- Base timer channel 0 corresponds to A/D converter unit 0.
- Base timer channel 2 corresponds to A/D converter unit 1.

- For compare match activation, the A/D activation is requested when the value in the lower 16 bits in the 32-bit free-run timer and the compare register of each activation channel match. The free-run timer to be used is selected by the RESSEL bit in the resource input setting register (RIC_RESIN), and is input to each activation channel. For details of RIC_RESIN, see "Resource Input Setting Register" in "CHAPTER: I/O PORTS."
  RIC_RESINs as many as the number of the activation channels are implemented.
  For relation between RIC_RESINs, free-run timer numbers, and activation channels, see the section of "5.2.3 Free-Run Timer Input."
- In compare match activation, the A/D activation is requested when the free-run timer value and the compare register match during any one of the up-count-only operations, down-count-only operations, or up/down (bidirectional) operations of the 32-bit free-run timer.
- For an activation request, either single mode or repeat mode can be set for each activation channel.
- In single mode, an activation factor causes an activation request. A/D conversion is performed once, and the activation request is canceled upon the completion of the A/D conversion.
- In repeat mode, an activation factor causes a continuous activation request. A/D conversion is performed repeatedly, and the activation request continues until the repeat mode is canceled.

A/D Conversion Data

- When A/D conversion is completed, the converted data is stored in the A/D data register. Each activation channel is equipped with an A/D data register.
- Each A/D data register contains an error flag bit and error status bit. The status of the A/D-converted data can be learned from the values of these bits.

Scan Conversion with the A/D Conversion Count Specified

- Scan conversion with the A/D conversion count specified can be performed for each activation channel.
- 1 type of scan conversion with the A/D conversion count specified can be specified for each 12-bit A/D converter unit.
- As the specification of an A/D conversion count, a number from 1 to 4 can be selected.
- For scan conversion with the A/D conversion count specified, continuous scan conversion mode or paused scan conversion mode can be selected.
- In continuous scan conversion mode, the specified activation channels are started in sequence. After the scan conversion at the last activation channel is completed, conversion is performed repeatedly from the beginning of the scan conversion.
- In paused scan conversion mode, the specified activation channels are started in sequence. After the scan conversion at the last activation channel is completed, operation is stopped. When the next activation factor is input, the scan conversion is performed from the beginning. However, activation factor input that occurs while scan conversion is in progress is ignored.

Range Comparison Function

- The range can be compared for each activation channel.
- Up to 4 types of upper and lower threshold settings can be made. Each activation channel selects 1 combination from among the 4 types of upper and lower threshold settings to perform range comparison.
- For range comparison, checking of whether a value is inside the range of the upper and lower thresholds or checking of whether it is outside the range can be selected.
- Noise reduction can be applied to the range comparison result by using the continuous detection function. The continuous detection function sets the range comparison flag according to the continuous detection of a range comparison result.
- As a continuous detection count, a number from 1 to 7 can be selected.
- The status of the continuous detection count for a range comparison result can be checked.
- For checking whether a value is outside the range for range comparison, the detection status of whether the value is over the upper threshold or below the lower threshold can be checked.

Interrupt Request

- Each activation channel can generate an interrupt request when the A/D conversion is completed.

Data Protection Function

- For each A/D data register, the data protection function can be set. The protection function is effective for factors that are not compare match activation.
- When the data protection function is enabled, the A/D activation request is masked until data is read from the A/D data register and the interrupt flag is cleared. There is no particularly specified order in reading data and clearing interrupt flag. In addition, whether to include the clearing of the interrupt flag in the protection conditions can be selected.
- The status that the device is requesting A/D activation or performing A/D conversion is reported by the A/D activation request in progress bit. In addition, if you want to forcibly stop the current A/D activation request or A/D conversion, it can be stopped by writing "0" in the A/D activation request in progress bit.

2.2. A/D Activation Arbitration Functions

This section explains the functions of the A/D activation arbitration.

- There is A/D activation arbitration for each 12-bit A/D converter unit. For assignment of each A/D activation channel to A/D activation arbitration 0 or 1, see "2.1. A/D Activation Compare Function."
- The A/D activation arbitration consists of the arbitration circuit, A/D activation trigger generation, and analog channel number selection.
- The A/D activation arbitration arbitrates an activation request from the A/D activation compare and generates the activation trigger, A/D conversion cancellation signal, and analog channel number.
- The A/D activation arbitration selects one from among the activation requests from different A/D activation compares to generate the activation trigger. The A/D activation arbitration performs priority control when activation requests from different A/D activation compares conflict with each other. The priority orders are determined according to the following rules: [an activation channel with a smaller channel number takes priority] (priority control based on channel numbers) and [compare match > activation request by the reload timer/PWM timer > software activation] (priority control based on activation factors). Activation requests that are not selected are kept waiting, and arbitration is again applied to them after the A/D conversion that is being processed is completed. Priority control based on activation factors is performed even during A/D conversion. At this time, the current conversion is interrupted and the activation factor with the higher priority order is processed. After the conversion with the higher priority order is completed, the processing for the interrupted activation factor is again arbitrated. In this arbitration, if no activation requests with a higher priority order in terms of channel numbers or activation factors are found, the processing for the interrupted activation factor is restarted.
  - When an activation factor with the same priority order occurs while A/D conversion is stopped:
    The activation factor with the smaller activation channel number is processed first.
  - When an activation factor with a different priority order occurs while A/D conversion is stopped:
    The activation factor with the higher priority order is processed first.
  - When an activation factor with a higher priority order occurs while A/D conversion is in progress:
    The current conversion is interrupted, and the activation factor with the higher priority order is processed. Arbitration is performed again after this conversion, and then the processing of the interrupted activation factor is restarted.
  - When an activation factor with a lower priority order occurs while A/D conversion is in progress:
    Arbitration is again performed after the current conversion is completed, and then the activation factor with the lower priority order is processed.
  - When an activation factor with the same priority order occurs while A/D conversion is in progress:
    Arbitration is again performed after the current conversion is completed, and then the activation factor with the same priority order is processed.
- The conversion cancellation signal is generated to forcibly stop the current conversion processing when the activation factor for which conversion is in progress becomes inactive and the other activation factors are also not active.
- As an analog channel number, an activation request analog number that is input from the activation channel of the activation request arbitration result is selected.

2.3. 12-Bit A/D Converter Control Functions

This chapter explains the functions of the 12-bit A/D converter control.

- There are 12-bit A/D converters 0 and 1, to each of which the analog input pin is assigned. For correspondence between the input pins and the units, see "2.1. A/D Activation Compare Function."

A function for A/D-converting the analog voltage that is input to the analog input pin (input voltage) into a digital value is supported and it has the following features.

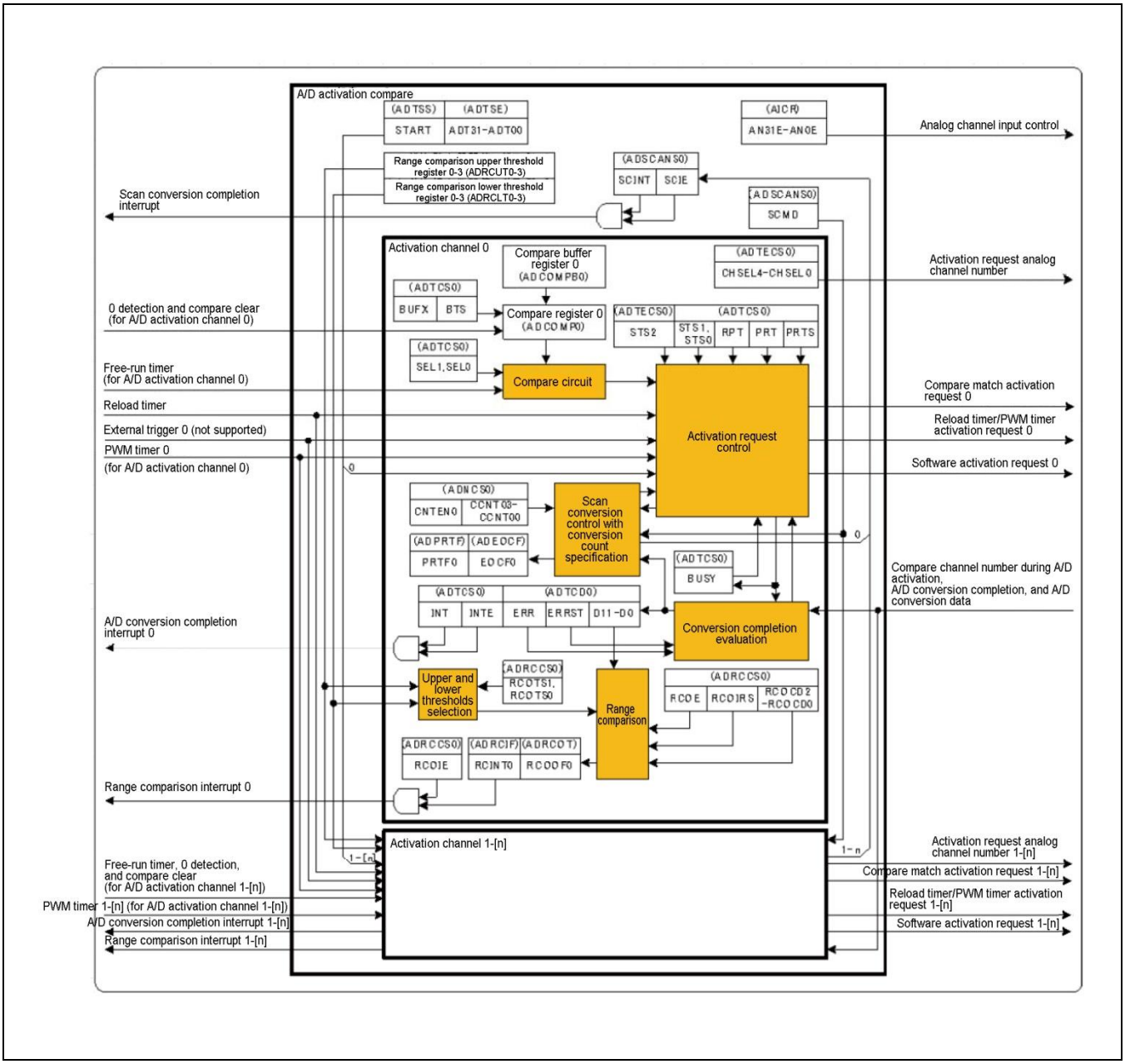
- The conversion system used is an RC successive approximation conversion system with a sample hold circuit.
- The analog input pin can be selected by the registers (make its setting in the A/D activation compare block).
- The activation signal is input as a pulse signal.
- 1 sequence of A/D conversion is performed in response to 1 activation factor.
- If another activation signal is input during A/D conversion, reactivation is performed. (Reactivation function)
- If the A/D conversion cancellation signal is received during A/D conversion, the current processing is stopped and initialized. (Forcible stop function)
- For setting the sampling time, the sampling time setting that is common to all the channels or sampling time setting for each channel can be selected.

3. Configuration

This section explains the configuration of the 12-bit A/D converter.

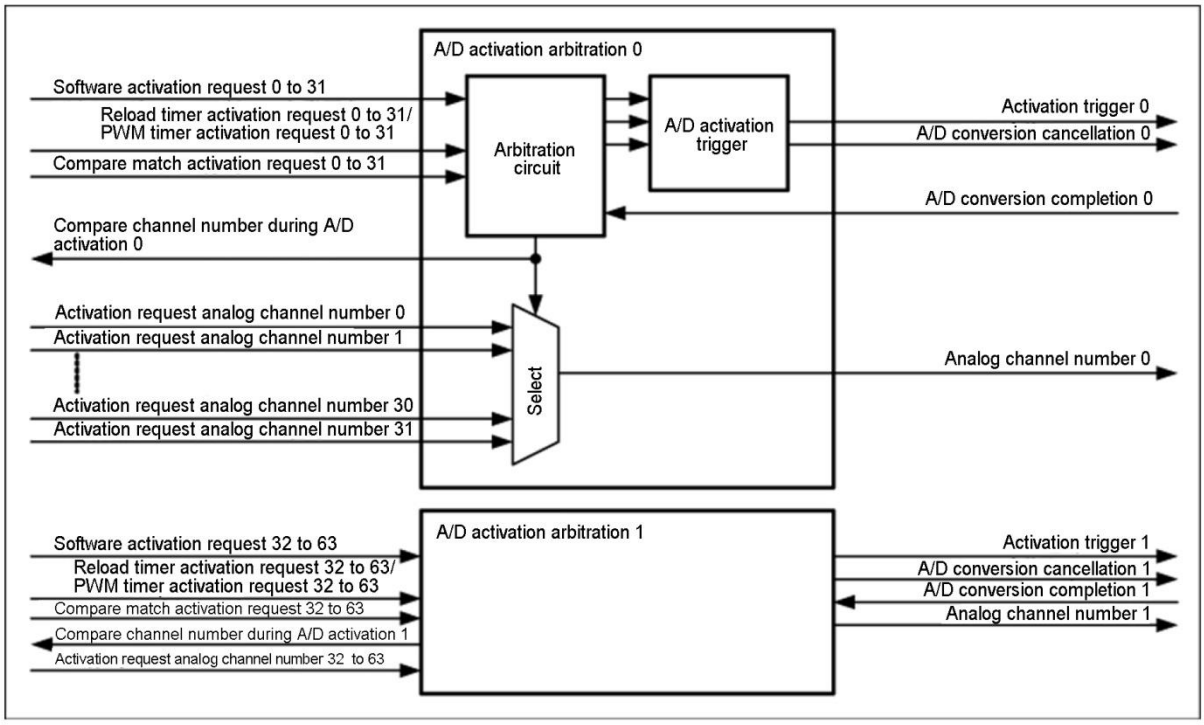
Configuration of the A/D activation Compare

Figure 3-1 Configuration of the A/D Activation Compare (n=31, AD Converter Unit 0)



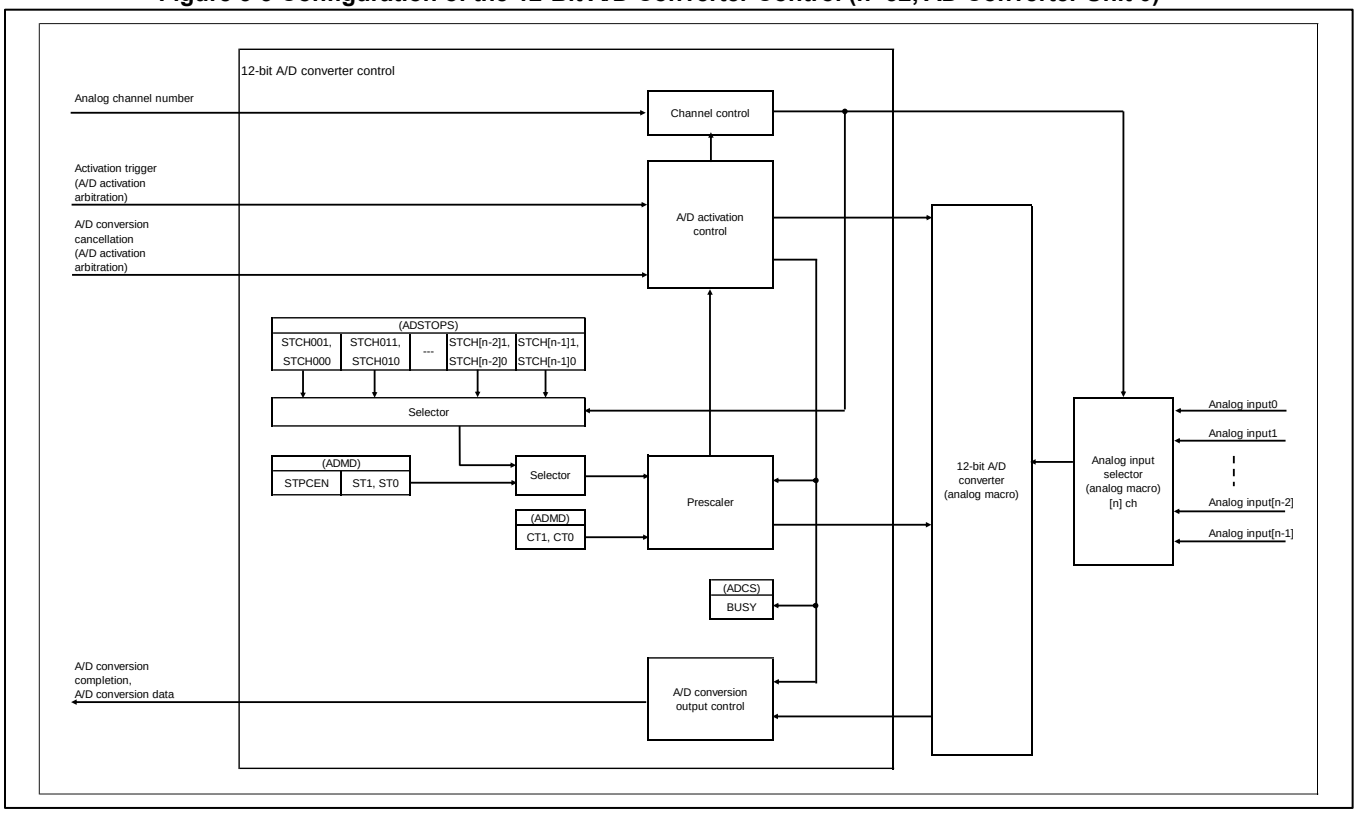
Configuration of the A/D activation Arbitration

Figure 3-2 Configuration of the A/D Activation Arbitration



Configuration of the 12-Bit A/D Converter Control

Figure 3-3 Configuration of the 12-Bit A/D Converter Control (n=32, AD Converter Unit 0)



4. Registers

This section explains the registers of the 12-bit A/D converter.

There is not the restriction by privileged processing mode, when these registers are read or written.

- Registers of the analog input control

Table 4-1 Registers of the Analog Input Control

Address	+3	+2	+1	+0
0x8800	Key code register (KEYCDR)			
0x8804	Analog input control register (ADER0) (Key code target register)			
0x8808	Analog input control register (ADER1) (Key code target register)			
0x81D4	Failure diagnosis function control register (ADFDCTR0)			
0x85D4	Failure diagnosis function control register (ADFDCTR1)			

- Registers of the A/D activation compare

Table 4-2 Registers of the A/D Activation Compare

Address	+3	+2	+1	+0
0x8004	Reserved	Reserved	Reserved	A/D software activation register 0 (ADTSS0)
0x8008	A/D software activation channel selection register 0 (ADTSE0)			
0x800C	Compare buffer register 1 (ADCOMPB1) Compare register 1 (ADCOMP1)		Compare buffer register 0 (ADCOMPB0) Compare register 0 (ADCOMP0)	
0x8010	Compare buffer register 3 (ADCOMPB3) Compare register 3 (ADCOMP3)		Compare buffer register 2 (ADCOMPB2) Compare register 2 (ADCOMP2)	
0x8014	Compare buffer register 5 (ADCOMPB5) Compare register 5 (ADCOMP5)		Compare buffer register 4 (ADCOMPB4) Compare register 4 (ADCOMP4)	
0x8018	Compare buffer register 7 (ADCOMPB7) Compare register 7 (ADCOMP7)		Compare buffer register 6 (ADCOMPB6) Compare register 6 (ADCOMP6)	
0x801C	Compare buffer register 9 (ADCOMPB9) Compare register 9 (ADCOMP9)		Compare buffer register 8 (ADCOMPB8) Compare register 8 (ADCOMP8)	
0x8020	Compare buffer register 11 (ADCOMPB11) Compare register 11 (ADCOMP11)		Compare buffer register 10 (ADCOMPB10) Compare register 10 (ADCOMP10)	
0x8024	Compare buffer register 13 (ADCOMPB13) Compare register 13 (ADCOMP13)		Compare buffer register 12 (ADCOMPB12) Compare register 12 (ADCOMP12)	
0x8028	Compare buffer register 15 (ADCOMPB15) Compare register 15 (ADCOMP15)		Compare buffer register 14 (ADCOMPB14) Compare register 14 (ADCOMP14)	
0x802C	Compare buffer register 17 (ADCOMPB17) Compare register 17 (ADCOMP17)		Compare buffer register 16 (ADCOMPB16) Compare register 16 (ADCOMP16)	
0x8030	Compare buffer register 19 (ADCOMPB19) Compare register 19 (ADCOMP19)		Compare buffer register 18 (ADCOMPB18) Compare register 18 (ADCOMP18)	
0x8034	Compare buffer register 21 (ADCOMPB21) Compare register 21 (ADCOMP21)		Compare buffer register 20 (ADCOMPB20) Compare register 20 (ADCOMP20)	
0x8038	Compare buffer register 23 (ADCOMPB23) Compare register 23 (ADCOMP23)		Compare buffer register 22 (ADCOMPB22) Compare register 22 (ADCOMP22)	
0x803C	Compare buffer register 25 (ADCOMPB25) Compare register 25 (ADCOMP25)		Compare buffer register 24 (ADCOMPB24) Compare register 24 (ADCOMP24)	

Address	+3	+2	+1	+0
0x8040	Compare buffer register 27 (ADCOMPB27) Compare register 27 (ADCOMP27)		Compare buffer register 26 (ADCOMPB26) Compare register 26 (ADCOMP26)	
0x8044	Compare buffer register 29 (ADCOMPB29) Compare register 29 (ADCOMP29)		Compare buffer register 28 (ADCOMPB28) Compare register 28 (ADCOMP28)	
0x8048	Compare buffer register 31 (ADCOMPB31) Compare register 31 (ADCOMP31)		Compare buffer register 30 (ADCOMPB30) Compare register 30 (ADCOMP30)	
0x804C	A/D activation trigger control status register 1 (ADTCS1)		A/D activation trigger control status register 0 (ADTCS0)	
0x8050	A/D activation trigger control status register 3 (ADTCS3)		A/D activation trigger control status register 2 (ADTCS2)	
0x8054	A/D activation trigger control status register 5 (ADTCS5)		A/D activation trigger control status register 4 (ADTCS4)	
0x8058	A/D activation trigger control status register 7 (ADTCS7)		A/D activation trigger control status register 6 (ADTCS6)	
0x805C	A/D activation trigger control status register 9 (ADTCS9)		A/D activation trigger control status register 8 (ADTCS8)	
0x8060	A/D activation trigger control status register 11 (ADTCS11)		A/D activation trigger control status register 10 (ADTCS10)	
0x8064	A/D activation trigger control status register 13 (ADTCS13)		A/D activation trigger control status register 12 (ADTCS12)	
0x8068	A/D activation trigger control status register 15 (ADTCS15)		A/D activation trigger control status register 14 (ADTCS14)	
0x806C	A/D activation trigger control status register 17 (ADTCS17)		A/D activation trigger control status register 16 (ADTCS16)	
0x8070	A/D activation trigger control status register 19 (ADTCS19)		A/D activation trigger control status register 18 (ADTCS18)	
0x8074	A/D activation trigger control status register 21 (ADTCS21)		A/D activation trigger control status register 20 (ADTCS20)	
0x8078	A/D activation trigger control status register 23 (ADTCS23)		A/D activation trigger control status register 22 (ADTCS22)	
0x807C	A/D activation trigger control status register 25 (ADTCS25)		A/D activation trigger control status register 24 (ADTCS24)	
0x8080	A/D activation trigger control status register 27 (ADTCS27)		A/D activation trigger control status register 26 (ADTCS26)	
0x8084	A/D activation trigger control status register 29 (ADTCS29)		A/D activation trigger control status register 28 (ADTCS28)	
0x8088	A/D activation trigger control status register 31 (ADTCS31)		A/D activation trigger control status register 30 (ADTCS30)	
0x808C	A/D data register 1 (ADTCD1)		A/D data register 0 (ADTCD0)	
0x8090	A/D data register 3 (ADTCD3)		A/D data register 2 (ADTCD2)	
0x8094	A/D data register 5 (ADTCD5)		A/D data register 4 (ADTCD4)	
0x8098	A/D data register 7 (ADTCD7)		A/D data register 6 (ADTCD6)	
0x809C	A/D data register 9 (ADTCD9)		A/D data register 8 (ADTCD8)	
0x80A0	A/D data register 11 (ADTCD11)		A/D data register 10 (ADTCD10)	
0x80A4	A/D data register 13 (ADTCD13)		A/D data register 12 (ADTCD12)	
0x80A8	A/D data register 15 (ADTCD15)		A/D data register 14 (ADTCD14)	

Address	+3	+2	+1	+0
0x80AC	A/D data register 17 (ADTCD17)		A/D data register 16 (ADTCD16)	
0x80B0	A/D data register 19 (ADTCD19)		A/D data register 18 (ADTCD18)	
0x80B4	A/D data register 21 (ADTCD21)		A/D data register 20 (ADTCD20)	
0x80B8	A/D data register 23 (ADTCD23)		A/D data register 22 (ADTCD22)	
0x80BC	A/D data register 25 (ADTCD25)		A/D data register 24 (ADTCD24)	
0x80C0	A/D data register 27 (ADTCD27)		A/D data register 26 (ADTCD26)	
0x80C4	A/D data register 29 (ADTCD29)		A/D data register 28 (ADTCD28)	
0x80C8	A/D data register 31 (ADTCD31)		A/D data register 30 (ADTCD30)	
0x80CC	A/D activation trigger extended control register 1 (ADTECS1)		A/D activation trigger extended control register 0 (ADTECS0)	
0x80D0	A/D activation trigger extended control register 3 (ADTECS3)		A/D activation trigger extended control register 2 (ADTECS2)	
0x80D4	A/D activation trigger extended control register 5 (ADTECS5)		A/D activation trigger extended control register 4 (ADTECS4)	
0x80D8	A/D activation trigger extended control register 7 (ADTECS7)		A/D activation trigger extended control register 6 (ADTECS6)	
0x80DC	A/D activation trigger extended control register 9 (ADTECS9)		A/D activation trigger extended control register 8 (ADTECS8)	
0x80E0	A/D activation trigger extended control register 11 (ADTECS11)		A/D activation trigger extended control register 10 (ADTECS10)	
0x80E4	A/D activation trigger extended control register 13 (ADTECS13)		A/D activation trigger extended control register 12 (ADTECS12)	
0x80E8	A/D activation trigger extended control register 15 (ADTECS15)		A/D activation trigger extended control register 14 (ADTECS14)	
0x80EC	A/D activation trigger extended control register 17 (ADTECS17)		A/D activation trigger extended control register 16 (ADTECS16)	
0x80F0	A/D activation trigger extended control register 19 (ADTECS19)		A/D activation trigger extended control register 18 (ADTECS18)	
0x80F4	A/D activation trigger extended control register 21 (ADTECS21)		A/D activation trigger extended control register 20 (ADTECS20)	
0x80F8	A/D activation trigger extended control register 23 (ADTECS23)		A/D activation trigger extended control register 22 (ADTECS22)	
0x80FC	A/D activation trigger extended control register 25 (ADTECS25)		A/D activation trigger extended control register 24 (ADTECS24)	
0x8100	A/D activation trigger extended control register 27 (ADTECS27)		A/D activation trigger extended control register 26 (ADTECS26)	
0x8104	A/D activation trigger extended control register 29 (ADTECS29)		A/D activation trigger extended control register 28 (ADTECS28)	
0x8108	A/D activation trigger extended control register 31 (ADTECS31)		A/D activation trigger extended control register 30 (ADTECS30)	
0x810C	Lower threshold setting register 0 (ADRCLT0)		Upper threshold setting register 0 (ADRCUT0)	
0x8110	Lower threshold setting register 1 (ADRCLT1)		Upper threshold setting register 1 (ADRCUT1)	
0x8114	Lower threshold setting register 2 (ADRCLT2)		Upper threshold setting register 2 (ADRCUT2)	

Address	+3	+2	+1	+0
0x8118	Lower threshold setting register 3 (ADRCLT3)		Upper threshold setting register 3 (ADRCUT3)	
0x811C	Range comparison control status register 3 (ADRCCS3)	Range comparison control status register 2 (ADRCCS2)	Range comparison control status register 1 (ADRCCS1)	Range comparison control status register 0 (ADRCCS0)
0x8120	Range comparison control status register 7 (ADRCCS7)	Range comparison control status register 6 (ADRCCS6)	Range comparison control status register 5 (ADRCCS5)	Range comparison control status register 4 (ADRCCS4)
0x8124	Range comparison control status register 11 (ADRCCS11)	Range comparison control status register 10 (ADRCCS10)	Range comparison control status register 9 (ADRCCS9)	Range comparison control status register 8 (ADRCCS8)
0x8128	Range comparison control status register 15 (ADRCCS15)	Range comparison control status register 14 (ADRCCS14)	Range comparison control status register 13 (ADRCCS13)	Range comparison control status register 12 (ADRCCS12)
0x812C	Range comparison control status register 19 (ADRCCS19)	Range comparison control status register 18 (ADRCCS18)	Range comparison control status register 17 (ADRCCS17)	Range comparison control status register 16 (ADRCCS16)
0x8130	Range comparison control status register 23 (ADRCCS23)	Range comparison control status register 22 (ADRCCS22)	Range comparison control status register 21 (ADRCCS21)	Range comparison control status register 20 (ADRCCS20)
0x8134	Range comparison control status register 27 (ADRCCS27)	Range comparison control status register 26 (ADRCCS26)	Range comparison control status register 25 (ADRCCS25)	Range comparison control status register 24 (ADRCCS24)
0x8138	Range comparison control status register 31 (ADRCCS31)	Range comparison control status register 30 (ADRCCS30)	Range comparison control status register 29 (ADRCCS29)	Range comparison control status register 28 (ADRCCS28)
0x813C	Range comparison threshold excess flag register 0 (ADRCOT0)			
0x8140	Range comparison flag register 0 (ADRCIF0)			
0x8144	Reserved	Reserved	Reserved	Scan conversion control status register 0 (ADSCANS0)
0x8148	Activation channel conversion count setting register 3 (ADNCS3)	Activation channel conversion count setting register 2 (ADNCS2)	Activation channel conversion count setting register 1 (ADNCS1)	Activation channel conversion count setting register 0 (ADNCS0)

Address	+3	+2	+1	+0
0x814C	Activation channel conversion count setting register 7 (ADNCS7)	Activation channel conversion count setting register 6 (ADNCS6)	Activation channel conversion count setting register 5 (ADNCS5)	Activation channel conversion count setting register 4 (ADNCS4)
0x8150	Activation channel conversion count setting register 11 (ADNCS11)	Activation channel conversion count setting register 10 (ADNCS10)	Activation channel conversion count setting register 9 (ADNCS9)	Activation channel conversion count setting register 8 (ADNCS8)
0x8154	Activation channel conversion count setting register 15 (ADNCS15)	Activation channel conversion count setting register 14 (ADNCS14)	Activation channel conversion count setting register 13 (ADNCS13)	Activation channel conversion count setting register 12 (ADNCS12)
0x8158	Data protection status flag register 0 (ADPRTF0)			
0x815C	Activation channel conversion completion flag register 0 (ADEOCF0)			
0x8160	A/D activation trigger control status clear register 1 (ADTCSC1)		A/D activation trigger control status clear register 0 (ADTCSC0)	
0x8164	A/D activation trigger control status clear register 3 (ADTCSC3)		A/D activation trigger control status clear register 2 (ADTCSC2)	
0x8168	A/D activation trigger control status clear register 5 (ADTCSC5)		A/D activation trigger control status clear register 4 (ADTCSC4)	
0x816C	A/D activation trigger control status clear register 7 (ADTCSC7)		A/D activation trigger control status clear register 6 (ADTCSC6)	
0x8170	A/D activation trigger control status clear register 9 (ADTCSC9)		A/D activation trigger control status clear register 8 (ADTCSC8)	
0x8174	A/D activation trigger control status clear register 11 (ADTCSC11)		A/D activation trigger control status clear register 10 (ADTCSC10)	
0x8178	A/D activation trigger control status clear register 13 (ADTCSC13)		A/D activation trigger control status clear register 12 (ADTCSC12)	
0x817C	A/D activation trigger control status clear register 15 (ADTCSC15)		A/D activation trigger control status clear register 14 (ADTCSC14)	
0x8180	A/D activation trigger control status clear register 17 (ADTCSC17)		A/D activation trigger control status clear register 16 (ADTCSC16)	
0x8184	A/D activation trigger control status clear register 19 (ADTCSC19)		A/D activation trigger control status clear register 18 (ADTCSC18)	
0x8188	A/D activation trigger control status clear register 21 (ADTCSC21)		A/D activation trigger control status clear register 20 (ADTCSC20)	
0x818C	A/D activation trigger control status clear register 23 (ADTCSC23)		A/D activation trigger control status clear register 22 (ADTCSC22)	
0x8190	A/D activation trigger control status clear register 25 (ADTCSC25)		A/D activation trigger control status clear register 24 (ADTCSC24)	
0x8194	A/D activation trigger control status clear register 27 (ADTCSC27)		A/D activation trigger control status clear register 26 (ADTCSC26)	
0x8198	A/D activation trigger control status clear register 29 (ADTCSC29)		A/D activation trigger control status clear register 28 (ADTCSC28)	

Address	+3	+2	+1	+0
0x819C	A/D activation trigger control status clear register 31 (ADTCSC31)		A/D activation trigger control status clear register 30 (ADTCSC30)	
0x81A0	Range comparison status register 3 (ADRCS3)	Range comparison status register 2 (ADRCS2)	Range comparison status register 1 (ADRCS1)	Range comparison status register 0 (ADRCS0)
0x81A4	Range comparison status register 7 (ADRCS7)	Range comparison status register 6 (ADRCS6)	Range comparison status register 5 (ADRCS5)	Range comparison status register 4 (ADRCS4)
0x81A8	Range comparison status register 11 (ADRCS11)	Range comparison status register 10 (ADRCS10)	Range comparison status register 9 (ADRCS9)	Range comparison status register 8 (ADRCS8)
0x81AC	Range comparison status register 15 (ADRCS15)	Range comparison status register 14 (ADRCS14)	Range comparison status register 13 (ADRCS13)	Range comparison status register 12 (ADRCS12)
0x81B0	Range comparison status register 19 (ADRCS19)	Range comparison status register 18 (ADRCS18)	Range comparison status register 17 (ADRCS17)	Range comparison status register 16 (ADRCS16)
0x81B4	Range comparison status register 23 (ADRCS23)	Range comparison status register 22 (ADRCS22)	Range comparison status register 21 (ADRCS21)	Range comparison status register 20 (ADRCS20)
0x81B8	Range comparison status register 27 (ADRCS27)	Range comparison status register 26 (ADRCS26)	Range comparison status register 25 (ADRCS25)	Range comparison status register 24 (ADRCS24)
0x81BC	Range comparison status register 31 (ADRCS31)	Range comparison status register 30 (ADRCS30)	Range comparison status register 29 (ADRCS29)	Range comparison status register 28 (ADRCS28)
0x81C0	Range comparison flag clear register 0 (ADRCIFC0)			
0x81C4	Reserved	Reserved	Reserved	Scan conversion control status clear register 0 (ADSCANSC0)
0x8404	Reserved	Reserved	Reserved	A/D software activation register 1 (ADTSS1)
0x8408	A/D software activation channel selection register 1 (ADTSE1)			
0x840C	Compare buffer register 33 (ADCOMPB33) Compare register 33 (ADCOMP33)		Compare buffer register 32 (ADCOMPB32) Compare register 32 (ADCOMP32)	
0x8410	Compare buffer register 35 (ADCOMPB35) Compare register 35 (ADCOMP35)		Compare buffer register 34 (ADCOMPB34) Compare register 34 (ADCOMP34)	
0x8414	Compare buffer register 37 (ADCOMPB37) Compare register 37 (ADCOMP37)		Compare buffer register 36 (ADCOMPB36) Compare register 36 (ADCOMP36)	
0x8418	Compare buffer register 39 (ADCOMPB39) Compare register 39 (ADCOMP39)		Compare buffer register 38 (ADCOMPB38) Compare register 38 (ADCOMP38)	
0x841C	Compare buffer register 41 (ADCOMPB41) Compare register 41 (ADCOMP41)		Compare buffer register 40 (ADCOMPB40) Compare register 40 (ADCOMP40)	
0x8420	Compare buffer register 43 (ADCOMPB43) Compare register 43 (ADCOMP43)		Compare buffer register 42 (ADCOMPB42) Compare register 42 (ADCOMP42)	

Address	+3	+2	+1	+0
0x8424	Compare buffer register 45 (ADCOMPB45) Compare register 45 (ADCOMP45)		Compare buffer register 44 (ADCOMPB44) Compare register 44 (ADCOMP44)	
0x8428	Compare buffer register 47 (ADCOMPB47) Compare register 47 (ADCOMP47)		Compare buffer register 46 (ADCOMPB46) Compare register 46 (ADCOMP46)	
0x842C	Compare buffer register 49 (ADCOMPB49) Compare register 49 (ADCOMP49)		Compare buffer register 48 (ADCOMPB48) Compare register 48 (ADCOMP48)	
0x8430	Compare buffer register 51 (ADCOMPB51) Compare register 51 (ADCOMP51)		Compare buffer register 50 (ADCOMPB50) Compare register 50 (ADCOMP50)	
0x8434	Compare buffer register 53 (ADCOMPB53) Compare register 53 (ADCOMP53)		Compare buffer register 52 (ADCOMPB52) Compare register 52 (ADCOMP52)	
0x8438	Compare buffer register 55 (ADCOMPB55) Compare register 55 (ADCOMP55)		Compare buffer register 54 (ADCOMPB54) Compare register 54 (ADCOMP54)	
0x843C	Compare buffer register 57 (ADCOMPB57) Compare register 57 (ADCOMP57)		Compare buffer register 56 (ADCOMPB56) Compare register 56 (ADCOMP56)	
0x8440H	Compare buffer register 59 (ADCOMPB59) Compare register 59 (ADCOMP59)		Compare buffer register 58 (ADCOMPB58) Compare register 58 (ADCOMP58)	
0x8444	Compare buffer register 61 (ADCOMPB61) Compare register 61 (ADCOMP61)		Compare buffer register 60 (ADCOMPB60) Compare register 60 (ADCOMP60)	
0x8448	Compare buffer register 63 (ADCOMPB63) Compare register 63 (ADCOMP63)		Compare buffer register 62 (ADCOMPB62) Compare register 62 (ADCOMP62)	
0x844C	A/D activation trigger control status register 33 (ADTCS33)		A/D activation trigger control status register 32 (ADTCS32)	
0x8450	A/D activation trigger control status register 35 (ADTCS35)		A/D activation trigger control status register 34 (ADTCS34)	
0x8454	A/D activation trigger control status register 37 (ADTCS37)		A/D activation trigger control status register 36 (ADTCS36)	
0x8458	A/D activation trigger control status register 39 (ADTCS39)		A/D activation trigger control status register 38 (ADTCS38)	
0x845C	A/D activation trigger control status register 41 (ADTCS41)		A/D activation trigger control status register 40 (ADTCS40)	
0x8460	A/D activation trigger control status register 43 (ADTCS43)		A/D activation trigger control status register 42 (ADTCS42)	
0x8464	A/D activation trigger control status register 45 (ADTCS45)		A/D activation trigger control status register 44 (ADTCS44)	
0x8468	A/D activation trigger control status register 47 (ADTCS47)		A/D activation trigger control status register 46 (ADTCS46)	
0x846C	A/D activation trigger control status register 49 (ADTCS49)		A/D activation trigger control status register 48 (ADTCS48)	
0x8470	A/D activation trigger control status register 51 (ADTCS51)		A/D activation trigger control status register 50 (ADTCS50)	
0x8474	A/D activation trigger control status register 53 (ADTCS53)		A/D activation trigger control status register 52 (ADTCS52)	
0x8478	A/D activation trigger control status register 55 (ADTCS55)		A/D activation trigger control status register 54 (ADTCS54)	
0x847C	A/D activation trigger control status register 57 (ADTCS57)		A/D activation trigger control status register 56 (ADTCS56)	

Address	+3	+2	+1	+0
0x8480	A/D activation trigger control status register 59 (ADTCS59)		A/D activation trigger control status register 58 (ADTCS58)	
0x8484	A/D activation trigger control status register 61 (ADTCS61)		A/D activation trigger control status register 60 (ADTCS60)	
0x8488	A/D activation trigger control status register 63 (ADTCS63)		A/D activation trigger control status register 62 (ADTCS62)	
0x848C	A/D data register 33 (ADTCD33)		A/D data register 32 (ADTCD32)	
0x8490	A/D data register 35 (ADTCD35)		A/D data register 34 (ADTCD34)	
0x8494	A/D data register 37 (ADTCD37)		A/D data register 36 (ADTCD36)	
0x8498	A/D data register 39 (ADTCD39)		A/D data register 38 (ADTCD38)	
0x849C	A/D data register 41 (ADTCD41)		A/D data register 40 (ADTCD40)	
0x84A0	A/D data register 43 (ADTCD43)		A/D data register 42 (ADTCD42)	
0x84A4	A/D data register 45 (ADTCD45)		A/D data register 44 (ADTCD44)	
0x84A8	A/D data register 47 (ADTCD47)		A/D data register 46 (ADTCD46)	
0x84AC	A/D data register 49 (ADTCD49)		A/D data register 48 (ADTCD48)	
0x84B0	A/D data register 51 (ADTCD51)		A/D data register 50 (ADTCD50)	
0x84B4	A/D data register 53 (ADTCD53)		A/D data register 52 (ADTCD52)	
0x84B8	A/D data register 55 (ADTCD55)		A/D data register 54 (ADTCD54)	
0x84BC	A/D data register 57 (ADTCD57)		A/D data register 56 (ADTCD56)	
0x84C0	A/D data register 59 (ADTCD59)		A/D data register 58 (ADTCD58)	
0x84C4	A/D data register 61 (ADTCD61)		A/D data register 60 (ADTCD60)	
0x84C8	A/D data register 63 (ADTCD63)		A/D data register 62 (ADTCD62)	
0x84CC	A/D activation trigger extended control register 33 (ADTECS33)		A/D activation trigger extended control register 32 (ADTECS32)	
0x84D0	A/D activation trigger extended control register 35 (ADTECS35)		A/D activation trigger extended control register 34 (ADTECS34)	
0x84D4	A/D activation trigger extended control register 37 (ADTECS37)		A/D activation trigger extended control register 36 (ADTECS36)	
0x84D8	A/D activation trigger extended control register 39 (ADTECS39)		A/D activation trigger extended control register 38 (ADTECS38)	
0x84DC	A/D activation trigger extended control register 41 (ADTECS41)		A/D activation trigger extended control register 40 (ADTECS40)	
0x84E0	A/D activation trigger extended control register 43 (ADTECS43)		A/D activation trigger extended control register 42 (ADTECS42)	
0x84E4	A/D activation trigger extended control register 45 (ADTECS45)		A/D activation trigger extended control register 44 (ADTECS44)	
0x84E8	A/D activation trigger extended control register 47 (ADTECS47)		A/D activation trigger extended control register 46 (ADTECS46)	
0x84EC	A/D activation trigger extended control register 49 (ADTECS49)		A/D activation trigger extended control register 48 (ADTECS48)	
0x84F0	A/D activation trigger extended control register 51 (ADTECS51)		A/D activation trigger extended control register 50 (ADTECS50)	
0x84F4	A/D activation trigger extended control register 53 (ADTECS53)		A/D activation trigger extended control register 52 (ADTECS52)	
0x84F8	A/D activation trigger extended control register 55 (ADTECS55)		A/D activation trigger extended control register 54 (ADTECS54)	

Address	+3	+2	+1	+0
0x84FC	A/D activation trigger extended control register 57 (ADTECS57)		A/D activation trigger extended control register 56 (ADTECS56)	
0x8500	A/D activation trigger extended control register 59 (ADTECS59)		A/D activation trigger extended control register 58 (ADTECS58)	
0x8504	A/D activation trigger extended control register 61 (ADTECS61)		A/D activation trigger extended control register 60 (ADTECS60)	
0x8508	A/D activation trigger extended control register 63 (ADTECS63)		A/D activation trigger extended control register 62 (ADTECS62)	
0x850C	Lower threshold setting register 4 (ADRCLT4)		Upper threshold setting register 4 (ADRCUT4)	
0x8510	Lower threshold setting register 5 (ADRCLT5)		Upper threshold setting register 5 (ADRCUT5)	
0x8514	Lower threshold setting register 6 (ADRCLT6)		Upper threshold setting register 6 (ADRCUT6)	
0x8518	Lower threshold setting register 7 (ADRCLT7)		Upper threshold setting register 7 (ADRCUT7)	
0x851C	Range comparison control status register 35 (ADRCCS35)	Range comparison control status register 34 (ADRCCS34)	Range comparison control status register 33 (ADRCCS33)	Range comparison control status register 32 (ADRCCS32)
0x8520	Range comparison control status register 39 (ADRCCS39)	Range comparison control status register 38 (ADRCCS38)	Range comparison control status register 37 (ADRCCS37)	Range comparison control status register 36 (ADRCCS36)
0x8524	Range comparison control status register 43 (ADRCCS43)	Range comparison control status register 42 (ADRCCS42)	Range comparison control status register 41 (ADRCCS41)	Range comparison control status register 40 (ADRCCS40)
0x8528	Range comparison control status register 47 (ADRCCS47)	Range comparison control status register 46 (ADRCCS46)	Range comparison control status register 45 (ADRCCS45)	Range comparison control status register 44 (ADRCCS44)
0x852C	Range comparison control status register 51 (ADRCCS51)	Range comparison control status register 50 (ADRCCS50)	Range comparison control status register 49 (ADRCCS49)	Range comparison control status register 48 (ADRCCS48)
0x8530	Range comparison control status register 55 (ADRCCS55)	Range comparison control status register 54 (ADRCCS54)	Range comparison control status register 53 (ADRCCS53)	Range comparison control status register 52 (ADRCCS52)
0x8534	Range comparison control status register 59 (ADRCCS59)	Range comparison control status register 58 (ADRCCS58)	Range comparison control status register 57 (ADRCCS57)	Range comparison control status register 56 (ADRCCS56)

Address	+3	+2	+1	+0
0x8538	Range comparison control status register 63 (ADRCCS63)	Range comparison control status register 62 (ADRCCS62)	Range comparison control status register 61 (ADRCCS61)	Range comparison control status register 60 (ADRCCS60)
0x853C	Range comparison threshold excess flag register 1 (ADRCOT1)			
0x8540	Range comparison flag register 1 (ADRCIF1)			
0x8544	Reserved	Reserved	Reserved	Scan conversion control status register 1 (ADSCANS1)
0x8548	Activation channel conversion count setting register 19 (ADNCS19)	Activation channel conversion count setting register 18 (ADNCS18)	Activation channel conversion count setting register 17 (ADNCS17)	Activation channel conversion count setting register 16 (ADNCS16)
0x854C	Activation channel conversion count setting register 23 (ADNCS23)	Activation channel conversion count setting register 22 (ADNCS22)	Activation channel conversion count setting register 21 (ADNCS21)	Activation channel conversion count setting register 20 (ADNCS20)
0x8550	Activation channel conversion count setting register 27 (ADNCS27)	Activation channel conversion count setting register 26 (ADNCS26)	Activation channel conversion count setting register 25 (ADNCS25)	Activation channel conversion count setting register 24 (ADNCS24)
0x8554	Activation channel conversion count setting register 31 (ADNCS31)	Activation channel conversion count setting register 30 (ADNCS30)	Activation channel conversion count setting register 29 (ADNCS29)	Activation channel conversion count setting register 28 (ADNCS28)
0x8558	Data protection status flag register 1 (ADPRTF1)			
0x855C	Activation channel conversion completion flag register 1 (ADEOCF1)			
0x8560	A/D activation trigger control status clear register 33 (ADTCSC33)		A/D activation trigger control status clear register 32 (ADTCSC32)	
0x8564	A/D activation trigger control status clear register 35 (ADTCSC35)		A/D activation trigger control status clear register 34 (ADTCSC34)	
0x8568	A/D activation trigger control status clear register 37 (ADTCSC37)		A/D activation trigger control status clear register 36 (ADTCSC36)	
0x856C	A/D activation trigger control status clear register 39 (ADTCSC39)		A/D activation trigger control status clear register 38 (ADTCSC38)	
0x8570	A/D activation trigger control status clear register 41 (ADTCSC41)		A/D activation trigger control status clear register 40 (ADTCSC40)	
0x8574	A/D activation trigger control status clear register 43 (ADTCSC43)		A/D activation trigger control status clear register 42 (ADTCSC42)	
0x8578	A/D activation trigger control status clear register 45 (ADTCSC45)		A/D activation trigger control status clear register 44 (ADTCSC44)	
0x857C	A/D activation trigger control status clear register 47 (ADTCSC47)		A/D activation trigger control status clear register 46 (ADTCSC46)	
0x8580	A/D activation trigger control status clear register 49 (ADTCSC49)		A/D activation trigger control status clear register 48 (ADTCSC48)	

Address	+3	+2	+1	+0
0x8584	A/D activation trigger control status clear register 51 (ADTCSC51)		A/D activation trigger control status clear register 50 (ADTCSC50)	
0x8588	A/D activation trigger control status clear register 53 (ADTCSC53)		A/D activation trigger control status clear register 52 (ADTCSC52)	
0x858C	A/D activation trigger control status clear register 55 (ADTCSC55)		A/D activation trigger control status clear register 54 (ADTCSC54)	
0x8590	A/D activation trigger control status clear register 57 (ADTCSC57)		A/D activation trigger control status clear register 56 (ADTCSC56)	
0x8594	A/D activation trigger control status clear register 59 (ADTCSC59)		A/D activation trigger control status clear register 58 (ADTCSC58)	
0x8598	A/D activation trigger control status clear register 61 (ADTCSC61)		A/D activation trigger control status clear register 60 (ADTCSC60)	
0x859C	A/D activation trigger control status clear register 63 (ADTCSC63)		A/D activation trigger control status clear register 62 (ADTCSC62)	
0x85A0	Range comparison status register 35 (ADRCS35)	Range comparison status register 34 (ADRCS34)	Range comparison status register 33 (ADRCS33)	Range comparison status register 32 (ADRCS32)
0x85A4	Range comparison status register 39 (ADRCS39)	Range comparison status register 38 (ADRCS38)	Range comparison status register 37 (ADRCS37)	Range comparison status register 36 (ADRCS36)
0x85A8	Range comparison status register 43 (ADRCS43)	Range comparison status register 42 (ADRCS42)	Range comparison status register 41 (ADRCS41)	Range comparison status register 40 (ADRCS40)
0x85AC	Range comparison status register 47 (ADRCS47)	Range comparison status register 46 (ADRCS46)	Range comparison status register 45 (ADRCS45)	Range comparison status register 44 (ADRCS44)
0x85B0	Range comparison status register 51 (ADRCS51)	Range comparison status register 50 (ADRCS50)	Range comparison status register 49 (ADRCS49)	Range comparison status register 48 (ADRCS48)
0x85B4	Range comparison status register 55 (ADRCS55)	Range comparison status register 54 (ADRCS54)	Range comparison status register 53 (ADRCS53)	Range comparison status register 52 (ADRCS52)
0x85B8	Range comparison status register 59 (ADRCS59)	Range comparison status register 58 (ADRCS58)	Range comparison status register 57 (ADRCS57)	Range comparison status register 56 (ADRCS56)
0x85BC	Range comparison status register 63 (ADRCS63)	Range comparison status register 62 (ADRCS62)	Range comparison status register 61 (ADRCS61)	Range comparison status register 60 (ADRCS60)
0x85C0	Range comparison flag clear register 1 (ADRCIFC1)			
0x85C4	Reserved	Reserved	Reserved	Scan conversion control status clear register 1 (ADSCANSC1)

■ Registers of the 12-bit A/D converter control

Table 4-3 Registers of the 12-Bit A/D Converter Control

Address	+3	+2	+1	+0
0x81C8	A/D mode setting register 0 (ADMD0)	A/D channel status register 0 (ADCH0)	A/D control status register 0 (ADCS0)	
0x81CC	Sampling time setting per A/D channel register 3 (ADSTPCS3)	Sampling time setting per A/D channel register 2 (ADSTPCS2)	Sampling time setting per A/D channel register 1 (ADSTPCS1)	Sampling time setting per A/D channel register 0 (ADSTPCS0)
0x81D0	Sampling time setting per A/D channel register 7 (ADSTPCS7)	Sampling time setting per A/D channel register 6 (ADSTPCS6)	Sampling time setting per A/D channel register 5 (ADSTPCS5)	Sampling time setting per A/D channel register 4 (ADSTPCS4)
0x85C8	A/D mode setting register 1 (ADMD1)	A/D channel status register 1 (ADCH1)	A/D control status register 1 (ADCS1)	
0x85CC	Sampling time setting per A/D channel register 11 (ADSTPCS11)	Sampling time setting per A/D channel register 10 (ADSTPCS10)	Sampling time setting per A/D channel register 9 (ADSTPCS9)	Sampling time setting per A/D channel register 8 (ADSTPCS8)
0x85D0	Sampling time setting per A/D channel register 15 (ADSTPCS15)	Sampling time setting per A/D channel register 14 (ADSTPCS14)	Sampling time setting per A/D channel register 13 (ADSTPCS13)	Sampling time setting per A/D channel register 12 (ADSTPCS12)

4.1. Registers of the Analog Input Control

This section explains the registers of the analog input control.

The analog input control has the key code register, analog input control registers, and failure diagnosis function control registers.

4.1.1. Key Code Register: KEYCDR

This section shows the bit configuration of the key code register.

This register makes the write setting of a register that has the function for protection against erroneous writing (ADER). If writing to this register is not done using the prescribed method, writing to the relevant register is invalid. This register is valid only for word access.

■ KEYCDR: address 8800_H (access: word)

bit	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
Field	KEY1	KEY0	SIZE1	SIZE0	Reserved			
R/W Attribute	R0,W	R0,W	R0,W	R0,W	R0,WX	R0,WX	R0,WX	R0,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

bit	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
Field	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	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
Field	Reserved				ADR11	ADR10	ADR09	ADR08
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,W	R0,W	R0,W	R0,W
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

bit	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Field	ADR07	ADR06	ADR05	ADR04	ADR03	ADR02	ADR01	ADR00
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 to bit30] KEY1 to KEY0: Key Code Setting Bits

You need to 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, you need to set the key codes again from the beginning.

Bit31:30		Description
0	0	1st key code
0	1	2nd key code
1	0	3rd key code
1	1	4th key code

[bit29 to bit28] SIZE[1:0]: Access Size Setting Bits

These bits set the access size for writing to a key code target register.

Write the same data to these bits when writing the key codes of "00", "01", "10", and "11" to KEY[1:0] in this order.

Bit29:28		Description
0	0	Set byte access.
0	1	Set half-word access.
1	0	Set word access.
1	1	Reserved

[bit27 to bit12] Reserved

Always write "0" to these bits.

[bit11 to bit0] ADR11 to ADR0: Address Specification Bits

These bits set the lower 12 bits of the address of a key code target register. Write the same data to these bits when writing the key codes of "00", "01", "10", and "11" to KEY[1:0] in this order.

bit11:0	Description
Set the lower 12 bits of the address of a key code target register.	

Note:

The following are the notes on key code setting.

- *You need to write "00", "01", "10", and "11" continuously to KEY[1:0] in this order in 4 write operations.*
  The key code setting becomes invalid immediately upon any writing to these bits in a different order. In such cases, you need to set the key codes again from the beginning.
- *You need to write the same pair of values to SIZE[1:0] and ADR[11:0] 4 times when writing data to KEY[1:0] 4 times.*
  The key code setting becomes invalid upon any writing of a different pair of values. In such cases, you need to set the key codes again from the beginning.
- *When the KEYCDR register is read during writing to KEY[1:0], the key code setting becomes invalid. In such cases, you need to set the key codes again from the beginning.*
- *When access to the address set in ADR[11:0] occurs during writing to KEY[1:0], the key code setting becomes invalid. In such cases, you need to set the key codes again from the beginning.*
  When access to any register other than the above occurs, the key code setting does not become invalid.
  Writing the remaining data to KEY[1:0] subsequently makes the key code setting valid.
- *Once the address set in KEYCDR:ADR[11:0] is accessed after KEY[1:0] is set normally, it is required to set the key codes again to access the address again.*

4.1.2. Analog Input Control Register: ADER

This section shows the bit configurations of the analog input control registers.

The analog input control registers (ADER0, ADER1) control analog input.

■ ADER0: Address 8804_H (access: byte, half word, word)

bit	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
Field	ADE31	ADE30	ADE29	ADE28	ADE27	ADE26	ADE25	ADE24
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	WS							
Initial Value	1	1	1	1	1	1	1	1

bit	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
Field	ADE23	ADE22	ADE21	ADE20	ADE19	ADE18	ADE17	ADE16
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	WS							
Initial Value	1	1	1	1	1	1	1	1

bit	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
Field	ADE15	ADE14	ADE13	ADE12	ADE11	ADE10	ADE09	ADE08
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	WS							
Initial Value	1	1	1	1	1	1	1	1

bit	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Field	ADE07	ADE06	ADE05	ADE04	ADE03	ADE02	ADE01	ADE00
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	WS							
Initial Value	1	1	1	1	1	1	1	1

[bit31 to bit0] ADE31 to ADE00: Analog Input Enable Bits

ADE31 to ADE00	Function
0	Port input/output mode
1	Analog input mode

- These bits control analog input pins.
- If these bits are "0", the analog input is disabled.
- If these bits are "1", the analog input is enabled.

Note:

- *This register is a key code target register. Key code settings are required for writing to these registers.*
  For the setting method, see "4.1.1. Key Code Register: KEYCDR."

■ ADER1: Address 8808_H (access: byte, half word, word)

bit	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
Field	ADE63	ADE62	ADE61	ADE60	ADE59	ADE58	ADE57	ADE56
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	WS							
Initial Value	1	1	1	1	1	1	1	1

bit	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
Field	ADE55	ADE54	ADE53	ADE52	ADE51	ADE50	ADE49	ADE48
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	WS							
Initial Value	1	1	1	1	1	1	1	1

bit	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
Field	ADE47	ADE46	ADE45	ADE44	ADE43	ADE42	ADE41	ADE40
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	WS							
Initial Value	1	1	1	1	1	1	1	1

bit	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Field	ADE39	ADE38	ADE37	ADE36	ADE35	ADE34	ADE33	ADE32
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	WS							
Initial Value	1	1	1	1	1	1	1	1

[bit31 to bit0] ADE63 to ADE32: Analog Input Enable Bits

ADE63 to ADE32	Function
0	Port input/output mode
1	Analog input mode

- These bits control analog input pins.
- If these bits are "0", the analog input is disabled.
- If these bits are "1", the analog input is enabled.

Note:

- *This register is a key code target register. Key code settings are required for writing to these registers.*
  For the setting method, see "4.1.1. Key Code Register: KEYCDR."

4.1.3. Failure Diagnosis Function Control Register: ADFDCTR0, ADFDCTR1

This section shows the bit configurations of the failure diagnosis function control registers.

The failure diagnosis function control registers control the failure diagnosis function of the A/D converter.

■ ADFDCTR0: Address 81D4$_H$ (access: byte, half word, word)

bit	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
Field	Reserved							
R/W Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX
Protection Attribute	-							
Initial Value	1	1	1	1	1	1	1	1

bit	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
Field	Reserved							
R/W Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX
Protection Attribute	-							
Initial Value	1	1	1	1	1	1	1	1

bit	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
Field	Reserved							
R/W Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX
Protection Attribute	-							
Initial Value	1	1	1	1	1	1	1	1

bit	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Field	Reserved						AVRH0SEL	AVRL0SEL
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W	R/W
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

[bit31 to bit2] Reserved

Always write "0" to these bits.

[bit1] AVRH0SEL: AVRH0 Voltage Input Setting Bit
[bit0] AVRL0SEL: AVRL0 Voltage Input Setting Bit

AVRH0SEL	AVRL0SEL	Function
0	0	Set neither the AVRH0 level voltage nor AVRL0 level voltage as analog input (ordinary A/D conversion).
0	1	Set the AVRL0 level voltage as analog input.
1	0	Set the AVRH0 level voltage as analog input.
1	1	This setting is prohibited (operation is not guaranteed).

- These bits are used to make the setting regarding 12-bit A/D converter unit 0.
- When either bit is set to "1", the analog input from the A/D converter analog input pin ANx (x: integer number) becomes invalid.

Notes:

- *After confirming that the A/D conversion has stopped, perform switching of AVRH0SEL and AVRL0SEL. After failure diagnosis, make the setting of AVRH0SEL=0 and AVRL0SEL=0.*
- *Setting of AVRH0SEL=1 and AVRL0SEL=1 is prohibited.*
  If this setting is made, the operation will be the same as when the setting of AVRH0SEL=0 and AVRL0SEL=0 is made. However, during the short time period in which the circuit switching begins and completes, through-current may occur between AVRH0 and AVRL0.

■ ADFDCTR1: Address 85D4$_H$ (access: byte, half word, word)

bit	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
Field	Reserved							
R/W Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX
Protection Attribute	-							
Initial Value	1	1	1	1	1	1	1	1

bit	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
Field	Reserved							
R/W Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX
Protection Attribute	-							
Initial Value	1	1	1	1	1	1	1	1

bit	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
Field	Reserved							
R/W Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX
Protection Attribute	-							
Initial Value	1	1	1	1	1	1	1	1

bit	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Field	Reserved						AVRH1SEL	AVRL1SEL
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W	R/W
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

[bit31 to bit2] Reserved

Always write "0" to these bits.

[bit1] AVRH1SEL: AVRH1 Voltage Input Setting Bit
[bit0] AVRL1SEL: AVRL1 Voltage Input Setting Bit

AVRH1SEL	AVRL1SEL	Function
0	0	Set neither the AVRH1 level voltage nor AVRL1 level voltage as analog input (ordinary A/D conversion).
0	1	Set the AVRL1 level voltage as analog input.
1	0	Set the AVRH1 level voltage as analog input.
1	1	This setting is prohibited (operation is not guaranteed).

- These bits are used to make the setting regarding 12-bit A/D converter unit 1.
- When either bit is set to "1", the analog input from the A/D converter analog input pin ANx (x: integer number) becomes invalid.

Notes:

- *After confirming that the A/D conversion has stopped, perform switching of AVRH1SEL and AVRL1SEL. After failure diagnosis, make the setting of AVRH1SEL=0 and AVRL1SEL=0.*
- *Setting of AVRH1SEL=1 and AVRL1SEL=1 is prohibited.*
  If this setting is made, the operation will be the same as when the setting of AVRH1SEL=0 and AVRL1SEL=0 is made. However, during the short time period in which the circuit switching begins and completes, through-current may occur between AVRH1 and AVRL1.

4.2. A/D Activation Compare Register

This section describes the A/D activation compare register.

The A/D activation compare has the following registers.

- A/D software activation register
- A/D software activation channel selection register
- Compare buffer register/compare register
- A/D activation trigger control status register
- A/D data register
- S/D activation trigger extended control register
- Upper threshold setting register
- Lower threshold setting register
- Range comparison control status register
- Range comparison threshold excess flag register
- Range comparison flag register
- Scan conversion control status register
- Activation channel conversion count setting register
- Data protection status flag register
- Activation channel conversion completion flag register
- A/D activation trigger control status clear register
- Range comparison status register
- Range comparison flag clear register
- Scan conversion control status clear register

4.2.1. A/D Software Activation Register: ADTSS0, ADTSS1

This section shows the bit configuration of the A/D software activation registers.

The A/D software activation register (ADTSS) is set to generate the A/D conversion start request. The activation channel is specified by the A/D software activation channel selection register (ADTSE).

- ADTSS0: Address 8004_H (access: byte, half word, word)
- ADTSS1: Address 8404_H (access: byte, half word, word)

bit	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Field	Reserved							START
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

[bit7 to bit1] Reserved

Always write "0" to these bits.

[bit0] START: A/D Conversion Activation Bit (Software)

START	Function
0	Do not activate the A/D conversion function.
1	Activate the A/D conversion function.

- This bit is used to start A/D conversion by the software.
- When "1" is written in this bit, A/D conversion is activated. The activation channel is specified by the A/D software activation channel selection register (ADTSE).
- This bit cannot be used as reconversion of A/D converter.

4.2.2. A/D Software Activation Channel Selection Register: ADTSE0, ADTSE1

This section shows the bit configuration of the A/D software activation channel selection registers.

The A/D software activation channel selection register (ADTSE) selects the activation channel to issue the A/D activation request.

■ ADTSE0: Address 8008_H (access: byte, half word, word)

bit	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
Field	ADT31	ADT30	ADT29	ADT28	ADT27	ADT26	ADT25	ADT24
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	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
Field	ADT23	ADT22	ADT21	ADT20	ADT19	ADT18	ADT17	ADT16
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	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
Field	ADT15	ADT14	ADT13	ADT12	ADT11	ADT10	ADT09	ADT08
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	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Field	ADT07	ADT06	ADT05	ADT04	ADT03	ADT02	ADT01	ADT00
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 to bit0] ADT31 to ADT00: Software Activation Channel Selection Bits

ADT31 to ADT00	Function
0	Disable software activation.
1	Enable software activation.

- These bits control the software activation of the activation channel.
- When these bits are "0", software activation is disabled.
- When these bits are "1", software activation is enabled.

■ ADTSE1: Address 8408_H (access: byte, half word, word)

bit	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
Field	ADT63	ADT62	ADT61	ADT60	ADT59	ADT58	ADT57	ADT56
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	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
Field	ADT55	ADT54	ADT53	ADT52	ADT51	ADT50	ADT49	ADT48
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	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
Field	ADT47	ADT46	ADT45	ADT44	ADT43	ADT42	ADT41	ADT40
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	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Field	ADT39	ADT38	ADT37	ADT36	ADT35	ADT34	ADT33	ADT32
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 to bit0] ADT63 to ADT32: Software Activation Channel Selection Bits

ADT63 to ADT32	Function
0	Disable software activation.
1	Enable software activation.

- These bits control the software activation of the activation channel.
- When these bits are "0", software activation is disabled.
- When these bits are "1", software activation is enabled.

4.2.3. Compare Buffer Register: ADCOMPB0 to ADCOMPB63/Compare Register: ADCOMP0 to ADCOMP63

This section shows the bit configurations of the compare buffer registers/compare registers.

The compare buffer register (ADCOMPB) is a 16-bit buffer register for the compare registers (ADCOMP). The compare register (ADCOMP) activates the A/D converter when it matches the value of the free-run timer.

Both ADCOMPB and ADCOMP registers exist in the same address.

- ADCOMPB0 to ADCOMPB31: Address $800C_H$ to $804A_H$ (access: half word, word)
- ADCOMPB32 to ADCOMPB63: Address $840C_H$ to $844A_H$ (access: half word, word)

bit	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
Field	CMP15	CMP14	CMP13	CMP12	CMP11	CMP10	CMP09	CMP08
R/W Attribute	W	W	W	W	W	W	W	W
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

bit	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Field	CMP07	CMP06	CMP05	CMP04	CMP03	CMP02	CMP01	CMP00
R/W Attribute	W	W	W	W	W	W	W	W
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

[bit15 to bit0] CMP15 to CMP00: Compare Value Buffer Bits

CMP15 to CMP00	Function
	Compare value buffer

- The compare buffer register is a buffer register for the A/D activation compare register (ADCOMP). When the buffer function is disabled (BUFX=1 for the A/D activation trigger control status register (ADTCS)) or the free-run timer is stopped, the compare buffer value is immediately transferred to the compare register.
- When the buffer function is enabled (BUFX=0 for the A/D activation trigger control status register (ADTCS)), if the compare buffer value matches the 32-bit free-run timer compare clear register or "0" is detected, it is transferred to the compare register.

Note:

- *For access to these registers, use half word or word access instructions.*

- ADCOMP0 to ADCOMP31: Address $800C_H$ to $804A_H$ (access: half word, word)
- ADCOMP32 to ADCOMP63: Address $840C_H$ to $844A_H$ (access: half word, word)

bit	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
Field	CMP15	CMP14	CMP13	CMP12	CMP11	CMP10	CMP09	CMP08
R/W Attribute	R	R	R	R	R	R	R	R
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

bit	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Field	CMP07	CMP06	CMP05	CMP04	CMP03	CMP02	CMP01	CMP00
R/W Attribute	R	R	R	R	R	R	R	R
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

[bit15 to bit0] CMP15 to CMP00: Compare Value Bits

CMP15 to CMP00	Function
	Compare value

- The compare value is updated via the compare buffer register.
- When the free-run timer and compare value matches in a register that stores the compare value to compare with the lower 16-bit count value of a 32-bit free-run timer, the compare register can activate A/D.
- The compare value that is stored in the compare register is immediately compared.
- When the A/D activation trigger control status register (ADTCS) is SEL1, SEL0=11_B, the compare match operation is not performed.

Note:

- *Use word or half word for reading to the compare register.*

4.2.4. A/D Activation Trigger Control Status Register: ADTCS0 to ADTCS63

This section shows the bit configuration of the A/D activation trigger control status registers.

The A/D activation trigger control status register (ADTCS) is used for A/D activation request confirmation, permission/prohibition of interrupt request, confirmation of interrupt request status, selection of activation factor, selection of conversion mode, control of protection function, selection of the compare value used for compare operation, buffer control of the compare value and selection of an analog input channel.

- ADTCS0 to ADTCS31: Address 804C$_H$ to 808A$_H$ (access: byte, half word, word)
- ADTCS32 to ADTCS63: Address 844C$_H$ to 848A$_H$ (access: byte, half word, word)

bit	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
Field	BUSY	INT	INTE	STS1	STS0	RPT	PRT	PRTS
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

bit	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Field	SEL1	SEL0	BUFX	BTS	Reserved			
R/W Attribute	R/W	R/W	R/W	R/W	R0/W0	R0/W0	R0/W0	R0/W0
Protection Attribute	-							
Initial Value	0	0	1	0	0	0	0	0

[bit15] BUSY: A/D Activation Request in Progress Bit

BUSY	Function
0	A/D activation has not been requested
1	A/D activation has been requested or conversion is in progress.

- This bit is a bit to indicate A/D activation request or conversion operation.
- During reading, when this bit is "0", it shows that A/D conversion has not been requested. When the bit is "1", it indicates that A/D conversion has been requested or conversion is in progress.
- The BUSY bit is read-only. Writing to the BUSY bit has no effect on operation.
- The BUSY bit is cleared to "0" by writing "1" to corresponding BUSY clear bit (ADTCSC[n].BUSYC).

[bit14] INT: Interrupt Request Flag Bit

INT	Function
0	A/D conversion has not been completed.
1	A/D conversion has been completed.

- When data is set in the A/D data register (ADTCD) by A/D conversion, this bit is set to "1".
- When this bit and the interrupt request enable bit (ADTCS:INTE) are "1", the interrupt request is generated.
- The INT bit is read-only. Writing to the INT bit has no effect on operation.
- The INT bit is cleared to "0" by writing "1" to the corresponding INT clear bit (ADTCSC[n].INTC).
- This bit is cleared when the A/D conversion end interrupt clear signal is "H".

Note:

- *When the software clear (writing ADTCSC[n].INTC="1") or clear by the interrupt clear signal ("H") occurs at the same time as the hardware set, the hardware set takes precedence.*

[bit13] INTE: Interrupt Request Enable Bit

INTE	Function
0	Disable interrupt request output.
1	Enable interrupt request output.

- This bit enables/disables interrupt output to the CPU.
- When this bit and the interrupt request flag bit (ADTCS:INT) is "1", interrupt request is generated.

[bit12, bit11] STS1, STS0: A/D Activation Factor Selection Bits

ADTECSn: STS2	STS1	STS0	Function
0	0	0	Software activation
0	0	1	Setting prohibited
0	1	0	Reload timer activation (rising edge) Partial wake-up activation (rising edge)
0	1	1	Compare match activation
1	0	0	PWM timer activation (rising edge)
1	0	1	Setting prohibited
1	1	0	
1	1	1	

- STS1 and STS0 bits, being combined with bit 8 (STS2) in the A/D activation trigger extended control register (ADTECS), select the activation factor of the A/D conversion.
- To move to the partial wake-up mode, set these bits to "010".
- For the A/D converter unit number and channel number which correspond to the partial wake-up mode, see "CHAPTER: PARTIAL WAKEUP CONTROL."
- The reload timer and PWM timer can be selected as one of the functions of the base timer.
- The correspondence between the reload timer and 12-bit A/D converter units is as follows.
  - Reload timer channel 0 corresponds to A/D converter unit 0.
  - Reload timer channel 2 corresponds to A/D converter unit 1.

- About PWM mode, please refer to Chapter: 12-Bit A/D Converter 5.2.8. PWM Timer Activation.

Notes:

- *Since the A/D activation factor selection bit is changed while writing is performed, change these bits when the activation factors of the current selection destination and the selection destination to be changed are not active and A/D conversion request is not in progress (ADTCS:BUSY=1).*
- *When the A/D activation request is not made, set these bits including ADTECS.STS2 as software activation ("000_B") and the corresponding bit of the ADTSE (activation channel) as software activation disabled (ADT bit=0).*
- *When the A/D activation factor selection bit is set, make sure that the 32-bit free-run timer is stopped.*
- *When these bits including ADTECS.STS2 are set to the reload timer activation ("010B"), even if the corresponding base timer is used as the PWM timer or PPG timer, the rising edge of the timer output is regarded as the A/D activation request and A/D conversion is started.*

[bit10] RPT: Repeat Conversion Selection Bit

RPT	Function
0	Single conversion
1	Repeat conversion

- This bit sets the A/D conversion mode.
- When it is set to "0", the single conversion mode is set. 1 A/D conversion request is made per activation factor. A/D conversion is conducted once.
- When this bit is set to "1", the repeat conversion mode is set. 1 activation factor initiates the successive A/D conversion request. A/D conversion is repeatedly performed until the mode is changed to the single conversion mode.

[bit9] PRT: A/D Data Register Protection Enable Bit

PRT	Function
0	Protection disabled
1	Protection enabled

- When this bit is set to "1", this bit protects overwriting data to the A/D data register. Moreover, the protection function is effective for factors that are not compare match activation (STS1, STS0=11).
- After the conversion data is stored in the A/D data register, until the factor set in A/D data register protection release selection bit (PRTS) is generated, it masks the next activation factor and protects overwriting data to the A/D data register.

Note:

- *Set the A/D data register protection enable bit before performing A/D conversion. Do not change the A/D data register protection enable bit while A/D conversion is being requested or the A/D data register is protected.*

[bit8] PRTS: A/D Data Register Protection Release Selection Bit

PRTS	Function
0	Data reading and interrupt flag clear
1	Data reading

- When the A/D data register protection function is enabled (PRT=1), this bit selects the activation request mask release condition.
- When this bit is set to "0", the A/D data register (ADTCD) reading and interrupt request flag bit (INT) clear become protection release conditions (in random order).
- When this bit is set to "1", the A/D data register (ADTCD) reading becomes a protection release condition.

Note:

- *The A/D data register protection release selection bit should be set before performing A/D conversion.*
  Do not change the A/D data register protection release selection bit while A/D conversion is being requested or the A/D data register is protected.

[bit7, bit6] SEL1, SEL0: Count Direction Selection Bits

SEL1	SEL0	Function
0	0	Both up/down count
0	1	Only up count
1	0	Only down count
1	1	Compare prohibited

- When these bits are set to "00_B", the compare match is performed during both up/down count in free-run timer.
- When these bits are set to "01_B", the compare match is performed during only up count in free-run timer.
- When these bits are set to "10_B", the compare match is performed during only down count in free-run timer.
- When these bits are set to "11_B", the compare match is not performed.
- When the selected free-run timer is suspended, the compare match is not performed.

Note:

- *When the 32-bit free-run timer is in up count mode, do not set these bits to "10".*

[bit5] BUFX: Compare Register Buffer Function Control Bit

BUFX	Function
0	Enabled
1	Disabled

- When this bit is set to "1", the buffer function is disabled.
- When this bit is set to "0", the buffer function is enabled.

[bit4] BTS: Compare Register Buffer Transfer Control Bit

BTS	Function
0	When "0" is detected
1	At compare clear

- This bit sets transfer conditions when the compare register buffer function is enabled (ADTCS.BUFX="0").
- When the compare register buffer function is enabled (ADTCS.BUFX="0"), the BTS setting is enabled.
- When this bit is set to "0", if the free-run timer is detected as "0", compare buffer register value is transferred to the compare register.
- When this bit is set to "1", if the free-run timer is detected as "compare clear", compare buffer register value is transferred to the compare register.

Note:

- *To set the compare register buffer transfer control bit, make sure that the 32-bit free-run timer is suspended.*

[bit3 to bit0] Reserved

Always write "0" to these bits.

4.2.5. A/D Data Register: ADTCD0 to ADTCD63

This section shows the bit configuration of the A/D data registers.

The A/D data register (ADTCD) stores the A/D conversion results.

- ADTCD0 to ADTCD31: Address 808C$_H$ to 80CA$_H$ (access: byte, half word, word)
- ADTCD32 to ADTCD63: Address 848C$_H$ to 84CA$_H$ (access: byte, half word, word)

bit	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
Field	ERR	ERRST	Reserved		D11	D10	D9	D8
R/W Attribute	R, WX	R,WX	R0,W0	R0,W0	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	1	0	0	0	0	0	0	0

bit	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Field	D7	D6	D5	D4	D3	D2	D1	D0
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] ERR: Conversion Data Error Flag Bit

ERR	Function
0	The conversion data is normal.
1	The conversion data is not normal.

- This bit is a bit to indicate that there is an error in A/D conversion data. When this bit is "1", the error information can be obtained from the ERRST bit value.
- This bit is set to "1" after reading.
- When new conversion results are written in this register, it is cleared to "0".
- When the A/D data register protection function is enabled (ADTCS.PRT=1) and factors are other than compare match activation (ADTECS STS2="0", ADTCS.STS1, STS0=11), "0" is read.

[bit14] ERRST: Conversion Data Error Status Bit (Only if ERR=1)

ERRST	Function
0	The conversion data are old results.
1	The conversion data was overwritten with new data.

- This bit is a flag to show error information of A/D conversion data when the ERR bit=1.
- When the ERR bit=1 and this bit=0, it shows that the conversion result of the CPU reading is old.
- When the ERR bit=1 and this bit=1, it shows the status of the conversion results of the CPU reading: the reading of the old conversion results using the CPU did not finish, and the old conversion data was lost due to the overwrite of the new conversion results.
- When the reading of the old conversion results using the CPU did not finish and the old conversion data was lost due to the overwrite of the new conversion results, this bit is set to "1".
- After reading, this bit is cleared to "0".
- When the A/D data register protection function is enabled (ADTCS.PRT=1) and factors are other than compare match activation (ADTECS.STS2="0", ADTCS.STS1, STS0=11), "0" is read.

[bit13, bit12] Reserved

Always write "0" to these bits.

[bit11to bit0] D11 to D0: A/D Data Bits

D11 to D0	Function
	Conversion data

- The A/D conversion results are stored and a register is rewritten for the completion of each conversion process.
- Under normal conditions, the last conversion value is stored.

Note:

- *Do not write in these registers.*

4.2.6. A/D Activation Trigger Extended Control Register: ADTECS0 to ADTECS63

This section shows the bit configurations of the A/D activation trigger extended control registers.

The A/D activation trigger extended control register (ADTECS) selects the activation factor and analog input channel.

ADTECS0 to 31: Address $80CC_H$ to $810A_H$ (Access: Byte, Half Word, Word)

bit	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
Field	Reserved							STS2
R/W Attribute	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R/W
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

bit	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Field	Reserved	Reserved	Reserved	CHSEL4	CHSEL3	CHSEL2	CHSEL1	CHSEL0
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

[bit15 to bit9] Reserved

Always write "0" to these bits.

[bit8] STS2: A/D Activation Factor Selection Bits

This bit selects the A/D conversion activation factor using the combination of bit12 and bit11 (STS1, STS0) of the A/D activation trigger control status register (ADTCS). For details, see "4.2.4 A/D Activation Trigger Control Status Register: ADTCS0 to ADTCS63."

Notes:

- *Since the A/D activation factor selection bit is changed while writing is performed, change this bit when the activation factors of the current selection destination and the selection destination to be changed are not active and A/D conversion request is not in progress (ADTCS:BUSY=1).*
- *When the A/D activation request is not made, set this bit including ADTCS.STS1 and ADTCS.STS0 as software activation ("000_B"), and the corresponding bit of the ADTSE (activation channel) as software activation disabled (ADT bit=0).*
- *When the A/D activation factor selection bit is set, make sure that the 32-bit free-run timer is stopped.*

[bit7 to bit5] Reserved

Always write "0" to these bits.

[bit4 to bit0] CHSEL4 to CHSEL0: Analog Channel Selection Bits

CHSEL4	CHSEL3	CHSEL2	CHSEL1	CHSEL0	Function
0	0	0	0	0	Channel 0
0	0	0	0	1	Channel 1
		:			:
0	0	1	1	0	Channel 6
0	0	1	1	1	Channel 7
0	1	0	0	0	Channel 8
		:			:
0	1	1	1	0	Channel 14
0	1	1	1	1	Channel 15
1	0	0	0	0	Channel 16
		:			:
1	1	1	1	0	Channel 30
1	1	1	1	1	Channel 31

Select an analog channel for the selected value.

Do not change the analog channel selection bit during A/D conversion.

■ ADTECS32 to 63: Address $84CC_H$ to $850A_H$ (access: byte, half word, word)

bit	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
Field	Reserved							STS2
R/W Attribute	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R/W
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

bit	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Field	Reserved	Reserved	Reserved	CHSEL4	CHSEL3	CHSEL2	CHSEL1	CHSEL0
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

[bit15 to bit9] Reserved

Always write "0" to these bits.

[bit8] STS2: A/D Activation Factor Selection Bit

This bit selects the A/D conversion activation factor using the combination of bit12 and bit11 (STS1, STS0) of the A/D activation trigger control status register (ADTCS). For details, see "4.2.4 A/D Activation Trigger Control Status Register: ADTCS0 to ADTCS63."

Notes:

- *Since the A/D activation factor selection bit is changed while writing is performed, change this bit when the activation factors of the current selection destination and the selection destination to be changed are not active and A/D conversion request is not in progress (ADTCS:BUSY=1).*
- *When the A/D activation request is not made, set this bit including ADTCS.STS1 and ADTCS.STS0 as software activation ("000_B"), and the corresponding bit of the ADTSE (activation channel) as software activation disabled (ADT bit=0).*
- *When the A/D activation factor selection bit is set, make sure that the 32-bit free-run timer is stopped.*

[bit7 to bit5] Reserved

Always write "0" to these bits.

[bit4 to bit0] CHSEL4 to CHSEL0: Analog Channel Selection Bits

CHSEL4	CHSEL3	CHSEL2	CHSEL1	CHSEL0	Function
0	0	0	0	0	Channel 32
0	0	0	0	1	Channel 33
:					:
0	0	1	1	0	Channel 38
0	0	1	1	1	Channel 39
0	1	0	0	0	Channel 40
:					:
0	1	1	1	0	Channel 46
0	1	1	1	1	Channel 47
1	0	0	0	0	Channel 48
:					:
1	1	1	1	0	Channel 62
1	1	1	1	1	Channel 63

4.2.7. Upper Threshold Setting Register: ADRCUT0 to ADRCUT7

This section shows the bit configuration of the upper threshold setting registers.

The upper threshold setting register (ADRCUT) sets the upper threshold to be used to compare ranges. It is possible to set 4 types of upper thresholds.

- ADRCUT0: Address 810C$_H$ (access: byte, half word, word)
- ADRCUT1: Address 8110$_H$ (access: byte, half word, word)
- ADRCUT2: Address 8114$_H$ (access: byte, half word, word)
- ADRCUT3: Address 8118$_H$ (access: byte, half word, word)
- ADRCUT4: Address 850C$_H$ (access: byte, half word, word)
- ADRCUT5: Address 8510$_H$ (access: byte, half word, word)
- ADRCUT6: Address 8514$_H$ (access: byte, half word, word)
- ADRCUT7: Address 8518$_H$ (access: byte, half word, word)

bit	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
Field	Reserved				C11	C10	C9	C8
R/W Attribute	R0,W0	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

bit	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Field	C7	C6	C5	C4	C3	C2	C1	C0
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 to bit12] Reserved

Always write "0" to these bits.

[bit11 to bit0] C11 to C0: Upper Threshold Bits

C11 to C0	Function
	Upper threshold

- These bits set the upper threshold to be used to compare ranges.

Note:

- *Do not change the upper threshold bit while A/D conversion is being requested.*

4.2.8. Lower Threshold Setting Register: ADRCLT0 to ADRCLT7

This section shows the bit configuration of the lower threshold setting registers.

The lower threshold setting register (ADRCLT) sets the lower threshold to be used to compare ranges. It is possible to set 4 types of lower thresholds.

- ADRCLT0: Address $810E_H$ (access: byte, half word, word)
- ADRCLT1: Address 8112_H (access: byte, half word, word)
- ADRCLT2: Address 8116_H (access: byte, half word, word)
- ADRCLT3: Address $811A_H$ (access: byte, half word, word)
- ADRCLT4: Address $850E_H$ (access: byte, half word, word)
- ADRCLT5: Address 8512_H (access: byte, half word, word)
- ADRCLT6: Address 8516_H (access: byte, half word, word)
- ADRCLT7: Address $851A_H$ (access: byte, half word, word)

bit	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
Field	Reserved				C11	C10	C9	C8
R/W Attribute	R0,W0	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

bit	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Field	C7	C6	C5	C4	C3	C2	C1	C0
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 to bit12] Reserved

Always write "0" to these bits.

[bit11 to bit0] C11 to C0: Lower Threshold Bits

C11 to C0	Function
	Lower threshold

- These bits set the lower threshold to be used to compare ranges.

Note:

- *Do not change the lower threshold bit while A/D conversion is being requested.*

4.2.9. Range Comparison Control Status Register: ADRCCS0 to ADRCCS63

This section shows the bit configuration of the range comparison control status registers.

The range comparison control status register (ADRCCS) performs continuous detection count instructions, inside/outside-range check selection, range comparison interrupt request permission/prohibition, range comparison execution permission/prohibition and upper and lower threshold selection.

- ADRCCS0 to ADRCCS31: Address $811C_H$ to $813B_H$ (access: byte, half word, word)
- ADRCCS32 to ADRCCS63: Address $851C_H$ to $853B_H$ (access: byte, half word, word)

bit	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Field	RCOCD2	RCOCD1	RCOCD0	RCOIRS	RCOIE	RCOE	RCOTS1	RCOTS0
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

[bit7 to bit5] RCOCD2 to RCOCD0: Continuous Detection Count Specification Bits

RCOCD2 to RCOCD0	Function
"000B"	Setting prohibited
"001B"	Set the continuous detection count to 1
"010B"	Set the continuous detection count to 2
"011B"	Set the continuous detection count to 3
"100B"	Set the continuous detection count to 4
"101B"	Set the continuous detection count to 5
"110B"	Set the continuous detection count to 6
"111B"	Set the continuous detection count to 7

- These bits specify the continuous detection count of range comparison results.
- When the detection count of range comparison results reaches the specified continuous detection count, the range comparison interrupt factor flag bit of the corresponding activation channel is set to "1".

Note:

- *Do not set the continuous detection count specification/status display bit to "000_B".*
- *Do not change the continuous detection count specification/status display bit while A/D conversion is being requested.*
- *Do not change the continuous detection count specification bit while the execution of range comparison is enabled (ADRCCS.RCOE="1").*

[bit4] RCOIRS: Inside/Outside-Range Check Selection Bit

RCOIRS	Description
0	Confirming that the data is outside range
1	Confirming that the data is inside range

- The A/D data bit selects range comparison conditions to define if data is inside range or outside range for upper and lower threshold bits which were selected using the upper and lower threshold selection bits.
- The range comparison conditions to check outside range (RCOIRS="0") are as follows.
  A/D data bit (ADTCD.D11 to D0) > Upper threshold bit (ADRCUT.C11 to C0)
  or
  A/D data bit (ADTCD.D11 to D0) < Lower threshold bit (ADRCLT.C11 to C0)
- The range comparison conditions to check inside range (RCOIRS="1") are as follows.
  A/D data bit (ADTCD.D11 to D0) ≤ Upper threshold bit (ADRCUT.C11 to C0)
  and A/D data bit (ADTCD.D11 to D0) ≥ Lower threshold bit (ADRCLT.C11 to C0)
- Upon outside range check during range comparison detection, the threshold excess flag bit can be used to check whether the value is over the upper threshold or below the lower threshold.

Note:

- *Do not change the inside/outside-range check section bit while A/D conversion is being requested.*

[bit3] RCOIE: Range Comparison Interrupt Request Enable Bit

RCOIE	Description
0	Range comparison interrupt disabled
1	Range comparison interrupt enabled

When the range comparison interrupt factor flag bit of the corresponding activation channel is set to "1" and this bit is set as range comparison interrupt request enabled, interrupt request is generated.

[bit2] RCOE: Range Comparison Execution Enable Bit

RCOE	Description
0	Range comparison execution disabled
1	Range comparison execution enabled

- This bit enables/disables range comparison execution.
- When the range comparison execution enable bit is "0", the execution of range comparison is disabled. In addition, the continuous detection count status is initialized to "000_B".
- When the range comparison execution enable bit is "1", range comparison execution is enabled.

[bit1, bit0] RCOTS1, RCOTS0: Upper and Lower Threshold Selection Bits

- ADRCCS0 to ADRCCS31(A/D converter unit 0)

RCOTS1	RCOTS0	Description
0	0	Select upper threshold setting register 0/lower threshold setting register 0
0	1	Select upper threshold setting register 1/lower threshold setting register 1
1	0	Select upper threshold setting register 2/lower threshold setting register 2
1	1	Select upper threshold setting register 3/lower threshold setting register 3

- ADRCCS32 to ADRCCS63 (A/D converter unit 1)

RCOTS1	RCOTS0	Description
0	0	Select upper threshold setting register 4/lower threshold setting register 4
0	1	Select upper threshold setting register 5/lower threshold setting register 5
1	0	Select upper threshold setting register 6/lower threshold setting register 6
1	1	Select upper threshold setting register 7/lower threshold setting register 7

- These bits select 1 combination out of 4 types of upper threshold setting registers 0 to 3/4 to 7 and lower threshold setting register 0 to 3/4 to 7.

Note:

- *Do not change the upper and lower threshold bit while A/D conversion is being requested.*

4.2.10. Range Comparison Threshold Excess Flag Register: ADRCOT0, ADRCOT1

This section shows the bit configurations of the range comparison threshold excess flag registers.

The range comparison threshold excess flag register (ADRCOT) shows whether the value is over the upper threshold or below the lower threshold after comparing the range using the outside range check setting.

■ ADRCOT0: Address 813C$_H$ (access: byte, half word, word)

bit	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
Field	RCOOF31	RCOOF30	RCOOF29	RCOOF28	RCOOF27	RCOOF26	RCOOF25	RCOOF24
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	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
Field	RCOOF23	RCOOF22	RCOOF21	RCOOF20	RCOOF19	RCOOF18	RCOOF17	RCOOF16
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	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
Field	RCOOF15	RCOOF14	RCOOF13	RCOOF12	RCOOF11	RCOOF10	RCOOF9	RCOOF8
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	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Field	RCOOF7	RCOOF6	RCOOF5	RCOOF4	RCOOF3	RCOOF2	RCOOF1	RCOOF0
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] RCOOF31 to RCOOF0: Conversion Data Error Flag Bits

RCOOF[n]	Function
0	Below lower threshold (A/D data < lower threshold bit)
1	Over upper threshold (A/D data > upper threshold bit)

(n=0 to 31)

- Upon outside range check, these bits show that the range comparison result is higher than the upper threshold setting register or lower than the lower threshold setting register.
- Upon outside range check and when the range comparison result is inside the range, the value of the threshold excess flag bit remains the same.

- When the range comparison interrupt factor flag bit of the corresponding activation channel is set to "1", the threshold excess flag bit is not updated and the value remains the same even if outside range check detects that the range comparison result is outside range.
- Upon inside-range check, the threshold excess flag bit has no meaning. The value remains the same.

■ ADRCOT1: Address 853C$_H$ (access: byte, half word, word)

bit	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
Field	RCOOF63	RCOOF62	RCOOF61	RCOOF60	RCOOF59	RCOOF58	RCOOF57	RCOOF56
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	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
Field	RCOOF55	RCOOF54	RCOOF53	RCOOF52	RCOOF51	RCOOF50	RCOOF49	RCOOF48
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	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
Field	RCOOF47	RCOOF46	RCOOF45	RCOOF44	RCOOF43	RCOOF42	RCOOF41	RCOOF40
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	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Field	RCOOF39	RCOOF38	RCOOF37	RCOOF36	RCOOF35	RCOOF34	RCOOF33	RCOOF32
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] RCOOF63 to RCOOF32: Conversion Data Error Flag Bits

RCOOF[n]	Function
0	Below lower threshold (A/D data < lower threshold bit)
1	Over upper threshold (A/D data > upper threshold bit)

(n=32 to 63)

- Upon outside range check, these bits show that the range comparison result is higher than the upper threshold setting register or lower than the lower threshold setting register.
- Upon outside range check and when the range comparison result is inside the range, the value of the threshold excess flag bit remains the same.
- When the range comparison interrupt factor flag bit of the corresponding activation channel is set to "1", the threshold excess flag bit is not updated and the value remains the same even if outside range check detects that the range comparison result is outside range.
- Upon inside-range check, the threshold excess flag bit has no meaning. The value remains the same.

4.2.11. Range Comparison Flag Register: ADRCIF0, ADRCIF1

This section shows the bit configurations of the range comparison flag registers.

The range comparison flag register (ADRCIF) shows interrupt factors for continuous detection of range comparison results.

■ ADRCIF0: Address 8140_H (access: byte, half word, word)

bit	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
Field	RCINT31	RCINT30	RCINT29	RCINT28	RCINT27	RCINT26	RCINT25	RCINT24
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	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
Field	RCINT23	RCINT22	RCINT21	RCINT20	RCINT19	RCINT18	RCINT17	RCINT16
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	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
Field	RCINT15	RCINT14	RCINT13	RCINT12	RCINT11	RCINT10	RCINT9	RCINT8
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	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Field	RCINT7	RCINT6	RCINT5	RCINT4	RCINT3	RCINT2	RCINT1	RCINT0
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] RCINT31 to RCINT0: Conversion Data Error Flag Bits

RCINT[n]	Function
0	Range comparison interrupt factor clear status
1	Generation status of the interrupt factor for continuous detection of range comparison results.

(n=0 to 31)

- RCINT bit is set to "1" by the continuous detection of range comparison results of the corresponding activation channel.
- When the RCINT bit of the corresponding activation channel and range comparison interrupt request permission is "1", range comparison interrupt request is generated.
- The RCINT bit is read-only. Writing to the RCINT bit has no effect on operation.
- The RCINT bit is cleared to "0" by writing "1" in the corresponding RCINT clear bit (ADRCIFC[n].RCINTC).

Note:

- *When the software clear (writing ADRCIFC.RCINTC="1") and hardware set are generated simultaneously, the hardware set takes precedence.*

■ ADRCIF1: Address 8540_H (access: byte, half word, word)

bit	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
Field	RCINT63	RCINT62	RCINT61	RCINT60	RCINT59	RCINT58	RCINT57	RCINT56
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	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
Field	RCINT55	RCINT54	RCINT53	RCINT52	RCINT51	RCINT50	RCINT49	RCINT48
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	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
Field	RCINT47	RCINT46	RCINT45	RCINT44	RCINT43	RCINT42	RCINT41	RCINT40
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	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Field	RCINT39	RCINT38	RCINT37	RCINT36	RCINT35	RCINT34	RCINT33	RCINT32
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] RCINT63 to RCINT32: Conversion Data Error Flag Bits

RCINT[n]	Function
0	Range comparison interrupt factor clear status
1	Generation status of the interrupt factor for continuous detection of range comparison results.

(n=32 to 63)

- RCINT bit is set to "1" by the continuous detection of range comparison results of the corresponding activation channel.
- When the RCINT bit of the corresponding activation channel and range comparison interrupt request enable bit is "1", range comparison interrupt request is generated.
- The RCINT bit is read-only. Writing to the RCINT bit has no effect on operation.
- The RCINT bit is cleared to "0" by writing "1" in the corresponding RCINT clear bit (ADRCIFC[n].RCINTC).

Note:

- *When the software clear (writing ADRCIFC.RCINTC="1") and hardware set are generated simultaneously, the hardware set takes precedence.*

4.2.12. Scan Conversion Control Status Register: ADSCANS0, ADSCANS1

This section shows the bit configuration of the scan conversion control status registers.

The scan conversion control status register (ADSCANS) performs the selection of continuous/pause scan mode at the time of conversion count specification, scan conversion completion interrupt request permission/prohibition and the display of scan conversion completion interrupt request status.

- ADSCANS0: Address 8144_H (access: byte, half word, word)
- ADSCANS1: Address 8544_H (access: byte, half word, word)

bit	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Field	SCINT	SCIE	SCMD	Reserved				
R/W Attribute	R, WX	R/W	R/W	R0/W0	R0/W0	R0/W0	R0/W0	R0/W0
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

[bit7] SCINT: Scan Conversion Completion Interrupt Factor Flag Bit

SCINT	Function
0	Status of scan conversion completion interrupt factor clear at the time of specification of conversion count by channel
1	Status of interrupt factor generation due to the completion of scan conversion at the time of specification of conversion count by channel

- The completion of scan conversion at the time of specification of conversion count by channel sets SCINT bit to "1".
- When the SCINT bit and scan conversion completion interrupt request enable bit are "1", scan conversion completion interrupt request at the time of specification of conversion count is generated.
- The SCINT bit is read-only. Writing to the SCINT bit has no effect on operation.
- The SCINT bit is cleared to "0" by writing "1" in the SCINT clear bit (ADSCANSC0.SCINTC, ADSCANSC1.SCINTC).

Note:

- *When the software clear (writing ADSCANSC0.SCINTC="1", ADSCANSC1.SCINTC="1") and hardware set are generated simultaneously, the hardware set takes precedence.*

[bit6] SCIE: Scan Conversion Completion Interrupt Request Enable Bit

SCIE	Description
0	Scan conversion completion interrupt disabled
1	Scan conversion completion interrupt enabled

When both the scan conversion completion interrupt factor flag bit and scan conversion completion interrupt request enable bit are set to "1", interrupt request is generated.

[bit5] SCMD: Continuous/Pause Scan Conversion Mode Selection Bit

SCMD	Description
0	Continuous scan conversion mode
1	Pause scan conversion mode

Select the scan conversion mode at the time of specification of conversion count by channel.

[bit4 to bit0] Reserved

Always write "0" to these bits.

4.2.13. Activation Channel Conversion Count Setting Register: ADNCS0 to ADNCS031

This section shows the bit configuration of the activation channel conversion count setting registers.

The activation channel conversion count setting register (ADNCS) performs the selection of conversion count specification scan conversion execution permission/prohibition, specification of conversion count by activation channel.

- ADNCSm (m=0 to 15): Address 8148_H to 8157_H (access: byte, half word, word)
- ADNCSm (m=16 to 31): Address 8548_H to 8557_H (access: byte, half word, word)

bit	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Field	CNTEN (m*2+1)	Reserved	CCNT (m*2+1)1	CCNT (m*2+1)0	CNTEN (m*2)	Reserved	CCNT (m*2)1	CCNT (m*2)0
R/W Attribute	R/W	R0,W0	R/W	R/W	R/W	R0,W0	R/W	R/W
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

[bit7, bit3] CNTEN: Conversion Count Specification Scan Conversion Execution Enable Bit

CNTEN[n]	Description
0	Conversion count specification scan conversion execution disabled
1	Conversion count specification scan conversion execution enabled

(n=0 to 63)

- These bits enable/disable the conversion count specification scan conversion execution for each activation channel.
- The scan conversion by specifying the conversion count for each channel can control only 1 scan conversion group for each 12-bit A/D converter.

[bit6, bit2] Reserved

Always write "0" to these bits.

[bit5, bit4, bit1, bit0] CCNT: Conversion Count Specification Bits

CCNT[n1]	CCNT[n]0	Description
0	0	Set the conversion count to 1.
0	1	Set the conversion count to 2.
1	0	Set the conversion count to 3.
1	1	Set the conversion count to 4.

(n=0 to 63)

- These bits control the scan conversion count when the conversion count specification scan conversion execution of the activation channel is enabled.

Note:

- *Do not change the conversion count specification bit while the corresponding activation channel is requesting A/D conversion.*

4.2.14. Data Protection Status Flag Register: ADPRTF0, ADPRTF1

This section shows the bit configurations of the data protection status flag registers.

The data protection status flag register (ADPRTF) shows the protection status of the A/D data register for each activation channel.

■ ADPRTF0: Address 8158_H (access: byte, half word, word)

bit	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
Field	PRTF31	PRTF30	PRTF29	PRTF28	PRTF27	PRTF26	PRTF25	PRTF24
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	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
Field	PRTF23	PRTF22	PRTF21	PRTF20	PRTF19	PRTF18	PRTF17	PRTF16
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	*1.* -							
Initial Value	0	0	0	0	0	0	0	0

bit	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
Field	PRTF15	PRTF14	PRTF13	PRTF12	PRTF11	PRTF10	PRTF9	PRTF8
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	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Field	PRTF7	PRTF6	PRTF5	PRTF4	PRTF3	PRTF2	PRTF1	PRTF0
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] PRTF31 to PRTF0: Data Protection Status Flag Bits

PRTF	Description
0	Data protection status disabled
1	Data protection status enabled

- These bits show the data protection status of the A/D data register for each activation channel.
- Writing has no effect on data protection status.

■ ADPRTF1: Address 8558_H (access: byte, half word, word)

bit	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
Field	PRTF63	PRTF62	PRTF61	PRTF60	PRTF59	PRTF58	PRTF57	PRTF56
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	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
Field	PRTF55	PRTF54	PRTF53	PRTF52	PRTF51	PRTF50	PRTF49	PRTF48
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	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
Field	PRTF47	PRTF46	PRTF45	PRTF44	PRTF43	PRTF42	PRTF41	PRTF40
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	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Field	PRTF39	PRTF38	PRTF37	PRTF36	PRTF35	PRTF34	PRTF33	PRTF32
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] PRTF63 to PRTF32: Data Protection Flag Bits

PRTF	Description
0	Data protection status disabled
1	Data protection status enabled

- These bits show the data protection status of the A/D data register for each activation channel.
- Writing has no effect on data protection status.

4.2.15. Activation Channel Conversion Completion Flag Register: ADEOCF0, ADEOCF1

This section shows the bit configurations of the activation channel conversion completion flag registers.

The activation channel conversion completion flag register (ADEOCF) shows the conversion count completion status at the time of conversion count specification scan conversion for each activation channel.

■ ADEOCF0: Address 815C$_H$ (access: byte, half word, word)

bit	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
Field	EOCF31	EOCF30	EOCF29	EOCF28	EOCF27	EOCF26	EOCF25	EOCF24
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	1	1	1	1	1	1	1	1

bit	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
Field	EOCF23	EOCF22	EOCF21	EOCF20	EOCF19	EOCF18	EOCF17	EOCF16
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	1	1	1	1	1	1	1	1

bit	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
Field	EOCF15	EOCF14	EOCF13	EOCF12	EOCF11	EOCF10	EOCF9	EOCF8
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	1	1	1	1	1	1	1	1

bit	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Field	EOCF7	EOCF6	EOCF5	EOCF4	EOCF3	EOCF2	EOCF1	EOCF0
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	1	1	1	1	1	1	1	1

[bit31 to bit0] EOCF31 to EOCF0: Conversion Count Completion Flag Bits

EOCF	Description
0	The specified conversion count has not been completed.
1	The specified conversion count has been completed.

- When the conversion count specification scan conversion execution of the corresponding activation channel is enabled (ADNCS.CNTEN[n]="1"), the display of conversion count completion flag bit (EOCF) is enabled.
- These bits show the completion status of the specified number of scan conversion execution (ADNCS.CCNT1, CCNT0).

- When EOCF="0", the specified number of scan conversion has not been completed.
- When EOCF="1", the specified number of scan conversion has been completed.
- When A/D activation has not been requested (ADTCS.BUSY="0") or scan conversion restarted from the beginning, conversion count completion flag bit (EOCF) is cleared to "0".
- Writing has no effect on data protection status.

Notes:

- *When the conversion count specification scan conversion execution of the corresponding activation channel is disabled (ADNCS.CNTEN[n]="0"), the read data of the conversion count completion flag bit (EOCF[n]) has no meaning.*
- *At the time of the continuous scan conversion mode (ADSCANS.SCMD="0"), conversion count completion flag bit (EOCF) which was set to the final channel of scan conversion restarts immediately after the completion of scan conversion. So, if it is set to "1", it is cleared to "0" immediately.*

■ ADEOCF1: Address 855C$_H$ (access: byte, half word, word)

bit	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
Field	EOCF63	EOCF62	EOCF61	EOCF60	EOCF59	EOCF58	EOCF57	EOCF56
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	1	1	1	1	1	1	1	1

bit	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
Field	EOCF55	EOCF54	EOCF53	EOCF52	EOCF51	EOCF50	EOCF49	EOCF48
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	1	1	1	1	1	1	1	1

bit	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
Field	EOCF47	EOCF46	EOCF45	EOCF44	EOCF43	EOCF42	EOCF41	EOCF40
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	1	1	1	1	1	1	1	1

bit	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Field	EOCF39	EOCF38	EOCF37	EOCF36	EOCF35	EOCF34	EOCF33	EOCF32
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	1	1	1	1	1	1	1	1

[bit31 to bit0] EOCF63 to EOCF32: Conversion Count Completion Flag Bits

EOCF	Description
0	The specified conversion count has not been completed.
1	The specified conversion count has been completed.

- When the conversion count specification scan conversion execution of the corresponding activation channel is enabled (ADNCS.CNTEN[n]="1"), the display of conversion count completion flag bit (EOCF) is enabled.
- These bits show the completion status of the specified number of scan conversion execution (ADNCS.CCNT1, CCNT0).
- When EOCF="0", the specified number of scan conversion has not been completed.
- When EOCF="1", the specified number of scan conversion has been completed.
- When A/D activation has not been requested (ADTCS.BUSY="0") or scan conversion restarted from the beginning, the conversion count completion flag bit (EOCF) is cleared to "0".
- Writing has no effect on data protection status.

Notes:

- *When the conversion count specification scan conversion execution of the corresponding activation channel is disabled (ADNCS.CNTEN[n]="0"), the read data of the conversion count completion flag bit (EOCF[n]) has no meaning.*
- *In continuous scan conversion mode (ADSCANS.SCMD="0"), the conversion count completion flag bit (EOCF) which was set to the final channel of scan conversion restarts immediately after the completion of scan conversion. So, if it is cleared to "0" immediately after it is set to "1".*

4.2.16. A/D Activation Trigger Control Status Clear Register: ADTCSC0 to ADTCSC63

This section shows the bit configuration of the A/D activation trigger control status clear registers.

The A/D activation trigger control status clear register (ADTCSC) clears the bits in the A/D activation trigger control status register (ADTCS[n]).

- ADTCSC0 to ADTCSC31: Address 8160_H to $819E_H$ (access: byte, half word, word)
- ADTCSC32 to ADTCSC63: Address 8560_H to $859E_H$ (access: byte, half word, word)

bit	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
Field	BUSYC	INTC	Reserved					
R/W Attribute	R0, W	R0, W	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	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Field	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

[bit15] BUSYC: BUSY Clear Bit

BUSYC	Function
0	This has no effect on the BUSYC bit and the corresponding A/D activation request in progress bit (ADTCS[n].BUSY).
1	Clear the corresponding A/D activation request in progress bit (ADTCS[n].BUSY).

- When "1" is written in the BUSYC bit, the corresponding A/D activation request in progress bit (ADTCS[n].BUSY) is cleared to "0".
- "0" is always read from the BUSYC bit.

[bit14] INTC: INT Clear Bit

INTC	Function
0	This has no effect on INTC bit and the corresponding interrupt request flag bit (ADTCS[n].INT).
1	Clear the corresponding interrupt request flag bit (ADTCS[n].INT).

- When "1" is written in the INTC bit, the corresponding interrupt request flag bit (ADTCS[n].INT) is cleared to "0".
- "0" is always read from the INTC bit.

[bit13 to bit0] Reserved

Always write "0" to these bits.

4.2.17. Range Comparison Status Register: ADRCS0 to ADRCS63

This section shows the bit configuration of the range comparison status registers.

The range comparison status register (ADRCS) performs the confirmation of continuous detection count status, inside/outside-range check selection, range comparison interrupt request permission/prohibition, range comparison execution permission/prohibition and upper and lower threshold selection set value confirmation.

- ADRCS0 to ADRCS31: Address $81A0_H$ to $81BF_H$ (access: byte, half word, word)
- ADRCS32 to ADRCS63: Address $85A0_H$ to $85BF_H$ (access: byte, half word, word)

bit	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Field	RCOCDS2	RCOCDS1	RCOCDS0	RCOIRSS	RCOIES	RCOES	RCOTSS1	RCOTSS0
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 to bit5] RCOCDS2 to RCOCDS0: Continuous Detection Count Status Display Bits

RCOCDS2 to RCOCDS0	Function
"000B"	Continuous detection status: 0
"001B"	Continuous detection status: 1
"010B"	Continuous detection status: 2
"011B"	Continuous detection status: 3
"100B"	Continuous detection status: 4
"101B"	Continuous detection status: 5
"110B"	Continuous detection status: 6
"111B"	Continuous detection status: 7

- These bits show the status of the continuous detection count of range comparison results.
- When the range comparison results reach the continuous count specification value (ADRCCS[n].RCOCD2 to RCOCD0), the range comparison interrupt factor flag bit of the corresponding activation channel is set to "1". In addition, continuous detection status is suspended at the continuous count specification value.
- The RCOCDS2 to RCOCDS0 bits are read-only. Writing data to the RCOCDS2 to RCOCDS0 bits has no effect on the operation.

[bit4] RCOIRSS: RCOIRS Set Value Display Bit

RCOIRSS	Function
0	The corresponding inside/outside-range check selection bit (ADRCCS[n].RCOIRS) is set to outside range.
1	The corresponding inside/outside-range check selection bit (ADRCCS[n].RCOIRS) is set to inside range.

- This bit confirms the set value of the corresponding inside/outside-range check selection bit (ADRCCS[n].RCOIRS).
- The RCOIRSS bit is read-only. Writing to the RCOIRSS bit has no effect on operation.

[bit3] RCOIES: RCOIE Set Value Display Bit

RCOIES	Function
0	The corresponding range comparison interrupt request enable bit (ADRCCS[n].RCOIE) is set to range comparison interrupt disabled.
1	The corresponding range comparison interrupt request enable bit (ADRCCS[n].RCOIE) is set to range comparison interrupt enabled.

- This bit confirms the set value of the corresponding range comparison interrupt request enable bit (ADRCCS[n].RCOIE).
- The RCOIES bit is read-only. Writing to the RCOIES bit has no effect on operation.

[bit2] RCOES: RCOE Set Value Display Bit

RCOES	Function
0	The corresponding range comparison execution enable bit (ADRCCS[n].RCOE) is set to range comparison execution disabled.
1	The corresponding range comparison execution enable bit (ADRCCS[n].RCOE) is set to range comparison execution enabled.

- This bit confirms the set value of the corresponding range comparison execution enable bit (ADRCCS[n].RCOE).
- The RCOES bit is read-only. Writing to the RCOES bit has no effect on operation.

[bit1, bit0] RCOTSS1, RCOTSS0: RCOTS Set Value Display Bits

RCOTSS1	RCOTSS0	Function
0	0	The corresponding upper and lower threshold selection bit (ADRCCS[n].RCOTS1, RCOTS0) is set to select upper/lower threshold setting register 0.
0	1	The corresponding upper and lower threshold selection bit (ADRCCS[n].RCOTS1, RCOTS0) is set to select upper/lower threshold setting register 1.
1	0	The corresponding upper and lower threshold selection bit (ADRCCS[n].RCOTS1, RCOTS0) is set to select upper/lower threshold setting register 2.
1	1	The corresponding upper and lower threshold selection bit (ADRCCS[n].RCOTS1, RCOTS0) is set to select upper/lower threshold setting register 3.

- This bit confirms the set value for the corresponding upper and lower threshold selection bit (ADRCCS[n].RCOTS1, RCOTS0).
- The RCOTSS1 and RCOTSS0 bits are read-only. Writing data to the RCOTSS1 and RCOTSS0 bits has no effect on the operation.

4.2.18. Range Comparison Flag Clear Register: ADRCIFC0, ADRCIFC1

This section shows the bit configurations of the range comparison flag clear registers.

The range comparison flag clear register (ADRCIFC) clears the bits in the range comparison flag register.

■ ADRCIFC0: Address $81C0_H$ (access: byte, half word, word)

bit	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
Field	RCINTC31	RCINTC30	RCINTC29	RCINTC28	RCINTC27	RCINTC26	RCINTC25	RCINTC24
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	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
Field	RCINTC23	RCINTC22	RCINTC21	RCINTC20	RCINTC19	RCINTC18	RCINTC17	RCINTC16
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	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
Field	RCINTC15	RCINTC14	RCINTC13	RCINTC12	RCINTC11	RCINTC10	RCINTC9	RCINTC8
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	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Field	RCINTC7	RCINTC6	RCINTC5	RCINTC4	RCINTC3	RCINTC2	RCINTC1	RCINTC0
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 to bit0] RCINTC31 to RCINTC0: RCINT Clear Bits

RCINTC[n]	Function
0	This has no effect on the RCINTC[n] bit and the corresponding range comparison interrupt factor flag bit (ADRCIF.RCINT[n])
1	Clear the corresponding range comparison interrupt factor flag bit (ADRCIF.RCINT[n]).

- When "1" is written in the RCINTC[n] bit, clear the corresponding range comparison interrupt factor flag bit (ADRCIF.RCINT[n]) to "0".
- "0" is always read from the RCINTC[n] bit.

■ ADRCIFC1: Address $85C0_H$ (access: byte, half word, word)

bit	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
Field	RCINTC 63	RCINTC 62	RCINTC 61	RCINTC 60	RCINTC 59	RCINTC 58	RCINTC 57	RCINTC 56
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	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
Field	RCINTC 55	RCINTC 54	RCINTC 53	RCINTC 52	RCINTC 51	RCINTC 50	RCINTC 49	RCINTC 48
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	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
Field	RCINTC 47	RCINTC 46	RCINTC 45	RCINTC 44	RCINTC 43	RCINTC 42	RCINTC 41	RCINTC 40
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	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Field	RCINTC 39	RCINTC 38	RCINTC 37	RCINTC 36	RCINTC 35	RCINTC 34	RCINTC 33	RCINTC 32
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 to bit0] RCINTC63 to RCINTC32: RCINT Clear Bits

RCINTC[n]	Function
0	This has no effect on the RCINTC[n] bit and the corresponding range comparison interrupt factor flag bit (ADRCIF.RCINT[n])
1	Clear the corresponding range comparison interrupt factor flag bit (ADRCIF.RCINT[n]).

- When "1" is written in the RCINTC[n] bit, clear the corresponding range comparison interrupt factor flag bit (ADRCIF.RCINT[n]) to "0".
- "0" is always read from the RCINTC[n] bit.

4.2.19. Scan Conversion Control Status Clear Register: ADSCANSC0, ADSCANSC1

This section shows the bit configuration of the scan conversion control status clear registers.

The scan conversion control status clear register (ADSCANSC) clears the bits in the scan conversion control status register (ADSCANS).

- ADSCANSC0: Address $81C4_H$ (access: byte, half word, word)
- ADSCANSC1: Address $85C4_H$ (access: byte, half word, word)

bit	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Field	SCINTC	Reserved						
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

[bit7] SCINTC: SCINT Clear Bit

SCINTC	Function
0	This has no effect on the SCINTC bit and the scan conversion completion interrupt factor flag bit (ADSCANS0.SCINT).
1	Clear the scan conversion completion interrupt factor flag bit (ADSCANS0.SCINT).

- When "1" is written in the SCINTC bit, clear the scan conversion completion interrupt factor flag bit (ADSCANS0.SCINT) to "0".
- "0" is always read from the SCINTC bit.

[bit6 to bit0] Reserved

Always write "0" to these bits.

4.3. Registers of the 12-Bit A/D Converter Control

This section explains the registers of the 12-bit A/D converter control.

The 12-bit A/D converter control has A/D control status registers, A/D channel status registers, A/D mode setting registers, and a sampling register for each A/D channel.

4.3.1. A/D Control Status Register: ADCS0, ADCS1

This section shows the bit configuration of the A/D control status registers.

The A/D control status registers (ADCS) have the function of verifying conversion.

- ADCS0: Address $81C8_H$ (access: byte, half word, word)
- ADCS1: Address $85C8_H$ (access: byte, half word, word)

bit	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
Field	BUSY	Reserved						
R/W Attribute	R,WX	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	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Field	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

[bit15] BUSY: A/D Conversion in Progress Bit

BUSY	Function
0	A/D conversion has stopped.
1	A/D conversion is in progress.

- This bit indicates the operation of the A/D converter.
- If the value read from this bit is "0", A/D conversion is stopped. If it is "1", A/D conversion is in progress.
- Writing does not change anything and does not produce any effect.

[bit14 to bit0] Reserved

Always write "0" to these bits.

4.3.2. A/D Channel Status Register: ADCH

This section shows the bit configuration of the A/D channel status registers.

During A/D conversion, the A/D channel status registers (ADCH) can read analog channel numbers for which the conversion is being performed.

- ADCH0: Address $81CA_H$ (access: byte, half word, word)
- ADCH1: Address $85CA_H$ (access: byte, half word, word)

bit	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Field	Reserved			CH4	CH3	CH2	CH1	CH0
R/W Attribute	R0,W0	R0,W0	R0,W0	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

[bit7 to bit5] Reserved

Always write "0" to these bits.

[bit4 to bit0] CH4 to CH0: Analog Channel Bits

CH4	CH3	CH2	CH1	CH0	Function	
					ADCH0	ADCH1
0	0	0	0	0	ch.0	ch.32
0	0	0	0	1	ch.1	ch.33
:					:	:
0	0	1	1	0	ch.6	ch.38
0	0	1	1	1	ch.7	ch.39
0	1	0	0	0	ch.8	ch.40
:					:	:
0	1	1	1	0	ch.14	ch.46
0	1	1	1	1	ch.15	ch.47
1	0	0	0	0	ch.16	ch.48
1	0	0	0	1	ch.17	ch.49
:					:	:
1	1	1	1	0	ch.30	ch.62
1	1	1	1	1	ch.31	ch.63

- During A/D conversion, these bits can check analog channel numbers for which the conversion is being performed.

4.3.3. A/D Mode Setting Register: ADMD

This section shows the bit configuration of the A/D mode setting registers.

The A/D mode setting registers (ADMD) have the function of setting the compare time and sampling time for A/D conversion.

- ADMD0: Address $81CB_H$ (access: byte, half word, word)
- ADMD1: Address $85CB_H$ (access: byte, half word, word)

bit	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Field	STPCEN	Reserved			CT1	CT0	ST1	ST0
R/W Attribute	R/W	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

[bit7] STPCEN: Sampling Time Setting per Channel Enable Bit

STPCEN	Description
0	Sampling time setting common to all channels
1	Sampling time setting for each channel

- This bit selects whether to set sampling time for A/D conversion for each channel or to set a common setting.
- If STPCEN is "0", sampling time common to all channels is used. Use the sampling time setting bits to set sampling time.
- If STPCEN is "1", you can set the sampling time for each channel. Use the sampling time setting per A/D channel bits to set the sampling time for each channel.

[bit6 to bit4] Reserved

Always write "0" to these bits.

[bit3, bit2] CT1, CT0: Compare Time Setting Bits

CT1	CT0	Function
0	0	28 peripheral clock cycles (A/D clock output: peripheral clock/2)
0	1	42 peripheral clock cycles (A/D clock output: peripheral clock/3)
1	0	56 peripheral clock cycles (A/D clock output: peripheral clock/4)
1	1	112 peripheral clock cycles (A/D clock output: peripheral clock/8)

- These bits select the compare time for A/D conversion.
- After analog input is fetched (the sampling time has elapsed) and the time set in these bits has elapsed, conversion result data is determined.

■ Example of settings

Peripheral Clock (MHz)	CT1	CT0	Compare Period (ns)
40	0	0	700
32	0	0	875
24	0	0	1166.7
16	0	0	1750

Notes:

- *Be sure to overwrite these bits during A/D operation before the conversion operation has stopped.*
- *Set the A/D clock output to 500 kHz or higher.*
  For example, if you want an A/D operation with a 4 MHz (High-speed) CR oscillation clock and set CT1=1 and CT0=1, then the A/D clock output (type.) is 500 kHz (division by 8). However, considering variation in characteristics, the clock output may become less than 500 kHz. Thus, do not set CT1=1 and CT0=1.

[bit1, bit0] ST1, ST0: Sampling Time Setting Bits

ST1	ST0	Function
0	0	12 peripheral clock cycles (A/D clock output: peripheral clock/2)
0	1	18 peripheral clock cycles (A/D clock output: peripheral clock/3)
1	0	24 peripheral clock cycles (A/D clock output: peripheral clock/4)
1	1	48 peripheral clock cycles (A/D clock output: peripheral clock/8)

- These bits select the sampling time for A/D conversion.
- When A/D is activated, the time and analog input set in these bits are fetched.

■ Example of settings (use conditions: AVcc=4.5 V to 5.5 V)

Peripheral Clock (MHz)	ST1	ST0	Sampling Period (ns)
40	1	1	1200
32	1	0	750
24	0	1	750
16	0	0	750

Notes:

- Be sure to overwrite these bits during A/D operation before the conversion operation has stopped.
- Set the A/D clock output to 500 kHz or higher.
  For example, if you want an A/D operation with a 4 MHz (High-speed) CR oscillation clock and set ST1 =1 and ST0=1, then the A/D clock output (type.) is 500 kHz (division by 8). However,
  considering variation in characteristics, the clock output may become less than 500 kHz. Thus, do not set ST1=1 and ST0=1.

4.3.4. Sampling Time Setting per A/D Channel Register: ADSTPCS

This section shows the bit configuration of the sampling time setting per A/D channel registers.

The sampling time setting per A/D channel register (ADSTPCS) sets the sampling time for A/D conversion for each channel.

- ADSTPCSm (m=0 to 7): Address $81CC_H$ to $81D3_H$ (access: byte, half word, word)
- ADSTPCSm (m=8 to 15): Address $85CC_H$ to $85D3_H$ (access: byte, half word, word)

bit	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Field	STCH (m*4+3)1	STCH (m*4+3)0	STCH (m*4+2)1	STCH (m*4+2)0	STCH (m*4+1)1	STCH (m*4+1)0	STCH (m*4)1	STCH (m*4)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

[bit7 to 0] STCHn1, STCHn0 (n=0 to 63): Sampling Time Setting Bits

STCHn1	STCHn0	Description
0	0	12 peripheral clock cycles (A/D clock output: peripheral clock/2)
0	1	18 peripheral clock cycles (A/D clock output: peripheral clock/3)
1	0	24 peripheral clock cycles (A/D clock output: peripheral clock/4)
1	1	48 peripheral clock cycles (A/D clock output: peripheral clock/8)

- These bits select the sampling time for A/D conversion for each channel.
- The sampling time setting for each channel (STCHn1, STCHn0) is enabled when the sampling time setting per channel enable bit (ADMD.STPCEN) is "1".
- The correspondence of the channel for which the sampling time is set to the analog channel is shown below.

Notes:

- Be sure to overwrite these bits during A/D operation before the conversion operation has stopped.
- Set the A/D clock output to 500 kHz or higher.
  For example, if you want an A/D operation with a 4 MHz (High-speed) CR oscillation clock and set ST1=1 and ST0=1, then the A/D clock output (type.) is 500 kHz (division by 8). However,
  considering variation in characteristics, the clock output may become less than 500 kHz. Thus, do not set STCHn1=1 and STCHn0=1.

Table 4-4 Correspondence of Sampling Time Setting for Each Channel to Analog Channel (A/D Converter Unit 0)

Sampling Time Setting per Channel Bits	Analog Channel Number
	ADSTPCS0 to 31
STCH001, STCH000	Channel 0
STCH011, STCH010	Channel 1
:	:
STCH061, STCH060	Channel 6
STCH071, STCH070	Channel 7
:	:
STCH141, STCH140	Channel 14
STCH151, STCH150	Channel 15
:	:
STCH300, STCH300	Channel 30
STCH311, STCH310	Channel 31

Table 4-5 Correspondence of Sampling Time Setting for Each Channel to Analog Channel (A/D Converter Unit 1)

Sampling Time Setting per Channel Bits	Analog Channel Number
	ADSTPCS32 to 63
STCH321, STCH320	Channel 32
STCH331, STCH330	Channel 33
:	:
STCH381, STCH380	Channel 38
STCH391, STCH390	Channel 39
:	:
STCH461, STCH460	Channel 46
STCH471, STCH470	Channel 47
:	:
STCH621, STCH620	Channel 62
STCH631, STCH630	Channel 63

5. Explanation of Operation

This section explains operation.

5.1. Interrupts From A/D Activation Compare

5.2. Operation of the A/D Activation Compare

5.3. Operation of the A/D Activation Arbitration

5.4. Operation of the 12-Bit A/D Converter

5.5. Failure Diagnosis Function of the 12-Bit A/D Converter

5.1. Interrupts From A/D Activation Compare

This section explains the interrupt control bits and interrupt factors of the A/D activation compare.

This section explains the interrupt control bits and interrupt factors of the A/D activation compare.

5.1.1. A/D Conversion End Interrupt

This section explains the A/D conversion end interrupt.

Table 5-1 Interrupt Control Bits and Interrupt Factor for A/D Conversion End Interrupt

	A/D Conversion End Interrupt
Interrupt Request Flag bit	INT:bit14 in the A/D activation trigger control status register (ADTCS)
Interrupt Request Enable bit	INTE:bit13 in the A/D activation trigger control status register (ADTCS)
Interrupt Factor	Writing A/D conversion results to the A/D data register

An A/D conversion end interrupt request can be generated when A/D conversion for the compare channel that activated the A/D converter ends. The A/D conversion end interrupt can be controlled for each activation channel.

When A/D conversion results are set in the A/D data register (ADTCD), the INT bit in the A/D activation trigger control status register (ADTCS) is set to "1". At this time, if the interrupt request is enabled (INTE in ADTCS=1), an interrupt request is output to the interrupt controller.

5.1.2. Conversion Count-Specified Scan Conversion Completion Interrupt

This section explains the conversion count-specified scan conversion completion interrupt.

Table 5-2 Interrupt Control Bits and Interrupt Factor When Conversion Count-Specified Scan Conversion Ends

	Conversion Count-Specified Scan Conversion Completion Interrupt
Interrupt Request Flag bit	SCINT:bit7 in the scan conversion control status register (ADSCANS)
Interrupt Request Enable bit	SCIE:bit6 in the scan conversion control status register (ADSCANS)
Interrupt Factor	After a scan conversion for which the conversion count is specified completes as many A/D conversions as specified for the final activation channel

When performing a scan conversion for which the conversion count is specified, the A/D activation compare can generate an interrupt request after as many A/D conversions as specified for the final activation channel have been completed. The conversion count-specified scan conversion completion interrupt is controlled for each 12-bit A/D converter unit.

After a scan conversion for which the conversion count is specified completes as many A/D conversions as specified for the final activation channel, the scan conversion completion interrupt factor flag bit (ADSCANS.SCINT) is set to "1". At this time, if the scan conversion completion interrupt request enable bit is "enable" (ADSCANS.SCIE="1"), an interrupt request is output to the interrupt controller. The conversion count-specified scan conversion execution enable bit (ADNCS.CNTEN) and the conversion count specification bit (ADNCS.CCNT1, CCNT0) control the scan conversion for which the conversion count is specified.

5.1.3. Range Comparison Interrupt

This section explains the range comparison interrupt.

Table 5-3 Interrupt Control Bits and Interrupt Factor for Range Comparison Interrupt

	Range Comparison Interrupt
Interrupt Request Flag bit	RCINT[n]:bit[n] in the range comparison flag register (ADRCIF)
Interrupt Request Enable bit	RCOIE:bit3 in the range comparison control status register (ADRCCS[n])
Interrupt Factor	After continuous detection function determination when range comparison is performed

(n=Relevant activation channel)

When range comparison execution is enabled, the A/D activation compare performs range comparison. This is done by comparing conversion results stored in the A/D data register (ADTCD) with one of the combinations of upper threshold setting registers 0 to 7 (ADRCUT0 to 7) and lower threshold setting registers 0 to 7 (ADRCLT0 to 7) for range comparison. If the continuity of the range comparison result is identified, a range comparison interrupt request can be generated. The range comparison interrupt can be controlled for each activation channel.
When range comparison execution is enabled (ADRCCS.RCOE="1") and A/D conversion result is set in the A/D data register (ADTCD), range comparison is performed. For range comparison execution, range comparison conditions can be selected using the inside/outside-range check selection bit (ADRCCS.RCOIRS).

- Range comparison conditions when outside-range check (ADRCCS.RCOIRS="0") is selected
  Lower threshold setting register > A/D conversion result
  or
  upper threshold setting register < A/D conversion result
- Range comparison conditions when inside-range check (ADRCCS.RCOIRS="1") is selected
  Lower threshold setting register ≤ A/D conversion result
  and
  upper threshold setting register ≥ A/D conversion result

If range comparison conditions are continuously detected, the range comparison interrupt factor flag bit (ADRCIF.RCINT) is set to "1". At this time, if the range comparison interrupt request enable bit is "enable" (ADRCCS.RCOIE="1"), an interrupt request is output to the interrupt controller. The timing when the range comparison interrupt request is output is delayed by 2 cycles of the peripheral clock from the A/D conversion end interrupt. The number of continuous detections can be selected from 1 to 7 using the continuous detection count specification bits (ADRCCS.RCOCD2 to RCOCD0).

5.2. Operation of the A/D Activation Compare

This section explains the operation of the A/D activation compare.

A/D activation can be requested with one of the following factors: software, reload timer, compare match (when the value of the lower 16 bits of the 32-bit free-run timer becomes the indicated value), or PWM timer.

5.2.1. A/D Activation

This section explains A/D activation.

Activation is requested of one of the 2 A/D converter units. An activation request can be generated for each analog channel. There can be up to 64 analog channels.
An activation request signal is generated to the A/D activation arbitration with one of the following factors: software, reload timer (rising), compare match (when the free-run timer matches the compare register value), or PWM timer. There are 3 types of A/D activation request signals for each activation channel: [software activation request], [reload timer/PWM timer activation request], and [compare match activation request]. One of them becomes active.
When A/D conversion for the relevant channel ends, the activation request is cleared and the conversion data is stored in the A/D data register. At this time, an interrupt can be generated.
Note that even if an activation factor is received while activation is being requested, activation request reactivation is not performed within the activation channel.

- Activation channels are allocated to the 2 A/D converter units. The allocation to each of the units is as follows.
  - Activation channels 0 to 31 : 12-bit A/D converter unit 0
  - Activation channels 32 to 63 : 12-bit A/D converter unit 1

5.2.2. Enabling A/D Activation

This section explains how to enable A/D activation.

Select an A/D activation factor using the A/D activation trigger control factor selection bits (ADTECS.STS2, ADTCS.STS1, STS0). Software, reload timer, compare match, or PWM timer can be selected. When the selected activation factor is generated, an activation request signal is generated to the A/D activation arbitration.

For activation channels that do not activate A/D, you can disable A/D activation requests. To do so, select software activation (ADTECS.STS2="0", ADTCS.STS1, STS0="00_B") and disable software activation for the relevant channels in the A/D software activation channel selection register (ADTSE).

5.2.3. Free-Run Timer Input

This section explains free-run timer input.

Free-run timer input that is used for compare match is independently input to each activation channel. If you want multiple activation channels to use the same free-run timer, make an appropriate setting using the free-run timer selector.

The free-run timer to be used is selected by the RESSEL bit in the resource input setting register (RIC_RESIN), and is input to each activation channel.
For details of RIC_RESIN, see "Resource Input Setting Register" in "CHAPTER: I/O PORT."

RIC_RESINs as many as the number of the activation channels are implemented.

For relationship among RIC_RESIN numbers, free-run timer numbers, and activation channels, see Table 5-4.

Table 5-4 Connection Relationship Between Free-Run Timers and A/D Converter Activation Channels

Reg Add	Resource	RESSEL Data[3:0]						
		0000	0001	0010	0011	0100	0101	Other Than Ones on the Left
RIC_RESIN 126 (0x00FC)	A/D activation channel 0	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	Setting prohibited
RIC_RESIN 127 (0x00FE)	A/D activation channel 1	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	Setting prohibited
RIC_RESIN 128 (0x0100)	A/D activation channel 2	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	Setting prohibited
RIC_RESIN 129 (0x0102)	A/D activation channel 3	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	Setting prohibited
RIC_RESIN 130 (0x0104)	A/D activation channel 4	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	Setting prohibited
RIC_RESIN 131 (0x0106)	A/D activation channel 5	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	Setting prohibited
RIC_RESIN 132 (0x0108)	A/D activation channel 6	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	Setting prohibited
RIC_RESIN 133 (0x010A)	A/D activation channel 7	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	Setting prohibited
RIC_RESIN 134 (0x010C)	A/D activation channel 8	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	Setting prohibited
RIC_RESIN 135 (0x010E)	A/D activation channel 9	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	Setting prohibited
RIC_RESIN 136 (0x0110)	A/D activation channel 10	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	Setting prohibited
RIC_RESIN 137 (0x0112)	A/D activation channel 11	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	Setting prohibited
RIC_RESIN 138 (0x0114)	A/D activation channel 12	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	Setting prohibited
RIC_RESIN 139 (0x0116)	A/D activation channel 13	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	Setting prohibited

Reg Add	Resource	RESSEL Data[3:0]						
		0000	0001	0010	0011	0100	0101	Other Than Ones on the Left
RIC_RESIN 140 (0x0118)	A/D activation channel 14	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	Setting prohibited
RIC_RESIN 141 (0x011A)	A/D activation channel 15	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	Setting prohibited
RIC_RESIN 142 (0x011C)	A/D activation channel 16	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	Setting prohibited
RIC_RESIN 143 (0x011E)	A/D activation channel 17	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	Setting prohibited
RIC_RESIN 144 (0x0120)	A/D activation channel 18	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	Setting prohibited
RIC_RESIN 145 (0x0122)	A/D activation channel 19	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	Setting prohibited
RIC_RESIN 146 (0x0124)	A/D activation channel 20	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	Setting prohibited
RIC_RESIN 147 (0x0126)	A/D activation channel 21	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	Setting prohibited
RIC_RESIN 148 (0x0128)	A/D activation channel 22	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	Setting prohibited
RIC_RESIN 149 (0x012A)	A/D activation channel 23	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	Setting prohibited
RIC_RESIN 150 (0x012C)	A/D activation channel 24	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	Setting prohibited
RIC_RESIN 151 (0x012E)	A/D activation channel 25	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	Setting prohibited
RIC_RESIN 152 (0x0130)	A/D activation channel 26	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	Setting prohibited
RIC_RESIN 153 (0x0132)	A/D activation channel 27	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	Setting prohibited
RIC_RESIN 154 (0x0134)	A/D activation channel 28	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	Setting prohibited
RIC_RESIN 155 (0x0136)	A/D activation channel 29	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	Setting prohibited
RIC_RESIN 156 (0x0138)	A/D activation channel 30	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	Setting prohibited
RIC_RESIN 157 (0x013A)	A/D activation channel 31	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	Setting prohibited
RIC_RESIN 158 (0x013C)	A/D activation channel 32	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	Setting prohibited
RIC_RESIN 159 (0x013E)	A/D activation channel 33	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	Setting prohibited
RIC_RESIN 160 (0x0140)	A/D activation channel 34	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	Setting prohibited

Reg Add	Resource	RESSEL Data[3:0]						
		0000	0001	0010	0011	0100	0101	Other Than Ones on the Left
RIC_RESIN 161 (0x0142)	A/D activation channel 35	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	Setting prohibited
RIC_RESIN 162 (0x0144)	A/D activation channel 36	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	Setting prohibited
RIC_RESIN 163 (0x0146)	A/D activation channel 37	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	Setting prohibited
RIC_RESIN 164 (0x0148)	A/D activation channel 38	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	Setting prohibited
RIC_RESIN 165 (0x014A)	A/D activation channel 39	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	Setting prohibited
RIC_RESIN 166 (0x014C)	A/D activation channel 40	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	Setting prohibited
RIC_RESIN 167 (0x014E)	A/D activation channel 41	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	Setting prohibited
RIC_RESIN 168 (0x0150)	A/D activation channel 42	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	Setting prohibited
RIC_RESIN 169 (0x0152)	A/D activation channel 43	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	Setting prohibited
RIC_RESIN 170 (0x0154)	A/D activation channel 44	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	Setting prohibited
RIC_RESIN 171 (0x0156)	A/D activation channel 45	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	Setting prohibited
RIC_RESIN 172 (0x0158)	A/D activation channel 46	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	Setting prohibited
RIC_RESIN 173 (0x015A)	A/D activation channel 47	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	Setting prohibited
RIC_RESIN 174 (0x015C)	A/D activation channel 48	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	Setting prohibited
RIC_RESIN 175 (0x015E)	A/D activation channel 49	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	Setting prohibited
RIC_RESIN 176 (0x0160)	A/D activation channel 50	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	Setting prohibited
RIC_RESIN 177 (0x0162)	A/D activation channel 51	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	Setting prohibited
RIC_RESIN 178 (0x0164)	A/D activation channel 52	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	Setting prohibited
RIC_RESIN 179 (0x0166)	A/D activation channel 53	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	Setting prohibited
RIC_RESIN 180 (0x0168)	A/D activation channel 54	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	Setting prohibited
RIC_RESIN 181 (0x016A)	A/D activation channel 55	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	Setting prohibited

Reg Add	Resource	RESSEL Data[3:0]						
		0000	0001	0010	0011	0100	0101	Other Than Ones on the Left
RIC_RESIN 182 (0x016C)	A/D activation channel 56	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	Setting prohibited
RIC_RESIN 183 (0x016E)	A/D activation channel 57	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	Setting prohibited
RIC_RESIN 184 (0x0170)	A/D activation channel 58	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	Setting prohibited
RIC_RESIN 185 (0x0172)	A/D activation channel 59	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	Setting prohibited
RIC_RESIN 186 (0x0174)	A/D activation channel 60	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	Setting prohibited
RIC_RESIN 187 (0x0176)	A/D activation channel 61	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	Setting prohibited
RIC_RESIN 188 (0x0178)	A/D activation channel 62	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	Setting prohibited
RIC_RESIN 189 (0x017A)	A/D activation channel 63	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	Setting prohibited

The following shows the detailed settings of the resource selection bits (RESSEL).

[bit3:0] RESSEL: Resource Selection Bits

Table 5-5 Relationship Between the RESSEL Bits and the Free-Run Timers to Be Selected

RESSEL3 to RESSEL0				Description
0	0	0	0	Select FRT0
0	0	0	1	Select FRT1
0	0	1	0	Select FRT2
0	0	1	1	Select FRT3
0	1	0	0	Select FRT4
0	1	0	1	Select FRT5
0	1	1	0	This setting is prohibited (operation is not guaranteed).
0	1	1	1	This setting is prohibited (operation is not guaranteed).
1	0	0	0	This setting is prohibited (operation is not guaranteed).
1	0	0	1	This setting is prohibited (operation is not guaranteed).
1	0	1	0	This setting is prohibited (operation is not guaranteed).
1	0	1	1	This setting is prohibited (operation is not guaranteed).

5.2.4. Analog Channel Selection

This section explains analog channel selection.

You can select the analog channel for which A/D conversion is to be performed, using the CHSEL bits in the A/D activation trigger extended control register (ADTECS).

5.2.5. Software Activation

This section explains software activation.

Set the A/D activation trigger control factor selection bits to software activation (ADTECS.STS2="0", ADTCS.STS1, STS0="00_B").
Enable software activation for the desired channels using the A/D software activation channel selection register (ADTSE). An activation request can be generated simultaneously for the multiple channels for which the above setting is configured.
Then, write "1" to the START bit in the A/D software activation register (ADTSS) to set a software activation request signal.

5.2.6. Reload Timer Activation

This section explains reload timer activation.

Set the A/D activation trigger control factor selection bits to reload timer activation (ADTECS.STS2="0", ADTCS.STS1, STS0="10_B").

The reload timer can be selected as one of the functions of the base timer.
The reload timer (base timer) activation trigger is input to each 12-bit A/D converter unit one by one. The correspondence between the base timers and 12-bit A/D converter units is as follows.

- Base timer channel 0 corresponds to A/D converter unit 0.
- Base timer channel 2 corresponds to A/D converter unit 1.

For information on the correspondence between each A/D converter unit of this series and activation channels, see "5.2.1.A/D Activation."

If you select the reload timer for the A/D activation request, A/D conversion starts based on a signal from the output pin (TOUT) of the reload timer.
For an output pin timing chart, see "Reload Timer Function" in "CHAPTER: BASE TIMER."

When a rising of the output pin of the reload timer is detected, the reload timer activation request signal is set.

Note:

- *If you set ADTECS.STS2 to "0" and ADTCS.STS1 and STS0 to "10_B" (reload timer activation), a rising edge of the timer output is regarded as the A/D activation request and A/D conversion starts, even when the corresponding base timer is used as a PWM or PPG timer.*

5.2.7. Compare Match Activation

This section explains compare match activation.

Set the A/D activation trigger control factor selection bits to compare match activation (ADTECS.STS2="0", ADTCS.STS1, STS0="11_B").
Each activation channel has a compare register. When the value of the lower 16 bits of the 32-bit free-run timer matches the value of the compare register, a compare match activation request is set.
Set the desired timer value in the compare register. If you want to request A/D activation at the same timing as when "0" of the free-run timer is detected, set "0000_H". If you want A/D activation to be performed at the same timing as when a compare clear match is found for the free-run timer, set the same value as the compare clear value of the free-run timer.

Settings for the 32-Bit Free-Run Timer

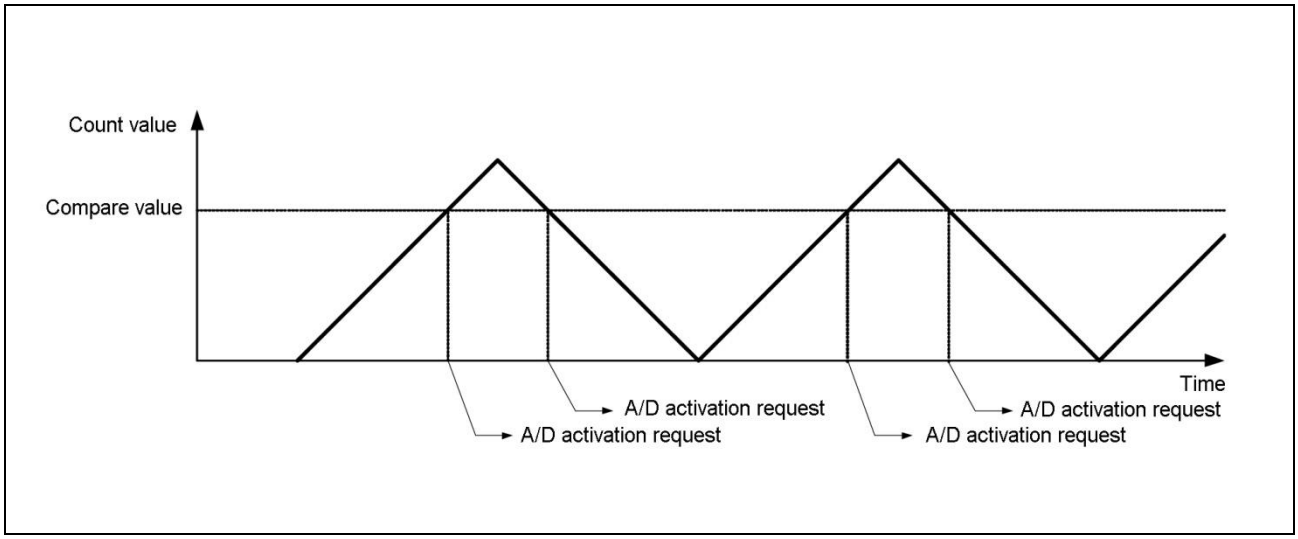
Although the 32-bit free-run timer can count a 32-bit value, only the value of the lower 16 bits can be compared with the compare register. If you use the 32-bit free-run timer as an activation trigger for the A/D converter, set the upper 16 bits of the compare clear register (CPCLR) to "0000_H". This prevents a carry to the upper 16 bits of the free-run timer.

Operation Mode in Compare Match Activation Mode

You can set the operation mode in A/D compare activation mode using the SEL1 and SEL0 bits in the A/D activation trigger control status register (ADTCS).
SEL1 and SEL0 can specify whether to compare the value of the compare register and the value of the lower 16 bits of the 32-bit free-run timer in both up-counting and down-counting, only up-counting, or only down-counting.
If SEL1 and SEL0 are set to "11_B", no activation request signal is generated to the A/D activation arbitration even when the free-run timer matches the compare register value.

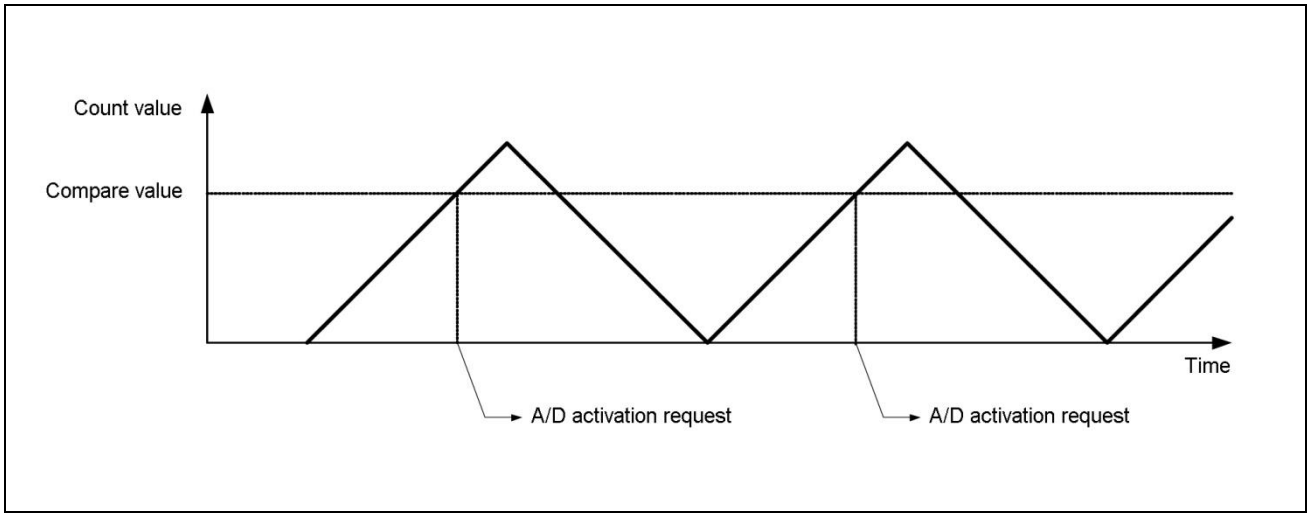
- SEL1, SEL0=00_B: Activation when a compare match occurs

Figure 5-1 SEL1, SEL0=00_B: Activation When a Compare Match Occurs in Up-Counting or Down-Counting



- SEL1, SEL0=01_B: Activation only when a compare match occurs in up-counting

Figure 5-2 SEL1, SEL0=01B: Activation Only When a Compare Match Occurs in Up-Counting



- SEL1, SEL0=10B: Activation only when a compare match occurs in down-counting

Figure 5-3 SEL1, SEL0=10B: Activation Only When a Compare Match Occurs in Down-Counting

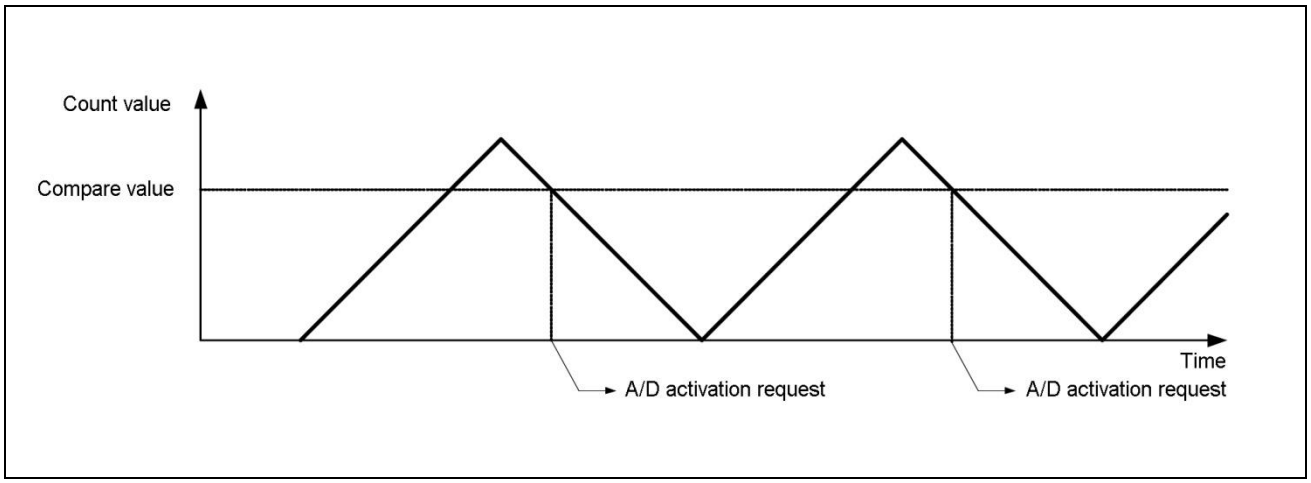
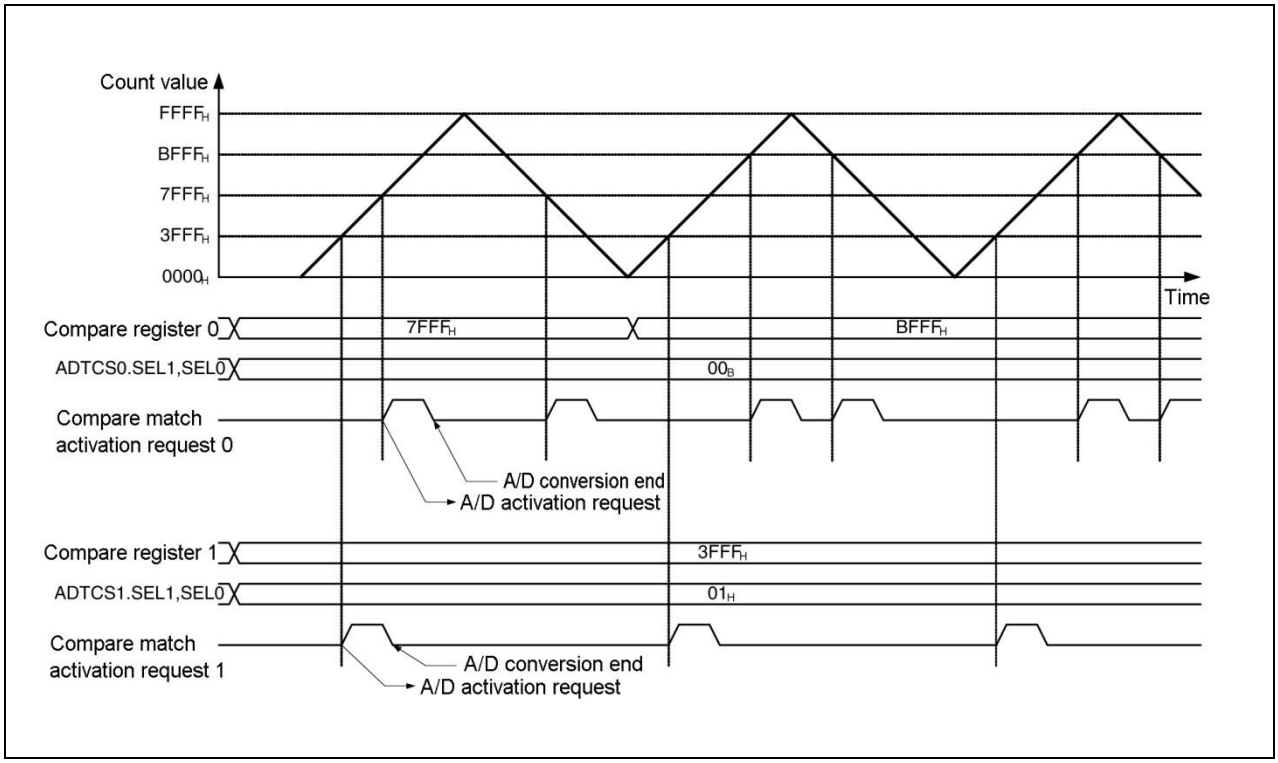


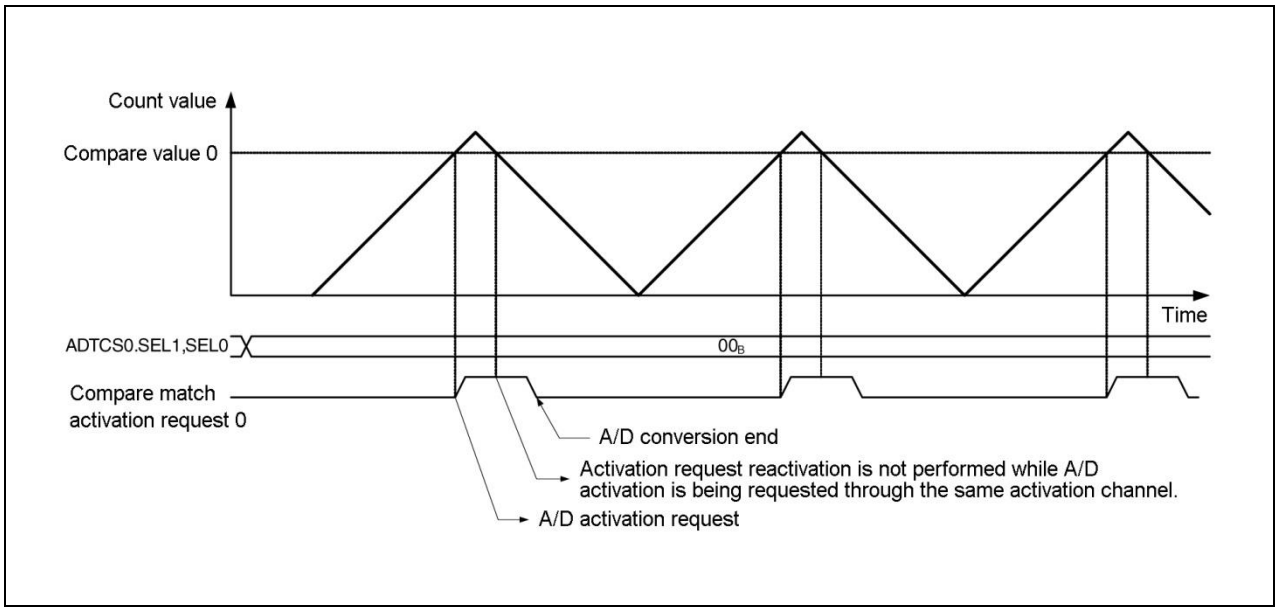
Figure 5-4 Activation Channel 0: A/D Activation in Up/Down-Counting and Activation Channel 1: A/D Activation in Up-Counting



About Setting a Compare Value for Compare Match Activation

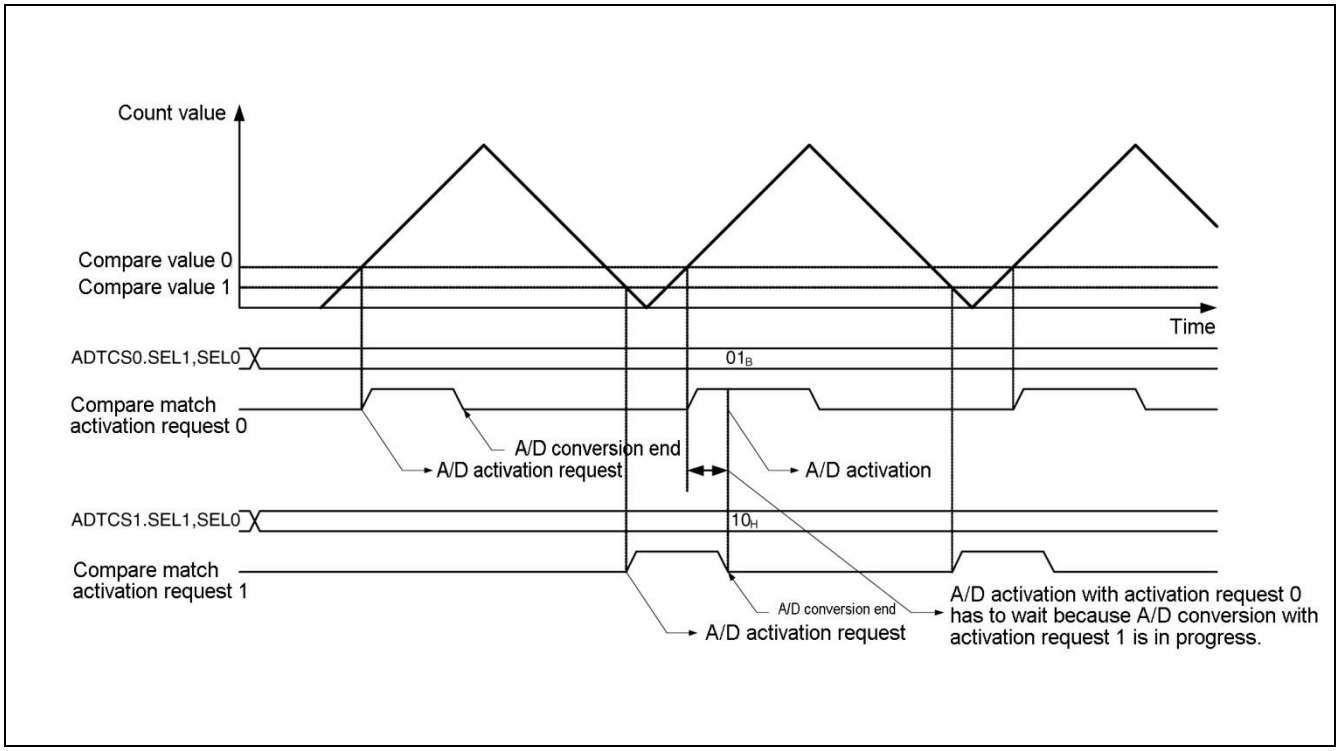
If the interval at which compare matches occur through the same activation channel that performs compare match activation is shorter than A/D conversion time, a compare match that occurs during the A/D conversion will be ignored.

Figure 5-5 Case Where the Interval at Which Compare Matches Occur Through the Same Activation Channel Is Shorter Than A/D Conversion Time



If the interval at which compare matches occur through the activation channels that activate the same A/D converter is shorter than A/D conversion time, the A/D conversion start with the activation request generated during the A/D conversion has to wait. The A/D conversion does not start at the intended timing, and the start of the A/D conversion is delayed.

Figure 5-6 Case Where the Interval at Which Compare Matches Occur Through Activation Channels Is Shorter Than A/D Conversion Time



Compare Register Buffer Function for Compare Match Activation

Write "0" to the BUFX bit in the A/D activation trigger control status register (ADTCS) to enable the buffer function of the compare register.

The buffer timing can be selected by the BTS bit in the ADTCS register. When BTS is "1", the value written to the compare buffer register is transferred to the compare register at the time of compare clear. When BTS is "0", it is transferred to the compare register at the time of "0" detection.

Table 5-6 Conditions for Transfer from the Compare Buffer Register to the Compare Register

BUFX	BTS	Conditions for Transfer from the Compare Buffer Register (ADCOMPB) to the Compare Register (ADCOMP)
0	0	When "0" is detected from the 32-bit free-run timer Or While the 32-bit free-run timer is stopped
0	1	When a compare clear of the 32-bit free-run timer occurs Or While the 32-bit free-run timer is stopped
1	X	Buffer function is not used. (Transfer immediately)

Figure 5-7 Compare Register 0: Buffer Function Is Enabled and Compare Register 1: Buffer Function Is Disabled

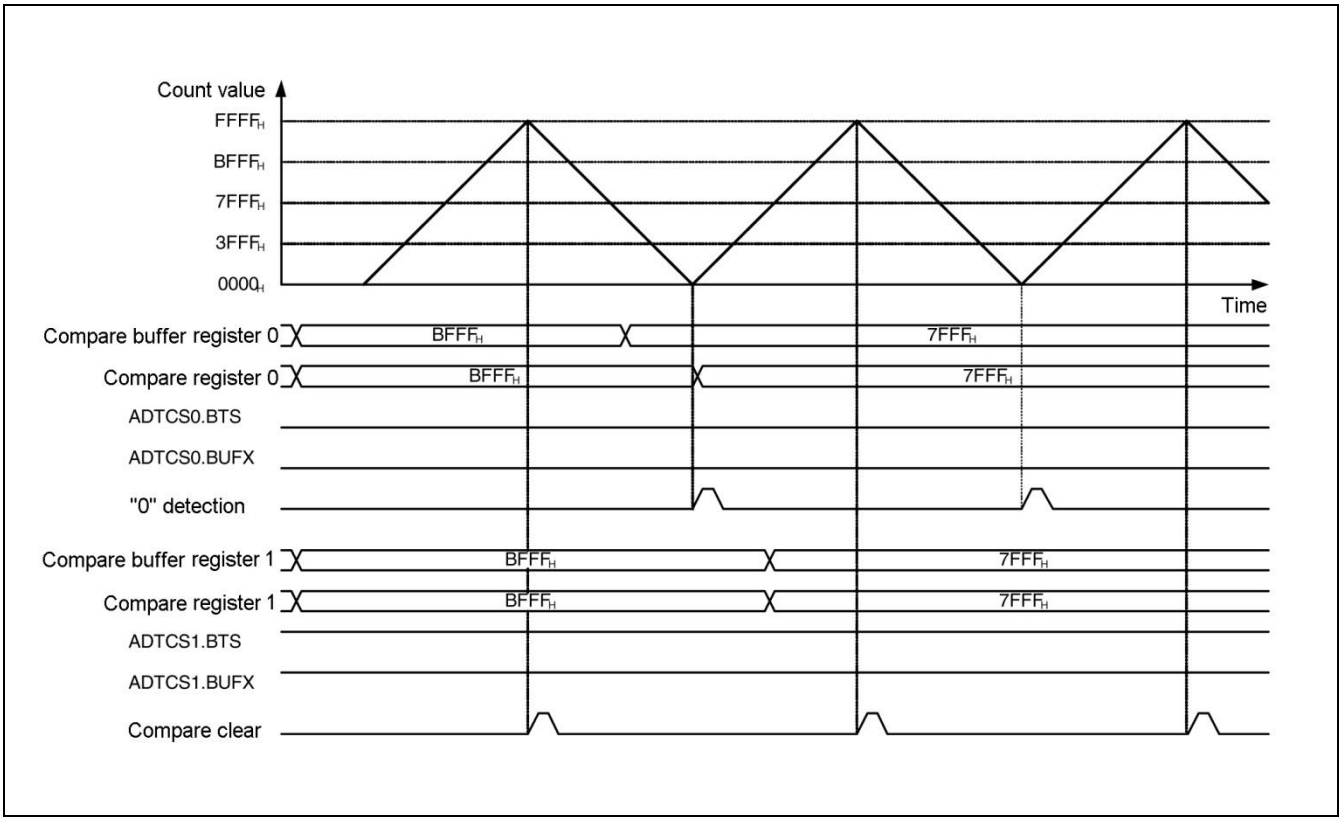


Figure 5-8 Data Transfer Timing for the Compare Register When Compare Matches Occur in 32-Bit Free-Run Timer Up-Counting

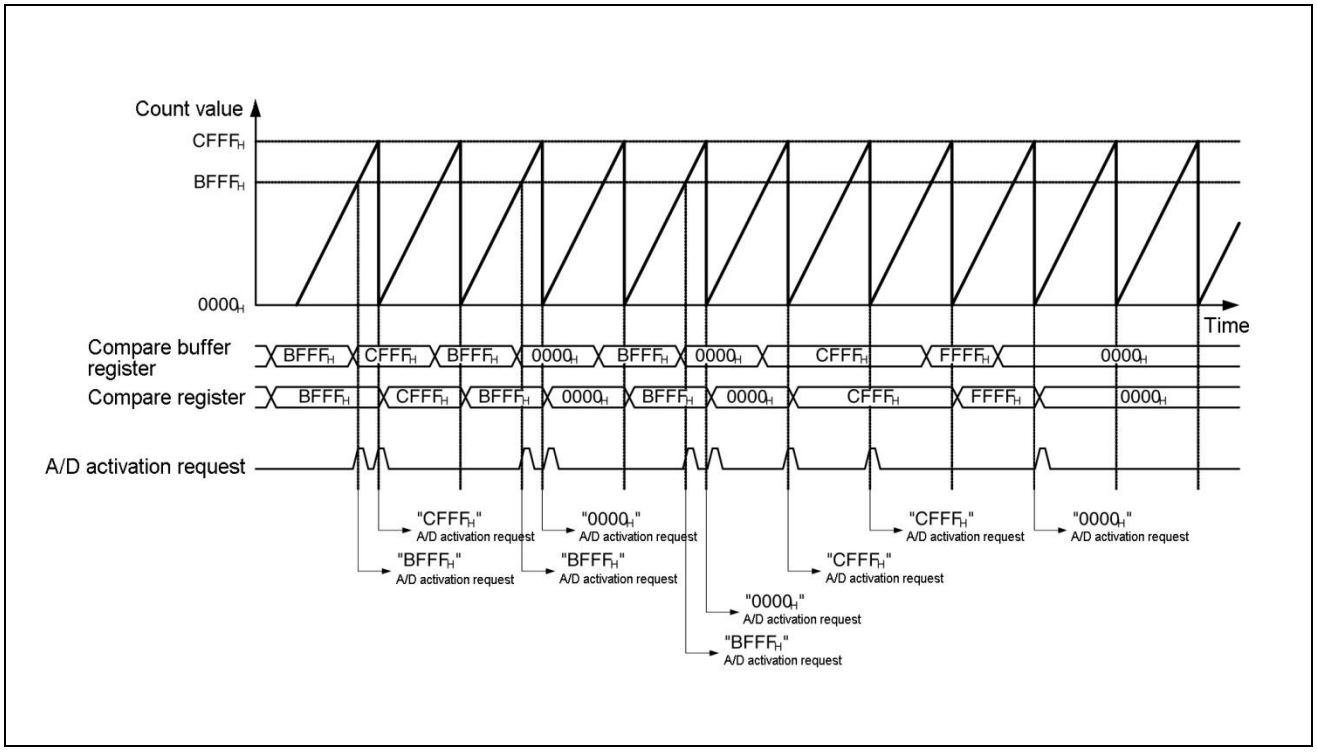


Figure 5-9 Data Transfer Timing for the Compare Register When "0" Is Detected in 32-Bit Free-Run Timer Up-Counting

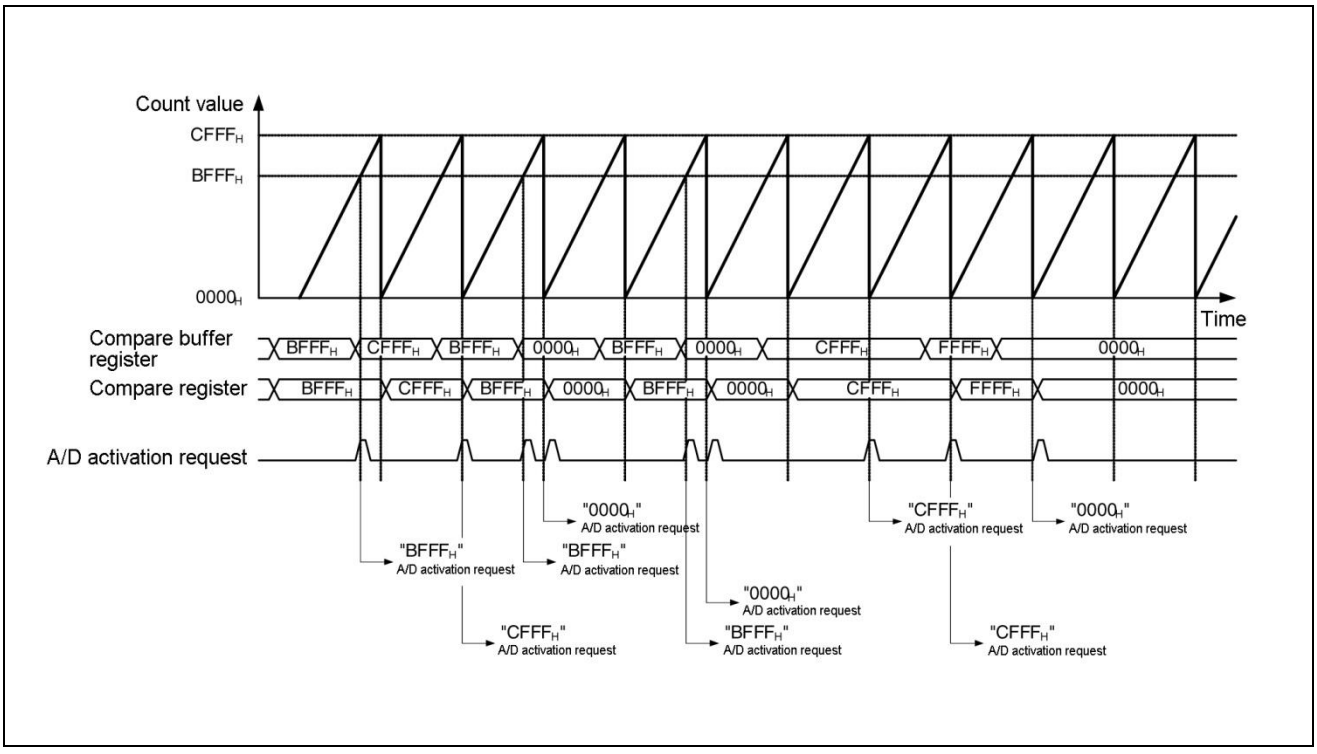


Figure 5-10 Data Transfer Timing for the Compare Register When Compare Matches Occur in 32-Bit Free-Run Timer UP/Down-Counting

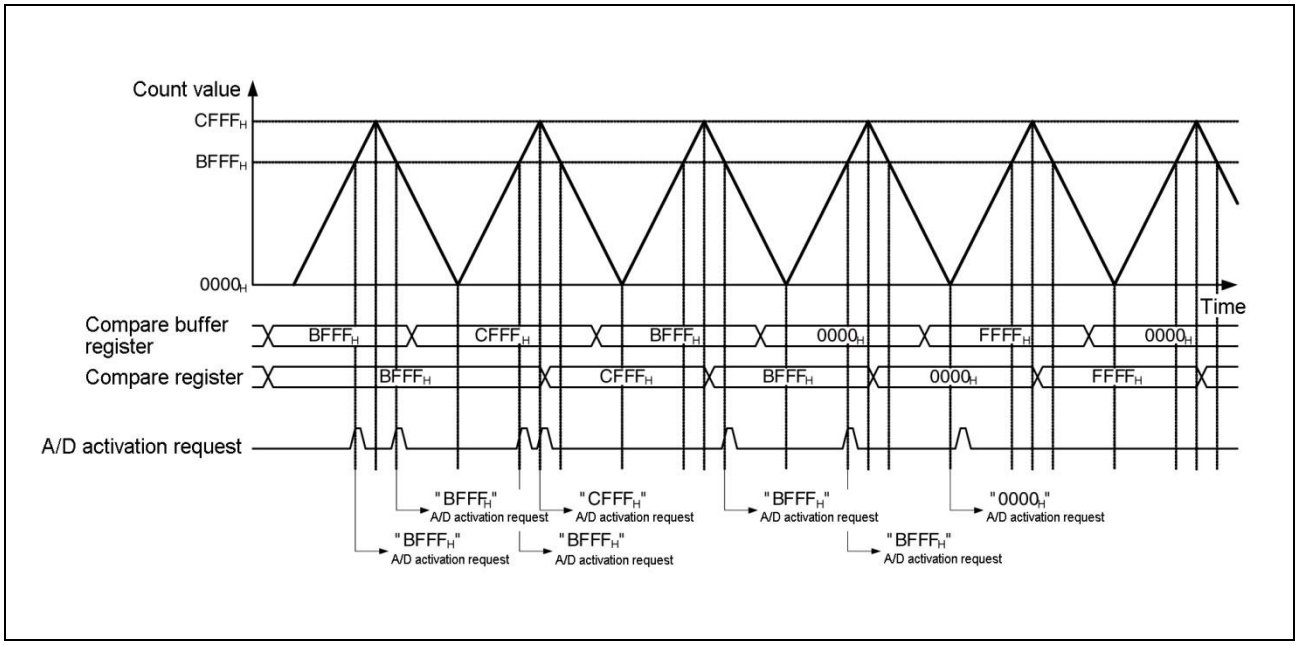
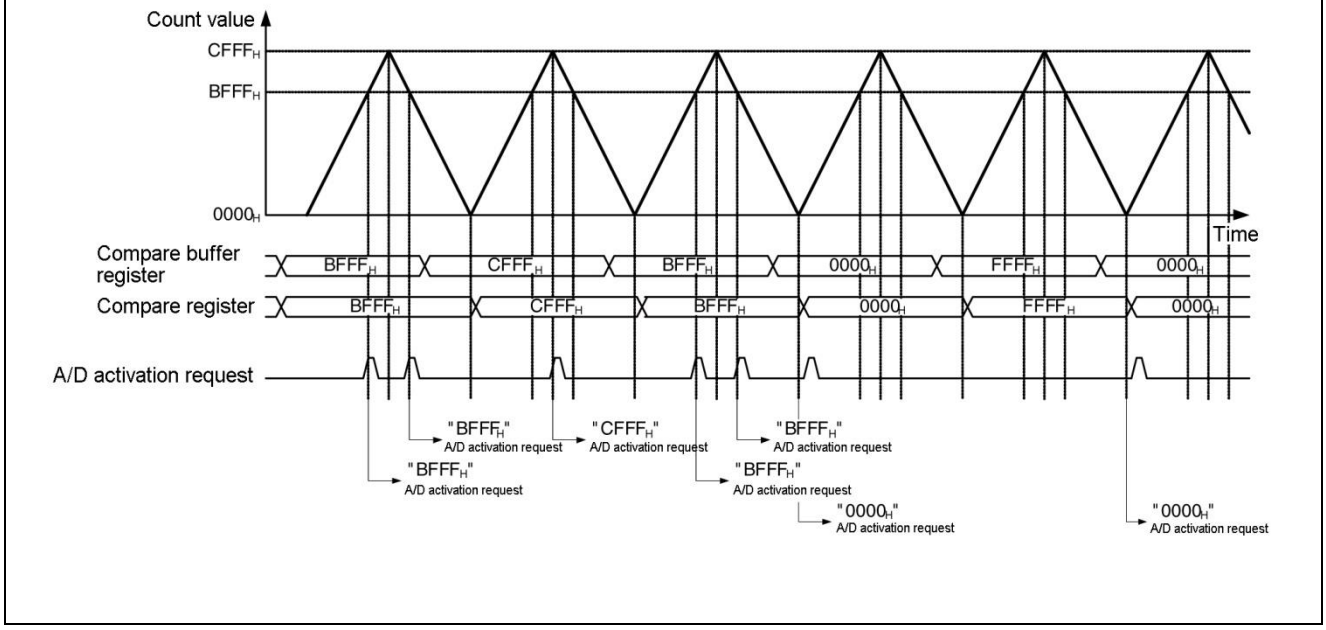


Figure 5-11 Data Transfer Timing for the Compare Register When "0" Is Detected in 32-Bit Free-Run Timer UP/Down-Counting



5.2.8. PWM Timer Activation

This section explains PWM timer activation.

Set the A/D activation trigger control factor selection bits to PWM timer activation (ADTECS.STS2="1", ADTCS.STS1, STS0="00_B").

The PWM timer can be selected as one of the functions of the base timer.
The PWM timer (base timer) activation trigger is input to each 12-bit A/D converter unit one by one. The correspondence between the base timers and 12-bit A/D converter units is as follows.

- Base timer channel 0 corresponds to A/D converter unit 0.
- Base timer channel 2 corresponds to A/D converter unit 1.

When using an activation trigger different PWM timer , make an appropriate setting using the PWM timer selector.
The PWM timer to be used is selected by the RESSEL bit and the PORTSEL bit in the resource input setting register (RIC_RESIN), and is input to each activation channel.
For details of RIC_RESIN, see "Resource Input Setting Register" in "CHAPTER: I/O PORT."
RIC_RESINs as many as the number of the activation channels are implemented.
For relationship among RIC_RESIN numbers, PWM timer numbers, and activation channels, see table 5-12.
For information on the correspondence between each A/D converter unit of this series and activation channels, see "5.2.1.A/D Activation."

The PWM timer generates an A/D activation trigger when the 16-bit down counter and the ADTR register are compared and match (H active).
For an output timing chart, see "PWM Timer Function" in "CHAPTER: BASE TIMER."

When a rising of the PWM timer is detected, the PWM timer activation request signal is set.

Table 5-12 Connection Relationship between PWM Timers and A/D Converter Activation Channels [bit3:0]PORTSEL: Resource Selection bits , [bit3:0]RESSEL: Resource Selection bits

Reg Add (Offset)	Resource	RESSEL Data[3:0] Select Basetimer ADTO signal ch				PORTSEL Data[3:0]
RIC_RESIN 228 (0x01C8)	A/D activation channel 3	0x0 BT00 ADTO	0x1 BT01 ADTO	0x2 BT02 ADTO	0x3 BT03 ADTO	0x8
		0x4 BT04 ADTO	0x5 BT05 ADTO	0x6 BT06 ADTO	0x7 BT07 ADTO	
		0x8 BT08 ADTO	0x9 BT09 ADTO	0xA BT10 ADTO	0xB BT11 ADTO	
		0xC BT12 ADTO	0xD BT13 ADTO	0xE BT14 ADTO	0xF BT15 ADTO	
RIC_RESIN 229 (0x01CA)		0x0 BT16 ADTO	0x1 BT17 ADTO	0x2 BT18 ADTO	0x3 BT19 ADTO	0x9
		0x4 BT20 ADTO	0x5 BT21 ADTO	0x6 BT22 ADTO	0x7 BT23 ADTO	
		0x8 BT24 ADTO	0x9 BT25 ADTO	0xA BT26 ADTO	0xB BT27 ADTO	
		0xC BT28 ADTO	0xD BT29 ADTO	0xE Setting prohibited	0xF Setting prohibited	
RIC_RESIN 232 (0x01D0)	A/D activation channel 5	0x0 BT00 ADTO	0x1 BT01 ADTO	0x2 BT02 ADTO	0x3 BT03 ADTO	0x8
		0x4 BT04 ADTO	0x5 BT05 ADTO	0x6 BT06 ADTO	0x7 BT07 ADTO	
		0x8 BT08 ADTO	0x9 BT09 ADTO	0xA BT10 ADTO	0xB BT11 ADTO	
		0xC BT12 ADTO	0xD BT13 ADTO	0xE BT14 ADTO	0xF BT15 ADTO	
RIC_RESIN 233 (0x01D2)		0x0 BT16 ADTO	0x1 BT17 ADTO	0x2 BT18 ADTO	0x3 BT19 ADTO	0x9
		0x4 BT20 ADTO	0x5 BT21 ADTO	0x6 BT22 ADTO	0x7 BT23 ADTO	
		0x8 BT24 ADTO	0x9 BT25 ADTO	0xA BT26 ADTO	0xB BT27 ADTO	
		0xC BT28 ADTO	0xD BT29 ADTO	0xE Setting prohibited	0xF Setting prohibited	
RIC_RESIN 234 (0x01D4)	A/D activation channel 6	0x0 BT00 ADTO	0x1 BT01 ADTO	0x2 BT02 ADTO	0x3 BT03 ADTO	0x8
		0x4 BT04 ADTO	0x5 BT05 ADTO	0x6 BT06 ADTO	0x7 BT07 ADTO	
		0x8 BT08 ADTO	0x9 BT09 ADTO	0xA BT10 ADTO	0xB BT11 ADTO	
		0xC BT12 ADTO	0xD BT13 ADTO	0xE BT14 ADTO	0xF BT15 ADTO	

Reg Add (Offset)	Resource	RESSEL Data[3:0] Select Basetimer ADTO signal ch				PORTSEL Data[3:0]
RIC_RESIN 235 (0x01D6)	A/D activation channel 6	0x0 BT16 ADTO	0x1 BT17 ADTO	0x2 BT18 ADTO	0x3 BT19 ADTO	0x9
		0x4 BT20 ADTO	0x5 BT21 ADTO	0x6 BT22 ADTO	0x7 BT23 ADTO	
		0x8 BT24 ADTO	0x9 BT25 ADTO	0xA BT26 ADTO	0xB BT27 ADTO	
		0xC BT28 ADTO	0xD BT29 ADTO	0xE Setting prohibited	0xF Setting prohibited	
RIC_RESIN 240 (0x01E0)	A/D activation channel 9	0x0 BT00 ADTO	0x1 BT01 ADTO	0x2 BT02 ADTO	0x3 BT03 ADTO	0x8
		0x4 BT04 ADTO	0x5 BT05 ADTO	0x6 BT06 ADTO	0x7 BT07 ADTO	
		0x8 BT08 ADTO	0x9 BT09 ADTO	0xA BT10 ADTO	0xB BT11 ADTO	
		0xC BT12 ADTO	0xD BT13 ADTO	0xE BT14 ADTO	0xF BT15 ADTO	
RIC_RESIN 241 (0x01E2)		0x0 BT16 ADTO	0x1 BT17 ADTO	0x2 BT18 ADTO	0x3 BT19 ADTO	0x9
		0x4 BT20 ADTO	0x5 BT21 ADTO	0x6 BT22 ADTO	0x7 BT23 ADTO	
		0x8 BT24 ADTO	0x9 BT25 ADTO	0xA BT26 ADTO	0xB BT27 ADTO	
		0xC BT28 ADTO	0xD BT29 ADTO	0xE Setting prohibited	0xF Setting prohibited	
RIC_RESIN 242 (0x01E4)	A/D activation channel 10	0x0 BT00 ADTO	0x1 BT01 ADTO	0x2 BT02 ADTO	0x3 BT03 ADTO	0x8
		0x4 BT04 ADTO	0x5 BT05 ADTO	0x6 BT06 ADTO	0x7 BT07 ADTO	
		0x8 BT08 ADTO	0x9 BT09 ADTO	0xA BT10 ADTO	0xB BT11 ADTO	
		0xC BT12 ADTO	0xD BT13 ADTO	0xE BT14 ADTO	0xF BT15 ADTO	
RIC_RESIN 243 (0x01E6)		0x0 BT16 ADTO	0x1 BT17 ADTO	0x2 BT18 ADTO	0x3 BT19 ADTO	0x9
		0x4 BT20 ADTO	0x5 BT21 ADTO	0x6 BT22 ADTO	0x7 BT23 ADTO	
		0x8 BT24 ADTO	0x9 BT25 ADTO	0xA BT26 ADTO	0xB BT27 ADTO	
		0xC BT28 ADTO	0xD BT29 ADTO	0xE Setting prohibited	0xF Setting prohibited	

Reg Add (Offset)	Resource	RESSEL Data[3:0] Select Basetimer ADTO signal ch				PORTSEL Data[3:0]
RIC_RESIN 246 (0x01EC)	A/D activation channel 12	0x0 BT00 ADTO	0x1 BT01 ADTO	0x2 BT02 ADTO	0x3 BT03 ADTO	0x8
		0x4 BT04 ADTO	0x5 BT05 ADTO	0x6 BT06 ADTO	0x7 BT07 ADTO	
		0x8 BT08 ADTO	0x9 BT09 ADTO	0xA BT10 ADTO	0xB BT11 ADTO	
		0xC BT12 ADTO	0xD BT13 ADTO	0xE BT14 ADTO	0xF BT15 ADTO	
RIC_RESIN 247 (0x01EE)		0x0 BT16 ADTO	0x1 BT17 ADTO	0x2 BT18 ADTO	0x3 BT19 ADTO	0x9
		0x4 BT20 ADTO	0x5 BT21 ADTO	0x6 BT22 ADTO	0x7 BT23 ADTO	
		0x8 BT24 ADTO	0x9 BT25 ADTO	0xA BT26 ADTO	0xB BT27 ADTO	
		0xC BT28 ADTO	0xD BT29 ADTO	0xE Setting prohibited	0xF Setting prohibited	
RIC_RESIN 248 (0x01F0)	A/D activation channel 13	0x0 BT00 ADTO	0x1 BT01 ADTO	0x2 BT02 ADTO	0x3 BT03 ADTO	0x8
		0x4 BT04 ADTO	0x5 BT05 ADTO	0x6 BT06 ADTO	0x7 BT07 ADTO	
		0x8 BT08 ADTO	0x9 BT09 ADTO	0xA BT10 ADTO	0xB BT11 ADTO	
		0xC BT12 ADTO	0xD BT13 ADTO	0xE BT14 ADTO	0xF BT15 ADTO	
RIC_RESIN 249 (0x01F2)		0x0 BT16 ADTO	0x1 BT17 ADTO	0x2 BT18 ADTO	0x3 BT19 ADTO	0x9
		0x4 BT20 ADTO	0x5 BT21 ADTO	0x6 BT22 ADTO	0x7 BT23 ADTO	
		0x8 BT24 ADTO	0x9 BT25 ADTO	0xA BT26 ADTO	0xB BT27 ADTO	
		0xC BT28 ADTO	0xD BT29 ADTO	0xE Setting prohibited	0xF Setting prohibited	
RIC_RESIN 250 (0x01F4)	A/D activation channel 14	0x0 BT00 ADTO	0x1 BT01 ADTO	0x2 BT02 ADTO	0x3 BT03 ADTO	0x8
		0x4 BT04 ADTO	0x5 BT05 ADTO	0x6 BT06 ADTO	0x7 BT07 ADTO	
		0x8 BT08 ADTO	0x9 BT09 ADTO	0xA BT10 ADTO	0xB BT11 ADTO	
		0xC BT12 ADTO	0xD BT13 ADTO	0xE BT14 ADTO	0xF BT15 ADTO	

Reg Add (Offset)	Resource	RESSEL Data[3:0] Select Basetimer ADTO signal ch				PORTSEL Data[3:0]
RIC_RESIN 251 (0x01F6)	A/D activation channel 14	0x0 BT16 ADTO	0x1 BT17 ADTO	0x2 BT18 ADTO	0x3 BT19 ADTO	0x9
		0x4 BT20 ADTO	0x5 BT21 ADTO	0x6 BT22 ADTO	0x7 BT23 ADTO	
		0x8 BT24 ADTO	0x9 BT25 ADTO	0xA BT26 ADTO	0xB BT27 ADTO	
		0xC BT28 ADTO	0xD BT29 ADTO	0xE Setting prohibited	0xF Setting prohibited	
RIC_RESIN 252 (0x01F8)	A/D activation channel 15	0x0 BT00 ADTO	0x1 BT01 ADTO	0x2 BT02 ADTO	0x3 BT03 ADTO	0x8
		0x4 BT04 ADTO	0x5 BT05 ADTO	0x6 BT06 ADTO	0x7 BT07 ADTO	
		0x8 BT08 ADTO	0x9 BT09 ADTO	0xA BT10 ADTO	0xB BT11 ADTO	
		0xC BT12 ADTO	0xD BT13 ADTO	0xE BT14 ADTO	0xF BT15 ADTO	
RIC_RESIN 253 (0x01FA)		0x0 BT16 ADTO	0x1 BT17 ADTO	0x2 BT18 ADTO	0x3 BT19 ADTO	0x9
		0x4 BT20 ADTO	0x5 BT21 ADTO	0x6 BT22 ADTO	0x7 BT23 ADTO	
		0x8 BT24 ADTO	0x9 BT25 ADTO	0xA BT26 ADTO	0xB BT27 ADTO	
		0xC BT28 ADTO	0xD BT29 ADTO	0xE Setting prohibited	0xF Setting prohibited	
RIC_RESIN 256 (0x0200)	A/D activation channel 17	0x0 BT00 ADTO	0x1 BT01 ADTO	0x2 BT02 ADTO	0x3 BT03 ADTO	0x8
		0x4 BT04 ADTO	0x5 BT05 ADTO	0x6 BT06 ADTO	0x7 BT07 ADTO	
		0x8 BT08 ADTO	0x9 BT09 ADTO	0xA BT10 ADTO	0xB BT11 ADTO	
		0xC BT12 ADTO	0xD BT13 ADTO	0xE BT14 ADTO	0xF BT15 ADTO	
RIC_RESIN 257 (0x0202)		0x0 BT16 ADTO	0x1 BT17 ADTO	0x2 BT18 ADTO	0x3 BT19 ADTO	0x9
		0x4 BT20 ADTO	0x5 BT21 ADTO	0x6 BT22 ADTO	0x7 BT23 ADTO	
		0x8 BT24 ADTO	0x9 BT25 ADTO	0xA BT26 ADTO	0xB BT27 ADTO	
		0xC BT28 ADTO	0xD BT29 ADTO	0xE Setting prohibited	0xF Setting prohibited	

Reg Add (Offset)	Resource	RESSEL Data[3:0] Select Basetimer ADTO signal ch				PORTSEL Data[3:0]
RIC_RESIN 258 (0x0204)	A/D activation channel 18	0x0 BT00 ADTO	0x1 BT01 ADTO	0x2 BT02 ADTO	0x3 BT03 ADTO	0x8
		0x4 BT04 ADTO	0x5 BT05 ADTO	0x6 BT06 ADTO	0x7 BT07 ADTO	
		0x8 BT08 ADTO	0x9 BT09 ADTO	0xA BT10 ADTO	0xB BT11 ADTO	
		0xC BT12 ADTO	0xD BT13 ADTO	0xE BT14 ADTO	0xF BT15 ADTO	
RIC_RESIN 259 (0x0206)		0x0 BT16 ADTO	0x1 BT17 ADTO	0x2 BT18 ADTO	0x3 BT19 ADTO	0x9
		0x4 BT20 ADTO	0x5 BT21 ADTO	0x6 BT22 ADTO	0x7 BT23 ADTO	
		0x8 BT24 ADTO	0x9 BT25 ADTO	0xA BT26 ADTO	0xB BT27 ADTO	
		0xC BT28 ADTO	0xD BT29 ADTO	0xE Setting prohibited	0xF Setting prohibited	
RIC_RESIN 260 (0x0208)	A/D activation channel 19	0x0 BT00 ADTO	0x1 BT01 ADTO	0x2 BT02 ADTO	0x3 BT03 ADTO	0x8
		0x4 BT04 ADTO	0x5 BT05 ADTO	0x6 BT06 ADTO	0x7 BT07 ADTO	
		0x8 BT08 ADTO	0x9 BT09 ADTO	0xA BT10 ADTO	0xB BT11 ADTO	
		0xC BT12 ADTO	0xD BT13 ADTO	0xE BT14 ADTO	0xF BT15 ADTO	
RIC_RESIN 261 (0x020A)		0x0 BT16 ADTO	0x1 BT17 ADTO	0x2 BT18 ADTO	0x3 BT19 ADTO	0x9
		0x4 BT20 ADTO	0x5 BT21 ADTO	0x6 BT22 ADTO	0x7 BT23 ADTO	
		0x8 BT24 ADTO	0x9 BT25 ADTO	0xA BT26 ADTO	0xB BT27 ADTO	
		0xC BT28 ADTO	0xD BT29 ADTO	0xE Setting prohibited	0xF Setting prohibited	
RIC_RESIN 262 (0x020C)	A/D activation channel 20	0x0 BT00 ADTO	0x1 BT01 ADTO	0x2 BT02 ADTO	0x3 BT03 ADTO	0x8
		0x4 BT04 ADTO	0x5 BT05 ADTO	0x6 BT06 ADTO	0x7 BT07 ADTO	
		0x8 BT08 ADTO	0x9 BT09 ADTO	0xA BT10 ADTO	0xB BT11 ADTO	
		0xC BT12 ADTO	0xD BT13 ADTO	0xE BT14 ADTO	0xF BT15 ADTO	

Reg Add (Offset)	Resource	RESSEL Data[3:0] Select Basetimer ADTO signal ch				PORTSEL Data[3:0]
RIC_RESIN 263 (0x020E)	A/D activation channel 20	0x0 BT16 ADTO	0x1 BT17 ADTO	0x2 BT18 ADTO	0x3 BT19 ADTO	0x9
		0x4 BT20 ADTO	0x5 BT21 ADTO	0x6 BT22 ADTO	0x7 BT23 ADTO	
		0x8 BT24 ADTO	0x9 BT25 ADTO	0xA BT26 ADTO	0xB BT27 ADTO	
		0xC BT28 ADTO	0xD BT29 ADTO	0xE Setting prohibited	0xF Setting prohibited	
RIC_RESIN 264 (0x0210)	A/D activation channel 21	0x0 BT00 ADTO	0x1 BT01 ADTO	0x2 BT02 ADTO	0x3 BT03 ADTO	0x8
		0x4 BT04 ADTO	0x5 BT05 ADTO	0x6 BT06 ADTO	0x7 BT07 ADTO	
		0x8 BT08 ADTO	0x9 BT09 ADTO	0xA BT10 ADTO	0xB BT11 ADTO	
		0xC BT12 ADTO	0xD BT13 ADTO	0xE BT14 ADTO	0xF BT15 ADTO	
RIC_RESIN 265 (0x0212)		0x0 BT16 ADTO	0x1 BT17 ADTO	0x2 BT18 ADTO	0x3 BT19 ADTO	0x9
		0x4 BT20 ADTO	0x5 BT21 ADTO	0x6 BT22 ADTO	0x7 BT23 ADTO	
		0x8 BT24 ADTO	0x9 BT25 ADTO	0xA BT26 ADTO	0xB BT27 ADTO	
		0xC BT28 ADTO	0xD BT29 ADTO	0xE Setting prohibited	0xF Setting prohibited	
RIC_RESIN 266 (0x0214)	A/D activation channel 22	0x0 BT00 ADTO	0x1 BT01 ADTO	0x2 BT02 ADTO	0x3 BT03 ADTO	0x8
		0x4 BT04 ADTO	0x5 BT05 ADTO	0x6 BT06 ADTO	0x7 BT07 ADTO	
		0x8 BT08 ADTO	0x9 BT09 ADTO	0xA BT10 ADTO	0xB BT11 ADTO	
		0xC BT12 ADTO	0xD BT13 ADTO	0xE BT14 ADTO	0xF BT15 ADTO	
RIC_RESIN 267 (0x0216)		0x0 BT16 ADTO	0x1 BT17 ADTO	0x2 BT18 ADTO	0x3 BT19 ADTO	
		0x4 BT20 ADTO	0x5 BT21 ADTO	0x6 BT22 ADTO	0x7 BT23 ADTO	
		0x8 BT24 ADTO	0x9 BT25 ADTO	0xA BT26 ADTO	0xB BT27 ADTO	
		0xC BT28 ADTO	0xD BT29 ADTO	0xE Setting prohibited	0xF Setting prohibited	

Reg Add (Offset)	Resource	RESSEL Data[3:0] Select Basetimer ADTO signal ch				PORTSEL Data[3:0]
RIC_RESIN 268 (0x0218)	A/D activation channel 23	0x0 BT00 ADTO	0x1 BT01 ADTO	0x2 BT02 ADTO	0x3 BT03 ADTO	0x8
		0x4 BT04 ADTO	0x5 BT05 ADTO	0x6 BT06 ADTO	0x7 BT07 ADTO	
		0x8 BT08 ADTO	0x9 BT09 ADTO	0xA BT10 ADTO	0xB BT11 ADTO	
		0xC BT12 ADTO	0xD BT13 ADTO	0xE BT14 ADTO	0xF BT15 ADTO	
RIC_RESIN 269 (0x021A)		0x0 BT16 ADTO	0x1 BT17 ADTO	0x2 BT18 ADTO	0x3 BT19 ADTO	0x9
		0x4 BT20 ADTO	0x5 BT21 ADTO	0x6 BT22 ADTO	0x7 BT23 ADTO	
		0x8 BT24 ADTO	0x9 BT25 ADTO	0xA BT26 ADTO	0xB BT27 ADTO	
		0xC BT28 ADTO	0xD BT29 ADTO	0xE Setting prohibited	0xF Setting prohibited	
RIC_RESIN 270 (0x021C)	A/D activation channel 24	0x0 BT00 ADTO	0x1 BT01 ADTO	0x2 BT02 ADTO	0x3 BT03 ADTO	0x8
		0x4 BT04 ADTO	0x5 BT05 ADTO	0x6 BT06 ADTO	0x7 BT07 ADTO	
		0x8 BT08 ADTO	0x9 BT09 ADTO	0xA BT10 ADTO	0xB BT11 ADTO	
		0xC BT12 ADTO	0xD BT13 ADTO	0xE BT14 ADTO	0xF BT15 ADTO	
RIC_RESIN 271 (0x021E)		0x0 BT16 ADTO	0x1 BT17 ADTO	0x2 BT18 ADTO	0x3 BT19 ADTO	0x9
		0x4 BT20 ADTO	0x5 BT21 ADTO	0x6 BT22 ADTO	0x7 BT23 ADTO	
		0x8 BT24 ADTO	0x9 BT25 ADTO	0xA BT26 ADTO	0xB BT27 ADTO	
		0xC BT28 ADTO	0xD BT29 ADTO	0xE Setting prohibited	0xF Setting prohibited	
RIC_RESIN 276 (0x0228)	A/D activation channel 27	0x0 BT00 ADTO	0x1 BT01 ADTO	0x2 BT02 ADTO	0x3 BT03 ADTO	0x8
		0x4 BT04 ADTO	0x5 BT05 ADTO	0x6 BT06 ADTO	0x7 BT07 ADTO	
		0x8 BT08 ADTO	0x9 BT09 ADTO	0xA BT10 ADTO	0xB BT11 ADTO	
		0xC BT12 ADTO	0xD BT13 ADTO	0xE BT14 ADTO	0xF BT15 ADTO	

Reg Add (Offset)	Resource	RESSEL Data[3:0] Select Basetimer ADTO signal ch				PORTSEL Data[3:0]
RIC_RESIN 277 (0x022A)	A/D activation channel 27	0x0 BT16 ADTO	0x1 BT17 ADTO	0x2 BT18 ADTO	0x3 BT19 ADTO	0x9
		0x4 BT20 ADTO	0x5 BT21 ADTO	0x6 BT22 ADTO	0x7 BT23 ADTO	
		0x8 BT24 ADTO	0x9 BT25 ADTO	0xA BT26 ADTO	0xB BT27 ADTO	
		0xC BT28 ADTO	0xD BT29 ADTO	0xE Setting prohibited	0xF Setting prohibited	
RIC_RESIN 278 (0x022C)	A/D activation channel 28	0x0 BT00 ADTO	0x1 BT01 ADTO	0x2 BT02 ADTO	0x3 BT03 ADTO	0x8
		0x4 BT04 ADTO	0x5 BT05 ADTO	0x6 BT06 ADTO	0x7 BT07 ADTO	
		0x8 BT08 ADTO	0x9 BT09 ADTO	0xA BT10 ADTO	0xB BT11 ADTO	
		0xC BT12 ADTO	0xD BT13 ADTO	0xE BT14 ADTO	0xF BT15 ADTO	
RIC_RESIN 279 (0x022E)		0x0 BT16 ADTO	0x1 BT17 ADTO	0x2 BT18 ADTO	0x3 BT19 ADTO	0x9
		0x4 BT20 ADTO	0x5 BT21 ADTO	0x6 BT22 ADTO	0x7 BT23 ADTO	
		0x8 BT24 ADTO	0x9 BT25 ADTO	0xA BT26 ADTO	0xB BT27 ADTO	
		0xC BT28 ADTO	0xD BT29 ADTO	0xE Setting prohibited	0xF Setting prohibited	
RIC_RESIN 280 (0x0230)	A/D activation channel 29	0x0 BT00 ADTO	0x1 BT01 ADTO	0x2 BT02 ADTO	0x3 BT03 ADTO	0x8
		0x4 BT04 ADTO	0x5 BT05 ADTO	0x6 BT06 ADTO	0x7 BT07 ADTO	
		0x8 BT08 ADTO	0x9 BT09 ADTO	0xA BT10 ADTO	0xB BT11 ADTO	
		0xC BT12 ADTO	0xD BT13 ADTO	0xE BT14 ADTO	0xF BT15 ADTO	
RIC_RESIN 281 (0x0232)		0x0 BT16 ADTO	0x1 BT17 ADTO	0x2 BT18 ADTO	0x3 BT19 ADTO	0x9
		0x4 BT20 ADTO	0x5 BT21 ADTO	0x6 BT22 ADTO	0x7 BT23 ADTO	
		0x8 BT24 ADTO	0x9 BT25 ADTO	0xA BT26 ADTO	0xB BT27 ADTO	
		0xC BT28 ADTO	0xD BT29 ADTO	0xE Setting prohibited	0xF Setting prohibited	

Reg Add (Offset)	Resource	RESSEL Data[3:0] Select Basetimer ADTO signal ch				PORTSEL Data[3:0]
RIC_RESIN 282 (0x0234)	A/D activation channel 30	0x0 BT00 ADTO	0x1 BT01 ADTO	0x2 BT02 ADTO	0x3 BT03 ADTO	0x8
		0x4 BT04 ADTO	0x5 BT05 ADTO	0x6 BT06 ADTO	0x7 BT07 ADTO	
		0x8 BT08 ADTO	0x9 BT09 ADTO	0xA BT10 ADTO	0xB BT11 ADTO	
		0xC BT12 ADTO	0xD BT13 ADTO	0xE BT14 ADTO	0xF BT15 ADTO	
RIC_RESIN 283 (0x0236)		0x0 BT16 ADTO	0x1 BT17 ADTO	0x2 BT18 ADTO	0x3 BT19 ADTO	0x9
		0x4 BT20 ADTO	0x5 BT21 ADTO	0x6 BT22 ADTO	0x7 BT23 ADTO	
		0x8 BT24 ADTO	0x9 BT25 ADTO	0xA BT26 ADTO	0xB BT27 ADTO	
		0xC BT28 ADTO	0xD BT29 ADTO	0xE Setting prohibited	0xF Setting prohibited	
RIC_RESIN 284 (0x0238)	A/D activation channel 31	0x0 BT00 ADTO	0x1 BT01 ADTO	0x2 BT02 ADTO	0x3 BT03 ADTO	0x8
		0x4 BT04 ADTO	0x5 BT05 ADTO	0x6 BT06 ADTO	0x7 BT07 ADTO	
		0x8 BT08 ADTO	0x9 BT09 ADTO	0xA BT10 ADTO	0xB BT11 ADTO	
		0xC BT12 ADTO	0xD BT13 ADTO	0xE BT14 ADTO	0xF BT15 ADTO	
RIC_RESIN 285 (0x023A)		0x0 BT16 ADTO	0x1 BT17 ADTO	0x2 BT18 ADTO	0x3 BT19 ADTO	0x9
		0x4 BT20 ADTO	0x5 BT21 ADTO	0x6 BT22 ADTO	0x7 BT23 ADTO	
		0x8 BT24 ADTO	0x9 BT25 ADTO	0xA BT26 ADTO	0xB BT27 ADTO	
		0xC BT28 ADTO	0xD BT29 ADTO	0xE Setting prohibited	0xF Setting prohibited	
RIC_RESIN 286 (0x023C)	A/D activation channel 32	0x0 BT00 ADTO	0x1 BT01 ADTO	0x2 BT02 ADTO	0x3 BT03 ADTO	0x8
		0x4 BT04 ADTO	0x5 BT05 ADTO	0x6 BT06 ADTO	0x7 BT07 ADTO	
		0x8 BT08 ADTO	0x9 BT09 ADTO	0xA BT10 ADTO	0xB BT11 ADTO	
		0xC BT12 ADTO	0xD BT13 ADTO	0xE BT14 ADTO	0xF BT15 ADTO	

Reg Add (Offset)	Resource	RESSEL Data[3:0] Select Basetimer ADTO signal ch				PORTSEL Data[3:0]
RIC_RESIN 287 (0x023E)	A/D activation channel 32	0x0 BT16 ADTO	0x1 BT17 ADTO	0x2 BT18 ADTO	0x3 BT19 ADTO	0x9
		0x4 BT20 ADTO	0x5 BT21 ADTO	0x6 BT22 ADTO	0x7 BT23 ADTO	
		0x8 BT24 ADTO	0x9 BT25 ADTO	0xA BT26 ADTO	0xB BT27 ADTO	
		0xC BT28 ADTO	0xD BT29 ADTO	0xE Setting prohibited	0xF Setting prohibited	
RIC_RESIN 288 (0x0240)	A/D activation channel 33	0x0 BT00 ADTO	0x1 BT01 ADTO	0x2 BT02 ADTO	0x3 BT03 ADTO	0x8
		0x4 BT04 ADTO	0x5 BT05 ADTO	0x6 BT06 ADTO	0x7 BT07 ADTO	
		0x8 BT08 ADTO	0x9 BT09 ADTO	0xA BT10 ADTO	0xB BT11 ADTO	
		0xC BT12 ADTO	0xD BT13 ADTO	0xE BT14 ADTO	0xF BT15 ADTO	
RIC_RESIN 289 (0x0242)		0x0 BT16 ADTO	0x1 BT17 ADTO	0x2 BT18 ADTO	0x3 BT19 ADTO	0x9
		0x4 BT20 ADTO	0x5 BT21 ADTO	0x6 BT22 ADTO	0x7 BT23 ADTO	
		0x8 BT24 ADTO	0x9 BT25 ADTO	0xA BT26 ADTO	0xB BT27 ADTO	
		0xC BT28 ADTO	0xD BT29 ADTO	0xE Setting prohibited	0xF Setting prohibited	
RIC_RESIN 290 (0x0244)	A/D activation channel 34	0x0 BT00 ADTO	0x1 BT01 ADTO	0x2 BT02 ADTO	0x3 BT03 ADTO	0x8
		0x4 BT04 ADTO	0x5 BT05 ADTO	0x6 BT06 ADTO	0x7 BT07 ADTO	
		0x8 BT08 ADTO	0x9 BT09 ADTO	0xA BT10 ADTO	0xB BT11 ADTO	
		0xC BT12 ADTO	0xD BT13 ADTO	0xE BT14 ADTO	0xF BT15 ADTO	
RIC_RESIN 291 (0x0246)		0x0 BT16 ADTO	0x1 BT17 ADTO	0x2 BT18 ADTO	0x3 BT19 ADTO	0x9
		0x4 BT20 ADTO	0x5 BT21 ADTO	0x6 BT22 ADTO	0x7 BT23 ADTO	
		0x8 BT24 ADTO	0x9 BT25 ADTO	0xA BT26 ADTO	0xB BT27 ADTO	
		0xC BT28 ADTO	0xD BT29 ADTO	0xE Setting prohibited	0xF Setting prohibited	

Reg Add (Offset)	Resource	RESSEL Data[3:0] Select Basetimer ADTO signal ch				PORTSEL Data[3:0]
RIC_RESIN 292 (0x0248)	A/D activation channel 35	0x0 BT00 ADTO	0x1 BT01 ADTO	0x2 BT02 ADTO	0x3 BT03 ADTO	0x8
		0x4 BT04 ADTO	0x5 BT05 ADTO	0x6 BT06 ADTO	0x7 BT07 ADTO	
		0x8 BT08 ADTO	0x9 BT09 ADTO	0xA BT10 ADTO	0xB BT11 ADTO	
		0xC BT12 ADTO	0xD BT13 ADTO	0xE BT14 ADTO	0xF BT15 ADTO	
RIC_RESIN 293 (0x024A)		0x0 BT16 ADTO	0x1 BT17 ADTO	0x2 BT18 ADTO	0x3 BT19 ADTO	0x9
		0x4 BT20 ADTO	0x5 BT21 ADTO	0x6 BT22 ADTO	0x7 BT23 ADTO	
		0x8 BT24 ADTO	0x9 BT25 ADTO	0xA BT26 ADTO	0xB BT27 ADTO	
		0xC BT28 ADTO	0xD BT29 ADTO	0xE Setting prohibited	0xF Setting prohibited	
RIC_RESIN 294 (0x024C)	A/D activation channel 36	0x0 BT00 ADTO	0x1 BT01 ADTO	0x2 BT02 ADTO	0x3 BT03 ADTO	0x8
		0x4 BT04 ADTO	0x5 BT05 ADTO	0x6 BT06 ADTO	0x7 BT07 ADTO	
		0x8 BT08 ADTO	0x9 BT09 ADTO	0xA BT10 ADTO	0xB BT11 ADTO	
		0xC BT12 ADTO	0xD BT13 ADTO	0xE BT14 ADTO	0xF BT15 ADTO	
RIC_RESIN 295 (0x024E)		0x0 BT16 ADTO	0x1 BT17 ADTO	0x2 BT18 ADTO	0x3 BT19 ADTO	0x9
		0x4 BT20 ADTO	0x5 BT21 ADTO	0x6 BT22 ADTO	0x7 BT23 ADTO	
		0x8 BT24 ADTO	0x9 BT25 ADTO	0xA BT26 ADTO	0xB BT27 ADTO	
		0xC BT28 ADTO	0xD BT29 ADTO	0xE Setting prohibited	0xF Setting prohibited	
RIC_RESIN 296 (0x0250)	A/D activation channel 37	0x0 BT00 ADTO	0x1 BT01 ADTO	0x2 BT02 ADTO	0x3 BT03 ADTO	0x8
		0x4 BT04 ADTO	0x5 BT05 ADTO	0x6 BT06 ADTO	0x7 BT07 ADTO	
		0x8 BT08 ADTO	0x9 BT09 ADTO	0xA BT10 ADTO	0xB BT11 ADTO	
		0xC BT12 ADTO	0xD BT13 ADTO	0xE BT14 ADTO	0xF BT15 ADTO	

Reg Add (Offset)	Resource	RESSEL Data[3:0] Select Basetimer ADTO signal ch				PORTSEL Data[3:0]
RIC_RESIN 297 (0x0252)	A/D activation channel 37	0x0 BT16 ADTO	0x1 BT17 ADTO	0x2 BT18 ADTO	0x3 BT19 ADTO	0x9
		0x4 BT20 ADTO	0x5 BT21 ADTO	0x6 BT22 ADTO	0x7 BT23 ADTO	
		0x8 BT24 ADTO	0x9 BT25 ADTO	0xA BT26 ADTO	0xB BT27 ADTO	
		0xC BT28 ADTO	0xD BT29 ADTO	0xE Setting prohibited	0xF Setting prohibited	
RIC_RESIN 298 (0x0254)	A/D activation channel 38	0x0 BT00 ADTO	0x1 BT01 ADTO	0x2 BT02 ADTO	0x3 BT03 ADTO	0x8
		0x4 BT04 ADTO	0x5 BT05 ADTO	0x6 BT06 ADTO	0x7 BT07 ADTO	
		0x8 BT08 ADTO	0x9 BT09 ADTO	0xA BT10 ADTO	0xB BT11 ADTO	
		0xC BT12 ADTO	0xD BT13 ADTO	0xE BT14 ADTO	0xF BT15 ADTO	
RIC_RESIN 299 (0x0256)		0x0 BT16 ADTO	0x1 BT17 ADTO	0x2 BT18 ADTO	0x3 BT19 ADTO	0x9
		0x4 BT20 ADTO	0x5 BT21 ADTO	0x6 BT22 ADTO	0x7 BT23 ADTO	
		0x8 BT24 ADTO	0x9 BT25 ADTO	0xA BT26 ADTO	0xB BT27 ADTO	
		0xC BT28 ADTO	0xD BT29 ADTO	0xE Setting prohibited	0xF Setting prohibited	
RIC_RESIN 300 (0x0258)	A/D activation channel 39	0x0 BT00 ADTO	0x1 BT01 ADTO	0x2 BT02 ADTO	0x3 BT03 ADTO	0x8
		0x4 BT04 ADTO	0x5 BT05 ADTO	0x6 BT06 ADTO	0x7 BT07 ADTO	
		0x8 BT08 ADTO	0x9 BT09 ADTO	0xA BT10 ADTO	0xB BT11 ADTO	
		0xC BT12 ADTO	0xD BT13 ADTO	0xE BT14 ADTO	0xF BT15 ADTO	
RIC_RESIN 301 (0x025A)		0x0 BT16 ADTO	0x1 BT17 ADTO	0x2 BT18 ADTO	0x3 BT19 ADTO	0x9
		0x4 BT20 ADTO	0x5 BT21 ADTO	0x6 BT22 ADTO	0x7 BT23 ADTO	
		0x8 BT24 ADTO	0x9 BT25 ADTO	0xA BT26 ADTO	0xB BT27 ADTO	
		0xC BT28 ADTO	0xD BT29 ADTO	0xE Setting prohibited	0xF Setting prohibited	

Reg Add (Offset)	Resource	RESSEL Data[3:0] Select Basetimer ADTO signal ch				PORTSEL Data[3:0]
RIC_RESIN 302 (0x025C)	A/D activation channel 40	0x0 BT00 ADTO	0x1 BT01 ADTO	0x2 BT02 ADTO	0x3 BT03 ADTO	0x8
		0x4 BT04 ADTO	0x5 BT05 ADTO	0x6 BT06 ADTO	0x7 BT07 ADTO	
		0x8 BT08 ADTO	0x9 BT09 ADTO	0xA BT10 ADTO	0xB BT11 ADTO	
		0xC BT12 ADTO	0xD BT13 ADTO	0xE BT14 ADTO	0xF BT15 ADTO	
RIC_RESIN 303 (0x025E)		0x0 BT16 ADTO	0x1 BT17 ADTO	0x2 BT18 ADTO	0x3 BT19 ADTO	0x9
		0x4 BT20 ADTO	0x5 BT21 ADTO	0x6 BT22 ADTO	0x7 BT23 ADTO	
		0x8 BT24 ADTO	0x9 BT25 ADTO	0xA BT26 ADTO	0xB BT27 ADTO	
		0xC BT28 ADTO	0xD BT29 ADTO	0xE Setting prohibited	0xF Setting prohibited	
RIC_RESIN 304 (0x0260)	A/D activation channel 41	0x0 BT00 ADTO	0x1 BT01 ADTO	0x2 BT02 ADTO	0x3 BT03 ADTO	0x8
		0x4 BT04 ADTO	0x5 BT05 ADTO	0x6 BT06 ADTO	0x7 BT07 ADTO	
		0x8 BT08 ADTO	0x9 BT09 ADTO	0xA BT10 ADTO	0xB BT11 ADTO	
		0xC BT12 ADTO	0xD BT13 ADTO	0xE BT14 ADTO	0xF BT15 ADTO	
RIC_RESIN 305 (0x0262)		0x0 BT16 ADTO	0x1 BT17 ADTO	0x2 BT18 ADTO	0x3 BT19 ADTO	0x9
		0x4 BT20 ADTO	0x5 BT21 ADTO	0x6 BT22 ADTO	0x7 BT23 ADTO	
		0x8 BT24 ADTO	0x9 BT25 ADTO	0xA BT26 ADTO	0xB BT27 ADTO	
		0xC BT28 ADTO	0xD BT29 ADTO	0xE Setting prohibited	0xF Setting prohibited	
RIC_RESIN 306 (0x0264)	A/D activation channel 42	0x0 BT00 ADTO	0x1 BT01 ADTO	0x2 BT02 ADTO	0x3 BT03 ADTO	0x8
		0x4 BT04 ADTO	0x5 BT05 ADTO	0x6 BT06 ADTO	0x7 BT07 ADTO	
		0x8 BT08 ADTO	0x9 BT09 ADTO	0xA BT10 ADTO	0xB BT11 ADTO	
		0xC BT12 ADTO	0xD BT13 ADTO	0xE BT14 ADTO	0xF BT15 ADTO	

Reg Add (Offset)	Resource	RESSEL Data[3:0] Select Basetimer ADTO signal ch				PORTSEL Data[3:0]
RIC_RESIN 307 (0x0266)	A/D activation channel 42	0x0 BT16 ADTO	0x1 BT17 ADTO	0x2 BT18 ADTO	0x3 BT19 ADTO	0x9
		0x4 BT20 ADTO	0x5 BT21 ADTO	0x6 BT22 ADTO	0x7 BT23 ADTO	
		0x8 BT24 ADTO	0x9 BT25 ADTO	0xA BT26 ADTO	0xB BT27 ADTO	
		0xC BT28 ADTO	0xD BT29 ADTO	0xE Setting prohibited	0xF Setting prohibited	
RIC_RESIN 308 (0x0268)	A/D activation channel 43	0x0 BT00 ADTO	0x1 BT01 ADTO	0x2 BT02 ADTO	0x3 BT03 ADTO	0x8
		0x4 BT04 ADTO	0x5 BT05 ADTO	0x6 BT06 ADTO	0x7 BT07 ADTO	
		0x8 BT08 ADTO	0x9 BT09 ADTO	0xA BT10 ADTO	0xB BT11 ADTO	
		0xC BT12 ADTO	0xD BT13 ADTO	0xE BT14 ADTO	0xF BT15 ADTO	
RIC_RESIN 309 (0x026A)		0x0 BT16 ADTO	0x1 BT17 ADTO	0x2 BT18 ADTO	0x3 BT19 ADTO	0x9
		0x4 BT20 ADTO	0x5 BT21 ADTO	0x6 BT22 ADTO	0x7 BT23 ADTO	
		0x8 BT24 ADTO	0x9 BT25 ADTO	0xA BT26 ADTO	0xB BT27 ADTO	
		0xC BT28 ADTO	0xD BT29 ADTO	0xE Setting prohibited	0xF Setting prohibited	
RIC_RESIN 314 (0x0274)	A/D activation channel 46	0x0 BT00 ADTO	0x1 BT01 ADTO	0x2 BT02 ADTO	0x3 BT03 ADTO	0x8
		0x4 BT04 ADTO	0x5 BT05 ADTO	0x6 BT06 ADTO	0x7 BT07 ADTO	
		0x8 BT08 ADTO	0x9 BT09 ADTO	0xA BT10 ADTO	0xB BT11 ADTO	
		0xC BT12 ADTO	0xD BT13 ADTO	0xE BT14 ADTO	0xF BT15 ADTO	
RIC_RESIN 315 (0x0276)		0x0 BT16 ADTO	0x1 BT17 ADTO	0x2 BT18 ADTO	0x3 BT19 ADTO	0x9
		0x4 BT20 ADTO	0x5 BT21 ADTO	0x6 BT22 ADTO	0x7 BT23 ADTO	
		0x8 BT24 ADTO	0x9 BT25 ADTO	0xA BT26 ADTO	0xB BT27 ADTO	
		0xC BT28 ADTO	0xD BT29 ADTO	0xE Setting prohibited	0xF Setting prohibited	

Reg Add (Offset)	Resource	RESSEL Data[3:0] Select Basetimer ADTO signal ch				PORTSEL Data[3:0]
RIC_RESIN 316 (0x0278)	A/D activation channel 47	0x0 BT00 ADTO	0x1 BT01 ADTO	0x2 BT02 ADTO	0x3 BT03 ADTO	0x8
		0x4 BT04 ADTO	0x5 BT05 ADTO	0x6 BT06 ADTO	0x7 BT07 ADTO	
		0x8 BT08 ADTO	0x9 BT09 ADTO	0xA BT10 ADTO	0xB BT11 ADTO	
		0xC BT12 ADTO	0xD BT13 ADTO	0xE BT14 ADTO	0xF BT15 ADTO	
RIC_RESIN 317 (0x027A)		0x0 BT16 ADTO	0x1 BT17 ADTO	0x2 BT18 ADTO	0x3 BT19 ADTO	0x9
		0x4 BT20 ADTO	0x5 BT21 ADTO	0x6 BT22 ADTO	0x7 BT23 ADTO	
		0x8 BT24 ADTO	0x9 BT25 ADTO	0xA BT26 ADTO	0xB BT27 ADTO	
		0xC BT28 ADTO	0xD BT29 ADTO	0xE Setting prohibited	0xF Setting prohibited	
RIC_RESIN 318 (0x027C)	A/D activation channel 48	0x0 BT00 ADTO	0x1 BT01 ADTO	0x2 BT02 ADTO	0x3 BT03 ADTO	0x8
		0x4 BT04 ADTO	0x5 BT05 ADTO	0x6 BT06 ADTO	0x7 BT07 ADTO	
		0x8 BT08 ADTO	0x9 BT09 ADTO	0xA BT10 ADTO	0xB BT11 ADTO	
		0xC BT12 ADTO	0xD BT13 ADTO	0xE BT14 ADTO	0xF BT15 ADTO	
RIC_RESIN 319 (0x027E)		0x0 BT16 ADTO	0x1 BT17 ADTO	0x2 BT18 ADTO	0x3 BT19 ADTO	0x9
		0x4 BT20 ADTO	0x5 BT21 ADTO	0x6 BT22 ADTO	0x7 BT23 ADTO	
		0x8 BT24 ADTO	0x9 BT25 ADTO	0xA BT26 ADTO	0xB BT27 ADTO	
		0xC BT28 ADTO	0xD BT29 ADTO	0xE Setting prohibited	0xF Setting prohibited	
RIC_RESIN 320 (0x0280)	A/D activation channel 49	0x0 BT00 ADTO	0x1 BT01 ADTO	0x2 BT02 ADTO	0x3 BT03 ADTO	0x8
		0x4 BT04 ADTO	0x5 BT05 ADTO	0x6 BT06 ADTO	0x7 BT07 ADTO	
		0x8 BT08 ADTO	0x9 BT09 ADTO	0xA BT10 ADTO	0xB BT11 ADTO	
		0xC BT12 ADTO	0xD BT13 ADTO	0xE BT14 ADTO	0xF BT15 ADTO	

Reg Add (Offset)	Resource	RESSEL Data[3:0] Select Basetimer ADTO signal ch				PORTSEL Data[3:0]
RIC_RESIN 321 (0x0282)	A/D activation channel 49	0x0 BT16 ADTO	0x1 BT17 ADTO	0x2 BT18 ADTO	0x3 BT19 ADTO	0x9
		0x4 BT20 ADTO	0x5 BT21 ADTO	0x6 BT22 ADTO	0x7 BT23 ADTO	
		0x8 BT24 ADTO	0x9 BT25 ADTO	0xA BT26 ADTO	0xB BT27 ADTO	
		0xC BT28 ADTO	0xD BT29 ADTO	0xE Setting prohibited	0xF Setting prohibited	
RIC_RESIN 322 (0x0284)	A/D activation channel 50	0x0 BT00 ADTO	0x1 BT01 ADTO	0x2 BT02 ADTO	0x3 BT03 ADTO	0x8
		0x4 BT04 ADTO	0x5 BT05 ADTO	0x6 BT06 ADTO	0x7 BT07 ADTO	
		0x8 BT08 ADTO	0x9 BT09 ADTO	0xA BT10 ADTO	0xB BT11 ADTO	
		0xC BT12 ADTO	0xD BT13 ADTO	0xE BT14 ADTO	0xF BT15 ADTO	
RIC_RESIN 323 (0x0286)		0x0 BT16 ADTO	0x1 BT17 ADTO	0x2 BT18 ADTO	0x3 BT19 ADTO	0x9
		0x4 BT20 ADTO	0x5 BT21 ADTO	0x6 BT22 ADTO	0x7 BT23 ADTO	
		0x8 BT24 ADTO	0x9 BT25 ADTO	0xA BT26 ADTO	0xB BT27 ADTO	
		0xC BT28 ADTO	0xD BT29 ADTO	0xE Setting prohibited	0xF Setting prohibited	
RIC_RESIN 324 (0x0288)	A/D activation channel 51	0x0 BT00 ADTO	0x1 BT01 ADTO	0x2 BT02 ADTO	0x3 BT03 ADTO	0x8
		0x4 BT04 ADTO	0x5 BT05 ADTO	0x6 BT06 ADTO	0x7 BT07 ADTO	
		0x8 BT08 ADTO	0x9 BT09 ADTO	0xA BT10 ADTO	0xB BT11 ADTO	
		0xC BT12 ADTO	0xD BT13 ADTO	0xE BT14 ADTO	0xF BT15 ADTO	
RIC_RESIN 325 (0x028A)		0x0 BT16 ADTO	0x1 BT17 ADTO	0x2 BT18 ADTO	0x3 BT19 ADTO	0x9
		0x4 BT20 ADTO	0x5 BT21 ADTO	0x6 BT22 ADTO	0x7 BT23 ADTO	
		0x8 BT24 ADTO	0x9 BT25 ADTO	0xA BT26 ADTO	0xB BT27 ADTO	
		0xC BT28 ADTO	0xD BT29 ADTO	0xE Setting prohibited	0xF Setting prohibited	

Reg Add (Offset)	Resource	RESSEL Data[3:0] Select Basetimer ADTO signal ch				PORTSEL Data[3:0]
RIC_RESIN 326 (0x028C)	A/D activation channel 52	0x0 BT00 ADTO	0x1 BT01 ADTO	0x2 BT02 ADTO	0x3 BT03 ADTO	0x8
		0x4 BT04 ADTO	0x5 BT05 ADTO	0x6 BT06 ADTO	0x7 BT07 ADTO	
		0x8 BT08 ADTO	0x9 BT09 ADTO	0xA BT10 ADTO	0xB BT11 ADTO	
		0xC BT12 ADTO	0xD BT13 ADTO	0xE BT14 ADTO	0xF BT15 ADTO	
RIC_RESIN 327 (0x028E)		0x0 BT16 ADTO	0x1 BT17 ADTO	0x2 BT18 ADTO	0x3 BT19 ADTO	0x9
		0x4 BT20 ADTO	0x5 BT21 ADTO	0x6 BT22 ADTO	0x7 BT23 ADTO	
		0x8 BT24 ADTO	0x9 BT25 ADTO	0xA BT26 ADTO	0xB BT27 ADTO	
		0xC BT28 ADTO	0xD BT29 ADTO	0xE Setting prohibited	0xF Setting prohibited	
RIC_RESIN 328 (0x0290)	A/D activation channel 53	0x0 BT00 ADTO	0x1 BT01 ADTO	0x2 BT02 ADTO	0x3 BT03 ADTO	0x8
		0x4 BT04 ADTO	0x5 BT05 ADTO	0x6 BT06 ADTO	0x7 BT07 ADTO	
		0x8 BT08 ADTO	0x9 BT09 ADTO	0xA BT10 ADTO	0xB BT11 ADTO	
		0xC BT12 ADTO	0xD BT13 ADTO	0xE BT14 ADTO	0xF BT15 ADTO	
RIC_RESIN 329 (0x0292)		0x0 BT16 ADTO	0x1 BT17 ADTO	0x2 BT18 ADTO	0x3 BT19 ADTO	0x9
		0x4 BT20 ADTO	0x5 BT21 ADTO	0x6 BT22 ADTO	0x7 BT23 ADTO	
		0x8 BT24 ADTO	0x9 BT25 ADTO	0xA BT26 ADTO	0xB BT27 ADTO	
		0xC BT28 ADTO	0xD BT29 ADTO	0xE Setting prohibited	0xF Setting prohibited	
RIC_RESIN 332 (0x0298)	A/D activation channel 55	0x0 BT00 ADTO	0x1 BT01 ADTO	0x2 BT02 ADTO	0x3 BT03 ADTO	0x8
		0x4 BT04 ADTO	0x5 BT05 ADTO	0x6 BT06 ADTO	0x7 BT07 ADTO	
		0x8 BT08 ADTO	0x9 BT09 ADTO	0xA BT10 ADTO	0xB BT11 ADTO	
		0xC BT12 ADTO	0xD BT13 ADTO	0xE BT14 ADTO	0xF BT15 ADTO	

Reg Add (Offset)	Resource	RESSEL Data[3:0] Select Basetimer ADTO signal ch				PORTSEL Data[3:0]
RIC_RESIN 333 (0x029A)	A/D activation channel 55	0x0 BT16 ADTO	0x1 BT17 ADTO	0x2 BT18 ADTO	0x3 BT19 ADTO	0x9
		0x4 BT20 ADTO	0x5 BT21 ADTO	0x6 BT22 ADTO	0x7 BT23 ADTO	
		0x8 BT24 ADTO	0x9 BT25 ADTO	0xA BT26 ADTO	0xB BT27 ADTO	
		0xC BT28 ADTO	0xD BT29 ADTO	0xE Setting prohibited	0xF Setting prohibited	
RIC_RESIN 334 (0x029C)	A/D activation channel 56	0x0 BT00 ADTO	0x1 BT01 ADTO	0x2 BT02 ADTO	0x3 BT03 ADTO	0x8
		0x4 BT04 ADTO	0x5 BT05 ADTO	0x6 BT06 ADTO	0x7 BT07 ADTO	
		0x8 BT08 ADTO	0x9 BT09 ADTO	0xA BT10 ADTO	0xB BT11 ADTO	
		0xC BT12 ADTO	0xD BT13 ADTO	0xE BT14 ADTO	0xF BT15 ADTO	
RIC_RESIN 335 (0x029E)		0x0 BT16 ADTO	0x1 BT17 ADTO	0x2 BT18 ADTO	0x3 BT19 ADTO	0x9
		0x4 BT20 ADTO	0x5 BT21 ADTO	0x6 BT22 ADTO	0x7 BT23 ADTO	
		0x8 BT24 ADTO	0x9 BT25 ADTO	0xA BT26 ADTO	0xB BT27 ADTO	
		0xC BT28 ADTO	0xD BT29 ADTO	0xE Setting prohibited	0xF Setting prohibited	
RIC_RESIN 336 (0x02A0)	A/D activation channel 57	0x0 BT00 ADTO	0x1 BT01 ADTO	0x2 BT02 ADTO	0x3 BT03 ADTO	0x8
		0x4 BT04 ADTO	0x5 BT05 ADTO	0x6 BT06 ADTO	0x7 BT07 ADTO	
		0x8 BT08 ADTO	0x9 BT09 ADTO	0xA BT10 ADTO	0xB BT11 ADTO	
		0xC BT12 ADTO	0xD BT13 ADTO	0xE BT14 ADTO	0xF BT15 ADTO	
RIC_RESIN 337 (0x02A2)		0x0 BT16 ADTO	0x1 BT17 ADTO	0x2 BT18 ADTO	0x3 BT19 ADTO	0x9
		0x4 BT20 ADTO	0x5 BT21 ADTO	0x6 BT22 ADTO	0x7 BT23 ADTO	
		0x8 BT24 ADTO	0x9 BT25 ADTO	0xA BT26 ADTO	0xB BT27 ADTO	
		0xC BT28 ADTO	0xD BT29 ADTO	0xE Setting prohibited	0xF Setting prohibited	

Reg Add (Offset)	Resource	RESSEL Data[3:0] Select Basetimer ADTO signal ch				PORTSEL Data[3:0]
RIC_RESIN 338 (0x02A4)	A/D activation channel 58	0x0 BT00 ADTO	0x1 BT01 ADTO	0x2 BT02 ADTO	0x3 BT03 ADTO	0x8
		0x4 BT04 ADTO	0x5 BT05 ADTO	0x6 BT06 ADTO	0x7 BT07 ADTO	
		0x8 BT08 ADTO	0x9 BT09 ADTO	0xA BT10 ADTO	0xB BT11 ADTO	
		0xC BT12 ADTO	0xD BT13 ADTO	0xE BT14 ADTO	0xF BT15 ADTO	
RIC_RESIN 339 (0x02A6)		0x0 BT16 ADTO	0x1 BT17 ADTO	0x2 BT18 ADTO	0x3 BT19 ADTO	0x9
		0x4 BT20 ADTO	0x5 BT21 ADTO	0x6 BT22 ADTO	0x7 BT23 ADTO	
		0x8 BT24 ADTO	0x9 BT25 ADTO	0xA BT26 ADTO	0xB BT27 ADTO	
		0xC BT28 ADTO	0xD BT29 ADTO	0xE Setting prohibited	0xF Setting prohibited	
RIC_RESIN 340 (0x02A8)	A/D activation channel 59	0x0 BT00 ADTO	0x1 BT01 ADTO	0x2 BT02 ADTO	0x3 BT03 ADTO	0x8
		0x4 BT04 ADTO	0x5 BT05 ADTO	0x6 BT06 ADTO	0x7 BT07 ADTO	
		0x8 BT08 ADTO	0x9 BT09 ADTO	0xA BT10 ADTO	0xB BT11 ADTO	
		0xC BT12 ADTO	0xD BT13 ADTO	0xE BT14 ADTO	0xF BT15 ADTO	
RIC_RESIN 341 (0x02AA)		0x0 BT16 ADTO	0x1 BT17 ADTO	0x2 BT18 ADTO	0x3 BT19 ADTO	0x9
		0x4 BT20 ADTO	0x5 BT21 ADTO	0x6 BT22 ADTO	0x7 BT23 ADTO	
		0x8 BT24 ADTO	0x9 BT25 ADTO	0xA BT26 ADTO	0xB BT27 ADTO	
		0xC BT28 ADTO	0xD BT29 ADTO	0xE Setting prohibited	0xF Setting prohibited	
RIC_RESIN 342 (0x02AC)	A/D activation channel 60	0x0 BT00 ADTO	0x1 BT01 ADTO	0x2 BT02 ADTO	0x3 BT03 ADTO	0x8
		0x4 BT04 ADTO	0x5 BT05 ADTO	0x6 BT06 ADTO	0x7 BT07 ADTO	
		0x8 BT08 ADTO	0x9 BT09 ADTO	0xA BT10 ADTO	0xB BT11 ADTO	
		0xC BT12 ADTO	0xD BT13 ADTO	0xE BT14 ADTO	0xF BT15 ADTO	

Reg Add (Offset)	Resource	RESSEL Data[3:0] Select Basetimer ADTO signal ch				PORTSEL Data[3:0]
RIC_RESIN 343 (0x02AE)	A/D activation channel 60	0x0 BT16 ADTO	0x1 BT17 ADTO	0x2 BT18 ADTO	0x3 BT19 ADTO	0x9
		0x4 BT20 ADTO	0x5 BT21 ADTO	0x6 BT22 ADTO	0x7 BT23 ADTO	
		0x8 BT24 ADTO	0x9 BT25 ADTO	0xA BT26 ADTO	0xB BT27 ADTO	
		0xC BT28 ADTO	0xD BT29 ADTO	0xE Setting prohibited	0xF Setting prohibited	
RIC_RESIN 344 (0x02B0)	A/D activation channel 61	0x0 BT00 ADTO	0x1 BT01 ADTO	0x2 BT02 ADTO	0x3 BT03 ADTO	0x8
		0x4 BT04 ADTO	0x5 BT05 ADTO	0x6 BT06 ADTO	0x7 BT07 ADTO	
		0x8 BT08 ADTO	0x9 BT09 ADTO	0xA BT10 ADTO	0xB BT11 ADTO	
		0xC BT12 ADTO	0xD BT13 ADTO	0xE BT14 ADTO	0xF BT15 ADTO	
RIC_RESIN 345 (0x02B2)		0x0 BT16 ADTO	0x1 BT17 ADTO	0x2 BT18 ADTO	0x3 BT19 ADTO	0x9
		0x4 BT20 ADTO	0x5 BT21 ADTO	0x6 BT22 ADTO	0x7 BT23 ADTO	
		0x8 BT24 ADTO	0x9 BT25 ADTO	0xA BT26 ADTO	0xB BT27 ADTO	
		0xC BT28 ADTO	0xD BT29 ADTO	0xE Setting prohibited	0xF Setting prohibited	
RIC_RESIN 346 (0x02B4)	A/D activation channel 62	0x0 BT00 ADTO	0x1 BT01 ADTO	0x2 BT02 ADTO	0x3 BT03 ADTO	0x8
		0x4 BT04 ADTO	0x5 BT05 ADTO	0x6 BT06 ADTO	0x7 BT07 ADTO	
		0x8 BT08 ADTO	0x9 BT09 ADTO	0xA BT10 ADTO	0xB BT11 ADTO	
		0xC BT12 ADTO	0xD BT13 ADTO	0xE BT14 ADTO	0xF BT15 ADTO	
RIC_RESIN 347 (0x02B6)		0x0 BT16 ADTO	0x1 BT17 ADTO	0x2 BT18 ADTO	0x3 BT19 ADTO	0x9
		0x4 BT20 ADTO	0x5 BT21 ADTO	0x6 BT22 ADTO	0x7 BT23 ADTO	
		0x8 BT24 ADTO	0x9 BT25 ADTO	0xA BT26 ADTO	0xB BT27 ADTO	
		0xC BT28 ADTO	0xD BT29 ADTO	0xE Setting prohibited	0xF Setting prohibited	

The following shows the detailed settings of the resource selection bits (RESSEL).

Table 5-13 Connection Relationship between Base Timer Timers and A/D Converter Activation Channels

RIC ADD	Resource	PORTSEL3 to PORTSEL0 (RIC_RESIN m)				RESSEL3 to RESSEL0				Description
RIC_RESIN m m:228～347	A/D activation Channel 0～63ch	0	0	0	0	X	X	X	X	BT00 corresponds to A/D converter unit 0 BT02 corresponds to A/D converter unit 1
		1	0	0	0	0	0	0	0	Select BT00 ADTO
		1	0	0	0	0	0	0	1	Select BT01 ADTO
		1	0	0	0	0	0	1	0	Select BT02 ADTO
		1	0	0	0	0	0	1	1	Select BT03 ADTO
		1	0	0	0	0	1	0	0	Select BT04 ADTO
		1	0	0	0	0	1	0	1	Select BT05 ADTO
		1	0	0	0	0	1	1	0	Select BT06 ADTO
		1	0	0	0	0	1	1	1	Select BT07 ADTO
		1	0	0	0	1	0	0	0	Select BT08 ADTO
		1	0	0	0	1	0	0	1	Select BT09 ADTO
		1	0	0	0	1	0	1	0	Select BT10 ADTO
		1	0	0	0	1	0	1	1	Select BT11 ADTO
		1	0	0	0	1	1	0	0	Select BT12 ADTO
		1	0	0	0	1	1	0	1	Select BT13 ADTO
		1	0	0	0	1	1	1	0	Select BT14 ADTO
		1	0	0	0	1	1	1	1	Select BT15 ADTO
RIC_RESIN m+1		0	0	0	0	X	X	X	X	BT00 corresponds to A/D converter unit 0 BT02 corresponds to A/D converter unit 1
		1	0	0	1	0	0	0	0	Select BT16 ADTO
		1	0	0	1	0	0	0	1	Select BT17 ADTO
		1	0	0	1	0	0	1	0	Select BT18 ADTO
		1	0	0	1	0	0	1	1	Select BT19 ADTO
		1	0	0	1	0	1	0	0	Select BT20 ADTO
		1	0	0	1	0	1	0	1	Select BT21 ADTO
		1	0	0	1	0	1	1	0	Select BT22 ADTO
		1	0	0	1	0	1	1	1	Select BT23 ADTO
		1	0	0	1	1	0	0	0	Select BT24 ADTO
		1	0	0	1	1	0	0	1	Select BT25 ADTO
		1	0	0	1	1	0	1	0	Select BT26 ADTO
		1	0	0	1	1	0	1	1	Select BT27 ADTO
		1	0	0	1	1	1	0	0	Select BT28 ADTO
		1	0	0	1	1	1	0	1	Select BT29 ADTO
		1	0	0	1	1	1	1	0	Setting prohibited
		1	0	0	1	1	1	1	1	Setting prohibited
		Other settings				x	x	x	x	Setting prohibited

Notes:

- *PORTSEL[3:0] setting, use the lower address of RIC_RESIN register allocated to each A/Dch.*
  For example, to set the A/D activation channel 3, is set at PORTSEL[3:0] in RIC_RESIN228.
- *The following numbers are not supported as activation channels in this model.*
  Channel No. 0, 1, 2, 4, 7, 8, 11, 16, 25, 26, 44, 45, 54, 63

5.2.9. Activation Request Modes

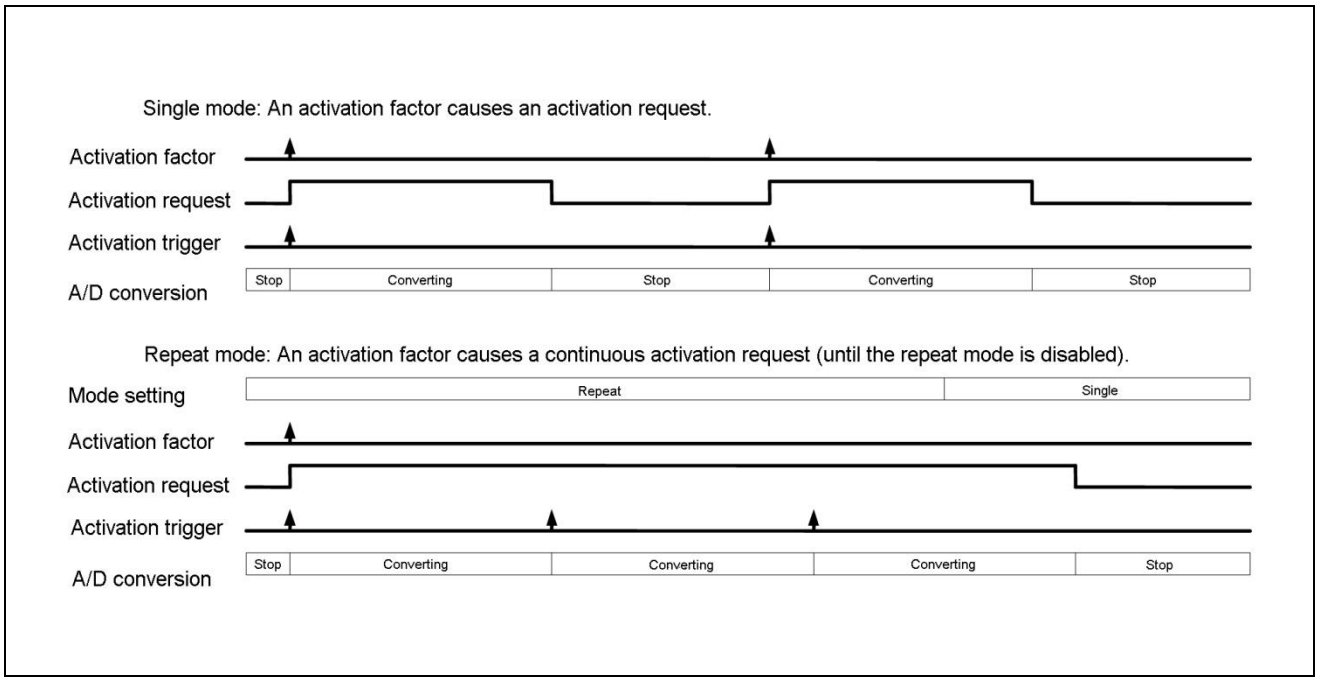
This section explains the activation request modes.

The activation request mode can be set for each activation channel. 2 activation modes are available: single mode and repeat mode. Set the mode in the RPT bit in the A/D activation trigger control status register (ADTCS).

In single mode, an activation factor causes an activation request. A/D conversion is performed once, and the activation request is canceled upon the completion of the A/D conversion.

In repeat mode, an activation factor causes a continuous activation request. A/D conversion is performed repeatedly, and the activation request continues until the repeat mode is canceled.

Figure 5-12 Activation Request Modes



5.2.10. A/D Conversion Data

This section explains A/D conversion data.

A/D conversion result data is stored in the A/D data bits (ADTCD.D11 to D0) for each activation channel. When the data protection function is disabled (ADTCS.PRT="0"), you can check the conversion data error flag bit (ADTCD.ERR) and the conversion data error status bit (ADTCD.ERRST) for the status of A/D conversion data stored in the A/D data bits (ADTCD.D11 to D0).
When the data protection function is enabled (ADTCS.PRT="1"), the conversion data error flag bit (ADTCD.ERR) and the conversion data error status bit (ADTCD.ERRST) are fixed to "0", respectively.

Table 5-7 Checking the Status of A/D Conversion Data (When the Data Protection Function Is Disabled)

ADTCD:ERR	ADTCD:ERRST	Status of A/D Conversion Data
0	0	Latest data (unread)
0	1	- (No meaning)
1	0	Old data (read) * Initial value
1	1	Latest data/overwriting occurred (unread) * Data may be abandoned.

5.2.11. Protection Function

This section explains the protection function.

The data protection function can be set for each A/D data register. The PRT bit in the A/D activation trigger control status register (ADTCS) sets the protection function. The protection function is effective for factors that are not compare match activation.

When the protection function is enabled and conversion result is stored in the A/D data register, data becomes protected. The cancel conditions for the data protected state can be selected using the PRTS bit in the ADTCS register.

- If the PRTS bit is "0", the activation request is masked until data is read from the A/D data register and the interrupt flag is cleared. There is no particularly specified order in reading data and clearing interrupt flag.
- If the PRTS bit is "1", the activation request is masked until data is read from the A/D data register.
- In the data protected state, the activation request signal is masked (inactivated) even when the next activation factor is generated. This protects unread data in the A/D data register from being overwritten with the next A/D conversion data.

5.2.12. About the Scan Conversion Mode

This section explains the scan conversion mode.

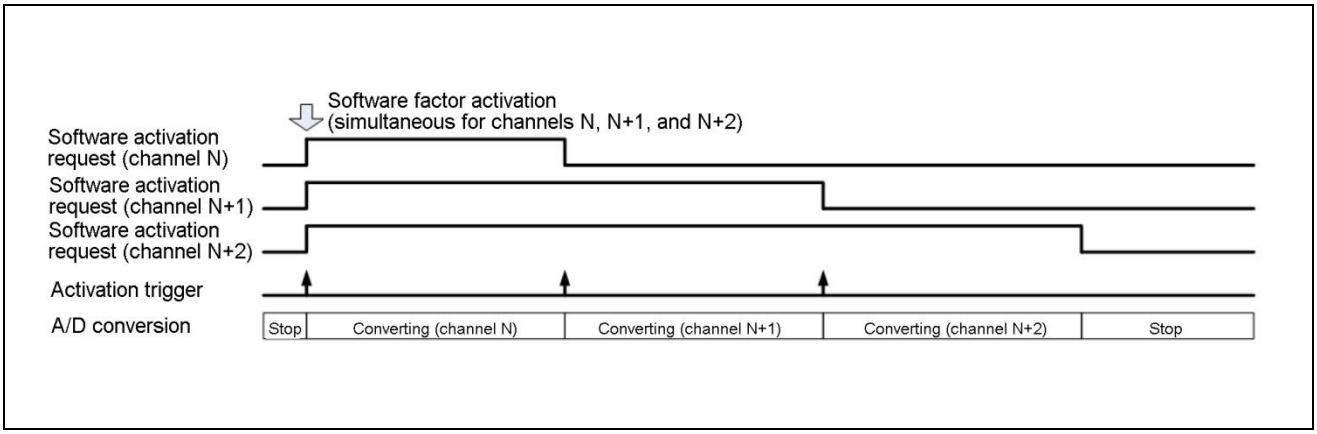
Scan conversion is an operation to perform A/D conversion sequentially from the activation channel with the smallest activation channel number. 1 type of scan conversion can be set for each 12-bit A/D converter unit.
The following scan conversion operations can be performed.

- Single scan conversion
- Continuous scan conversion
- Continuous scan conversion when the conversion count is specified for each channel
- Pause scan conversion when the conversion count is specified for each channel

Single Scan Conversion

To execute single scan conversion, set the repeat conversion selection bit to single mode (ADTCS.RPT="0") for each of the activation channels for which scan conversion is to be performed. Then, execute A/D activation simultaneously with the same activation factor.
The A/D conversion of scan conversion is performed sequentially from the activation channel with the smallest channel number for which A/D activation is performed by the A/D activation arbitration described later. After A/D conversion for the last activation channel ends, scan conversion stops.

Figure 5-13 Single Scan Conversion



Continuous Scan Conversion

To execute continuous scan conversion, set the repeat conversion selection bit to repeat mode (ADTCS.RPT="1") for each of the activation channels for which scan conversion is to be performed. Enable the data protection function (ADTCS.PRT="1"). Then, execute A/D activation simultaneously with the same activation factor.
If the repeat mode is set for multiple activation channels that correspond to the same A/D converter, only a certain channel is processed in order of priority by the A/D activation arbitration described later.
In this case, use the activation channels with the protection function enabled.

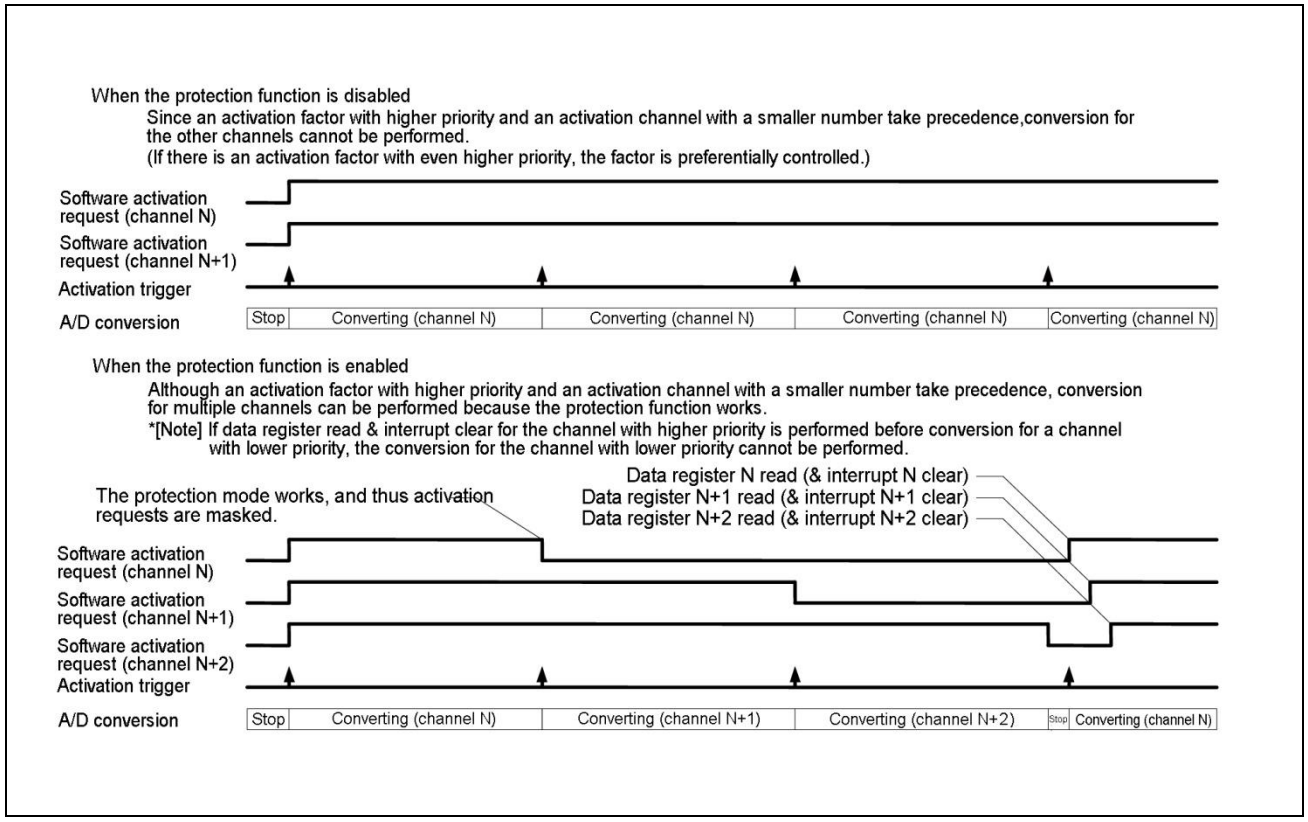
- When the data protection function is disabled (ADTCS.PRT="0")
  In A/D activation arbitration, an activation factor with higher priority and an activation channel with a smaller activation channel number take precedence. If the repeat mode is set (ADTCS.RPT="1"), an activation request is continuously output. Thus, conversion for activation channels with lower priority is not performed.

- When the data protection function is enabled (ADTCS.PRT="1")
  In A/D activation arbitration, an activation factor with higher priority and an activation channel with a smaller activation channel number take precedence. If the repeat mode is set (ADTCS.RPT="1"), an activation request is continuously output. However, if A/D conversion result is stored in the A/D data register (ADTCD), data becomes protected and the activation request is masked. This makes it possible to perform conversion for the next activation channel.

Note:

- *Note the following when you set multiple repeat modes (ADTCS.RPT="1") when the data protection function is enabled (ADTCS.PRT="1"): If the data protected state of an activation channel with higher priority is canceled before conversion for an activation channel with lower priority, the conversion for the activation channel with lower priority is not performed.*

Figure 5-14 Continuous Scan Conversion



Continuous Scan Conversion When the Conversion Count is Specified for Each Channel

To execute continuous scan conversion when the conversion count is specified for each channel, set the continuous scan conversion mode (ADSCANS.SCMD="0"). For each of the activation channels for which scan conversion is to be performed, set the repeat conversion selection bit to repeat mode (ADTCS.RPT="1"), enable execution of conversion count-specified scan conversion (ADNCS.CNTEN="1"), and set the conversion count (ADNCS.CCNT1, CCNT0="count"). Then, execute A/D activation simultaneously with the same activation factor.

In continuous scan conversion when the conversion count is specified for each channel, A/D conversion with conversion count specified (ADNCS.CCNT1, CCNT0) is performed for the activation channels for which scan conversion is to be performed in the order from the smallest activation channel number. After completion of the conversion, transition to A/D activation for the next activation channel occurs. You can check the conversion count completion flag bits (ADEOCF.EOCF) for the activation channels for which A/D conversion with conversion count specified is completed.

Then, after A/D conversion with the conversion count specified is completed for the last activation channel, the scan conversion completion interrupt factor flag bit (ADSCANS.SCINT) is set to "1" and the scan conversion is automatically repeated from the beginning. The conversion count completion flag bit (ADEOCF.EOCF) for each activation channel is automatically cleared to "0" when the scan conversion is repeated from the beginning.

Conversion count specification (ADNCS.CCNT1, CCNT0) can select a value from 1 to 4 times for each activation channel.

The data protection function can be enabled or disabled for each activation channel. When the data protection function is enabled (ADTCS.PRT="1"), you can check the data protection status flag bits (ADPRTF.PRTF) for the data protected state of each activation channel.

- When the data protection function is disabled (ADTCS.PRT="0")
  A/D conversion is continuously performed without gaps. A/D conversion data is not protected, and data from the last A/D conversion performed for an activation channel is stored.
- When the data protection function is enabled (ADTCS.PRT="1")
  A/D conversion data is protected. Therefore, A/D conversion for the same activation channel is stopped while data is protected. However, if scan conversion proceeds to a different activation channel, a different A/D data register (ADTCD) is used. Thus, A/D conversion is continuously performed without gaps.

Notes:

- *In the case of continuous scan conversion, the conversion count completion flag bit (ADEOCF.EOCF) for the last activation channel for which scan conversion is to be performed is set to "1". However, soon after that, it is automatically cleared to "0".*
- *Suppose that the same activation factor as for the channels for which scan conversion is to be performed is set for an activation channel larger than the last channel of the continuous scan conversion. In this case, the activation channel larger than the last channel is evaluated once by the A/D activation arbitration described later after the last channel of the continuous scan conversion is completed.*
- *The data protection function is effective for factors that are not compare match activation (ADTECS.STS2="0", ADTCS.STS1, STS0="11_B").*
- *When the data protection function is enabled (ADTCS.PRT="1"), cancel the data protected state before the next scan conversion is repeatedly performed. A/D conversion for activation channels in the data protected state is not performed.*

Figure 5-15 Continuous Scan Conversion When the Conversion Count Is Specified for Each Channel (When the Data Protection Function Is Disabled)

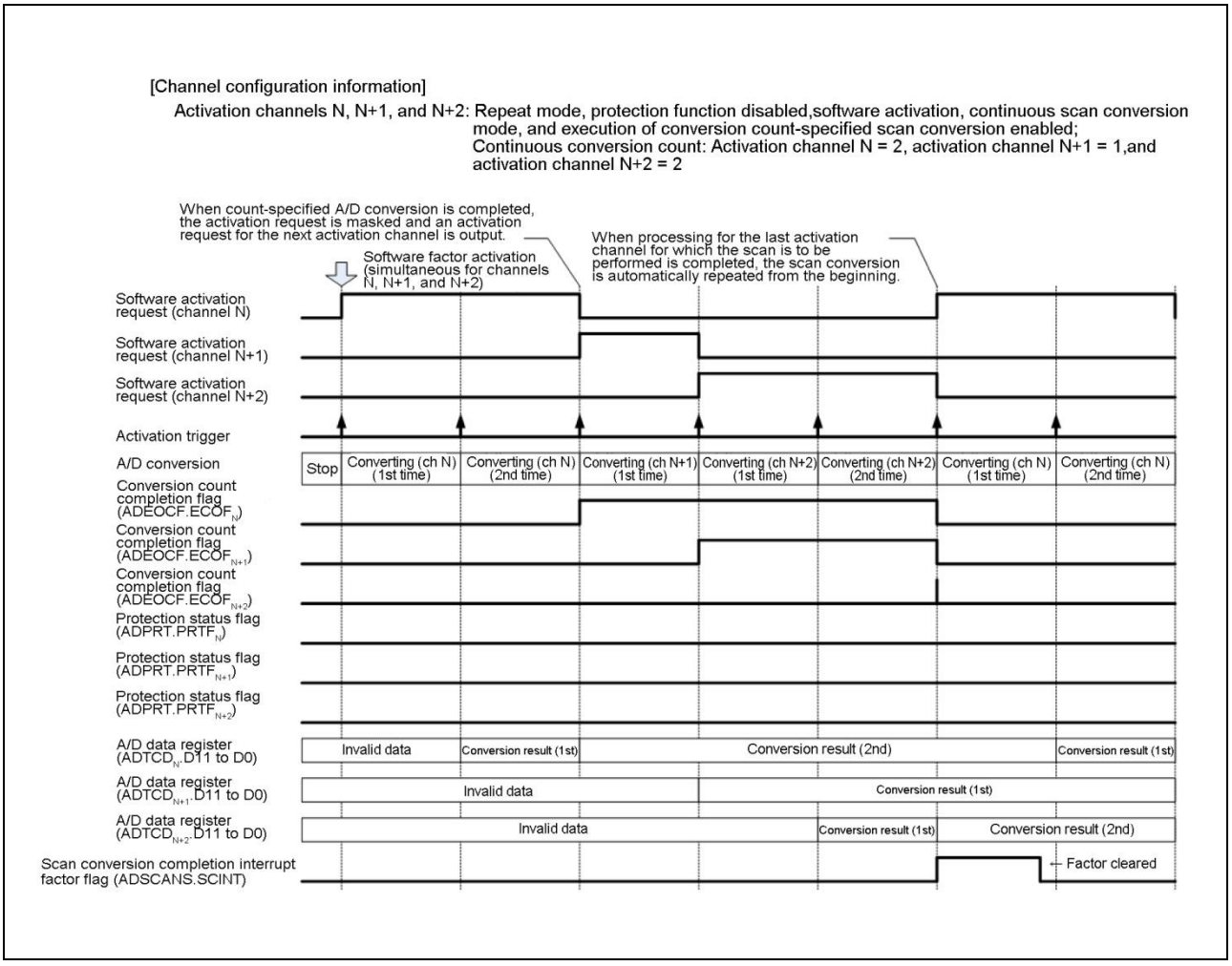
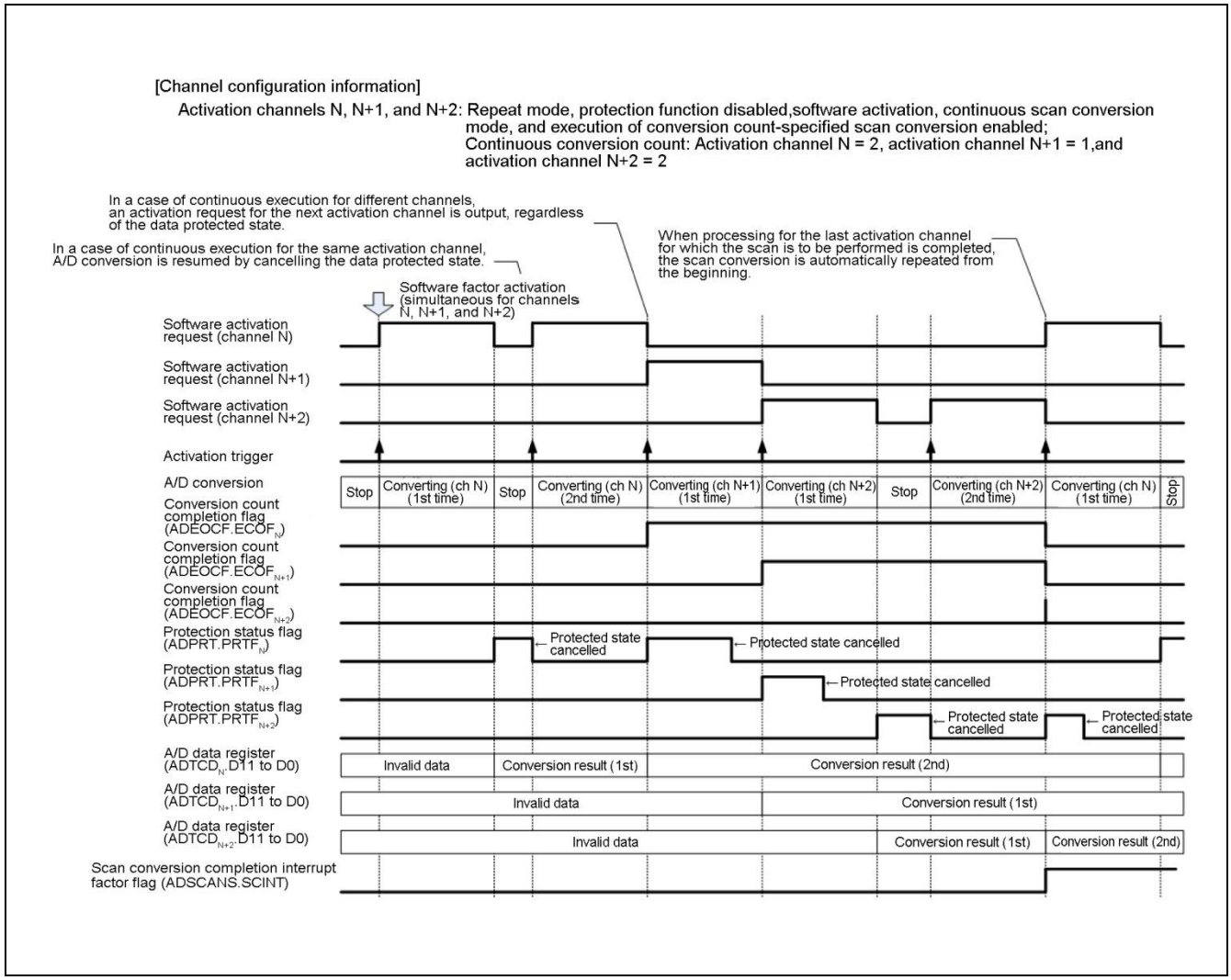


Figure 5-16 Continuous Scan Conversion When the Conversion Count Is Specified for Each Channel (When the Data Protection Function Is Enabled)



Pause Scan Conversion When the Conversion Count is Specified for Each Channel

To execute pause scan conversion when the conversion count is specified for each channel, set the pause scan conversion mode (ADSCANS.SCMD="1"). For each of the activation channels for which scan conversion is to be performed, set the repeat conversion selection bit to repeat mode (ADTCS.RPT="1"), enable execution of conversion count-specified scan conversion (ADNCS.CNTEN="1"), and set the conversion count (ADNCS.CCNT1, CCNT0="count"). Then, execute A/D activation simultaneously with the same activation factor.

In pause scan conversion when the conversion count is specified for each channel, A/D conversion with conversion count specified (ADNCS.CCNT1, CCNT0) is performed for the activation channels for which scan conversion is to be performed in the order from the smallest activation channel number. After completion of the conversion, transition to A/D activation for the next activation channel occurs. You can check the conversion count completion flag bits (ADEOCF.EOCF) for the activation channels for which A/D conversion with conversion count specified is completed.

Then, after A/D conversion with the conversion count specified is completed for the last activation channel, the scan conversion completion interrupt factor flag bit (ADSCANS.SCINT) is set to "1" and the scan conversion is paused. The scan conversion is resumed (recovers from the paused state) when an A/D activation factor is generated for any of the channels for which scan conversion is to be performed. The conversion count completion flag bit (ADEOCF.EOCF) for each activation channel is automatically cleared to "0" when the scan conversion is resumed (recovers from the paused state).

Conversion count specification (ADNCS.CCNT1, CCNT0) can select a value from 1 to 4 times for each activation channel.

The data protection function can be enabled or disabled for each activation channel. When the data protection function is enabled (ADTCS.PRT="1"), you can check the data protection status flag bits (ADPRTF.PRTF) for the data protected state of each activation channel.

- When the data protection function is disabled (ADTCS.PRT="0")
  A/D conversion in a state other than paused is continuously performed without gaps. A/D conversion data is not protected, and data from the last A/D conversion performed for an activation channel is stored.
- When the data protection function is enabled (ADTCS.PRT="1")
  A/D conversion data is protected. Therefore, A/D conversion for the same activation channel is stopped while data is protected. However, if scan conversion proceeds to a different activation channel, a different A/D data register (ADTCD) is used. Thus, A/D conversion is continuously performed without gaps.

Notes:

- *Suppose that the same activation factor as for the channels for which scan conversion is to be performed is set for an activation channel larger than the last channel of the pause scan conversion. In this case, A/D conversion for the activation channel larger than the last channel can be performed only during a pause period.*
- *The data protection function is effective for factors that are not compare match activation (ADTECS.STS2="0", ADTCS.STS1, STS0="11_B").*
- *When the data protection function is enabled (ADTCS.PRT="1"), cancel the data protected state before the next scan conversion starts. A/D conversion for activation channels in the data protected state is not performed.*

Figure 5-17 Pause Scan Conversion When the Conversion Count Is Specified for Each Channel (When the Data Protection Function Is Disabled)

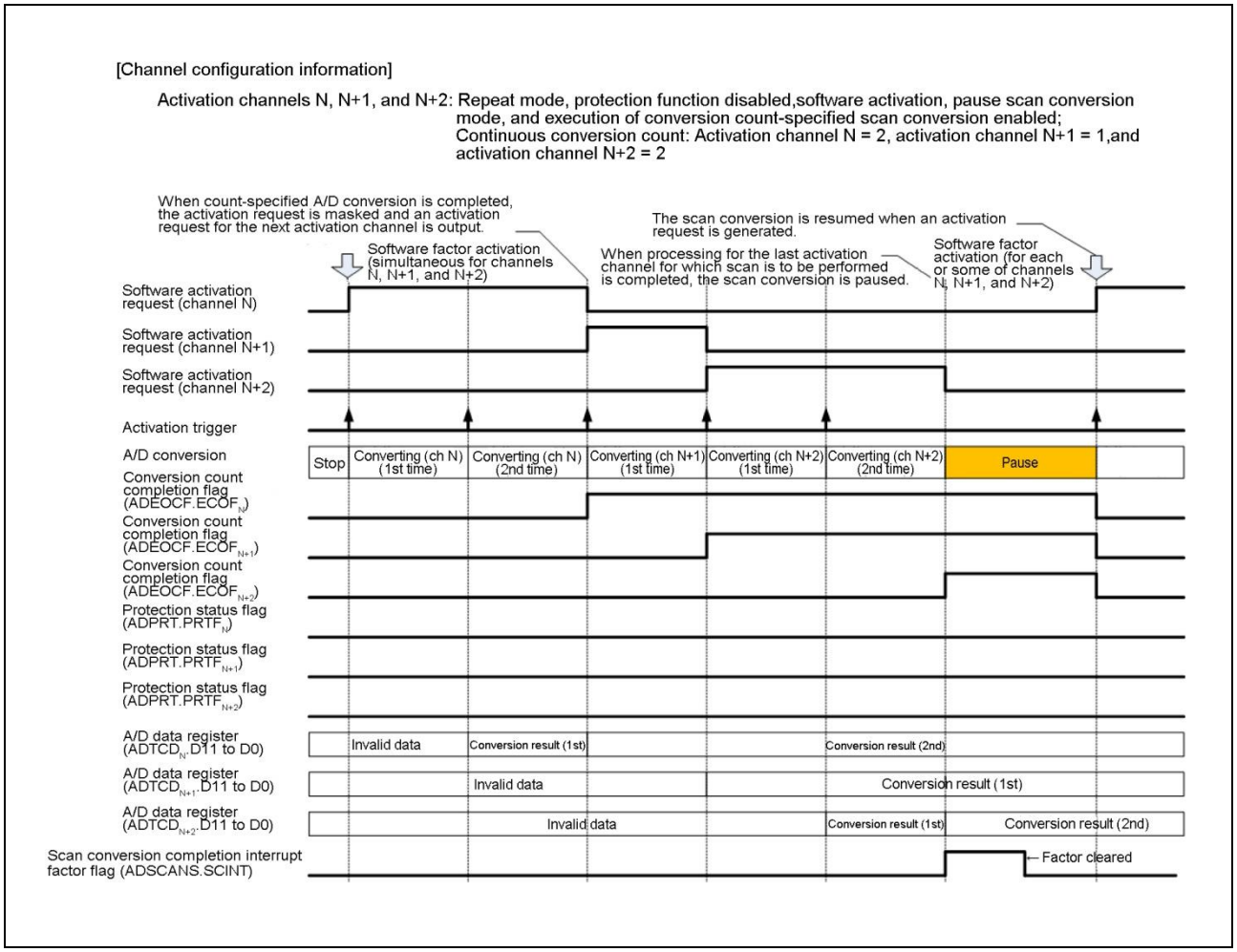
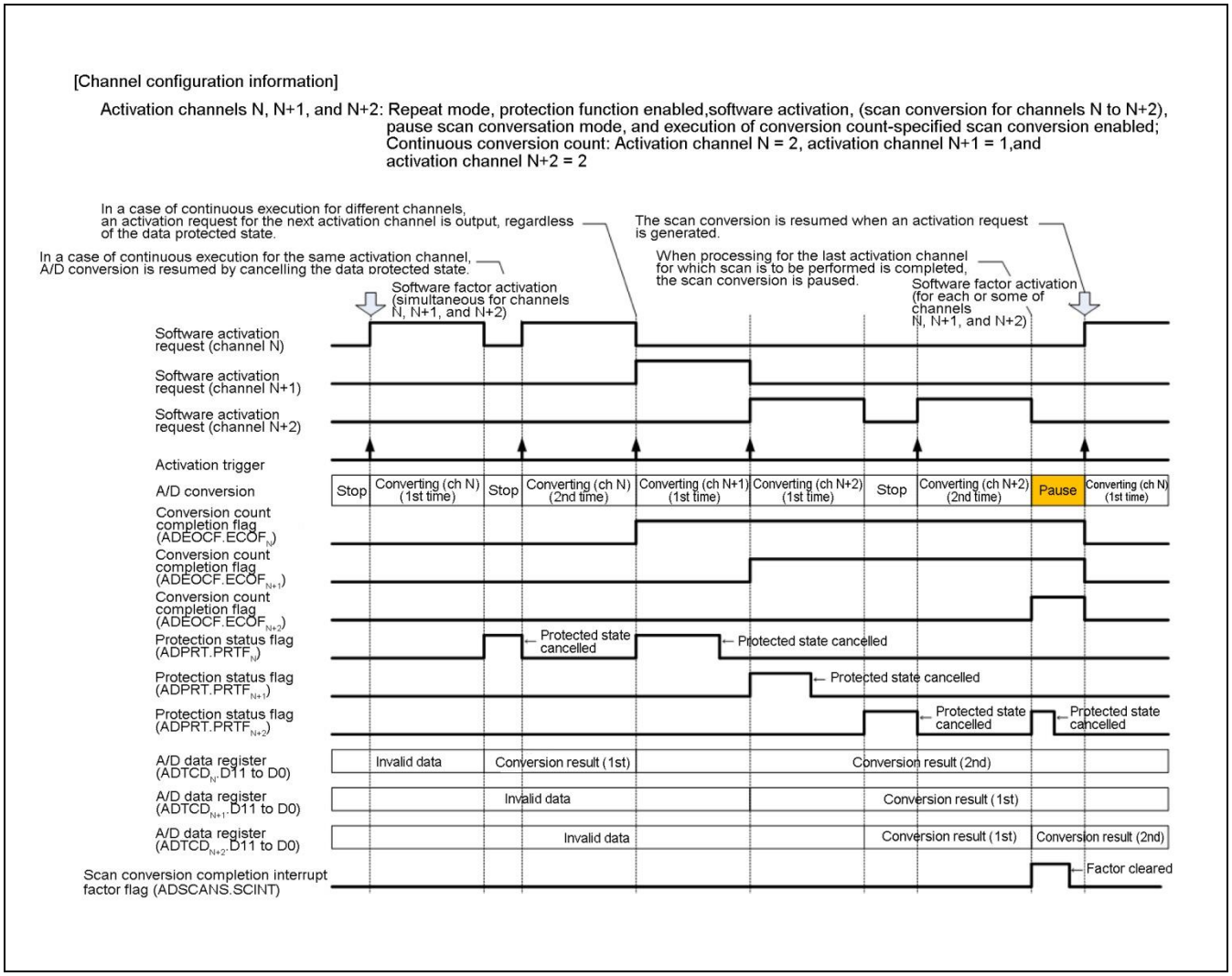


Figure 5-18 Pause Scan Conversion When the Conversion Count Is Specified for Each Channel (When the Data Protection Function Is Enabled)



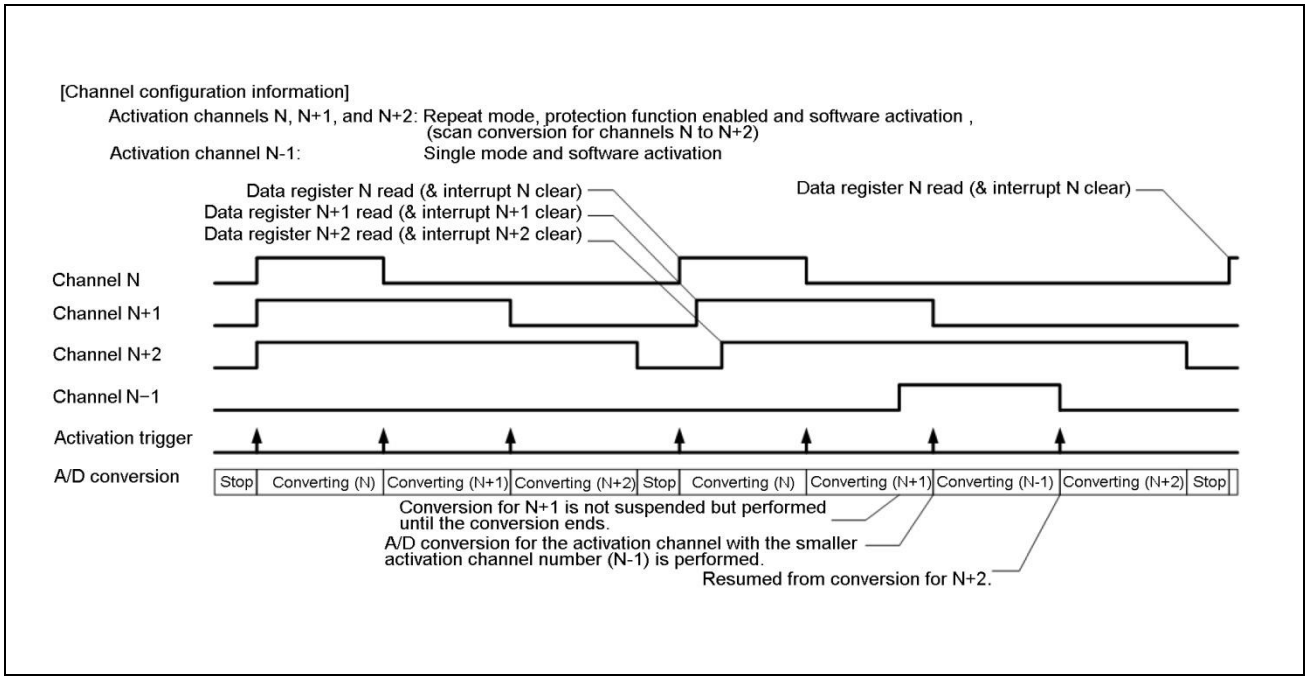
5.2.13. About High-Priority Activation Request Operation for Another Activation Channel during Scan Conversion

This section explains High-priority activation request operation for another activation channel during scan conversion.

If scan conversion is performed for the activation channels that correspond to the same A/D converter unit, priority control is performed for the activation channels based on the activation factor and activation channel information. The A/D activation arbitration described later performs this priority control.

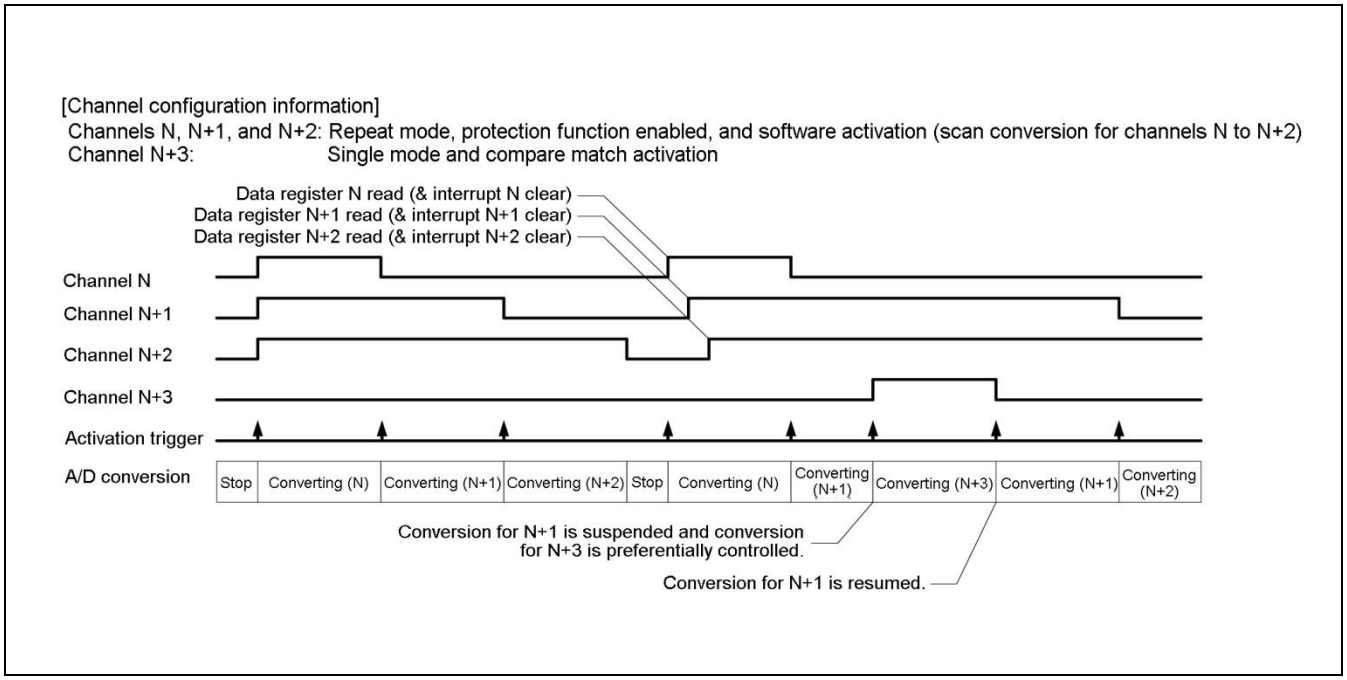
Suppose a case where a request with the same activation factor is generated from an activation channel with an activation channel number smaller than that of the activation channel for which scan conversion is to be performed. In this case, the scan conversion is temporarily stopped after the in-progress A/D conversion ends, and resumed after A/D conversion for the activation channel with a smaller activation channel number ends.

Figure 5-19 Case Where a Request with the Same Activation Factor Is Generated from an Activation Channel with a Smaller Activation Channel Number during Continuous Scan Conversion



Suppose a case where an activation factor with a priority higher than that of the activation channel for which scan conversion is to be performed is generated from another activation channel. In this case, the scan conversion is suspended. After the High-priority A/D conversion ends, the A/D conversion is resumed from the activation channel at which the scan conversion was suspended.

Figure 5-20 Operation for a Request with the Highest-Priority Activation Factor from Another Activation Channel during Continuous Scan Conversion



5.2.14. Forcible Termination of an Activation Request

This section explains forcible termination of an activation request.

The A/D activation request in progress bit (ADTCS0-63.BUSY) in the A/D activation trigger control status register can notify you of a state where an A/D activation request or conversion is in progress.
If you want to forcibly terminate the current A/D activation request or conversion, write "0" to this bit.

5.2.15. Range Comparison Function

This section explains the range comparison function.

Upper and Lower Threshold Settings for Range Comparison

The upper threshold setting register (ADRCUT) and the lower threshold setting register (ADRCLT) can configure 4 types of settings for each unit. Select 1 from 4 types of combinations of upper and lower threshold setting registers using the upper and lower threshold selection bits (ADRCCS.RCOTS1, RCOTS0) for each activation channel.

Table 5-8 Upper and Lower Threshold Selection for Range Comparison

Upper and Lower Threshold Selection Bits (ADRCCS:RCOTS1, RCOTS0)	Selection Result	
	A/D Converter Unit 0	A/D Converter Unit 1
"00B"	Upper threshold setting register 0 (ADRCUT0)/ Lower threshold setting register 0 (ADRCLT0)	Upper threshold setting register 4 (ADRCUT4)/ Lower threshold setting register 4 (ADRCLT4)
"01B"	Upper threshold setting register 1 (ADRCUT1)/ Lower threshold setting register 1 (ADRCLT1)	Upper threshold setting register 5 (ADRCUT5)/ Lower threshold setting register 5 (ADRCLT5)
"10B"	Upper threshold setting register 2 (ADRCUT2)/ Lower threshold setting register 2 (ADRCLT2)	Upper threshold setting register 6 (ADRCUT6)/ Lower threshold setting register 6 (ADRCLT6)
"11B"	Upper threshold setting register 3 (ADRCUT3)/ Lower threshold setting register 3 (ADRCLT3)	Upper threshold setting register 7 (ADRCUT7)/ Lower threshold setting register 7 (ADRCLT7)

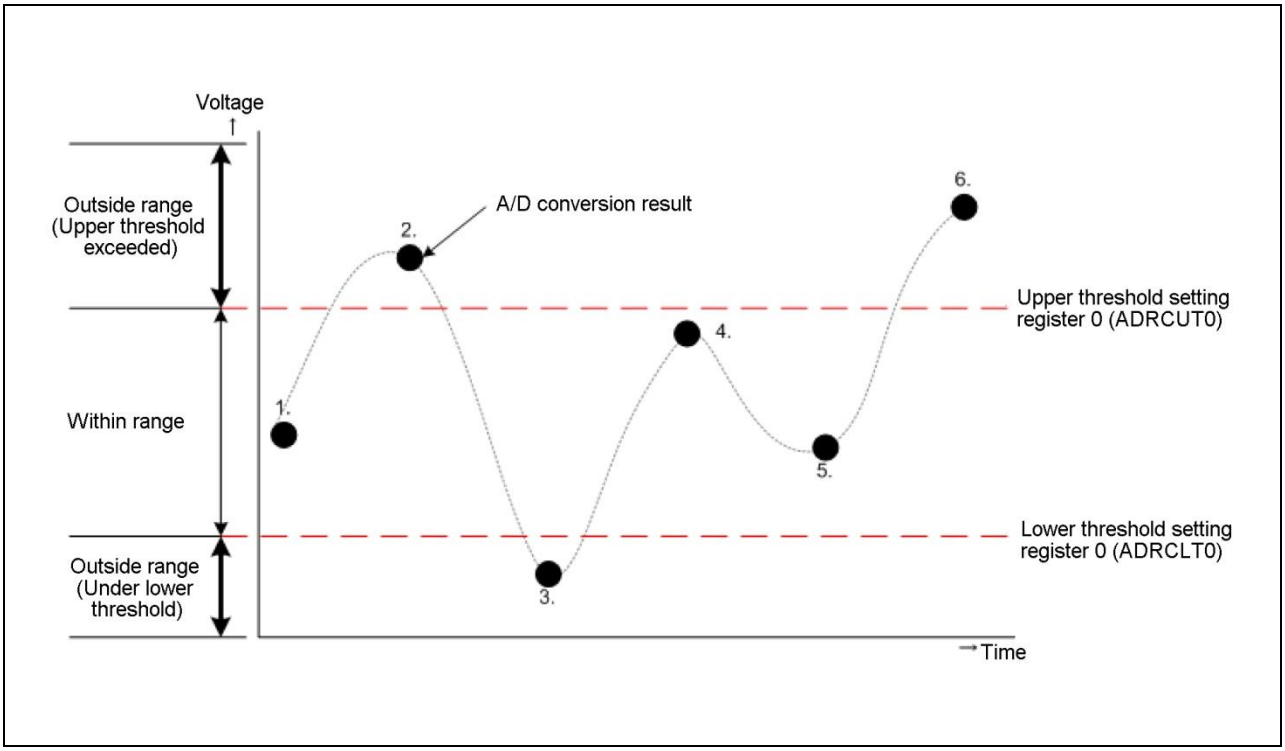
Range Comparison Operation

Range comparison is performed when range comparison is enabled (ADRCCS.RCOE="1"), A/D conversion ends, and then data is stored in the A/D data bits (ADTCD.D11 to D0).
Range comparison compares the upper and lower threshold setting registers (ADRCUT/ADRCLT) selected by the upper and lower threshold selection bits for range comparison (ADRCCS.RCOTS1, RCOTS0) with the A/D data bits (ADTCD.D11 to D0).
The range comparison result is input to the continuous detection function.

Table 5-9 Upper and Lower Threshold Selection for Range Comparison

Range Comparison Result	Outside-Range Check (ADRCCS: RCOIRS="0")	Inside-Range Check (ADRCCS: RCOIRS="1")
Outside range (upper threshold exceeded) A/D data bits > Upper threshold setting register	Detected	Not detected
Within range A/D data bits ≤ Upper threshold setting register and A/D data bits ≥ Lower threshold setting register	Not detected	Detected
Outside range (under lower threshold) A/D data bits < Lower threshold setting register	Detected	Not detected

Figure 5-21 Range Comparison Conditions



Continuous Detection Function for Range Comparison Result

The continuous detection function performs continuous detection of range comparison result and reduces noise, etc.

When as many detection states of range comparison result as the count set by the continuous detection count specification for range comparison (ADRCCS.RCOCD2 to RCOCD0) are continuously detected, the range comparison interrupt factor flag bit (ADRCIF.RCINT) is set to "1". If range comparison result comes to be "not detected" even once during continuous detection, the continuous detection measurement is cleared to 0 times and the measurement is performed again.

Table 5-10 Operating Conditions for the Continuous Detection Function

Continuous Detection Measurement Operation	Controlled for Each Activation Channel. Always Performed When Range Comparison Execution is Enabled (ADRCCS:RCOE="1").
Continuous Detection Count	− A value can be selected from 1 to 7 times using the continuous detection count specification (ADRCCS:RCOCD2 to RCOCD0). − The status of the detection count can be checked with the indication of the continuous detection count status (ADRCCS:RCOCD2 to RCOCD0).
Clearing Conditions	− When range comparison execution is disabled (ADRCCS:RCOE="0") − When range comparison result is "not detected."
Increment Conditions	− When range comparison result is "detected" However, when the count reaches the continuous detection count specification (ADRCCS:RCOCD2 to RCOCD0), it stops at the specified continuous detection count value.

Notes:

- *Even if range comparison result changes from the "upper threshold exceeded" state to the "under lower threshold" state at the time of outside-range check, the continuous detection measurement is not cleared to 0 times and the continuous detection continues.*
- *If you want to initialize the continuous detection count status of range comparison result, disable range comparison execution and then enable it (ADRCCS.RCOE="0" -> "1") while A/D conversion is not being requested.*
- *The range comparison interrupt factor flag (ADRCIF.RCINT) is set to "1" due to continuous detection when 2 clocks (peripheral clock) have elapsed since the interrupt control bit flag was set (ADTCS.INT="1") due to an A/D conversion end interrupt.*

Control of the Exceeded Flag for Range Comparison

When an outside-range check is performed for range comparison (ADRCCS.RCOIRS="0"), you can check the range comparison threshold excess flag bit for each activation channel (ADRCOT.RCOOF[n]) for indications of "upper threshold exceeded" and "under lower threshold".

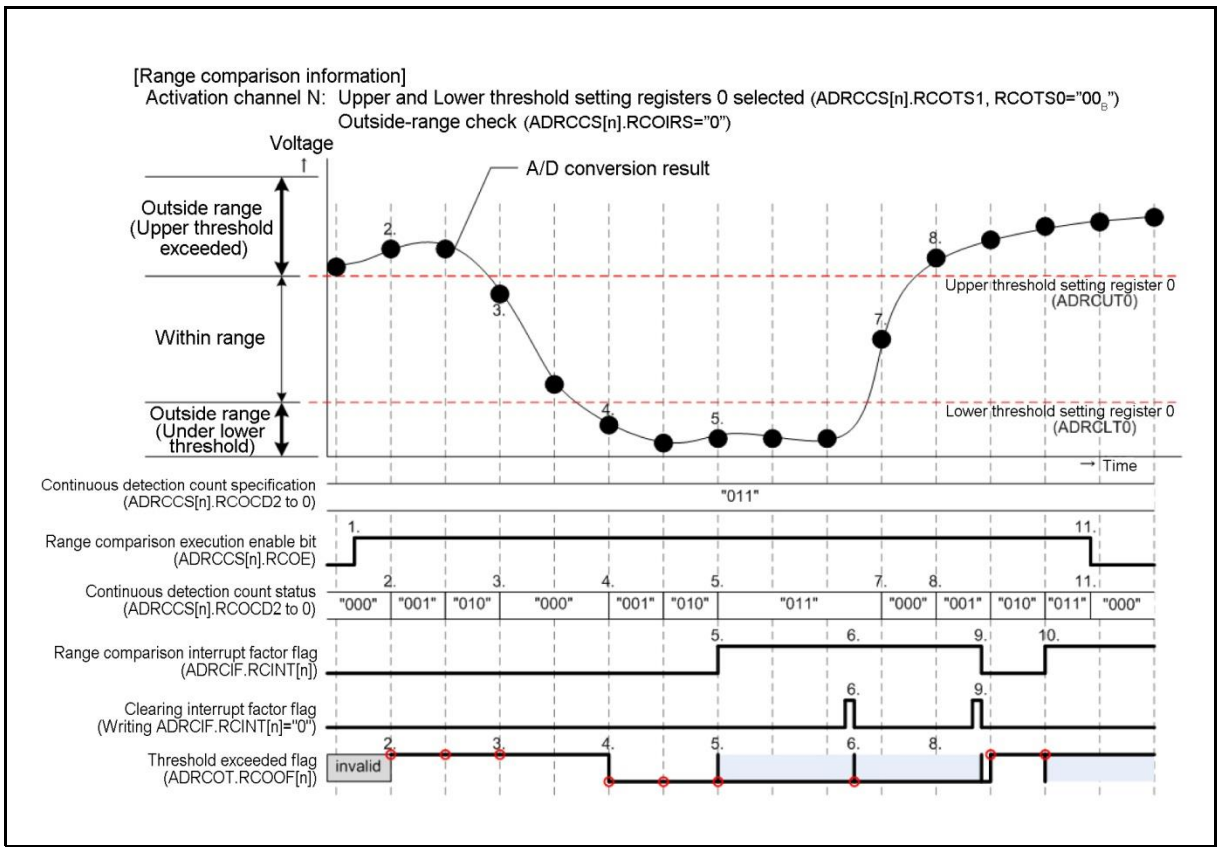
Table 5-11 Upper and Lower Threshold Selection for Range Comparison

Range Comparison Result	Range Comparison Threshold Excess Flag Bit (ADRCOT:RCOOF)	
	Outside-Range Check (ADRCCS: RCOIRS="0")	Inside-Range Check (ADRCCS: RCOIRS="1")
Outside range (upper threshold exceeded) A/D data bits > Upper threshold setting register	"1"	Previous value retained
Within range A/D data bits ≤ Upper threshold setting register and A/D data bits ≥ Lower threshold setting register	Previous value retained	Previous value retained
Outside range (under lower threshold) A/D data bits < Lower threshold setting register	"0"	Previous value retained

The range comparison threshold excess flag bit (ADRCOT.RCOOF) retains the value set in itself while the range comparison interrupt factor flag (ADRCIF.RCINT) is "1".

Example of Range Comparison Operation

Figure 5-22 Example of Range Comparison Operation



1. When range comparison execution is disabled (ADRCCS[n].RCOE="0"), the continuous detection count status (ADRCCS[n].RCOCD2 to RCOCD0) is initialized to "000_B".
   When range comparison execution is enabled (ADRCCS[n].RCOE="1"), range comparison operation starts.
2. When the range comparison result is "upper threshold exceeded", the continuous detection count status (ADRCCS[n].RCOCD2 to RCOCD0) is incremented.
   The threshold excess flag indicates "upper threshold exceeded" (ADRCOT.RCOOF[n]="1").
3. Since the range comparison result is found to be "within range" before the continuous detection count specification (ADRCCS[n]. RCOCD2 to RCOCD0="011_B") is reached, the continuous detection count status is initialized (ADRCCS[n].RCOCD2 to RCOCD0="000_B").
   The threshold excess flag (ADRCOT.RCOOF[n]) retains the previous value.
4. When the range comparison result is "under lower threshold", the continuous detection count status (ADRCCS[n].RCOCD2 to RCOCD0) is incremented.
   The threshold excess flag indicates "under lower threshold" (ADRCOT.RCOOF[n]="0").
5. Since the range comparison result repeatedly reaches the continuous detection count specification (ADRCCS[n].RCOCD2 to RCOCD0="011_B"), the range comparison interrupt factor flag (ADRCIF.RCINT[n]) is set to "1".
   The threshold excess flag (ADRCOT.RCOOF[n]) is set to the threshold excess state at the point when the range comparison interrupt factor flag is set (ADRCIF.RCINT[n]="1"). The set state is retained until the range comparison interrupt factor flag is cleared (ADRCIF.RCINT[n]="0").
6. If clearing the range comparison interrupt factor flag (ADRCIF.RCINT[n]="0") conflicts with a continuous detection state, the continuous detection state takes precedence. The range comparison interrupt factor flag is set (ADRCIF.RCINT[n]="1"), and the threshold excess flag (ADRCOT.RCOOF[n]) is set to the threshold excess state again.
7. When the range comparison result is "within range," the continuous detection count status is initialized (ADRCCS[n].RCOCD2 to RCOCD0="000_B") even if the range comparison interrupt factor flag is set (ADRCIF.RCINT[n]="1").
8. Even when the range comparison interrupt factor flag is set (ADRCIF.RCINT[n]="1"), the continuous detection count status (ADRCCS[n].RCOCD2 to RCOCD0) is incremented when the range comparison result is "upper threshold exceeded."
   However, since the range comparison interrupt factor flag is set (ADRCIF.RCINT[n]="1"), the threshold excess flag (ADRCOT.RCOOF[n]) retains the previous value.
9. Writing "0" to the range comparison interrupt factor flag (ADRCIF.RCINT[n]="0") clears the range comparison interrupt factor flag (ADRCIF.RCINT[n]="0").
   The threshold excess flag (ADRCOT.RCOOF[n]) is released from a retention state.
10. Since the range comparison result repeatedly reaches the continuous detection count specification (ADRCCS[n].RCOCD2 to RCOCD0="011_B"), the range comparison interrupt factor flag (ADRCIF.RCINT[n]) is set to "1".
    The threshold excess flag (ADRCOT.RCOOF[n]) is set to the threshold excess state at the point when the range comparison interrupt factor flag is set (ADRCIF.RCINT[n]="1"). The set state is retained until the range comparison interrupt factor flag is cleared (ADRCIF.RCINT[n]="0").
11. When range comparison execution is disabled (ADRCCS[n].RCOE="0"), the continuous detection count status (ADRCCS[n].RCOCD2 to RCOCD0) is initialized to "000_B".
    The range comparison interrupt factor flag (ADRCIF.RCINT[n]) and the threshold excess flag (ADRCOT.RCOOF[n]) are not cleared when range comparison execution is disabled (ADRCCS[n].RCOE="0").

5.3. Operation of the A/D Activation Arbitration

This section explains the operation of the A/D activation arbitration.

A/D activation requests from the A/D activation compare are arbitrated to generate an A/D activation trigger. In addition, it determines the analog channel for which A/D conversion is to be performed.

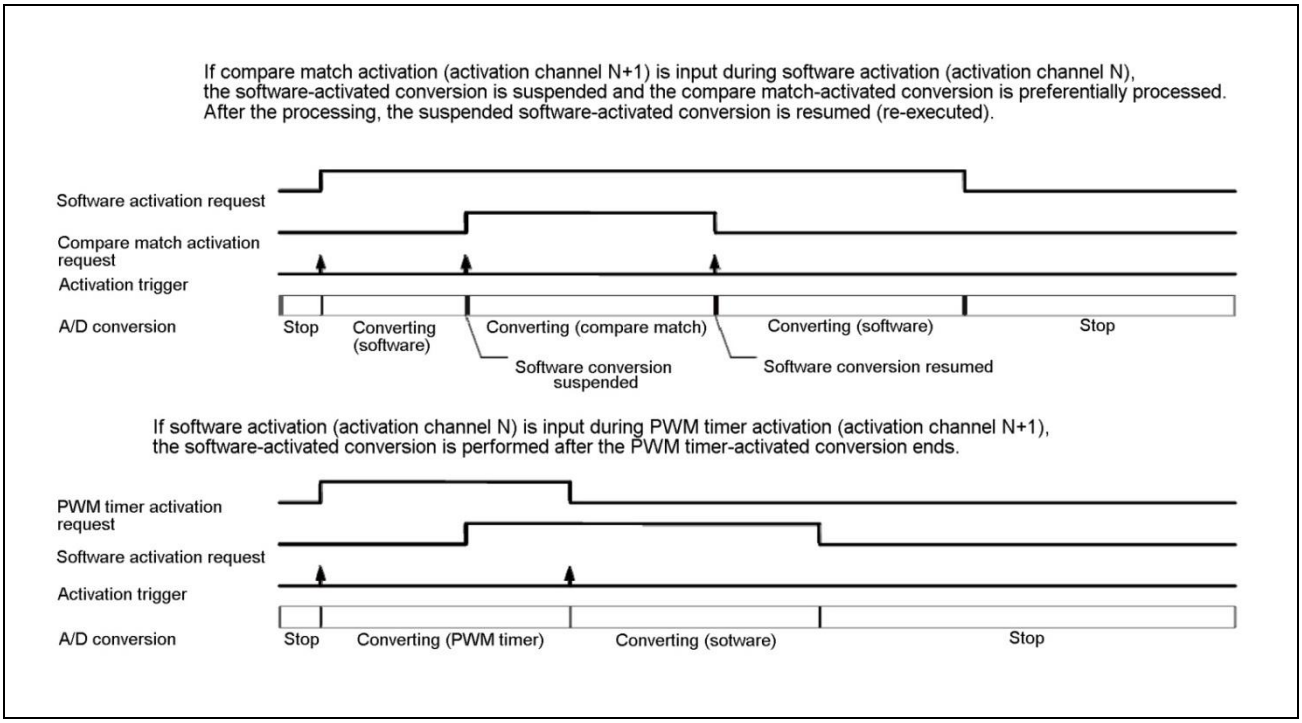
5.3.1. Arbitration for an A/D Activation Trigger

This section explains arbitration for an A/D activation trigger.

An A/D activation trigger is generated after one of activation requests from A/D activation compare channels is selected. 3 types of activation requests are input from each channel of the A/D activation compare: software activation request, reload timer activation request//PWM timer activation, and compare match activation request. Then, an A/D activation trigger signal is generated. If activation requests conflict with one another, one with the smaller compare channel number takes precedence. Activation requests that are not selected are kept waiting, and arbitration is again applied to them after the A/D conversion that is being processed is completed.
The priority based on the activation factor of the activation arbitration is [compare match activation request > reload timer activation request/PWM timer activation request > software activation request]. If there are activation factors with the same priority, one with a smaller activation channel number takes precedence.

- If activation factors with the same priority are generated while A/D conversion is stopped:
  The request from an activation channel with a smaller activation channel number is processed first.
- If activation factors with different priority are generated while A/D conversion is stopped:
  An activation factor with higher priority is processed first.
- If an activation factor with higher priority is generated while A/D conversion is in progress:
  The current conversion is suspended, and the activation factor with higher priority is processed. Then, arbitration is performed again after the conversion, and the suspended activation factor is processed again.
- If an activation factor with lower priority is generated while A/D conversion is in progress:
  Arbitration is performed again after the current conversion ends, and the activation factor with lower priority is processed.
- If an activation factor with the same priority is generated while A/D conversion is in progress:
  Arbitration is performed again after the current conversion ends, and the activation factor with the same priority is processed.

Figure 5-23 Activation Arbitration



5.3.2. Analog Channel Selection

This section explains analog channel selection.

Together with A/D activation requests, analog channel numbers for which A/D conversion is to be performed are input from the A/D activation compare.

The A/D activation arbitration selects the activation-request analog channel number of the selected A/D activation compare channel.

5.3.3. A/D Conversion Cancellation Function

This section explains the A/D conversion cancellation function.

This function generates an A/D conversion cancellation signal to forcibly terminate the current conversion processing when the activation request from the requester becomes inactive during A/D conversion. If an activation factor from a different activation channel is active when the activation request from the requester becomes inactive, no A/D conversion cancellation signal is generated but an A/D activation trigger is generated from the active activation factor.

5.4. Operation of the 12-Bit A/D Converter

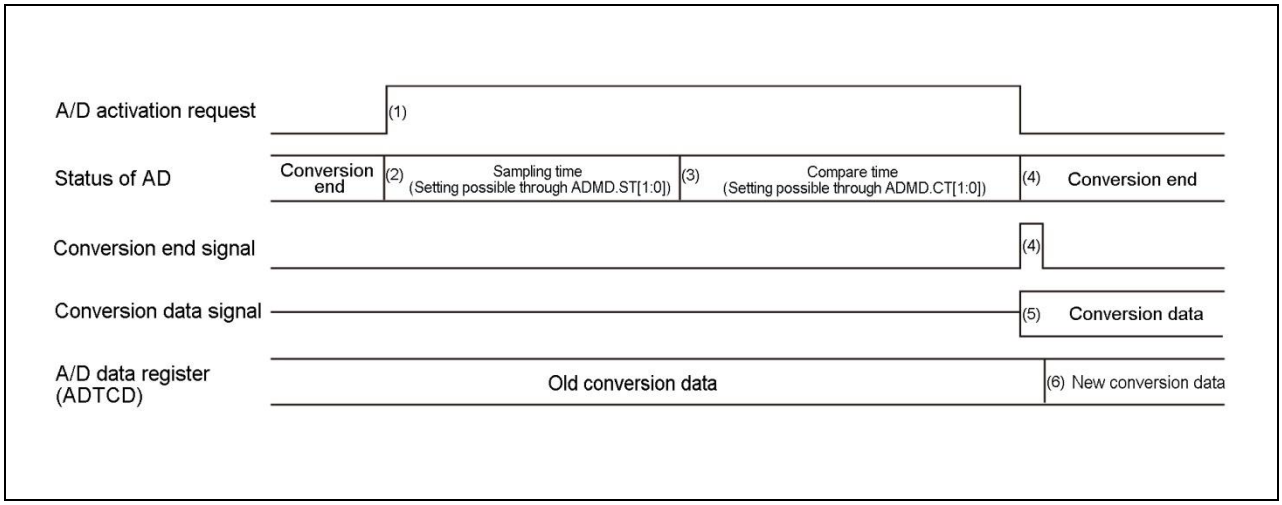
This section explains the operation of the 12-bit A/D converter.

The 12-bit A/D converter controls A/D conversion.

5.4.1. Operation Timing

This section explains the operation timing.

Figure 5-24 Operation Timing of the 12-Bit A/D Converter



(1) A/D conversion starts using the set activation factor.
(2) A sampling operation starts upon receiving the activation request described in (1).
(3) After the sampling time set by ADMD.ST[1:0] has elapsed, a compare operation starts.
(4) After the compare time set by ADMD.CT[1:0] has elapsed, a conversion end signal rises and the conversion is completed.
(5) A/D conversion data is output.
(6) New conversion data is stored in the A/D data register (ADTCD).

5.4.2. About Activation Factors

This section explains activation factors.

There are the following activation factors for A/D conversion: software activation, reload timer activation, compare match activation, and PWM timer activation. ADTECS.STS2 and ADTCS.STS[1:0] select the factor.

If you do not request A/D activation, set these bits to software activation ("000_B") and set an appropriate value in the relevant bit in ADTSE to disable software activation (ADTSE.ADT=0).

5.4.3. About A/D Conversion

This section explains A/D conversion.

1 sequence of A/D conversion is performed in response to 1 activation trigger.

5.4.4. About Reactivation

This section explains reactivation.

If an activation trigger signal is input during A/D conversion, the current conversion is stopped/initialized and the A/D conversion is reactivated.
Thus, the start of A/D conversion reactivation is delayed by several clocks, compared to the normal activation (the start of A/D conversion while A/D conversion is stopped).

5.4.5. About A/D Conversion Cancellation

This section explains A/D conversion cancellation.

If an A/D conversion cancellation signal is received during A/D conversion, the current conversion is stopped/initialized.

5.4.6. Analog Channel Selection Control

This section explains the analog channel selection control.

Besides an activation trigger, information on the analog channel for which conversion is to be performed is input.

The analog channel information at the point when the activation trigger is active is retained and used for analog channel selection.

5.4.7. About A/D Conversion Time

This section explains the A/D conversion time.

The A/D conversion time is the sum of sampling time and compare time.

Sampling Time

For sampling time, you can select whether sampling is to be performed for each channel or using a common setting through the sampling time setting per channel enable bit (ADMD.STPCEN).

- If ADMD.STPCEN is "0", sampling time common to all channels is used. Use the sampling time setting bits (ADMD.ST1, ST0) to set sampling time.
- If ADMD.STPCEN is "1", you can set sampling time for each channel. Use the sampling time setting per channel bits (ADSTPCS.STCHn1, STCHn0: n=00 to 63) to set sampling time for each channel.

Table 5-12 Sampling Time Corresponding to Peripheral Clock Frequency

ST1, STCHn1	ST0, STCHn0	Function	Sampling Time (Peripheral Clock Frequency)			
			(40 MHz)	(32 MHz)	(24 MHz)	(16 MHz)
0	0	12 peripheral clock cycles	300 ns	375 ns	500 ns	750 ns
0	1	18 peripheral clock cycles	450 ns	562.5 ns	750 ns	1125 ns
1	0	24 peripheral clock cycles	600 ns	750 ns	1000 ns	1500 ns
1	1	48 peripheral clock cycles	1200 ns	1500 ns	2000 ns	3000 ns

Note:

- *Be sure to overwrite these bits during A/D operation before the conversion operation has stopped.*

Compare Time

- Set compare time in the compare time setting bits (ADMD.CT1, CT0).

Table 5-13 Compare Time Corresponding to Peripheral Clock Frequency

CT1	CT0	Function	Compare Time (Peripheral Clock Frequency)			
			(40 MHz)	(32 MHz)	(24 MHz)	(16 MHz)
0	0	28 peripheral clock cycles	700 ns	875 ns	1166.7 ns	1750 ns
0	1	42 peripheral clock cycles	1050 ns	1312.5 ns	1750 ns	2625 ns
1	0	56 peripheral clock cycles	1400 ns	1750 ns	2333.4 ns	3500 ns
1	1	112 peripheral clock cycles	2800 ns	3500 ns	4666.7 ns	7000 ns

Note:

- *Be sure to overwrite these bits during A/D operation before the conversion operation has stopped.*

5.4.8. A/D Conversion End and A/D Data Capture

This section explains the A/D conversion end and A/D data capture.

When A/D conversion ends (the predetermined cycles have elapsed) normally without reactivation or cancellation, the received conversion data is captured and output. At this time, an A/D conversion end signal is generated.

5.4.9. Power Down

This section explains power down.

Power down is in effect during in PSS mode.

Note:

- *Before a transition to the PSS mode occurs, stop AD conversion.*

5.5. Failure Diagnosis Function of the 12-Bit A/D Converter

This section explains the failure diagnosis function of the 12-bit A/D converter.

The 12-bit A/D converter has a failure diagnosis function: AVRH and AVRL voltage conversion diagnosis.

5.5.1. AVRH and AVRL Voltage Conversion Diagnosis

This section explains AVRH and AVRL voltage conversion diagnosis.

This product has the function of fixing the input voltage level of the internal A/D converter to the AVRH or AVRL level and perform A/D conversion.

The following registers can set AVRH level voltage input.

- 12-bit A/D converter unit 0:
  AVRH0SEL bit in the failure diagnosis function control register (ADFDCTR0)
- 12-bit A/D converter unit 1:
  AVRH1SEL bit in the failure diagnosis function control register (ADFDCTR1)

The following registers can set AVRL level voltage input.

- 12-bit A/D converter unit 0:
  AVRL0SEL bit in the failure diagnosis function control register (ADFDCTR0)
- 12-bit A/D converter unit 1:
  AVRL1SEL bit in the failure diagnosis function control register (ADFDCTR1)

For 12-bit A/D converter unit 0,

if you set AVRH0SEL or AVRL0SEL to "1", the analog input from the A/D converter analog input pin ANx (x: integer number) becomes invalid as shown in Figure 5-25.
This is also applicable to 12-bit A/D converter unit 1.

Figure 5-25 Configuration of the Analog Input Switching Circuit of A/D Converter Unit 0 (AN=0 to 31)

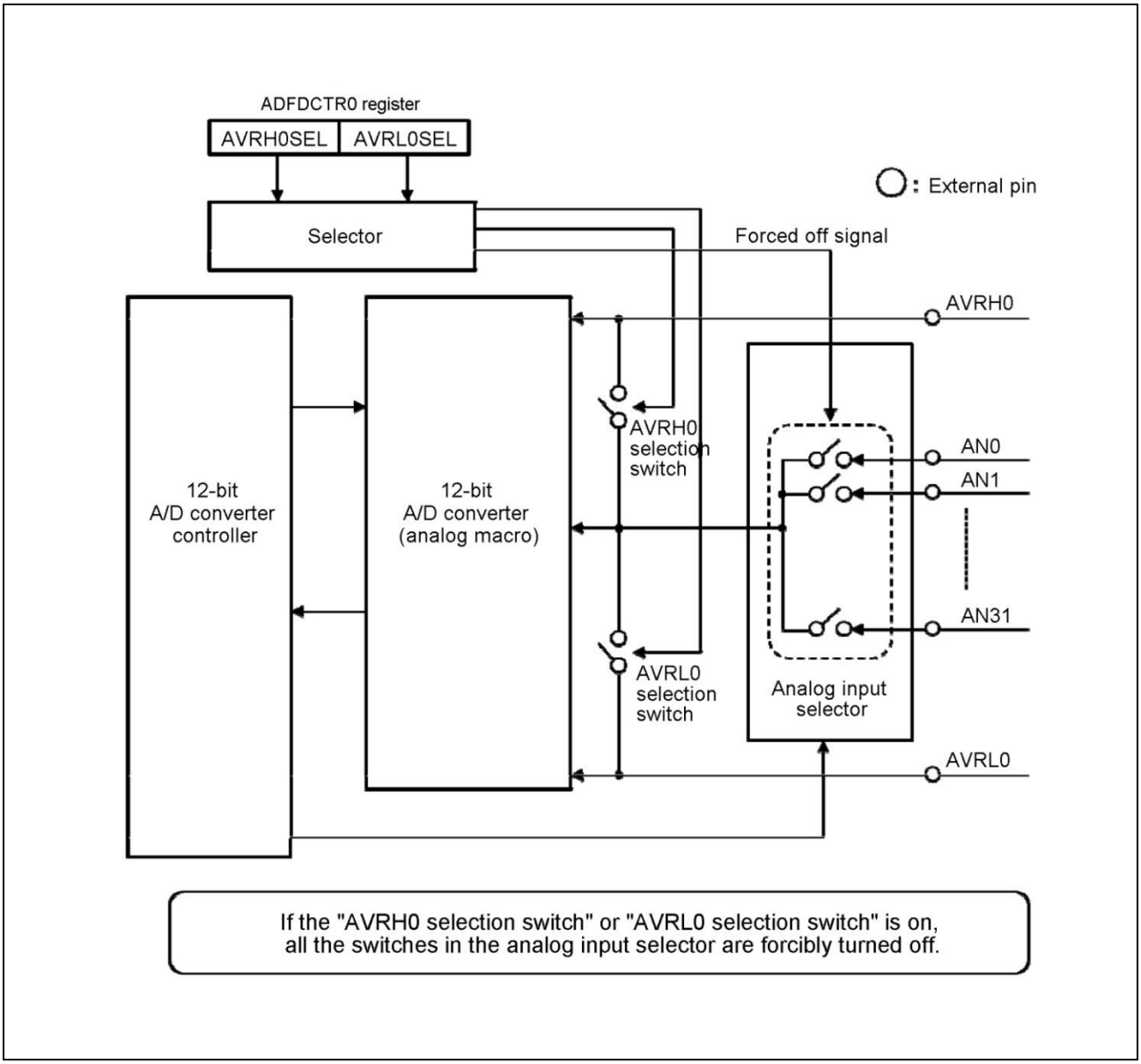
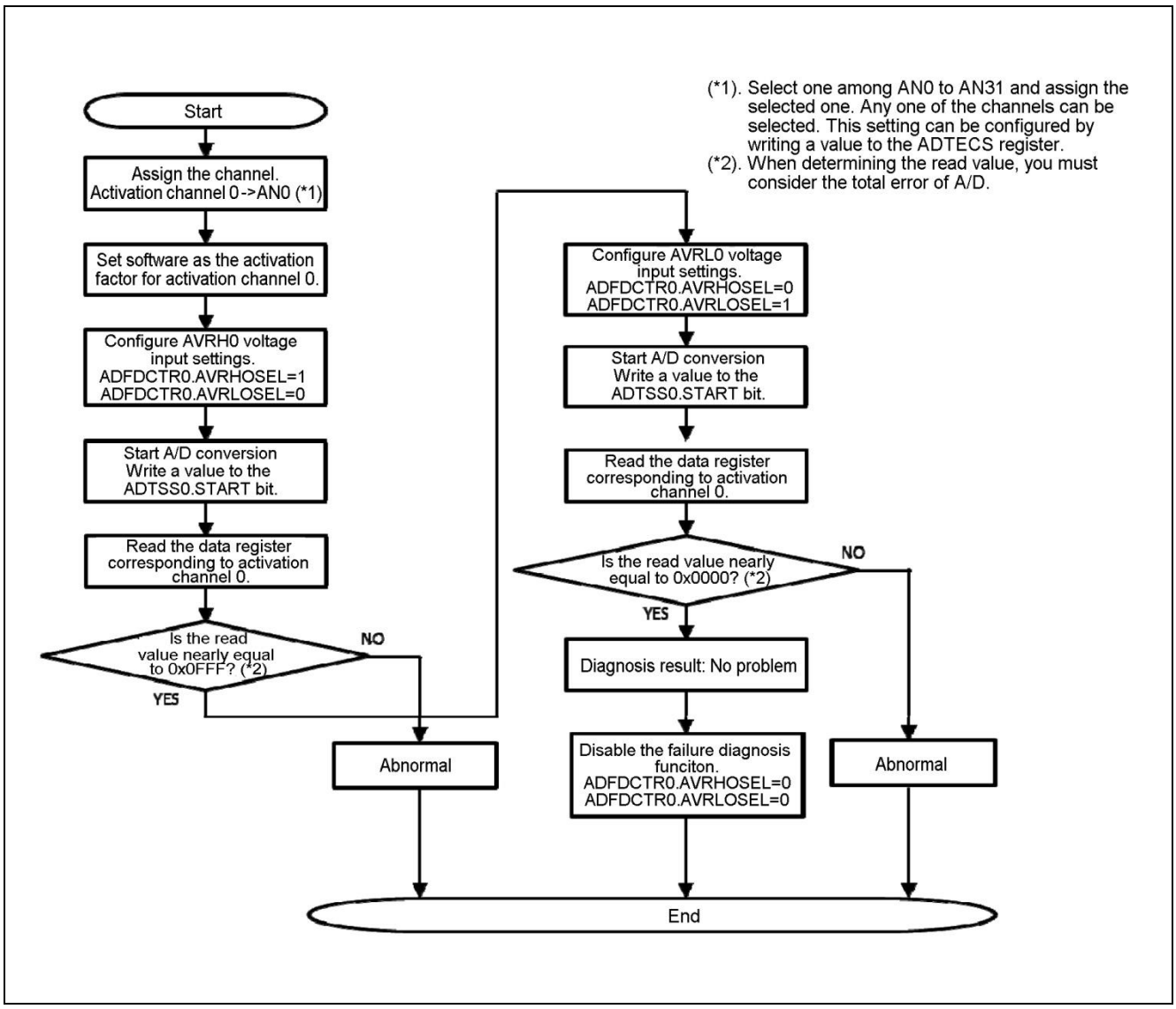


Figure 5-26 shows an example of a failure diagnosis program that uses this function.

Figure 5-26 Example of a Failure Diagnosis Program for Diagnosis of A/D Converter Unit 0 (AN0 to 31)



6. Notes

This section provides notes.

Notes on the A/D Activation Compare Usage

- About the selection setting

  Be sure to configure the selection setting while the free-run timer is stopped.

- About 32-bit free-run timer settings

  Although the 32-bit free-run timer can count a 32-bit value, only the value of the lower 16 bits can be compared with the compare register. If you use the 32-bit free-run timer as an activation trigger for the A/D converter, set the upper 16 bits of the compare clear register (CPCLR) to "0000_H". This prevents a carry to the upper 16 bits of the free-run timer.

- About the protection settings of A/D data registers

  Set a value in the PRT and PRTS bits before starting A/D conversion operation. Do not change the settings of these bits while A/D conversion is in progress and the A/D data register is protected.

  To cancel the protection function of an A/D data register, stop A/D conversion, and then perform the protection cancellation operation set by the PRTS bit or disable the protection function using the PRT bit.

  Suppose that you changed the PRTS bit while the A/D data register is protected. In this case, to cancel the protection of the A/D data register, perform the protection cancellation operation (reading the A/D data register or a clear operation by writing "0" to the interrupt request flag bit) set by the PRTS bit after changing the PRTS bit. For example, while PRT is 1, PRTS is 1, and the A/D data register is protected, you clear the interrupt request flag bit and then set PRTS to 0. In this case, to cancel the protection, you must read the A/D data register and perform a clear operation again by writing "0" to the interrupt request flag bit.

- About compare match activation

  If 0000_H and the same value as the value set in the compare clear register of the free-run timer are set in the compare register (ADCOMP), an A/D activation request signal is generated at the time of compare match. This is regardless of whether the count direction set by the count direction selection bits (SEL0 and SEL1 bits) in the A/D activation trigger control register (ADTCS) is up or down.

- About the setting of the count direction selection bits in the A/D activation trigger control status register

  When the free-run timer is in up count mode, it is prohibited to set the count direction selection bits (SEL0 and SEL1 bits) in the A/D activation trigger control status register (ADTCS) to (SEL1, SEL0)=(1, 0) (only down-counting).

- About range comparison

  The execution timing of range comparison is after A/D conversion result is set in the A/D data register (ADTCD) when A/D conversion ends. Thus, reflection of range comparison result is delayed by 2 cycles (peripheral clock) from the point when an A/D conversion end interrupt request is generated. Then, the result is reflected to the RCOCD2 to RCOCD0 bits, RCOOF bit, and RCINT bit, and a range comparison interrupt request is generated.

  When an outside-range check is performed for range comparison, the continuous detection measurement is not cleared to 0 times and the continuous detection continues. This is applicable even if range comparison result changes from the "upper threshold exceeded" state to the "under lower threshold" state.

  If you want to initialize the continuous detection count status of range comparison result, disable

range comparison execution and then enable it (ADRCCS.RCOE="0" -> "1") while A/D conversion is not being requested.

- About continuous scan conversion without conversion count specification for each channel

  If the repeat mode is set for multiple activation channels that correspond to the same A/D converter unit, only a certain channel is processed in order of priority by the A/D activation arbitration described later. Thus, if the repeat mode is set for multiple activation channels, use the activation channels after enabling the data protection function.

  Note the following when you set the repeat mode for multiple activation channels that correspond to the same A/D converter unit: If the data protected state of an activation channel with higher priority is canceled before conversion for an activation channel with lower priority, the conversion for the activation channel with lower priority is not performed. Thus, if you cancel the data protected state before the conversion for the last activation channel is performed for the continuous scan conversion, the conversion for the activation channel with lower priority is not performed and the scan conversion is not performed normally.

- About continuous scan conversion when the conversion count is specified for each channel

  Set the repeat conversion mode for activation channels for which execution of conversion count-specified scan conversion is enabled.

  In continuous scan conversion mode when the conversion count is specified for each channel, the EOCF bit set for the last channel of the scan conversion is cleared to "0" immediately after it is set to "1". This is because the scan conversion restarts from the beginning immediately after the scan conversion ends.

  Suppose the following case in continuous scan conversion mode when the conversion count is specified for each channel: The same activation factor as for the channels for which scan conversion is to be performed is set for an activation channel larger than the last channel of the continuous scan conversion. In this case, the activation channel larger than the last channel is evaluated once by the A/D activation arbitration described later after the last channel of the continuous scan conversion is completed.

  Suppose the following case in pause scan conversion mode when the conversion count is specified for each channel: The same activation factor as for the channels for which scan conversion is to be performed is set for an activation channel larger than the last channel of the pause scan conversion. In this case, A/D conversion for the activation channel larger than the last channel can be performed only during a pause period.

  While the data protection function is enabled, there is no restriction on the cancellation timing of the data protected state. However, if the activation channel for which scan conversion is performed is in the data protected state, the scan conversion is stopped until the data protected state is canceled.

- About the settings of the ADMD register and ADSTPCS

  Be sure to rewrite bits in the A/D mode setting register (ADMD) and the sampling time setting register (ADSTPCS) before A/D conversion operation is performed, while the A/D conversion operation is stopped.

- About reset

  A/D converter unit 0 is reset at power-off and register values are initialized. However, A/D converter unit 1 is not reset at power-off and register values are not initialized.

CHAPTER 57: LCD Controller

This chapter explains the LCD controller.

1. Overview
2. Features
3. Configuration
4. Operation
5. Setting
6. Registers
7. Q&A
8. Sample Program
9. Notes

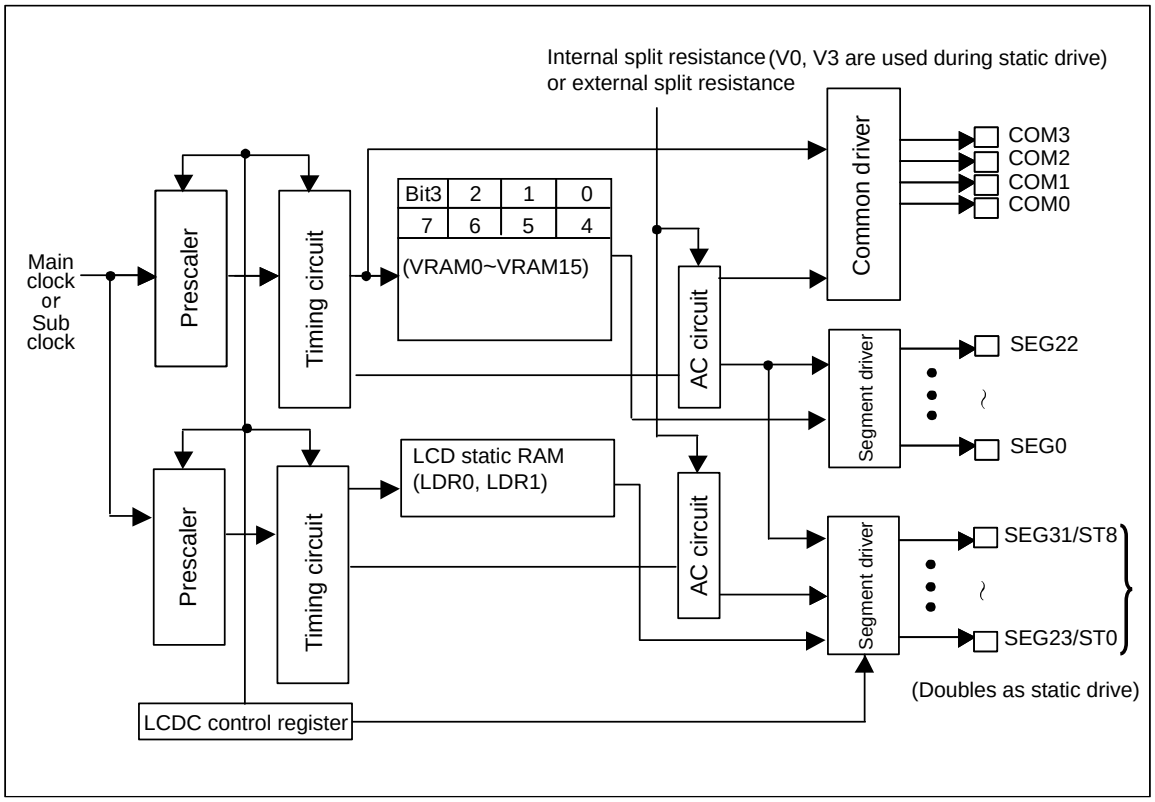
LCDC-T01P00R02L01-E1-GBM02

1. Overview

This section explains the overview of the LCD controller.

The LCD enables the selection of duty from 1/2, 1/3, and 1/4, and up to 128-element display.

Furthermore, it enables up to 8-element display as static LCD output.



2. Features

This section explains the features of the LCD controller.

<Duty drive>

- Number: 1 (4-common × 32-segment)
- Display: Up to 128-element (on 1/4 duty)
- Duty: Selection from three types (1/2, 1/3, 1/4)
- Bias: Selection from 1/2 and 1/3

Combination of Bias, Duty, and Common Output

Bias	1/2 Duty Output Mode	1/3 Duty Output Mode	1/4 Duty Output Mode
1/2 bias	○	×	×
1/3 bias	×	○	○

○ : Recommended mode

× : Prohibited

- Frame cycle: Selectable from four types (For a clock, a main clock or sub clock can be used.)
- Driver: Built-in (Internal division resistor), or an external division resistor can be connected to V0 to V3 pins.
- Data memory: Built-in 16-byte data memory for display
- No display selection: Enabled
  - Pin: COM0 to COM3, SEG0 to SEG31, V0, V1, V2, V3 pins are used for general-purpose ports also and switching is enabled.
    SEG23 to SEG31 pins are used as static drive pins (ST0 to ST8) also and switching is enabled.
    VCC can be selected instead of V3.
- During duty driving, up to 4 common outputs (COM0 to COM3) and 32 segment outputs (SEG0 to SEG31) can be used.
  - Others: The external division resistor also can block a current when LCD is stopped.

< Static drive >

- Number: 1 (1-common × 8-segment)
- Display: Up to 8 element
- Frame cycle: Selectable from four types (For a clock, a main clock or sub clock can be used.)
- Pin: ST0 to ST8 pins are used also as general-purpose ports and duty drive pins (SEG23 to SEG31) and switching is enabled.
- During static driving, up to 1 static common output and 8 static segment outputs (ST0 to ST8) can be used.

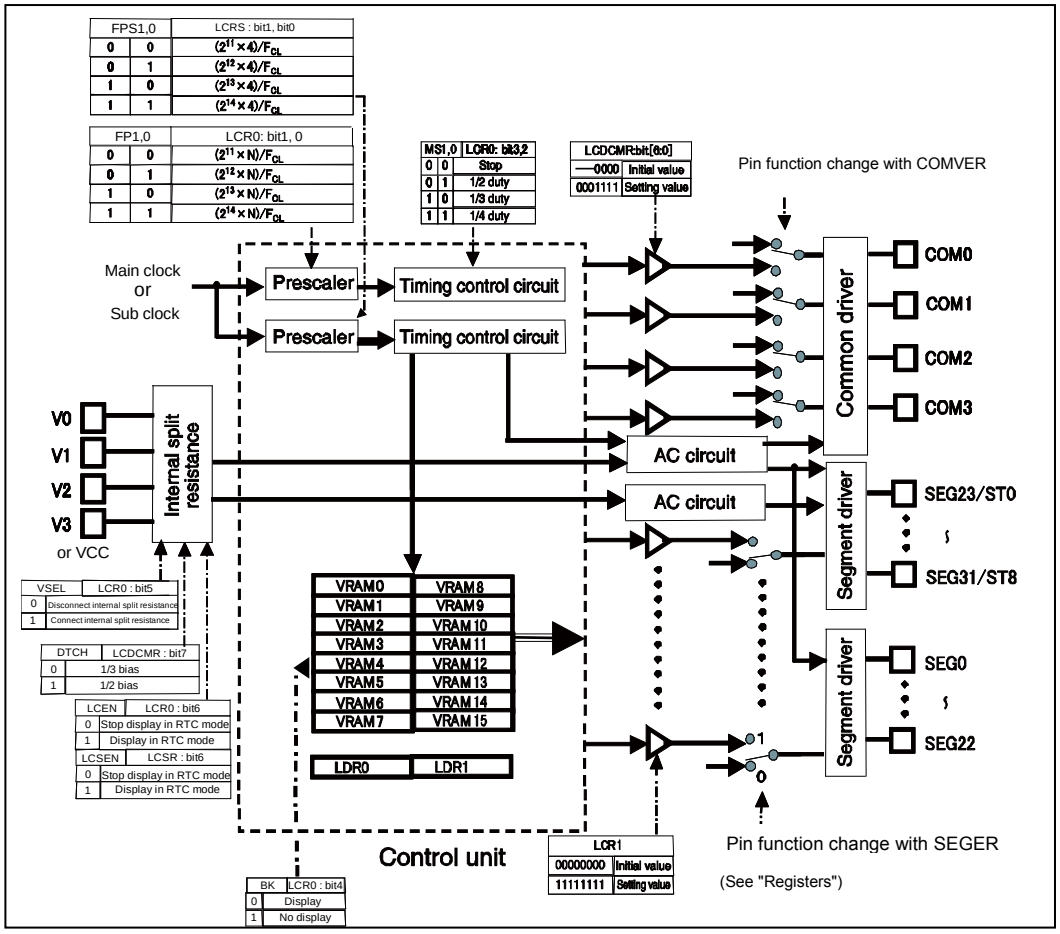
3. Configuration

This section shows the configuration of the LCD controller.

■ Configuration Diagram of LCD Controller

Figure 3-1 shows the configuration diagram of the LCD controller.

Figure 3-1 Configuration Diagram



4. Operation

The following sections explain the operation of the LCD controller.

4.1. Operation of LCD Controller/Driver (LCDC)

4.2. 1/2 Duty Output Waveform

4.3. 1/3 Duty Output Waveform

4.4. 1/4 Duty Output Waveform

4.5. Static Drive Output Waveform

4.1. Operation of LCD Controller/Driver (LCDC)

The operation of LCD controller/driver (LCDC) is shown below.

1. **Data Memory for Display**

<Duty drive>

Set a value to the data memory for display (VRAM) in advance.

<Static drive>

Set a value to the data memory for display (LDR0, LDR1) in advance.

2. **Write Necessary Settings to Each Register.**

3. **Output Pin**

<Duty drive>

When a clock for frame cycle generation oscillates, waveforms that drive the LCD are output to the common/segment output pins (COM0 to COM3, SEG0 to SEG31).

The content of the VRAM is read automatically in synchronization with the timing of the common signal and output from the segment output pin.

(When the bit of VRAM is set to "1", a selection waveform is output from the segment output pin.

When the bit of VRAM is set to "0", a non-selection waveform is output from the segment output pin.)

Non-selection level waveforms are output from the COM2 and COM3 pins in 1/2 duty display mode and from the COM3 pin in 1/3 duty display mode.

<Static drive>

When a clock for frame cycle generation oscillates, waveforms that drive the LCD are output to the common/segment output pins (ST0 to ST8).

4. **Output Waveform**

<Duty drive>

Output waveforms drive in 2 frames alternating waveform according to the duty setting.

<Static drive>

Output waveforms drive in 1 frame alternating waveform.

5. **Operation in the PSS Timer Mode (Main Oscillation Operation/ Sub Oscillation Operation)**

<Duty drive>

For operation enable (LCEN = "1"), the LCD is displayed.

Please do not stop the clock oscillation which selected by the CSS bit during the transition to PSS mode.

LCD display stop when power shutdown.

<Static drive>

For operation enable (LCSEN = "1"), the LCD is displayed.

Please do not stop the clock oscillation which selected by the SCSS bit during the transition to PSS mode.

LCD display stop when power shutdown

6. Blanking Function

<Duty drive>

Light of the LCD can be turned off with selection of non-display (BK = "1") of blanking.

However, non-selection waveforms are output.

7. LCD Stop State

<Duty drive>

When display operation of the LCD is stopped (MS [1:0] = "00"), both the common/segment output pins become "L" level.

<Static drive>

When display operation of the LCD is stopped (LCS [3:0] = "0000"), both the common/segment output pins become "L" level.

When the LCD with static drive (ST0 to ST8) is switched from display to non-display, set LDR0:ST8 = "0" and LDR1: ST [7:0] = "00000000" without change of the static drive selection port (LCS [3:0]).

The same potential pulses are output from the static drive pins (ST0 to ST8).

4.2. 1/2 Duty Output Waveform

1/2 duty output waveform is shown below.

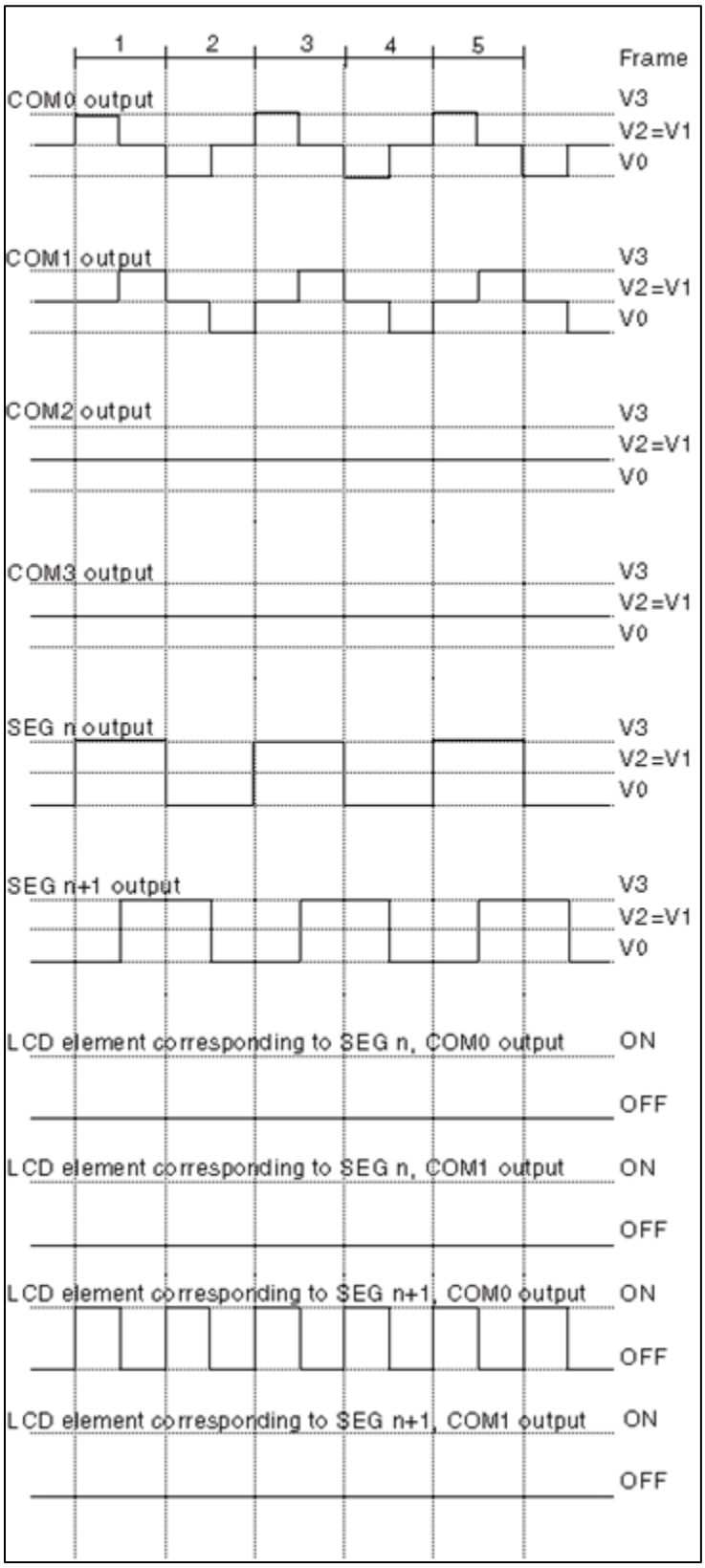
Only COM0 output and COM1 output are used for LCD display. COM2 output and COM3 output are not used.

1/2 Bias Output Waveform Case

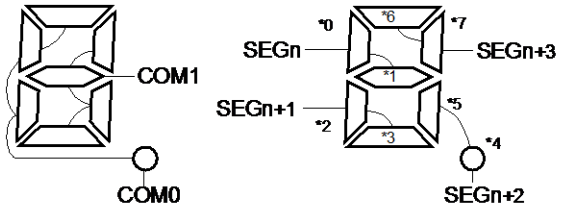
Liquid crystal elements that have maximum difference in potential between the common output and segment output light up.

■ Example of Content of Data Memory for Display

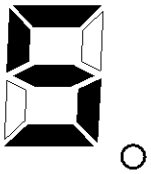
Segment	Content of Data Memory for Display			
	COM3 Output	COM2 Output	COM1 Output	COM0 Output
SEG n output	-	-	0	0
SEG n+1 output	-	-	0	1


1
2
3
4
5
Frame
COM0 output
COM1 output
COM2 output
COM3 output
SEG n output
SEG n+1 output
V3
V2=V1
V0
LCD element corresponding to SEG n, COM0 output
LCD element corresponding to SEG n, COM1 output
LCD element corresponding to SEG n+1, COM0 output
LCD element corresponding to SEG n+1, COM1 output
ON
OFF

■ LCD panel connection case and display data case (1/2 duty drive method)



Example for displaying 5:

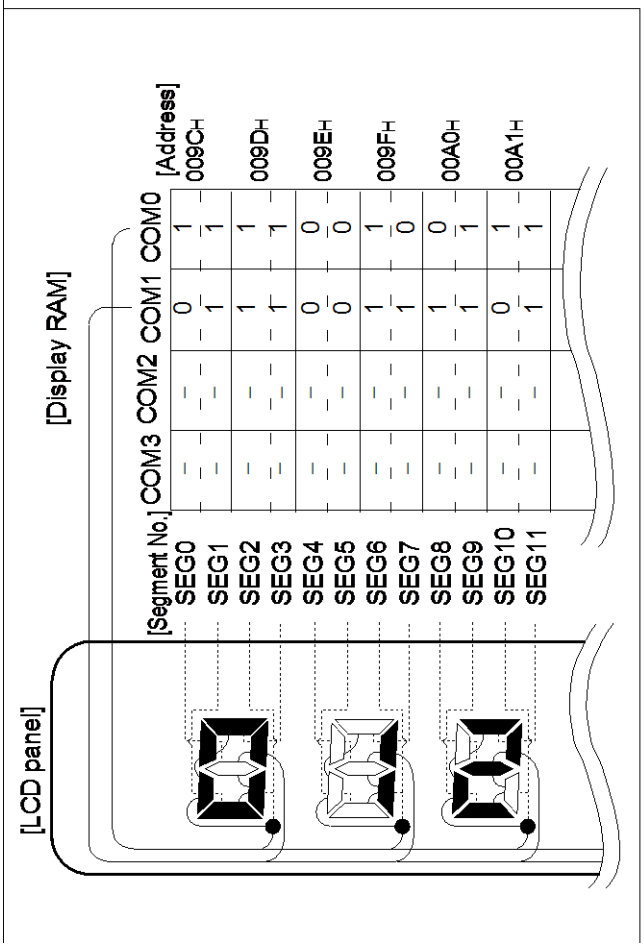


Address	COM3	COM2	COM1	COM0	
nH	bit3	bit2	bit1*1	bit0*0	SEGn
	bit7	bit6	bit5*3	bit4*2	SEGn+1
n+1H	bit3	bit2	bit1*5	bit0*4	SEGn+2
	bit7	bit6	bit5*7	bit4*6	SEGn+3

Address	COM3	COM2	COM1	COM0	
3950H	–	–	1	1	SEG0
	–	–	1	0	SEG1
3951H	–	–	1	0	SEG2
	–	–	0	1	SEG3

*0 to *7 : Indicates correspondence with display RAM bit2, 3, 6, 7 are unused

0 : OFF
1 : ON



LCD display	Example of data correspondence from 0 to 9							
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0	–	–	1	1	–	–	0	1
	–	–	1	1	–	–	1	1
1	–	–	0	0	–	–	0	0
	–	–	1	0	–	–	1	1
2	–	–	1	1	–	–	1	0
	–		1	1			0	1
3	–	–	1	0	–	–	1	0
	–	–	1	1	–	–	1	1
4	–	–	0	0	–	–	1	1
	–	–	1	0	–	–	1	1
5	–	–	1	0	–	–	1	1
	–	–	0	1	–	–	1	1
6	–	–	1	1	–	–	1	1
	–	–	0	1	–	–	1	1
7	–	–	0	0	–	–	0	1
	–	–	1	1	–	–	1	1
8	–	–	1	1	–	–	1	1
	–	–	1	1	–	–	1	1
9	–	–	1	0	–	–	1	1
	–	–	1	1	–	–	1	1

4.3. 1/3 Duty Output Waveform

1/3 duty output waveform is shown below.

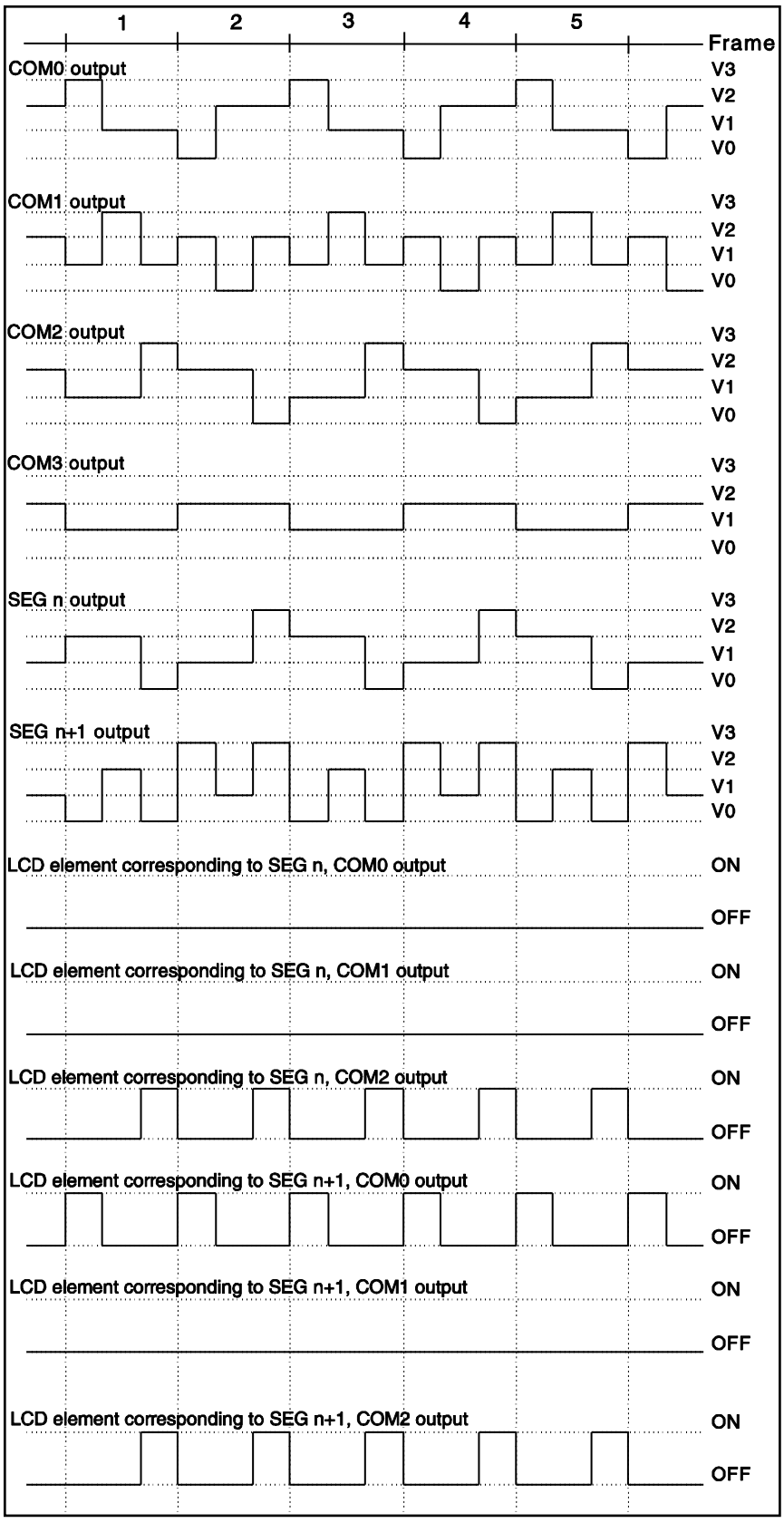
For 1/3 duty output mode, COM0 output, COM1 output, and COM2 output are used for LCD display. COM3 output is not used.

1/3 Bias Output Waveform Case

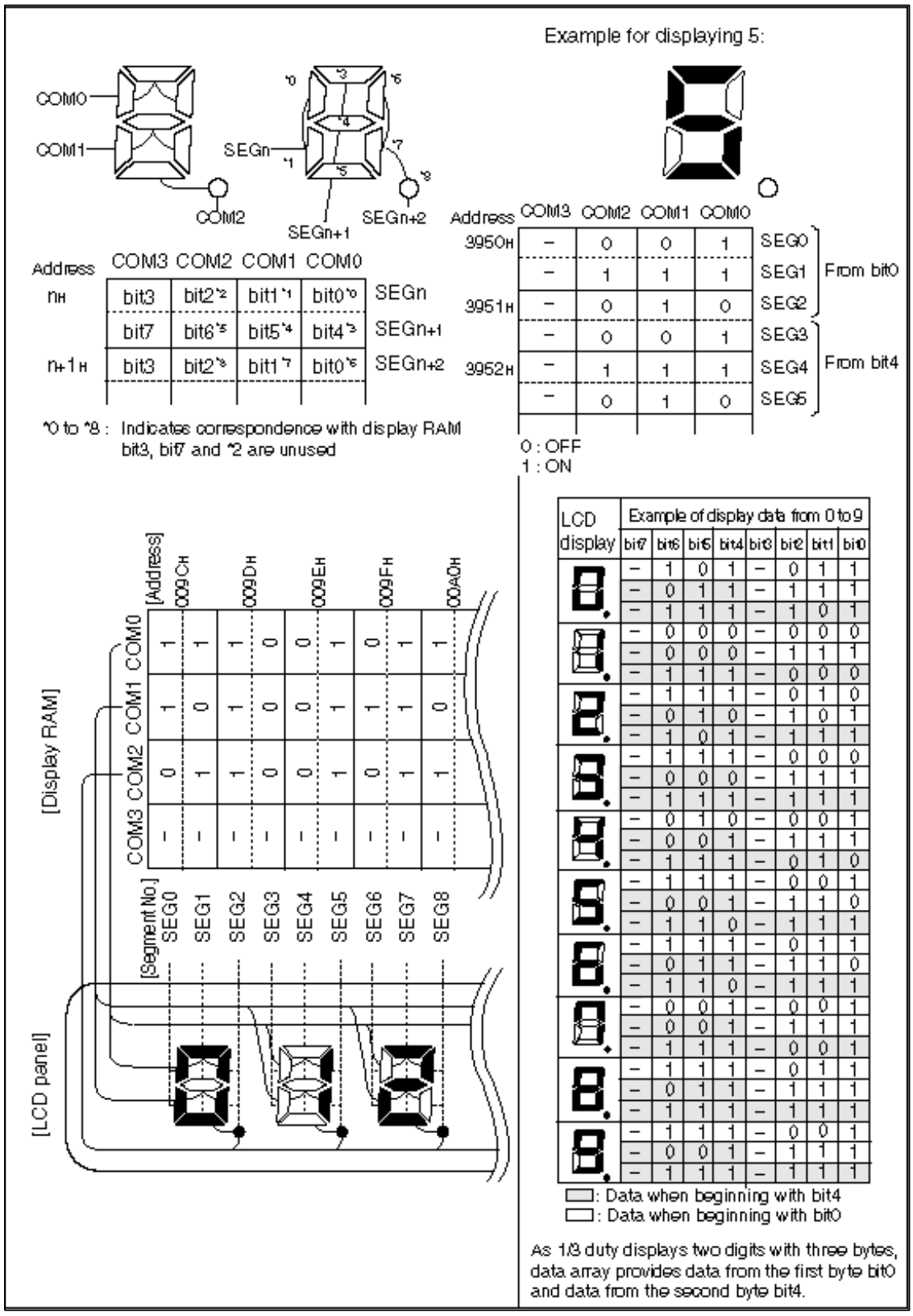
Liquid crystal elements that have maximum difference in potential between the common output and segment output light up.

■ Example of Content of Data Memory for Display

Segment	Content of Data Memory for Display			
	COM3 Output	COM2 Output	COM1 Output	COM0 Output
SEG n output	-	1	0	0
SEG n+1 output	-	1	0	1


1
2
3
4
5
Frame
COM0 output
V3
V2
V1
V0
COM1 output
V3
V2
V1
V0
COM2 output
V3
V2
V1
V0
COM3 output
V3
V2
V1
V0
SEG n output
V3
V2
V1
V0
SEG n+1 output
V3
V2
V1
V0
LCD element corresponding to SEG n, COM0 output
ON
OFF
LCD element corresponding to SEG n, COM1 output
ON
OFF
LCD element corresponding to SEG n, COM2 output
ON
OFF
LCD element corresponding to SEG n+1, COM0 output
ON
OFF
LCD element corresponding to SEG n+1, COM1 output
ON
OFF
LCD element corresponding to SEG n+1, COM2 output
ON
OFF

■ LCD panel connection case and display data case (1/3 duty drive method)



4.4. 1/4 Duty Output Waveform

1/4 duty output waveform is shown below.

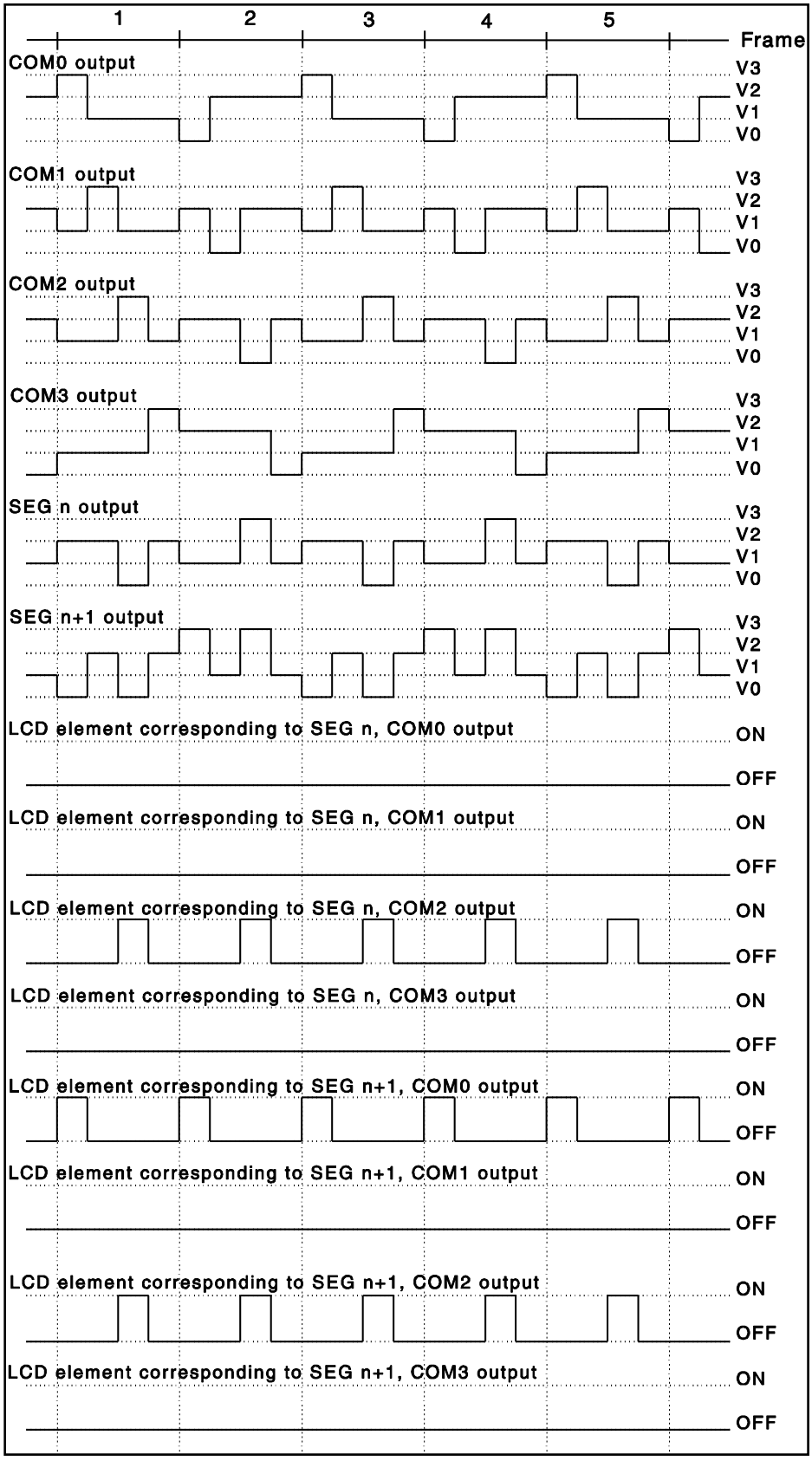
For 1/4 duty output mode, all of COM0 output, COM1 output, COM2 output, and COM3 output are used for LCD display.

1/4 Bias Output Waveform Case

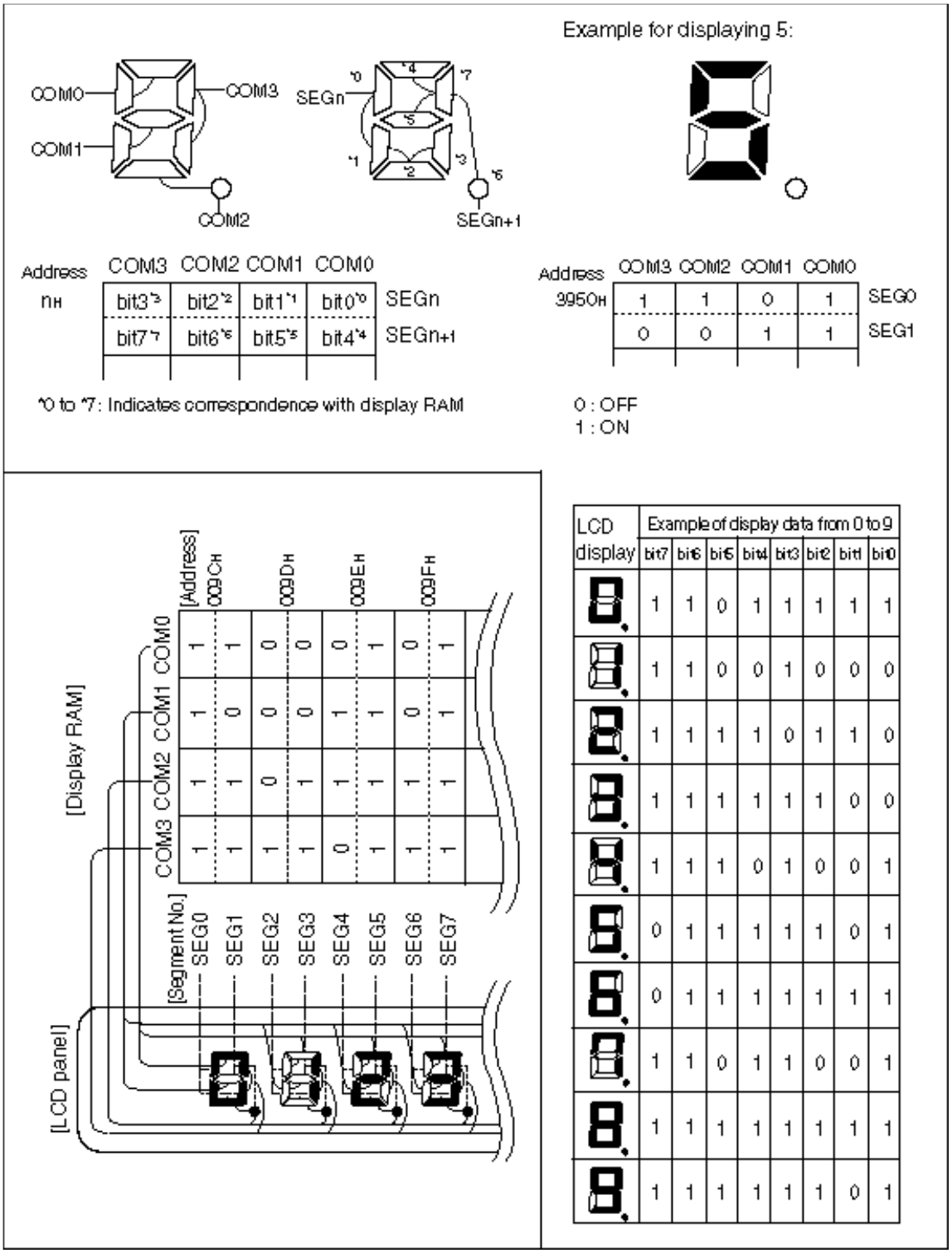
For LCD, liquid crystal elements that have maximum difference in potential between the common output and segment output light up.

■ Example of Content of Data Memory for Display

Segment	Content of Data Memory for Display			
	COM3 Output	COM2 Output	COM1 Output	COM0 Output
SEG n output	0	1	0	0
SEG n+1 output	0	1	0	1


1
2
3
4
5
Frame
COM0 output
V3
V2
V1
V0
COM1 output
V3
V2
V1
V0
COM2 output
V3
V2
V1
V0
COM3 output
V3
V2
V1
V0
SEG n output
V3
V2
V1
V0
SEG n+1 output
V3
V2
V1
V0
LCD element corresponding to SEG n, COM0 output
ON
OFF
LCD element corresponding to SEG n, COM1 output
ON
OFF
LCD element corresponding to SEG n, COM2 output
ON
OFF
LCD element corresponding to SEG n, COM3 output
ON
OFF
LCD element corresponding to SEG n+1, COM0 output
ON
OFF
LCD element corresponding to SEG n+1, COM1 output
ON
OFF
LCD element corresponding to SEG n+1, COM2 output
ON
OFF
LCD element corresponding to SEG n+1, COM3 output
ON
OFF

■ LCD panel connection case and display data case (1/4 duty drive method)



LCD display	Example of display data from 0 to 9							
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0	1	1	0	1	1	1	1	1
1	1	1	0	0	1	0	0	0
2	1	1	1	1	0	1	1	0
3	1	1	1	1	1	1	0	0
4	1	1	1	0	1	0	0	1
5	0	1	1	1	1	1	0	1
6	0	1	1	1	1	1	1	1
7	1	1	0	1	1	0	0	1
8	1	1	1	1	1	1	1	1
9	1	1	1	1	1	1	0	1

4.5. Static Drive Output Waveform

Static drive output waveform is shown below.

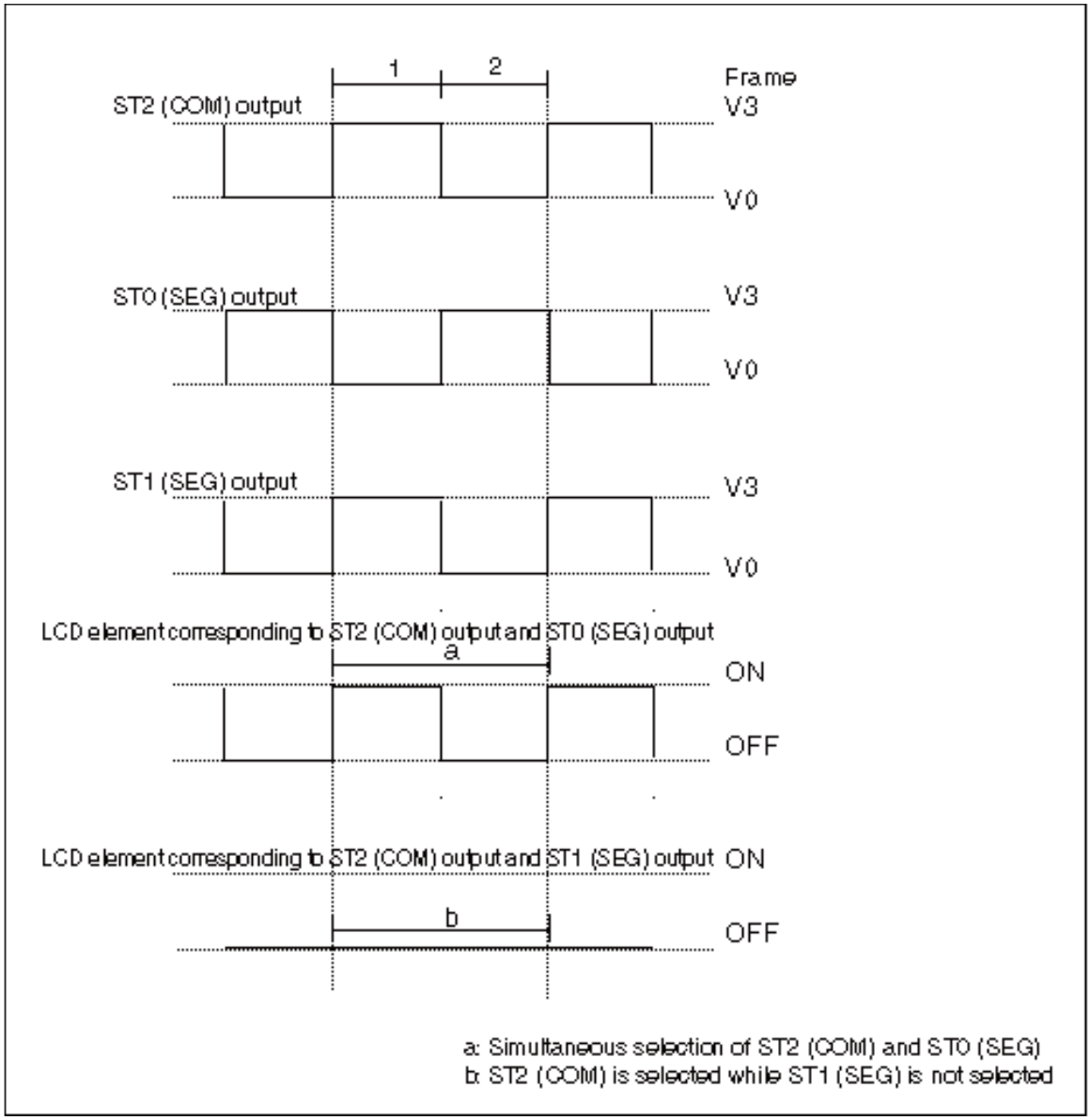
For the static drive output mode, the common/segment output pins (ST0 to ST8) are used for LCD display.

Static Drive Output Waveform Case

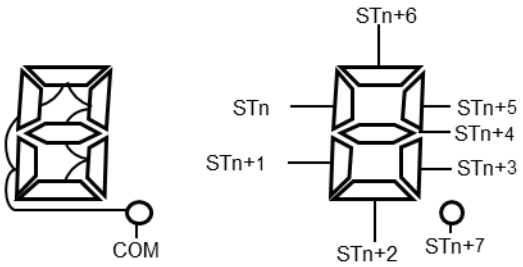
For LCD, liquid crystal elements that have maximum difference in potential between the common output and segment output light up.

- Example of Content of Data Memory for Display (Example of static SEG output to the output pins ST0 to ST1 and static COM output to ST2)

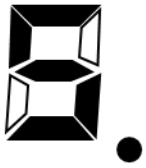
Content of Data Memory for Display (LDR0, LDR1)		
ST2 Output LDR0[2]	**ST1 Output LDR0[1]**	**ST0 Output LDR0[0]**
0	0	1


1
2
Frame
ST2 (COM) output
V3
V0
ST0 (SEG) output
V3
V0
ST1 (SEG) output
V3
V0
LCD element corresponding to ST2 (COM) output and ST0 (SEG) output
a
ON
OFF
LCD element corresponding to ST2 (COM) output and ST1 (SEG) output
ON
b
OFF
a: Simultaneous selection of ST2 (COM) and ST0 (SEG)
b: ST2 (COM) is selected while ST1 (SEG) is not selected

■ LCD panel connection case and display data case (Static drive method)



Example for displaying 5:



Register Address		—	—	—	—	—	—	—	ST8 (COM)
LCR0	00ACH	—	—	—	—	—	—	—	0

Register Address		ST7	ST6	ST5	ST4	ST3	ST2	ST1	ST0
LCR1	00ADH	1	1	0	1	1	1	0	1

SEG

0 : OFF
1 : ON

LCD display	Example of data correspondence from 0 to 9 (LCR1)							
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	1	1	1	0	1	1	1	1
	0	0	1	0	1	0	0	0
	1	1	1	1	0	1	1	0
	1	1	1	1	1	1	0	0
	1	0	1	1	1	0	0	1
	1	1	0	1	1	1	0	1
	1	1	0	1	1	1	1	1
	1	1	1	0	1	0	0	1
	1	1	1	1	1	1	1	1
	1	1	1	1	1	1	0	1

5. Setting

This section shows the setting of the LCD controller.

<Duty drive>

Setting Necessary for Use of LCD

Setting	Set Register	Set Method
Preliminary setting	LCD control register 1 (LCR1) Common pin switching register (LCDCMR)	See 6.3, 6.4
Division resistor setting	LCD control register 0 (LCR0)	See 7.9, 7.11
Port setting	Segment output register (SEGER) Common output V pin control register(COMVER)	See 6.8, 6.9
Display data setting	Data memory for display (VRAM)	See 7.2
Frame cycle setting	LCD control register 0 (LCR0)	See 7.3
Duty selection (activation)		See 7.5
Display selection		See 7.7

Setting Necessary for Cancellation of LCD Display

Setting	Set Register	Set Method
Non-display selection	LCD control register 0 (LCR0)	See 7.7

Setting Necessary for LCD Operation Stop

Setting	Set Register	Set Method
LCD operation stop	LCD control register 0 (LCR0)	See 7.6

Setting Necessary for LCD Display during PSS Timer Mode (Main Oscillation Operation/ Sub Oscillation Operation)

Setting	Set Register	Set Method
Display selection during PSS timer mode(main oscillation operation/ sub oscillation operation)	LCD control register 0 (LCR0)	See 7.8
Transition to PSS timer mode(main oscillation operation/ sub oscillation operation)	See chapter of "LOW-POWER CONSUMPTION".	-

< Static drive >

Setting Necessary for Use of LCD

Setting	Set Register	Set Method
Preliminary setting	LCD control register 1 (LCR1) Common pin switching register (LCDCMR)	See 6.3, 6.4
Port setting	Set the pin to LCDC output. See chapter of "I/O PORTS".	See 0, 6.9
	LCD control register (LCRS)	See 6.5
Display data setting	Data memory for display (LDR0,LDR1)	See 6.6
Frame cycle setting	LCD control register (LCRS)	See 7.3

Setting Necessary for LCD Operation Stop

Setting	Set Register	Set Method
LCD operation stop	LCD control register (LCRS)	See 7.6

Setting Necessary for LCD Display during PSS Timer Mode (Main Oscillation Operation/ Sub Oscillation Operation)

Setting	Set Register	Set Method
Display selection during PSS timer mode(main oscillation operation/ sub oscillation operation)	LCD control register (LCRS)	See 7.8
Transition to PSS timer mode(main oscillation operation/ sub oscillation operation)	See chapter of "LOW-POWER CONSUMPTION".	-

6. Registers

This section explains the registers of the LCD controller.

- Register map

Table 6-1 Registers of the LCD Controller

Register Abbreviation	Register Name	Reference
LCR0	LCD control register 0	6.1
VRAMn(n=0 to15)	Data memory for display	6.2
LCR1	LCDC control register 1	6.3
LCDCMR	Common pin switching register	6.4
LCRS	LCDC Static control register	6.5
LDR	Static LCD display data register	6.6

Table 6-2 Registers of LCD Port-related

Register Abbreviation	Register Name	Reference
LCD_KEYCDR	key code register	6.7
SEGER	Segment output Register	6.8
COMVER	Common output V pin control register	6.9

6.1. LCD Control Register 0: LCR0

The bit configuration of the LCD control register 0 is shown below.

This register selects a frame cycle with its clock to be used, selects display mode, enables the LCD controller to operate in the condition for selecting display/non-display mode, and controls LCD drive power.

■ LCR0

bit	7	6	5	4	3	2	1	0
Field	CSS	LCEN	VSEL	BK	MS1	MS0	FP1	FP0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	1	0	0	0	0

[bit7] CSS: Clock selection Bit

Select a clock to be used for this module.

Bit	Operation
0	Main clock
1	Sub clock

For the single clock products, set "0".

[bit6] LCEN: PSS Timer Mode (Main Oscillation Operation / Sub Oscillation Operation) Operation Enable

Bit	Operation
0	LCD display stop in the PSS timer mode(main oscillation operation/ sub oscillation operation)
1	LCD display in the PSS timer mode(main oscillation operation/ sub oscillation operation)

When LCEN="1", do not stop the clock oscillation which selected by the CSS bit during the transition to PSS mode.

LCD display stop when power shutdown.

[bit5] VSEL: LCD Drive Power Control

Bit	Operation
0	Internal division resistor disconnection
1	Internal division resistor connection

When an external division resistor is to be connected, the LCD drive power control bit (VSEL) must be set to "0".

[bit4] BK: Blanking selection

Bit	Operation
0	LCD display
1	LCD no display

[bit3, bit2] MS1, MS0: Display Mode selection

MS1	MS0	Display Mode
0	0	LCD operation stop
0	1	1/2 duty output mode (Time division number: N=2, COM0, COM1)
1	0	1/3 duty output mode (Time division number: N=3, COM0 to COM2)
1	1	1/4 duty output mode (Time division number: N=4, COM0 to COM3)

When the display mode selection bits (MS [1:0]) are set to "00", the LCD controller's operation is stopped, and the common pin/segment pin output "L" level.

[bit1, bit0] FP1, FP0: Frame Cycle

FP1	FP0	Frame Cycle
0	0	For CSS=0 : $(2^{11} \times N)/F_{CL}$ For CSS=1 : $(2^{3} \times N)/F_{CL}$
0	1	For CSS=0 : $(2^{12} \times N)/F_{CL}$ For CSS=1 : $(2^{4} \times N)/F_{CL}$
1	0	For CSS=0 : $(2^{13} \times N)/F_{CL}$ For CSS=1 : $(2^{5} \times N)/F_{CL}$
1	1	For CSS=0 : $(2^{14} \times N)/F_{CL}$ For CSS=1 : $(2^{6} \times N)/F_{CL}$

F_{CL} : Main clock (LCR0:CSS=0) or sub clock (LCR0:CSS=1)

N : Time division number (depends on the setting of the display mode selection bits MS1, MS0)

Select the setting for FP1 and FP0 that is the optimum condition for the frame frequency of the LCD panel to be used.

6.2. Data Memory for Display: VRAM

The data memory for display is shown below.

Memory (VRAM) area for setting of data for display

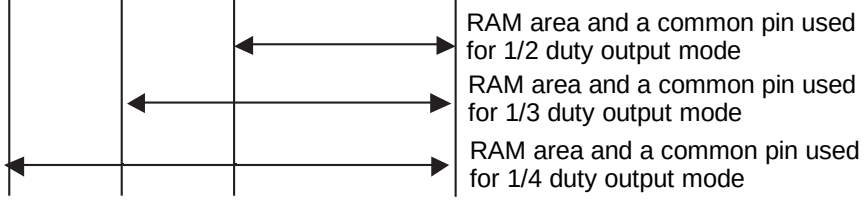
- VRAM0 (SEG0, SEG1)
- VRAM1 (SEG2, SEG3)
- VRAM2 (SEG4, SEG5)
- VRAM3 (SEG6, SEG7)
- VRAM4 (SEG8, SEG9)
- VRAM5 (SEG10, SEG11)
- VRAM6 (SEG12, SEG13)
- VRAM7 (SEG14, SEG15)
- VRAM8 (SEG16, SEG17)
- VRAM9 (SEG18, SEG19)
- VRAM10 (SEG20, SEG21)
- VRAM11 (SEG22, SEG23)
- VRAM12 (SEG24, SEG25)
- VRAM13 (SEG26, SEG27)
- VRAM14 (SEG28, SEG29)
- VRAM15 (SEG30, SEG31)

bit	7	6	5	4	3	2	1	0
Field	D07	D06	D05	D04	D03	D02	D01	D00
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 RAM for display can be read /written with any timing regardless of the LCD controller/driver operation.

Correspondence between VRAM and Common Pins/Segment Pins

VRAM0	bit3	bit2	bit1	bit0	SEG0
	bit7	bit6	bit5	bit4	SEG1
VRAM1	bit3	bit2	bit1	bit0	SEG2
	bit7	bit6	bit5	bit4	SEG3
VRAM2	bit3	bit2	bit1	bit0	SEG4
	bit7	bit6	bit5	bit4	SEG5
VRAM3	bit3	bit2	bit1	bit0	SEG6
	bit7	bit6	bit5	bit4	SEG7
VRAM4	bit3	bit2	bit1	bit0	SEG8
	bit7	bit6	bit5	bit4	SEG9
VRAM5	bit3	bit2	bit1	bit0	SEG10
	bit7	bit6	bit5	bit4	SEG11
VRAM6	bit3	bit2	bit1	bit0	SEG12
	bit7	bit6	bit5	bit4	SEG13
VRAM7	bit3	bit2	bit1	bit0	SEG14
	bit7	bit6	bit5	bit4	SEG15
VRAM8	bit3	bit2	bit1	bit0	SEG16
	bit7	bit6	bit5	bit4	SEG17
VRAM9	bit3	bit2	bit1	bit0	SEG18
	bit7	bit6	bit5	bit4	SEG19
VRAM10	bit3	bit2	bit1	bit0	SEG20
	bit7	bit6	bit5	bit4	SEG21
VRAM11	bit3	bit2	bit1	bit0	SEG22
	bit7	bit6	bit5	bit4	SEG23
VRAM12	bit3	bit2	bit1	bit0	SEG24
	bit7	bit6	bit5	bit4	SEG25
VRAM13	bit3	bit2	bit1	bit0	SEG26
	bit7	bit6	bit5	bit4	SEG27
VRAM14	bit3	bit2	bit1	bit0	SEG28
	bit7	bit6	bit5	bit4	SEG29
VRAM15	bit3	bit2	bit1	bit0	SEG30
	bit7	bit6	bit5	bit4	SEG31
	COM3	COM2	COM1	COM0	



6.3. LCDC Control Register 1: LCR1

The bit configuration of the LCDC control register 1 is shown below.

■ LCR1

bit	7	6	5	4	3	2	1	0
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1
Protection attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

[bit7 to bit0] Reserved

When the LCD is used, set "11111111" always.

6.4. Common Pin Switching Register: LCDCMR

The bit configuration of the common pin switching register is shown below.

■ LCDCMR

bit	7	6	5	4	3	2	1	0
Field	DTCH	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R/W	R/W0	R/W0	R/W0	R/W1	R/W1	R/W1	R/W1
Protection attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

[bit7] DTCH: Bias selection

Bit	Operation
0	1/3 bias
1	1/2 bias

[bit6 to bit4] Reserved

When the LCD is used, set "000" always.

[bit3 to bit0] Reserved

When the LCD is used, set "1111" always.

6.5. LCDC Static Control Register: LCRS

The bit configuration of the LCDC static control register is shown below.

■ LCRS

bit	7	6	5	4	3	2	1	0
Field	SCSS	LCSEN	LCS3	LCS2	LCS1	LCS0	FPS1	FPS0
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] SCSS: Frame Cycle Generation Clock for Static Drive selection Bit

Bit	Operation
0	Main clock
1	Sub clock

For the single clock products, set "0".

[bit6] LCSEN: PSS Timer Mode (Main Oscillation Operation / Sub Oscillation Operation) Operation Enable

Bit	Operation
0	LCD display stop in the PSS timer mode(main oscillation operation/ sub oscillation operation)
1	LCD display in the PSS timer mode(main oscillation operation/ sub oscillation operation)

When LCSEN="1", please do not stop the clock oscillation which selected by the SCSS bit during the transition to PSS mode.

LCD display stop when power shutdown.

[bit5:2] LCS3 to LCS0: Static Drive selection

LCS3	LCS2	LCS1	LCS0	Static Drive Selection Port
				Common / Segment Output
0	0	0	0	OFF
0	0	0	1	ST0, ST1
0	0	1	0	ST0 to ST2
0	0	1	1	ST0 to ST3
0	1	0	0	ST0 to ST4
0	1	0	1	ST0 to ST5
0	1	1	0	ST0 to ST6
0	1	1	1	ST0 to ST7
1	X	X	X	ST0 to ST8

Note:

- *For the use method for the static drive selection, refer to "4.5 Static Drive Output Waveform".*

[bit1:0] FPS1, FPS0: Frame Cycle

FPS1	FPS0	Frame Cycle
0	0	For SCSS=0 : $(2^{11} \times 4)/F_{CL}$ For SCSS=1 : $(2^{3} \times 4)/F_{CL}$
0	1	For SCSS=0 : $(2^{12} \times 4)/F_{CL}$ For SCSS=1 : $(2^{4} \times 4)/F_{CL}$
1	0	For SCSS=0 : $(2^{13} \times 4)/F_{CL}$ For SCSS=1 : $(2^{5} \times 4)/F_{CL}$
1	1	For SCSS=0 : $(2^{14} \times 4)/F_{CL}$ For SCSS=1 : $(2^{6} \times 4)/F_{CL}$

F_{CL} : Main clock (LCRS:SCSS=0) or sub clock (LCRS:SCSS=1)

Select the setting for FPS1 and FPS0 that is the optimum condition for the frame frequency of the LCD panel to be used.

The frame cycle during static drive is equal to the cycle of the 1/4 duty output mode.

The static drive is enabled when SEGER setting and any setting other than LCS [3:0] = "0000" are executed.

The duty drive is enabled when SEGER and LCS [3:0] = "0000" are set.

6.6. Static LCD Display Data Register: LDR

The bit configuration of the static LCD display data register is shown below.

■ LDR0

bit	15	14	13	12	11	10	9	8
Field	-		-	-	-	-	-	ST8
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

[bit7:1] Undefined Bits

The read value is always "0". Writing has no effect on operation.

[bit0] ST8: Static Output Data

This bit is static output data for ST8.

■ LDR1 : Address 05BD$_H$ (Access : Byte, Half-word, Word)

bit	7	6	5	4	3	2	1	0
Field	ST7	ST6	ST5	ST4	ST3	ST2	ST1	ST0
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

[bit7:0] ST7 to ST0: Static Output Data

These bits are static output data for ST0 to ST7.

Static Output Data Correspondence Table

LDR	Operation
ST8	ST8 static output data
ST7	ST7 static output data
ST6	ST6 static output data
ST5	ST5 static output data
ST4	ST4 static output data
ST3	ST3 static output data
ST2	ST2 static output data
ST1	ST1 static output data
ST0	ST0 static output data

This register is for setting of data output to the LCD static drive port (set by the LCRS register). The LCD static drive is executed by inverting output of the specified data periodically.

Common output pins are not specified especially. Assign common output to one of segments. For example, if ST0 to ST8 are assigned static drive pins, ST8 is a common pin and set to LDR0:ST8 = "0". Furthermore, by setting LDR1: ST [7:0] = "11111111", the LCD selected with ST0 to ST7 is displayed.

6.7. Key Code Register: LCD_KEYCDR

This section shows the bit configuration of the key code register.

This register makes the write setting of a register that has the function for protection against erroneous writing (SEGER, COMVER). If writing to this register is not done using the prescribed method, writing to the relevant register is invalid. This register is valid only for word access.

■ KEYCDR

bit	31	30	29	28	27	26	25	24
Field	KEY1	KEY0	SIZE1	SIZE0	Reserved			
Attribute	R0,W	R0,W	R0,W	R0,W	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							
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				ADR11	ADR10	ADR09	ADR08
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	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	ADR07	ADR06	ADR05	ADR04	ADR03	ADR02	ADR01	ADR00
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

[bit31:30] KEY1 to KEY0: Key Code Setting Bits

You need to 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, you need to set the key codes again from the beginning.

Bit31:30		Description
0	0	1st key code
0	1	2nd key code
1	0	3rd key code
1	1	4th key code

[bit29:28] SIZE [1:0]: Access Size Setting Bits

These bits set the access size for writing to a key code target register.

Write the same data to these bits when writing the key codes of "00", "01", "10", and "11" to KEY [1:0] in this order.

Bit29:28		Description
0	0	Set byte access.
0	1	Set half-word access.
1	0	Set word access.
1	1	Reserved

[bit27:12] Reserved

Always write "0" to these bits.

[bit11:0] ADR11 to ADR0: Address Specification Bits

These bits set the lower 12 bits of the address of a key code target register. Write the same data to these bits when writing the key codes of "00", "01", "10", and "11" to KEY [1:0] in this order.

Bit11:0	Description
Set the lower 12 bits of the address of a key code target register.	

Notes:

The following are the notes on key code setting.

- *You need to write "00", "01", "10", and "11" continuously to KEY [1:0] in this order in 4 write operations.*
  The key code setting becomes invalid immediately upon any writing to these bits in a different order. In such cases, you need to set the key codes again from the beginning.

- *You need to write the same pair of values to SIZE [1:0] and ADR [11:0] 4 times when writing data to KEY [1:0] 4 times.*
  The key code setting becomes invalid upon any writing of a different pair of values. In such cases, you need to set the key codes again from the beginning.
- *When the KEYCDR register is read during writing to KEY [1:0], the key code setting becomes invalid. In such cases, you need to set the key codes again from the beginning.*
- *When access to the address set in ADR [11:0] occurs during writing to KEY [1:0], the key code setting becomes invalid. In such cases, you need to set the key codes again from the beginning.*
  When access to any register other than the above occurs, the key code setting does not become invalid.
  Writing the remaining data to KEY [1:0] subsequently makes the key code setting valid.
- *Once the address set in KEYCDR: ADR [11:0] is accessed after KEY [1:0] is set normally, it is required to set the key codes again to access the address again.*

6.8. Segment Output Register: SEGER

This section shows the bit configuration of the Segment output Register.

Segment output register (SEGER) is a register which control the segment output enable of LCD controller duty driving driver.

■ SEGER

bit	31	30	29	28	27	26	25	24
Field	SEGE7	SEGE6	SEGE5	SEGE4	SEGE3	SEGE2	SEGE1	SEGE0
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	SEGE15	SEGE14	SEGE13	SEGE12	SEGE11	SEGE10	SEGE9	SEGE8
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	SEGE23	SEGE22	SEGE21	SEGE20	SEGE19	SEGE18	SEGE17	SEGE16
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	SEGE31	SEGE30	SEGE29	SEGE28	SEGE27	SEGE26	SEGE25	SEGE24
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

[bit31:0] SEGEn (n=0 to 31)

SEGEn(n=0 to 31)	Operation
0	Disable LCDC segment output
1	Enable LCDC segment output

6.9. Common Output V Pin Control Register: COMVER

This section shows the bit configuration of the common output V pin control register.

Common output V pin control register (COMVER) is a register which control the common output enable of LCD controller static driving driver, input enable and selection of LCD controller reference voltage.

■ COMVER

bit	31	30	29	28	27	26	25	24
Field	VE3	VE2	VE1	VE0	COME3	COME2	COME1	COME0
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	Reserved							
Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX
Protection attribute	-	-	-	-	-	-	-	-
Initial value	1	1	1	1	1	1	1	1

bit	15	14	13	12	11	10	9	8
Field	Reserved							
Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX
Protection attribute	-	-	-	-	-	-	-	-
Initial value	1	1	1	1	1	1	1	1

bit	7	6	5	4	3	2	1	0
Field	Reserved							
Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX
Protection attribute	-	-	-	-	-	-	-	-
Initial value	1	1	1	1	1	1	1	1

[bit31] VE3

VE3	Operation
1	Use V3 pin as the LCDC V3 reference voltage input
0	VCC is used for the LCDC V3 reference voltage

Please to switch on the main oscillation state when switching external V3 and VCC as LCDC reference supply V3 input.

[bit30:28] VEn (n=0 to 2)

VEn(n=0 to 2)	Operation
0	Disable LCDC reference voltage input
1	Enable LCDC reference voltage input

[bit27:24] COMEn (n=0 to 3)

COMEn(n=0 to 3)	Operation
0	Disable LCDC common output
1	Enable LCDC common output

[bit23:0] Reserved

The read value is always "1". Writing has no effect on operation.

7. Q&A

This section shows Q&A of the LCD controller.

7.1. How Can I Set Pins to COM Output Pins or SEG Output Pins?

7.2. How to Set VRAM?

7.3 How Can I Setting the Frame Cycle?7.4. How Can I Set the Bias?

7.5. How Can I Set the Duty?

7.6. How Can I Control the LCD Operation Start/Stop?

7.7. How Can I Execute/Cancel the Display?

7.8. How Can I Display during the PSS Timer Mode (Main Oscillation Operation/ Sub Oscillation Operation)?

7.9. How Can I Select Either Internal or External for the Division Resistor?

7.10. How Can I Select Pin of V3 Voltage?

7.11. How Can I Select Either Internal or External for the Division Resistor?

7.12. How Can I Adjust the Brightness When the Internal Division Resistor is Used?

7.13. How Can I Block the Current with the External Division Resistor When the LCD Stops?

7.14. How Can I Display/Non-Display the LCD with Static Drive (ST0 to ST8)?

7.1. How Can I Set Pins to COM Output Pins or SEG Output Pins?

This section shows how to set pins to COM output pins or SEG output pins.

COM output and SEG output setting.

With software setting, ports can be switched to COM, SEG output.

Set pins for COM output and SEG output to peripheral output. See following table for pin setting.

<Duty drive>

Pin	V0-V3 and COM and SEG Setting Method	Port Setting Method
V0	Set pins to LCDC V0/V1/V2/V3 (LCDC reference power input). See 6.9.	See "Port Configuration 5.6 Analog Function Input or Output"
V1		
V2		
V3		
COM0		
COM1		
COM2		
COM3		
SEG0	See 6.8.	
SEG1		
SEG2		
SEG3		
SEG4		
SEG5		
SEG6		
SEG7		
SEG8		
SEG9		
SEG10		
SEG11		
SEG12		
SEG13		
SEG14		
SEG15		
SEG16		
SEG17		
SEG18		

Pin	V0-V3 and COM and SEG Setting Method	Port Setting Method
SEG19	See 6.8.	See "Port Configuration 5.6 Analog Function Input or Output"
SEG20		
SEG21		
SEG22		
SEG23/ST0		
SEG24/ST1		
SEG25/ST2		
SEG26/ST3		
SEG27/ST4		
SEG28/ST5		
SEG29/ST6		
SEG30/ST7		
SEG31/ST8		

<Static drive >

COM output and SEG output setting.

With software setting, ports can be switched to COM, SEG output.

Set pins for static drive to peripheral output. See 6.8. for pin setting.

Pin	V0-V3 and SEG Setting Method	Port Setting Method
V0	Set pins to LCDC V0/V3 (LCDC reference power input). See 6.9.	See "Port Configuration 5.6 Analog Function Input or Output"
V3		
SEG23/ST0	See 6.8.	
SEG24/ST1		
SEG25/ST2		
SEG26/ST3		
SEG27/ST4		
SEG28/ST5		
SEG29/ST6		
SEG30/ST7		
SEG31/ST8		

7.2. How to Set VRAM?

This section shows how to set VRAM.

<Duty drive>

The matrix of pins and bit locations of VRAM (n) is shown below. (n = 0 to 15)

1/2 Duty

Pin	COM1	COM0
SEG 2n	bit1	bit0
SEG 2n+1	bit5	bit4

1/3 Duty

Pin	COM2	COM1	COM0
SEG 2n	bit2	bit1	bit0
SEG 2n+1	bit6	bit5	bit4

1/4 Duty

Pin	COM3	COM2	COM1	COM0
SEG 2n	bit3	bit2	bit1	bit0
SEG 2n+1	bit7	bit6	bit5	bit4

(Non-selection waveforms are output from irrelevant pins.)

Example: 1/4 duty

When the bit6 of VRAMn is set to "1", the selection waveform is output from the SEGn+1 of COM2.

For bits with "0"setting, non-selection waveforms are output to pins.

7.3. How Can I Setting the Frame Cycle?

This section shows how to set pins to COM output pins or SEG output pins.

<Duty drive>

The frame cycle can be set with the frame cycle bits (LCR0: FP [1:0]).The following settings are available.

Frame Cycle (When a Main Clock Is Selected)	Selection Value
	Frame Cycle Bits (FP[1:0])
$(2^{11} \times N)$/ Main clock frequency	Set "00".
$(2^{12} \times N)$/ Main clock frequency	Set "01".
$(2^{13} \times N)$/ Main clock frequency	Set "10".
$(2^{14} \times N)$/ Main clock frequency	Set "11".

N (Time division number) = Value of MS [1:0] + "1"

Frame Cycle (When a Sub Clock Is Selected)	Selection Value
	Frame Cycle Bits (FP[1:0])
$(2^{3} \times N)$/ Sub clock frequency	Set "00".
$(2^{4} \times N)$/ Sub clock frequency	Set "01".
$(2^{5} \times N)$/ Sub clock frequency	Set "10".
$(2^{6} \times N)$/ Sub clock frequency	Set "11".

N (Time division number) = Value of MS [1:0] + "1"

< Static drive >

The frame cycle can be set with the frame cycle bits (LCRS: FPS [1:0]).The following settings are available.

Frame Cycle	Selection Value
	Frame Cycle Bits (FPS[1:0])
$(2^{11} \times 4)$/Main clock (F_{CL})	Set "00".
$(2^{12} \times 4)$/ Main clock (F_{CL})	Set "01".
$(2^{13} \times 4)$/ Main clock (F_{CL})	Set "10".
$(2^{14} \times 4)$/ Main clock (F_{CL})	Set "11".

7.4. How Can I Set the Bias?

This section shows how to set the bias.

<Duty drive>

Set the bias selection bit (LCDCMR: DTCH).

Bias	Bias Selection Bit (DTCH)
To set 1/3 bias	Set "0".
To set 1/2 bias	Set "1".

7.5. How Can I Set the Duty?

This section shows how to set the duty.

<Duty drive>

Set the display mode selection bits (LCR0: MS [1:0]).

Control Detail	Display Mode Selection Bits (MS[1:0])	N (Time Division Number)
LCD operation stop (Pin output "L")	Set "00".	-
To set 1/2 duty output mode	Set "01".	2
To set 1/3 duty output mode	Set "10".	3
To set 1/4 duty output mode	Set "11".	4

The display mode selection bits are also the control bit for operation start/stop.

7.6. How Can I Control the LCD Operation Start/Stop?

This section shows how to control the LCD operation start/stop.

<Duty drive>

Operation start/stop can be controlled with the display mode selection bits (LCR0: MS [1:0]).

See "7.5 How can I set the duty?".

<Static drive>

Operation start/stop can be controlled with the display mode selection bits (LCRS: LCS [3:0]).

7.7. How Can I Execute/Cancel the Display?

This section shows how to execute/cancel the LCD display.

<Duty drive>

The two methods below are available.

■ Setting of the blanking selection bit (LCR0:BK)

Control Detail	Blanking Selection Bit (BK)
To execute LCD display	Set "0".
To cancel LCD display (A non-selection waveform is output to a segment pin.)	Set "1".

■ Cancellation of display with operation stop by the display mode selection bits (LCR0: MS [1:0]).

Control Detail	Display Mode Selection Bits (MS[1:0])
LCD operation stop ("L" output from the common pin and segment pin.)	Set "00".

7.8. How Can I Display during the PSS Timer Mode (Main Oscillation Operation/ Sub Oscillation Operation)?

This section shows how to display during the PSS timer mode (main oscillation operation/ sub oscillation operation).

<Duty drive>

Set the PSS timer mode (main oscillation operation/ sub oscillation operation) operation enable bit (LCR0: LCEN).

Control Detail	PSS Timer Mode(Main Oscillation Operation/ Sub Oscillation Operation) Operation Enable Bit (LCEN)
For no LCD display in the PSS timer mode(main oscillation operation/ sub oscillation operation)	Set "0".
For LCD display in the PSS timer mode(main oscillation operation/ sub oscillation operation)	Set "1".

When LCEN="1", please do not stop the clock oscillation which selected by the CSS bit during the transition to PSS mode. LCD display stop when power shutdown.

<Static drive>

Set the PSS timer mode (main oscillation operation/ sub oscillation operation) operation enable bit (LCRS: LCSEN).

Control Detail	PSS Timer Mode(Main Oscillation Operation/ Sub Oscillation Operation) Operation Enable Bit (LCSEN)
For no LCD display in the PSS timer mode(main oscillation operation/ sub oscillation operation)	Set "0".
For LCD display in the PSS timer mode(main oscillation operation/ sub oscillation operation)	Set "1".

When LCSEN="1", please do not stop the clock oscillation which selected by the SCSS bit during the transition to PSS mode. LCD display stop when power shutdown.

7.9. How Can I Select Either Internal or External for the Division Resistor?

This section shows how to select either internal or external for the division resistor.

<Duty drive>

Set the LCD drive power control bit (LCR0: VSEL).

Control Detail	LCD Drive Power Control Bit (VSEL)
For use of the external division resistor (The internal division resistor is disconnected.)	Set "0".
For use of the internal division resistor (The internal division resistor is connected.)	Set "1".

7.10. How Can I Select Pin of V_3 Voltage?

This section shows how to select the pin of V_3 voltage.

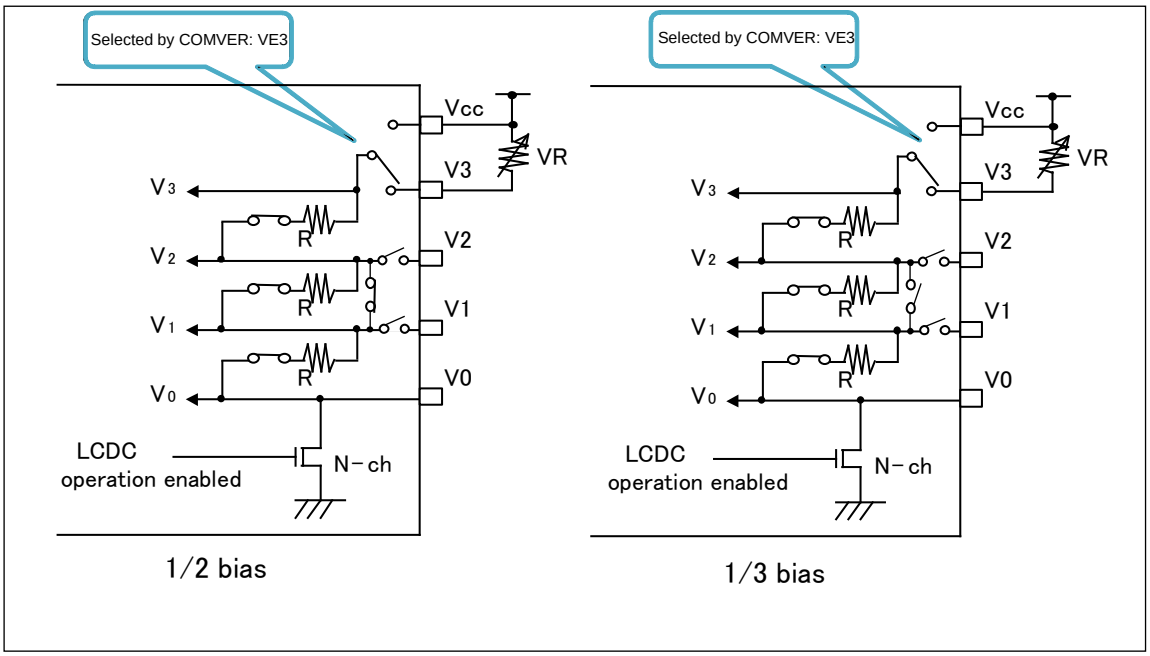
Set the COMVER: VE3 bit. See chapter of 6.9 for details.

7.11. How Can I Select Either Internal or External for the Division Resistor?

This section shows how to select either internal or external for the division resistor.

■ When the internal division resistor is selected

Even when the internal division resistor is used if the V3 pin is used as V_3 voltage, the external resistor must be connected between Vcc and V3.



■ When the external division resistor is selected

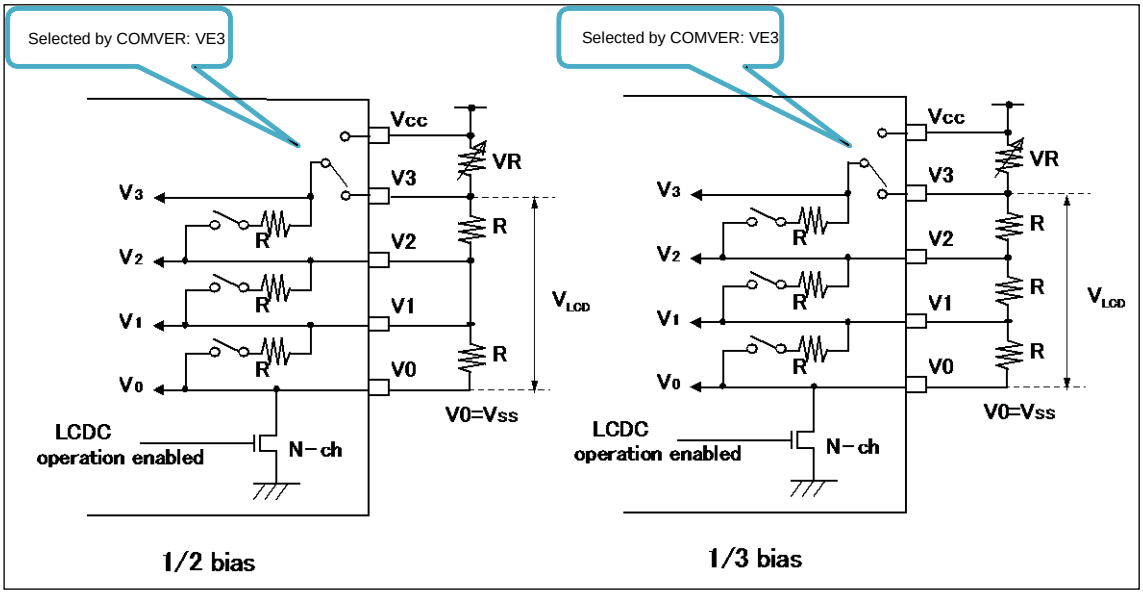
Voltage for the LCD drive is set with the external division resistor connected to the power pins for LCD drive (V0 to V3).

Setting of LCD drive voltage

	V3	V2	V1	V0
1/2 bias	VLCD	1/2VLCD	1/2VLCD	VSS
1/3 bias	VLCD	2/3VLCD	1/3VLCD	VSS

V_0 to V_3 : Voltage of V0 to V3 pins

V_{LCD} : Operation voltage of the LCD



To avoid the effect of the internal division resistor, the LCD drive power control bit (LCR0: VSEL) must be set to "0" to disconnect the internal division resistor.

7.12. How Can I Adjust the Brightness When the Internal Division Resistor is Used?

This section shows how to adjust the brightness when the internal division resistor is used.

If desired brightness cannot be obtained with the use of the internal division resistor, adjust the voltage V3 by putting a variable resister (VR) between the outer terminals Vcc and V3.

7.13. How Can I Block the Current with the External Division Resistor When the LCD Stops?

This section shows how to block the current with the external division resistor when the LCD stops.

The V0 pin is connected to the Vss (GND) via a transistor internally. Therefore, when the external division resistor is used, by connecting the Vss side of the external division resistor to the V0 pin, the current which flows on the LCD controller which is stopped can be blocked. Block the current with the display mode selection bits (MS [1:0] = "00").

7.14. How Can I Display/Non-Display the LCD with Static Drive (ST0 to ST8)?

This section shows how to display/non-display the LCD with static drive (ST0 to ST8).

When the LCD with static drive (ST0 to ST8) is switched to non-display after reset, set LDR0:ST8 = "0" and LDR1: ST [7:0] = "00000000", and set the static drive selection port (LCS [3:0]).

The same potential pulses are output from the static drive pins (ST0 to ST8).

When the LCD with static drive (ST0 to ST8) is switched from non-display to display, set LDR0:ST8 and LDR1: ST [7:0] without change of (LCS [3:0]).

When the LCD with static drive (ST0 to ST8) is switched from display to non-display, set LDR0:ST8 = "0" and LDR1: ST [7:0] = "00000000" without change of the static drive selection port (LCS [3:0]).

The same potential pulses are output from the static drive pins (ST0 to ST8).

8. Sample Program

This section shows sample program.

Setting procedure 1

With the 1/2 duty drive method, make the LCD display a four-digit number, "0 1 2 3".

Initial setting (LCDC)

<Initial setting>

1. - Port

	Register name
COM, SEG output setting for the port	See 6.8. and 6.9.

2. - Setting of VRAM

	Register name
Setting of VRAM	VRAM00-VRAM07

3. - Setting of control register

	Register name
Fixed value	LCR1
Bias setting	LCDCMR
Setting of control register	LCR0
	. LCEN
	. VSEL
	. BK
	. MS[1:0]
	. FP[1:0]

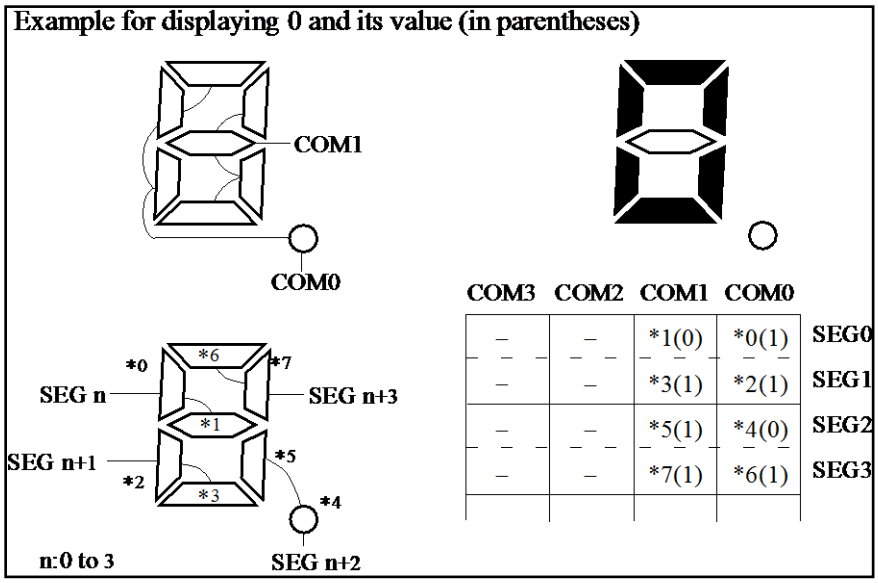
<Others>

(Note)

Clock-related setting and setting of _set_il (numerical value) in advance are required. See chapter of "CLOCK SYSTEM" and chapter of "INTERRUPT CONTROLLER" for details.

Program 1

```
void LCD_sample_1(void)
{
        LCDC_initial();
}
void LCDC_initial(void)
{
        PORT_SETTING_LCDC_OUT();    /* Set the LCD controller pin to */
                                    /* peripheral output.   */

        IO_VRAM00 = 0x31;
        IO_VRAM01 = 0x32;
        IO_VRAM02 = 0x00;
        IO_VRAM03 = 0x22;
        IO_VRAM04 = 0x32;
        IO_VRAM05 = 0x30;
        IO_VRAM06 = 0x22;
        IO_VRAM07 = 0x32;

        IO_LCR1.byte = 0xFF;        /* Set to FF. */
        IO_LCDCMR. byte = 0x0F;     /* 1/2 bias */
        IO_LCR0.byte = 0x04;        /* Setting value =0000_0100 */
                                    /* bit7 = 0 Clock selection    Main clock */
                                    /* bit6 = 0 LCEN    Display stopped with PSS mode   */
                                    /* bit5 = 0 VSEL Internal division resistor disconnected
                                    */
                                    /* bit4 = 0 BK Blanking selection bit */
                                    /* bit3-2 = 01 MS[1:0]          1/2 duty mode */
                                    /* bit1-0 = 00 FP[1:0] */
}
```


Example for displaying 0 and its value (in parentheses)
COM1
COM0
*0
*6
*7
SEG n
SEG n+3
*1
SEG n+1
*5
*2
*3
*4
n:0 to 3
SEG n+2
COM3 COM2 COM1 COM0
– – *1(0) *0(1) SEG0
– – *3(1) *2(1) SEG1
– – *5(1) *4(0) SEG2
– – *7(1) *6(1) SEG3

Setting procedure 2

With the1/3 duty drive method, make the LCD display a four-digit number , "0 1 2 3 ".

Initial setting (LCDC)

<Initial setting>

1. - Port

	Register name
COM, SEG output setting for the port	See 6.8. and 6.9.

2. - Setting of VRAM

	Register name
Setting of VRAM	VRAM00-VRAM05

3. - Setting of control register

Fixed value	LCR1
Bias setting	LCDCMR
Setting of control register	LCR0
	. LCEN
	. VSEL
	. BK
	. MS[1:0]
	. FP[1:0]

<Others>

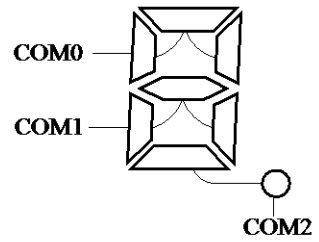
(Note)

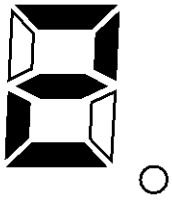
Clock-related setting and setting of _set_il (numerical value) in advance are required. See chapter of "CLOCK SYSTEM" and chapter of "INTERRUPT CONTROLLER" for details.

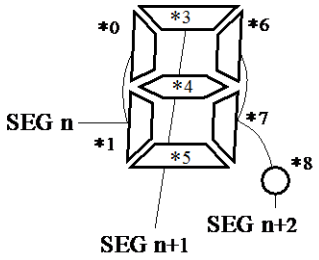
Program 2

```
void LCD_sample_2(void)
{
        LCDC_initial();
}
void LCDC_initial(void)
{
        PORT_SETTING_LCDC_OUT();        /* Set the LCD controller pin to */
                                        /* peripheral output.    */

        IO_VRAM00 = 0x53;
        IO_VRAM01 = 0x03;
        IO_VRAM02 = 0x30;
        IO_VRAM03 = 0x72;
        IO_VRAM04 = 0x01;
        IO_VRAM05 = 0x37;

        IO_LCR1.byte = 0xFF;            /* Set to FF. */
        IO_LCDCMR. byte = 0x00;         /* 1/3 bias */
        IO_LCR0.byte = 0x08;            /* Setting value =0000_1000 */
                                        /* bit7 = 0 Clock selection Main clock */
                                        /* bit6 = 0 LCEN Display stopped with PSS mode */
                                        /* bit5 = 0 VSEL Internal division resistor disconnected */
                                        /* bit4 = 0 BK Blanking selection bit */
                                        /* bit3-2 = 10 MS[1:0]  1/3 duty   mode */
                                        /* bit1-0 = 00 FP[1:0] */
}
```

Example for displaying 2 and its value (in parentheses)







COM3	COM2	COM1	COM0		
–	(-)	*1(1)	*0(0)	SEG0	First digit
–	*5(1)	*4(1)	*3(1)	SEG1	
–	*8(0)	*7(0)	*6(1)	SEG2	
–	–	1	0	SEG3	Second digit
–	1	1	1	SEG4	
–	0	0	1	SEG5	

Setting procedure 3

With the 1/4 duty drive method, make the LCD display a four-digit number, "0 1 2 3 ".

Initial setting (LCD)

<Initial setting>

1. - PORT

	Register name
COM, SEG output setting for the port	See 6.8. and 6.9.

2. - Setting of VRAM

	Register name
Setting of VRAM	VRAM00-VRAM03

3. - Setting of control register

	registername . bit name
Fixed value	LCR1
Bias setting	LCDCMR
Setting of control register	LCR0
	. LCEN
	. VSEL
	. BK
	. MS[1:0]
	. FP[1:0]

<Others>

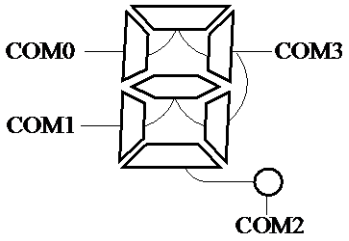
(Note)

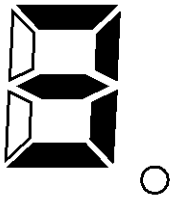
Clock-related setting and setting of _set_il (numerical value) in advance are required. See chapter of "CLOCK SYSTEM" and chapter of "INTERRUPT CONTROLLER" for details.

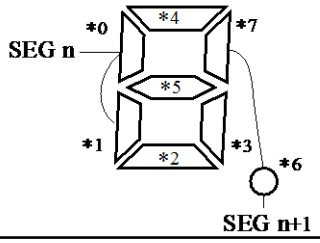
Program 3

```
void LCD_sample_3(void)
{
        LCD_initial();
}

void lcdc_initial(void)
{
        PORT_SETTING_LCDC_OUT();        /* Set the LCD controller pin to */
                                        /* peripheral output. */

        IO_VRAM00 = 0x9F;
        IO_VRAM01 = 0x88;
        IO_VRAM02 = 0xB6;
        IO_VRAM03 = 0xBC;

        IO_LCR1.byte = 0xFF;            /* Set to FF. */
        IO_LCDCMR. byte = 0x00;         /* 1/3 bias */
        IO_LCR0.byte = 0x0C;            /* Setting value =0000_1100 */
                                        /* bit7 = 0 Clock selection Main clock */
                                        /* bit6 = 0 LCEN   Display stopped with PSS mode */
                                        /* bit5 = 0 VSEL Internal division resistor disconnected */
                                        /* bit4 = 0 BK Blanking selection bit */
                                        /* bit3-2 = 11 MS[1:0]  1/4 duty mode */
                                        /* bit1-0 = 00 FP[1:0] */
}
```

Example for displaying 3 and its value (in parentheses)







COM3	COM2	COM1	COM0	
*3(1)	*2(1)	*1(0)	*0(1)	SEG0
*7(0)	*6(0)	*5(1)	*4(1)	SEG1

Setting procedure 4

With the static drive method, make the LCD display an one-digit number.

Initial setting (LCD)

<Initial setting>

1. - PORT — Register name

COM, SEG output setting for the port	See 6.8. and 6.9.

2. - Data setting — Register name

Data setting	LDR0 LDR1

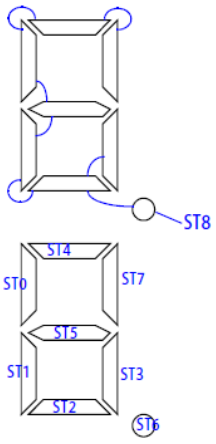
3. - Setting of control register — Register name . bit name

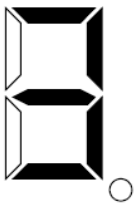
Fixed value	LCR1
Bias setting	LCDCMR
Setting of control register	LCRS
	. LCSEN
	. LCS[3:0]
	. FPS[1:0]

Program 4

```
void LCD_sample_4(void)
{
        LCD_initial();
}

void lcdc_initial(void)
{
        PORT_SETTING_LCDC_OUT();        /* Set the LCD controller pin to */
                                        /* peripheral output.   */

        IO_LDR0.byte = 0x00;
        IO_LDR1.byte = 0xBC;

        IO_LCR1.byte = 0xFF;            /* Set to FF. */
        IO_LCDCMR. byte = 0x00;         /* 1/3 bias */
        IO_LCRS.byte = 0x20;            /* Setting value =0001_0000 */
                                        /* bit7 = 0 Clock selection Main clock */
                                        /* bit6 = 0 LCSEN   Non-display with PSS mode   */
                                        /* bit5-2 = 1000   LCS[3:0]   ST0 to ST8 */
                                        /* bit1-0 = 00 FPS[1:0]   Frame cycle setting */
}
```

Example) Values to display 3





LDR0/ LDR1 register

ST 8	ST 7	ST 6	ST 5	
0	1	0	1	
ST 4	ST 3	ST 2	ST 1	ST 0
1	1	1	0	0

9. Notes

This section shows notes.

- Since resistance value of the external division resisters depends on the LCD to be used, connect the resistors with a suitable value.
- Non-selection level waveforms are output from the COM2 and COM3 pins in 1/2 duty display mode and from the COM3 pin in 1/3 duty display mode.
- If the settings of the LCD drive power control (VSEL), duty selection (MS [1:0]), frame cycle selection (FP [1:0]), etc. are not proper, the LCD does not display correctly.
- When neither the LCD nor general-purpose ports are used, connect a pull-up or pull-down resistor to V3 to V0 pins.
- The static drive is enabled when any setting other than LCS [3:0] = "0000" is set. The duty drive is enabled when LCS [3:0] = "0000" is set.
- Setting of display/non-display of the LCD with static drive (ST0 to ST8)

  When the LCD with static drive (ST0 to ST8) is switched to non-display after reset, set LDR0:ST8 = "0" and LDR1:ST [7:0] = "00000000", and set the static drive selection port (LCS [3:0]). The same potential pulses are output from the static drive pins (ST0 to ST8).

When the LCD with static drive (ST0 to ST8) is switched from non-display to display, set LDR0:ST8 and LDR1:ST [7:0] without change of the static drive selection port (LCS [3:0]).
When the LCD with static drive (ST0 to ST8) is switched from display to non-display, set LDR0:ST8 = "0" and LDR1:ST [7:0] = "00000000" without change of the static drive selection port (LCS [3:0]).

The same potential pulses are output from the static drive pins (ST0 to ST8).

- When the LCD is used (static drive, duty drive), set the following.
  LCR1 [7:0] = "11111111"
  LCDCMR [3:0] = "1111"

In case of power shutdown, it works as following.

- When port setting is LCD output, output the "Low".
- LCD control is power shutdown, registers (LCR0, VRAMn (n=0 to 15), LCR1, LCDCMR, LCRS, LDR) of LCD control is initialized when returning from power shutdown.
- Registers (LCD_KEYCDR, SEGER, COMVER) of LCD port-related is not initialized.

CHAPTER 58: Sound Generator

This chapter explains the functions and operations of the Sound Generator(SG).

1. Overview
2. Configuration
3. Operations
4. Registers

SG-T01P00R02L01-E1-GBM01

1. Overview

The Sound Generator generates and outputs a tone pulse signal, a PWM pulse signal, or mixed one of both signals according to the setting by MCU. It is enabled to set the frequency of the tone pulse signal, sound volume (amplitude of PWM pulse), and length of sound. The Sound Generator consists of the following registers and counters.

- DMA Transfer Update Enable Register (SGDER)
- Sound Control Register (SGCR)
- Amplitude Data Register (SGAR)
- Frequency Data Register (SGFR)
- Time Cycle Register (SGTCR)
- Tone Output Number Register (SGNR)
- Increase and Decrease Data Register (SGIDR)
- PWM Cycle Data Register (SGPCR)
- DMA Transfer Intermediate Register (SGDMAR)
- Interrupt Clear Register (SGCCR)
- PWM pulse generation counter
- Frequency counter
- Decrement counter
- Tone pulse counter

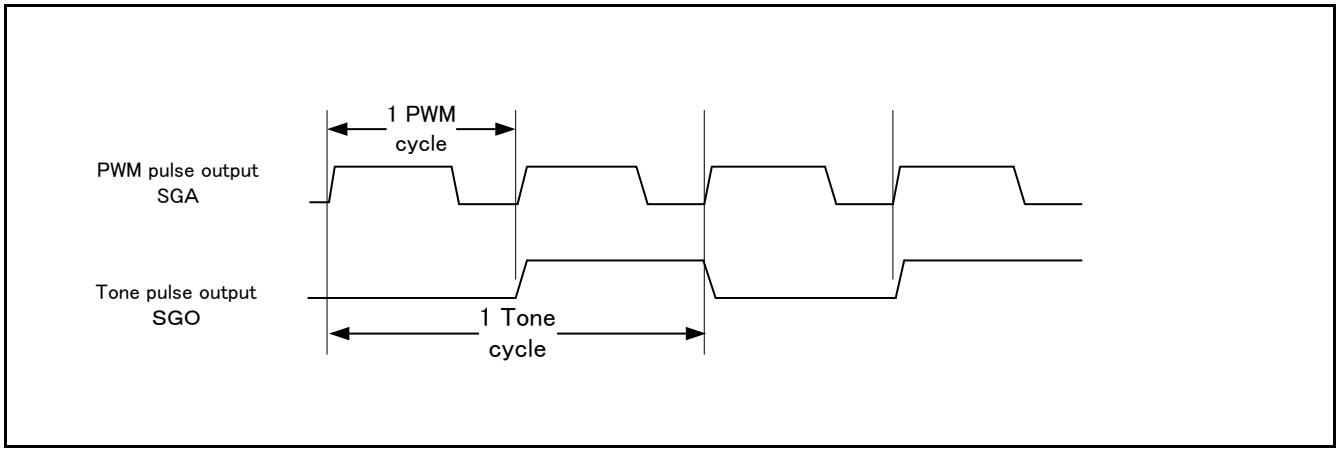
Functions of Sound Generator

Table 1-1 Functions of Sound Generator

No	Item	Function
1	Operation clock	Bus clock (BCLK)
2	Clock input	For clock input of the Sound Generator, divide the bus clock (BCLK) and use it. - 1 x BCLK - 1/2 x BCLK - 1/4 x BCLK - 1/8 x BCLK
3	Waveform	Rectangular wave for sound (tone pulse output, sound output from SGO pin)
4	Volume of sound	It is enabled to set their arbitrary level (PWM pulse output). (Amplitude output from SGA pin)
5	Frequency	It is enabled to set their arbitrary level for sound signal frequency. (Frequency setting and PWM cycle setting)
6	Length of sound	It is enabled to set their arbitrary level.
7	Interrupt	- It is enabled to generate interrupt requests at the end of outputting sound at programmed length (An overflow of Tone pulse counter). - It is enabled to generate interrupts by writing "1" to the Start bit (SGCR.ST) when in DMA mode (SGCR.DMA="1").

Output of Sound Generator

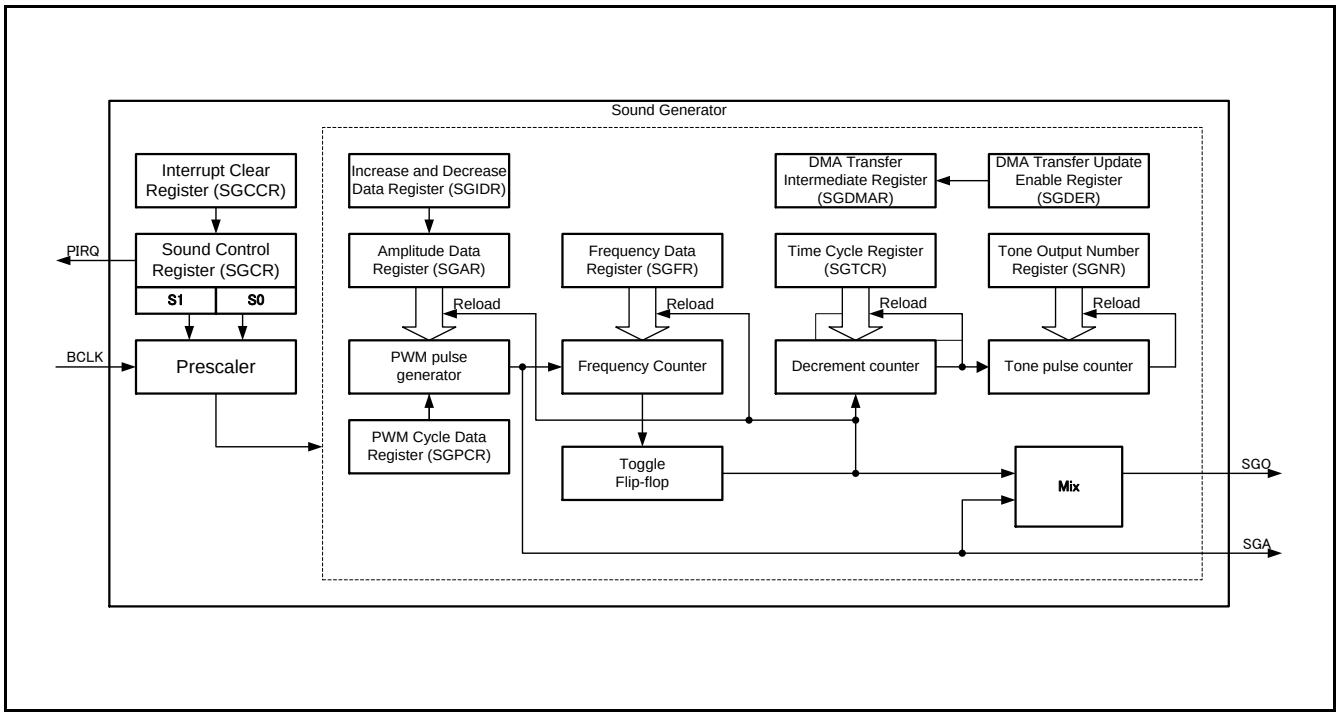
Figure 1-1 Example of output



2. Configuration

This chapter shows the block diagram of Sound Generator.

Figure 2-1 Block Diagram

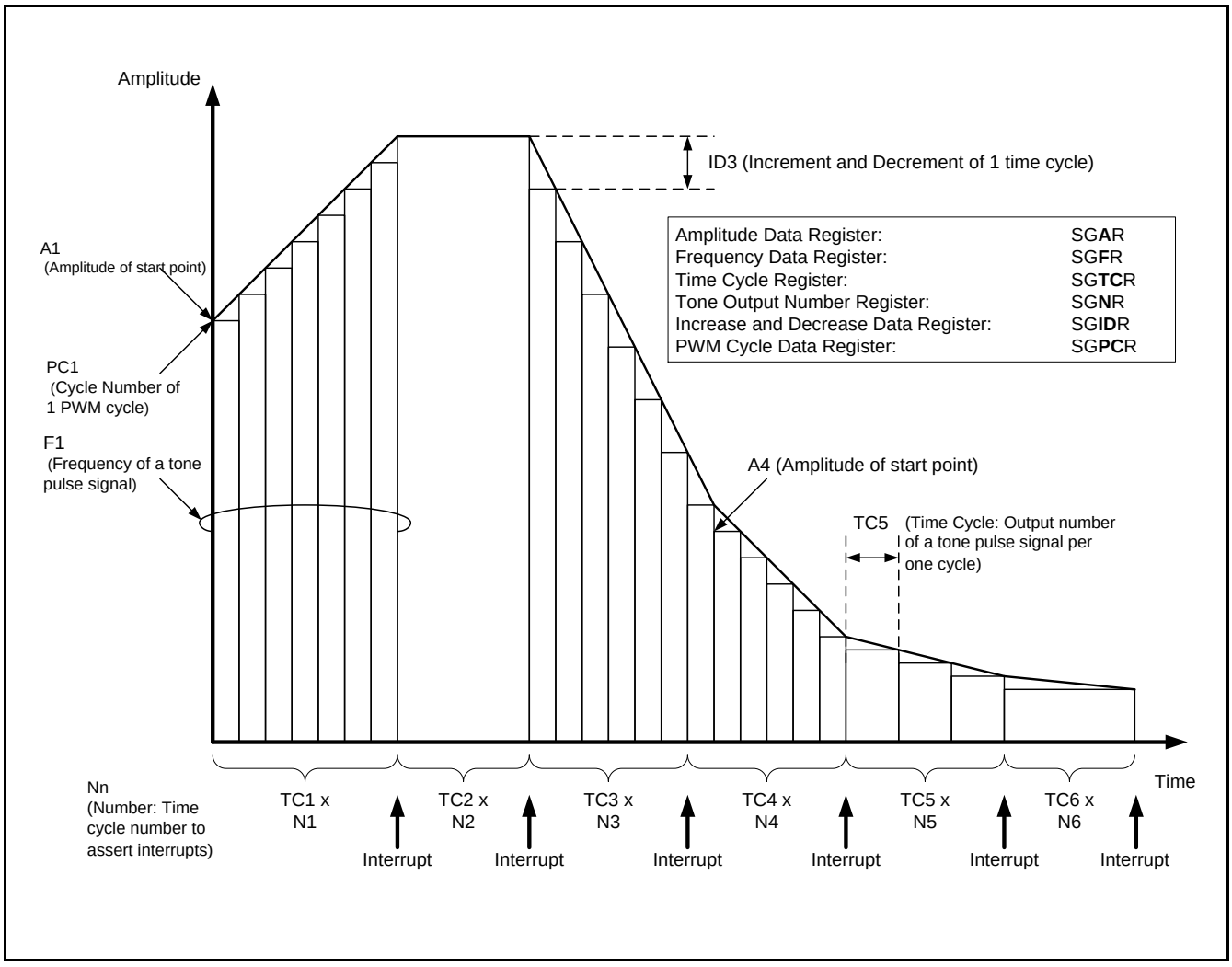


3. Operations

This chapter describes the Sound Generator operation.

The concept of Sound Generator operation is shown in the diagram as below.

Figure 3-1 The operation of Sound Generator (Conceptual Diagram)



Set the various values in the register as follows:

Amplitude of start point (An) (SGAR)	- to the Amplitude Data Register
Frequency of a tone pulse signal (Fn) (SGFR)	- to the Frequency Data Register
Output number of a tone pulse signal per one cycle (TCn)	- to the Time Cycle Register (SGTCR)
Time cycle number to assert interrupts (Nn)	- to the Tone Output Number Register

	(SGNR)
Increment and Decrement of 1 time cycle (IDn)	- to the Increase and Decrease Data Register (SGIDR)
Cycle Number of 1 PWM cycle (PCn)	- to the PWM Cycle Data Register (SGPCR)
Other information about Sound Generator control (SGCR)	- to the Sound Control Register

The Sound Generator outputs the tone pulse signal and the amplitude data according to the setting above.

In the conceptual diagram shown above, it outputs 6 steps of signal.

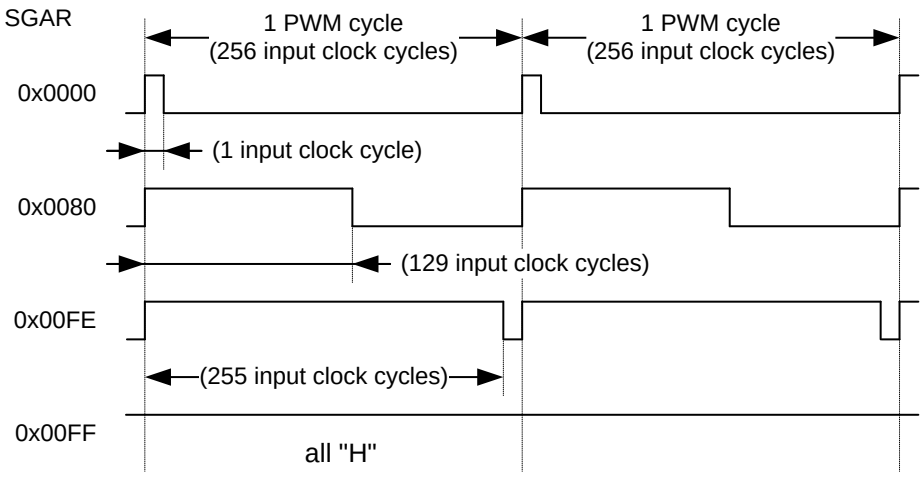
Each register, such as Amplitude Data Register, is supposed to be written at the following timing.

- Before starting signal output
- Each time after asserting an interrupt

3.1. Relation between the Amplitude Data Register (SGAR) and PWM Pulse

This section describes the relation between the Amplitude Data Register (SGAR) and a PWM pulse.

Figure 3-2 The Relation Between the Amplitude Data Register (SGAR) and PWM Pulse



The amplitude data is output as a PWM (Pulse Width Modulation) waveform on the SGA pin.

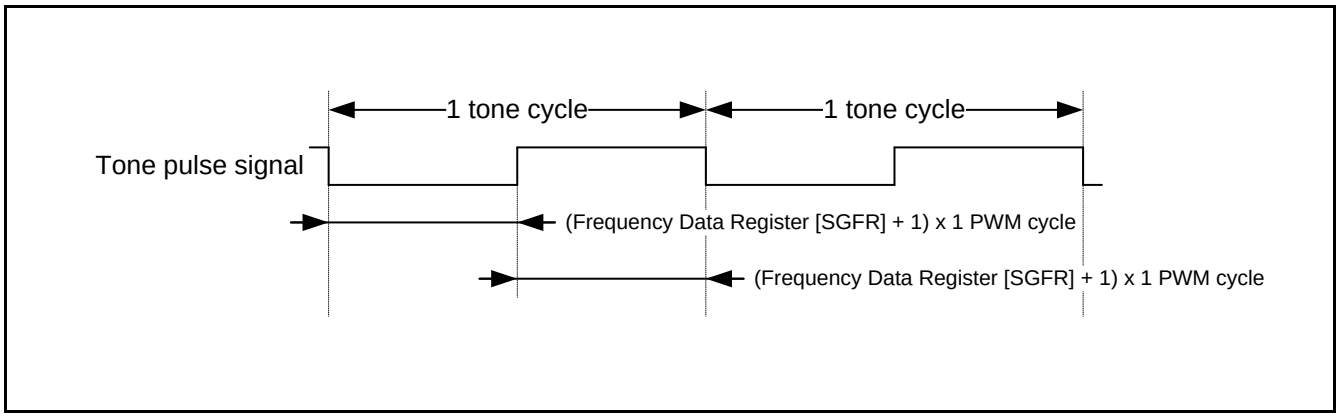
The length of a PWM cycle is programmable in the PWM Cycle Data Register (SGPCR). This length is based on the count of input clock. The initial value is 256 input clock cycle (SGPCR=0x00FF).

The value [Amplitude Data Register (SGAR) + 1] means the number of input clock cycles in which the SGA pin is "H" during 1 PWM cycle. Moreover, when the Amplitude Data Register (SGAR) is greater than or equal to the PWM Cycle Data Register (SGPCR), the SGA pin output is always "H".

3.2. Relation between the Frequency Data Register (SGFR) and Tone Pulse Signal

This section describes the relation between the Frequency Data Register (SGFR) and a tone pulse signal.

Figure 3-3 The Relation between the Frequency Data Register (SGFR) and Tone Pulse Signal



The tone pulse signal transits between "L" and "H" at every cycle of "(Frequency Data Register [SGFR] + 1) x 1 PWM cycle)". This signal is generated by an internal toggle flip-flop.

When the Tone output bit (TONE) of the Sound Control Register (SGCR) is "0", the tone pulse signal is mixed with a PWM pulse, and is output from the SGO pin. And when the Tone output bit (TONE) of the Sound Control Register (SGCR) is "1", the tone pulse signal is output without being mixed.

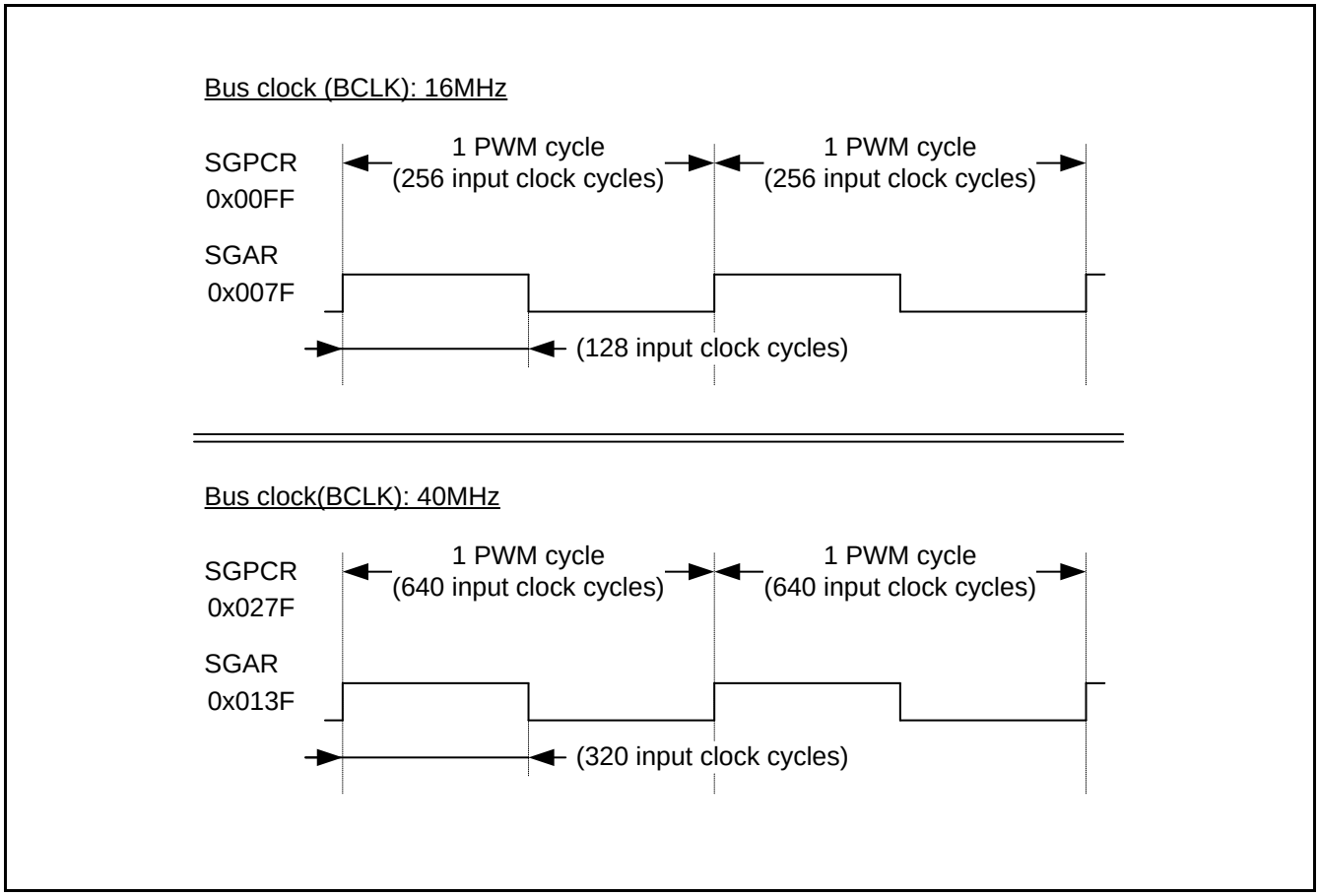
Note:

- *During operation, in the case of changing a register value, meet the following either conditions.*
  (1)Sound generator is in the stop state (SGCR.BUSY="0").
  (2)The changing comes between an interrupt occurrence and falling edge of the first tone pulse.
  (The limit time = (Frequency Data Register [SGFR] + 1) x 1 PWM cycle)
  When it meets neither condition, the sound output cannot guarantee the expected duty ratio.

3.3. Relation between the PWM Cycle Data Register (SGPCR) and PWM cycle

This section describes the relation between the PWM Cycle Data Register (SGPCR) and PWM cycle.

Figure 3-4 Relation between the PWM Cycle Data Register (SGPCR) and PWM cycle



The length of a PWM cycle is programmable in the PWM Cycle Data Register (SGPCR). This length is based on the count of input clock, and defined as "PWM Cycle Data Register [SGPCR] + 1". The input clock is created by dividing the bus clock.

The PWM cycle is a reference clock for a tone pulse signal (or a mixed signal of the tone pulse signal and a PWM pulse signal), and a PWM pulse signal.

To generate the same sound output both by the bus clock at 16 MHz and 40 MHz, for example, it can be made by changing the values of the PWM Cycle Data Register (SGPCR) and the Amplitude Data Register (SGAR). Because the bus clock proportion of 16 MHz and 40 MHz is 1:2.5, the same sound can be made by setting the values of these two registers to 1:2.5.

3.4. Relation between the DMA Transfer Update Enable Register (SGDER) and DMA Settings

This section describes the relation between the DMA Transfer Update Enable Register (SGDER) and DMA settings.

3.4.1. The Number of DMA Transfers

DMA Transfer Update Enable Register (SGDER) determines the number of DMA transfer, once or twice.

If all of SGDER.ARE1, SGDER.ARE0, SGDER.FRE and SGDER.NRE are "0", or

all of SGDER.TCRE, SGDER.IDRE, SGDER.PCRE1 and SGDER.PCRE0 are "0",

the DMA transfer occurs only once. Otherwise, the DMA transfer occurs twice.

3.4.2. DMA Transfer size

The DMA transfer size (1 byte, 2 bytes or 4 bytes) depends on the setting of the DMA Transfer Update Enable Register (SGDER).

The greater value of the following two becomes the DMA transfer size:

The value of SGDER.ARE1, SGDER.ARE0, SGDER.FRE, and SGDER.NRE,

The value of SGDER.TCRE, SGDER.IDRE, SGDER.PCRE1, and SGDER.PCRE0

The transfer size of 3 bytes is regarded as 4 bytes.

3.4.3. Transfer Byte Position in the DMA Transfer Intermediate Register (SGDMAR)

The DMA transfer byte position in the "DMA Transfer Intermediate Register (SGDMAR)" depends on the setting of the "DMA Transfer Update Enable Register (SGDER)" and the DMA transfer size.

If the size of a DMA transfer is less than 4 bytes, the byte position of DMA transfer is left-aligned.

The table below shows the relations between
"the setting of the DMA Transfer Update Enable Register (SGDER)"
and
"the byte position in the DMA Transfer Intermediate Register (SGDMAR)".
The byte position in the DMA Transfer Intermediate Register (SGDMAR) corresponds to the selection of the following registers.

- Amplitude Data Register (SGAR [15:0])
- Frequency Data Register (SGFR [7:0])
- Tone Output Number Register (SGNR [7:0])

The transfer size #1 is calculated by the setting of {SGDER.ARE1, SGDER.ARE0, SGDER.FRE, and SGDER.NRE} in the "DMA Transfer Update Enable Register (SGDER)". When this transfer size #1 is not 4 bytes, the transfer byte position is left-aligned.

Table 3-1 The Relations between the setting of the DMA Transfer Update Enable Register (SGDER) and the Transfer Byte Position of SGDMAR #1

No.	Setting of SGDER				Transfer Size #1 *	Transfer Byte Position of SGDMAR			
	ARE1	ARE0	FRE	NRE		SGAR [15:8]	SGAR [7:0]	SGFR [7:0]	SGNR [7:0]
1	0	0	0	0	0	-	-	-	-
2	1	0	0	0	1	SGDMAR [31:24]	-	-	-
3	0	1	0	0	1	-	SGDMAR [31:24]	-	-
4	1	1	0	0	2	SGDMAR [31:24]	SGDMAR [23:16]	-	-
5	0	0	1	0	1	-	-	SGDMAR [31:24]	-
6	1	0	1	0	2	SGDMAR [31:24]	-	SGDMAR [23:16]	-
7	0	1	1	0	2	-	SGDMAR [31:24]	SGDMAR [23:16]	-
8	1	1	1	0	4	SGDMAR [31:24]	SGDMAR [23:16]	SGDMAR [15:8]	-
9	0	0	0	1	1	-	-	-	SGDMAR [31:24]
10	1	0	0	1	2	SGDMAR [31:24]	-	-	SGDMAR [23:16]
11	0	1	0	1	2	-	SGDMAR [31:24]	-	SGDMAR [23:16]
12	1	1	0	1	4	SGDMAR [31:24]	SGDMAR [23:16]	-	SGDMAR [15:8]
13	0	0	1	1	2	-	-	SGDMAR [31:24]	SGDMAR [23:16]
14	1	0	1	1	4	SGDMAR [31:24]	-	SGDMAR [23:16]	SGDMAR [15:8]
15	0	1	1	1	4	-	SGDMAR [31:24]	SGDMAR [23:16]	SGDMAR [15:8]
16	1	1	1	1	4	SGDMAR [31:24]	SGDMAR [23:16]	SGDMAR [15:8]	SGDMAR [7:0]

*: The transfer size which is calculated by the setting of SGDER.ARE1, SGDER.ARE0, SGDER.FRE, and SGDER.NRE.

- : Do not care.

The table below shows the relations between
"the setting of the DMA Transfer Update Enable Register (SGDER)"
and
"the byte position in the DMA Transfer Intermediate Register (SGDMAR)".
The byte position in the DMA Transfer Intermediate Register (SGDMAR) corresponds to the selection of the following registers.
- Time Cycle Register (SGTCR[7:0])
- Increase and Decrease Data Register (SGIDR[7:0])
- PWM Cycle Data Register (SGPCR[15:0])

The transfer size #2 is calculated by the setting of {SGDER.TCRE, SGDER.IDRE, SGDER.PCRE1, and SGDER.PCRE0} in the DMA Transfer Update Enable Register (SGDER). When this transfer size #2 is not 4 bytes, the transfer byte position is left-aligned.

Table 3-2 The Relations Between the setting of the DMA Transfer Update Enable Register (SGDER) and Transfer Byte Position of SGDMAR #2

No.	Setting of SGDER				Transfer Size #2 *	Transfer Byte Position of SGDMAR			
	TCRE	IDRE	PCRE1	PCRE0		SGTCR [7:0]	SGIDR [7:0]	SGPCR [15:8]	SGPCR [7:0]
1	0	0	0	0	0	-	-	-	-
2	1	0	0	0	1	SGDMAR [31:24]	-	-	-
3	0	1	0	0	1	-	SGDMAR [31:24]	-	-
4	1	1	0	0	2	SGDMAR [31:24]	SGDMAR [23:16]	-	-
5	0	0	1	0	1	-	-	SGDMAR [31:24]	-
6	1	0	1	0	2	SGDMAR [31:24]	-	SGDMAR [23:16]	-
7	0	1	1	0	2	-	SGDMAR [31:24]	SGDMAR [23:16]	-
8	1	1	1	0	4	SGDMAR [31:24]	SGDMAR [23:16]	SGDMAR [15:8]	-
9	0	0	0	1	1	-	-	-	SGDMAR [31:24]
10	1	0	0	1	2	SGDMAR [31:24]	-	-	SGDMAR [23:16]
11	0	1	0	1	2	-	SGDMAR [31:24]	-	SGDMAR [23:16]
12	1	1	0	1	4	SGDMAR [31:24]	SGDMAR [23:16]	-	SGDMAR [15:8]
13	0	0	1	1	2	-	-	SGDMAR [31:24]	SGDMAR [23:16]
14	1	0	1	1	4	SGDMAR [31:24]	-	SGDMAR [23:16]	SGDMAR [15:8]
15	0	1	1	1	4	-	SGDMAR [31:24]	SGDMAR [23:16]	SGDMAR [15:8]
16	1	1	1	1	4	SGDMAR [31:24]	SGDMAR [23:16]	SGDMAR [15:8]	SGDMAR [7:0]

*: The transfer size which is calculated by the setting of SGDER.TCRE, SGDER.IDRE, SGDER.PCRE1, and SGDER.PCRE0.

- : Do not care.

3.4.4. DMA Transfer Image

This section shows an example of DMA transfer image, under the setting of DMA Transfer Update Enable Register (SGDER):

[SGDER.ARE1, SGDER.ARE0, SGDER.FRE, SGDER.NRE] ="1001"

[SGDER.TCRE, SGDER.IDRE, SGDER.PCRE1, SGDER.PCRE0] ="0100"

Condition:

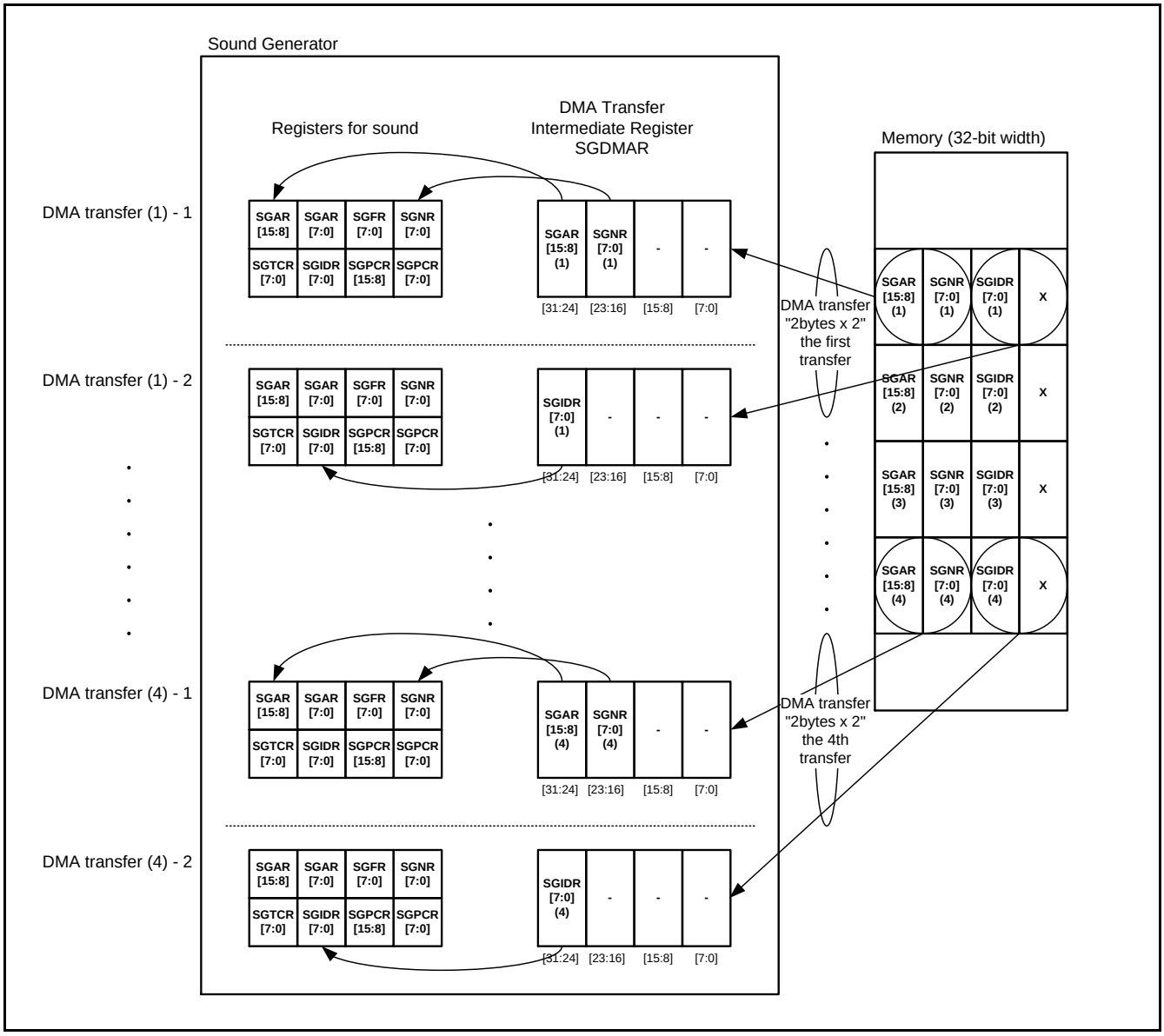
The number of times of DMA transfer : 2 times

DMA transfer size : 2 bytes

Transfer byte position of the DMA Transfer Intermediate Register (SGDMAR):

The first	SGDMAR[31:24]	← The amplitude data (upper byte) / SGAR[15:8]
	SGDMAR[23:16]	← The tone output number / SGNR[7:0]
	SGDMAR[15:0]	← Do not care
The second	SGDMAR[31:24]	← The increment and decrement data / SGIDR[7:0]
	SGDMAR[23:0]	← Do not care

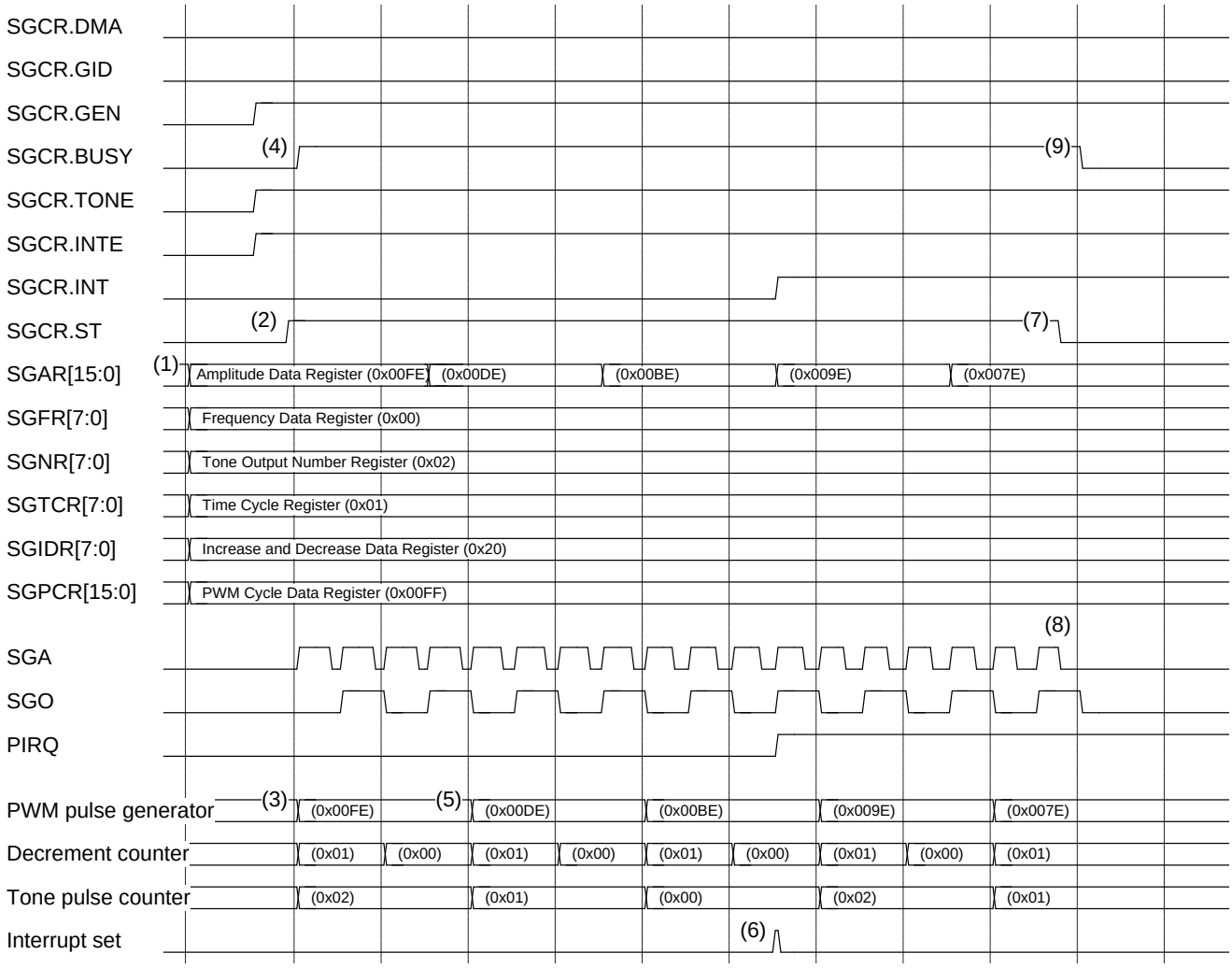
Figure 3-5 DMA Transfer Image



3.5. Sound Generator Operation

This section describes the Sound Generator operation as below

Figure 3-6 Sound Generator operation



(1) The reload values are written to the Amplitude Data Register (SGAR), the Frequency Data Register (SGFR), the Tone Output Number Register (SGNR), and the Time Cycle Register (SGTCR) by software. Also the selection of increment and decrement of the amplitude is written to the Increase and Decrease Data Register (SGIDR) by software as well as the number of cycles in 1 PWM cycle to the PWM Cycle Data Register (SGPCR).

Moreover, set other information to the Sound Control Register (SGCR) to control the Sound Generator. Initialize the Interrupt status bit (SGCR.INT) and set the Interrupt enable bit (SGCR.INTE).

(2) Write "1" to the Start bit (SGCR.ST).

(3) By setting "1" to the Start bit (SGCR.ST), the Amplitude Data Register (SGAR) value is loaded into the PWM pulse generator, the Frequency Data Register (SGFR) value into the Frequency counter, the Tone Output Number Register (SGNR) value into the Tone pulse counter, the Time Cycle Register (SGTCR) value into the Decrement counter.

(4) The operation flag (SGCR.BUSY) is automatically set.

(5) Due to the operation that the Decrement counter counts the number of tone pulses until it reaches the reload value, the Amplitude Data Register (SGAR) value decreases according to the setting of the Automatic increase/decrease enable bit (SGCR.GEN) and the Increase/decrease setting bit (SGCR.GID).

(6) The Tone pulse counter counts the number of tone pulses. When the following conditions are satisfied, it sets the Interrupt status bit (SGCR.INT) and asserts the Interrupt request (PIRQ).
- Tone pulse counter is 0x00
- Decrement counter is 0x00
- At the rising edge of SGO

(7) Write "0" to the Start bit (SGCR.ST). The Sound Generator keeps operating until the Busy status bit (SGCR.BUSY) turns "0".

(8) The Sound Generator stops when the current tone cycle finishes.

(9) The Busy status bit (SGCR.BUSY) turns "0".

DMA transfer start interrupt setting enable bit (SGCR.DMA)

To assert the first Interrupt request (PIRQ) after starting (SGCR.ST="1"), there are two modes for the request. And the selection depends on the setting of the "DMA transfer start interrupt setting enable bit (SGCR.DMA)".

- Normal mode :
  When Sound Generator outputs the tone pulses, to the number programmed in the Time Cycle Register (SGTCR)
- DMA mode :
  Immediately after the setting of Start bit (writing "1" to SGCR.ST)
  (In this case, the Interrupt request (PIRQ) is regarded as the DMA transfer request.)

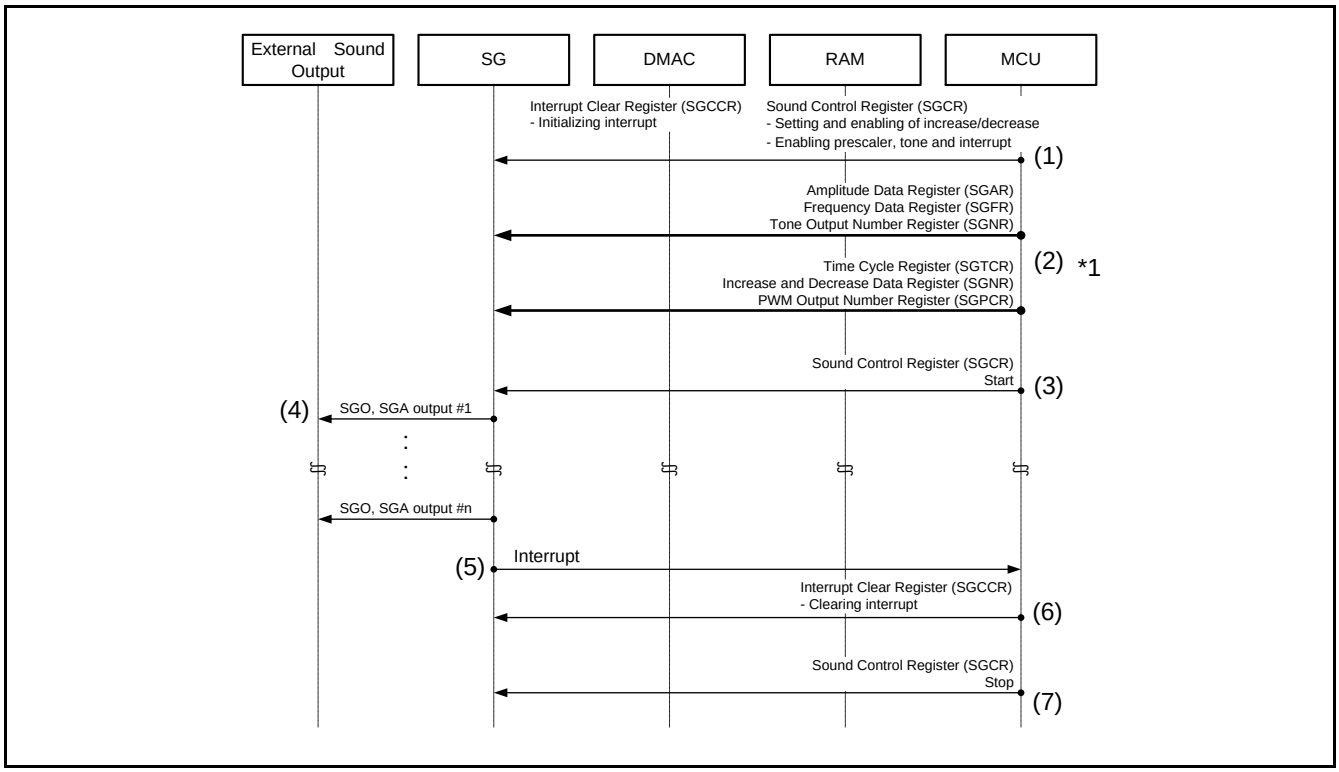
DMA transfer

The following registers are set through the DMA Transfer Intermediate Register (SGDMAR).
- Amplitude Data Register (SGAR)
- Frequency Data Register (SGFR)
- Time Cycle Register (SGTCR)
- Tone Output Number Register (SGNR)
- Increase and Decrease Data Register (SGIDR)
- PWM Cycle Data Register (SGPCR)

3.5.1. Single operation of Sound Generator by MCU

This section shows the single operation flow of the Sound Generator by MCU.

Figure 3-7 Single operation of Sound Generator by MCU



(1) Set the information to control the Sound Generator to the Sound Control Register (SGCR) by software. Initialize the Interrupt status bit (SGCR.INT) and set the Interrupt enable bit (SGCR.INTE).

(2) Set registers "Amplitude Data Register (SGAR)", "Frequency Data Register (SGFR)", "Tone Output Number Register (SGNR)", "Time Cycle Register (SGTCR)", "Increase and Decrease Data Register (SGIDR)" and "PWM Cycle Data Register (SGPCR)" by software. (*: Set only the necessary registers.)

(3) Write "1" to the Start bit (SGCR.ST).

(4) The outputs of SGO and SGA start.

(5) The Tone pulse counter counts the number of tone pulses. When the following conditions are satisfied, the interrupt is generated.
- Tone pulse counter is 0x00
- Decrement counter is 0x00
- At the rising edge of SGO

(6) MCU clears the interrupt.

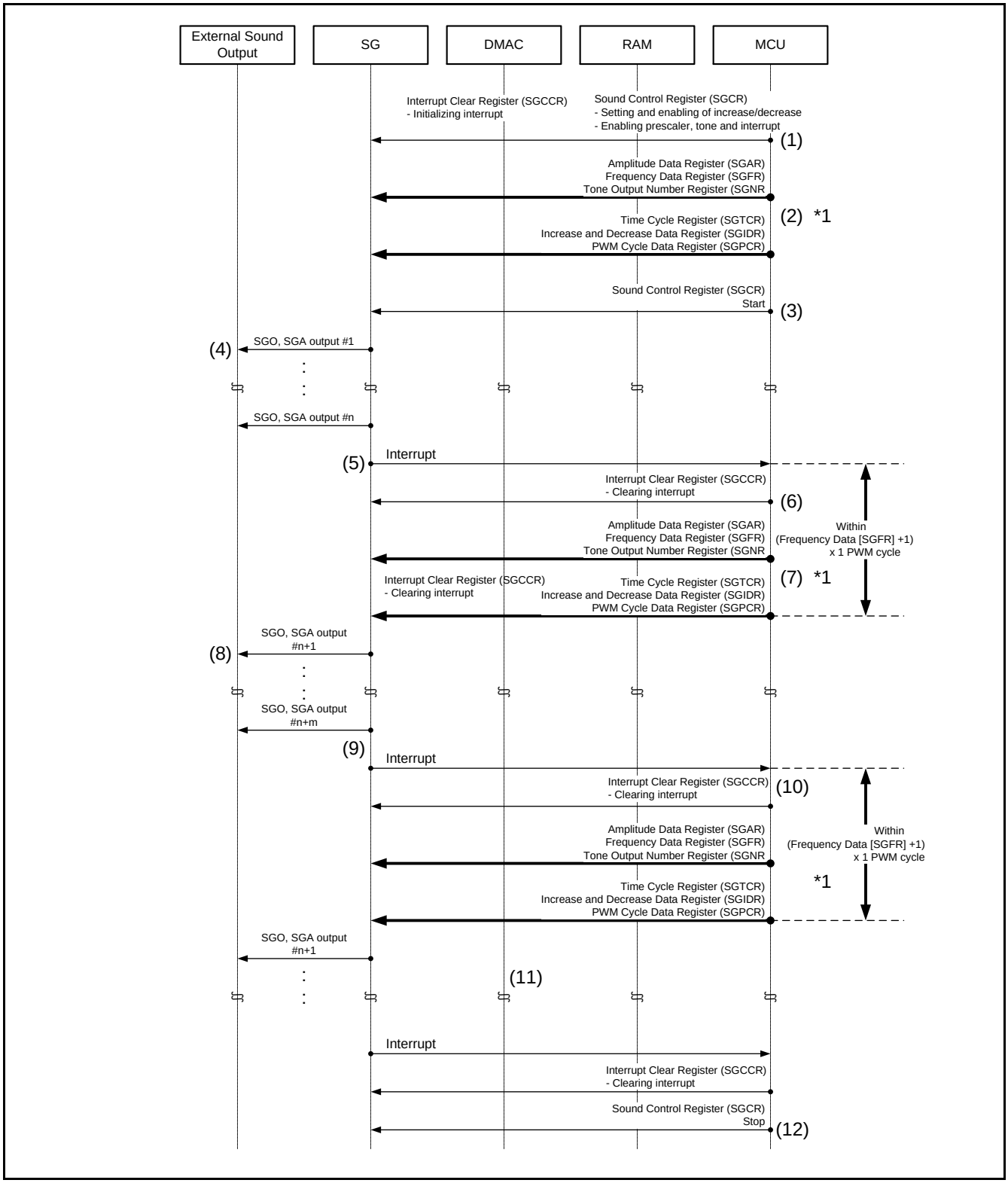
(7) Software writes "0" to the Start bit (SGCR.ST) to stop outputting the sound.

3.6. Continuous operation of Sound Generator by MCU

This section shows the continuous operation flow of the Sound Generator by MCU.

The flow from (7) is different from that of the single operation.

Figure 3-8 Continuous operation of Sound Generator by MCU



(1) Set the information to control the Sound Generator to the Sound Control Register (SGCR) by software. Initialize the Interrupt status bit (SGCR.INT) and set the Interrupt enable bit (SGCR.INTE).

(2) Set registers "Amplitude Data Register (SGAR)", "Frequency Data Register (SGFR)", "Tone Output Number Register (SGNR)", "Time Cycle Register (SGTCR)", "Increase and Decrease Data Register (SGIDR)" and "PWM Cycle Data Register (SGPCR)" by software. (*: Set only the necessary registers.)

(3) Write "1" to the Start bit (SGCR.ST).

(4) The outputs of SGO and SGA start.

(5) The Tone pulse counter counts the number of tone pulses. When the following conditions are satisfied, the interrupt is generated.
- Tone pulse counter is 0x00
- Decrement counter is 0x00
- At the rising edge of SGO

(6) MCU clears the interrupt.

(7) Set registers "Amplitude Data Register (SGAR)", "Frequency Data Register (SGFR)", "Tone Output Number Register (SGNR)", "Time Cycle Register (SGTCR)", "Increase and Decrease Data Register (SGIDR)" and "PWM Cycle Data Register (SGPCR)" by software. (*: Set only the necessary registers.)

(8) The Sound Generator keeps outputting SGO and SGA, according to the register settings above.

(9) The Tone pulse counter counts the number of tone pulses. When the following conditions are satisfied, the interrupt is generated.
- Tone pulse counter is 0x00
- Decrement counter is 0x00
- At the rising edge of SGO

(10) MCU clears the interrupt.

(11) Repeat the flow from 7 to 11 to continue outputting the sound.

(12) Software writes "0" to the Start bit (SGCR.ST) to stop outputting the sound.

Notes:

- *The software must finish the procedure from step 5 to 7, within the following time.*
  The limit time = (Frequency Data Register [SGFR] + 1) x 1 PWM cycle
- *In case of switching the selection of increase/decrease, it is needed to write "Increase/decrease setting bit (SGCR.GID)" and the "Automatic increase/decrease enable bit (SGCR.GEN)" in the Sound Control Register (SGCR) within above-mentioned limit time.*

3.7. Sound Generator operation with DMA

This section shows the flow of the Sound Generator operation with DMA.

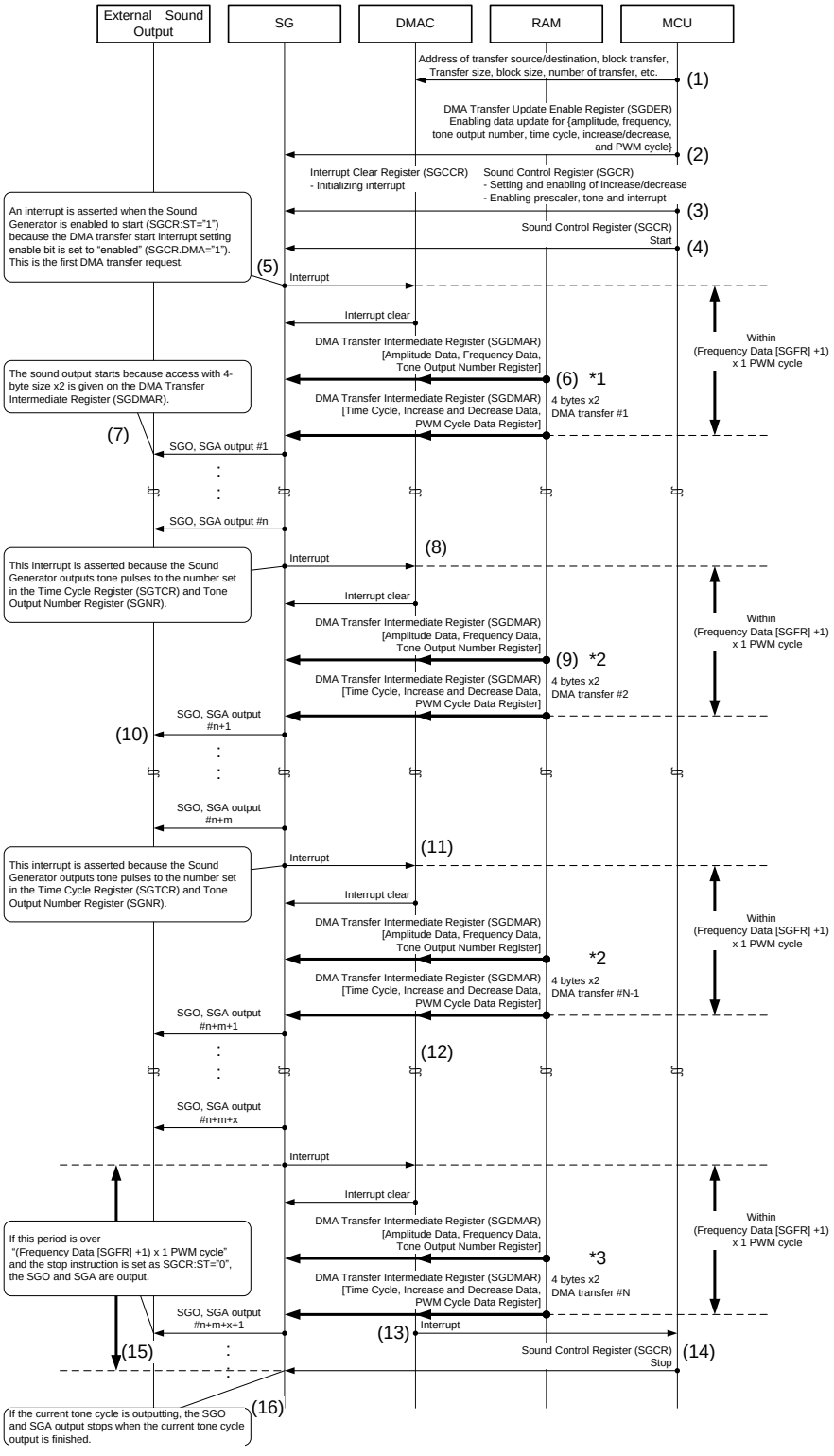
DMAC writes a register related to DMA in the Sound Generator. The timing of the first interrupt with DMA is different from the flow by MCU. The DMAC writes functional registers in the Sound Generator, through the DMA Transfer Intermediate Register (SGDMAR) which works as a window register.

Note:

- *Software needs to set the Interrupt enable bit (SGCR.INTE="1") in order to use the Interrupt request (PIRQ) as a DMA transfer request.*

3.7.1. In Case of DMA Transfer with 4-Byte size x 2 is Made N Times

Figure 3-9 Sound Generator operation with DMA (When in DMA Transfer with 4-Byte size x 2 is Made N Times)



(1) Software makes initial settings to those registers which are needed in DMA transfer. The DMA transfer is based on a block data of "4-byte size x 2", and this block can be repeated N times. DMAC sets following registers to prepare for a DMA transfer, through the "DMA Transfer Intermediate Register (SGDMAR)".
- Amplitude Data Register (SGAR)
- Frequency Data Register (SGFR)
- Tone Output Number Register (SGNR)
- Time Cycle Register (SGTCR)
- Increase and Decrease Data Register (SGIDR)
- PWM Cycle Data Register (SGPCR)

The destination address of the DMA transfer is a fixed one on the "DMA Transfer Intermediate Register (SGDMAR)".

(2) Software configures the "DMA Transfer Update Enable Register (SGDER)" to enable the automatic update of the following registers during DMA transfer.
- Amplitude Data Register (SGAR)
- Frequency Data Register (SGFR)
- Tone Output Number Register (SGNR)
- Time Cycle Register (SGTCR)
- Increase and Decrease Data Register (SGIDR)
- PWM Cycle Data Register (SGPCR)

(3) Software initialize Interrupt status bit (SGCR.INT) by writing "1" to Interrupt status clear bit (SGCCR.INTC). Then, the software configures the Sound Control Register (SGCR) in the needed mode, and this must include the following bit operations.
- DMA transfer start interrupt setting enable bit (SGCR.DMA) to "1"(enabled)
- Interrupt enable bit (SGCR.INTE) to "1"(enabled)

(4) Write "1" to the Start bit (SGCR.ST).

(5) The interrupt occurs immediately after setting Start bit (SGCR.ST), since the Sound Generator is enabled on DMA transfer (SGCR.DMA="1"). An Interrupt request (PIRQ) is asserted, and this interrupt is used as a DMA transfer request.

(6) DMAC clears the interrupt, and write registers in the Sound Generator through the "DMA Transfer Intermediate Register (SGDMAR)". This operation configures the following registers by 2 steps.
[1st step]
- Amplitude Data Register (SGAR)
- Frequency Data Register (SGFR)
- Tone Output Number Register (SGNR)

[2nd step]
- Time Cycle Register (SGTCR)
- Increase and Decrease Data Register (SGIDR)
- PWM Cycle Data Register (SGPCR)

(*: DMA block size must to be "4-byte size x 2" for the access to "DMA Transfer Intermediate Register (SGDMAR)")

(7) The outputs of SGO and SGA start, according to the register settings above.

(8) The Tone pulse counter counts the number of tone pulses. When the following conditions are satisfied, the interrupt is generated.
- Tone pulse counter is 0x00
- Decrement counter is 0x00
- At the rising edge of SGO

(9) DMAC clears the interrupt, and write registers in the Sound Generator through the "DMA Transfer Intermediate Register (SGDMAR)".
(*2: DMA block size must to be "4-byte size x 2" for the access to "DMA Transfer Intermediate Register (SGDMAR)")

(10) The Sound Generator keeps outputting SGO and SGA, according to the register settings above.

(11) The Tone pulse counter counts the number of tone pulses. When the following conditions are satisfied, the interrupt is generated.
- Tone pulse counter is 0x00
- Decrement counter is 0x00
- At the rising edge of SGO

DMAC doesn't assert an interrupt to MCU until the DMAC completes all DMA transfer (with 4-byte size x2, N times).

(12) Repeat the flow from 9 to 11 to continue outputting the sound.

(13) When DMAC completes all DMA transfer (with 4-byte size x2, N times), it asserts an interrupt to MCU.

(14) Software writes "0" to the Start bit (SGCR.ST) to stop outputting the sound.
(15) When above operation (14) was made within the following time, the Nth DMA transfer doesn't come to the output of SGO and SGA. The Sound Generator stops driving SGO and SGA just before outputting the data of Nth DMA transfer.
The limit time = (Frequency Data Register [SGFR] + 1) x 1 PWM cycle
(*3: The data of the Nth DMA transfer are written to the Sound Generator, however, they are not output. DMAC issues this Nth DMA transfer only to assert an interrupt toward the MCU.)

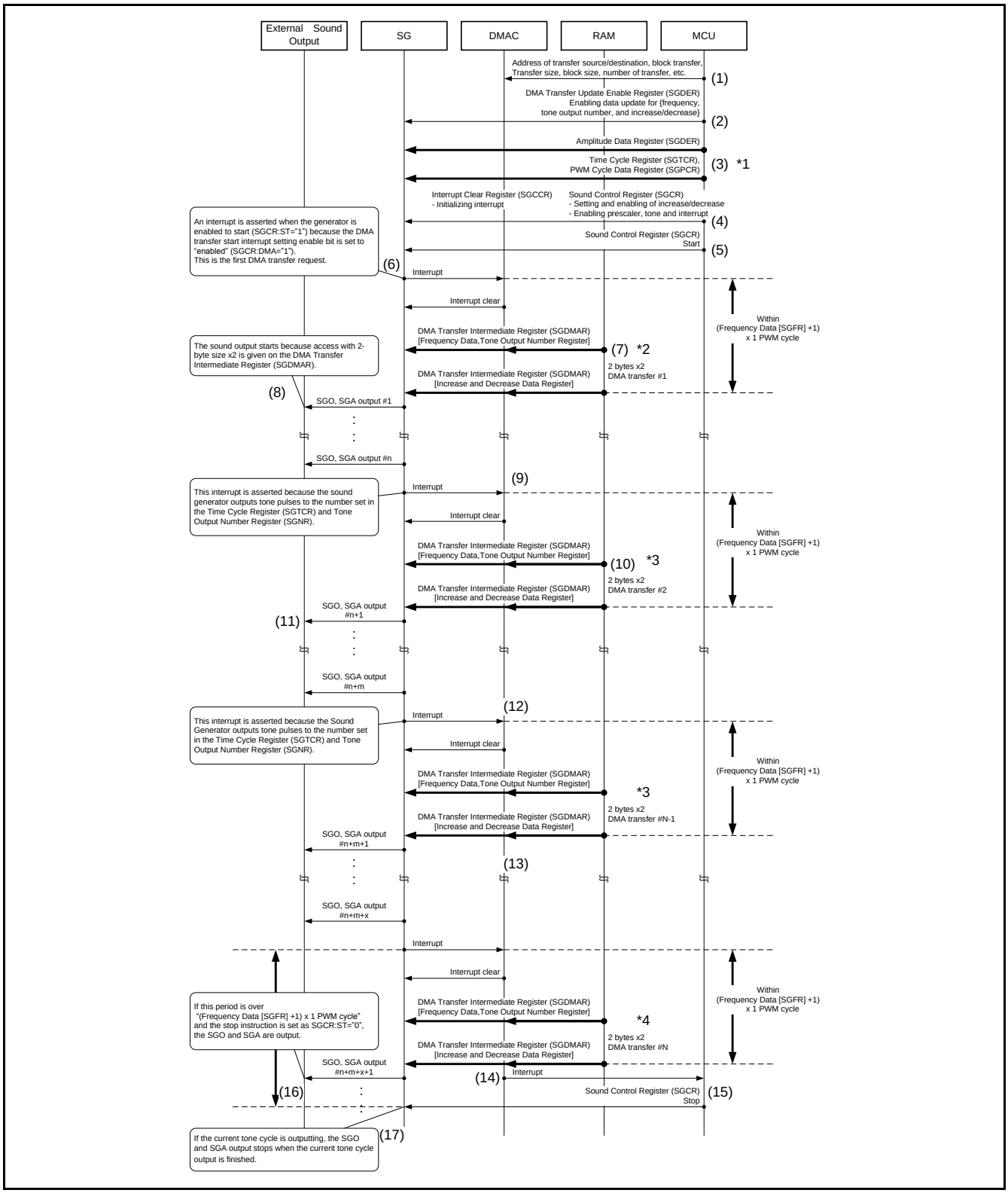
(16) If above (15) is not done within the limit time, the Sound Generator keeps driving SGO and SGA in order to output the data of the Nth DMA transfer. Then, the Sound Generator stops driving SGO and SGA after the end of all data.
(*3: The data of the Nth DMA transfer are written to the Sound Generator, and they are output to the end.)

Notes:

- *The DMAC must finish the sequence from step 5 to 6, within the following time.*
  The limit time = (Frequency Data Register [SGFR] + 1) x 1 PWM cycle
- *The DMA transfer error means the occurrence of delay in the sound data setting. It causes unsteady sound output. In that case, please fix the priority of DMA transfer in the system to finish all data transfer within the limit time.*

3.7.2. In Case of DMA Transfer with 2-Byte size x 2 is Made N Times

Figure 3-10 Sound Generator operation with DMA (When DMA Transfer with 2-Byte size x 2 is Made N Times)



(1) Software makes initial settings to those registers which are needed in DMA transfer. The DMA transfer is based on a block data of "2-byte size x 2", and this block can be repeated N times. DMAC sets following registers to prepare for a DMA transfer, through the "DMA Transfer Intermediate Register (SGDMAR)".
- Frequency Data Register (SGFR)
- Tone Output Number Register (SGNR)
- Increase and Decrease Data Register (SGIDR)
The destination address of the DMA transfer is a fixed one on the "DMA Transfer Intermediate Register (SGDMAR)".

(2) Software configures the "DMA Transfer Update Enable Register (SGDER)" to enable the automatic update of the following registers during DMA transfer.
- Frequency Data Register (SGFR)
- Tone Output Number Register (SGNR)
- Increase and Decrease Data Register (SGIDR)

(3) Software configures the "DMA Transfer Update Enable Register (SGDER)" to disable the automatic update of the following registers during DMA transfer.
- Amplitude Data Register (SGAR)
- Time Cycle Register (SGTCR)
- PWM Cycle Data Register (SGPCR)
(*: Set values to these registers which are not automatically updated.)

(4) Software initialize Interrupt status bit (SGCR.INT) by writing "1" to Interrupt status clear bit (SGCCR.INTC). Then, the software configures the Sound Control Register (SGCR) in the needed mode, and this must include the following bit operations.
- DMA transfer start interrupt setting enable bit (SGCR.DMA) to "1"(enabled)
- Interrupt enable bit (SGCR.INTE) to "1"(enabled)

(5) Write "1" to the Start bit (SGCR.ST).

(6) The interrupt occurs immediately after setting Start bit (SGCR.ST), since the Sound Generator is enabled on DMA transfer (SGCR.DMA="1"). An Interrupt request (PIRQ) is asserted, and this interrupt is used as a DMA transfer request.

(7) DMAC clears the interrupt, and write registers in the Sound Generator through the "DMA Transfer Intermediate Register (SGDMAR)". This operation configures the following registers by 2 steps.
[1st step]
- Frequency Data Register (SGFR)
- Tone Output Number Register (SGNR)
[2nd step]
- Increase and Decrease Data Register (SGIDR)
(*2: DMA block size must to be "2-byte size x 2" for the access to "DMA Transfer Intermediate Register (SGDMAR)")

(8) The outputs of SGO and SGA start, according to the register settings above.

(9) The Tone pulse counter counts the number of tone pulses. When the following conditions are satisfied, the interrupt is generated.
- Tone pulse counter is 0x00
- Decrement counter is 0x00
- At the rising edge of SGO

(10) DMAC clears the interrupt, and write registers in the Sound Generator through the "DMA Transfer Intermediate Register (SGDMAR)".

(*3: DMA block size must to be "2-byte size x 2" for the access to "DMA Transfer Intermediate Register (SGDMAR)")

(11) The Sound Generator keeps outputting SGO and SGA, according to the register settings above.

(12) The Tone pulse counter counts the number of tone pulses. When the following conditions are satisfied, the interrupt is generated.
- Tone pulse counter is 0x00
- Decrement counter is 0x00
- At the rising edge of SGO

DMAC doesn't assert an interrupt to MCU until the DMAC completes all DMA transfer (with 2-byte size x2, N times).

(13) Repeat the flow from 10 to 12 to continue outputting the sound.

(14) When DMAC completes all DMA transfer (with 2-byte size x2, N times), it asserts an interrupt to MCU.

(15) Software writes "0" to the Start bit (SGCR.ST) to stop outputting the sound.

(16) When above operation (15) was made within the following time, the Nth DMA transfer doesn't come to the output of SGO and SGA. The Sound Generator stops driving SGO and SGA just before outputting the data of Nth DMA transfer.
The limit time = (Frequency Data Register [SGFR] + 1) x 1 PWM cycle
(*4: The data of the Nth DMA transfer are written to the Sound Generator, however, they are not output. DMAC issues this Nth DMA transfer only to assert an interrupt toward the MCU.)

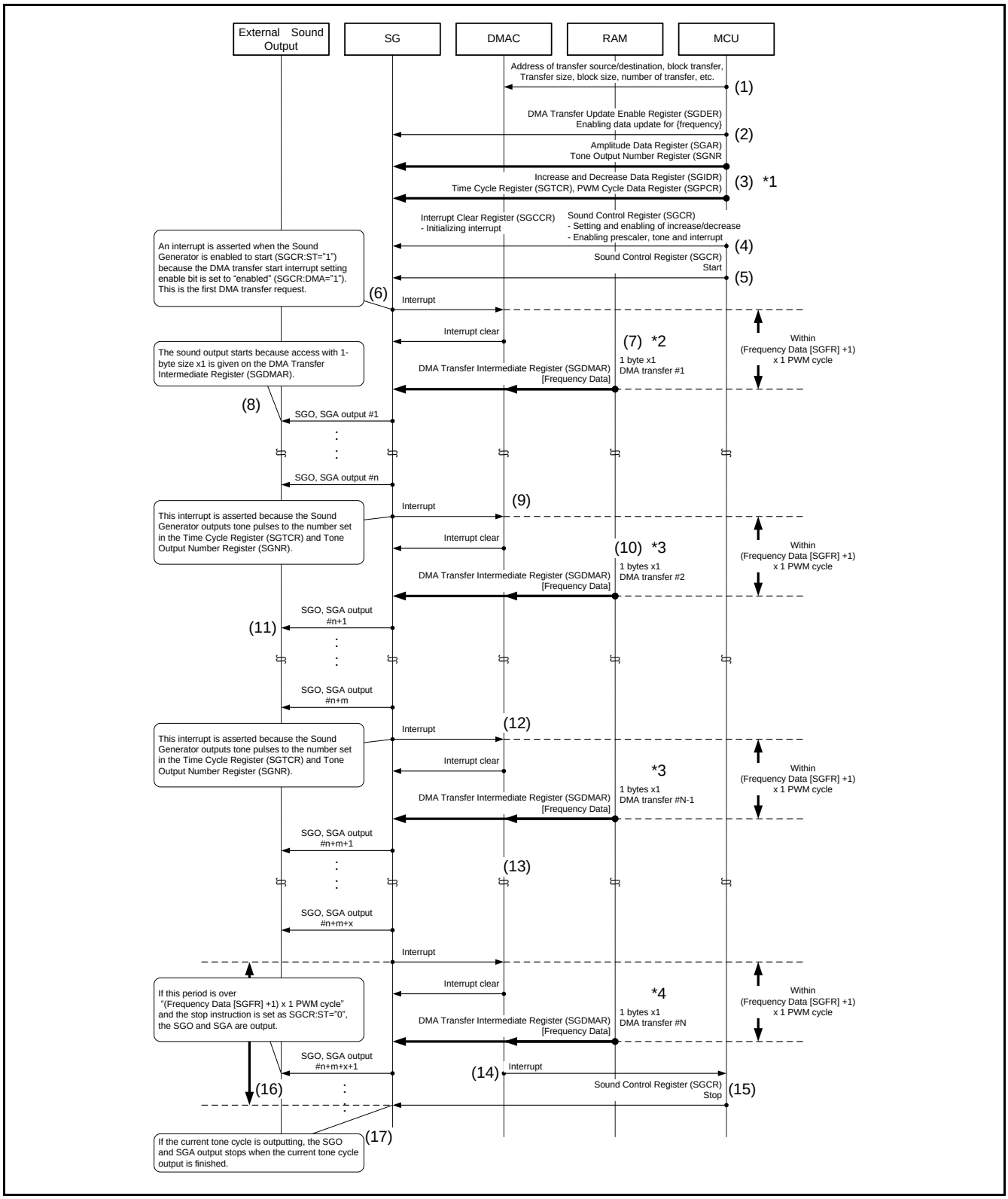
(17) If above (16) is not done within the limit time, the Sound Generator keeps driving SGO and SGA in order to output the data of the Nth DMA transfer. Then, the Sound Generator stops driving SGO and SGA after the end of all data.
(*4: The data of the Nth DMA transfer are written to the Sound Generator, and they are output to the end.)

Notes:

- *The DMAC must finish the sequence from step 6 to 7, within the following time.*
  The limit time = (Frequency Data Register [SGFR] + 1) x 1 PWM cycle
- *The DMA transfer error means the occurrence of delay in the sound data setting. It causes unsteady sound output. In that case, please fix the priority of DMA transfer in the system to finish all data transfer within the limit time.*

3.7.3. In Case of DMA Transfer with 1-Byte size x 1 is Made N Times

Figure 3-11 Sound Generator operation with DMA (When in DMA Transfer with 1-Byte size x 1 is Made N Times)



(1) Software makes initial settings to those registers which are needed in DMA transfer. The DMA transfer is based on a block data of "1-byte size x 2", and this block can be repeated N times. DMAC sets following registers to prepare for a DMA transfer, through the "DMA Transfer Intermediate Register (SGDMAR)".
- Frequency Data Register (SGFR)
The destination address of the DMA transfer is a fixed one on the "DMA Transfer Intermediate Register (SGDMAR)".

(2) Software configures the "DMA Transfer Update Enable Register (SGDER)" to enable the automatic update of the following registers during DMA transfer.
- Frequency Data Register (SGFR)

(3) Software configures the "DMA Transfer Update Enable Register (SGDER)" to disable the automatic update of the following registers during DMA transfer.
- Amplitude Data Register (SGAR)
- Tone Output Number Register (SGNR)
- Time Cycle Register (SGTCR)
- Increase and Decrease Data Register (SGIDR)
- PWM Cycle Data Register (SGPCR)
(*: Set values to these registers which are not automatically updated.)

(4) Software initialize Interrupt status bit (SGCR.INT) by writing "1" to Interrupt status clear bit (SGCCR.INTC). Then, the software configures the Sound Control Register (SGCR) in the needed mode, and this must include the following bit operations.
- DMA transfer start interrupt setting enable bit (SGCR.DMA) to "1"(enabled)
- Interrupt enable bit (SGCR.INTE) to "1"(enabled)

(5) Write "1" to the Start bit (SGCR.ST).

(6) The interrupt occurs immediately after setting Start bit (SGCR.ST), since the Sound Generator is enabled on DMA transfer (SGCR.DMA="1"). An Interrupt request (PIRQ) is asserted, and this interrupt is used as a DMA transfer request.

(7) DMAC clears the interrupt, and write registers in the Sound Generator through the "DMA Transfer Intermediate Register (SGDMAR)". This operation configures the following register.
- Frequency Data Register (SGFR)
(*2: DMA block size must to be "1-byte size x 1" for the access to "DMA Transfer Intermediate Register (SGDMAR)")

(8) The outputs of SGO and SGA start, according to the register settings above.

(9) The Tone pulse counter counts the number of tone pulses. When the following conditions are satisfied, the interrupt is generated.
- Tone pulse counter is 0x00
- Decrement counter is 0x00
- At the rising edge of SGO

(10) DMAC clears the interrupt, and write registers in the Sound Generator through the "DMA Transfer Intermediate Register (SGDMAR)".
(*3: DMA block size must to be "1-byte size x 1" for the access to "DMA Transfer Intermediate Register (SGDMAR)")

(11) The Sound Generator keeps outputting SGO and SGA, according to the register settings above.

(12) The Tone pulse counter counts the number of tone pulses. When the following conditions are satisfied, the interrupt is generated.
- Tone pulse counter is 0x00

- Decrement counter is 0x00
- At the rising edge of SGO

DMAC doesn't assert an interrupt to MCU until the DMAC completes all DMA transfer (with 1-byte size x1, N times).

(13) Repeat the flow from 10 to 12 to continue outputting the sound.

(14) When DMAC completes all DMA transfer (with 1-byte size x1, N times), it asserts an interrupt to MCU.

(15) Software writes "0" to the Start bit (SGCR.ST) to stop outputting the sound.

(16) When above operation (15) was made within the following time, the Nth DMA transfer doesn't come to the output of SGO and SGA. The Sound Generator stops driving SGO and SGA just before outputting the data of Nth DMA transfer.
The limit time = (Frequency Data Register [SGFR] + 1) x 1 PWM cycle
(*4: The data of the Nth DMA transfer are written to the Sound Generator, however, they are not output. DMAC issues this Nth DMA transfer only to assert an interrupt toward the MCU.)

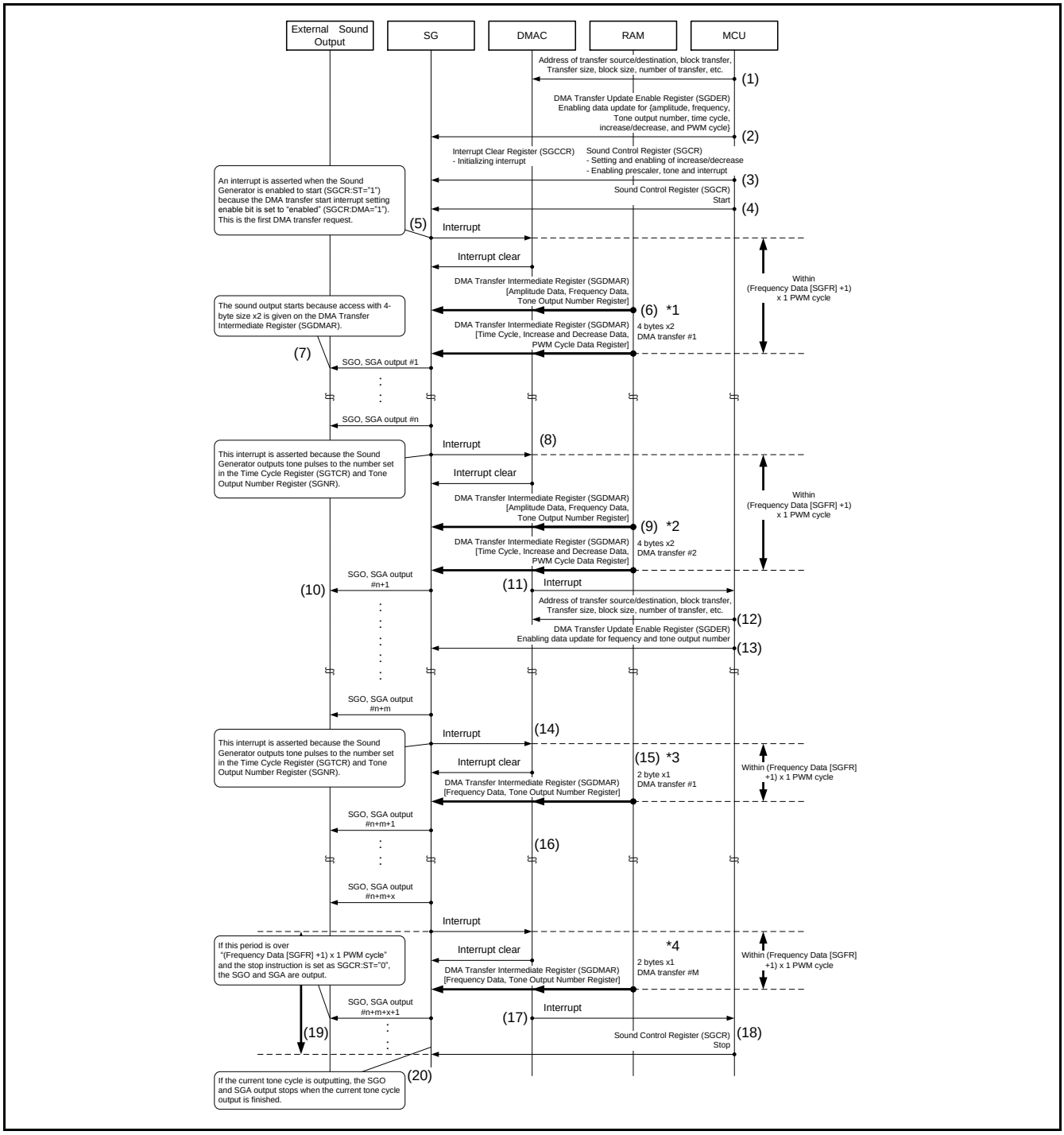
(17) If above (16) is not done within the limit time, the Sound Generator keeps driving SGO and SGA in order to output the data of the Nth DMA transfer. Then, the Sound Generator stops driving SGO and SGA after the end of all data.
(*4: The data of the Nth DMA transfer are written to the Sound Generator, and they are output to the end.)

Notes:

- *The DMAC must finish the sequence from step 6 to 7, within the following time.*
  The limit time = (Frequency Data Register [SGFR] + 1) x 1 PWM cycle
- *The DMA transfer error means the occurrence of delay in the sound data setting. It causes unsteady sound output. In that case, please fix the priority of DMA transfer in the system to finish all data transfer within the limit time.*

3.7.4. In Case of DMA Transfer with 4-Byte size x 2 is Made N Times and DMA Transfer with 2-Byte size x 1 is Made M Times (The Case the Transfer Byte size is Changed While Sound Outputting)

Figure 3-12 Sound Generator operation with DMA
(The case the transfer byte size is changed while sound is being output)



(1) Software makes initial settings to those registers which are needed in DMA transfer. The DMA transfer is based on a block data of "4-byte size x 2", and this block can be repeated N times. DMAC sets following registers to prepare for a DMA transfer, through the "DMA Transfer Intermediate Register (SGDMAR)".
- Amplitude Data Register (SGAR)
- Frequency Data Register (SGFR)
- Tone Output Number Register (SGNR)
- Time Cycle Register (SGTCR)
- Increase and Decrease Data Register (SGIDR)
- PWM Cycle Data Register (SGPCR)

The destination address of the DMA transfer is a fixed one on the "DMA Transfer Intermediate Register (SGDMAR)".

(2) Software configures the "DMA Transfer Update Enable Register (SGDER)" to enable the automatic update of the following registers during DMA transfer.
- Amplitude Data Register (SGAR)
- Frequency Data Register (SGFR)
- Tone Output Number Register (SGNR)
- Time Cycle Register (SGTCR)
- Increase and Decrease Data Register (SGIDR)
- PWM Cycle Data Register (SGPCR)

(3) Software initialize Interrupt status bit (SGCR.INT) by writing "1" to Interrupt status clear bit (SGCCR.INTC). Then, the software configures the Sound Control Register (SGCR) in the needed mode, and this must include the following bit operations.
- DMA transfer start interrupt setting enable bit (SGCR.DMA) to "1"(enabled)
- Interrupt enable bit (SGCR.INTE) to "1"(enabled)

(4) Write "1" to the Start bit (SGCR.ST).

(5) The interrupt occurs immediately after setting Start bit (SGCR.ST), since the Sound Generator is enabled on DMA transfer (SGCR.DMA="1"). An Interrupt request (PIRQ) is asserted, and this interrupt is used as a DMA transfer request.

(6) DMAC clears the interrupt, and write registers in the Sound Generator through the "DMA Transfer Intermediate Register (SGDMAR)". This operation configures the following registers by 2 steps.
[1st step]
- Amplitude Data Register (SGAR)
- Frequency Data Register (SGFR)
- Tone Output Number Register (SGNR)

[2nd step]
- Time Cycle Register (SGTCR)
- Increase and Decrease Data Register (SGIDR)
- PWM Cycle Data Register (SGPCR)

(*: DMA block size must to be "4-byte size x 2" for the access to "DMA Transfer Intermediate Register (SGDMAR)")

(7) The outputs of SGO and SGA start, according to the register settings above.

(8) The Tone pulse counter counts the number of tone pulses. When the following conditions are satisfied, the interrupt is generated.
- Tone pulse counter is 0x00
- Decrement counter is 0x00
- At the rising edge of SGO

(9) DMAC clears the interrupt, and write registers in the Sound Generator through the "DMA Transfer Intermediate Register (SGDMAR)".
(*2: DMA block size must to be "4-byte size x 2" for the access to "DMA Transfer Intermediate Register (SGDMAR)")

(10) The Sound Generator keeps outputting SGO and SGA, according to the register settings above.

(11) When DMAC completes all DMA transfer (with 4-byte size x2, N times), it asserts an interrupt to MCU.

(12) Software makes settings to those registers which are needed in DMA transfer. The DMA transfer is based on a block data of "2-byte size x 1", and this block can be repeated M times. DMAC sets following registers to prepare for a DMA transfer, through the "DMA Transfer Intermediate Register (SGDMAR)".
- Frequency Data Register (SGFR)
- Tone Output Number Register (SGNR)
The destination address of the DMA transfer is a fixed one on the "DMA Transfer Intermediate Register (SGDMAR)".

(13) Software configures the "DMA Transfer Update Enable Register (SGDER)" to enable the automatic update of the following registers during DMA transfer.
- Frequency Data Register (SGFR)
- Tone Output Number Register (SGNR)

(14) The Tone pulse counter counts the number of tone pulses. When the following conditions are satisfied, the interrupt is generated.
- Tone pulse counter is 0x00
- Decrement counter is 0x00
- At the rising edge of SGO

(15) DMAC clears the interrupt, and write registers in the Sound Generator through the "DMA Transfer Intermediate Register (SGDMAR)".
(*3: DMA block size must to be "2-byte size x 1" for the access to "DMA Transfer Intermediate Register (SGDMAR)")

(16) Repeat the flow from 14 to 15 to continue outputting the sound.

(17) When DMAC completes all DMA transfer (with 2-byte size x1, M times), it asserts an interrupt to MCU.

(18) Software writes "0" to the Start bit (SGCR.ST) to stop outputting the sound.

(19) When above operation (18) was made within the following time, the Mth DMA transfer doesn't come to the output of SGO and SGA. The Sound Generator stops driving SGO and SGA just before outputting the data of Nth DMA transfer.
The limit time = (Frequency Data Register [SGFR] + 1) x 1 PWM cycle
(*4: The data of the Mth DMA transfer are written to the Sound Generator, however, they are not output. DMAC issues this Mth DMA transfer only to assert an interrupt toward the MCU.)

(20) If above (19) is not done within the limit time, the Sound Generator keeps driving SGO and SGA in order to output the data of the Mth DMA transfer. Then, the Sound Generator stops driving SGO and SGA after the end of all data.
(*4: The data of the Mth DMA transfer are written to the Sound Generator, and they are output to the end.)

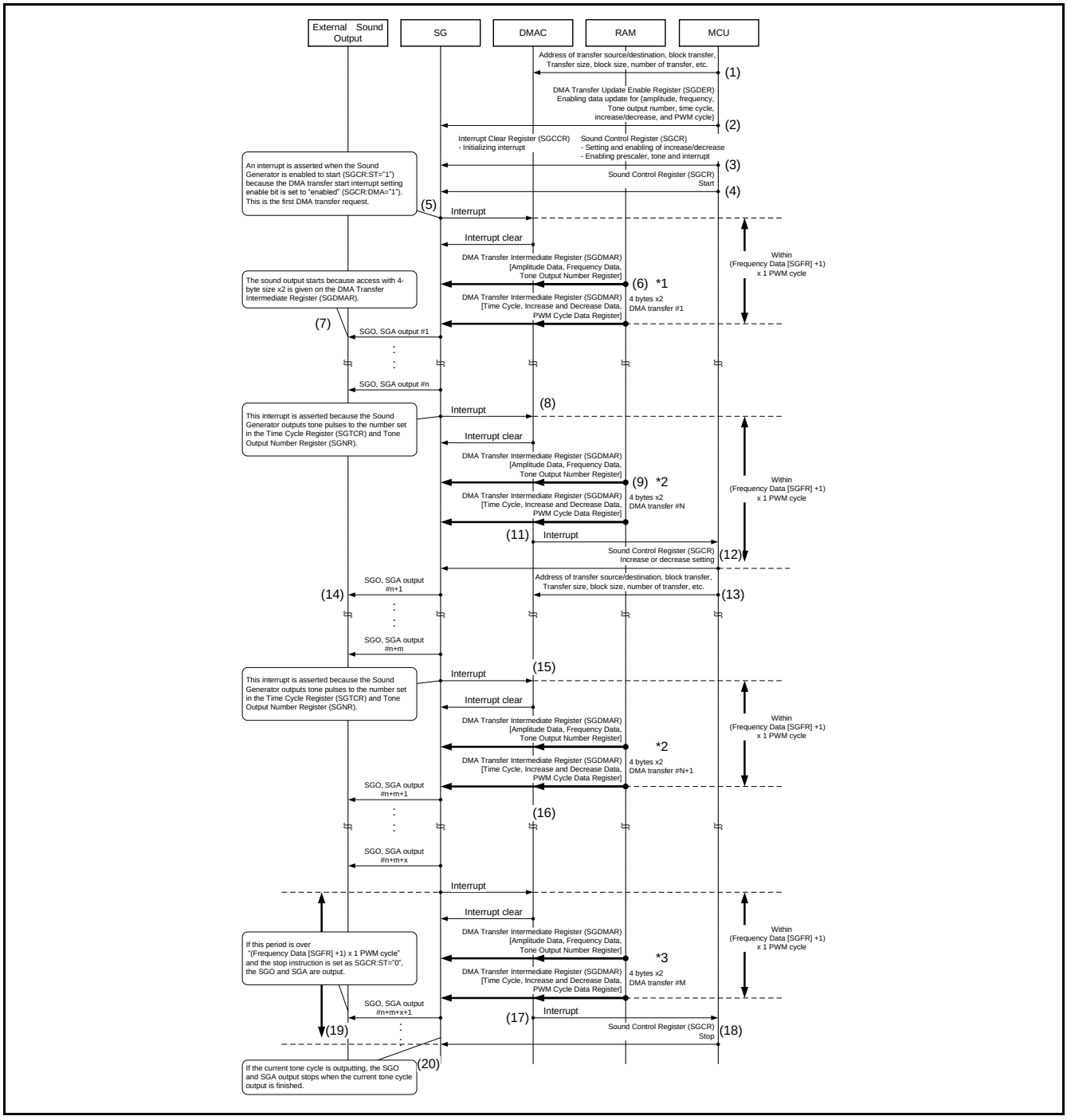
Notes:

- *The DMAC must finish the sequence from step 5 to 6, within the following time. The limit time = (Frequency Data Register [SGFR] + 1) x 1 PWM cycle*
- *The DMA transfer error means the occurrence of delay in the sound data setting. It causes unsteady sound output. In that case, please fix the priority of DMA transfer in the system to finish all data transfer within the limit time.*

3.7.5. In Case of DMA Transfer with 4-Byte size x 2 is made N Times and DMA Transfer with 4-Byte size x 2 is Made M Times (The Case the Transfer Byte size and the Increase/Decrease setting is Changed While Sound Outputting)

Figure 3-13 Sound Generator operation with DMA

(The case the transfer byte size and increase/decrease setting is changed while sound is being output)



(1) Software makes initial settings to those registers which are needed in DMA transfer. The DMA transfer is based on a block data of "4-byte size x 2", and this block can be repeated N times. DMAC sets following registers to prepare for a DMA transfer, through the "DMA Transfer Intermediate Register (SGDMAR)".
- Amplitude Data Register (SGAR)
- Frequency Data Register (SGFR)
- Tone Output Number Register (SGNR)
- Time Cycle Register (SGTCR)
- Increase and Decrease Data Register (SGIDR)
- PWM Cycle Data Register (SGPCR)

The destination address of the DMA transfer is a fixed one on the "DMA Transfer Intermediate Register (SGDMAR)".

(2) Software configures the "DMA Transfer Update Enable Register (SGDER)" to enable the automatic update of the following registers during DMA transfer.
- Amplitude Data Register (SGAR)
- Frequency Data Register (SGFR)
- Tone Output Number Register (SGNR)
- Time Cycle Register (SGTCR)
- Increase and Decrease Data Register (SGIDR)
- PWM Cycle Data Register (SGPCR)

(3) Software initialize Interrupt status bit (SGCR.INT) by writing "1" to Interrupt status clear bit (SGCCR.INTC). Then, the software configures the Sound Control Register (SGCR) in the needed mode, and this must include the following bit operations.
- DMA transfer start interrupt setting enable bit (SGCR.DMA) to "1"(enabled)
- Interrupt enable bit (SGCR.INTE) to "1"(enabled)

(4) Write "1" to the Start bit (SGCR.ST).

(5) The interrupt occurs immediately after setting Start bit (SGCR.ST), since the Sound Generator is enabled on DMA transfer (SGCR.DMA="1"). An Interrupt request (PIRQ) is asserted, and this interrupt is used as a DMA transfer request.

(6) DMAC clears the interrupt, and write registers in the Sound Generator through the "DMA Transfer Intermediate Register (SGDMAR)". This operation configures the following registers by 2 steps.
[1st step]
- Amplitude Data Register (SGAR)
- Frequency Data Register (SGFR)
- Tone Output Number Register (SGNR)

[2nd step]
- Time Cycle Register (SGTCR)
- Increase and Decrease Data Register (SGIDR)
- PWM Cycle Data Register (SGPCR)

(*: DMA block size must to be "4-byte size x 2" for the access to "DMA Transfer Intermediate Register (SGDMAR)")

(7) The outputs of SGO and SGA start, according to the register settings above.

(8) The Tone pulse counter counts the number of tone pulses. When the following conditions are satisfied, the interrupt is generated.
- Tone pulse counter is 0x00
- Decrement counter is 0x00
- At the rising edge of SGO

(9) DMAC clears the interrupt, and write registers in the Sound Generator through the "DMA Transfer Intermediate Register (SGDMAR)".
(*2: DMA block size must to be "4-byte size x 2" for the access to "DMA Transfer Intermediate Register (SGDMAR)")

(10) The Sound Generator keeps outputting SGO and SGA, according to the register settings above.

(11) When DMAC completes all DMA transfer (with 4-byte size x2, N times), it asserts an interrupt to MCU.

(12) Change the increase/decrease setting of the Sound Control Register (SGCR) by software (SGCR.GID).

(13) Software makes settings to the DMAC. The DMA transfer is based on a block data of "4-byte size x 2 cycle", and this block can be repeated M times.

(14) The Sound Generator keeps outputting SGO and SGA.

(15) The Tone pulse counter counts the number of tone pulses. When the following conditions are satisfied, the interrupt is generated.
- Tone pulse counter is 0x00
- Decrement counter is 0x00
- At the rising edge of SGO

Then, DMAC clears the interrupt, and write registers in the Sound Generator through the "DMA Transfer Intermediate Register (SGDMAR)".
(*2: DMA block size must to be "4-byte size x 2" for the access to "DMA Transfer Intermediate Register (SGDMAR)")

(16) Repeat the step 15 to continue outputting the sound.

(17) When DMAC completes all DMA transfer (with 4-byte size x2, M times), it asserts an interrupt to MCU.

(18) Software writes "0" to the Start bit (SGCR.ST) to stop outputting the sound.

(19) When above operation (18) was made within the following time, the Mth DMA transfer doesn't come to the output of SGO and SGA. The Sound Generator stops driving SGO and SGA just before outputting the data of Nth DMA transfer.
The limit time = (Frequency Data Register [SGFR] + 1) x 1 PWM cycle
(*4: The data of the Mth DMA transfer are written to the Sound Generator, however, they are not output. DMAC issues this Mth DMA transfer only to assert an interrupt toward the MCU.)

(20) If above (19) is not done within the limit time, the Sound Generator keeps driving SGO and SGA in order to output the data of the Mth DMA transfer. Then, the Sound Generator stops driving SGO and SGA after the end of all data.
(*4: The data of the Mth DMA transfer are written to the Sound Generator, and they are output to the end.)

Notes:

- *The DMAC must finish the sequence from step 5 to 6, within the following time.*
  The limit time = (Frequency Data Register [SGFR] + 1) x 1 PWM cycle
- *The DMA transfer error means the occurrence of delay in the sound data setting. It causes unsteady sound output. In that case, please fix the priority of DMA transfer in the system to finish all data transfer within the limit time.*
- *The DMAC must finish the sequence from step 8 to 12 (including the time for changing "the increase/decrease setting"), within the following time. The increase/decrease setting change comes into effect on the sound output from step 14, using the data of the Mth transfer with 4-byte size x 2.*
  The limit time = (Frequency Data Register [SGFR] + 1) x 1 PWM cycle

3.8. Interrupt of Sound Generator

The interrupt from Sound Generator comes from the judgment of counting tone pulses.

Interrupt of Sound Generator

The interrupt control bit and its factor is shown in the following table.

Table 3-3 The Interrupt bit of Sound Generator and Its Factor

	Interrupt bit of Sound Generator (singular)
Interrupt status bit	Register: Sound Control Register (SGCR) Bit: Interrupt status bit (INT: bit1)
Interrupt enable bit	Register: Sound Control Register (SGCR) Bit: Interrupt enable bit (INTE: bit2)
Interrupt factor	Tone pulse count is greater than or equal to (Time Cycle Register [SGTCR] + 1) x (Tone Output Number Register [SGNR] + 1)

During the tone pulses are counted, when the count reaches the multiple of "Time Cycle Register [SGTCR] + 1" and "Tone Output Number Register [SGNR] + 1", the Interrupt status bit (SGCR.INT) is set. When the interrupt is allowed by the setting of SGCR.INTE, an Interrupt request (PIRQ) is subsequently asserted (to be "H").

4. Registers

This chapter shows the list of registers in Sound Generator.

There is not the restriction by privileged processing mode, when these registers are read or written.

Registers of Sound Generator

Table 4-1 The List of Registers of Sound Generator

Contracted Name	Register Name	Reference
SGDER	DMA Transfer Update Enable Register	4.1
SGCR	Sound Control Register	4.2
SGAR	Amplitude Data Register	4.3
SGFR	Frequency Data Register	4.4
SGNR	Tone Output Number Register	4.5
SGTCR	Time Cycle Register	4.6
SGIDR	Increase and Decrease Data Register	4.7
SGPCR	PWM Cycle Data Register	4.8
SGDMAR	DMA Transfer Intermediate Register	4.9
SGCCR	Interrupt Clear Register	4.10

The Meaning of the Register Bit Property Code

Table 4-2 The Meaning of the Register Bit Property Code

R	Read-Only Register. Only Read access is available, and Write access is ignored.
W	Write-Only Register. Only Write access is available, and Read access makes nothing.
R/W	Read/Write Register. Both Read access and Write access are available.
-	Both Read access and Write access make nothing.

4.1. DMA Transfer Update Enable Register (SGDER)

The DMA Transfer Update Enable Register (SGDER) is to enable the update of the registers (SGAR, SGFR, SGNR, SGTCR, SGIDR, SGPCR) which can be updated when in DMA transfer.

This setting is made for respective registers.

The Sound Generator identifies the register to be updated by DMA transfer according to the setting of this register.

It also recognizes the number of times of DMA transfer and the transfer byte size according to the setting of this register, as well as the valid byte position of the DMA Transfer Intermediate Register (SGDMAR).

bit	7	6	5	4	3	2	1	0
Field	ARE1	ARE0	FRE	NRE	TCRE	IDRE	PCRE1	PCRE0
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] ARE1 : Amplitude data (Upper Byte) Update enable bit

This bit is to enable the update of the amplitude data (upper byte) of the Amplitude Data Register (SGAR) through the DMA Transfer Intermediate Register when in DMA transfer.

bit	Description
0	The update of the amplitude data (upper byte) is disabled.
1	The update of the amplitude data (upper byte) is enabled.

[bit6] ARE0 : Amplitude data (Lower Byte) Update enable bit

This bit is to enable the update of the amplitude data (lower byte) of the Amplitude Data Register (SGAR) through the DMA Transfer Intermediate Register when in DMA transfer.

bit	Description
0	The update of the amplitude data (lower byte) is disabled.
1	The update of the amplitude data (lower byte) is enabled.

[bit5] FRE : Frequency data Update enable bit

This bit is to enable the update of the frequency data of the Frequency Data Register (SGFR) through the DMA Transfer Intermediate Register when in DMA transfer.

bit	Description
0	The update of the frequency data is disabled.
1	The update of the frequency data is enabled.

[bit4] NRE : Tone output Number Update enable bit

This bit is to enable the update of the tone output number of the Tone Output Number Register (SGNR) through the DMA Transfer Intermediate Register when in DMA transfer.

bit	Description
0	The update of the tone output number is disabled.
1	The update of the tone output number is enabled.

[bit3] TCRE : Time cycle Update enable bit

This bit is to enable the update of the time cycle of the Time Cycle Register (SGTCR) through the DMA Transfer Intermediate Register when in DMA transfer.

bit	Description
0	The update of the time cycle is disabled.
1	The update of the time cycle is enabled.

[bit2] IDRE : Increase and Decrease data Update enable bit

This bit is to enable the update of the increase and decrease data of the Increase and Decrease Data Register (SGIDR) through the DMA Transfer Intermediate Register when in DMA transfer.

bit	Description
0	The update of the increase and decrease data is disabled.
1	The update of the increase and decrease data is enabled.

[bit1] PCRE1 : PWM cycle data (Upper Byte) Update enable bit

This bit is to enable the update of the PWM cycle data (upper byte) of the PWM Cycle Data Register (SGPCR) through the DMA Transfer Intermediate Register when in DMA transfer.

bit	Description
0	The update of the PWM cycle data (upper byte) is disabled.
1	The update of the PWM cycle data (upper byte) is enabled.

[bit0] PCRE0 : PWM cycle data (Lower Byte) Update enable bit

This bit is to enable the update of the PWM cycle data (lower byte) of the PWM Cycle Data Register (SGPCR) through the DMA Transfer Intermediate Register when in DMA transfer.

bit	Description
0	The update of the PWM cycle data (lower byte) is disabled.
1	The update of the PWM cycle data (lower byte) is enabled.

4.2. Sound Control Register (SGCR)

The Sound Control Register (SGCR) is to control interrupts and operation status of the Sound Generator.

bit	15	14	13	12	11	10	9	8
Field	Reserved	SRST	DMA	GID	GEN	Reserved	BUSY	Reserved
Attribute	-	W	R/W	R/W	R/W	-	R	-
Protection attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	S1	S0	TONE	SGOOE	SGAOE	INTE	INT	ST
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W
Protection attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

[bit15] Reserved : Reserved bit

Always write "0" to this bit. The read value is "0".

[bit14] SRST : Software Reset bit

This bit is used to issue a software reset in the Sound Generator.

When in the read access, the read value is always "0". Writing "0" to this bit has no effect.

bit	Description
	Write
0	No effect.
1	Writing "1" issues a software reset.

[bit13] DMA : DMA Transfer Start Interrupt setting enable bit

This bit is to enable DMA transfer start interrupt.

bit	Description
0	Writing "1" to the Start bit (ST) does not make the Interrupt status bit (INT) set.
1	Writing "1" to the Start bit (ST) makes the Interrupt status bit (INT) set.

Note:

- *Do not change this setting while operating (ST="1").*

[bit12] GID : Increase/Decrease setting bit

This bit is for automatic increase/decrease setting of the sound, working with the Time Cycle Register (SGTCR), the Increase and Decrease Data Register (SGIDR), and the automatic increase/decrease enable bit (GEN).

This bit determines whether the value of Amplitude Data Register (SGAR) increases or decreases.

bit	Description
0	The value stored in the Amplitude Data Register (SGAR) decreases.
1	The value stored in the Amplitude Data Register (SGAR) increases.

Notes:

- *This bit is valid only when the automatic increase/decrease enable bit is set to "enabled"(GEN="1").*
- *If this bit is changed during the operation, it is reflected at the time when the value of "Amplitude Data Register (SGAR)" is automatically updated.*

[bit11] GEN : Automatic Increase/Decrease enable bit

This bit is for automatic increase/decrease setting of the sound, working with the Time Cycle Register (SGTCR), the Increase and Decrease Data Register (SGIDR), and the Increase/decrease setting bit (GID).

bit	Description
0	The automatic increase or decrease is disabled.
1	The automatic increase or decrease is enabled.

Note:

- *When this bit turns from "enabled" to "disabled", the Amplitude Data Register (SGAR) holds the value at that time.*

[bit10] Reserved : Reserved bit

Always write "0" to this bit. The read value is "0".

[bit9] BUSY : Busy Status bit

This bit is to indicate whether the Sound Generator is in operation or not.

This bit is only for read access, and the write access to this bit has no effect.

bit	Description
	Read
0	Sound Generator is inactive. This bit is cleared under any of following conditions. - Software reset is set (SRST="1"). - ST bit is cleared by writing "0". And the operation is completed after 1 tone cycle (current tone cycle) has finished.
1	Sound Generator is active. This bit is set when ST bit is written "1".

[bit8] Reserved : Reserved bit

Always write "0" to this bit. The read value is "0".

[bit7:6] S1, S0 : Operation clock Select bits

The combination of these bits controls the internal clock division, to determine the bit rate on the output sound.

S1	S0	Description
0	0	The division ratio at the Prescaler is 1.
0	1	The division ratio at the Prescaler is 2.
1	0	The division ratio at the Prescaler is 4.
1	1	The division ratio at the Prescaler is 8.

[bit5] TONE : Tone output bit

This bit controls the type of SGO output signal.

bit	Description
0	SGO outputs the mixed signal of tone pulse and PWM pulse.
1	SGO outputs a simple square wave (tone pulse), that is generated by the internal toggle flip-flop.

[bit4] SGOOE : SGO Signal output enable bit

This bit controls whether to enable the output of SGO.

bit	Description
0	SGO output is disabled.
1	SGO output is enabled.

[bit3] SGAOE : SGA Signal output enable bit

This bit controls whether to enable the output of SGA.

bit	Description
0	SGA output is disabled.
1	SGA output is enabled.

[bit2] INTE : Interrupt enable bit

This bit enables the interrupt request of the Sound Generator.

bit	Description
0	The interrupt by INT bit is masked, and the status of INT bit is not reflected to the output on PIRQ.
1	The interrupt by INT bit is not masked, and the status of INT bit is reflected to the output on PIRQ.

[bit1] INT : Interrupt Status bit

This is an interrupt flag of the Sound Generator.

This bit is only for read access, and the write access to this bit has no effect.

bit	Description	
	Read	Write
0	Sound Generator hasn't detected the end of tone pulses. This bit is cleared under any of following conditions. - Software reset is set (SRST="1"). - DMA transfer - Interrupt Clear bit is set (SGCCR.INTC="1")	No effect.
1	Sound Generator has detected the end of tone pulses. This bit is set under the following condition. - The number of tone pulse count is greater than or equal to (Time Cycle Register [SGTCR] + 1) x (Tone Output Number Register [SGNR] + 1)	No effect.

[bit0] ST : Start bit

This bit is a start trigger of the Sound Generator.

bit	Description
0	Writing "0" to this bit stops Sound Generator.
1	Writing "1" to this bit starts Sound Generator.

Notes:

- *Whether the Sound Generator is stopped or not can be seen on the status bit BUSY.*
- *This bit is cleared also when a software reset is issued (SRST = "1").*

4.3. Amplitude Data Register (SGAR)

The Amplitude Data Register (SGAR) stores the reload value for a PWM pulse generator.

The register value represents sound amplitude and is reloaded into the PWM pulse generator at the end of every tone cycle.

bit	15	14	13	12	11	10	9	8
Field	SGAR[15:8]							
Attribute	R/W							
Protection attribute	-							
Initial value	0x00							

bit	7	6	5	4	3	2	1	0
Field	SGAR[7:0]							
Attribute	R/W							
Protection attribute	-							
Initial value	0x00							

[bit15:0] SGAR : Amplitude data bits

These bits store the reload value for the PWM pulse generator,
and the value is written either in the following ways.
(1)Software writes a value to this register.
(2)DMAC writes a value through DMA Transfer Intermediate Register (SGDMAR).

When in decreasing the value of this register, 0x0000 is the lower limit and the value doesn't roll over to 0xFFFF from it.

When in increasing the value of this register, 0xFFFF is the upper limit and the value doesn't roll over to 0x0000 from it.

In any case, the Sound Generator keeps operating until the Start bit (SGCR.ST) is cleared.

Notes:

- *The number of clock cycles in high width of 1 PWM is equal to "SGAR + 1".*
- *During operation, in the case of changing a register value, meet the following either conditions.*
  (1)Sound generator is in the stop state (SGCR.BUSY="0").
  (2)The changing comes between an interrupt occurrence and falling edge of the first tone pulse.
  (The limit time = (Frequency Data Register [SGFR] + 1) x 1 PWM cycle)
  When it meets neither condition, the sound output cannot guarantee the expected duty ratio.

4.4. Frequency Data Register (SGFR)

The Frequency Data Register (SGFR) stores the reload value for a Frequency counter.

bit	7	6	5	4	3	2	1	0
Field	SGFR[7:0]							
Attribute	R/W							
Protection attribute	-							
Initial value	0x00							

[bit7:0] SGFR : Frequency data bits

These bits store the reload value for the Frequency counter,
and the value is written either in the following ways.
(1)Software writes a value to this register.
(2)DMAC writes a value through DMA Transfer Intermediate Register (SGDMAR).

Notes:

- *The number of PWM pulses is equal to "SGFR + 1".*
- *During operation, in the case of changing a register value, meet the following either conditions.*
  (1)Sound generator is in the stop state (SGCR.BUSY="0").
  (2)The changing comes between an interrupt occurrence and falling edge of the first tone pulse.
  (The limit time = (Frequency Data Register [SGFR] + 1) x 1 PWM cycle)
  When it meets neither condition, the sound output cannot guarantee the expected duty ratio.

4.5. Tone Output Number Register (SGNR)

The Tone Output Number Register (SGNR) stores the reload value for a Tone pulse counter.

The Tone pulse counter accumulates the number of tone pulses (or the number of sound increment/decrement). When the accumulated value reaches the reload value, it sets the Interrupt status bit (SGCR.INT). This operation is to reduce the frequency of interrupts.

bit	7	6	5	4	3	2	1	0
Field	SGNR[7:0]							
Attribute	R/W							
Protection attribute	-							
Initial value	0x00							

[bit7:0] SGNR : Tone output Number bits

These bits store the reload value for the Tone pulse counter,
and the value is written either in the following ways.
(1)Software writes a value to this register.
(2)DMAC writes a value through DMA Transfer Intermediate Register (SGDMAR).

Notes:

- *The number of pulses is equal to "SGNR + 1".*
- *During operation, in the case of changing a register value, meet the following either conditions.*
  (1)Sound generator is in the stop state (SGCR.BUSY="0").
  (2)The changing comes between an interrupt occurrence and falling edge of the first tone pulse.
  (The limit time = (Frequency Data Register [SGFR] + 1) x 1 PWM cycle)
  When it meets neither condition, the sound output cannot guarantee the expected duty ratio.

4.6. Time Cycle Register (SGTCR)

The Time Cycle Register (SGTCR) stores the reload value for a Decrement counter.

bit	7	6	5	4	3	2	1	0
Field	SGTCR[7:0]							
Attribute	R/W							
Protection attribute	-							
Initial value	0x00							

[bit7:0] SGTCR : Time cycle bits

These bits store the reload value for the Decrement counter,
and the value is written either in the following ways.
(1)Software writes a value to this register.
(2)DMAC writes a value through DMA Transfer Intermediate Register (SGDMAR).

Notes:

- *The number of pulses is equal to "SGTCR + 1".*
- *When SGTCR is 0x00, the automatic increase or decrease operation on the Amplitude Data Register (SGAR) occurs at each tone cycle.*
- *During operation, in the case of changing a register value, meet the following either conditions.*
  (1)Sound generator is in the stop state (SGCR.BUSY="0").
  (2)The changing comes between an interrupt occurrence and falling edge of the first tone pulse.
  (The limit time = (Frequency Data Register [SGFR] + 1) x 1 PWM cycle)
  When it meets neither condition, the sound output cannot guarantee the expected duty ratio.

4.7. Increase and Decrease Data Register (SGIDR)

The Increase and Decrease Data Register (SGIDR) stores the increment or decrement for the Amplitude Data Register (SGIDR).

bit	7	6	5	4	3	2	1	0
Field	SGIDR[7:0]							
Attribute	R/W							
Protection attribute	-							
Initial value	0x00							

[bit7:0] SGIDR : Increase and Decrease data bits

These bits store the increment or decrement for the Amplitude Data Register (SGAR) , and the value is written either in the following ways.

(1)Software writes a value to this register.

(2)DMAC writes a value through DMA Transfer Intermediate Register (SGDMAR).

Note:

- *During operation, in the case of changing a register value, meet the following either conditions.*
  (1)Sound generator is in the stop state (SGCR.BUSY="0").
  (2)The changing comes between an interrupt occurrence and falling edge of the first tone pulse.
  (The limit time = (Frequency Data Register [SGFR] + 1) x 1 PWM cycle)
  When it meets neither condition, the sound output cannot guarantee the expected duty ratio.

4.8. PWM Cycle Data Register (SGPCR)

The PWM Cycle Data Register (SGPCR) stores the number of cycles in 1 PWM cycle.

bit	15	14	13	12	11	10	9	8
Field	SGPCR[15:8]							
Attribute	R/W							
Protection attribute	-							
Initial value	0x00							

bit	7	6	5	4	3	2	1	0
Field	SGPCR[7:0]							
Attribute	R/W							
Protection attribute	-							
Initial value	0xFF							

[bit15:0] SGPCR : PWM cycle Number data bits

These bits store the number of cycles in 1PWM cycle,
and the value is written either in the following ways.
(1)Software writes a value to this register.
(2)DMAC writes a value through DMA Transfer Intermediate Register (SGDMAR).

Notes:

- *The number of cycles in 1 PWM is equal to "SGPCR + 1".*
- *During operation, in the case of changing a register value, meet the following either conditions.*
  (1)Sound generator is in the stop state (SGCR.BUSY="0").
  (2)The changing comes between an interrupt occurrence and falling edge of the first tone pulse.
  (The limit time = (Frequency Data Register [SGFR] + 1) x 1 PWM cycle)
  When it meets neither condition, the sound output cannot guarantee the expected duty ratio.

4.9. DMA Transfer Intermediate Register (SGDMAR)

The DMA Transfer Intermediate Register (SGDMAR) is available in DMA transfer, and works as a window to access several registers. The details are described in "3.4 Relation between the DMA Transfer Update Enable Register (SGDER) and DMA Settings".

bit	31	30	29	28	27	26	25	24
Field	SGDMAR[31:24]							
Attribute	W							
Protection attribute	-							
Initial value	0x00							

bit	23	22	21	20	19	18	17	16
Field	SGDMAR[23:16]							
Attribute	W							
Protection attribute	-							
Initial value	0x00							

bit	15	14	13	12	11	10	9	8
Field	SGDMAR[15:8]							
Attribute	W							
Protection attribute	-							
Initial value	0x00							

bit	7	6	5	4	3	2	1	0
Field	SGDMAR[7:0]							
Attribute	W							
Protection attribute	-							
Initial value	0x00							

[bit31:0] SGDMAR : DMA Transfer data bits

This register is used to write the following registers:

- Amplitude Data Register (SGAR)
- Frequency Data Register (SGFR)
- Tone Output Number Register (SGNR)
- Time Cycle Register (SGTCR)
- Increase and Decrease Data Register (SGIDR)
- PWM Cycle Data Register (SGPCR)

The details are described in "3.4 Relation between the DMA Transfer Update Enable Register (SGDER) and DMA Settings".

The read value is always 0x00000000.

Notes:

- *If the number of DMA transfer is greater than "the number determined by the DMA Transfer Update Enable Register (SGDER)", the value of this register is updated.*
- *When in reading corresponding registers (not through this register), please make sure the setting of "DMA Transfer Update Enable Register (SGDER)" is available for the target registers.*
- *As for the enabled setting register of the DMA Transfer Update Enable Register (SGDER), access is enabled to both the DMA Transfer Intermediate Register (SGDMAR) and individual registers (SGAR, SGFR, SGNR, SGTCR, SGIDR, SGPCR).*
- *During operation, in the case of changing a register value, meet the following either conditions.*
  (1)Sound generator is in the stop state (SGCR.BUSY="0").
  (2)The changing comes between an interrupt occurrence and falling edge of the first tone pulse. (The limit time = (Frequency Data Register [SGFR] + 1) x 1 PWM cycle)
  When it meets neither condition, the sound output cannot guarantee the expected duty ratio.

4.10. Interrupt Clear Register (SGCCR)

The Interrupt Clear Register (SGCCR) clears the Interrupt status bit (SGCR.INT) in the Sound Control Register (SGCR).

bit	15	14	13	12	11	10	9	8
Field	Reserved							
Attribute	-							
Protection attribute	-							
Initial value	0x00							

bit	7	6	5	4	3	2	1	0
Field	Reserved						INTC	Reserved
Attribute	-						W	-
Protection attribute	-						-	-
Initial value	0x00						0	0

Note:

- *Use 16- or 32-bit data access for the SGCCR register.*

[bit15:2] Reserved : Reserved bits

Always write "0" to these bits. The read value is "0".

[bit1] INTC : Interrupt Status Clear bit

When this bit is set, Interrupt status bit (SGCR.INT) is cleared.

The read value is "0".

bit	Description
	Write
0	No effect.
1	Interrupt status bit (SGCR.INT) is cleared.

[bit0] Reserved : Reserved bit

Always write "0" to this bit. The read value is "0".

CHAPTER 59: Stepper Motor Controller

This chapter explains the functions and operations of the Stepper Motor Controller (SMC).

1. Overview
2. Configuration
3. Operation of the Stepper Motor Controller
4. Registers

SMC-T01P00R01L01-E1-GBM01

1. Overview

This section lists the main features for Stepper Motor Controller.

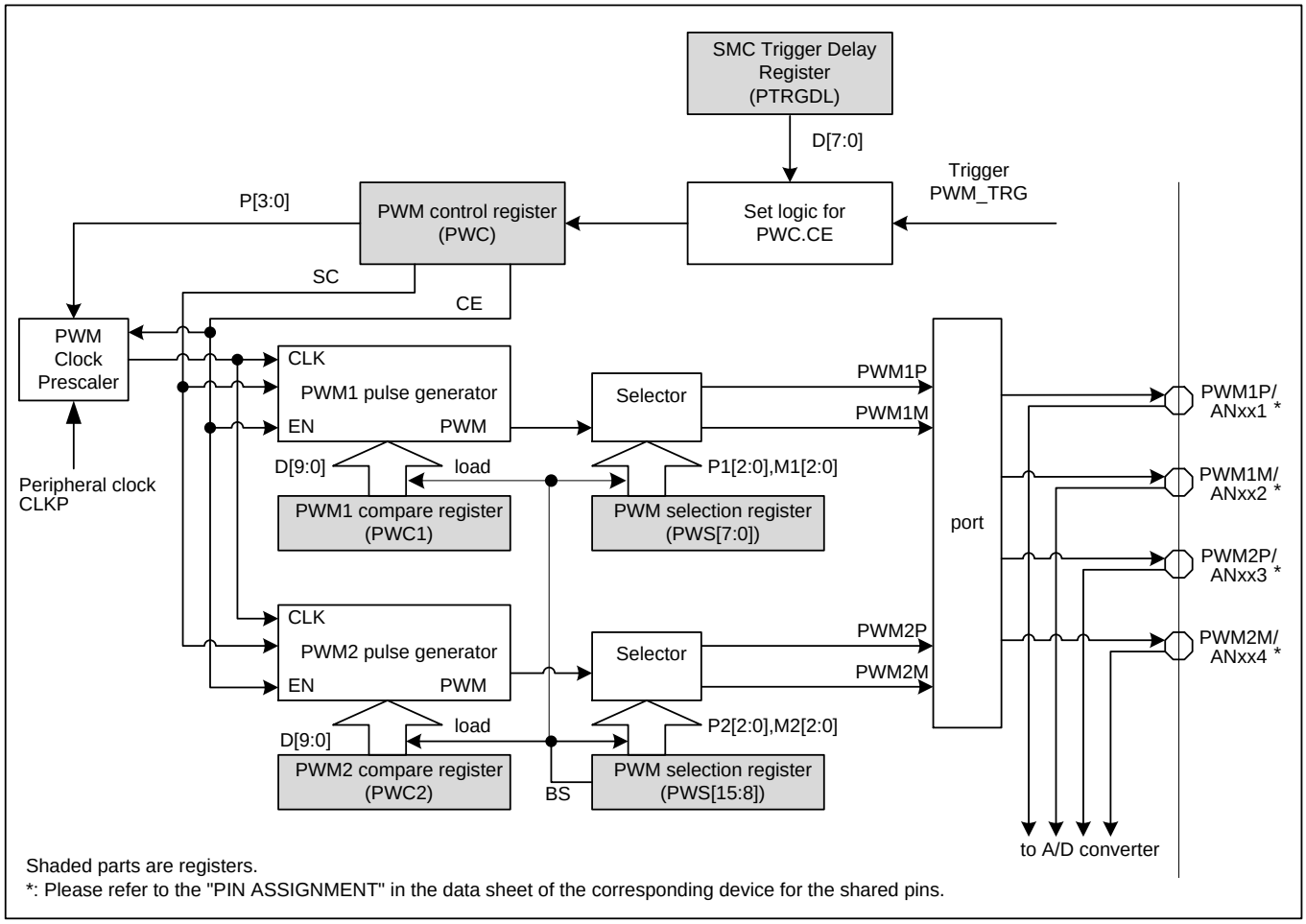
■ **Features of the Stepper Motor Controller:**

- Four high-current output drivers: Stepper Motor Controller provides four driver outputs with high current driving capability. Two motor coils can be connected directly to four stepper motor controller pins. The motor rotation is controlled by a combination of the PWM Pulse Generator and selector logic circuits.
- Two PWM Pulse Generators with 10-bit/8-bit operation mode (selected by software):
  - For the PWM pulse width in 8-bit operation mode, the duty can be specified within the range from 0% to 99.6%.
  - For the PWM pulse width in 10-bit operation mode, the duty can be specified within the range from 0% to 99.9%.
- A synchronization mechanism that enables synchronous operation of PWM Pulse Generators.
- A variable operating Clock Prescaler (the division ratio from a Peripheral clock can be selected by software).
  - Output selector logic ("H", "L", PWM pulse and "Hi-Z" signal levels can be selected).

2. Configuration

This section shows block diagrams of the Stepper Motor Controller.

Figure 59-1 Block Diagram of the Stepper Motor Controller



Shaded parts are registers.

*: Please refer to the "PIN ASSIGNMENT" in the data sheet of the corresponding device for the shared pins.

PWM Clock Prescaler
The PWM Clock Prescaler divides the Peripheral clock (CLKP) and generates the PWM operating clock. The PWM operating clock is used by the PWM Pulse Generators.

PWM1 Pulse Generator / PWM2 Pulse Generator
The PWM Pulse Generators generate the pulse-width modulated signals according to the waveform configuration and the selected operation mode.

Selector
This sub-module, based on configuration, selects the level of the signal that will be sent to the Stepper Motor Controller output pins.

Port
The Port controls the I/O port for the Stepper Motor Controller pins.

PWM Control Register (PWC)
The PWM Control Register (PWC) starts/stops the Stepper Motor Controller, sets the PWM Prescaler and selects the operation mode for the PWM Pulse Generators.

PWM1 and PWM2 Compare Registers (PWC1, PWC2)
The PWM1 and PWM2 Compare Registers determine the PWM pulse width.

PWM Selection Register (PWS)
The PWM Selection Register selects the signal levels ("H" and "L", PWM pulse and "Hi-Z") to be output to the Stepper Motor Controller pins. It also sets the update instruction for the system to synchronize the PWM Pulse Generators.

SMC Trigger Delay Register (PTRGDL)
The SMC Trigger Delay Register (PTRGDL) is used to delay the start of the operation by 0 to 255 clock cycles.

Set logic for PWC.CE
The Set logic for PWC.CE receives the trigger and waits for a configured delay time and sets automatically PWC.CE bit.

3. Operation of the Stepper Motor Controller

This section describes the operation of the Stepper Motor Controller.

3.1 Operation of PWM Pulse Generator

3.2 Selection of Motor Drive Signals

3.3 Synchronization System of Stepper Motor Controller

3.4 Operation of the SMC Trigger Delay Function

3.1. Operation of PWM Pulse Generator

The PWM period is determined by the selected division ratio from a Peripheral clock (CLKP) and the operation mode (8-bit/10-bit) of the PWM Pulse Generator:

- During configuration, software can select the PWM operating clock by specifying the division ratio for the Peripheral clock (CLKP). The division ratio for the Peripheral clock is defined by setting the PWM operating clock prescaler bits (PWC.P[3:0]).
- Software can also configure the operation mode for the counter that controls the PWM period. When the Operation Mode Switching bit (PWC.SC) is set to "0", 8-bit operation mode is selected (PWM period = 256 counts). If this bit is set to "1", 10-bit operation mode is selected (PWM period = 1024 counts).

Examples of operating clock values for different configurations are given in Table 59-1.

Table 59-1 Relation between PWM Operating Clock and PWM Period for Different Operation Modes

PWC P[3:0]	PWM Operating Clock (CLK)	PWM Period [µs] (CLKP = 16 MHz)		PWM Period [µs] (CLKP = 60 MHz)	
		PWC SC = "0"	PWC SC = "1"	PWC SC = "0"	PWC SC = "1"
0000	CLKP	16	64	4.267	17.067
0001	CLKP/4	64	256	17.067	68.267
0010	CLKP/5	80	320	21.333	85.333
0011	CLKP/6	96	384	25.6	102.4
0100	CLKP/8	128	512	34.133	136.533
0101	CLKP/10	160	640	42.667	170.667
0110	CLKP/12	192	768	51.2	204.8
0111	CLKP/16	256	1024	68.267	273.067
1000	CLKP/2	32	128	8.533	34.133
1001	CLKP/8	128	512	34.133	136.533
1010	CLKP/10	160	640	42.667	170.667
1011	CLKP/12	192	768	51.2	204.8
1100	CLKP/16	256	1024	68.267	273.067
1101	CLKP/20	320	1280	85.333	341.333
1110	CLKP/24	384	1536	102.4	409.6
1111	CLKP/32	512	2048	136.533	546.133

CLKP: Peripheral clock

The PWM pulse width (duty cycle) is determined by the PWM Compare Data Value (PWC1.D[9:0], PWC2.D[9:0]). In 8-bit operation mode (PWC.SC = "0"), the PWM Compare Data Value is taken from the PWC1.D[7:0], PWC2.D[7:0] bits and PWC1.D[9:8], PWC2.D[9:8] bits are "don't care". In 10-bit operation mode (PWC.SC = "1"), the PWM Compare Data Value is taken from the PWC1.D[9:0], PWC2.D[9:0] bits. Below are the ranges in which PWM Compare Data Values (PWC1.D[9:0], PWC2.D[9:0]) can be set:

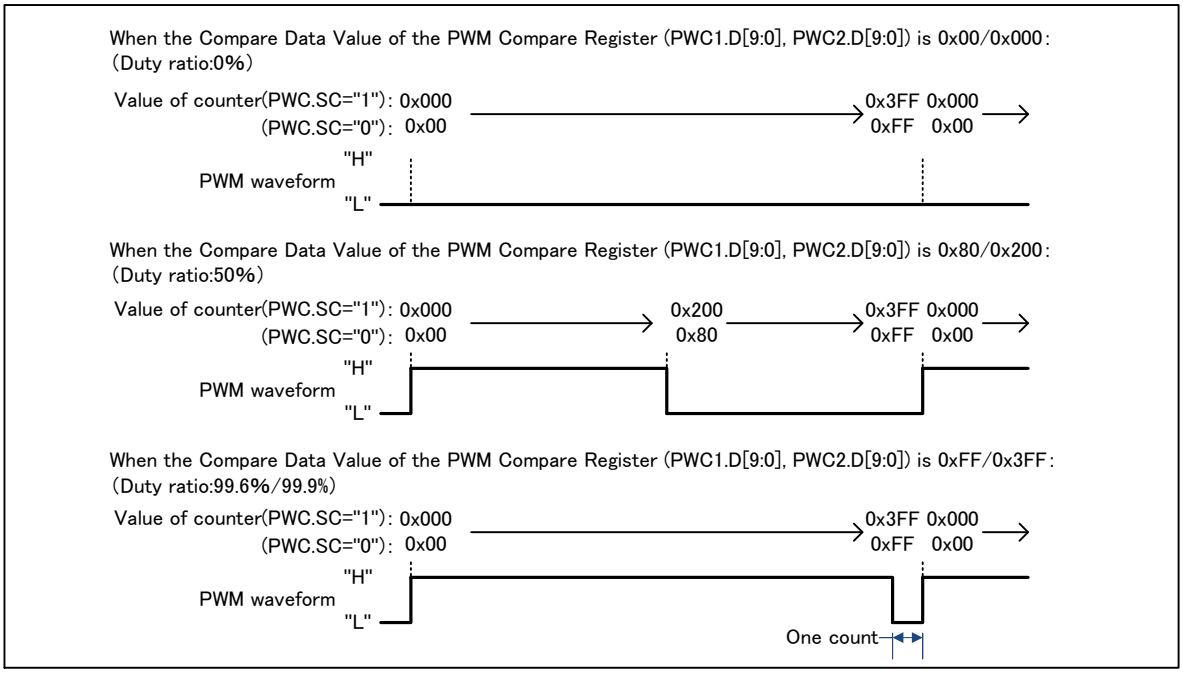
- In 8-bit operation mode (PWC.SC = "0") 0% to 99.6% (0x00 to 0xFF)
- In 10-bit operation mode (PWC.SC = "1") 0% to 99.9% (0x000 to 0x3FF)

The PWM Pulse Generator is started/stopped using the Count Enable bit (PWC.CE). When this bit is set to "1", the PWM Pulse Generator starts the counter that controls the PWM period. The counter

increments its value from 0x00 (0x000) at the rising edge of the PWM operating clock. The PWM Pulse Generator output remains at the "H" level until the counter value matches the PWM Compare Data Value. Once the PWM Compare Data Value is reached, the PWM Pulse Generator output changes to "L" and remains at the "L" level until the counter value overflows: 0xFF → 0x00 (0x3FF → 0x000).

Figure 59-2 shows the PWM waveforms generated by the PWM Pulse Generator.

Figure 59-2 Examples of PWM1 and PWM2 Output Waveforms



If the PWM Pulse Generator is stopped (PWC.CE = "0"), the setting of the PWM pulse width (duty cycle) and the output pin selection settings are not reflected on the Stepper Motor Controller outputs.

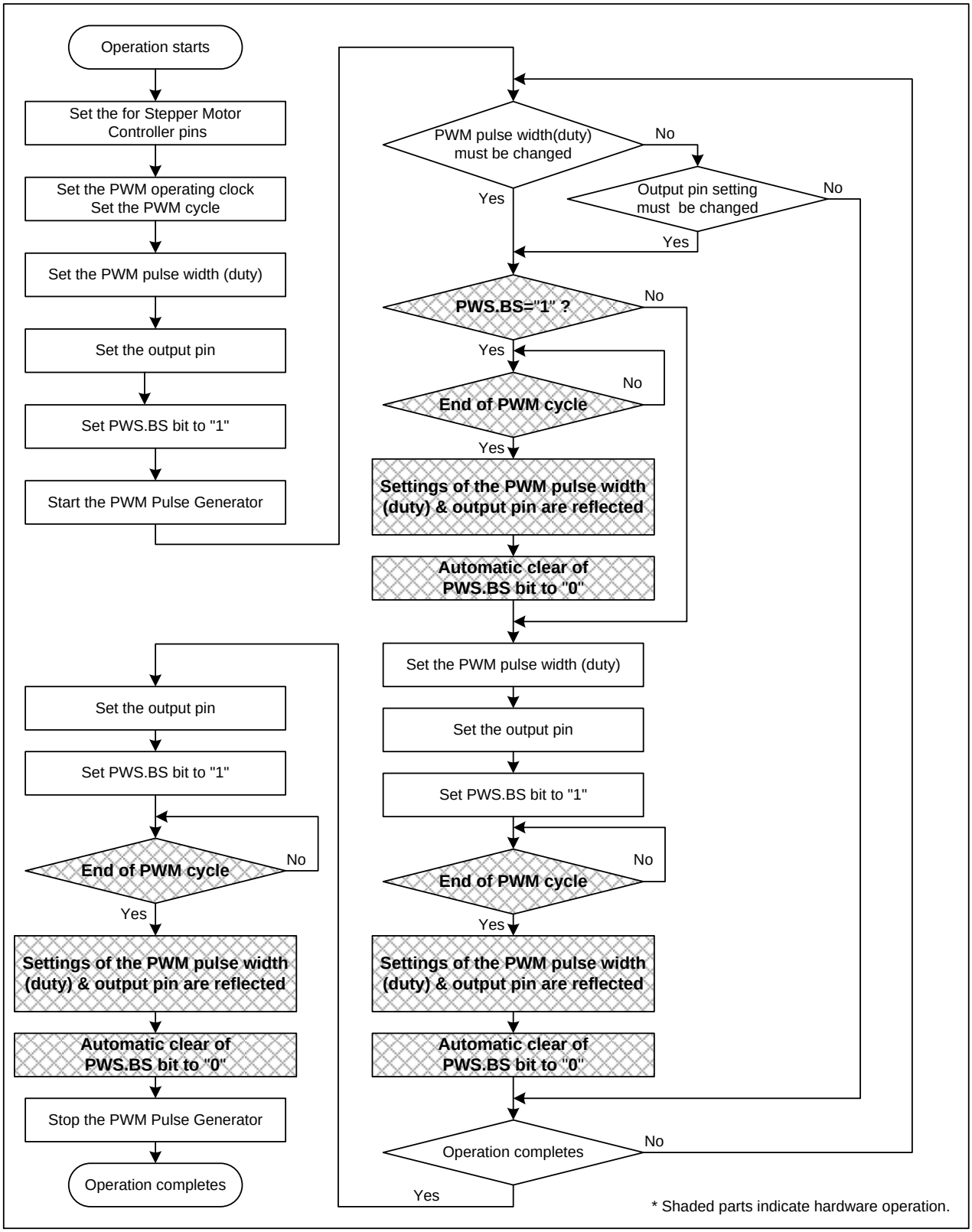
Notes:

- *If the PWM Pulse Generator is started(PWC.CE="1"), operate following steps because the PWM Pulse Generator.Compare Data Value must be initialized.*
  1. Set the PWM Compare Register.Compare Data Value(PWC1.D[9:0], PWC2.D[9:0])
  2. Set the PWM Selection Register.Output Selection bits(PWS.BS="1", PWS.P2[2:0], PWS.M2[2:0], PWS.P1[2:0], PWS.M1[2:0])
  3. Set the PWM Control Register.Count Enable bit(PWC.CE="1")
- *If the PWM Pulse Generator is stopped when the Output Update bit (PWS.BS) is "1", the PWS.BS bit will retain the value "1" until the PWM Pulse Generator is started and the beginning of the new PWM cycle is detected. As a result, the setting of the PWM pulse width (duty cycle) and the output pin selection settings are reflected on the Stepper Motor Controller outputs. To avoid this scenario, clear PWC.CE only after PWS.BS is automatically cleared by the SMC.*
- *If the PWS.BS bit is "1", keep the following registers value because the setting of the PWM pulse width (duty cycle) and the output pin selection settings are not reflected at the same time.*
  - PWM1 Compare Register.Compare Data Value(PWC1.D[9:0])
  - PWM2 Compare Register.Compare Data Value(PWC2.D[9:0])

- PWM Selection Register.Output Selection bits(PWS.P2[2:0], PWS.M2[2:0], PWS.P1[2:0], PWS.M1[2:0])

An example for the Stepper Motor Controller setting procedure is given on the Figure 59-3 .

Figure 59-3 Example of the Setting Procedure for the Stepper Motor Controller



3.2. Selection of Motor Drive Signals

The user can select the signal at each output pin of the Stepper Motor Controller according to the Table 59-2 and Table 59-3. The selection of output pin is done by configuring the PWM Selection Register (PWS.P2[2:0], PWS.M2[2:0], PWS.P1[2:0], PWS.M1[2:0]).

Table 59-2 Selection of Motor Drive Signals and Setting of PWM1 Selection Bits

PWS.P1[2:0]	PWM1P
000	L
001	H
01X	PWM pulse
1XX	Hi-Z

PWS.M1[2:0]	PWM1M
000	L
001	H
01X	PWM pulse
1XX	Hi-Z

X: don’t care

Table 59-3 Selection of Motor Drive Signals and Setting of PWM2 Selection Bits

PWS.P2[2:0]	PWM2P
000	L
001	H
01X	PWM pulse
1XX	Hi-Z

PWS.M2[2:0]	PWM2M
000	L
001	H
01X	PWM pulse
1XX	Hi-Z

X: don’t care

3.3. Synchronization System of Stepper Motor Controller

The Stepper Motor Controller contains a dedicated system developed to enable synchronization of PWM1 and PWM2 Pulse Generators, as well as Stepper Motor Controller output pin selection. The synchronization system is controlled by the PWM Selection Register Output Update bit (PWS.BS). Until the Output Update bit (PWS.BS) is set to "1", changes made to the PWM Compare Registers (PWC1, PWC2) and the PWM Selection Register (PWS) are not reflected on the output. Configuring the Stepper Motor Controller output pin selection and setting the Output Update bit (PWS.BS) simultaneously (in the same software access) is allowed.

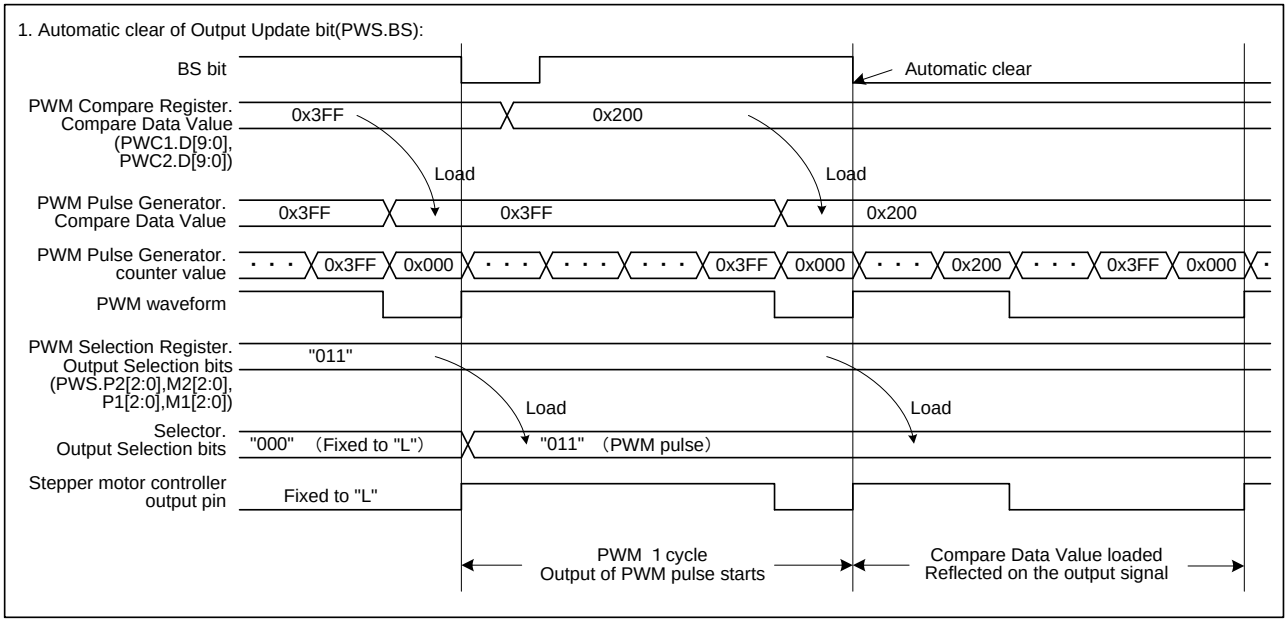
When the Output Update bit (PWS.BS) is set to "1" by software, and the end of current PWM cycle is detected, PWM Pulse Generator and motor drive selection logic will load values of the corresponding registers. The Output Update bit (PWS.BS) is automatically cleared to "0" at the beginning of the next PWM cycle.

If the Output Update bit (PWS.BS) is equal to "1" and the software access is initiated to set this bit to "1" at the beginning of the PWM cycle (at the same time when an automatic clear of Output Update bit (PWS.BS) is issued), the Output Update bit (PWS.BS) will retain the value "1" (i.e. no change is made to the PWS.BS bit) and the automatic clearing is canceled for this PWM cycle.

The following scenario is shown on Figure 59-4:

1. Automatic clear of the Output Update bit (PWS.BS): load operation is executed and reflected on the output.

Figure 59-4 Load Timing of PWM Compare Register and PWM Selection Register



3.4. Operation of the SMC Trigger Delay Function

After the Stepper Motor Controller receives a trigger input (PWM_TRG), it waits for a configured delay time to elapse before the Count Enable bit (PWC.CE) is automatically set and the Stepper Motor Controller starts operating. The delay time is configured in the SMC Trigger Delay Register (PTRGDL). The Stepper Motor Controller will continue to operate until the Count Enable bit (PWC.CE) is reset by software.

Figure 59-5 demonstrates the software flow for receiving a trigger input (PWM_TRG). An example, illustrating how the trigger starts Stepper Motor Controller, is shown on Figure 59-6.

Note:

- *The trigger should only be received when the Stepper Motor Controller are not running (i.e. the Count Enable bit (PWC.CE) is cleared to "0").*

Figure 59-5 Flowchart for Triggering Stepper Motor Controller

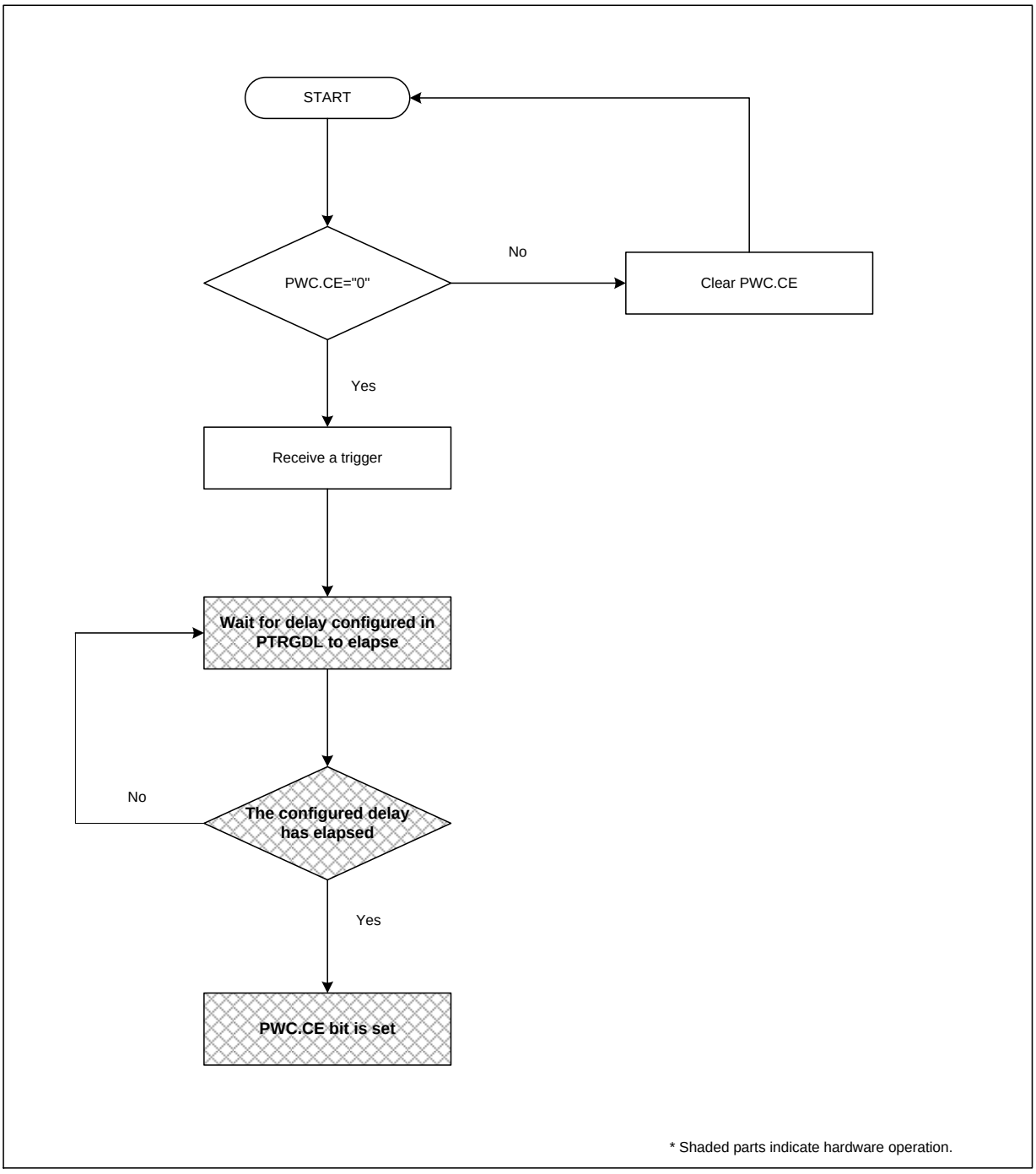


Figure 59-6 Timing diagram for Triggering of Stepper Motor Controller

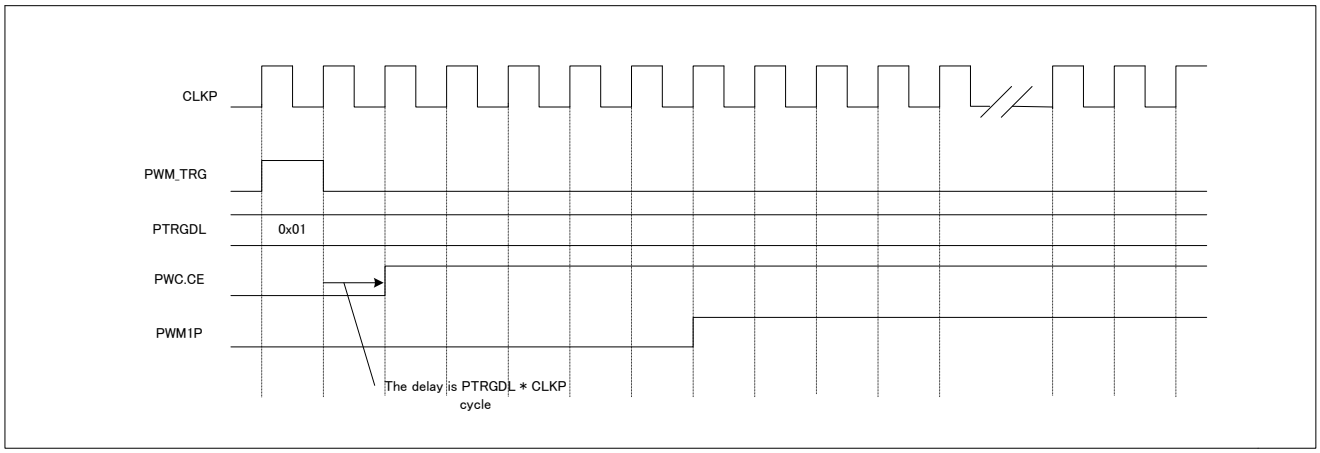


Figure 59-6 shows how the trigger starts Stepper Motor Controller (PTRGDL=0x01). If the SMC Trigger Delay Register (PTRGDL) setting is 0x00, PWC.CE is set immediately after receiving the trigger input (PWM_TRG). If the SMC Trigger Delay Register (PTRGDL) setting is 0x01, PWC.CE is set with a delay of one clock cycle after receiving the trigger input (PWM_TRG).

4. Registers

This section describes the registers of the Stepper Motor Controller.

There is not the restriction by privileged processing mode, when these registers are read or written.

■ The following registers are available for one Stepper Motor Controller:

PWM Control Register (PWC)

PWM1 and PWM2 Compare Registers (PWC1, PWC2)

PWM Selection Register (PWS)

PWM Selection Set Register (PWSS)

SMC Trigger Delay Register (PTRGDL)

Table 59-4 List of registers for the Stepper Motor Controller

Abbreviation	Register Name	Reference
PWC	PWM Control Register	4.1
PWC1	PWM1 Compare Register	4.2.1
PWC2	PWM2 Compare Register	4.2.2
PWS	PWM Selection Register	4.3
PWSS	PWM Selection Set Register	4.4
PTRGDL	SMC Trigger Delay Register	4.5

4.1. PWM Control Register (PWC)

The PWM Control Register starts/stops the Stepper Motor Controller, sets the PWM prescaler and selects the operation mode for the generation of the PWM pulse.

bit	15	14	13	12	11	10	9	8
Field	Reserved						P[3]	Reserved
Attribute	-						R/W	-
Protection attribute	-						-	-
Initial value	000000						0	0

bit	7	6	5	4	3	2	1	0
Field	Reserved	P[2:0]			CE	SC	Reserved	
Attribute	-	R/W			R/W	R/W	-	
Protection attribute	-	-			-	-	-	
Initial value	0	0			0	0	00	

[bit15:10] Reserved: Reserved Bits

When writing, always write "0".
When reading, "0" is always read.

[bit9] P[3]: PWM Operating Clock Prescaler Bit

P[3] bit, in combination with P[2:0] bits, specifies the clock input signal to the PWM Pulse Generators.

[bit8:7] Reserved: Reserved Bits

When writing, always write "0".
When reading, "0" is always read.

[bit6:4] P[2:0]: PWM Operating Clock Prescaler Bits

P[2:0] bits, in combination with P[3] bit, specify the clock input signal to the PWM Pulse Generators.

P[3]	P[2:0]	Description
0	000	PWM Operating clock is equal to the Peripheral clock (CLKP)
0	001	PWM Operating clock is equal to CLKP/4
0	010	PWM Operating clock is equal to CLKP/5
0	011	PWM Operating clock is equal to CLKP/6
0	100	PWM Operating clock is equal to CLKP/8
0	101	PWM Operating clock is equal to CLKP/10
0	110	PWM Operating clock is equal to CLKP/12
0	111	PWM Operating clock is equal to CLKP/16
1	000	PWM Operating clock is equal to CLKP/2
1	001	PWM Operating clock is equal to CLKP/8
1	010	PWM Operating clock is equal to CLKP/10
1	011	PWM Operating clock is equal to CLKP/12
1	100	PWM Operating clock is equal to CLKP/16
1	101	PWM Operating clock is equal to CLKP/20
1	110	PWM Operating clock is equal to CLKP/24
1	111	PWM Operating clock is equal to CLKP/32

CLKP: Peripheral clock

Note:

- *Configuration of the PWM Operating Clock Prescaler bits (P[3:0]) shall be done while counting operation is disabled (PWC.CE = "0").*

[bit3] CE: Count Enable Bit

The CE bit enables operation of the PWM Pulse Generators. When CE bit is set to "1", the PWM Pulse Generator starts its operating. The PWM2 Pulse Generator starts with the delay of one Peripheral clock cycle (CLKP) after the PWM1 Pulse Generator is started.

The CE bit can be set and cleared by the software. This bit also can be set to "1" by receiving trigger. To receive trigger, the CE bit must first be cleared to "0".

Bit	Description
0	Initializes and stops the operation of the PWM Pulse Generators
1	Starts the operation of the PWM Pulse Generators

Notes:

- *If the PWM Pulse Generator is started(CE="1"), operate following steps because the PWM Pulse Generator.Compare Data Value must be initialized.*
  1. Set the PWM Compare Register.Compare Data Value(PWC1.D[9:0], PWC2.D[9:0])
  2. Set the PWM Selection Register.Output Selection bits(PWS.BS="1", P2[2:0], M2[2:0], P1[2:0], M1[2:0])
  3. Set the PWM Control Register.Count Enable bit(CE="1")
- *If the PWM Pulse Generator is stopped (CE="0") when the PWM pulse is selected by the Output Selection bits (PWS.P2[2:0], M2[2:0], P1[2:0], M1[2:0]), the pin for which the PWM pulse is selected is fixed to the "L" level.*
- *If the trigger input is used, write "0" to the CE bit only when the CE bit is "1". Because the CE bit can be cleared at the same time when the CE bit is set by the trigger.*
- *The CE bit shall be set to "1" after the setting of the PWM Operating Clock Prescaler bits (PWC.P[3:0]) and the Operation Mode Switching bit (PWC.SC) is completed.*

[bit2] SC: Operation Mode Switching Bit

This bit specifies the operation mode of the PWM Pulse Generators

Bit	Description
0	8-bit operation mode is selected
1	10-bit operation mode is selected

Note:

- *The PWC.CE bit shall be set to "1" after the setting of the PWM Operating Clock Prescaler bits (PWC.P[3:0]) and the Operation Mode Switching bit (SC) is completed.*

[bit1:0] Reserved: Reserved bits

When writing, always write "0".
When reading, "0" is always read.

4.2. PWM1 and PWM2 Compare Registers (PWC1, PWC2)

The value of the 8/10 bit PWM1 and PWM2 Compare Registers determines the width of the PWM pulse. A value of 0x00 (0x000) indicates that the PWM duty cycle is 0%, while a value of 0xFF (0x3FF) indicates that the PWM duty cycle is 99.6% (99.9%).

4.2.1. PWM1 Compare Register(PWC1)

bit	15	14	13	12	11	10	9	8
Field	Reserved						D[9:8]	
Attribute	-						R/W	
Protection attribute	-						-	
Initial value	000000						XX	

bit	7	6	5	4	3	2	1	0
Field	D[7:0]							
Attribute	R/W							
Protection attribute	-							
Initial value	XXXXXXXX							

[bit15:10] Reserved: Reserved Bits

When writing, always write "0".

When reading, "0" is always read.

[bit9:0] D[9:0]: PWM1 Compare Data Value

The PWM1 Compare Data Value D[9:0] are used to set the PWM1 duty cycle. When the Operation Mode Switching bit is set to 8-bit operation mode (PWC.SC= "0"), the compare data value is taken from D[7:0] bits and D[9:8] bits are "don't care". When the Operation Mode Switching bit is set to 10-bit operation mode (PWC.SC= "1"), the compare data value is taken from D[9:0] bits.

D[9:0]	Description
0x(0)00	0%
…	Set the PWM pulse width
0x(3)FF	(99.9%) 99.6%

Notes:

- *PWM Pulse Generator.Compare Data Value is updated at the end of PWM cycle, after PWS.BS bit is set to "1".*
- *When PWC.SC bit is set to "0" (PWM Pulse Generators are operating in 8-bit mode), D[9:8], PWC2.D[9:8] bits are "don't care".*
- *The software must always make a 16-bit write access to PWM1 and PWM2 Compare Registers (PWC1, PWC2) to ensure consistency of data. However, 8-bit write access to PWM1 or PWM2 Compare Register (PWC1, PWC2) is possible and will not result in error response.*
- *If the PWS.BS bit is "1" , don't change the PWM Compare Register.Compare Data Value(D[9:0], PWC2.D[9:0]) and the PWM Selection Register.Output Selection bits (PWS.P2[2:0], PWS.M2[2:0], PWS.P1[2:0], PWS.M1[2:0]).*

4.2.2. PWM2 Compare Register(PWC2)

bit	15	14	13	12	11	10	9	8
Field	Reserved						D[9:8]	
Attribute	-						R/W	
Protection attribute	-						-	
Initial value	000000						XX	

bit	7	6	5	4	3	2	1	0
Field	D[7:0]							
Attribute	R/W							
Protection attribute	-							
Initial value	XXXXXXXX							

[bit15:10] Reserved: Reserved Bits

When writing, always write "0".

When reading, "0" is always read.

[bit9:0] D[9:0]: PWM2 Compare Data Value

The PWM2 Compare Data Value D[9:0] bits are used to set the PWM2 duty cycle. When the Operation Mode Switching bit is set to 8-bit operation mode (PWC.SC= "0"), the compare data value is taken from D[7:0] bits and D[9:8] bits are "don't care". When the Operation Mode Switching bit is set to 10-bit operation mode (PWC.SC= "1"), the compare data value is taken from D[9:0] bits.

D[9:0]	Description
0x(0)00	0%
…	Set the PWM pulse width
0x(3)FF	(99.9%) 99.6%

Notes:

- *PWM Pulse Generator.Compare Data Value is updated at the end of PWM cycle, after PWS.BS bit is set to "1".*
- *When PWC.SC bit is set to "0" (PWM Pulse Generators are operating in 8-bit mode), PWC1.D[9:8], D[9:8] bits are "don't care".*
- *The software must always make a 16-bit write access to PWM1 and PWM2 Compare Registers (PWC1, PWC2) to ensure consistency of data. However, 8-bit write access to PWM1 or PWM2 Compare Register (PWC1, PWC2) is possible and will not result in error response.*
- *If the PWS.BS bit is "1" , don't change the PWM Compare Register.Compare Data Value(PWC1.D[9:0], D[9:0]) and the PWM Selection Register.Output Selection bits (PWS.P2[2:0], PWS.M2[2:0], PWS.P1[2:0], PWS.M1[2:0]).*

4.3. PWM Selection Register (PWS)

The PWM Selection Register (PWS) sets the output state of the Stepper Motor Controller pins ("L", "H", PWM pulse or "Hi-Z") and allows the update of the PWM duty cycle or output pin states.

bit	15	14	13	12	11	10	9	8
Field	Reserved	BS	P2[2:0]			M2[2:0]		
Attribute	-	R/W	R/W			R/W		
Protection attribute	-	-	-			-		
Initial value	0	0	000			000		

bit	7	6	5	4	3	2	1	0
Field	Reserved		P1[2:0]			M1[2:0]		
Attribute	-		R/W			R/W		
Protection attribute	-		-			-		
Initial value	00		000			000		

[bit15] Reserved: Reserved Bit

When writing, always write "0".
When reading, "0" is always read.

[bit14] BS: Output Update Bit

The Output Update bit (BS) synchronously updates the Selector.Output Selection bits and PWM Pulse Generator.Compare Data Value (See Figure3-3). To change the settings for the PWM duty cycle or output pin states, the BS bit must be set to "1" after (or at the same time) a new configuration is written to the PWM1 and PWM2 Compare Registers (PWC1, PWC2) and PWM Selection Register (PWS). If the BS bit is set to "1" while PWM Pulse Generators are operating, the new settings are loaded at the end of the current PWM cycle. The BS bit is cleared automatically at the beginning of the following PWM cycle. As long as BS bit is "0", the PWM duty cycle and output pin states are not changed.

This bit can also be set by the PWSS.BSS bit.

Bit	Description
0	Disables the update of the setting for the PWM output
1	Enables the update of the setting for the PWM output

Notes:

- *If the PWM Pulse Generator is started(PWC.CE="1"), operate following steps because the PWM Pulse Generator.Compare Data Value must be initialized.*
  1. Set the PWM Compare Register.Compare Data Value(PWC1.D[9:0], PWC2.D[9:0])
  2. Set the PWM Selection Register.Output Selection bits(BS="1", PWS.P2[2:0], PWS.M2[2:0], PWS.P1[2:0], PWS.M1[2:0])
  3. Set the PWM Control Register:Count Enable bit(PWC.CE="1")
- *If the BS bit is "1" , don't change the PWM Compare Register.Compare Data Value(PWC1.D[9:0], PWC2.D[9:0]) and the PWM Selection Register.Output Selection bits (PWS.P2[2:0], PWS.M2[2:0], PWS.P1[2:0], PWS.M1[2:0]).*

[bit13:11] P2[2:0]: Plus Output 2 Selection Bits

The P2[2:0] bits select the output signal for PWM2P.

P2[2:0]	Description
000	"L": PWM2P output is fixed to low level
001	"H": PWM2P output is fixed to high level
01X	PWM pulse is driven on the PWM2P output
1XX	"Hi-Z": PWM2P output is set to high impedance

X: don't care

Note:

- *If the PWS.BS bit is "1" , don't change the PWM Compare Register.Compare Data Value(PWC1.D[9:0], PWC2.D[9:0]) and the PWM Selection Register.Output Selection bits (P2[2:0], PWS.M2[2:0], PWS.P1[2:0], PWS.M1[2:0]).*

[bit10:8] M2[2:0]: Minus Output 2 Selection Bits

The M2[2:0] bits selects the output signal for PWM2M.

M2[2:0]	Description
000	"L": PWM2M output is fixed to low level
001	"H": PWM2M output is fixed to high level
01X	PWM pulse is driven on the PWM2M output
1XX	"Hi-Z": PWM2M output is set to high impedance

X: don't care

Note:

- *If the PWS.BS bit is "1" , don't change the PWM Compare Register.Compare Data Value(PWC1.D[9:0], PWC2.D[9:0]) and the PWM Selection Register.Output Selection bits (PWS.P2[2:0], M2[2:0], PWS.P1[2:0], PWS.M1[2:0]).*

[bit7:6] Reserved: Reserved Bit

When writing, always write "0".

When reading, "0" is always read.

[bit5:3] P1[2:0]: Plus Output 1 Selection Bits

The P1[2:0] bits select the output signal for PWM1P.

P1[2:0]	Description
000	"L": PWM1P output is fixed to low level
001	"H": PWM1P output is fixed to high level
01X	PWM pulse is driven on the PWM1P output
1XX	"Hi-Z": PWM1P output is set to high impedance

X: don't care

Note:

- *If the PWS.BS bit is "1" , don't change the PWM Compare Register.Compare Data Value(PWC1.D[9.0], PWC2.D[9:0]) and the PWM Selection Register.Output Selection bits (PWS.P2[2:0], PWS.M2[2:0], P1[2:0], PWS.M1[2:0]).*

[bit2:0] M1[2:0]: Minus Output 1 Selection Bits

The M1[2:0] bits selects the output signal for PWM1M.

M1[2:0]	Description
000	"L": PWM1M output is fixed to low level
001	"H": PWM1M output is fixed to high level
01X	PWM pulse is driven on the PWM1M output
1XX	"Hi-Z": PWM1M output is set to high impedance

X: don’t care

Note:

- *If the PWS.BS bit is "1" , don't change the PWM Compare Register.Compare Data Value(PWC1.D[9:0], PWC2.D[9:0]) and the PWM Selection Register.Output Selection bits (PWS.P2[2:0], PWS.M2[2:0], PWS.P1[2:0], M1[2:0]).*

4.4. PWM Selection Set Register (PWSS)

The PWM Selection Set Register is used to set the value of the PWS.BS bit.

bit	15	14	13	12	11	10	9	8
Field	Reserved	BSS	Reserved					
Attribute	-	W	-					
Protection attribute	-	-	-					
Initial value	0	0	000000					

bit	7	6	5	4	3	2	1	0
Field	Reserved							
Attribute	-							
Protection attribute	-							
Initial value	00000000							

[bit15] Reserved: Reserved Bit

When writing, always write "0".
When reading, "0" is always read.

[bit14] BSS: Set Bit for the Output Update Bit

This bit sets the value of PWS.BS bit.

Bit	Description
0	No effect
1	Sets the PWS.BS bit to "1". Reading of this bit returns "0".

Notes:

- *If the PWM Pulse Generators is started(PWC.CE="1"), operate following steps because the PWM Pulse Generator.Compare Data Value must be initialized.*
  1. Set the PWM Compare Register.Compare Data Value(PWC1.D[9:0], PWC2.D[9:0])
  2. Set the PWM Selection Register.Output Selection bits(PWS.BS="1", PWS.P2[2:0], PWS.M2[2:0], PWS.P1[2:0], PWS.M1[2:0])
  3. Set the PWM Control Register.Count Enable bit(PWC.CE="1")
- *If the PWS.BS bit is "1" , don't change the PWM Compare Register.Compare Data Value(PWC1.D[9:0], PWC2.D[9:0]) and the PWM Selection Register.Output Selection bits (PWS.P2[2:0], PWS.M2[2:0], PWS.P1[2:0], PWS.M1[2:0]).*

[bit13:0] Reserved: Reserved Bits

When writing, always write "0".
When reading, "0" is always read.

4.5. SMC Trigger Delay Register (PTRGDL)

The SMC Trigger Delay Register (PTRGDL) is used to delay the start of the operation by 0 to 255 clock cycles.

bit	15	14	13	12	11	10	9	8
Field	Reserved							
Attribute	-							
Protection attribute	-							
Initial value	00000000							

bit	7	6	5	4	3	2	1	0
Field	D[7:0]							
Attribute	R/W							
Protection attribute	-							
Initial value	00000000							

[bit15:8] Reserved: Reserved Bits

When writing, always write "0".
When reading, "0" is always read.

[bit7:0] D[7:0]: Trigger Delay Bits

The Trigger Delay bits (D[7:0]) configure a delay of the trigger input that starts the PWM operation from 0 to 255 clock cycles.

D[7:0]	Description
0x00	No delay (delay disabled)
0x01	Delay of 1 clock cycle between the trigger and the clock cycle in which the PWM generation starts
…	…
0xFF	Delay of 255 clock cycles between the trigger and the clock cycle in which the PWM generation starts

CHAPTER 60: Trigger Configuration of Stepper Motor Controller

This chapter explains the trigger configuration of stepper motor controller.

1. Overview
2. Configuration and Block Diagram
3. Operation
4. Registers

SMCTRG-T01P00R01L01-E1-GBM01

1. Overview

The chapter describes the function and its operation of both SMC trigger selection register and SMC trigger register. See the chapter of Stepper motor controller with this chapter.

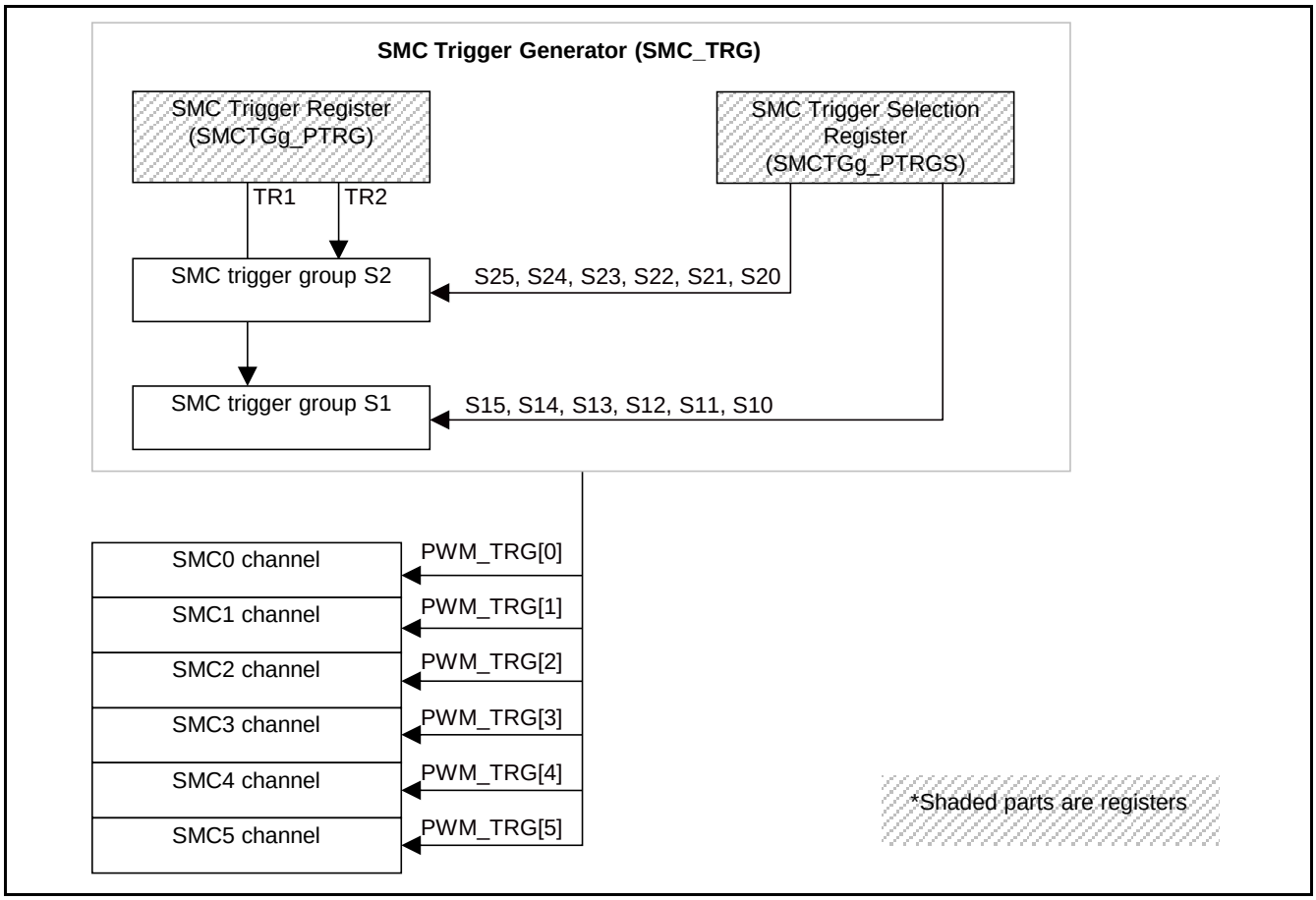
2. Configuration and Block Diagram

See the block diagram below.

Notes:

- *The suffix 'g' in the register name indicates that the register is an instance 'g' of the SMC Trigger Generator.*
- *The suffix 'i' in the register name indicates that the register is an instance 'i' of the SMC channel.*

Figure 2-1 Block Diagram of the SMC Module



3. Operation

The trigger configuration and its operation is described in 4.

4. Registers

Abbreviated Register Name	Register Name	Reference
SMCTGg_PTRGS	SMC Trigger Selection Register	4.1
SMCTGg_PTRG	SMC Trigger Register	4.2

4.1. SMC Trigger Selection Register (SMCTGg_PTRGS)

The SMC Trigger Selection Register (SMCTGg_PTRGS) selects multiple SMC channels that shall be triggered by the SMC Trigger Register to start operation synchronously or with a delay. The SMC channels can be combined into two trigger groups.

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved		S25	S24	S23	S22	S21	S20
ACCESS_TYPE	R0,WX		R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Rp/Wp							
INITIAL_VALUE	0		0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved		S15	S14	S13	S12	S11	S10
ACCESS_TYPE	R0,WX		R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Rp/Wp							
INITIAL_VALUE	0		0	0	0	0	0	0

[bit13] S25 : Trigger Enable for Operation of SMC<6*g+5>

This bit selects SMC<6*g+5> for the SMC trigger group S2 (triggered by SMCTGg_PTRG:TR2).

S25	Description
0	SMC<6*g+5> is not triggered by SMCTGg_PTRG:TR2
1	SMC<6*g+5> is triggered by SMCTGg_PTRG:TR2

[bit12] S24 : Trigger Enable for Operation of SMC<6*g+4>

This bit selects SMC<6*g+4> for the SMC trigger group S2 (triggered by SMCTGg_PTRG:TR2).

S24	Description
0	SMC<6*g+4> is not triggered by SMCTGg_PTRG:TR2
1	SMC<6*g+4> is triggered by SMCTGg_PTRG:TR2

[bit11] S23 : Trigger Enable for Operation of SMC<6*g+3>

This bit selects SMC<6*g+3> for the SMC trigger group S2 (triggered by SMCTGg_PTRG:TR2).

S23	Description
0	SMC<6*g+3> is not triggered by SMCTGg_PTRG:TR2
1	SMC<6*g+3> is triggered by SMCTGg_PTRG:TR2

[bit10] S22 : Trigger Enable for Operation of SMC<6*g+2>

This bit selects SMC<6*g+2> for the SMC trigger group S2 (triggered by SMCTGg_PTRG:TR2).

S22	Description
0	SMC<6*g+2> is not triggered by SMCTGg_PTRG:TR2
1	SMC<6*g+2> is triggered by SMCTGg_PTRG:TR2

[bit9] S21 : Trigger Enable for Operation of SMC<6*g+1>

This bit selects SMC<6*g+1> for the SMC trigger group S2 (triggered by SMCTGg_PTRG:TR2).

S21	Description
0	SMC<6*g+1> is not triggered by SMCTGg_PTRG:TR2
1	SMC<6*g+1> is triggered by SMCTGg_PTRG:TR2

[bit8] S20 : Trigger Enable for Operation of SMC<6*g+0>

This bit selects SMC<6*g+0> for the SMC trigger group S2 (triggered by SMCTGg_PTRG:TR2).

S20	Description
0	SMC<6*g+0> is not triggered by SMCTGg_PTRG:TR2
1	SMC<6*g+0> is triggered by SMCTGg_PTRG:TR2

[bit5] S15 : Trigger Enable for Operation of SMC<6*g+5>

This bit selects SMC<6*g+5> for the SMC trigger group S1 (triggered by SMCTGg_PTRG:TR1).

S15	Description
0	SMC<6*g+5> is not triggered by SMCTGg_PTRG:TR1
1	SMC<6*g+5> is triggered by SMCTGg_PTRG:TR1

[bit4] S14 : Trigger Enable for Operation of SMC<6*g+4>

This bit selects SMC<6*g+4> for the SMC trigger group S1 (triggered by SMCTGg_PTRG:TR1).

S14	Description
0	SMC<6*g+4> is not triggered by SMCTGg_PTRG:TR1
1	SMC<6*g+4> is triggered by SMCTGg_PTRG:TR1

[bit3] S13 : Trigger Enable for Operation of SMC<6*g+3>

This bit selects SMC<6*g+3> for the SMC trigger group S1 (triggered by SMCTGg_PTRG:TR1).

S13	Description
0	SMC<6*g+3> is not triggered by SMCTGg_PTRG:TR1
1	SMC<6*g+3> is triggered by SMCTGg_PTRG:TR1

[bit2] S12 : Trigger Enable for Operation of SMC<6*g+2>

This bit selects SMC<6*g+2> for the SMC trigger group S1 (triggered by SMCTGg_PTRG:TR1).

S12	Description
0	SMC<6*g+2> is not triggered by SMCTGg_PTRG:TR1
1	SMC<6*g+2> is triggered by SMCTGg_PTRG:TR1

[bit1] S11 : Trigger Enable for Operation of SMC<6*g+1>

This bit selects SMC<6*g+1> for the SMC trigger group S1 (triggered by SMCTGg_PTRG:TR1).

S11	Description
0	SMC<6*g+1> is not triggered by SMCTGg_PTRG:TR1
1	SMC<6*g+1> is triggered by SMCTGg_PTRG:TR1

[bit0] S10 : Trigger Enable for Operation of SMC<6*g+0>

This bit selects SMC<6*g+0> for the SMC trigger group S1 (triggered by SMCTGg_PTRG:TR1).

S10	Description
0	SMC<6*g+0> is not triggered by SMCTGg_PTRG:TR1
1	SMC<6*g+0> is triggered by SMCTGg_PTRG:TR1

4.2. SMC Trigger Register (SMCTGg_PTRG)

The SMC Trigger Register is used to generate trigger signals for the two trigger groups configured in the SMC Trigger Selection Register (SMCTGg_PTRGS). The trigger group can be used to synchronously start several SMC channels. The operation (count enable) starts depending on the delay configured in the SMCi_PTRGDL register.

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved							
ACCESS_TYPE	R0,WX							
PROT_TYPE	Rp/Wp							
INITIAL_VALUE	0							

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved						TR2	TR1
ACCESS_TYPE	R0,WX						R0,W	R0,W
PROT_TYPE	Rp/Wp							
INITIAL_VALUE	0						0	0

[bit1] TR2 : SMC Trigger 2

This bit triggers the SMC trigger group S2 of SMCTGg. The configuration in SMCTGg_PTRGS:S2x determines which SMC channels are triggered. The trigger signal starts the delay counters (if enabled in SMCi_PTRGDL). If the delay has elapsed or the delay is disabled, the Count Enable (SMCi_PWC:CE) bits of the selected SMC channels will be set and the operation starts (PWM generation and output control).

TR2	Description
0	No effect
1	Triggers selected SMC channels via the Set logic for SMCi_PWC:CE

Notes:

- *This bit is cleared automatically to '0' after one clock cycle.*
- *A second trigger should not be applied to a group which has already been triggered. If this happens, the running delay counter will reset its value.*

[bit0] TR1 : SMC Trigger 1

This bit triggers the SMC trigger group S1 of SMCTGg. The configuration in SMCTGg_PTRGS:S1x determines which SMC channels are triggered. The trigger signal starts the delay counters (if enabled in SMCi_PTRGDL). If the delay has elapsed or the delay is disabled, the Count Enable (SMCi_PWC:CE) bits of the selected SMC channels will be set and the operation starts (PWM generation and output control).

TR1	Description
0	No effect
1	Triggers selected SMC channels via the Set logic for SMCi_PWC:CE

Notes:

- *This bit is cleared automatically to '0' after one clock cycle.*
- *A second trigger should not be applied to a group which has already been triggered. If this happens, the running delay counter will reset its value.*

CHAPTER 61: Partial Wakeup Control

This chapter explains the partial wakeup control function and its operations.

1. Overview
2. Configuration
3. Explanation of Operation
4. Registers
5. Precautions for Using This Device

PWU-T00P00R04L00-E1-GBB01

1. Overview

This section provides an overview of the partial wakeup control.

Partial wakeup (PWU) mode is implemented in this device.

PWU mode is one of the PSS (Power Saving State) modes. In this mode, only the functions required for voltage monitoring by the A/D converter are operating.

Effective use of this function enables monitoring of voltage, such as in sensor output, with low-power consumption and without starting the CPU.

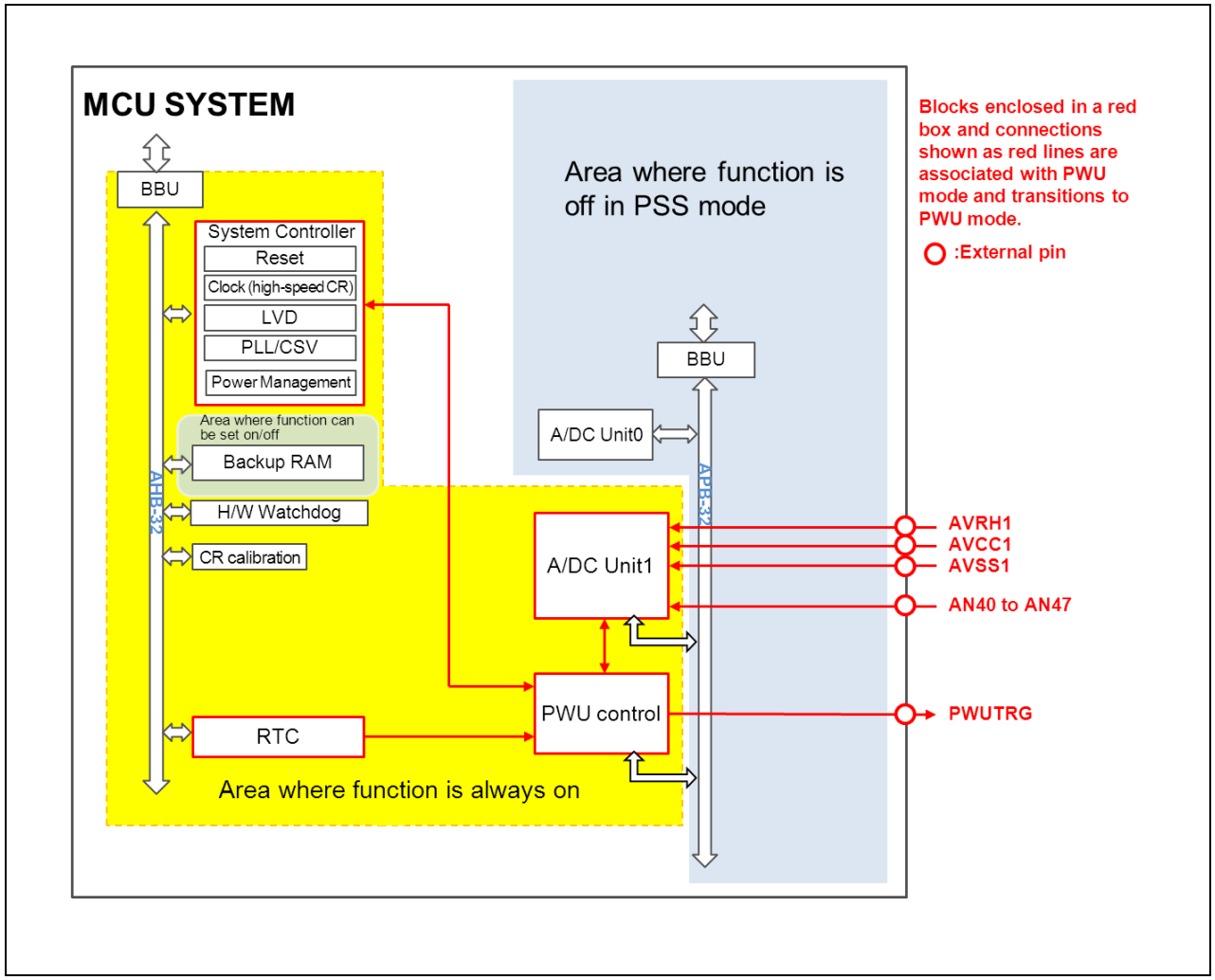
2. Configuration

This section explains the partial wakeup configuration.

Relationship Between the entire System and the partial wakeup Function

Figure 2-1 shows the relationship between the entire system and the partial wakeup function.

Figure 2-1 Relationship Between the Entire System and the Partial Wakeup Function (Describing Only the Relationship with Related Blocks)



■ RTC

This block counts the cycles for transitioning to PWU mode. The cycles can be set in steps of 8 ms, between 8 and 64 ms. For the setting method, see "3. Explanation of Operation."

Note:

The term of 8ms is generated from the following calculation in RTC.
0.25[s] ÷ 32 = 7.8125[ms] (≈ 8[ms])
Moreover as RTC operates with low-speed CR (typical 100KHz) in PWU mode, the actual cycles for transitioning to PWU mode become as the following.

7.81[ms] (The description in this chapter is “8ms”)
15.62[ms] (The description in this chapter is “16ms”)
23.43[ms] (The description in this chapter is “24ms”)
31.24[ms] (The description in this chapter is “32ms”)
39.05[ms] (The description in this chapter is “40ms”)
46.86[ms] (The description in this chapter is “48ms”)
54.67[ms] (The description in this chapter is “56ms”)
62.48[ms] (The description in this chapter is “64ms”)

■ System controller

- This block manages the device state (PSS or RUN).
- The block turns on the high-speed CR oscillator when transitioning to PWU mode.

■ PWU control

- This block controls the high-speed CR oscillator, the "PWUTRG" pin output function, and A/DC Unit1.
- In PWU mode, the block outputs "H" from the "PWUTRG" pin.
- The block starts A/D conversion within a certain time after the PWUTRG pin outputs "H". The time until A/D conversion start can be set in steps of 50 us, between 50 and 5100 us.

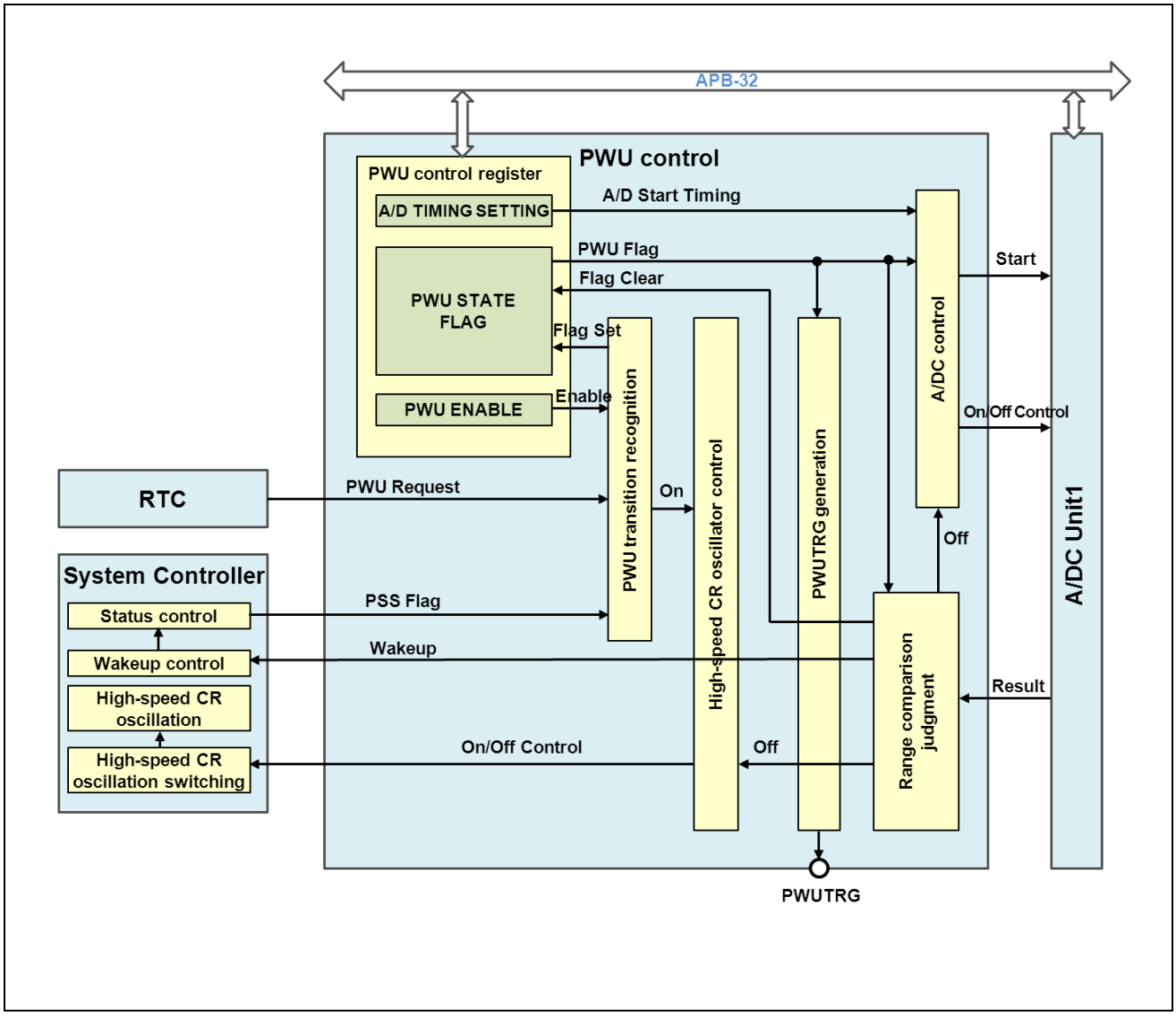
■ A/DC Unit1

- In PWU mode, the A/D converter can be used with the 8 channels from AN40 to AN47 out of all channels.
- In PWU mode, only the range comparison function of the A/D converter can be used. For details, see "CHAPTER: 12-BIT A/D CONVERTER."

Block Diagram of the partial wakeup Function

Figure 2-2 is a block diagram of the partial wakeup function.

Figure 2-2 Block Diagram of the Partial Wakeup Function



- PWU control register

  The configuration includes the "partial wakeup mode enable bit" and the "A/D conversion start time setting register." For details, see "CHAPTER 61:4. Registers."

- PWU transition recognition

  This block judges whether the conditions for transitioning to PWU mode are met. Also, the block turns on the high-speed CR oscillator.

- High-speed CR oscillator control
  This block is used for On/Off control of the high-speed CR oscillator in PWU mode.

- PWUTRG generation
  In PWU mode, this block outputs "H" from the "PWUTRG" pin.

- A/DC control
  This block is used to turn A/DC on or off and to adjust the timing of conversion start in PWU mode.
  The block starts A/D conversion within a certain time after the PWUTRG pin outputs "H".
  The time until A/D conversion start can be set in steps of 50 us, between 50 and 5100 us.

- Range comparison judgment
  This block determines, from an A/D conversion range comparison, whether to transition to RUN mode or PSS mode.
  To transition to RUN mode, it generates a Wakeup signal.
  After the completion of A/D conversion, the block generates a signal for turning off the high-speed CR oscillator and A/DC Unit1.

3. Explanation of Operation

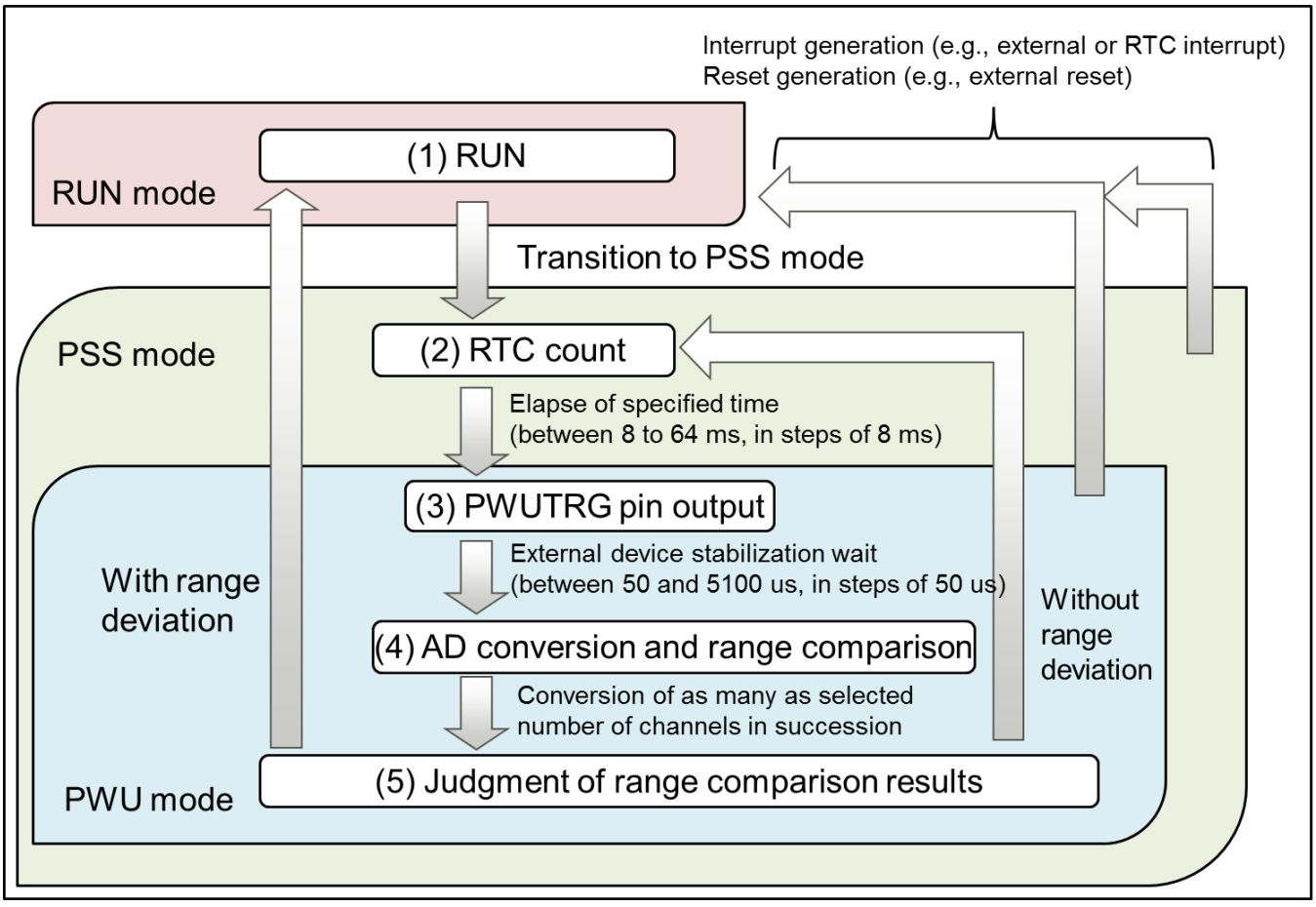
This section explains the operations of partial wakeup control.

3.1. State Transition Diagram

Figure 3-1 shows the relationship between partial wakeup mode and other modes.

The figure outlines the operations. For a more detailed explanation of the operations, see the next section, "Flowchart."

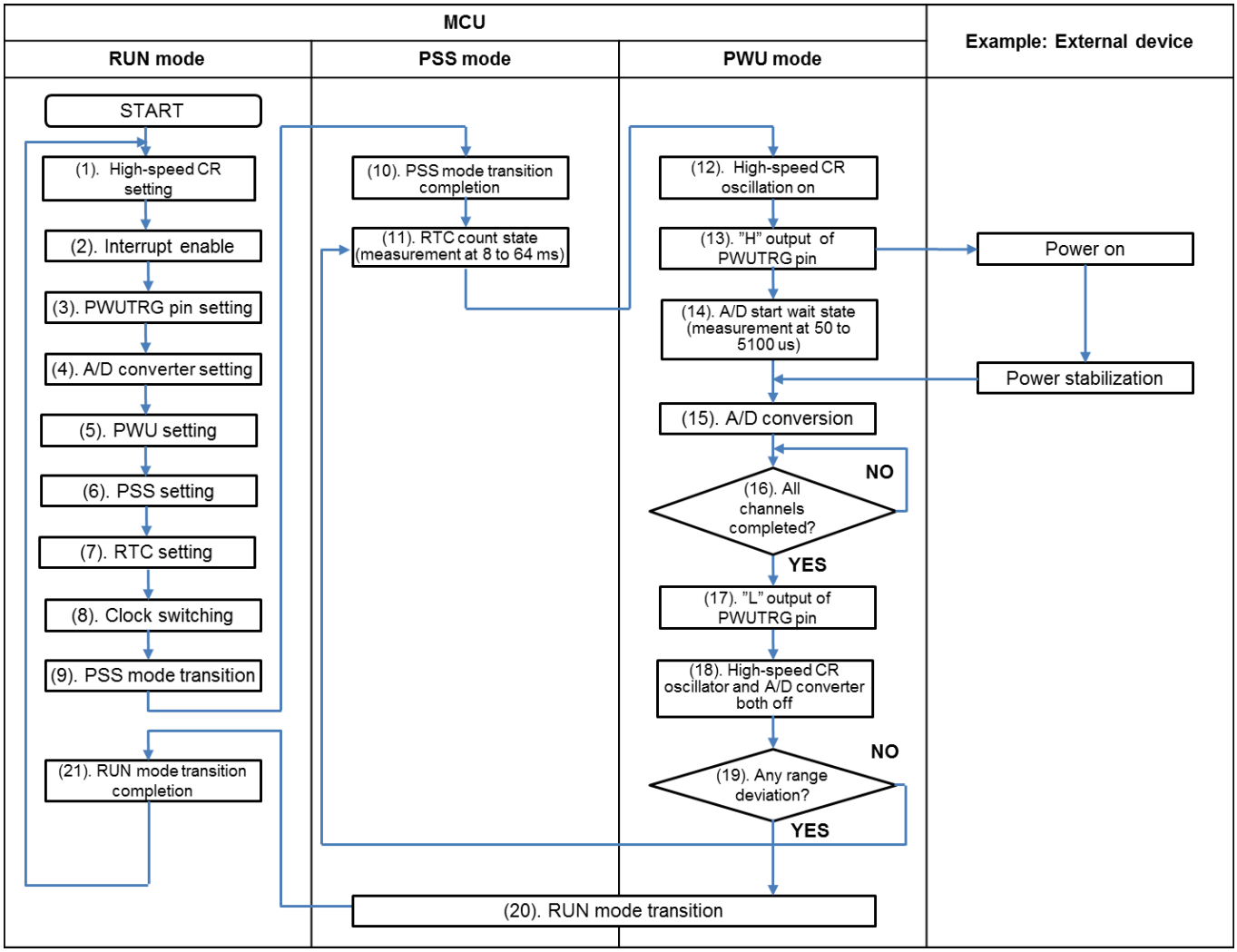
Figure 3-1 Partial Wakeup State Transition Diagram



3.2. Flowchart

This section uses the flowchart in Figure 3-2 to explain the partial wakeup function in detail.

Figure 3-2 Partial Wakeup Function Flowchart



(1) High-Speed CR Setting (Processing by Software)

(1)-1. Calibration

Software performs high-speed CR oscillation calibration and corrects any deviation in the A/D conversion time and any deviation in the period from PWUTRG pin output to A/D conversion start (external device stabilization wait time). Such deviations are due to process variations and variations depending on the use conditions.

For the setting method for the above, see "CHAPTER: CR CALIBRATION."

(1)-2. Stabilization wait time

Set the PSCL bit and CMPR bit in SYSC_FCRCTCPR register to initial (default) value.

These settings are applied to “(12) High-speed CR oscillation on.”

If stabilization wait time is set to large value and PWU mode time exceeds PWU transition cycle,

This device becomes illegal state.

In this state, the transition to RUN mode can’t occur by A/D converter range deviation.

For the setting method for the above, see "CHAPTER: SOURCE CLOCK TIMER."

(2) Interrupt Enable (Processing by Software)

You can treat enabled interrupt factors as return factors from PSS mode and PWU mode.
For any return due to an interrupt, there is a transition to the "(20)." processing.

The following can be specified: the interrupt of a resource not in the Power Down state, even in PSS mode.

Examples include external interrupt, RTC, NMI, low-voltage detection*1.

(*1 When used as an interrupt at the abnormal state detection)

For the method of enabling interrupts, see the chapters on the respective resources.
Note that this section does not cover A/DC-related settings because they are handled in the "(4)." processing.

(3) PWUTRG Pin Setting (Processing by Software)

Here, software makes the port output setting for output of the PWUTRG function.

You can set it by writing appropriate value to the POF bit in the corresponding PPC_PCFGRijj (i=0 to 4, jj=00 to 31) register.
For a corresponding register and setting, see "APPENDIX: VARIOUS SETTINGS OF I/O PORTS."

Note that this section does not cover settings related to "retaining the pin state in PSS mode" because they are handled in the "(6)." processing.

(4) A/D Converter Setting (Processing by Software)

Table 3-1 lists the setting values for writable registers related to A/D converter unit 1, corresponding to PWU mode.

The following values are degrees of limitations.

3: Always set values according to the "setting value."

2: The recommended setting is based on the "setting value." It can be changed to match the use conditions.

1: Set a value appropriate to the use conditions.

Interpret the values as described.

For a detailed explanation of the registers and the setting method, see "CHAPTER: 12-BIT A/D CONVERTER."

Note:

- *In PWU mode, the number of activation channel which can be used is maximum 8.*

Table 3-1 A/D Converter Settings

Register Name	Setting Value	Degree of Limitation
KEYCDR	Before writing to the ADER1 register, write to this register by using the prescribed method and then release the write protection.	3
ADER1	Set the bit corresponding to the pin used for analog input to "1", and set the other pins to "0". Except for AN40 to AN47, analog input is not supported, so set the ADEn (n=32 to 39, 48 to 63) bits to "0".	3
ADTSS1	Do not write.	3
ADTSE1	Set all bits to "0". (Software start is prohibited.)	3
ADCOMPB32 to ADCOMPB63	Do not use with the PWU function. (There is no effect on operation.)	1
ADTCS32 to ADTCS63 * Select only the registers corresponding to the numbers assigned as activation channels.	INTE=1'b0	3
	{STS1,STS0}=2'b10	3
	RPT=1'b1	3
	PRT=1'b0	3
	PRTS = Do not use with the PWU function. (There is no effect on operation.)	1
	{SEL1, SEL0} = Do not use with the PWU function. (There is no effect on operation.)	1
	BUFX = Do not use with the PWU function. (There is no effect on operation.)	1
	BTS = Do not use with the PWU function. (There is no effect on operation.)	1

Register Name	Setting Value	Degree of Limitation
ADTECS32 to ADTECS63 * Select only the registers corresponding to the numbers assigned as activation channels.	STS2=1'b0	3
	Value of {CHSEL4, CHSEL3, CHSEL2, CHSEL1, CHSEL0}=5'b01000 to 5'b01111 Activation channels and analog input pins are associated. Since only analog input pins AN40 to AN47 are associated, the settings must associate these pins.	3
ADRCUT4 to ADRCUT7	Set a value appropriate to the use conditions.	1
ADRCLT4 to ADRCLT7	Set a value appropriate to the use conditions.	1
ADRCCS32 to ADRCCS63 * Select only the registers corresponding to the numbers assigned as activation channels.	{RCOCD2, RCOCD1, RCOCD0}=3'b001	3
	RCOIRS = Set a value appropriate to the use conditions.	1
	RCOIE=1'b1	3
	RCOE=1'b1	3
	{RCOTS1, RCOTS0} = Set a value appropriate to use conditions.	1
ADSCANS1	SCIE=1'b1	3
	SCMD=1'b1	3
ADNCS16 to ADNCS31 * Select only the registers corresponding to the numbers assigned as activation channels.	CNTEN=1'b1	3
	{CCNT1, CCNT0}=2'b00	3
ADTCSC32 to ADTCSC63 * Select only the registers corresponding to the numbers assigned as activation channels.	Do not use with the PWU function.	1
ADRCIFC1	Write 1 to clear the range comparison interrupt flag corresponding to the number assigned as the activation channel. (Do not transition to PWU mode with interrupts left enabled.)	3
ADSCANSC1	Write 1 to clear the scan conversion completion interrupt flag. (Do not transition to PWU mode with interrupts left enabled.)	3
ADMD1	STPCEN=1'b0	2
	{CT1, CT0}=2'b00 (2'b11 is a prohibited setting.) The built-in high-speed CR oscillation clock (4 MHz) is divided according to the setting, and the compare time is measured using the divided clock. For details and notes on the settings, see "CHAPTER: 12-BIT A/D CONVERTER."	2
	{ST1, ST0}=2'b00 (2'b11 is a prohibited setting.) The built-in high-speed CR oscillation clock (4 MHz) is divided according to the setting, and the sampling time is measured using the divided clock. For details and notes on the settings, see "CHAPTER: 12-BIT A/D CONVERTER."	2
ADSTPCS8 to ADSTPCS15 * Select only the registers corresponding to the numbers assigned as activation channels.	{STCH1, STCH0}=2'b00 (2'b11 is a prohibited setting.) The built-in high-speed CR oscillation clock (4 MHz) is divided according to the setting, and the sampling time is measured using the divided clock. For details and notes on the settings, see "CHAPTER: 12-BIT A/D CONVERTER."	2

(5) PWU Setting (Processing by Software)

Here, software makes the following settings.

(5)-1. Enable the PWU function.

(5)-2. Set the external device stabilization wait time, which is the period between PWUTRG pin output and A/D conversion start.

You can select stabilization wait times from 50 to 5100 us, in steps of 50 us.

For the method of setting the above, see "CHAPTER 61:4. Registers" in this chapter.

(6) PSS Setting (Processing by Software)

Table 3-2 lists the setting values for the writable registers in the PSS profile register group and the writable registers of the system special setting registers. The following values are degrees of limitations.

3: Always set values according to the "setting value."

2: The recommended setting is based on the "setting value." It can be changed to match the use conditions.

1: Set a value appropriate to the use conditions.

Interpret the values as described.

For a detailed explanation of the registers and the setting method, see "CHAPTER: LOW-POWER CONSUMPTION."

Table 3-2 PSS Profile Register Group Settings

Register Name	Setting Value	Degree of Limitation
SYSC0_PSSPDCFGR	PD6_0EN=1'b0	3
	PD4_1EN = Set a value appropriate to the Backup RAM use conditions.	1
	PD4_0EN = Set a value appropriate to the Backup RAM use conditions.	1
	PD2EN=1'b0	3
SYSC0_PSSCKSRER	SSCG0EN=1'b0	3
	PLL0EN=1'b0	3
	MOSCEN=1'b0	3
	SCROSCEN=1'b1	3
	CROSCEN=1'b0	3
SYSC0_PSSCKSELR	CDMCUCCSL=3'b111	3
SYSC0_PSSCKER	ENCLKMCUCP=1'b0	3
	ENCLKMCUCH=1'b0	3
SYSC0_PSSCKDIVR	MCUCPDIV=4'b0000	3
	MCUCHDIV=5'b00000	3

Register Name	Setting Value	Degree of Limitation
SYSC0_PSSPLLxCNTR	PLLxISEL=1'b0	2
	PLLxDIVN=8'b00001101	2
	PLLxDIVM=4'b0001	2
	PLLxDIVL=2'b00	2
SYSC0_PSSSSCGxCNTR0	SSCGxISEL=1'b0	2
	SSCGxDIVN=8'b00001101	2
	SSCGxDIVM=4'b0001	2
	SSCGxDIVL=2'b00	2
SYSC0_PSSSSCGxCNTR1	SSCGxSSEN=1'b0	2
	SSCGxFREQ=2'b00	2
	SSCGxMODE=1'b0	2
	SSCGxRATE=10'b0000101001	2
SYSC0_PSSLVDCFGR	LVDL1S = Set a value appropriate to the use conditions.	1
	LVDL1V = Set a value appropriate to the use conditions.	1
	LVDL1E = Set a value appropriate to the use conditions.	1
	LVDH1S = Set a value appropriate to the use conditions.	1
	LVDH1V = Set a value appropriate to the use conditions.	1
	LVDH1E = Set a value appropriate to the use conditions.	1
SYSC0_PSSCSVCFGR	SSCG0CSVE=1'b0	3
	PLL0CSVE=1'b0	3
	MOCSVE=1'b0	3
SYSC0_PSSREGCFGR	RMSEL=1'b1	3
SYSC0_SPECFGR	HOLDIO_PD6_0=1’b0	3
	HOLDIO_PD2=1'b1 The setting by this bit is reflected immediately after writing register, it is not after PSS profile change.	3
	PSSPADCTRL=1'b0	3
	BRAMSC=1'b1 If the setting is PD4_0EN=1 or PD4_1EN=1, exercise Backup RAM sleep control to suppress current dissipation in PSS mode.	3
SYSC1_PSSCKSELR0	LAPP1ACSL=1’b0	3
	LAPP0ACSL=1’b0	3
	LCP1ACSL=1’b0	3
	LCP0ACSL=1’b0	3
	CD0CSL=3’b111 If this setting is not made, a PSS profile error occurs at a PSS mode transition.	3

(7) RTC Setting (Processing by Software)

The following shows examples of RTC settings. For details on the setting method, see "CHAPTER: REAL-TIME CLOCK."

Example of settings

(7)-1. Initialization settings

Sequence	Set Contents *1	Set Timing
1-1.	Write "1" to the ST bit in the RTC_WTCR register.	1st setting only after POR
1-2.	Write "0" to the ST bit in the RTC_WTCR register.	1st setting only after POR
1-3.	Wait until the CLK_STS bit in the RTC_WTSR register becomes 1'b0.	The setting “1-3” is required to set “1-4”. 1st setting only after POR (RTC Clock = 100KHz)
1-4.	Write "2'b10" to the RCKSEL bit in the RTC_WTCR register, write "1" to the CSM bit, and select the low-speed CR clock.	
1-5.	Set each bit in the RTC_WRT register.	1st setting only after POR

*1 Don’t write to the registers by bit level.
In bit-level-write, the no-change-bit have the function which the same value as read data is written to the register. In this case, the bit having difference function between read and write is changed to the other setting.
Therefore the bits no change is intended will be changed.

(7)-2. Calibration setting (for correcting any deviation in the count values due to variations in low-speed oscillation)

Sequence	Set Contents *1	Set Timing
2-1.	Write a value to the DURMW bit in the RTC_DURMW register, and set the calibration period. Normally, set the calibration period to 0.25[s]. To set a time 0.25[s] or shorter, a value of 0.25[s] divided by 1, 2, 4, 8, or 16 can be selected for the period. However, the lower the value set, the lower the accuracy becomes. (Related to the SCAL[2:0] bit) When the main clock is 4 MHz and the calibration period is set to 0.25[s], set DURMW bit value="0xF4240" (1'000'000 as a decimal).	1st setting only after POR

Sequence	Set Contents *1	Set Timing
2-2.	Set the related bits in the RTC_WTCR register as described below.	
	− Set the SCAL [2:0] bit according to the RTC_DURMW value. If the calibration period is set to 0.25[s], set "'3'b000".	1st setting only after POR
	− Set the CCKSEL bit to 1'b1, and select the low-speed CR.	1st setting only after POR (RTC Clock = 100KHz)
	− Set the ENUP bit according to the use conditions. To update the calibration result automatically as hardware, set "1'b1". To update it with software, set "1'b0".	1st setting only after POR
	− Set the ACAL bit to "1'b0", and disable the auto-calibration.	1st setting only after POR
2-3.	Write "1'b1" to the CALDC bit in the RTC_WINC register, and clear the calibration interrupt.	Clear the interrupt flag after interrupt occurs
2-4.	Write "1'b1" to the MTRG bit in the RTC_WTCR register, and start calibration.	Set in the timing you want to calibrate.(e.g. Post-PWU)
2-5.	Wait until the CALD bit in the RTC_WINS register becomes "1'b1".	The calibration is available on real time count.

*1 Refer to the previous table.

(7)-3. Partial wakeup setting - Case 1: When the ENUP bit in (7)-2-2 is set to "1'b1". -

Sequence	Set Contents *1	Set Timing
3-1.	Set the related bits in the RTC_PWUTRGCR register as described below.	
	Set the MD bit to "1'b1".	1st setting only after POR
	For the SEL bit, set the cycles for transitioning to PWU mode in accordance with the purpose. Values ranging from 8 to 64 ms can be selected in units of 8 ms.	The period of PWU can be switched.
3-2.	Wait until the BUSY bit in the RTC_PWUTRGSR register becomes 1'b0.	Set if RTC_PWUTRGCR register is changed.

*1 Refer to the previous table.

(7)-3. Partial wakeup setting - Case 2: When the ENUP bit in (7)-2-2 is set to "1'b0". -

Sequence	Set Contents *1	Set Timing
3-1.	Read the value of RTC_CNTCAL, execute [-1 processing], and then execute [divide-by-32 processing]. The RTC_CNTCAL includes the counter value of 0.25[s]. By dividing by 32, it generates a count value of approximately 8 ms.	Set if the calibration (the setting of “2-3”-”2-5”) is completed.
3-2.	Read the value of RTC_CNTCAL, and execute [-1 processing].	

Sequence	Set Contents *1	Set Timing
	Write the post processing value to the WTBR bit in the RTC_WTBR register.	
3-3.	Set the related bits in the RTC_PWUTRGCR register as described below.	
	− Set the C8MRL bit to the value calculated in "(7)-3-1."	Set if the calibration (the setting of “2-3”-”2-5”) is completed.
	− Set the MD bit to "1'b0".	1st setting only after POR
	− For the SEL bit, set the cycles for transitioning to PWU mode in accordance with the purpose. Values ranging from 8 to 64 ms can be selected in units of 8 ms.	The period of PWU can be switched.
3-4.	Wait until the BUSY bit in the RTC_PWUTRGSR register becomes 1'b0.	Set if RTC_PWUTRGCR register is changed.

*1 Refer to the previous table.

(7)-4. RTC count start

Sequence	Set Contents *1	Set Timing
4-1.	Set the RTC_WINE register, and enable the required interrupts.	1st setting only after POR
4-2.	Write "1" to each bit in the RTC_WINC register, and clear the interrupt flag.	Clear the interrupt flags after interrupts occur
4-3.	Write "1" to the ST bit in the RTC_WTCR register, and start counting.	1st setting only after POR (RTC Clock = 100KHz)

*1 Refer to the previous table.

(8) Clock Switching (Processing by Software)

Here, software makes the following settings.

(8)-1. Clock gear down operation

If the PLL clock is being used for operating the internal circuit, use the clock gear down function to set the clock frequency lower in stages to reduce the current fluctuations due to clock switching.

For details on how to use the clock gear down function, see the "Clock Gear" section in "CHAPTER: CLOCK SYSTEM."

(8)-2. Clock adjustment

To transition to the PSS, establish a 1:1:1:1 relationship between the CPU clock, the memory configuration clock, the SCU clock, and the MCUCH clock.

The absence of this relationship between these clocks may cause a malfunction.

Additionally, PLL and SSCG-PLL must be disabled before PSS mode transition.

Example of settings

Because the SCU clock operates with the high-speed CR oscillation clock,

switch the CPU clock (CLK_CPU), memory configuration clock (CLK_MEMC), and MCUCH clock (CLK_SYSC0H) so that they all operate with the high-speed CR oscillation clock.

(8)-2-1. Set the CD0CSL bit in the SYSC1_RUNCKSELR0 register to "All-0".

(8)-2-2. Set the HPMDIV bit and the SYSDIV bit in the SYSC1_RUNCKDIVR0 register to "All-0".

(8)-2-3. Set the CDMCUCCSL bit in the SYSC0_RUNCKSELR register to "All-0".

(8)-2-4. Set the MCUCHDIV bit in the SYSC0_RUNCKDIVR register to "All-0".

(8)-2-5. Set the PLL0EN and SSCG0EN bit in the SYSC0_RUNCKSRER register to "All-0".
(PLL and SSCG-PLL are disabled.)

(8)-2-6. Set the SYSC0_RUNPLL0CNTR, SYSC0_RUNSSCG0CNTR0 and SYSC0_RUNSSCG0CNTR1
register to appropriate value to avoid profile error.

For details on the combination of violation settings,
see "Profile" in "CHAPTER: LOW-POWER CONSUMPTION".

(8)-2-7. Set the SYSC1_RUNENR register to "0xAB".

(8)-2-8. Set the SYSC0_TRGRUNCNTR register to "0xAB". (The RUN profile update starts.)

(8)-2-9. Wait until the RUNDF0 bit in the SYSC0_SYSSTSR register becomes "1".

(9) PSS Mode Transition (Processing by Software)

See "Operation Procedure" in "CHAPTER: LOW-POWER CONSUMPTION," and transition to PSS mode.

(10) PSS Mode Transition Completion (Processing by Hardware)

There has been a transition to PSS mode according to the PSS profile register settings at "(6)."

(11) RTC Count State (Processing by Hardware)

Hardware waits until the RTC count value reaches the cycles set at "(7)."

After the cycles are reached, there is a transition to PWU mode.

For a transition from RUN mode to PSS mode, the RTC is not initialized. For this reason, after the transition from RUN mode to PSS mode, the time until the transition to PWU mode is less than the setting at "(7)."

(The time depends on the RTC count value at the transition to PSS mode.)

(12) High-Speed CR Oscillation on (Processing by Hardware)

Hardware turns on high-speed CR oscillation.

Stabilization wait time is inserted to turn on high-speed CR oscillation.

This wait time is set at “(1).”

PWU mode operates with the output clock of the high-speed CR oscillation.

(13) "H" Output of the PWUTRG Pin (Processing by Hardware)

Hardware changes PWUTRG to "H".

PWUTRG is "L" before this processing.

(14) A/D Start Wait State (Processing by Hardware)

Hardware waits until it arrives at the time set at "(5)."

The power supply of external devices is assumed to be stable during this wait period.

When the time comes, A/D conversion starts.

(15) A/D Conversion (Processing by Hardware)

A/D conversion is executed according to the settings at "(4)."

(16) A/D Conversion Judgment (Processing by Hardware)

Hardware waits until the conversion of all selected channels (maximum of 8 channels) completes.

(17) "L" Output of the PWUTRG Pin (Processing by Hardware)

Hardware switches PWUTRG output from "H" to "L".

(18) High-Speed CR Oscillator and A/D Converter both Off (Processing by Hardware)

Hardware turns off the high-speed CR oscillator and A/D converter.

(19) Range Comparison Judgment (Processing by Hardware)

Hardware judges whether a range deviation has arisen as a result of the A/D conversion at "(15)."

If no deviation has arisen, there is a transition to PSS mode again.

If a deviation has arisen, a WAKEUP signal is issued, followed by a transition to RUN mode to wake up the CPU.

(20) RUN Mode Transition (Processing by Hardware)

For details on the operation, see "Operation Procedure" in "CHAPTER: LOW-POWER CONSUMPTION."

(21) RUN Mode Transition Completion (Processing by Hardware)

The transition to RUN mode has completed.
There are the following notes to design software.

(21)-1. Regarding PSS enable setting

Set PSSEN0 bit in SYSC0_PSSENR register to "0x00".
Otherwise, written data to registers in SYSC0 is invalid. Bus error will occur.

But the set to PSSEN1 bit in SYSC1_PSSENR register is not required
because of initialization with power down.

(21)-2. Regarding the judgment of wakeup factor

If the following “(21)-2-1” and “(21)-2-2” both are "0" before "(9) PSS mode transition (processing by software)" and "1" after RUN mode transition, you are able to judge RUN mode transition occurred by A/DC range deviation.

(21)-2-1. Any bits among RCINT32 to RCINT63 in ADRCIF1 register

(21)-2-2. SCINT bit in ADSCANS1 register

Moreover, in case that you set in "(2). Interrupt enable" process,
RUN mode transition might occur by the factor you set.

You need to confirm these interrupt flag.

(21)-3. Regarding A/D Converter use after RUN mode transition

When A/D Converter is executed in PWU mode, A/D Converter becomes "Pause state". *1

In case of Pause state, BUSY flag is activated.
In case that BUSY flag is activated, the change of A/D Converter setting is impossible.
For example, When CNTEN bit in ADNCS16 to ADNCS31 registers is changed,

A/D Conversion started unexpectedly.
When you change A/D Converter setting after run mode transition,
clear the BUSY flag before changing the setting.

Example of settings
Write "1" to BUSYC bit in ADTCSC32 to ADTCSC63 register.

*1. Regarding "Pause state",
see the explanation of “Pause scan conversion when the conversion count is specified for each channel” in ”5.2.12. About the Scan Conversion Mode” in "CHAPTER: 12-BIT A/D CONVERTER."

3.3. Timing charts

Figure 3-3, Figure 3-4, and Figure 3-5 are timing charts related to the transition to partial wakeup mode.

Figure 3-3 With a Range Deviation Arising after a Transition from PSS Mode to PWU Mode

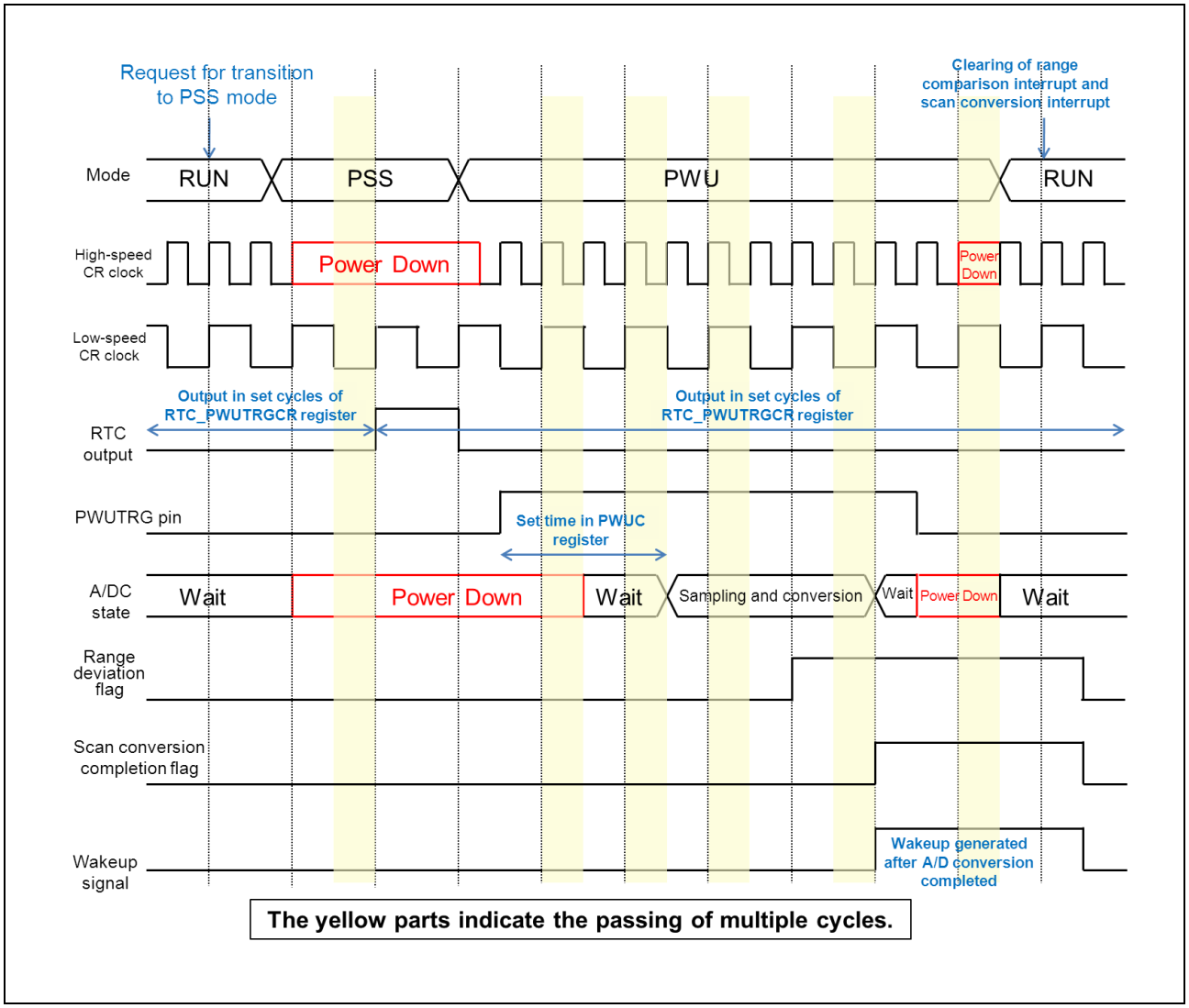


Figure 3-4 Without a Range Deviation Arising after a Transition from PSS Mode to PWU Mode

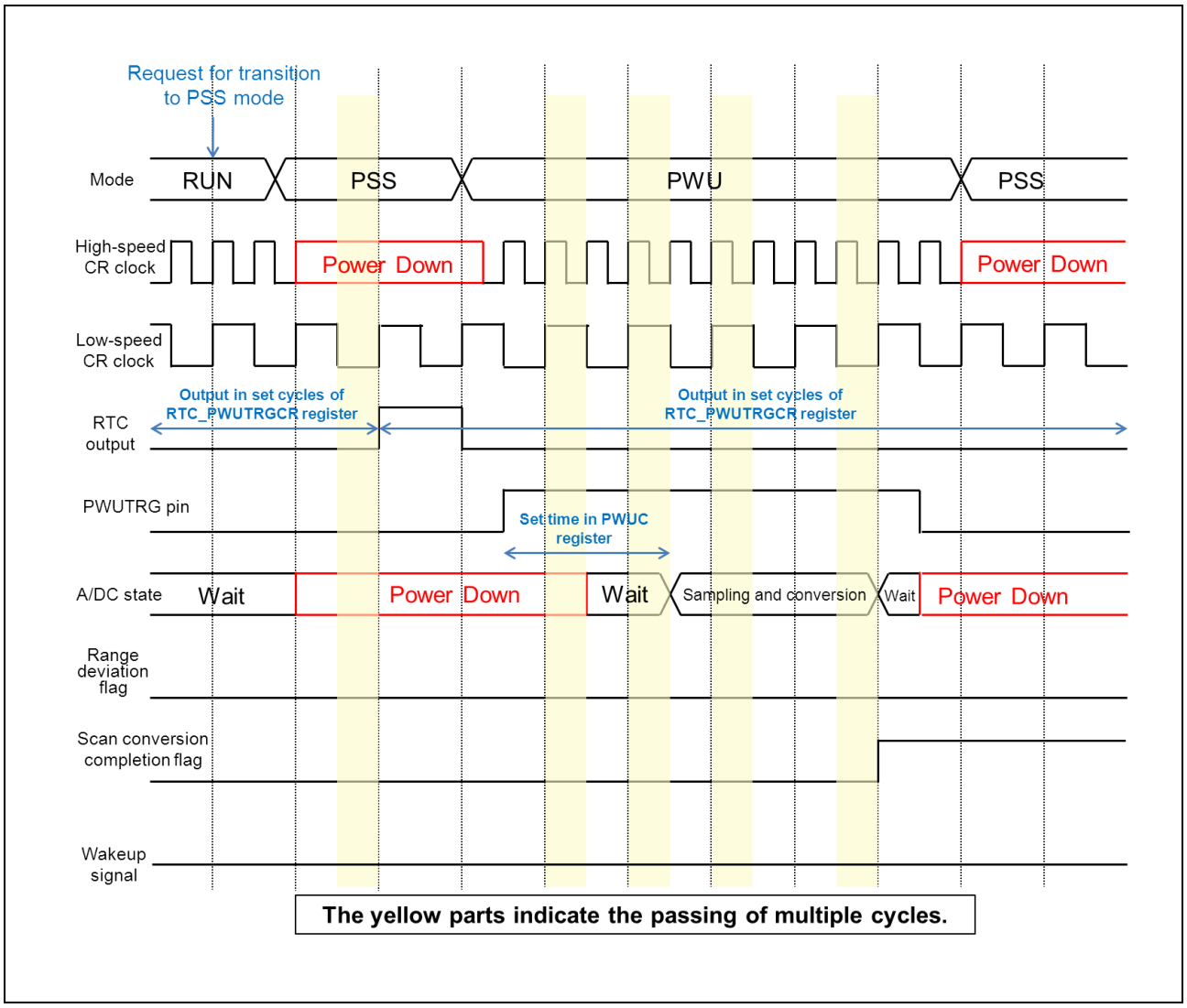
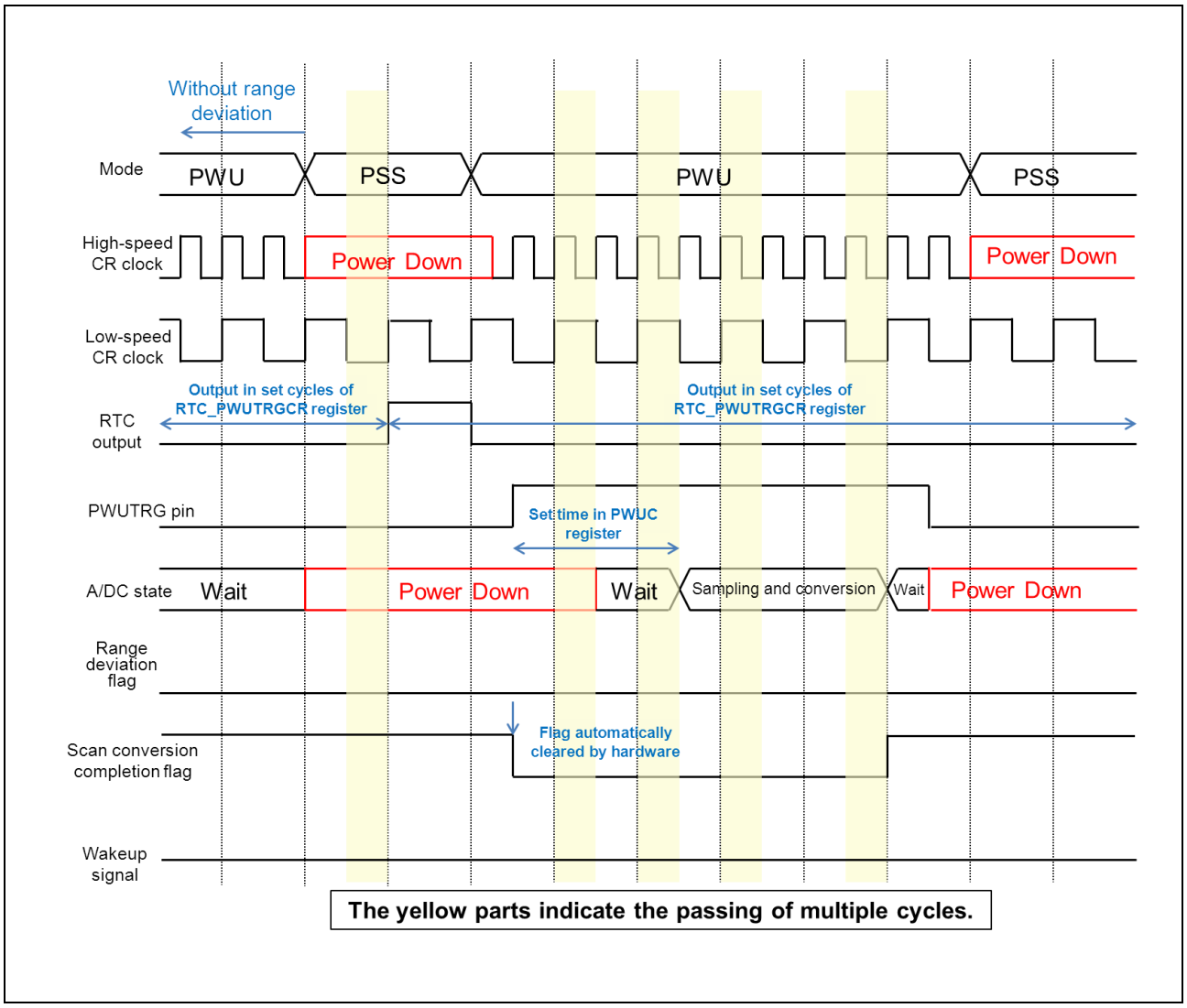


Figure 3-5 Until a Restart of A/D Conversion after Figure 3-4.



Note:

- *The clocks in the figures do not match the actual number of cycles.*

4. Registers

This section explains the partial wakeup control register.

Table 4-1 List of the Partial Wakeup Control Register

Abbreviated Register Name	Register Name	Reference
PWUC	PWU control register	4.1

Table 4-2 Memory Map and Initial Values of the Partial Wakeup Control Register

Offset	+3	+2	+1	+0
0x00000200	Reserved	Reserved	Reserved	PWUC 00000000

The offset value is the value for the "ADC1/PWU" area start address.

The partial wakeup register is deployed after the register of A/D converter unit 1.

4.1. PWU Control Register (PWUC)

The PWU control register (PWUC) is used to enable PWU mode and to set the period between PWUTRG pin output and A/D conversion start.

bit	7	6	5	4	3	2	1	0
Field	PWUE	ADSTS6	ADSTS5	ADSTS4	ADSTS3	ADSTS2	ADSTS1	ADSTS0
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] PWUE: PWU Mode Enable Bit

Bit	Description
0	Disable transition to PWU mode.
1	Enable transition to PWU mode.

- This bit is used to enable transition to PWU mode.
- When this bit is "1", there is a transition from PSS mode to PWU mode.

[bit6:0] ADSTS [6:0]: A/D Conversion Start Time Setting Bits

ADSTS [6:0]	Setting	Value for High-Speed CR Oscillator, 4 MHz
0000000	200 cycles of high-speed CR oscillation cycle	50 us
0000001	400 cycles of high-speed CR oscillation cycle	100 us
0000010	600 cycles of high-speed CR oscillation cycle	150 us
…	…	…
n	(n+1)*200 cycles of high-speed CR oscillation cycle	(n+1)*50 us
…	…	…
1100011	20000 cycles of high-speed CR oscillation cycle	5000 us
1100100	20200 cycles of high-speed CR oscillation cycle	5050 us
1100101	20400 cycles of high-speed CR oscillation cycle	5100 us
110011x	20400 cycles of high-speed CR oscillation cycle	5100 us
1101xxx		
111xxxx		

These bits are used to set the period between PWUTRG pin output and A/D conversion start.

5. Precautions for Using This Device

This section provides notes on using the partial wakeup control function.

Time until Transition to PWU Mode

For a transition from RUN mode to PSS mode, the RTC is not initialized.
For this reason, after the transition from RUN mode to PSS mode, the time until the transition to PWU mode is less than the setting at "(7)."
(The time depends on the RTC count value at the transition to PSS mode.)

Restrictions on A/D Conversion Data

If an interrupt factor other than a range comparison interrupt causes a return to RUN mode during A/D conversion, the A/D conversion results (values recorded in the A/D data register) are not guaranteed.
Examples of this factor include external interrupts and RTC interrupts.

Restrictions on the Range Comparison Function

The range comparison function sets the upper and lower limit thresholds for voltage and judges A/D conversion results.
8 analog input channels can be used in PWU mode.
However, for pairs of the upper limit thresholds and lower limit thresholds that can be set, select from 4 sets.

Restrictions on Port Settings

- Before transitioning to PSS mode, be sure to use the port function to set for output of the PWUTRG function.
- The state of ports at the generation of various resets is "output Hi-Z" and "input disabled."

Restrictions on Software Watchdog Timer Settings

If the settings of the software watchdog timer in PSS mode contradict the contents of the SYSC0_PSSCKSRER register that have been set in "PSS Setting (Processing by Software)" in Section "3.Explanation of Operation," a PSS profile error occurs. It cannot transition to PSS mode.
Be careful when setting the software watchdog timer.

For details on the combination of violation settings, see "Profile" in "CHAPTER: LOW-POWER CONSUMPTION".

For details on the setting procedure examples of the software watchdog timer, see " Example of set procedure " in "CHAPTER: SOFTWARE WATCHDOG TIMER."

Restrictions on the Cycles for transitioning to PWU Mode

The term of 8ms is generated from the following calculation in RTC.

0.25[s] ÷ 32 = 7.8125[ms] (≈ 8[ms])

Moreover as RTC operates with low-speed CR (typical 100KHz) in PWU mode, the actual cycles for transitioning to PWU mode become as the following.

7.81[ms] (The description in this chapter is “8ms”)
15.62[ms] (The description in this chapter is “16ms”)
23.43[ms] (The description in this chapter is “24ms”)
31.24[ms] (The description in this chapter is “32ms”)
39.05[ms] (The description in this chapter is “40ms”)
46.86[ms] (The description in this chapter is “48ms”)
54.67[ms] (The description in this chapter is “56ms”)
62.48[ms] (The description in this chapter is “64ms”)

CHAPTER 62: APPENDIX: Various Settings of I/O Ports

This appendix provides an overview of the I/O ports.

1. Input Level Setting
2. Output Drive Capacity Setting
3. Output Resource Selection
4. Resource Input Selection

SETTINGIO-T01P00R01L01-E1-GBM02

1. Input Level Setting

This section shows the I/O port input level settings.

PIL[1:0]	Description
00	CMOS hysteresis input (0.7 Vcc/0.3 Vcc)
01	Automotive input (0.8 Vcc/0.5 Vcc)
10	Setting prohibited
11	Setting prohibited

For details, see "OVERVIEW: 10. List of Pin Functions" and "OVERVIEW: 11. I/O Circuit Type."

2. Output Drive Capacity Setting

This section shows the I/O port output drive capacity settings.

ODR[1:0]	I_{OL}	I_{OH}
00	2mA	-2mA
01	1mA	-1mA
10	30mA	-30mA
11	Setting prohibited	Setting prohibited

The ports sharing the stepping motor controller output are available as 1mA/2mA/30mA settings.

For details, see "OVERVIEW: 10. List of Pin Functions" and "OVERVIEW: 11. I/O Circuit Type."

While a port is set to the stepping motor controller output, output drive capacity is 30mA regardless of the ODR setting.

3. Output Resource Selection

This section shows the output resource assignments and settings of each pin.

Table 3-1 List of Output Resources

Register (Offset)	Port	Resource Functional Outputs							
		POF=0	POF=1	POF=2	POF=3	POF=4	POF=5	POF=6	POF=7
PPC_PCFGR000 (0x0000)	P000	GPIO_PODR0:POD0	-	-	SOUT2	-	-	-	MAD00
PPC_PCFGR001 (0x0002)	P001	GPIO_PODR0:POD1	-	-	SCS20	-	-	-	MDATA15
PPC_PCFGR003 (0x0006)	P003	GPIO_PODR0:POD3	-	-	SCS22	-	-	-	MDATA14
PPC_PCFGR005 (0x000A)	P005	GPIO_PODR0:POD5	-	-	-	-	-	-	MDATA13
PPC_PCFGR006 (0x000C)	P006	GPIO_PODR0:POD6	-	SOUT3	-	-	-	SDA3	MDATA12
PPC_PCFGR007 (0x000E)	P007	GPIO_PODR0:POD7	-	SCK3	-	-	-	SCL3	MDATA11
PPC_PCFGR008 (0x0010)	P008	GPIO_PODR0:POD8	TIOA0	SCS3	-	-	-	-	MDATA10
PPC_PCFGR009 (0x0012)	P009	GPIO_PODR0:POD9	TIOA1	-	-	-	-	-	MDATA09
PPC_PCFGR010 (0x0014)	P010	GPIO_PODR0:POD10	TIOA2	SOUT11	-	-	-	SDA11	MDATA08
PPC_PCFGR012 (0x0018)	P012	GPIO_PODR0:POD12	TIOA3	SCK11	-	OUT5	-	SCL11	MOEX
PPC_PCFGR013 (0x001A)	P013	GPIO_PODR0:POD13	TIOA4	SCS110	-	OUT6	-	-	MWEX
PPC_PCFGR015 (0x001E)	P015	GPIO_PODR0:POD15	TIOA5	SCS111	-	OUT7	-	-	MCSX0
PPC_PCFGR016 (0x0020)	P016	GPIO_PODR0:POD16	TIOA6	SCS112	-	OUT8	-	-	MCSX1
PPC_PCFGR017 (0x0022)	P017	GPIO_PODR0:POD17	TIOA7	SCS113	-	OUT9	-	-	MDQM0
PPC_PCFGR018 (0x0024)	P018	GPIO_PODR0:POD18	TIOA8	-	-	OUT10	-	-	MDQM1

Register (Offset)	Port	Resource Functional Outputs							
		POF=0	POF=1	POF=2	POF=3	POF=4	POF=5	POF=6	POF=7
PPC_PCFGR019 (0x0026)	P019	GPIO_PODR0:POD19	-	-	-	OUT11	-	-	MAD15
PPC_PCFGR020 (0x0028)	P020	GPIO_PODR0:POD20	-	SOUT0	-	-	-	SDA0	-
PPC_PCFGR021 (0x002A)	P021	GPIO_PODR0:POD21	-	SCK0	SCK4	-	-	SCL0	-
PPC_PCFGR022 (0x002C)	P022	GPIO_PODR0:POD22	-	-	-	-	-	-	-
PPC_PCFGR023 (0x002E)	P023	GPIO_PODR0:POD23	-	SCS0	-	-	-	-	MAD16
PPC_PCFGR024 (0x0030)	P024	GPIO_PODR0:POD24	-	-	SOUT4	-	-	-	MAD17
PPC_PCFGR027 (0x0036)	P027	GPIO_PODR0:POD27	TIOA4	-	SCS42	-	-	-	MAD18
PPC_PCFGR028 (0x0038)	P028	GPIO_PODR0:POD28	-	-	-	OUT0	-	-	MAD19
PPC_PCFGR029 (0x003A)	P029	GPIO_PODR0:POD29	-	SOUT1	-	OUT1	-	-	MAD20
PPC_PCFGR030 (0x003C)	P030	GPIO_PODR0:POD30	-	-	SCS43	OUT2	-	-	MAD21
PPC_PCFGR031 (0x003E)	P031	GPIO_PODR0:POD31	-	SCS1	-	OUT3	-	-	MAD22
PPC_PCFGR100 (0x0040)	P100	GPIO_PODR1:POD00	-	SCK1	-	OUT4	-	-	MAD23
PPC_PCFGR101 (0x0042)	P101	GPIO_PODR1:POD01	-	-	-	OUT5	-	-	MDATA07
PPC_PCFGR103 (0x0046)	P103	GPIO_PODR1:POD03	-	-	-	OUT6	-	-	MDATA06
PPC_PCFGR105 (0x004A)	P105	GPIO_PODR1:POD05	TIOA9	-	-	OUT7	-	-	MDATA04
PPC_PCFGR106 (0x004C)	P106	GPIO_PODR1:POD06	-	-	-	OUT8	TX1	-	MDATA04
PPC_PCFGR107 (0x004E)	P107	GPIO_PODR1:POD07	TIOA10	-	-	OUT9	-	TOT0	MDATA03
PPC_PCFGR108 (0x0050)	P108	GPIO_PODR1:POD08	TIOA11	-	-	OUT10	-	-	-
PPC_PCFGR109 (0x0052)	P109	GPIO_PODR1:POD09	TIOA12	-	-	OUT11	-	TOT1	MCLK
PPC_PCFGR112 (0x0058)	P112	GPIO_PODR1:POD12	TIOA13	-	-	-	-	-	MDATA02
PPC_PCFGR113 (0x005A)	P113	GPIO_PODR1:POD13	TIOA5	-	-	-	-	TOT2	MDATA01
PPC_PCFGR114 (0x005C)	P114	GPIO_PODR1:POD14	TIOA6	-	-	-	-	-	MDATA00
PPC_PCFGR115 (0x005E)	P115	GPIO_PODR1:POD15	-	-	-	-	-	TOT3	-

Register (Offset)	Port	Resource Functional Outputs							
		POF=0	POF=1	POF=2	POF=3	POF=4	POF=5	POF=6	POF=7
PPC_PCFGR117 (0x0062)	P117	GPIO_PODR1:POD17	-	-	-	-	-	-	-
PPC_PCFGR118 (0x0064)	P118	GPIO_PODR1:POD18	-	-	-	-	-	TOT16	-
PPC_PCFGR119 (0x0066)	P119	GPIO_PODR1:POD19	-	SCS90	-	-	-	-	-
PPC_PCFGR120 (0x0068)	P120	GPIO_PODR1:POD20	-	SCS91	-	-	-	TOT17	-
PPC_PCFGR122 (0x006C)	P122	GPIO_PODR1:POD22	TIOA11	SCS92	-	-	-	SGA0	-
PPC_PCFGR123 (0x006E)	P123	GPIO_PODR1:POD23	TIOA12	SCS93	-	-	-	SGO0	-
PPC_PCFGR126 (0x0074)	P126	GPIO_PODR1:POD26	-	-	-	-	-	SGA1	-
PPC_PCFGR127 (0x0076)	P127	GPIO_PODR1:POD27	-	-	-	-	-	SGO1	-
PPC_PCFGR128 (0x0078)	P128	GPIO_PODR1:POD28	-	-	-	-	-	SGA2	-
PPC_PCFGR129 (0x007A)	P129	GPIO_PODR1:POD29	-	-	-	-	-	SGO2	-
PPC_PCFGR130 (0x007C)	P130	GPIO_PODR1:POD30	-	-	-	-	-	-	-
PPC_PCFGR131 (0x007E)	P131	GPIO_PODR1:POD31	-	SOUT9	-	-	-	SDA9	-
PPC_PCFGR202 (0x0084)	P202	GPIO_PODR2:POD02	-	SCK9	-	-	-	SCL9	-
PPC_PCFGR203 (0x0086)	P203	GPIO_PODR2:POD03	-	-	-	-	-	-	-
PPC_PCFGR204 (0x0088)	P204	GPIO_PODR2:POD04	-	-	-	-	-	-	-
PPC_PCFGR205 (0x008A)	P205	GPIO_PODR2:POD05	-	-	-	-	-	-	-
PPC_PCFGR206 (0x008C)	P206	GPIO_PODR2:POD06	-	SCS43	-	-	-	-	-
PPC_PCFGR207 (0x008E)	P207	GPIO_PODR2:POD07	-	SCK4	SCL4	-	-	SPISEL3	-
PPC_PCFGR208 (0x0090)	P208	GPIO_PODR2:POD08	TIOA19	SCS42	-	-	-	SPISEL2	-
PPC_PCFGR209 (0x0092)	P209	GPIO_PODR2:POD09	TIOA20	SOUT4	SDA4	-	-	SPISELL1	-
PPC_PCFGR210 (0x0094)	P210	GPIO_PODR2:POD10	TIOA21	-	-	-	-	SPICLK	-
PPC_PCFGR211 (0x0096)	P211	GPIO_PODR2:POD11	TIOA22	SCS40	-	-	-	SPIDAT0	-
PPC_PCFGR212 (0x0098)	P212	GPIO_PODR2:POD12	TIOA13	SCS41	SCS80	-	-	SPIDAT2	-

Register (Offset)	Port	Resource Functional Outputs							
		POF=0	POF=1	POF=2	POF=3	POF=4	POF=5	POF=6	POF=7
PPC_PCFGR213 (0x009A)	P213	GPIO_PODR2:POD13	TIOA14	-	-	-	-	SPIDAT1	-
PPC_PCFGR214 (0x009C)	P214	GPIO_PODR2:POD14	TIOA15	-	SOUT8	-	-	SPISEL0	-
PPC_PCFGR215 (0x009E)	P215	GPIO_PODR2:POD15	TIOA16	-	SCK8	-	-	SPIDAT3	-
PPC_PCFGR218 (0x00A4)	P218	GPIO_PODR2:POD18	-	-	-	-	-	-	-
PPC_PCFGR219 (0x00A6)	P219	GPIO_PODR2:POD19	-	-	-	-	-	-	-
PPC_PCFGR220 (0x00A8)	P220	GPIO_PODR2:POD20	-	SCS83	-	-	TX2	-	-
PPC_PCFGR222 (0x00AC)	P222	GPIO_PODR2:POD22	-	-	-	-	-	-	-
PPC_PCFGR223 (0x00AE)	P223	GPIO_PODR2:POD23	-	SCS81	-	-	-	-	-
PPC_PCFGR224 (0x00B0)	P224	GPIO_PODR2:POD24	-	SCS80	-	-	TX0	-	-
PPC_PCFGR225 (0x00B2)	P225	GPIO_PODR2:POD25	-	SOUT8	-	-	-	SDA8	-
PPC_PCFGR226 (0x00B4)	P226	GPIO_PODR2:POD26	TIOA17	SCK8	-	-	-	SCL8	-
PPC_PCFGR229 (0x00BA)	P229	GPIO_PODR2:POD29	TIOA25	-	-	OUT0	-	PWM1P0	-
PPC_PCFGR230 (0x00BC)	P230	GPIO_PODR2:POD30	TIOA26	-	-	OUT1	-	PWM1M0	-
PPC_PCFGR231 (0x00BE)	P231	GPIO_PODR2:POD31	TIOA27	-	-	OUT2	-	PWM2P0	-
PPC_PCFGR300 (0x00C0)	P300	GPIO_PODR3:POD00	TIOA28	-	-	OUT3	-	PWM2M0	-
PPC_PCFGR301 (0x00C2)	P301	GPIO_PODR3:POD01	TIOA18	-	-	OUT4	-	PWM1P1	-
PPC_PCFGR302 (0x00C4)	P302	GPIO_PODR3:POD02	TIOA19	-	-	-	-	PWM1M1	-
PPC_PCFGR304 (0x00C8)	P304	GPIO_PODR3:POD04	TIOA20	-	-	-	-	PWM2P1	-
PPC_PCFGR305 (0x00CA)	P305	GPIO_PODR3:POD05	TIOA29	-	-	-	-	PWM2M1	-
PPC_PCFGR307 (0x00CE)	P307	GPIO_PODR3:POD07	-	SCS102	-	-	-	PWM1P2	-
PPC_PCFGR308 (0x00D0)	P308	GPIO_PODR3:POD08	TIOA28	-	-	-	-	PWM1M2	-
PPC_PCFGR309 (0x00D2)	P309	GPIO_PODR3:POD09	TIOA29	-	-	-	-	PWM2P2	-
PPC_PCFGR312 (0x00D8)	P312	GPIO_PODR3:POD12	-	SCS101	-	-	-	PWM2M2	-

Register (Offset)	Port	Resource Functional Outputs							
		POF=0	POF=1	POF=2	POF=3	POF=4	POF=5	POF=6	POF=7
PPC_PCFGR313 (0x00DA)	P313	GPIO_PODR3:POD13	-	SOUT10	SDA10	-	-	PWM1P3	-
PPC_PCFGR314 (0x00DC)	P314	GPIO_PODR3:POD14	TIOA7	SCK10	SCL10	-	-	PWM1M3	-
PPC_PCFGR315 (0x00DE)	P315	GPIO_PODR3:POD15	TIOA8	SCS100	-	-	TX1	PWM2P3	-
PPC_PCFGR317 (0x00E2)	P317	GPIO_PODR3:POD17	TIOA9	-	-	-	-	PWM2M3	-
PPC_PCFGR321 (0x00EA)	P321	-	GPIO_PODR3: POD21	-	-	PWUTRG	-	-	-
PPC_PCFGR322 (0x00EC)	TRST	TRST	GPIO_PODR3: POD22	-	-	-	-	-	-
PPC_PCFGR323 (0x00EE)	TDO	TDO	GPIO_PODR3: POD23	-	-	-	-	-	-
PPC_PCFGR324 (0x00F0)	TDI	TDI	GPIO_PODR3: POD24	-	-	-	-	-	-
PPC_PCFGR327 (0x00F6)	P327	GPIO_PODR3:POD27	-	-	-	-	WOT	-	-
PPC_PCFGR331 (0x00FE)	P331	GPIO_PODR3:POD31	-	-	-	-	-	SGA0	MCSX3
PPC_PCFGR400 (0x0100)	P400	GPIO_PODR4:POD00	-	-	-	-	-	SGO0	MCSX2
PPC_PCFGR401 (0x0102)	P401	GPIO_PODR4:POD01	-	-	-	-	TX1	-	-
PPC_PCFGR402 (0x0104)	P402	-	-	-	-	-	-	-	-
PPC_PCFGR403 (0x0106)	P403	GPIO_PODR4:POD03	-	-	-	TRACEDATA0	-	SGA1	-
PPC_PCFGR404 (0x0108)	P404	GPIO_PODR4:POD04	-	SCS12	-	TRACEDATA1	-	SGO1	MAD14
PPC_PCFGR405 (0x010A)	P405	GPIO_PODR4:POD05	-	-	-	TRACEDATA2	-	SGA2	MAD13
PPC_PCFGR406 (0x010C)	P406	GPIO_PODR4:POD06	-	SOUT12	-	TRACEDATA3	-	SGO2	MAD12
PPC_PCFGR407 (0x010E)	P407	GPIO_PODR4:POD07	-	SCK12	SCK10	TRACEDATA4	-	-	MAD11
PPC_PCFGR408 (0x0110)	P408	GPIO_PODR4:POD08	-	-	-	TRACEDATA5	-	-	MAD10
PPC_PCFGR409 (0x0112)	P409	GPIO_PODR4:POD09	TIOA24	SOUT2	-	TRACEDATA6	-	TOT18	MAD09
PPC_PCFGR411 (0x0116)	P411	GPIO_PODR4:POD11	-	SCK2	SCS101	TRACEDATA7	-	-	MAD08

Register (Offset)	Port	Resource Functional Outputs							
		POF=0	POF=1	POF=2	POF=3	POF=4	POF=5	POF=6	POF=7
PPC_PCFGR413 (0x011A)	P413	GPIO_PODR4:POD13	-	SCS20	SCS103	-	-	TOT19	MAD07
PPC_PCFGR414 (0x011C)	P414	GPIO_PODR4:POD14	-	-	SCS21	-	-	-	MAD06
PPC_PCFGR416 (0x0120)	P416	GPIO_PODR4:POD16	TIOA22	-	-	-	-	TOT32	MAD05
PPC_PCFGR417 (0x0122)	P417	GPIO_PODR4:POD17	TIOA23	-	SOUT10	-	-	-	MAD04
PPC_PCFGR418 (0x0124)	P418	GPIO_PODR4:POD18	-	-	SCS22	-	-	TOT33	MAD03
PPC_PCFGR420 (0x0128)	P420	GPIO_PODR4:POD20	-	-	SCK2	TRACECLK	-	-	MAD02
PPC_PCFGR421 (0x012C)	P421	GPIO_PODR4:POD21	-	-	-	TRACECTL	-	-	MAD01

Note:

- *The hyphen indicates that setting is prohibited.*

4. Resource Input Selection

This section shows the settings of resource input setting registers and the assignments of input resources.

Table 4-1 List of Resource Inputs

Register (Offset)	Resource	RESSEL[3:0] PORTSEL[11:8]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN6 (0x000C)	SIN2	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P408	P421	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN7 (0x000E)	SCK2	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P411	P420	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN8 (0x0010)	SCS2	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P413	P001	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN12 (0x0018)	SIN4	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P210	P023	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN13 (0x001A)	SCK4	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P207	P021	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN15 (0x001E)	SIN8	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P222	P213	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN16 (0x0020)	SCK8	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P226	P215	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] PORTSEL[11:8]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN17 (0x0022)	SCS8	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P224	P212	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN21 (0x002A)	SIN10	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P317	P416	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN22 (0x002C)	SCK10	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P314	P407	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN66 (0x0084)	RX0	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	P225	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN67 (0x0086)	RX1	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P402	P317	P107	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN68 (0x0088)	INT0	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P225	P009	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN70 (0x008C)	INT2	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P402	P107	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN71 (0x008E)	INT3	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P022	P108	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] PORTSEL[11:8]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN72 (0x0090)	INT4	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	P117	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN73 (0x0092)	INT5	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P130	P118	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN74 (0x0094)	INT6	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P210	P202	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN75 (0x0096)	INT7	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P222	P207	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN76 (0x0098)	INT8	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P229	P213	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN77 (0x009A)	INT9	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	P215	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN78 (0x009C)	INT10	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	P313	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN79 (0x009E)	INT11	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P405	P317	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] PORTSEL[11:8]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN80 (0x00A0)	INT12	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	P421	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN81 (0x00A2)	INT13	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	P411	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN82 (0x00A4)	INT14	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P418	P413	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN83 (0x00A6)	INT15	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	P417	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN84 (0x00A8)	TEXT0	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P019	P027	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN85 (0x00AA)	TEXT1	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P020	P127	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN86 (0x00AC)	TEXT2	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P218	P128	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN87 (0x00AE)	TEXT3	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P219	P205	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] PORTSEL[11:8]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN88 (0x00B0)	TEXT4	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P304	P206	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN89 (0x00B2)	TEXT5	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P305	P207	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN90 (0x00B4)	OCU0	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
	OCU1	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN91 (0x00B6)	OCU0_MOD 0	RESSEL (0-7)	set 0	set 1	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN92 (0x00B8)	OCU1_MOD 1	RESSEL (0-7)	set 0	set 1	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN93 (0x00BA)	OCU2	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
	OCU3	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] PORTSEL[11:8]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN94 (0x00BC)	OCU2_MOD 0	RESSEL (0-7)	set 0	set 1	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN95 (0x00BE)	OCU3_MOD 1	RESSEL (0-7)	set 0	set 1	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN96 (0x00C0)	OCU4	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
	OCU5	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN97 (0x00C2)	OCU4_MOD 0	RESSEL (0-7)	set 0	set 1	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN98 (0x00C4)	OCU5_MOD 1	RESSEL (0-7)	set 0	set 1	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN99 (0x00C6)	OCU6	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
	OCU7	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] PORTSEL[11:8]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN100 (0x00C8)	OCU6_MOD 0	RESSEL (0-7)	set 0	set 1	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN101 (0x00CA)	OCU7_MOD 1	RESSEL (0-7)	set 0	set 1	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN102 (0x00CC)	OCU8	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
	OCU9	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN103 (0x00CE)	OCU8_MOD 0	RESSEL (0-7)	set 0	set 1	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN104 (0x00D0)	OCU9_MOD 1	RESSEL (0-7)	set 0	set 1	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN105 (0x00D2)	OCU10	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
	OCU11	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] PORTSEL[11:8]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN106 (0x00D4)	OCU10_MOD0	RESSEL (0-7)	set 0	set 1	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN107 (0x00D6)	OCU11_MOD1	RESSEL (0-7)	set 0	set 1	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN108 (0x00D8)	ICU0	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
	ICU1	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN109 (0x00DA)	IN0	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P401	P308	P210	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN110 (0x00DC)	IN1	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P402	P309	P211	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN111 (0x00DE)	ICU2	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
	ICU3	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] PORTSEL[11:8]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN112 (0x00E0)	IN2	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P403	P312	P212	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN113 (0x00E2)	IN3	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P404	P313	P213	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN114 (0x00E4)	ICU4	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
	ICU5	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN115 (0x00E6)	IN4	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P405	P314	P214	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN116 (0x00E8)	IN5	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P416	P315	P215	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN117 (0x00EA)	ICU6	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
	ICU7	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] PORTSEL[11:8]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN118 (0x00EC)	IN6	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P005	P129	P220	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN119 (0x00EE)	IN7	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P006	P130	P222	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN120 (0x00F0)	ICU8	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
	ICU9	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN121 (0x00F2)	IN8	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P007	P131	P223	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN122 (0x00F4)	IN9	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P008	P202	P224	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN123 (0x00F6)	ICU10	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
	ICU11	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] PORTSEL[11:8]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN124 (0x00F8)	IN10	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P009	P203	P225	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN125 (0x00FA)	IN11	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P010	P204	P226	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN129 (0x0102)	ADC3	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN131 (0x0106)	ADC5	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN132 (0x0108)	ADC6	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN135 (0x010E)	ADC9	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN136 (0x0110)	ADC10	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN138 (0x0114)	ADC12	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] PORTSEL[11:8]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN139 (0x0116)	ADC13	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN140 (0x0118)	ADC14	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN141 (0x011A)	ADC15	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN143 (0x011E)	ADC17	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN144 (0x0120)	ADC18	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN145 (0x0122)	ADC19	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN146 (0x0124)	ADC20	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN147 (0x0126)	ADC21	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] PORTSEL[11:8]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN148 (0x0128)	ADC22	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN149 (0x012A)	ADC23	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN150 (0x012C)	ADC24	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN153 (0x0132)	ADC27	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN154 (0x0134)	ADC28	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN155 (0x0136)	ADC29	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN156 (0x0138)	ADC30	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN157 (0x013A)	ADC31	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] PORTSEL[11:8]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN158 (0x013C)	ADC32	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN159 (0x013E)	ADC33	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN160 (0x0140)	ADC34	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN161 (0x0142)	ADC35	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN162 (0x0144)	ADC36	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN163 (0x0146)	ADC37	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN164 (0x0148)	ADC38	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN165 (0x014A)	ADC39	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] PORTSEL[11:8]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN166 (0x014C)	ADC40	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN167 (0x014E)	ADC41	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN168 (0x0150)	ADC42	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN169 (0x0152)	ADC43	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN172 (0x0158)	ADC46	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN173 (0x015A)	ADC47	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN174 (0x015C)	ADC48	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN175 (0x015E)	ADC49	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] PORTSEL[11:8]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN176 (0x0160)	ADC50	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN177 (0x0162)	ADC51	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN178 (0x0164)	ADC52	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN179 (0x0166)	ADC53	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN181 (0x016A)	ADC55	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN182 (0x016C)	ADC56	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN183 (0x016E)	ADC57	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN184 (0x0170)	ADC58	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] PORTSEL[11:8]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN185 (0x0172)	ADC59	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN186 (0x0174)	ADC60	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN187 (0x0176)	ADC61	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN188 (0x0178)	ADC62	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT5	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN190 (0x017C)	RX2	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P222	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN193 (0x0182)	TIOA5	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P015	P113	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN194 (0x0184)	TIOA7	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P017	P314	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN195 (0x0186)	TIOA9	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P105	P317	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] PORTSEL[11:8]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN196 (0x0188)	TIOA11	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P108	P122	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN197 (0x018A)	TIOA13	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P112	P212	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN200 (0x0190)	TIOA19	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P208	P302	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN202 (0x0194)	TIOA23	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	P417	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN205 (0x019A)	TIOA29	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P305	P309	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN207 (0x019E)	TIN1	RESSEL (0-7)	P108	TOT0	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN208 (0x01A0)	TIN2	RESSEL (0-7)	P112	TOT1	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN209 (0x01A2)	TIN3	RESSEL (0-7)	P114	TOT2	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] PORTSEL[11:8]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN211 (0x01A6)	TIN17	RESSEL (0-7)	P119	TOT16	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN212 (0x01A8)	TIN18	RESSEL (0-7)	P408	TOT17	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN213 (0x01AA)	TIN19	RESSEL (0-7)	P411	TOT18	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN215 (0x01AE)	TIN33	RESSEL (0-7)	P417	TOT32	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN216 (0x01B0)	AIN8	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P000	P203	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN217 (0x01B2)	BIN8	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P001	P204	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN218 (0x01B4)	ZIN8	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P003	P205	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN219 (0x01B6)	AIN9	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P005	P223	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] PORTSEL[11:8]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN220 (0x01B8)	BIN9	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P006	P224	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN221 (0x01BA)	ZIN9	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P007	P225	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN222 | RIC_RESIN227	Reserved	-	-							
RIC_RESIN228 (0x01C8)	ADC3	RESSEL (0-7)	BT00	BT01	BT02	BT03	BT04	BT05	BT06	BT07
		RESSEL (8-15)	BT08	BT09	BT10	BT11	BT12	BT13	BT14	BT15
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	BT00-15	-	-	-	-	-	-	-
RIC_RESIN229 (0x01CA)		RESSEL (0-7)	BT16	BT17	BT18	BT19	BT20	BT21	BT22	BT23
		RESSEL (8-15)	BT24	BT25	BT26	BT27	BT28	BT29	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	BT16-29	-	-	-	-	-	-
RIC_RESIN230 | RIC_RESIN231	Reserved	-	-							
RIC_RESIN232 (0x01D0)	ADC5	RESSEL (0-7)	BT00	BT01	BT02	BT03	BT04	BT05	BT06	BT07
		RESSEL (8-15)	BT08	BT09	BT10	BT11	BT12	BT13	BT14	BT15
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	BT00-15	-	-	-	-	-	-	-
RIC_RESIN233 (0x01D2)		RESSEL (0-7)	BT16	BT17	BT18	BT19	BT20	BT21	BT22	BT23
		RESSEL (8-15)	BT24	BT25	BT26	BT27	BT28	BT29	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	BT16-29	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] PORTSEL[11:8]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN234 (0x01D4)	ADC6	RESSEL (0-7)	BT00	BT01	BT02	BT03	BT04	BT05	BT06	BT07
		RESSEL (8-15)	BT08	BT09	BT10	BT11	BT12	BT13	BT14	BT15
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	BT00-15	-	-	-	-	-	-	-
RIC_RESIN235 (0x01D6)		RESSEL (0-7)	BT16	BT17	BT18	BT19	BT20	BT21	BT22	BT23
		RESSEL (8-15)	BT24	BT25	BT26	BT27	BT28	BT29	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	BT16-29	-	-	-	-	-	-
RIC_RESIN236 | RIC_RESIN239	Reserved	-	-							
RIC_RESIN240 (0x01E0)	ADC9	RESSEL (0-7)	BT00	BT01	BT02	BT03	BT04	BT05	BT06	BT07
		RESSEL (8-15)	BT08	BT09	BT10	BT11	BT12	BT13	BT14	BT15
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	BT00-15	-	-	-	-	-	-	-
RIC_RESIN241 (0x01E2)		RESSEL (0-7)	BT16	BT17	BT18	BT19	BT20	BT21	BT22	BT23
		RESSEL (8-15)	BT24	BT25	BT26	BT27	BT28	BT29	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	BT16-29	-	-	-	-	-	-
RIC_RESIN242 (0x01E4)	ADC10	RESSEL (0-7)	BT00	BT01	BT02	BT03	BT04	BT05	BT06	BT07
		RESSEL (8-15)	BT08	BT09	BT10	BT11	BT12	BT13	BT14	BT15
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	BT00-15	-	-	-	-	-	-	-
RIC_RESIN243 (0x01E6)		RESSEL (0-7)	BT16	BT17	BT18	BT19	BT20	BT21	BT22	BT23
		RESSEL (8-15)	BT24	BT25	BT26	BT27	BT28	BT29	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	BT16-29	-	-	-	-	-	-
RIC_RESIN244 | RIC_RESIN245	Reserved	-	-							

Register (Offset)	Resource	RESSEL[3:0] PORTSEL[11:8]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN246 (0x01EC)	ADC12	RESSEL (0-7)	BT00	BT01	BT02	BT03	BT04	BT05	BT06	BT07
		RESSEL (8-15)	BT08	BT09	BT10	BT11	BT12	BT13	BT14	BT15
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	BT00-15	-	-	-	-	-	-	-
RIC_RESIN247 (0x01EE)		RESSEL (0-7)	BT16	BT17	BT18	BT19	BT20	BT21	BT22	BT23
		RESSEL (8-15)	BT24	BT25	BT26	BT27	BT28	BT29	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	BT16-29	-	-	-	-	-	-
RIC_RESIN248 (0x01F0)	ADC13	RESSEL (0-7)	BT00	BT01	BT02	BT03	BT04	BT05	BT06	BT07
		RESSEL (8-15)	BT08	BT09	BT10	BT11	BT12	BT13	BT14	BT15
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	BT00-15	-	-	-	-	-	-	-
RIC_RESIN249 (0x01F2)		RESSEL (0-7)	BT16	BT17	BT18	BT19	BT20	BT21	BT22	BT23
		RESSEL (8-15)	BT24	BT25	BT26	BT27	BT28	BT29	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	BT16-29	-	-	-	-	-	-
RIC_RESIN250 (0x01F4)	ADC14	RESSEL (0-7)	BT00	BT01	BT02	BT03	BT04	BT05	BT06	BT07
		RESSEL (8-15)	BT08	BT09	BT10	BT11	BT12	BT13	BT14	BT15
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	BT00-15	-	-	-	-	-	-	-
RIC_RESIN251 (0x01F6)		RESSEL (0-7)	BT16	BT17	BT18	BT19	BT20	BT21	BT22	BT23
		RESSEL (8-15)	BT24	BT25	BT26	BT27	BT28	BT29	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	BT16-29	-	-	-	-	-	-
RIC_RESIN252 (0x01F8)	ADC15	RESSEL (0-7)	BT00	BT01	BT02	BT03	BT04	BT05	BT06	BT07
		RESSEL (8-15)	BT08	BT09	BT10	BT11	BT12	BT13	BT14	BT15
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	BT00-15	-	-	-	-	-	-	-
RIC_RESIN253 (0x01FA)		RESSEL (0-7)	BT16	BT17	BT18	BT19	BT20	BT21	BT22	BT23
		RESSEL (8-15)	BT24	BT25	BT26	BT27	BT28	BT29	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	BT16-29	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] PORTSEL[11:8]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN254 | RIC_RESIN255	Reserved	-	-							
RIC_RESIN256 (0x0200)	ADC17	RESSEL (0-7)	BT00	BT01	BT02	BT03	BT04	BT05	BT06	BT07
		RESSEL (8-15)	BT08	BT09	BT10	BT11	BT12	BT13	BT14	BT15
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	BT00-15	-	-	-	-	-	-	-
RIC_RESIN257 (0x0202)		RESSEL (0-7)	BT16	BT17	BT18	BT19	BT20	BT21	BT22	BT23
		RESSEL (8-15)	BT24	BT25	BT26	BT27	BT28	BT29	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	BT16-29	-	-	-	-	-	-
RIC_RESIN258 (0x0204)	ADC18	RESSEL (0-7)	BT00	BT01	BT02	BT03	BT04	BT05	BT06	BT07
		RESSEL (8-15)	BT08	BT09	BT10	BT11	BT12	BT13	BT14	BT15
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	BT00-15	-	-	-	-	-	-	-
RIC_RESIN259 (0x0206)		RESSEL (0-7)	BT16	BT17	BT18	BT19	BT20	BT21	BT22	BT23
		RESSEL (8-15)	BT24	BT25	BT26	BT27	BT28	BT29	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	BT16-29	-	-	-	-	-	-
RIC_RESIN260 (0x0208)	ADC19	RESSEL (0-7)	BT00	BT01	BT02	BT03	BT04	BT05	BT06	BT07
		RESSEL (8-15)	BT08	BT09	BT10	BT11	BT12	BT13	BT14	BT15
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	BT00-15	-	-	-	-	-	-	-
RIC_RESIN261 (0x020A)		RESSEL (0-7)	BT16	BT17	BT18	BT19	BT20	BT21	BT22	BT23
		RESSEL (8-15)	BT24	BT25	BT26	BT27	BT28	BT29	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	BT16-29	-	-	-	-	-	-
RIC_RESIN262 (0x020C)	ADC20	RESSEL (0-7)	BT00	BT01	BT02	BT03	BT04	BT05	BT06	BT07
		RESSEL (8-15)	BT08	BT09	BT10	BT11	BT12	BT13	BT14	BT15
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	BT00-15	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] PORTSEL[11:8]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN263 (0x020E)	ADC20	RESSEL (0-7)	BT16	BT17	BT18	BT19	BT20	BT21	BT22	BT23
		RESSEL (8-15)	BT24	BT25	BT26	BT27	BT28	BT29	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	BT16-29	-	-	-	-	-	-
RIC_RESIN264 (0x0210)	ADC21	RESSEL (0-7)	BT00	BT01	BT02	BT03	BT04	BT05	BT06	BT07
		RESSEL (8-15)	BT08	BT09	BT10	BT11	BT12	BT13	BT14	BT15
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	BT00-15	-	-	-	-	-	-	-
RIC_RESIN265 (0x0212)		RESSEL (0-7)	BT16	BT17	BT18	BT19	BT20	BT21	BT22	BT23
		RESSEL (8-15)	BT24	BT25	BT26	BT27	BT28	BT29	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	BT16-29	-	-	-	-	-	-
RIC_RESIN266 (0x0214)	ADC22	RESSEL (0-7)	BT00	BT01	BT02	BT03	BT04	BT05	BT06	BT07
		RESSEL (8-15)	BT08	BT09	BT10	BT11	BT12	BT13	BT14	BT15
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	BT00-15	-	-	-	-	-	-	-
RIC_RESIN267 (0x0216)		RESSEL (0-7)	BT16	BT17	BT18	BT19	BT20	BT21	BT22	BT23
		RESSEL (8-15)	BT24	BT25	BT26	BT27	BT28	BT29	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	BT16-29	-	-	-	-	-	-
RIC_RESIN268 (0x0218)	ADC23	RESSEL (0-7)	BT00	BT01	BT02	BT03	BT04	BT05	BT06	BT07
		RESSEL (8-15)	BT08	BT09	BT10	BT11	BT12	BT13	BT14	BT15
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	BT00-15	-	-	-	-	-	-	-
RIC_RESIN269 (0x021A)		RESSEL (0-7)	BT16	BT17	BT18	BT19	BT20	BT21	BT22	BT23
		RESSEL (8-15)	BT24	BT25	BT26	BT27	BT28	BT29	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	BT16-29	-	-	-	-	-	-
RIC_RESIN270 (0x021C)	ADC24	RESSEL (0-7)	BT00	BT01	BT02	BT03	BT04	BT05	BT06	BT07
		RESSEL (8-15)	BT08	BT09	BT10	BT11	BT12	BT13	BT14	BT15
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	BT00-15	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] PORTSEL[11:8]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN271 (0x021E)	ADC24	RESSEL (0-7)	BT16	BT17	BT18	BT19	BT20	BT21	BT22	BT23
		RESSEL (8-15)	BT24	BT25	BT26	BT27	BT28	BT29	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	BT16-29	-	-	-	-	-	-
RIC_RESIN272 | RIC_RESIN275	Reserved	-	-							
RIC_RESIN276 (0x0228)	ADC27	RESSEL (0-7)	BT00	BT01	BT02	BT03	BT04	BT05	BT06	BT07
		RESSEL (8-15)	BT08	BT09	BT10	BT11	BT12	BT13	BT14	BT15
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	BT00-15	-	-	-	-	-	-	-
RIC_RESIN277 (0x022A)		RESSEL (0-7)	BT16	BT17	BT18	BT19	BT20	BT21	BT22	BT23
		RESSEL (8-15)	BT24	BT25	BT26	BT27	BT28	BT29	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	BT16-29	-	-	-	-	-	-
RIC_RESIN278 (0x022C)	ADC28	RESSEL (0-7)	BT00	BT01	BT02	BT03	BT04	BT05	BT06	BT07
		RESSEL (8-15)	BT08	BT09	BT10	BT11	BT12	BT13	BT14	BT15
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	BT00-15	-	-	-	-	-	-	-
RIC_RESIN279 (0x022E)		RESSEL (0-7)	BT16	BT17	BT18	BT19	BT20	BT21	BT22	BT23
		RESSEL (8-15)	BT24	BT25	BT26	BT27	BT28	BT29	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	BT16-29	-	-	-	-	-	-
RIC_RESIN280 (0x0230)	ADC29	RESSEL (0-7)	BT00	BT01	BT02	BT03	BT04	BT05	BT06	BT07
		RESSEL (8-15)	BT08	BT09	BT10	BT11	BT12	BT13	BT14	BT15
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	BT00-15	-	-	-	-	-	-	-
RIC_RESIN281 (0x0232)		RESSEL (0-7)	BT16	BT17	BT18	BT19	BT20	BT21	BT22	BT23
		RESSEL (8-15)	BT24	BT25	BT26	BT27	BT28	BT29	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	BT16-29	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] PORTSEL[11:8]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN282 (0x0234)	ADC30	RESSEL (0-7)	BT00	BT01	BT02	BT03	BT04	BT05	BT06	BT07
		RESSEL (8-15)	BT08	BT09	BT10	BT11	BT12	BT13	BT14	BT15
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	BT00-15	-	-	-	-	-	-	-
RIC_RESIN283 (0x0236)		RESSEL (0-7)	BT16	BT17	BT18	BT19	BT20	BT21	BT22	BT23
		RESSEL (8-15)	BT24	BT25	BT26	BT27	BT28	BT29	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	BT16-29	-	-	-	-	-	-
RIC_RESIN284 (0x0238)	ADC31	RESSEL (0-7)	BT00	BT01	BT02	BT03	BT04	BT05	BT06	BT07
		RESSEL (8-15)	BT08	BT09	BT10	BT11	BT12	BT13	BT14	BT15
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	BT00-15	-	-	-	-	-	-	-
RIC_RESIN285 (0x023A)		RESSEL (0-7)	BT16	BT17	BT18	BT19	BT20	BT21	BT22	BT23
		RESSEL (8-15)	BT24	BT25	BT26	BT27	BT28	BT29	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	BT16-29	-	-	-	-	-	-
RIC_RESIN286 (0x023C)	ADC32	RESSEL (0-7)	BT00	BT01	BT02	BT03	BT04	BT05	BT06	BT07
		RESSEL (8-15)	BT08	BT09	BT10	BT11	BT12	BT13	BT14	BT15
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	BT00-15	-	-	-	-	-	-	-
RIC_RESIN287 (0x023E)		RESSEL (0-7)	BT16	BT17	BT18	BT19	BT20	BT21	BT22	BT23
		RESSEL (8-15)	BT24	BT25	BT26	BT27	BT28	BT29	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	BT16-29	-	-	-	-	-	-
RIC_RESIN288 (0x0240)	ADC33	RESSEL (0-7)	BT00	BT01	BT02	BT03	BT04	BT05	BT06	BT07
		RESSEL (8-15)	BT08	BT09	BT10	BT11	BT12	BT13	BT14	BT15
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	BT00-15	-	-	-	-	-	-	-
RIC_RESIN289 (0x0242)		RESSEL (0-7)	BT16	BT17	BT18	BT19	BT20	BT21	BT22	BT23
		RESSEL (8-15)	BT24	BT25	BT26	BT27	BT28	BT29	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	BT16-29	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] PORTSEL[11:8]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN290 (0x0244)	ADC34	RESSEL (0-7)	BT00	BT01	BT02	BT03	BT04	BT05	BT06	BT07
		RESSEL (8-15)	BT08	BT09	BT10	BT11	BT12	BT13	BT14	BT15
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	BT00-15	-	-	-	-	-	-	-
RIC_RESIN291 (0x0246)		RESSEL (0-7)	BT16	BT17	BT18	BT19	BT20	BT21	BT22	BT23
		RESSEL (8-15)	BT24	BT25	BT26	BT27	BT28	BT29	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	BT16-29	-	-	-	-	-	-
RIC_RESIN292 (0x0248)	ADC35	RESSEL (0-7)	BT00	BT01	BT02	BT03	BT04	BT05	BT06	BT07
		RESSEL (8-15)	BT08	BT09	BT10	BT11	BT12	BT13	BT14	BT15
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	BT00-15	-	-	-	-	-	-	-
RIC_RESIN293 (0x024A)		RESSEL (0-7)	BT16	BT17	BT18	BT19	BT20	BT21	BT22	BT23
		RESSEL (8-15)	BT24	BT25	BT26	BT27	BT28	BT29	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	BT16-29	-	-	-	-	-	-
RIC_RESIN294 (0x024C)	ADC36	RESSEL (0-7)	BT00	BT01	BT02	BT03	BT04	BT05	BT06	BT07
		RESSEL (8-15)	BT08	BT09	BT10	BT11	BT12	BT13	BT14	BT15
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	BT00-15	-	-	-	-	-	-	-
RIC_RESIN295 (0x024E)		RESSEL (0-7)	BT16	BT17	BT18	BT19	BT20	BT21	BT22	BT23
		RESSEL (8-15)	BT24	BT25	BT26	BT27	BT28	BT29	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	BT16-29	-	-	-	-	-	-
RIC_RESIN296 (0x0250)	ADC37	RESSEL (0-7)	BT00	BT01	BT02	BT03	BT04	BT05	BT06	BT07
		RESSEL (8-15)	BT08	BT09	BT10	BT11	BT12	BT13	BT14	BT15
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	BT00-15	-	-	-	-	-	-	-
RIC_RESIN297 (0x0252)		RESSEL (0-7)	BT16	BT17	BT18	BT19	BT20	BT21	BT22	BT23
		RESSEL (8-15)	BT24	BT25	BT26	BT27	BT28	BT29	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	BT16-29	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] PORTSEL[11:8]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN298 (0x0254)	ADC38	RESSEL (0-7)	BT00	BT01	BT02	BT03	BT04	BT05	BT06	BT07
		RESSEL (8-15)	BT08	BT09	BT10	BT11	BT12	BT13	BT14	BT15
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	BT00-15	-	-	-	-	-	-	-
RIC_RESIN299 (0x0256)		RESSEL (0-7)	BT16	BT17	BT18	BT19	BT20	BT21	BT22	BT23
		RESSEL (8-15)	BT24	BT25	BT26	BT27	BT28	BT29	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	BT16-29	-	-	-	-	-	-
RIC_RESIN300 (0x0258)	ADC39	RESSEL (0-7)	BT00	BT01	BT02	BT03	BT04	BT05	BT06	BT07
		RESSEL (8-15)	BT08	BT09	BT10	BT11	BT12	BT13	BT14	BT15
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	BT00-15	-	-	-	-	-	-	-
RIC_RESIN301 (0x025A)		RESSEL (0-7)	BT16	BT17	BT18	BT19	BT20	BT21	BT22	BT23
		RESSEL (8-15)	BT24	BT25	BT26	BT27	BT28	BT29	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	BT16-29	-	-	-	-	-	-
RIC_RESIN302 (0x025C)	ADC40	RESSEL (0-7)	BT00	BT01	BT02	BT03	BT04	BT05	BT06	BT07
		RESSEL (8-15)	BT08	BT09	BT10	BT11	BT12	BT13	BT14	BT15
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	BT00-15	-	-	-	-	-	-	-
RIC_RESIN303 (0x025E)		RESSEL (0-7)	BT16	BT17	BT18	BT19	BT20	BT21	BT22	BT23
		RESSEL (8-15)	BT24	BT25	BT26	BT27	BT28	BT29	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	BT16-29	-	-	-	-	-	-
RIC_RESIN304 (0x0260)	ADC41	RESSEL (0-7)	BT00	BT01	BT02	BT03	BT04	BT05	BT06	BT07
		RESSEL (8-15)	BT08	BT09	BT10	BT11	BT12	BT13	BT14	BT15
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	BT00-15	-	-	-	-	-	-	-
RIC_RESIN305 (0x0262)		RESSEL (0-7)	BT16	BT17	BT18	BT19	BT20	BT21	BT22	BT23
		RESSEL (8-15)	BT24	BT25	BT26	BT27	BT28	BT29	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	BT16-29	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] PORTSEL[11:8]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN306 (0x0264)	ADC42	RESSEL (0-7)	BT00	BT01	BT02	BT03	BT04	BT05	BT06	BT07
		RESSEL (8-15)	BT08	BT09	BT10	BT11	BT12	BT13	BT14	BT15
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	BT00-15	-	-	-	-	-	-	-
RIC_RESIN307 (0x0266)		RESSEL (0-7)	BT16	BT17	BT18	BT19	BT20	BT21	BT22	BT23
		RESSEL (8-15)	BT24	BT25	BT26	BT27	BT28	BT29	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	BT16-29	-	-	-	-	-	-
RIC_RESIN308 (0x0268)	ADC43	RESSEL (0-7)	BT00	BT01	BT02	BT03	BT04	BT05	BT06	BT07
		RESSEL (8-15)	BT08	BT09	BT10	BT11	BT12	BT13	BT14	BT15
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	BT00-15	-	-	-	-	-	-	-
RIC_RESIN309 (0x026A)		RESSEL (0-7)	BT16	BT17	BT18	BT19	BT20	BT21	BT22	BT23
		RESSEL (8-15)	BT24	BT25	BT26	BT27	BT28	BT29	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	BT16-29	-	-	-	-	-	-
RIC_RESIN310 | RIC_RESIN313	Reserved	-	-							
RIC_RESIN314 (0x0274)	ADC46	RESSEL (0-7)	BT00	BT01	BT02	BT03	BT04	BT05	BT06	BT07
		RESSEL (8-15)	BT08	BT09	BT10	BT11	BT12	BT13	BT14	BT15
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	BT00-15	-	-	-	-	-	-	-
RIC_RESIN315 (0x0276)		RESSEL (0-7)	BT16	BT17	BT18	BT19	BT20	BT21	BT22	BT23
		RESSEL (8-15)	BT24	BT25	BT26	BT27	BT28	BT29	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	BT16-29	-	-	-	-	-	-
RIC_RESIN316 (0x0278)	ADC47	RESSEL (0-7)	BT00	BT01	BT02	BT03	BT04	BT05	BT06	BT07
		RESSEL (8-15)	BT08	BT09	BT10	BT11	BT12	BT13	BT14	BT15
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	BT00-15	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] PORTSEL[11:8]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN317 (0x027A)	ADC47	RESSEL (0-7)	BT16	BT17	BT18	BT19	BT20	BT21	BT22	BT23
		RESSEL (8-15)	BT24	BT25	BT26	BT27	BT28	BT29	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	BT16-29	-	-	-	-	-	-
RIC_RESIN318 (0x027C)	ADC48	RESSEL (0-7)	BT00	BT01	BT02	BT03	BT04	BT05	BT06	BT07
		RESSEL (8-15)	BT08	BT09	BT10	BT11	BT12	BT13	BT14	BT15
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	BT00-15	-	-	-	-	-	-	-
RIC_RESIN319 (0x027E)		RESSEL (0-7)	BT16	BT17	BT18	BT19	BT20	BT21	BT22	BT23
		RESSEL (8-15)	BT24	BT25	BT26	BT27	BT28	BT29	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	BT16-29	-	-	-	-	-	-
RIC_RESIN320 (0x0280)	ADC49	RESSEL (0-7)	BT00	BT01	BT02	BT03	BT04	BT05	BT06	BT07
		RESSEL (8-15)	BT08	BT09	BT10	BT11	BT12	BT13	BT14	BT15
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	BT00-15	-	-	-	-	-	-	-
RIC_RESIN321 (0x0282)		RESSEL (0-7)	BT16	BT17	BT18	BT19	BT20	BT21	BT22	BT23
		RESSEL (8-15)	BT24	BT25	BT26	BT27	BT28	BT29	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	BT16-29	-	-	-	-	-	-
RIC_RESIN322 (0x0284)	ADC50	RESSEL (0-7)	BT00	BT01	BT02	BT03	BT04	BT05	BT06	BT07
		RESSEL (8-15)	BT08	BT09	BT10	BT11	BT12	BT13	BT14	BT15
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	BT00-15	-	-	-	-	-	-	-
RIC_RESIN323 (0x0286)		RESSEL (0-7)	BT16	BT17	BT18	BT19	BT20	BT21	BT22	BT23
		RESSEL (8-15)	BT24	BT25	BT26	BT27	BT28	BT29	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	BT16-29	-	-	-	-	-	-
RIC_RESIN324 (0x0288)	ADC51	RESSEL (0-7)	BT00	BT01	BT02	BT03	BT04	BT05	BT06	BT07
		RESSEL (8-15)	BT08	BT09	BT10	BT11	BT12	BT13	BT14	BT15
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	BT00-15	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] PORTSEL[11:8]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN325 (0x028A)	ADC51	RESSEL (0-7)	BT16	BT17	BT18	BT19	BT20	BT21	BT22	BT23
		RESSEL (8-15)	BT24	BT25	BT26	BT27	BT28	BT29	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	BT16-29	-	-	-	-	-	-
RIC_RESIN326 (0x028C)	ADC52	RESSEL (0-7)	BT00	BT01	BT02	BT03	BT04	BT05	BT06	BT07
		RESSEL (8-15)	BT08	BT09	BT10	BT11	BT12	BT13	BT14	BT15
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	BT00-15	-	-	-	-	-	-	-
RIC_RESIN327 (0x028E)		RESSEL (0-7)	BT16	BT17	BT18	BT19	BT20	BT21	BT22	BT23
		RESSEL (8-15)	BT24	BT25	BT26	BT27	BT28	BT29	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	BT16-29	-	-	-	-	-	-
RIC_RESIN328 (0x0290)	ADC53	RESSEL (0-7)	BT00	BT01	BT02	BT03	BT04	BT05	BT06	BT07
		RESSEL (8-15)	BT08	BT09	BT10	BT11	BT12	BT13	BT14	BT15
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	BT00-15	-	-	-	-	-	-	-
RIC_RESIN329 (0x0292)		RESSEL (0-7)	BT16	BT17	BT18	BT19	BT20	BT21	BT22	BT23
		RESSEL (8-15)	BT24	BT25	BT26	BT27	BT28	BT29	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	BT16-29	-	-	-	-	-	-
RIC_RESIN330 | RIC_RESIN331	Reserved	-	-							
RIC_RESIN332 (0x0298)	ADC55	RESSEL (0-7)	BT00	BT01	BT02	BT03	BT04	BT05	BT06	BT07
		RESSEL (8-15)	BT08	BT09	BT10	BT11	BT12	BT13	BT14	BT15
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	BT00-15	-	-	-	-	-	-	-
RIC_RESIN333 (0x029A)		RESSEL (0-7)	BT16	BT17	BT18	BT19	BT20	BT21	BT22	BT23
		RESSEL (8-15)	BT24	BT25	BT26	BT27	BT28	BT29	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	BT16-29	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] PORTSEL[11:8]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN334 (0x029C)	ADC56	RESSEL (0-7)	BT00	BT01	BT02	BT03	BT04	BT05	BT06	BT07
		RESSEL (8-15)	BT08	BT09	BT10	BT11	BT12	BT13	BT14	BT15
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	BT00-15	-	-	-	-	-	-	-
RIC_RESIN335 (0x029E)		RESSEL (0-7)	BT16	BT17	BT18	BT19	BT20	BT21	BT22	BT23
		RESSEL (8-15)	BT24	BT25	BT26	BT27	BT28	BT29	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	BT16-29	-	-	-	-	-	-
RIC_RESIN336 (0x02A0)	ADC57	RESSEL (0-7)	BT00	BT01	BT02	BT03	BT04	BT05	BT06	BT07
		RESSEL (8-15)	BT08	BT09	BT10	BT11	BT12	BT13	BT14	BT15
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	BT00-15	-	-	-	-	-	-	-
RIC_RESIN337 (0x02A2)		RESSEL (0-7)	BT16	BT17	BT18	BT19	BT20	BT21	BT22	BT23
		RESSEL (8-15)	BT24	BT25	BT26	BT27	BT28	BT29	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	BT16-29	-	-	-	-	-	-
RIC_RESIN338 (0x02A4)	ADC58	RESSEL (0-7)	BT00	BT01	BT02	BT03	BT04	BT05	BT06	BT07
		RESSEL (8-15)	BT08	BT09	BT10	BT11	BT12	BT13	BT14	BT15
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	BT00-15	-	-	-	-	-	-	-
RIC_RESIN339 (0x02A6)		RESSEL (0-7)	BT16	BT17	BT18	BT19	BT20	BT21	BT22	BT23
		RESSEL (8-15)	BT24	BT25	BT26	BT27	BT28	BT29	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	BT16-29	-	-	-	-	-	-
RIC_RESIN340 (0x02A8)	ADC59	RESSEL (0-7)	BT00	BT01	BT02	BT03	BT04	BT05	BT06	BT07
		RESSEL (8-15)	BT08	BT09	BT10	BT11	BT12	BT13	BT14	BT15
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	BT00-15	-	-	-	-	-	-	-
RIC_RESIN341 (0x02AA)		RESSEL (0-7)	BT16	BT17	BT18	BT19	BT20	BT21	BT22	BT23
		RESSEL (8-15)	BT24	BT25	BT26	BT27	BT28	BT29	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	BT16-29	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] PORTSEL[11:8]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN342 (0x02AC)	ADC60	RESSEL (0-7)	BT00	BT01	BT02	BT03	BT04	BT05	BT06	BT07
		RESSEL (8-15)	BT08	BT09	BT10	BT11	BT12	BT13	BT14	BT15
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	BT00-15	-	-	-	-	-	-	-
RIC_RESIN343 (0x02AE)		RESSEL (0-7)	BT16	BT17	BT18	BT19	BT20	BT21	BT22	BT23
		RESSEL (8-15)	BT24	BT25	BT26	BT27	BT28	BT29	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	BT16-29	-	-	-	-	-	-
RIC_RESIN344 (0x02B0)	ADC61	RESSEL (0-7)	BT00	BT01	BT02	BT03	BT04	BT05	BT06	BT07
		RESSEL (8-15)	BT08	BT09	BT10	BT11	BT12	BT13	BT14	BT15
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	BT00-15	-	-	-	-	-	-	-
RIC_RESIN345 (0x02B2)		RESSEL (0-7)	BT16	BT17	BT18	BT19	BT20	BT21	BT22	BT23
		RESSEL (8-15)	BT24	BT25	BT26	BT27	BT28	BT29	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	BT16-29	-	-	-	-	-	-
RIC_RESIN346 (0x02B4)	ADC62	RESSEL (0-7)	BT00	BT01	BT02	BT03	BT04	BT05	BT06	BT07
		RESSEL (8-15)	BT08	BT09	BT10	BT11	BT12	BT13	BT14	BT15
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	BT00-15	-	-	-	-	-	-	-
RIC_RESIN347 (0x02B6)		RESSEL (0-7)	BT16	BT17	BT18	BT19	BT20	BT21	BT22	BT23
		RESSEL (8-15)	BT24	BT25	BT26	BT27	BT28	BT29	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	BT16-29	-	-	-	-	-	-
RIC_RESIN348 | RIC_RESIN349	Reserved	-	-							
RIC_RESIN350 (0x02BC)	SCL0	RESSEL (0-7)	noise filter disable	noise filter disable	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] PORTSEL[11:8]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN351 (0x02BE)	SDA0	RESSEL (0-7)	noise filter disable	noise filter disable	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN356 (0x02C8)	SCL3	RESSEL (0-7)	noise filter disable	noise filter disable	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN357 (0x02CA)	SDA3	RESSEL (0-7)	noise filter disable	noise filter disable	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN358 (0x02CC)	SCL4	RESSEL (0-7)	noise filter disable	noise filter disable	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN359 (0x02CE)	SDA4	RESSEL (0-7)	noise filter disable	noise filter disable	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN360 (0x02D0)	SCL8	RESSEL (0-7)	noise filter disable	noise filter disable	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN361 (0x02D2)	SDA8	RESSEL (0-7)	noise filter disable	noise filter disable	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN362 (0x02D4)	SCL9	RESSEL (0-7)	noise filter disable	noise filter disable	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] PORTSEL[11:8]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN363 (0x02D6)	SDA9	RESSEL (0-7)	noise filter disable	noise filter disable	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN364 (0x02D8)	SCL10	RESSEL (0-7)	noise filter disable	noise filter disable	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN365 (0x02DA)	SDA10	RESSEL (0-7)	noise filter disable	noise filter disable	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN366 (0x02DC)	SCL11	RESSEL (0-7)	noise filter disable	noise filter disable	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN367 (0x02DE)	SDA11	RESSEL (0-7)	noise filter disable	noise filter disable	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Note:

- *The hyphen indicates that setting is prohibited except for the initial RESSEL (0) and PORTSEL (0) values.*

CHAPTER 63: APPENDIX: State Transition

This appendix explains State transition.

1. Overview
2. Diagram of State transition
3. Fetching the Operation Mode
4. Changes to PSS and RUN

STATET-T01P00R01L01-E1-GBM02

1. Overview

This section gives a brief overview of State transition

Refer to the Low-power chapter for the detailed information for performing a change state.

2. Diagram of State Transition

This section shows diagram of state transitions.

The device state transitions for this series are shown below.

Figure 2-1 Diagram of Device State Transitions

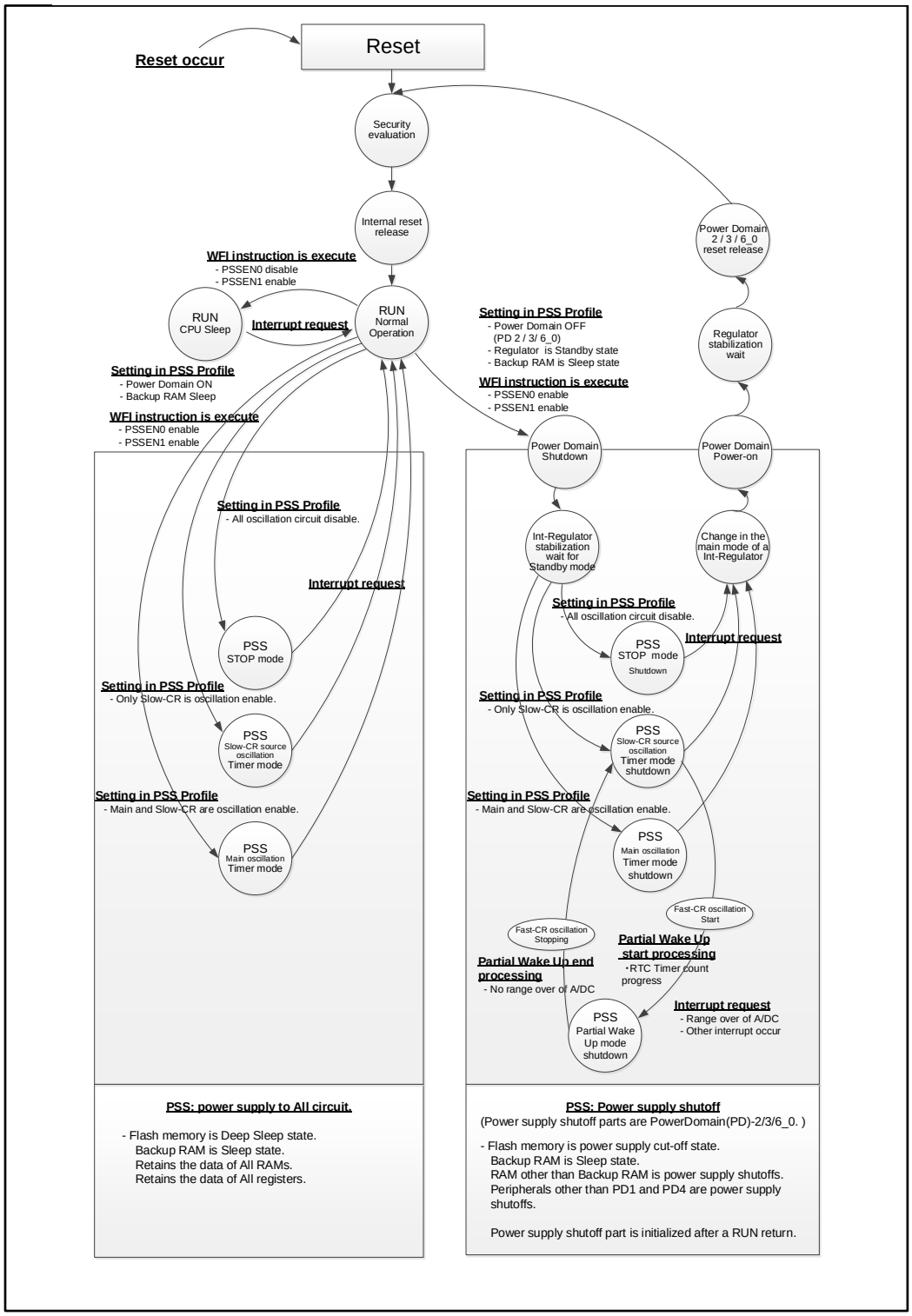


Table 2-1 RUN/PSS State Definitions

S6J312xHzC* supports only the below defined operating state.

Internal state and a frequency definition

		RUN Normal Operation 128MHz S6J312xHzC*	RUN CPU Sleep 128MHz S6J312xHzC*	PSS Slow-CR source oscillation Timer mode	PSS Main oscillation Timer mode	PSS STOP mode	PSS PWU mode Shutdown	PSS Slow-CR source oscillation Timer mode shutdown	PSS Main oscillation Timer mode shutdown	PSS STOP mode Shutdown
Clocks	CLK_CPU	128MHz	<-	Prohibition of an oscillation						
	CLK_FCLK	64MHz								
	CLK_ATB	64MHz								
	CLK_DBG	64MHz								
	CLK_HPM	32MHz								
	CLK_HPM2	16MHz								
	CLK_DMA	32MHz								
	CLK_MEMC	32MHz								
	CLK_EXTBUS	25MHz								
	CLK_SYSC1	32MHz								
	CLK_TRC	64MHz								
	CLK_HSSPI	32MHz								
	CLK_SYSC0H	32MHz								
	CLK_COMH	32MHz								
	CLK_RAM0H	32MHz								
	CLK_RAM1H	32MHz								
	CLK_SYSC0P	32MHz								
	CLK_COMP	32MHz								
	CLK_HAPP0A0	32MHz								
	CLK_HAPP0A1	32MHz								
	CLK_HAPP1B0	32MHz								
	CLK_HAPP1B1	32MHz								
	CLK_LLPBM	128MHz								
	CLK_LLPBM2	64MHz								
	CLK_LCP	64MHz								
	CLK_LCP0	32MHz								
	CLK_LCP0A	32MHz								
	CLK_LCP1	32MHz								
	CLK_LCP1A	32MHz								
	CLK_LAPP0	32MHz								
	CLK_LAPP0A	32MHz								
	CLK_LAPP1	32MHz								
	CLK_LAPP1A	32MHz								
oscillation state of a source clock	Slow-CR	100KHz	<-	<-	<-	Prohibition of an oscillation	100KHz	100KHz	<-	Prohibition of an oscillation
	Fast-CR	4MHz	<-	<-			Prohibition of an oscillation (Fast-CR is unrelated to the PSS profile to oscillate.)	Prohibition of an oscillation		
	Main oscillator	4MHz	<-	Prohibition of an oscillation	4MHz	Prohibition of an oscillation			4MHz	Prohibition of an oscillation
	PLL	When a source clock is not supplied by SSCG: 512MHz (When a source clock is supplied by SSCG: 480MHz)	<-	Prohibition of an oscillation						
	SSCG_PLL	When a source clock is not supplied by SSCG: disable (When a source clock is supplied by SSCG: 512MHz)	<-							

The macro operating state in each mode, and the state of the power domain.

		RUN Normal Operation 128MHz S6J312xHzC*	RUN CPU Sleep 128MHz S6J312xHzC*	PSS Slow-CR source oscillation Timer mode	PSS Main oscillation Timer mode	PSS STOP mode	PSS PWU mode Shutdown	PSS Slow-CR source oscillation Timer mode shutdown	PSS Main oscillation Timer mode shutdown	PSS STOP mode Shutdown
CPU(PD3)	-	enable	disable	disable	disable	disable	disable (Power down)	disable (Power down)	disable (Power down)	disable (Power down)
FLASH(PD3)	-	enable	disable	disable (Deep Sleep)	disable (Deep Sleep)	disable (Deep Sleep)	disable (Power down)	disable (Power down)	disable (Power down)	disable (Power down)
Backup RAM(PD4)	-	enable	disable	disable (Sleep)	disable (Sleep)	disable (Sleep)	disable (Sleep)	disable (Sleep)	disable (Sleep)	disable (Sleep)
Int-Regulator	-	Main mode	Main mode	Main mode	Main mode	Main mode	Standby mode	Standby mode	Standby mode	Standby mode
Power Domain	PD2/PD3	ON	ON	ON	ON	ON	OFF	OFF	OFF	OFF
	PD4	ON	ON	ON	ON	ON	ON	ON	ON	ON
	PD6	ON	ON	ON	ON	ON	OFF	OFF	OFF	OFF

The conditions for changing in each state.

		RUN Normal Operation 128MHz S6J312xHzC*	RUN CPU Sleep 128MHz S6J312xHzC*	PSS Slow-CR source oscillation Timer mode	PSS Main oscillation Timer mode	PSS STOP mode	PSS PWU mode Shutdown	PSS Slow-CR source oscillation Timer mode shutdown	PSS Main oscillation Timer mode shutdown	PSS STOP mode Shutdown
Setup for changes	PSSEN0	-	disable	enable	enable	enable	enable	enable	enable	enable
	PSSEN1	-	enable	enable	enable	enable	enable	enable	enable	enable
Changes start command		-	Execution of a WFI command.							

*x: A/9 、z: A/B

3. Fetching the Operation Mode

This section describes the Fetching the Operation Mode.

The operation mode is fetched by sampling the RST (Reset).

The following shows an operation sequence from an occurrence of reset cause to the determination of an operation mode.

Figure 3-1 Operation Mode Fetch Timing Chart

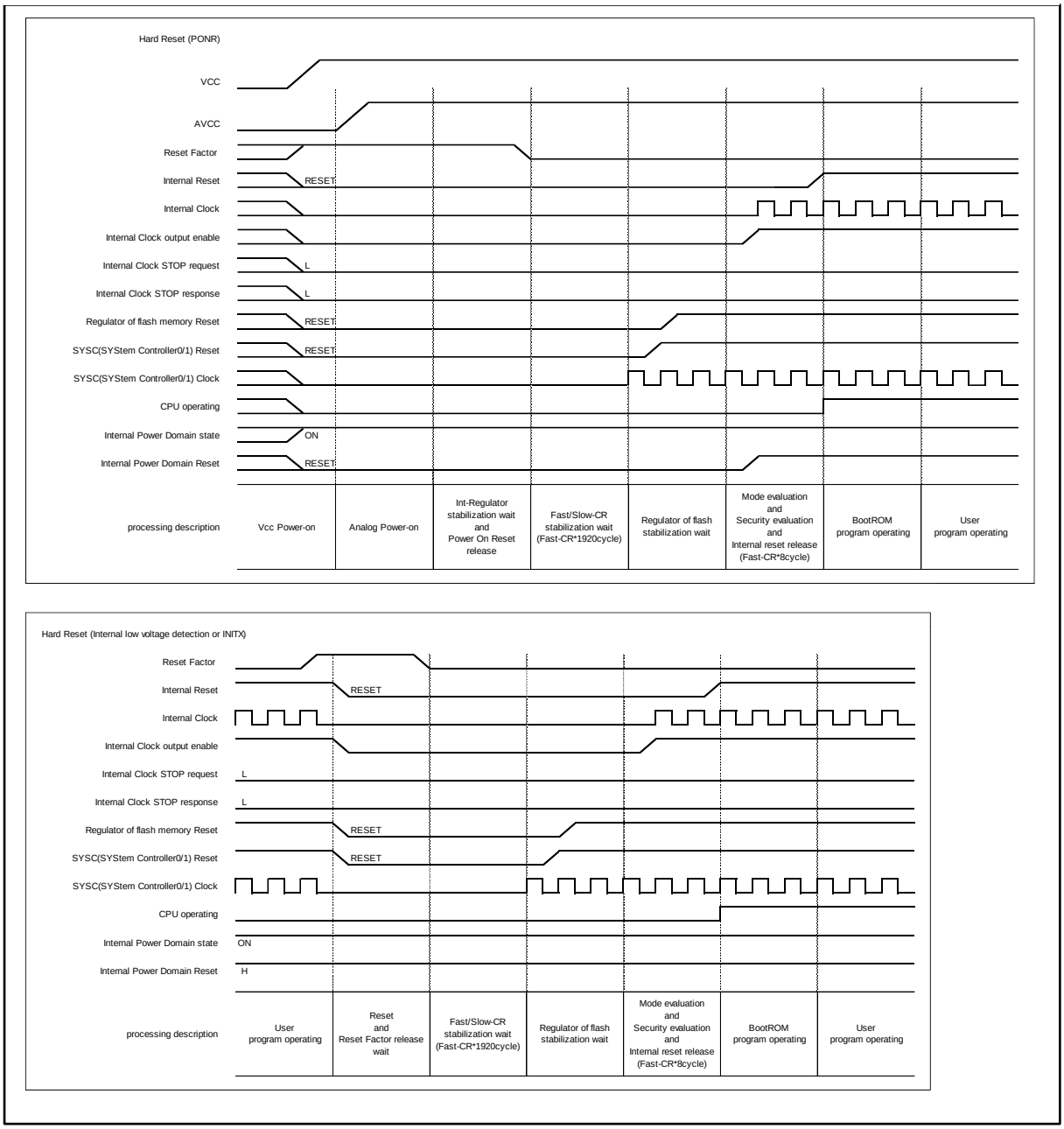


Figure 3-2 Operation Mode Fetch Timing Chart

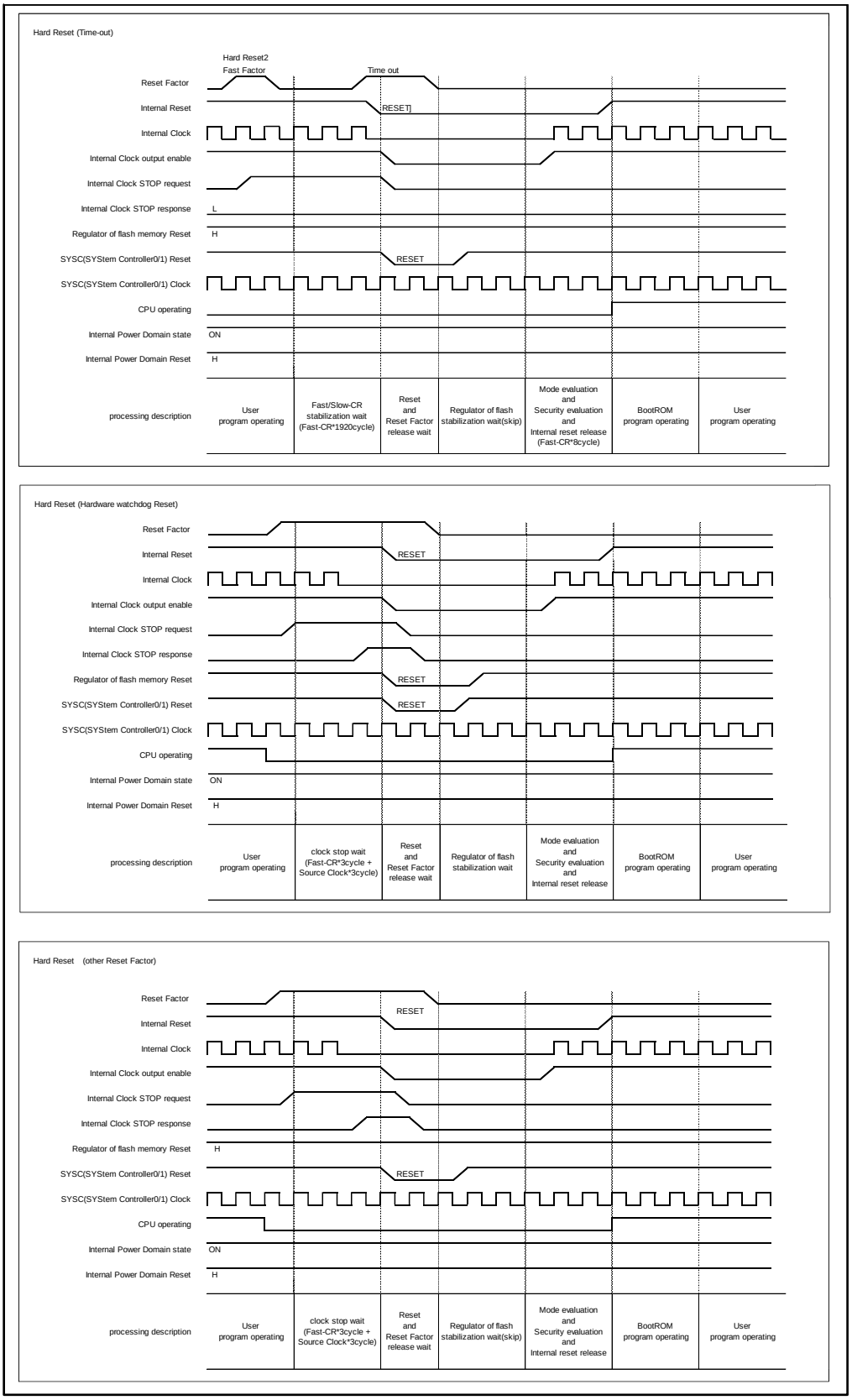
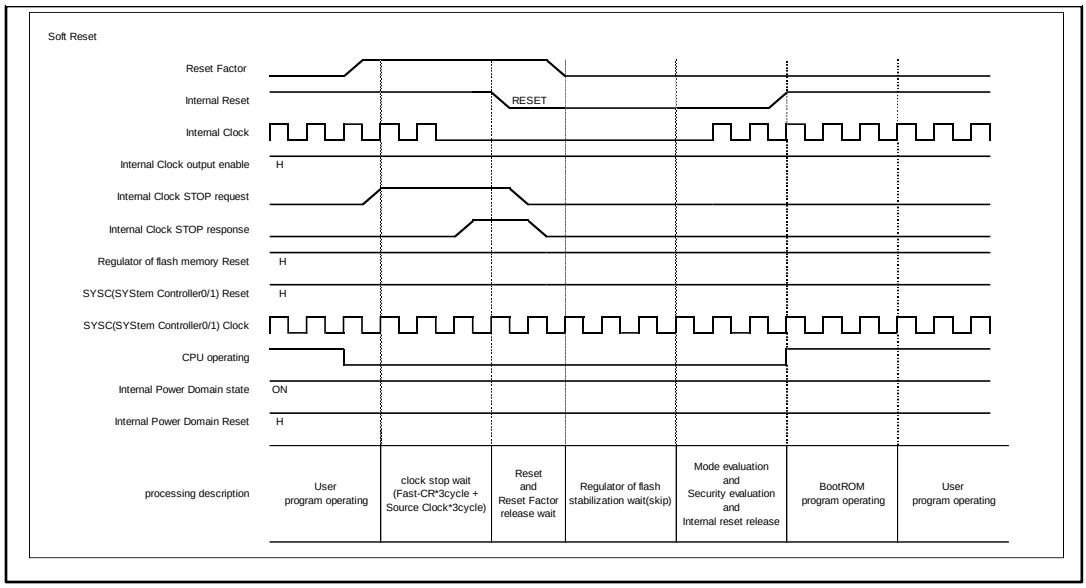


Figure 3-3 Operation Mode Fetch Timing Chart



4. Changes to PSS and RUN

This section shows the sequence which changes to PSS and RUN.

Figure 4-1 Changes to PSS Timing Chart

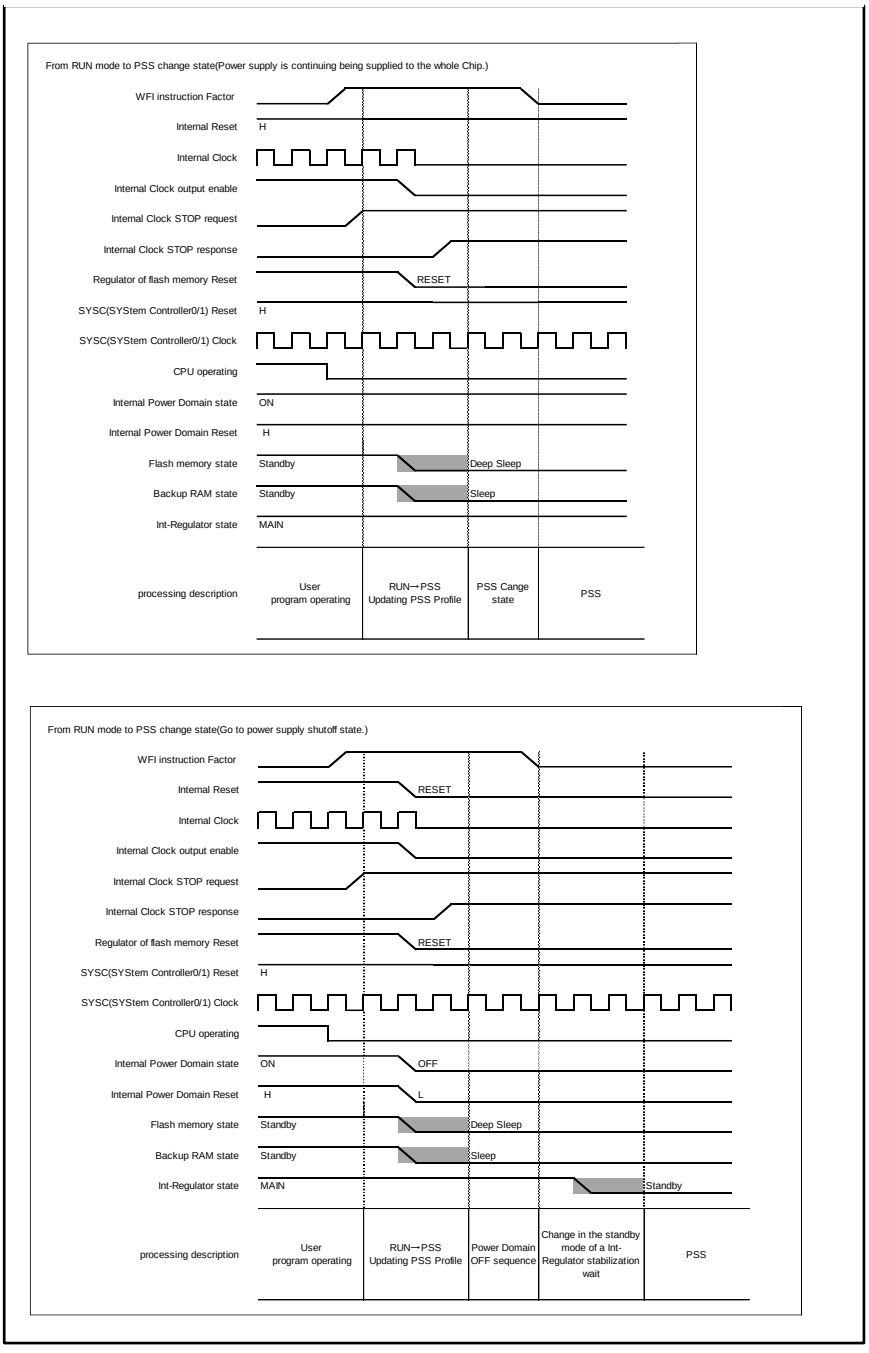


Figure 4-2 Changes to PSS Timing Chart

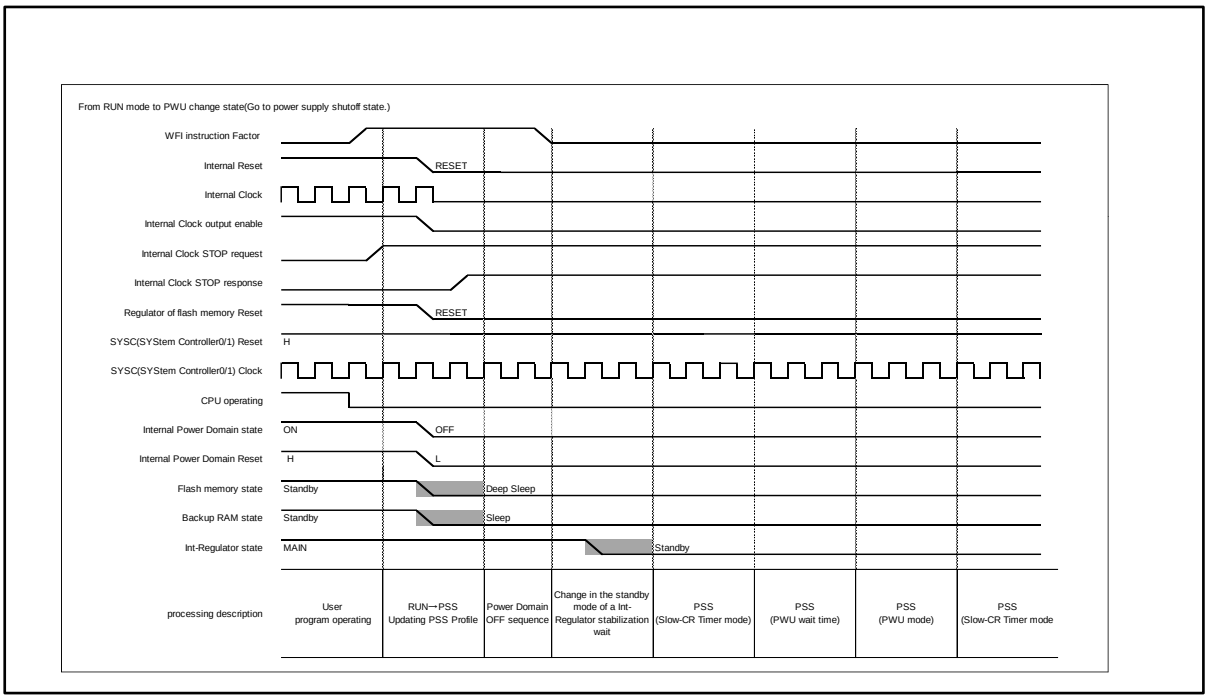
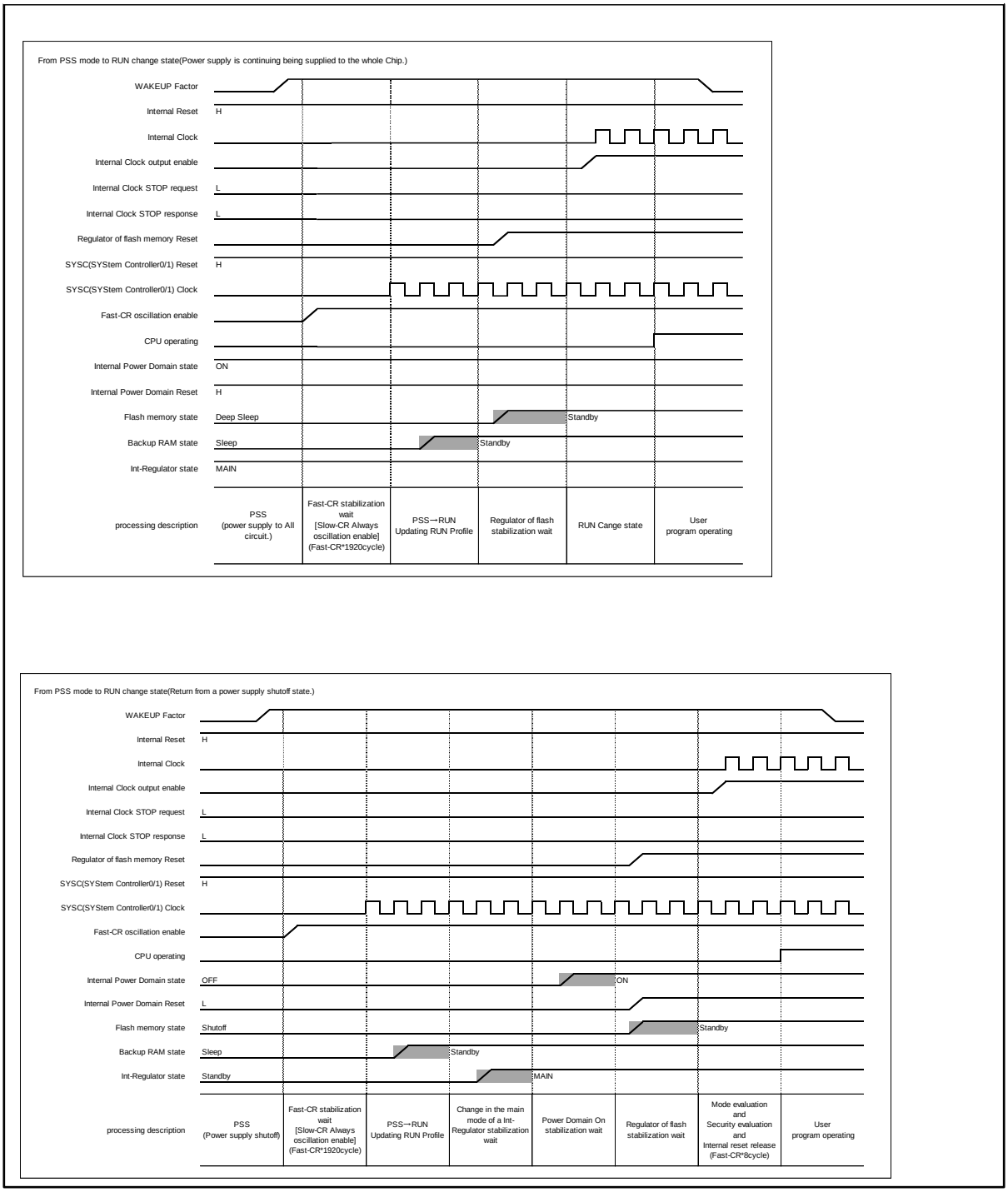


Figure 4-3 Changes to RUN Timing Chart



CHAPTER 64: APPENDIX: I/O Map

This chapter explains I/O MAP.

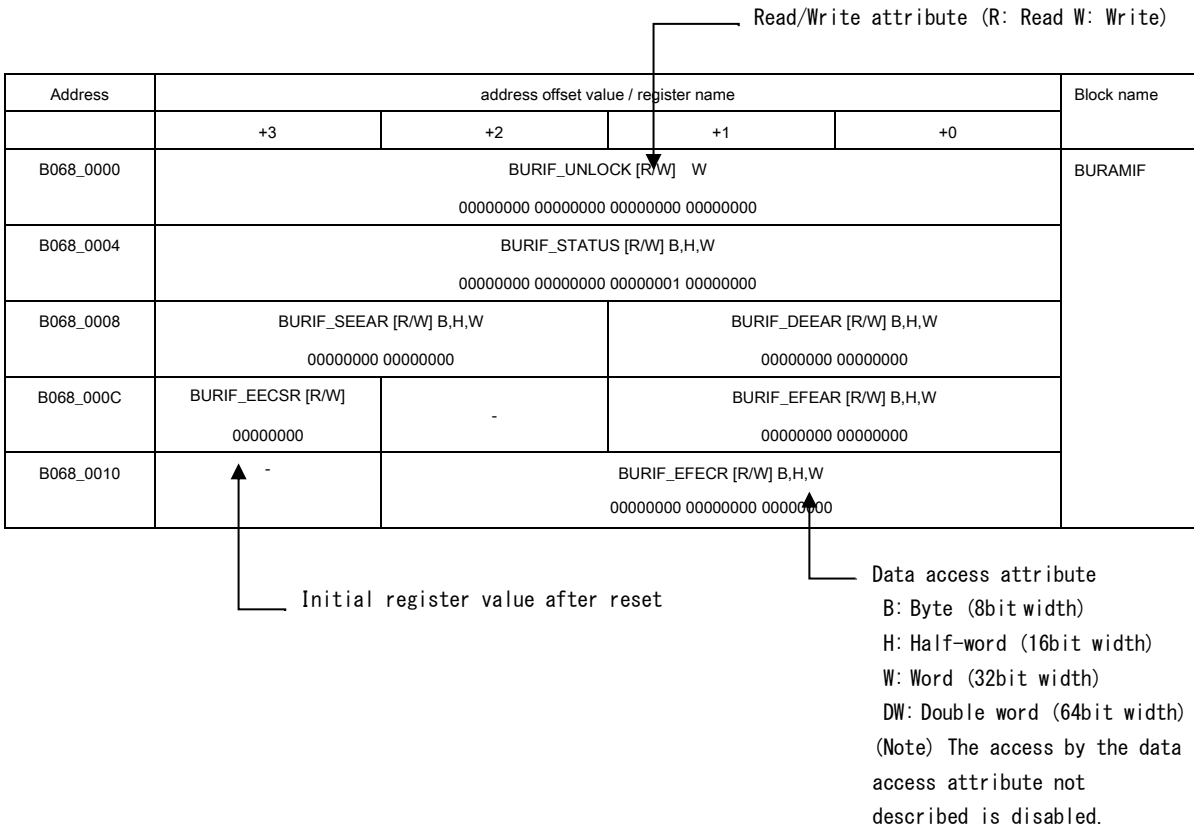
1. I/O MAP
2. I/O Map (S6J3120 series)
3. APPS#5

IOMAP-T01P00R01L01-E1-GBM03

1. I/O MAP

The following I/O map shows the relationship between memory space and registers for peripheral resources.

Figure 1-1 Legend of I/O Map



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

Note:

- *The access by the data access attribute not described is disabled.*

2. I/O Map (S6J3120 Series)

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
009F_0000 to 009F_0084	-				BootROM SW I/F
009F_0088	DDR_DSM [R/W] 11111111 11111111 11111111 11111111				
009F_008C	-				
009F_0090	DDR_DSKM0 [R/W] 11111111 11111111 11111111 11111111				
009F_0094	-				
009F_0098	DDR_DSKM1 [R/W] 11111111 11111111 11111111 11111111				
009F_009C	-				
009F_00A0	DDR_DSKM2 [R/W] 11111111 11111111 11111111 11111111				
009F_00A4	-				
009F_00A8	DDR_DSKM3 [R/W] 11111111 11111111 11111111 11111111				
009F_00AC to 009F_00BC	-				
009F_00C0	BDR_SBMM [R/W] 11111111 11111111 11111111 11111111				
009F_00C4	-				
009F_00C8	BDR_SBSM [R/W] 11111111 11111111 11111111 11111111				
009F_00CC to 009F_00DC	-				
009F_00E0	BDR_DWEM [R/W] 11111111 11111111 11111111 11111111				
009F_00E4	-				
009F_00E8	BDR_ABVM [R/W] 11111111 11111111 11111111 11111111				
009F_00EC	-				
009F_00F0	BDR_ABVEM [R/W] 11111111 11111111 11111111 11111111				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
009F_00F4 to 009F_00FC	-				
009F_0100	WDR_INTM [R/W] 11111111 11111111 11111111 11111111				
009F_0104	-				
009F_0108	WDR_TRG0CFGM [R/W] 11111111 11111111 11111111 11111111				
009F_010C	-				
009F_0110	WDR_TRG1CFGM [R/W] 11111111 11111111 11111111 11111111				
009F_0114	-				
009F_0118	WDR_RUNLLM [R/W] 11111111 11111111 11111111 11111111				
009F_011C	-				
009F_0120	WDR_RUNULM [R/W] 11111111 11111111 11111111 11111111				
009F_0124	-				
009F_0128	WDR_PSSLLM [R/W] 11111111 11111111 11111111 11111111				
009F_012C	-				
009F_0130	WDR_PSSULM [R/W] 11111111 11111111 11111111 11111111				
009F_0134	-				
009F_0138	WDR_RSTDLYM [R/W] 11111111 11111111 11111111 11111111				
009F_013C	-				
009F_0140	WDR_CFGM [R/W] 11111111 11111111 11111111 11111111				
009F_0144	-				
009F_0148	WDR_CEM [R/W] 11111111 11111111 11111111 11111111				
009F_014C	-				
B000_0000 to B00F_FFFC	-				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B010_0000	EBI_UNLOCK [W] W 00000000 00000000 00000000 00000000				EBI
B010_0004	EBI_LSTSR [R] W 00000000 00000000 00000000 00000001				
B010_0008	EBI_SFMR0 [R/W] W 00000000 00000000 00000000 00000000				
B010_000C	EBI_SFMR1 [R/W] W 00000000 00000000 00000000 00000000				
B010_0010	EBI_SFMR2 [R/W] W 00000000 00000000 00000000 00000000				
B010_0014	EBI_SFMR3 [R/W] W 00000000 00000000 00000000 00000000				
B010_0018	EBI_SFMR4 [R/W] W 00000000 00000000 00000000 00000000				
B010_001C	EBI_SFMR5 [R/W] W 00000000 00000000 00000000 00000000				
B010_0020	EBI_SFMR6 [R/W] W 00000000 00000000 00000000 00000000				
B010_0024	EBI_SFMR7 [R/W] W 00000000 00000000 00000000 00000000				
B010_0028	EBI_SFACCR0[R/W] W 00000101 01011111 11110000 00001111				
B010_002C	EBI_SFACCR1 [R/W] W 00000101 01011111 11110000 00001111				
B010_0030	EBI_SFACCR2 [R/W] W 00000101 01011111 11110000 00001111				
B010_0034	EBI_SFACCR3 [R/W] W 00000101 01011111 11110000 00001111				
B010_0038	EBI_SFACCR4 [R/W] W 00000101 01011111 11110000 00001111				
B010_003C	EBI_SFACCR5 [R/W] W 00000101 01011111 11110000 00001111				
B010_0040	EBI_SFACCR6 [R/W] W 00000101 01011111 11110000 00001111				
B010_0044	EBI_SFACCR7 [R/W] W 00000101 01011111 11110000 00001111				
B010_0048	EBI_SFADDCR0 [R/W] W 00000000 00001111 00000000 00000000				
B010_004C	EBI_SFADDCR1 [R/W] W 00000000 00001111 00000000 00010000				
B010_0050	EBI_SFADDCR2 [R/W] W 00000000 00001111 00000000 00100000				
B010_0054	EBI_SFADDCR3 [R/W] W 00000000 00001111 00000000 00110000				
B010_0058	EBI_SFADDCR4 [R/W] W 00000000 00001111 00000000 01000000				
B010_005C	EBI_SFADDCR5 [R/W] W 00000000 00001111 00000000 01010000				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B010_0060	EBI_SFADDCR6 [R/W] W 00000000 00001111 00000000 01100000				
B010_0064	EBI_SFADDCR7 [R/W] W 00000000 00001111 00000000 01110000				
B010_0068	-				
B010_006C	-				
B010_0070	-				
B010_0074	-				
B010_0078	-				
B010_007C	EBI_ERRR [R/W] W 00000000 00000000 00000000 00000000				
B010_0080 to B010_03FC	-				
B010_0400 to B010_0FFC	-				Reserved
B010_1000	DDRHSSPI0_MCTRL [R/W] B,H,W 00000000 00000000 00000000 00000000				DDRHSSPI
B010_1004	DDRHSSPI0_PCC0 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B010_1008	DDRHSSPI0_PCC1 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B010_100C	DDRHSSPI0_PCC2 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B010_1010	DDRHSSPI0_PCC3 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B010_1014	DDRHSSPI0_TXF [R] B,H,W 00000000 00000000 00000000 00000000				
B010_1018	DDRHSSPI0_TXE [R/W] B,H,W 00000000 00000000 00000000 00000000				
B010_101C	DDRHSSPI0_TXC [R/W] B,H,W 00000000 00000000 00000000 00000000				
B010_1020	DDRHSSPI0_RXF [R] B,H,W 00000000 00000000 00000000 00000000				
B010_1024	DDRHSSPI0_RXE [R/W] B,H,W 00000000 00000000 00000000 00000000				
B010_1028	DDRHSSPI0_RXC [R/W] B,H,W 00000000 00000000 00000000 00000000				
B010_102C	DDRHSSPI0_FAULTF [R] B,H,W 00000000 00000000 00000000 00000000				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B010_1030	DDRHSSPI0_FAULTC [R/W] B,H,W 00000000 00000000 00000000 00000000				
B010_1034	-		DDRHSSPI0_DMAEN [R/W] B,H,W 00000000	DDRHSSPI0_DMCFG [R/W] B,H,W 00000000	
B010_1038	DDRHSSPI0_DMTRP [R/W] B,H,W 00000000	DDRHSSPI0_DMPSEL [R/W] B,H,W 00000000	-	DDRHSSPI0_DMSTART [R/W] B,H,W 00000000	
B010_103C	DDRHSSPI0_DMBCS [R] H,W 00000000 00000000		DDRHSSPI0_DMBCC [R/W] H,W 00000000 00000000		
B010_1040	DDRHSSPI0_DMFIFOSTATUS [R] B,H,W 00000000 00000000 00000000 00000000				
B010_1044	DDRHSSPI0_DMFIFOCFG [R/W] B,H,W 00000000 00000000 00001000 00001111				
B010_1048	DDRHSSPI0_TXFIFO0 [R/W] B,H,W(*1) 00000000 00000000 00000000 00000000				
B010_104C	DDRHSSPI0_TXFIFO1 [R/W] B,H,W(*1) 00000000 00000000 00000000 00000000				
B010_1050	DDRHSSPI0_TXFIFO2 [R/W] B,H,W(*1) 00000000 00000000 00000000 00000000				
B010_1054	DDRHSSPI0_TXFIFO3 [R/W] B,H,W(*1) 00000000 00000000 00000000 00000000				
B010_1058	DDRHSSPI0_TXFIFO4 [R/W] B,H,W(*1) 00000000 00000000 00000000 00000000				
B010_105C	DDRHSSPI0_TXFIFO5 [R/W] B,H,W(*1) 00000000 00000000 00000000 00000000				
B010_1060	DDRHSSPI0_TXFIFO6 [R/W] B,H,W(*1) 00000000 00000000 00000000 00000000				
B010_1064	DDRHSSPI0_TXFIFO7 [R/W] B,H,W(*1) 00000000 00000000 00000000 00000000				
B010_1068	DDRHSSPI0_TXFIFO8 [R/W] B,H,W(*1) 00000000 00000000 00000000 00000000				
B010_106C	DDRHSSPI0_TXFIFO9 [R/W] B,H,W(*1) 00000000 00000000 00000000 00000000				
B010_1070	DDRHSSPI0_TXFIFO10 [R/W] B,H,W(*1) 00000000 00000000 00000000 00000000				
B010_1074	DDRHSSPI0_TXFIFO11 [R/W] B,H,W(*1) 00000000 00000000 00000000 00000000				
B010_1078	DDRHSSPI0_TXFIFO12 [R/W] B,H,W(*1) 00000000 00000000 00000000 00000000				
B010_107C	DDRHSSPI0_TXFIFO13 [R/W] B,H,W(*1) 00000000 00000000 00000000 00000000				
B010_1080	DDRHSSPI0_TXFIFO14 [R/W] B,H,W(*1) 00000000 00000000 00000000 00000000				
B010_1084	DDRHSSPI0_TXFIFO15 [R/W] B,H,W(*1) 00000000 00000000 00000000 00000000				
B010_1088	DDRHSSPI0_TXFIFO16 [R/W] B,H,W(*1) 00000000 00000000 00000000 00000000				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B010_108C	DDRHSSPI0_TXFIFO17 [R/W] B,H,W(*1) 00000000 00000000 00000000 00000000				
B010_1090	DDRHSSPI0_TXFIFO18 [R/W] B,H,W(*1) 00000000 00000000 00000000 00000000				
B010_1094	DDRHSSPI0_TXFIFO19 [R/W] B,H,W(*1) 00000000 00000000 00000000 00000000				
B010_1098	DDRHSSPI0_TXFIFO20 [R/W] B,H,W(*1) 00000000 00000000 00000000 00000000				
B010_109C	DDRHSSPI0_TXFIFO21 [R/W] B,H,W(*1) 00000000 00000000 00000000 00000000				
B010_10A0	DDRHSSPI0_TXFIFO22 [R/W] B,H,W(*1) 00000000 00000000 00000000 00000000				
B010_10A4	DDRHSSPI0_TXFIFO23 [R/W] B,H,W(*1) 00000000 00000000 00000000 00000000				
B010_10A8	DDRHSSPI0_RXFIFO0 [R] B,H,W(*1) 00000000 00000000 00000000 00000000				
B010_10AC	DDRHSSPI0_RXFIFO1 [R] B,H,W(*1) 00000000 00000000 00000000 00000000				
B010_10B0	DDRHSSPI0_RXFIFO2 [R] B,H,W(*1) 00000000 00000000 00000000 00000000				
B010_10B4	DDRHSSPI0_RXFIFO3 [R] B,H,W(*1) 00000000 00000000 00000000 00000000				
B010_10B8	DDRHSSPI0_RXFIFO4 [R] B,H,W(*1) 00000000 00000000 00000000 00000000				
B010_10BC	DDRHSSPI0_RXFIFO5 [R] B,H,W(*1) 00000000 00000000 00000000 00000000				
B010_10C0	DDRHSSPI0_RXFIFO6 [R] B,H,W(*1) 00000000 00000000 00000000 00000000				
B010_10C4	DDRHSSPI0_RXFIFO7 [R] B,H,W(*1) 00000000 00000000 00000000 00000000				
B010_10C8	DDRHSSPI0_RXFIFO8 [R] B,H,W(*1) 00000000 00000000 00000000 00000000				
B010_10CC	DDRHSSPI0_RXFIFO9 [R] B,H,W(*1) 00000000 00000000 00000000 00000000				
B010_10D0	DDRHSSPI0_RXFIFO10 [R] B,H,W(*1) 00000000 00000000 00000000 00000000				
B010_10D4	DDRHSSPI0_RXFIFO11 [R] B,H,W(*1) 00000000 00000000 00000000 00000000				
B010_10D8	DDRHSSPI0_RXFIFO12 [R] B,H,W(*1) 00000000 00000000 00000000 00000000				
B010_10DC	DDRHSSPI0_RXFIFO13 [R] B,H,W(*1) 00000000 00000000 00000000 00000000				
B010_10E0	DDRHSSPI0_RXFIFO14 [R] B,H,W(*1) 00000000 00000000 00000000 00000000				
B010_10E4	DDRHSSPI0_RXFIFO15 [R] B,H,W(*1) 00000000 00000000 00000000 00000000				
B010_10E8	DDRHSSPI0_RXFIFO16 [R] B,H,W(*1) 00000000 00000000 00000000 00000000				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B010_10EC	DDRHSSPI0_RXFIFO17 [R] B,H,W(*1) 00000000 00000000 00000000 00000000				
B010_10F0	DDRHSSPI0_RXFIFO18 [R] B,H,W(*1) 00000000 00000000 00000000 00000000				
B010_10F4	DDRHSSPI0_RXFIFO19 [R] B,H,W(*1) 00000000 00000000 00000000 00000000				
B010_10F8	DDRHSSPI0_RXFIFO20 [R] B,H,W(*1) 00000000 00000000 00000000 00000000				
B010_10FC	DDRHSSPI0_RXFIFO21 [R] B,H,W(*1) 00000000 00000000 00000000 00000000				
B010_1100	DDRHSSPI0_RXFIFO22 [R] B,H,W(*1) 00000000 00000000 00000000 00000000				
B010_1104	DDRHSSPI0_RXFIFO23 [R] B,H,W(*1) 00000000 00000000 00000000 00000000				
B010_1108	DDRHSSPI0_RDCSDC1 [R/W] H,W 00000000 00000000		DDRHSSPI0_RDCSDC0 [R/W] H,W 00000000 00000000		
B010_110C	DDRHSSPI0_RDCSDC3 [R/W] H,W 00000000 00000000		DDRHSSPI0_RDCSDC2 [R/W] H,W 00000000 00000000		
B010_1110	DDRHSSPI0_RDCSDC5 [R/W] H,W 00000000 00000000		DDRHSSPI0_RDCSDC4 [R/W] H,W 00000000 00000000		
B010_1114	DDRHSSPI0_RDCSDC7 [R/W] H,W 00000000 00000000		DDRHSSPI0_RDCSDC6 [R/W] H,W 00000000 00000000		
B010_1118	DDRHSSPI0_RDCSDC9 [R/W] H,W 00000000 00000000		DDRHSSPI0_RDCSDC8 [R/W] H,W 00000000 00000000		
B010_111C	DDRHSSPI0_RDCSDC11 [R/W] H,W 00000000 00000000		DDRHSSPI0_RDCSDC10 [R/W] H,W 00000000 00000000		
B010_1120	DDRHSSPI0_MID [R] B,H,W 00000000 00000000 00000000 00000001				
B010_1124	DDRHSSPI0_CSPREFETCHADDR [R] B,H,W 00000000 00000000 00000000 00000000				
B010_1128	DDRHSSPI0_SDAT ASAMPLEPTCNT3 [R/W] B,H,W 00000000	DDRHSSPI0_SDAT ASAMPLEPTCNT2 [R/W] B,H,W 00000000	DDRHSSPI0_SDAT ASAMPLEPTCNT1 [R/W] B,H,W 00000000	DDRHSSPI0_SDAT ASAMPLEPTCNT0 [R/W] B,H,W 00000000	
B010_112C	DDRHSSPI0_SDAT ASAMPLEPTCNT7 [R/W] B,H,W 00000000	DDRHSSPI0_SDAT ASAMPLEPTCNT6 [R/W] B,H,W 00000000	DDRHSSPI0_SDAT ASAMPLEPTCNT5 [R/W] B,H,W 00000000	DDRHSSPI0_SDAT ASAMPLEPTCNT4 [R/W] B,H,W 00000000	
B010_1130	DDRHSSPI0_SDAT ASAMPLEPTLFT3 [R/W] B,H,W 00000000	DDRHSSPI0_SDAT ASAMPLEPTLFT2 [R/W] B,H,W 00000000	DDRHSSPI0_SDAT ASAMPLEPTLFT1 [R/W] B,H,W 00000000	DDRHSSPI0_SDAT ASAMPLEPTLFT0 [R/W] B,H,W 00000000	
B010_1134	DDRHSSPI0_SDAT ASAMPLEPTLFT7 [R/W] B,H,W 00000000	DDRHSSPI0_SDAT ASAMPLEPTLFT6 [R/W] B,H,W 00000000	DDRHSSPI0_SDAT ASAMPLEPTLFT5 [R/W] B,H,W 00000000	DDRHSSPI0_SDAT ASAMPLEPTLFT4 [R/W] B,H,W 00000000	
B010_1138	DDRHSSPI0_SDAT ASAMPLEPTRGH3 [R/W] B,H,W 00000000	DDRHSSPI0_SDAT ASAMPLEPTRGH2 [R/W] B,H,W 00000000	DDRHSSPI0_SDAT ASAMPLEPTRGH1 [R/W] B,H,W 00000000	DDRHSSPI0_SDAT ASAMPLEPTRGH0 [R/W] B,H,W 00000000	
B010_113C	DDRHSSPI0_SDAT ASAMPLEPTRGH7 [R/W] B,H,W 00000000	DDRHSSPI0_SDAT ASAMPLEPTRGH6 [R/W] B,H,W 00000000	DDRHSSPI0_SDAT ASAMPLEPTRGH5 [R/W] B,H,W 00000000	DDRHSSPI0_SDAT ASAMPLEPTRGH4 [R/W] B,H,W 00000000	

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B010_1140	-				
B010_1144	DDRHSSPI0_DLP [R/W] B,H,W 00000000 00000000 00000000 00000000				
B010_1148	DDRHSSPI0_DLPSAMPLESTATUS [R] B,H,W 00000000 00000000 00000000 00000000				
B010_114C	DDRHSSPI0_CSCFG [R/W] B,H,W 00000000 00000000 00000000 00000000				
B010_1150	DDRHSSPI0_CSITIME [R/W] B,H,W 00000000 00000000 11111111 11111111				
B010_1154	DDRHSSPI0_CSAEXT [R/W] B,H,W 00000000 00000000 00000000 00000000				
B010_1158	DDRHSSPI0_CSPBUFFERCFG [R/W] B,H,W 00000000 00000000 00000000 00001111				
B010_115C	DDRHSSPI0_CSPBUFFERSTATUS [R] B,H,W 00000000 00000000 00000000 00000000				
B010_1160	-				
B010_1164 to B010_13FC	-				
B010_1400 to B010_7FFC	-				Reserved
B010_8000	SRCFG_CFG0 [R/W] B,H,W,DW 00000011 00000011 00000001 00000000				Scratch Pad RAM
B010_8004	SRCFG_CFG1 [R/W] B,H,W,DW 00000000 00000000 00000000 00000000				
B010_8008	SRCFG_CFG2 [R/W] B,H,W,DW 00000000 00000000 00000000 00000000				
B010_800C	SRCFG_KEY [R/W] B,H,W,DW 00000000 00000000 00000000 00000000				
B010_8010	SRCFG_ERRFLG [R/W] B,H,W,DW 00000000 00000000 00000000 00000000				
B010_8014	SRCFG_INTE [R/W] B,H,W,DW 00000000 00000000 00000000 00000000				
B010_8018	SRCFG_ECCE [R/W] B,H,W,DW 00000000 00000000 00000000 00000001				
B010_801C	-				
B010_8020	SRCFG_ERRADR [R] B,H,W,DW 00000000 00000000 00000000 00000000				
B010_8024	SRCFG_MID [R] B,H,W,DW 00000000 00000100 00000011 00000000				
B010_8028 to B010_80FC	-				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B010_8100 to B02F_FFFC	-				Reserved
B030_0000	SYSC1_PROTKEYR [R/W] W 00000000 00000000 00000000 00000000				Protection Register Area
B030_0004 to B030_007C	-				
B030_0080	SYSC1_RUNCKSELR0 [R/W] B,H,W 00000000 00000000 00000000 00000000				RUN Profile register area
B030_0084	SYSC1_RUNCKSELR1 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B030_0088	SYSC1_RUNCKSELR2 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B030_008C	SYSC1_RUNCKER0 [R/W] B,H,W 11111111 10111111 01001111 11110001				
B030_0090	SYSC1_RUNCKER1 [R/W] B,H,W 00011111 00011111 00011111 00000001				
B030_0094	SYSC1_RUNCKER2 [R/W] B,H,W 00000000 00000000 00011111 00011111				
B030_0098	SYSC1_RUNCKDIVR0 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B030_009C	SYSC1_RUNCKDIVR1 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B030_00A0	SYSC1_RUNCKDIVR2 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B030_00A4	SYSC1_RUNCKDIVR3 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B030_00A8	SYSC1_RUNCKDIVR4 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B030_00AC	-				
B030_00B0	-				
B030_00B4	-				
B030_00B8	-				
B030_00BC	-				
B030_00C0 to B030_00F8	-				
B030_00FC	SYSC1_RUNENR [R/W] B 00000000 00000000 00000000 00000000				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B030_0100	SYSC1_PSSCKSELR0 [R/W] B,H,W 00000000 00000000 00000000 00000000				PSS Profile register area
B030_0104	SYSC1_PSSCKSELR1 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B030_0108	SYSC1_PSSCKSELR2 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B030_010C	SYSC1_PSSCKER0 [R/W] B,H,W 11111111 10111111 01001111 11110000				
B030_0110	SYSC1_PSSCKER1 [R/W] B,H,W 00011111 00011111 00011111 00000001				
B030_0114	SYSC1_PSSCKER2 [R/W] B,H,W 00000000 00000000 00011111 00011111				
B030_0118	SYSC1_PSSCKDIVR0 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B030_011C	SYSC1_PSSCKDIVR1 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B030_0120	SYSC1_PSSCKDIVR2 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B030_0124	SYSC1_PSSCKDIVR3 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B030_0128	SYSC1_PSSCKDIVR4 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B030_012C	-				
B030_0130	-				
B030_0134	-				
B030_0138	-				
B030_013C	-				
B030_0140 to B030_0178	-				
B030_017C	SYSC1_PSSENR [R/W] B 00000000 00000000 00000000 00000000				
B030_0180	SYSC1_APPCKSELR0 [R/W] B,H,W 00000000 00000000 00000000 00000000				APP Profile register area
B030_0184	SYSC1_APPCKSELR1 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B030_0188	SYSC1_APPCKSELR2 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B030_018C	SYSC1_APPCKER0 [R/W] B,H,W 11111111 10111111 01001111 11110001				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B030_0190	SYSC1_APPCKER1 [R/W] B,H,W 00011111 00011111 00011111 00000001				
B030_0194	SYSC1_APPCKER2 [R/W] B,H,W 00000000 00000000 00011111 00011111				
B030_0198	SYSC1_APPCKDIVR0 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B030_019C	SYSC1_APPCKDIVR1 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B030_01A0	SYSC1_APPCKDIVR2 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B030_01A4	SYSC1_APPCKDIVR3 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B030_01A8	SYSC1_APPCKDIVR4 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B030_01AC	-				
B030_01B0	-				
B030_01B4	-				
B030_01B8	-				
B030_01BC	-				
B030_01C0 to B030_01FC	-				
B030_0200	SYSC1_STSCKSELR0 [R/W] B,H,W 00000000 00000000 00000000 00000000				STS Profile register area
B030_0204	SYSC1_STSCKSELR1 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B030_0208	SYSC1_STSCKSELR2 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B030_020C	SYSC1_STSCKER0 [R/W] B,H,W 11111111 10111111 01001111 11110001				
B030_0210	SYSC1_STSCKER1 [R/W] B,H,W 00011111 00011111 00011111 00000001				
B030_0214	SYSC1_STSCKER2 [R/W] B,H,W 00000000 00000000 00011111 00011111				
B030_0218	SYSC1_STSCKDIVR0 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B030_021C	SYSC1_STSCKDIVR1 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B030_0220	SYSC1_STSCKDIVR2 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B030_0224	SYSC1_STSCKDIVR3 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B030_0228	SYSC1_STSCKDIVR4 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B030_022C	-				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B030_0230	-				
B030_0234	-				
B030_0238	-				
B030_023C	-				
B030_0240 to B030_7FFC	-				
B030_8000	SWDG0_PROT [R/W] W 00000000 00000000 00000000 00000000				SWDT
B030_8004	-				
B030_8008	SWDG0_CNT [R/W] B,H,W 00000000 00000000 00000000 00000000				
B030_800C	SWDG0_RSTCAUSE [R/W] B,H,W 00000000 00000000 00000000 000XXXXX				
B030_8010	SWDG0_TRG0 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B030_8014	-				
B030_8018	SWDG0_TRG1 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B030_801C	-				
B030_8020	SWDG0_INT [R/W] B,H,W 00000000 00000010 00000000 00000000				
B030_8024	SWDG0_INTCLR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B030_8028	-				
B030_802C	SWDG0_TRG0CFG [R/W] B,H,W 00000000 00000000 00000000 00000000				
B030_8030	SWDG0_TRG1CFG [R/W] B,H,W 00000000 00000000 00000000 00000000				
B030_8034	SWDG0_RUNLLS [R/W] B,H,W 00000000 00000000 00000000 00000000				
B030_8038	SWDG0_RUNULS [R/W] B,H,W 00000001 00000000 00000000 00000000				
B030_803C	SWDG0_PSSLLS [R/W] B,H,W 00000000 00000000 00000000 00000000				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B030_8040	SWDG0_PSSULS [R/W] B,H,W 10000000 00000000 00000000 00000000				
B030_8044	SWDG0_RSTDLY [R/W] B,H,W 00000000 00000000 00000000 00000000				
B030_8048	SWDG0_CFG [R/W] B,H,W 00000001 00000000 00000000 00000011				
B030_804C	SWDG0_RUNLLC [R/W] B,H,W 00000000 00000000 00000000 00000000				
B030_8050	SWDG0_RUNULC [R/W] B,H,W 00000001 00000000 00000000 00000000				
B030_8054	SWDG0_PSSLLC [R/W] B,H,W 00000000 00000000 00000000 00000000				
B030_8058	SWDG0_PSSULC [R/W] B,H,W 10000000 00000000 00000000 00000000				
B030_805C to B03F_FFFC	-				
B040_0000	IRC0_NMIVAS [R/W] B,H,W 00000000 00000000 00000000 00000000				IRC
B040_0004	-		IRC0_NMIST [R/W] B,H,W 00001111 00100000		
B040_0008	IRC0_IRQVAS [R/W] B,H,W 00000000 00000000 00000000 00000000				
B040_000C	IRC0_IRQST [R/W] B,H,W 00000001 00011111 00000010 00000000				
B040_0010	IRC0_NMIVA0 [R/W] B,H,W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXX00				
B040_0010 + m*4 m=0-31(B040_ 008C)	IRC0_NMIVAm [R/W] B,H,W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXX00				
B040_0090 + r*4 r=0-511(B040_ 088C)	IRC0_IRQVAr [R/W] B,H,W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXX00				
B040_0890	IRC0_NMIPL0 [R/W] B,H,W 00001111 00001111 00001111 00000000				
B040_0890 + i*4 i=1-7(B040_08 AC)	IRC0_NMIPLi [R/W] B,H,W 00001111 00001111 00001111 00001111				
B040_08B0 + i*4 i=0-127(B040_ 0AAC)	IRC0_IRQPLi [R/W] B,H,W 00011111 00011111 00011111 00011111				
B040_0AB0	IRC0_NMIS [R/W] B,H,W 00000000 00000000 00000000 00000000				
B040_0AB4	IRC0_NMIR [R/W] B,H,W 00000000 00000000 00000000 00000000				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B040_0AB8	IRC0_NMISIS [R/W] B,H,W 00000000 00000000 00000000 00000000				
B040_0ABC	-				
B040_0AC0 + i*4 i=0-15(B040_0AFC)	IRC0_IRQSi [R/W] B,H,W 00000000 00000000 00000000 00000000				
B040_0B00 + i*4 i=0-15(B040_0B3C)	IRC0_IRQRi [R/W] B,H,W 00000000 00000000 00000000 00000000				
B040_0B40 + i*4 i=0-15(B040_0B7C)	IRC0_IRQSISi [R/W] B,H,W 00000000 00000000 00000000 00000000				
B040_0B80 + i*4 i=0-15(B040_0BBC)	IRC0_IRQCESi [R/W] B,H,W 00000000 00000000 00000000 00000000				
B040_0BC0 + i*4 i=0-15(B040_0BFC)	IRC0_IRQCECi [R/W] B,H,W 00000000 00000000 00000000 00000000				
B040_0C00 + i*4 i=0-15(B040_0C3C)	IRC0_IRQCEi [R/W] B,H,W 00000000 00000000 00000000 00000000				
B040_0C40	-			IRC0_NMIHC [R/W] B,H,W 00000000	
B040_0C44	IRC0_NMIHS [R/W] B,H,W 00000000 00000000 00000000 00000000				
B040_0C48	-		IRC0_IRQHC [R/W] H,W 00000000 00000000		
B040_0C4C	-				
B040_0C50 + i*4 i=0-15(B040_0C8C)	IRC0_IRQHSi [R/W] B,H,W 00000000 00000000 00000000 00000000				
B040_0C90	-			IRC0_IRQPLM [R/W] B,H,W 00100000	
B040_0C94	-				
B040_0C98	IRC0_CSR [R/W] B,H,W 00000000 00000001 00000000 00000000				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B040_0C9C to B040_0CA4	-				
B040_0CA8	IRC0_NMIRS [R/W] B,H,W 00000000 00000000 00000000 00000000				
B040_0CAC	IRC0_NMIPS [R/W] B,H,W 00000000 00000000 00000000 00000000				
B040_0CB0 + i*4 i=0-15(B040_0CEC)	IRC0_IRQRSi [R/W] B,H,W 00000000 00000000 00000000 00000000				
B040_0CF0 + i*4 i=0-15(B040_0D2C)	IRC0_IRQPSi [R/W] B,H,W 00000000 00000000 00000000 00000000				
B040_0D30	IRC0_UNLOCK [R/W] W 00000000 00000000 00000000 00000000				
B040_0D34	-				
B040_0D38	-				
B040_0D3C	IRC0_IRQEEVA [R/W] B,H,W 00000000 00000000 00000000 00000000				
B040_0D40	IRC0_EEI [R/W] B,H,W 00000000 00000000 00000000 00000000				
B040_0D44	-			IRC0_EAN [R/W] B,H,W 00000000	
B040_0D48	-			IRC0_ET [R/W] B,H,W 00000000	
B040_0D4C	IRC0_EEB0 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B040_0D50	IRC0_EEB1 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B040_0D54	-		IRC0_EEB2 [R/W] B,H,W 00000000 00000000		
B040_0D58 to B040_0D5C	-				
B040_0D60	-				
B040_0D64	-				
B040_0D68	-				
B040_0D6C	-				
B040_0D70	-				
B040_0D74	-				
B040_0D78 to B040_7FFC	-				
B040_8000	TPU_UNLOCK [R/W] W 00000000 00000000 00000000 00000000				TPU

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B040_8004	TPU_LST [R/W] B,H,W 00000000 00000000 00000000 00000001				
B040_8008	TPU_CFG [R/W] B,H,W 00000000 00000000 00000000 00000000				
B040_800C	TPU_TIR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B040_8010	TPU_TST [R/W] B,H,W 00000000 00000000 00000000 00000000				
B040_8014	TPU_TIE [R/W] B,H,W 00000000 00000000 00000000 00000000				
B040_8018	-				
B040_801C to B040_802C	-				
B040_8030 + m*4 m=0-7(B040_804C)	TPU_TCN0m [R/W] B,H,W 00000000 00000000 00000000 00000000				
B040_8050 + m*4 m=0-7(B040_806C)	TPU_TCN1m [R/W] B,H,W 00000000 00000000 00000000 00000000				
B040_8070 + m*4 m=0-7(B040_808C)	TPU_TCCm [R/W] B,H,W 00000000 00000000 00000000 00000000				
B040_8090 to B040_FFFC	-				
B041_0000	TRCFG_TCMCFG0 [R/W] B,H,W 00000000 00000000 00000001 00000000				TCRAM I/F
B041_0004	TRCFG_TCMCFG1 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B041_0008	TRCFG_TCMUNLOCK [R/W] W 00000000 00000000 00000000 00000000				
B041_000C	-				
B041_0010	-				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B041_0014	-				
B041_0018	-				
B041_001C	-				
B041_0020	-				
B041_0024	-				
B041_0028	-				
B041_002C	-				
B041_0030	TRCFG_TEAR0 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B041_0034	TRCFG_TEAR1 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B041_0038	TRCFG_TEAR2 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B041_003C	TRCFG_TASAR [R/W] B,H,W 00000000 00000000		TRCFG_TAEAR [R/W] B,H,W 00111111 11111111		
B041_0040	TRCFG_TTCR [R/W] B,H,W 00000000 00001100		TRCFG_TICR [R/W] B,H,W 00000000	TRCFG_TFECR [R/W] B,H,W 00000000	
B041_0044	TRCFG_TKCCR [R/W] B,H,W 00000000	-	TRCFG_TSCR [R/W] B,H,W 00000000	TRCFG_TSRCR [R/W] B,H,W 00000000	
B041_0048 to B041_0FFC	-				
B041_1000	TCFCFG_FCPROTKEY [R/W] W 00000000 00000000 00000000 00000000				TCFlash
B041_1004	-				
B041_1008	TCFCFG_FCFGR [R/W] B,H,W,DW 00000000 00000000 00000000 00000011				
B041_100C	-				
B041_1010	TCFCFG_FECCCTRL [R/W] B,H,W,DW 00000000 00000000 00000000 00000000				
B041_1014	-				
B041_1018	-				
B041_101C	TCFCFG_FECCEIR [R/W] B,H,W,DW 00000000 00000000 00000000 00000000				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B041_1020	TCFCFG_FICTRL [R/W] B,H,W,DW 00000000 00000000 00000000 00000000				
B041_1024	-				
B041_1028	TCFCFG_FDATEIR(lower32) [R/W] B,H,W,DW 00000000 00000000 00000000 00000000				
B041_102C	TCFCFG_FDATEIR(upper32) [R/W] B,H,W,DW 00000000 00000000 00000000 00000000				
B041_1030 to B041_1034	-				
B041_1038	TCFCFG_FSTAT [R/W] B,H,W,DW 00000000 00000000 00000001 00000101				
B041_103C to B041_104C	-				
B041_1050	TCFCFG_FSECIR [R/W] B,H,W,DW 00000000 00000000 00000000 00000000				
B041_1054	TCFCFG_FECCEAR [R/W] B,H,W,DW 00000000 00000000 00000000 00000000				
B041_1058	-				
B041_105C	-				
B041_1060	-				
B041_1064	-				
B041_1068	-				
B041_106C to B041_107C	-				
B041_1080	TCFCFG_FUCEDIR [R/W] B,H,W,DW 00000000 00000000 00000000 00000000				
B041_1084	TCFCFG_FUCEAR [R/W] B,H,W,DW 00000000 00000000 00000000 00000000				
B041_1088 to B041_108C	-				
B041_1090	TCFCFG_BRCFG [R/W] B,H,W,DW 00000000 00010000 00000000 11111111				
B041_1094	-				
B041_1098	TCFCFG_BRAT [R/W] B,H,W,DW 00000000 00000000 00000000 00000001				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B041_109C to B041_10FC	-				
B041_1100 to B041_111C	TCFCFG_TCMBUF0 [R/W] W,DW [151:128] = 10000000 00000000 00000000 [255:152] , [127:0] = All "0"				
B041_1120 to B041_113C	TCFCFG_TCMBUF1 [R/W] W,DW [151:128] = 10000000 00000000 00000000 [255:152] , [127:0] = All "0"				
B041_1140 to B041_115C	TCFCFG_TCMBUF2 [R/W] W,DW [151:128] = 10000000 00000000 00000000 [255:152] , [127:0] = All "0"				
B041_1160 to B041_117C	TCFCFG_TCMBUF3 [R/W] W,DW [151:128] = 10000000 00000000 00000000 [255:152] , [127:0] = All "0"				
B041_1180 to B041_119C	TCFCFG_TCMBUF4 [R/W] W,DW [151:128] = 10000000 00000000 00000000 [255:152] , [127:0] = All "0"				
B041_11A0 to B041_11BC	TCFCFG_TCMBUF5 [R/W] W,DW [151:128] = 10000000 00000000 00000000 [255:152] , [127:0] = All "0"				
B041_11C0 to B041_11DC	TCFCFG_TCMBUF6 [R/W] W,DW [151:128] = 10000000 00000000 00000000 [255:152] , [127:0] = All "0"				
B041_11E0 to B041_11FC	TCFCFG_TCMBUF7 [R/W] W,DW [151:128] = 10000000 00000000 00000000 [255:152] , [127:0] = All "0"				
B041_1200 to B041_121C	TCFCFG_TCMBUF8 [R/W] W,DW [151:128] = 10000000 00000000 00000000 [255:152] , [127:0] = All "0"				
B041_1220 to B041_123C	TCFCFG_TCMBUF9 [R/W] W,DW [151:128] = 10000000 00000000 00000000 [255:152] , [127:0] = All "0"				
B041_1240 to B041_125C	TCFCFG_TCMBUF10 [R/W] W,DW [151:128] = 10000000 00000000 00000000 [255:152] , [127:0] = All "0"				
B041_1260 to B041_127C	TCFCFG_TCMBUF11 [R/W] W,DW [151:128] = 10000000 00000000 00000000 [255:152] , [127:0] = All "0"				
B041_1280 to B041_129C	TCFCFG_TCMBUF12 [R/W] W,DW [151:128] = 10000000 00000000 00000000 [255:152] , [127:0] = All "0"				
B041_12A0 to B041_12BC	TCFCFG_TCMBUF13 [R/W] W,DW [151:128] = 10000000 00000000 00000000 [255:152] , [127:0] = All "0"				
B041_12C0 to B041_12DC	TCFCFG_TCMBUF14 [R/W] W,DW [151:128] = 10000000 00000000 00000000 [255:152] , [127:0] = All "0"				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B041_12E0 to B041_12FC	TCFCFG_TCMBUF15 [R/W] W,DW [151:128] = 10000000 00000000 00000000 [255:152] , [127:0] = All "0"				
B041_1300 to B041_13BC	-				
B041_13C0	TCFCFG0_SECSTAT [R/W] B,H,W 00000000 00000000 0000000* XX1**1**				
B041_13C4	TCFCFG0_SER [R/W] B,H,W ******** ******** ******** ********				
B041_13C8	TCFCFG0_SSR [R/W] B,H,W ******** ******** ******** ********				
B041_13CC	TCFCFG0_CEER [R/W] B,H,W ******** ******** ******** ********				
B041_13D0	TCFCFG0_SOER [R/W] W ******** ******** ******** ********				
B041_13D4	TCFCFG0_SWPOER [R/W] W ******** ******** ******** ********				
B041_13D8	TCFCFG0_WSWP [R/W] B,H,W ******** ******** ******** ********				
B041_13DC	TCFCFG0_C0SWP [R/W] B,H,W ******** ******** ******** ********				
B041_13E0	TCFCFG0_C1SWP [R/W] B,H,W ******** ******** ******** ********				
B041_13E4	TCFCFG0_C2SWP [R/W] B,H,W ******** ******** ******** ********				
B041_13E8	TCFCFG0_C3SWP [R/W] B,H,W ******** ******** ******** ********				
B041_13EC to B041_1FFC	-				
B041_2000	WFCFG_CPR [R/W] W 00000000 00000000 00000000 00000000				WorkFlash
B041_2004	-				
B041_2008	WFCFG_CR [R/W] B,H,W,DW 00000000 00000000 00000000 00000010				
B041_200C	WFCFG_ECR [R/W] B,H,W,DW 00000000 00000000 00000000 00000000				
B041_2010	WFCFG_WCR [R/W] B,H,W,DW 00000000 00000000 00000001 00000000				
B041_2014	WFCFG_WSR [R/W] B,H,W,DW 00000000 00000000 00000000 00000000				
B041_2018	-				
B041_201C	WFCFG_EEIR [R/W] B,H,W,DW 00000000 00000000 00000000 00000000				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B041_2020	-				
B041_2024	WFCFG_ICR [R/W] B,H,W,DW 00000000 00000000 00000000 00000000				
B041_2028	WFCFG_SR [R/W] B,H,W,DW 00000000 00000000 00000001 00000001				
B041_202C	WFCFG_SECIR [R/W] B,H,W,DW 00000000 00000000 00000000 00000000				
B041_2030	WFCFG_EEAR [R/W] B,H,W,DW 00000000 00000000 00000000 00000000				
B041_2034	-				
B041_2038	WFCFG_EMENR [R/W] B,H,W,DW 00000000 00000000 00000000 00000000				
B041_203C to B041_2050	-				
B041_2054	WFCFG_SEQCM [R/W] B,H,W,DW 00000000 11111111 00000000 00000000				
B041_2058	WFCFG_ARBERR [R/W] B,H,W,DW 00000000 00000000 00000000 00000000				
B041_205C	WFCFG_ARBCLR [R/W] B,H,W,DW 00000000 00000000 00000000 00000000				
B041_2060	WFCFG_BERR [R/W] B,H,W,DW 00000000 00000000 00000000 00000000				
B041_2064	WFCFG_BERRCLR [R/W] B,H,W,DW 00000000 00000000 00000000 00000000				
B041_2068	FCFG_WARBR [R/W] B,H,W,DW 00000001 00000000 00000000 00000000				
B041_206C	WFCFG_UCESR [R/W] B,H,W,DW 00000000 00000000 00000000 00000000				
B041_2070	WFCFG_UCEAR [R/W] B,H,W,DW 00000000 00000000 00000000 00000000				
B041_2074	-				
B041_2078	WFCFG_DBEIR(lower32) [R/W] B,H,W,DW 00000000 00000000 00000000 00000000				
B041_207C	WFCFG_DBEIR(upper32) [R/W] B,H,W,DW 00000000 00000000 00000000 00000000				
B041_2080 to B041_20FC	-				
B041_2100 to B04F_FFFC	-				Reserved
B050_0000	00000000 00000000 00010000 00000011				DAP ROM table

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B050_0004	00000000 00000000 00100000 00000011				
B050_0008	00000000 00000000 00110000 00000011				
B050_000C	00000000 00000000 01000000 00000011				
B050_0010	00000000 00001000 00000000 00000011				
B050_0014	00000000 00000000 00000000 00000000				
B050_0018 to B050_0FFC	-				
B050_1000 to B050_1FFC	ETB				see "CoreSight™ Components Technical Reference Manual"
B050_2000 to B050_2FFC	CTI#4				see "CoreSight™ Components Technical Reference Manual"
B050_3000 to B050_3FFC	TPIU				see "CoreSight™ Components Technical Reference Manual"
B050_4000 to B057_FFFC	TRACE FUNNEL				see "CoreSight™ Components Technical Reference Manual"
B058_0000	00000000 00000001 00000000 00000011				CR5_RomTable
B058_0004	00000000 00000001 00100000 00000010				
B058_0008	00000000 00000001 10000000 00000011				
B058_000C	00000000 00000001 10010000 00000010				
B058_0010	00000000 00000001 11000000 00000011				
B058_0014	00000000 00000001 11010000 00000010				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B058_0018	00000000 00000000 00000000 00000000				
B058_001C to B058_FFFC	-				
B059_0000 to B059_1FFC	CORE0				see "Cortex™-R5 Revision: r1p2 Technical Reference Manual"
B059_2000 to B059_7FFC	-				see "ARM® Debug Interface v5 Architecture Specification"
B059_8000 to B059_8FFC	CTI#0				see "CoreSight™ Components Technical Reference Manual"
B059_9000 to B059_BFFC	-				see "ARM® Debug Interface v5 Architecture Specification"
B059_C000 to B059_CFFC	ETM0				see "CoreSight™ ETM-R5™ Revision: r0p0 Technical Reference Manual"
B059_D000 to B05B_FFFC	-				see "ARM® Debug Interface v5 Architecture Specification"
B05C_0000 to B05F_FFFC	-				Reserved
B060_0000	SYSC0_PROTKEYR [R/W] W 00000000 00000000 00000000 00000000				Protection register Area
B060_0004 to B060_007C	-				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B060_0080	SYSC0_RUNPDCFGR [R/W] B,H,W 00000011 11110011 00010001 00000000				RUN profile register area
B060_0084	SYSC0_RUNCKSRER [R/W] B,H,W 00000000 00000000 00000000 00000111				
B060_0088	SYSC0_RUNCKSELR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B060_008C	SYSC0_RUNCKER [R/W] B,H,W 00000000 00000000 00000000 00000011				
B060_0090	SYSC0_RUNCKDIVR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B060_0094	SYSC0_RUNPLL0CNTR [R/W] B,H,W 00000000 00001101 00000001 00000000				
B060_0098	-				
B060_009C	-				
B060_00A0	-				
B060_00A4	SYSC0_RUNSSCG0CNTR0 [R/W] B,H,W 00000000 00001101 00000001 00000000				
B060_00A8	SYSC0_RUNSSCG0CNTR1 [R/W] B,H,W 00000000 00000000 00000000 00101001				
B060_00AC	-				
B060_00B0	-				
B060_00B4	-				
B060_00B8	-				
B060_00BC	-				
B060_00C0	-				
B060_00C4	SYSC0_RUNLVDCFGR [R/W] B,H,W 00000101 00000000 00001101 00000000				
B060_00C8	SYSC0_RUNCSVCFGR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B060_00CC	SYSC0_RUNREGCFGR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B060_00D0 to B060_00F8	-				
B060_00FC	SYSC0_TRGRUNCNTR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B060_0100	SYSC0_PSSPDCFGR [R/W] B,H,W 00000011 11110011 00010001 00000000				PSS profile register area
B060_0104	SYSC0_PSSCKSRER [R/W] B,H,W 00000000 00000000 00000000 00000111				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B060_0108	SYSC0_PSSCKSELR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B060_010C	SYSC0_PSSCKER [R/W] B,H,W 00000000 00000000 00000000 00000011				
B060_0110	SYSC0_PSSCKDIVR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B060_0114	SYSC0_PSSPLL0CNTR [R/W] B,H,W 00000000 00001101 00000001 00000000				
B060_0118	-				
B060_011C	-				
B060_0120	-				
B060_0124	SYSC0_PSSSSCG0CNTR0 [R/W] B,H,W 00000000 00001101 00000001 00000000				
B060_0128	SYSC0_PSSSSCG0CNTR1 [R/W] B,H,W 00000000 00000000 00000000 00101001				
B060_012C	-				
B060_0130	-				
B060_0134	-				
B060_0138	-				
B060_013C	-				
B060_0140	-				
B060_0144	SYSC0_PSSLVDCFGR [R/W] B,H,W 00000101 00000000 00001101 00000000				
B060_0148	SYSC0_PSSCSVCFGR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B060_014C	SYSC0_PSSREGCFGR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B060_0150 to B060_0178	-				
B060_017C	SYSC0_PSSENR [R/W] B 00000000 00000000 00000000 00000000				
B060_0180	SYSC0_APPPDCFGR [R/W] B,H,W 00000011 11110011 00010001 00000000				APP profile register area
B060_0184	SYSC0_APPCKSRER [R/W] B,H,W 00000000 00000000 00000000 00000111				
B060_0188	SYSC0_APPCKSELR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B060_018C	SYSC0_APPCKER [R/W] B,H,W 00000000 00000000 00000000 00000011				
B060_0190	SYSC0_APPCKDIVR [R/W] B,H,W 00000000 00000000 00000000 00000000				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B060_0194	SYSC0_APPPLL0CNTR [R/W] B,H,W 00000000 00001101 00000001 00000000				
B060_0198	-				
B060_019C	-				
B060_01A0	-				
B060_01A4	SYSC0_APPSSCG0CNTR0 [R/W] B,H,W 00000000 00001101 00000001 00000000				
B060_01A8	SYSC0_APPSSCG0CNTR1 [R/W] B,H,W 00000000 00000000 00000000 00101001				
B060_01AC	-				
B060_01B0	-				
B060_01B4	-				
B060_01B8	-				
B060_01BC	-				
B060_01C0	-				
B060_01C4	SYSC0_APPLVDCFGR [R/W] B,H,W 00000101 00000000 00001101 00000000				
B060_01C8	SYSC0_APPCSVCFGR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B060_01CC	SYSC0_APPREGCFGR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B060_01D0 to B060_01FC	-				
B060_0200	SYSC0_STSPDCFGR [R/W] B,H,W 00000011 11110011 00010001 00000000				STS profile register area
B060_0204	SYSC0_STSCKSRER [R/W] B,H,W 00000000 00000000 00000000 00000111				
B060_0208	SYSC0_STSCKSELR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B060_020C	SYSC0_STSCKER [R/W] B,H,W 00000000 00000000 00000000 00000011				
B060_0210	SYSC0_STSCKDIVR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B060_0214	SYSC0_STSPLL0CNTR [R/W] B,H,W 00000000 00001101 00000001 00000000				
B060_0218	-				
B060_021C	-				
B060_0220	-				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B060_0224	SYSC0_STSSSCG0CNTR0 [R/W] B,H,W 00000000 00001101 00000001 00000000				
B060_0228	SYSC0_STSSSCG0CNTR1 [R/W] B,H,W 00000000 00000000 00000000 00101001				
B060_022C	-				
B060_0230	-				
B060_0234	-				
B060_0238	-				
B060_023C	-				
B060_0240	-				
B060_0244	SYSC0_STSLVDCFGR [R/W] B,H,W 10000101 00000000 10001101 00000000				
B060_0248	SYSC0_STSCSVCFGR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B060_024C	SYSC0_STSREGCFGR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B060_0250 to B060_027C	-				
B060_0280	SYSC0_SYSIDR [R,WX] B,H,W				System register area
	00010000_000000000_01000101_11001111				
B060_0284	SYSC0_SYSPFIDR [R,WX] B,H,W				
	00000000 00010001 00000011 00000000				
B060_0288	SYSC0_SYSSTSR [R/W] B,H,W 00000000 00000000 00000000 00000001				
B060_028C	SYSC0_SYSINTER [R/W] B,H,W 00000000 00000000 00000000 00000000				
B060_0290	SYSC0_SYSICLR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B060_0294	SYSC0_SYSERRIR0 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B060_0298	SYSC0_SYSERRIR1 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B060_029C	SYSC0_SYSERRICLR0 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B060_02A0	SYSC0_SYSERRICLR1 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B060_02A4	SYSC0_SYSPROSTSR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B060_02A8	SYSC0_SYSRUNPEFR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B060_02AC	SYSC0_SYSPSSPEFR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B060_02B0 to B060_02FC	-				
B060_0300	SYSC_CSVMOCFGR0 [R/W] B,H,W				CSV

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
	00000000 00000000 00000000 00000000				
B060_0304	SYSC_CSVMOCFGR1 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B060_0308	-				
B060_030C	-				
B060_0310	SYSC_CSVPLL0CFGR0 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B060_0314	SYSC_CSVPLL0CFGR1 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B060_0318	-				
B060_031C	-				
B060_0320 to B060_032C	-				
B060_0330	SYSC_CSVSP0CFGR0 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B060_0334	SYSC_CSVSP0CFGR1 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B060_0338 to B060_034C	-				
B060_0350	-				
B060_0354	-				
B060_0358	-				
B060_035C	-				
B060_0360	SYSC_CSVOUTER [R/W] B,H,W 00000000 00000000 00000000 00000000				
B060_0364	SYSC_CSVTESTR [R/W] B,H,W 00000000 00000000 00000000 00000000				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B060_0368 to B060_037C	-				Reserved
B060_0380	SYSC_RSTCNTR [R/W] B,H,W 00000000 00000000 00000000 00000000				RESET
B060_0384	-				
B060_0388	-				
B060_038C	-				
B060_0390	SYSC_RSTCAUSEUR [R/W] B,H,W 00000000 00000000 00000000 00000001				
B060_0394	SYSC_EXCSVRSTCAUSEUR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B060_0398	SYSC_PDRSTCAUSEUR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B060_039C	-				
B060_03A0	SYSC_RSTCAUSEBT [R/W] B,H,W 00000000 00000000 00000000 00000001				
B060_03A4	SYSC_EXCSVRSTCAUSEBT [R/W] B,H,W 00000000 00000000 00000000 00000000				
B060_03A8	SYSC_PDRSTCAUSEBT [R/W] B,H,W 00000000 00000000 00000000 00000000				
B060_03AC	-				
B060_03B0	-				
B060_03B4	SYSC_PDRSTSTATUS [R] B,H,W 00000000 00000000 00000000 00000000				
B060_03B8 to B060_03FC	-				Reserved
B060_0400	SYSC_FCRCTTRGR [R/W] B,H,W 00000000 00000000 00000000 00000000				SCT
B060_0404	SYSC_FCRCTCNTR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B060_0408	SYSC_FCRCTCPR [R/W] B,H,W 00000000 00000110 00000000 00011110				
B060_040C	SYSC_FCRCTSTR [R] B,H,W 00000000 00000000 00000000 00000000				
B060_0410	SYSC_FCRCTINTER [R/W] B,H,W 00000000 00000000 00000000 00000000				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B060_0414	SYSC_FCRCTICLR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B060_0418 to B060_047C	-				
B060_0480	SYSC_SCRCTTRGR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B060_0484	SYSC_SCRCTCNTR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B060_0488	SYSC_SCRCTCPR [R/W] B,H,W 00000000 00000110 00000000 00000001				
B060_048C	SYSC_SCRCTSTR [R] B,H,W 00000000 00000000 00000000 00000000				
B060_0490	SYSC_SCRCTINTER [R/W] B,H,W 00000000 00000000 00000000 00000000				
B060_0494	SYSC_SCRCTICLR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B060_0498 to B060_04FC	-				
B060_0500	SYSC_MOCTTRGR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B060_0504	SYSC_MOCTCNTR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B060_0508	SYSC_MOCTCPR [R/W] B,H,W 00000000 00000110 00000010 00000000				
B060_050C	SYSC_MOCTSTR [R] B,H,W 00000000 00000000 00000000 00000000				
B060_0510	SYSC_MOCTINTER [R/W] B,H,W 00000000 00000000 00000000 00000000				
B060_0514	SYSC_MOCTICLR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B060_0518 to B060_05FC	-				
B060_0600	SYSC_CRCNTR [R/W] B,H,W 00000000 00000000 00001111 00010010				Clock System
B060_0604	SYSC_MOSCCNTR [R/W] B,H,W 00000000 00000000 00000001 00000000				
B060_0608	-				
B060_060C	SYSC_PLLSSCGSTCNTR [R/W] B,H,W 00000000 00000000 00000000 11111111				
B060_0610	SYSC_PLL0CGCNTR [R/W] B,H,W 00000000 11111111 00000000 00000001				
B060_0614	-				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B060_0618 to B060_061C	-				
B060_0620	SYSC_SSCG0CGCNTR [R/W] B,H,W 00000000 11111111 00000000 00000001				
B060_0624 to B060_067C	-				
B060_0680	SYSC0_SPECFGR [R/W] B,H,W 00000000 01000000 00000000 00001110				Special register area
B060_0684 to B060_06FC	-				Reserved
B060_0700	SYSC0_JTAGDETECT [R/W] B,H,W 00000000 00000000 00000000 00000000				Debug register area
B060_0704	SYSC0_JTAGCNFG [R/W] B,H,W 00000000 00000000 00000000 00000001 (*12)				
B060_0708	SYSC0_JTAGWAKEUP [R/W] B,H,W 00000000 00000000 00000000 00000000				
B060_070C to B060_07FC	-				
B060_0800	MODEC_MODER [R/W] B,H,W *0000000 00000000 000*0000 00000000				Mode
B060_0804	-				
B060_0808	-				
B060_080C	-				
B060_0810	-				
B060_0814	-				
B060_0818	-				
B060_081C to B060_BFFC	-				
B060_C000	HWDG_PROT [R/W] W 00000000 00000000 00000000 00000000				HWDT
B060_C004	-				
B060_C008	HWDG_CNT [R/W] B,H,W 00000000 00000000 00000000 00000000 (*13)				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B060_C00C	HWDG_RSTCAUSE [R/W] B,H,W 00000000 00000000 00000000 000XXXXX				
B060_C010	HWDG_TRG0 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B060_C014	-				
B060_C018	HWDG_TRG1 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B060_C01C	-				
B060_C020	HWDG_INT [R/W] B,H,W 00000000 00000010 00000000 00000000				
B060_C024	HWDG_INTCLR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B060_C028	-				
B060_C02C	HWDG_TRG0CFG [R/W] B,H,W 00000000 00000000 00000000 00000000				
B060_C030	HWDG_TRG1CFG [R/W] B,H,W 00000000 00000000 00000000 00000000				
B060_C034	HWDG_RUNLLS [R/W] B,H,W 00000000 00000000 00000000 00000000				
B060_C038	HWDG_RUNULS [R/W] B,H,W 00000001 00000000 00000000 00000000				
B060_C03C	HWDG_PSSLLS [R/W] B,H,W 00000000 00000000 00000000 00000000				
B060_C040	HWDG_PSSULS [R/W] B,H,W 10000000 00000000 00000000 00000000				
B060_C044	HWDG_RSTDLY [R/W] B,H,W 00000000 00000000 00000000 00000000				
B060_C048	HWDG_CFG [R/W] B,H,W 00000000 00000000 00000000 00000XXX				
B060_C04C	HWDG_RUNLLC [R/W] B,H,W 00000000 00000000 00000000 00000000				
B060_C050	HWDG_RUNULC [R/W] B,H,W 00000001 00000000 00000000 00000000				
B060_C054	HWDG_PSSLLC [R/W] B,H,W 00000000 00000000 00000000 00000000				
B060_C058	HWDG_PSSULC [R/W] B,H,W 10000000 00000000 00000000 00000000				
B060_C05C to B060_FFFC	-				
B061_0000 to B061_7FFC	-				Reserved

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B061_8000	RTC_WTCR [R/W] B,H,W 00000000 00000000 00000000 00000000				RTC register area
B061_8004	RTC_WTSR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B061_8008	RTC_WINS [R/W] 00000000 00000000 00000000 00000000				
B061_800C	RTC_WINE [R/W] 00000000 00000000 00000000 00000000				
B061_8010	RTC_WINC [R/W] B,H,W 00000000 00000000 00000000 00000000				
B061_8014	RTC_WTBR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B061_8018	RTC_WRT [R/W] B,H,W 00000000 00000000 00000000 00000000				
B061_801C	RTC_RTR1 [R/W] H,W 00000000 00000000 00000000 00000000				
B061_8020	RTC_CNTCAL [R/W] B,H,W 00000000 00000000 00000000 00000000				
B061_8024	RTC_CNTPCAL [R/W] B,H,W 00000000 00000000 00000000 00000000				
B061_8028	RTC_DURMW [R/W] B,H,W 00000000 00000000 00000000 00000000				
B061_802C	RTC_CALTRG [R/W] B,H,W 00000000 00000000 00000000 00000000				
B061_8030	-				
B061_8034	RTC_PWUTRGCR [R/W] B,H,W 00000010 00000000 00000000 00000000				
B061_8038	RTC_PWUTRGSR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B061_803C to B061_FFFC	-				
B062_0000	EIC00_ENIR [R/W] B,H,W 00000000 00000000 00000000 00000000				EIC register area
B062_0004	EIC00_ENISR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B062_0008	EIC00_ENICR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B062_000C	EIC00_EIRR [R/W] B,H,W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
B062_0010	EIC00_EIRCR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B062_0014	EIC00_NFER [R/W] B,H,W 00000000 00000000 00000000 00000000				
B062_0018	EIC00_NFESR [R/W] B,H,W 00000000 00000000 00000000 00000000				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B062_001C	EIC00_NFECR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B062_0020	EIC00_ELVR0 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B062_0024	EIC00_ELVR1 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B062_0028	-				
B062_002C	-				
B062_0030	EIC00_NMIR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B062_0034	EIC00_DRER [R/W] B,H,W 00000000 00000000 00000000 00000000				
B062_0038	EIC00_DRESR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B062_003C	EIC00_DRECR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B062_0040	EIC00_DRFR [R/W] B,H,W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
B062_0044 to B063_FFFC	-				
B064_0000 to B065_FFFC	-				Reserved
B066_0000 to B067_FFFC	-				Reserved
B068_0000	BURIF_UNLOCK [R/W] W 00000000 00000000 00000000 00000000				BURAMIF
B068_0004	BURIF_STATUS [R/W] B,H,W 00000000 00000000 00000001 00000000				
B068_0008	BURIF_SEEAR [R/W] B,H,W 00000000 00000000		BURIF_DEEAR [R/W] B,H,W 00000000 00000000		
B068_000C	BURIF_EECSR [R/W] B,H,W 00000000	-	BURIF_EFEAR [R/W] B,H,W 00000000 00000000		
B068_0010	-	BURIF_EFECR [R/W] B,H,W 00000000 00000000 00000000			
B068_0014	-				
B068_0018	BURIF_TEAR0 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B068_001C	BURIF_TEAR1 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B068_0020	BURIF_TEAR2 [R/W] B,H,W 00000000 00000000 00000000 00000000				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B068_0024	BURIF_TAEAR [R/W] B,H,W(*3) 0****111 11111111		BURIF_TASAR [R/W] B,H,W 00000000 00000000		
B068_0028	BURIF_TFECR [R/W] B,H,W 00000000	BURIF_TICR [R/W] B,H,W 00000000	BURIF_TTCR [R/W] B,H,W 00000000 00001100		
B068_002C	BURIF_TSRCR [R/W] B,H,W 00000000	-	-	BURIF_TKCCR [R/W] B,H,W 00000000	
B068_0030	BURIF_EVENTCLR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B068_0034 to B068_7FFC	-				
B068_8000	EICU0_CNFGR [R/W] B,H,W 00000000 00000000 00000000 00000000				EXTERNAL INTERRUPT CAPTURE
B068_8004	EICU0_IRENR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B068_8008	EICU0_SPLR0 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B068_800C	EICU0_SPLR1 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B068_8010	EICU0_SPLR2 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B068_8014	EICU0_SPLR3 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B068_8018	EICU0_SPLR4 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B068_801C	EICU0_SPLR5 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B068_8020	EICU0_SPLR6 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B068_8024	EICU0_SPLR7 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B068_8028 to B068_83FC	-				
B068_8400	CU_CUTD1 [R/W] B,H,W 11000011 01010000		CU_CUCR1 [R/W] B,H,W 11111111 00000000		CR calibration
B068_8404	CU_CUTR1 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B068_8408	CU_CUCRC1 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B068_840C to B068_87FC	-				
B068_8800	MCG_IRSR0 [R/W] B,H,W 00000000 00000000 00000000 00000000				MCU CONFIG register area
B068_8804	MCG_IRSR1 [R/W] B,H,W 00000000 00000000 00000000 00000000				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B068_8808	MCG_IRSR2 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B068_880C	-				
B068_8810	-				
B068_8814 to B068_8BFC	-				
B068_8C00	CANP_CTR [R/W] B.H.W 00000000 00000000 00000001 00000000				CAN_FD Prescaler
B068_8C04	CANP_STR [R/W] B.H.W 00000000 00000000 00000000 00000100				
B068_8C08 to B068_FFFC	-				
B069_0000 to B06F_FFFC	-				Reserved
B070_0000 to B07F_FFFC	-				Reserved
B080_0000 to B0FF_FFFC	-				BIT RMW alias for MCU config Gr. (Covers B060_0000 - B06F_FFFF)
B100_0000 to B10F_FFFC	-				BIT RMW alias for SYSC1 (Covers B030_0000 - B031_FFFF)
B110_0000 to B11F_FFFC	-				BIT RMW alias for MEMC (Covers B040_0000 - B041_FFFF)
B120_0000 to B1FF_FFFC	-				Reserved
B200_0000	SHE_CMD [R/W] B,H,W 00000000 00000000 00000000 00000000				SHE CONFIG register

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B200_0004	SHE_CMDCANCEL [R/W] B,H,W 00000000 00000000 00000000 00000000				
B200_0008	SHE_CLKCTRL [R/W] B,H,W 00000000 00000000 00000000 00000000				
B200_000C	SHE_STATUS [R] B,H,W 00000000 00000000 00000000 00000000				
B200_0010	SHE_ERC [R] B,H,W 00000000 00010000 00000000 00000000				
B200_0014	SHE_CLKSTAT [R] B,H,W 00000000 00000000 00000000 00000000				
B200_0018	SHE_MID [R] B,H,W 00000000 00001111 00000010 00000000				
B200_001C	SHE_IRQ [R] B,H,W 00000000 00000000 00000000 00000000				
B200_0020	SHE_IRQEN [R/W] B,H,W 00000000 00000000 00000000 00000000				
B200_0024	SHE_IRQCLR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B200_0028	SHE_IMSTADDR [R/W] B,H,W 00000000 00000000 00000000 00000000 (*11)				
B200_002C	SHE_OMSTADDR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B200_0030	SHE_IMSTCNT [R/W] B,H,W 00000000 00000000 00000000 00000000 (*11)				
B200_0034	SHE_OMSTCNT [R/W] B,H,W 00000000 00000000 00000000 00000000				
B200_0038	SHE_IMSTSTART [R/W] B,H,W 00000000 00000000 00000000 00000000				
B200_003C	SHE_OMSTSTART [R/W] B,H,W 00000000 00000000 00000000 00000000				
B200_0040	SHE_IFIFOCFG [R/W] B,H,W 00000000 00000001 00000000 00000000				
B200_0044	SHE_OFIFOCFG [R/W] B,H,W 00000000 00000001 00000000 00000000				
B200_0048	SHE_COMPARE0 [R/W] B,H,W 11111111 11111111 11111111 11111111				
B200_004C	SHE_COMPARE1 [R/W] B,H,W 00000111 11111111 11111111 11111111				
B200_0050	SHE_COMPACC [R] B,H,W 00000000 00000000 00000000 00000000				
B200_0054	SHE_MSTSTATUS [R] B,H,W 00000000 00000001 00000000 00000001				
B200_0058	SHE_IMSTERRADDR [R] B,H,W 00000000 00000000 00000000 00000000				
B200_005C	SHE_OMSTERRADDR [R] B,H,W 00000000 00000000 00000000 00000000				
B200_0060	SHE_FIFOSTATUS [R] B,H,W 00000000 00000000 00000000 00000000				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B200_0064	SHE_FIFOLOAD [R] B,H,W 00000000 00110000 00000000 00110000				
B200_0068	SHE_DATACNT0 [R] B,H,W 00000000 00000000 00000000 00000000				
B200_006C	SHE_DATACNT1 [R] B,H,W 00000000 00000000 00000000 00000000				
B200_0070 to B200_00FC	-				
B200_0100	SHE_IFIFOWRDATA0 [R/W] W 00000000 00000000 00000000 00000000				
B200_0104	SHE_IFIFOWRDATA1 [R/W] W 00000000 00000000 00000000 00000000				
B200_0108	SHE_IFIFOWRDATA2 [R/W] W 00000000 00000000 00000000 00000000				
B200_010C	SHE_IFIFOWRDATA3 [R/W] W 00000000 00000000 00000000 00000000				
B200_0110	SHE_IFIFOWRDATA4 [R/W] W 00000000 00000000 00000000 00000000				
B200_0114	SHE_IFIFOWRDATA5 [R/W] W 00000000 00000000 00000000 00000000				
B200_0118	SHE_IFIFOWRDATA6 [R/W] W 00000000 00000000 00000000 00000000				
B200_011C	SHE_IFIFOWRDATA7 [R/W] W 00000000 00000000 00000000 00000000				
B200_0120	SHE_IFIFOWRDATA8 [R/W] W 00000000 00000000 00000000 00000000				
B200_0124	SHE_IFIFOWRDATA9 [R/W] W 00000000 00000000 00000000 00000000				
B200_0128	SHE_IFIFOWRDATA10 [R/W] W 00000000 00000000 00000000 00000000				
B200_012C	SHE_IFIFOWRDATA11 [R/W] W 00000000 00000000 00000000 00000000				
B200_0130	SHE_IFIFOWRDATA12 [R/W] W 00000000 00000000 00000000 00000000				
B200_0134	SHE_IFIFOWRDATA13 [R/W] W 00000000 00000000 00000000 00000000				
B200_0138	SHE_IFIFOWRDATA14 [R/W] W 00000000 00000000 00000000 00000000				
B200_013C	SHE_IFIFOWRDATA15 [R/W] W 00000000 00000000 00000000 00000000				
B200_0140	SHE_IFIFOWRDATA16 [R/W] W 00000000 00000000 00000000 00000000				
B200_0144	SHE_IFIFOWRDATA17 [R/W] W 00000000 00000000 00000000 00000000				
B200_0148	SHE_IFIFOWRDATA18 [R/W] W 00000000 00000000 00000000 00000000				
B200_014C	SHE_IFIFOWRDATA19 [R/W] W 00000000 00000000 00000000 00000000				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B200_0150	SHE_IFIFOWRDATA20 [R/W] W 00000000 00000000 00000000 00000000				
B200_0154	SHE_IFIFOWRDATA21 [R/W] W 00000000 00000000 00000000 00000000				
B200_0158	SHE_IFIFOWRDATA22 [R/W] W 00000000 00000000 00000000 00000000				
B200_015C	SHE_IFIFOWRDATA23 [R/W] W 00000000 00000000 00000000 00000000				
B200_0160	SHE_IFIFOWRDATA24 [R/W] W 00000000 00000000 00000000 00000000				
B200_0164	SHE_IFIFOWRDATA25 [R/W] W 00000000 00000000 00000000 00000000				
B200_0168	SHE_IFIFOWRDATA26 [R/W] W 00000000 00000000 00000000 00000000				
B200_016C	SHE_IFIFOWRDAT27 [R/W] W 00000000 00000000 00000000 00000000				
B200_0170	SHE_IFIFOWRDATA28 [R/W] W 00000000 00000000 00000000 00000000				
B200_0174	SHE_IFIFOWRDATA29 [R/W] W 00000000 00000000 00000000 00000000				
B200_0178	SHE_IFIFOWRDATA30 [R/W] W 00000000 00000000 00000000 00000000				
B200_017C	SHE_IFIFOWRDATA31 [R/W] W 00000000 00000000 00000000 00000000				
B200_0180	SHE_OFIFORDDATA0 [R] W 00000000 00000000 00000000 00000000				
B200_0184	SHE_OFIFORDDATA1 [R] W 00000000 00000000 00000000 00000000				
B200_0188	SHE_OFIFORDDATA2 [R] W 00000000 00000000 00000000 00000000				
B200_018C	SHE_OFIFORDDATA3 [R] W 00000000 00000000 00000000 00000000				
B200_0190	SHE_OFIFORDDATA4 [R] W 00000000 00000000 00000000 00000000				
B200_0194	SHE_OFIFORDDATA5 [R] W 00000000 00000000 00000000 00000000				
B200_0198	SHE_OFIFORDDATA6 [R] W 00000000 00000000 00000000 00000000				
B200_019C	SHE_OFIFORDDATA7 [R] W 00000000 00000000 00000000 00000000				
B200_01A0	SHE_OFIFORDDATA8 [R] W 00000000 00000000 00000000 00000000				
B200_01A4	SHE_OFIFORDDATA9 [R] W 00000000 00000000 00000000 00000000				
B200_01A8	SHE_OFIFORDDATA10 [R] W 00000000 00000000 00000000 00000000				
B200_01AC	SHE_OFIFORDDATA11 [R] W 00000000 00000000 00000000 00000000				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B200_01B0	SHE_OFIFORDDATA12 [R] W 00000000 00000000 00000000 00000000				
B200_01B4	SHE_OFIFORDDATA13 [R] W 00000000 00000000 00000000 00000000				
B200_01B8	SHE_OFIFORDDATA14 [R] W 00000000 00000000 00000000 00000000				
B200_01BC	SHE_OFIFORDDATA15 [R] W 00000000 00000000 00000000 00000000				
B200_01C0	SHE_OFIFORDDATA16 [R] W 00000000 00000000 00000000 00000000				
B200_01C4	SHE_OFIFORDDATA17 [R] W 00000000 00000000 00000000 00000000				
B200_01C8	SHE_OFIFORDDATA18 [R] W 00000000 00000000 00000000 00000000				
B200_01CC	SHE_OFIFORDDATA19 [R] W 00000000 00000000 00000000 00000000				
B200_01D0	SHE_OFIFORDDATA20 [R] W 00000000 00000000 00000000 00000000				
B200_01D4	SHE_OFIFORDDATA21 [R] W 00000000 00000000 00000000 00000000				
B200_01D8	SHE_OFIFORDDATA22 [R] W 00000000 00000000 00000000 00000000				
B200_01DC	SHE_OFIFORDDATA23 [R] W 00000000 00000000 00000000 00000000				
B200_01E0	SHE_OFIFORDDATA24 [R] W 00000000 00000000 00000000 00000000				
B200_01E4	SHE_OFIFORDDATA25 [R] W 00000000 00000000 00000000 00000000				
B200_01E8	SHE_OFIFORDDATA26 [R] W 00000000 00000000 00000000 00000000				
B200_01EC	SHE_OFIFORDDATA27 [R] W 00000000 00000000 00000000 00000000				
B200_01F0	SHE_OFIFORDDATA28 [R] W 00000000 00000000 00000000 00000000				
B200_01F4	SHE_OFIFORDDATA29 [R] W 00000000 00000000 00000000 00000000				
B200_01F8	SHE_OFIFORDDATA30 [R] W 00000000 00000000 00000000 00000000				
B200_01FC	SHE_OFIFORDDATA31 [R] W 00000000 00000000 00000000 00000000				
B200_0200 to B20F_03FC	-				
B200_0400	MPUXSHE_CTRL0 [R/W] B 00000000 00000000 00000001 00000000				MPU AXI
B200_0404	MPUXSHE_NMIEN [R/W] B,H,W 00000000 00000000 00000000 00000001				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B200_0408	MPUXSHE_WERRC [R] B,H,W 00000000 00000000 00000XXX XXXXXXX0				
B200_040C	MPUXSHE_WERRA [R] B,H,W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
B200_0410	MPUXSHE_RERRC [R] B,H,W 00000000 00000000 00000XXX XXXXXXX0				
B200_0414	MPUXSHE_RERRA [R] B,H,W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
B200_0418	MPUXSHE_CTRL1 [R/W] B 00000000 00000000 00000000 00000000				
B200_041C	MPUXSHE_SADDR1 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B200_0420	MPUXSHE_EADDR1 [R/W] B,H,W 00000000 00000000 00000000 01111111				
B200_0424	MPUXSHE_CTRL2 [R/W] B 00000000 00000000 00000000 00000000				
B200_0428	MPUXSHE_SADDR2 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B200_042C	MPUXSHE_EADDR2 [R/W] B,H,W 00000000 00000000 00000000 01111111				
B200_0430	MPUXSHE_CTRL3 [R/W] B 00000000 00000000 00000000 00000000				
B200_0434	MPUXSHE_SADDR3 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B200_0438	MPUXSHE_EADDR3 [R/W] B,H,W 00000000 00000000 00000000 01111111				
B200_043C	MPUXSHE_CTRL4 [R/W] B 00000000 00000000 00000000 00000000				
B200_0440	MPUXSHE_SADDR4 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B200_0444	MPUXSHE_EADDR4 [R/W] B,H,W 00000000 00000000 00000000 01111111				
B200_0448	MPUXSHE_CTRL5 [R/W] B 00000000 00000000 00000000 00000000				
B200_044C	MPUXSHE_SADDR5 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B200_0450	MPUXSHE_EADDR5 [R/W] B,H,W 00000000 00000000 00000000 01111111				
B200_0454	MPUXSHE_CTRL6 [R/W] B 00000000 00000000 00000000 00000000				
B200_0458	MPUXSHE_SADDR6 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B200_045C	MPUXSHE_EADDR6 [R/W] B,H,W 00000000 00000000 00000000 01111111				
B200_0460	MPUXSHE_CTRL7 [R/W] B 00000000 00000000 00000000 00000000				
B200_0464	MPUXSHE_SADDR7 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B200_0468	MPUXSHE_EADDR7 [R/W] B,H,W 00000000 00000000 00000000 01111111				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B200_046C	MPUXSHE_CTRL8 [R/W] B 00000000 00000000 00000000 00000000				
B200_0470	MPUXSHE_SADDR8 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B200_0474	MPUXSHE_EADDR8 [R/W] B,H,W 00000000 00000000 00000000 01111111				
B200_0478	MPUXSHE_UNLOCK [R/W] W 00000000 00000000 00000000 00000000				
B200_047C	MPUXSHE_MID [R] B,H,W 00000000 00000000 00000000 00000000				
B200_0480 to B20F_FFFC	-				
B210_0000 to B46F_FFFC	-				Reserved
B470_0000	DMA0_A0 [R/W] B,H,W 00000000 00001111 00000000 00000000				DMAC #0
B470_0004	DMA0_B0 [R/W] B,H,W 00000000 00000000 00110011 01111111				
B470_0008	DMA0_SA0 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_000C	DMA0_DA0 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_0010	DMA0_C0 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_0014	DMA0_D0 [R/W] B 00000000 00000000 00000000 00000000				
B470_0018	DMA0_SASHDW0 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_001C	DMA0_DASHDW0 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_0020	DMA0_E0 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_0024 to B470_003C	-				
B470_0040	DMA0_A1 [R/W] B,H,W 00000000 00001111 00000000 00000000				
B470_0044	DMA0_B1 [R/W] B,H,W 00000000 00000000 00110011 01111111				
B470_0048	DMA0_SA1 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_004C	DMA0_DA1 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_0050	DMA0_C1 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_0054	DMA0_D1 [R/W] B 00000000 00000000 00000000 00000000				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B470_0058	DMA0_SASHDW1 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_005C	DMA0_DASHDW1 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_0060	DMA0_E1 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_0064 to B470_007C	-				
B470_0080	DMA0_A2 [R/W] B,H,W 00000000 00001111 00000000 00000000				
B470_0084	DMA0_B2 [R/W] B,H,W 00000000 00000000 00110011 01111111				
B470_0088	DMA0_SA2 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_008C	DMA0_DA2 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_0090	DMA0_C2 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_0094	DMA0_D2 [R/W] B 00000000 00000000 00000000 00000000				
B470_0098	DMA0_SASHDW2 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_009C	DMA0_DASHDW2 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_00A0	DMA0_E2 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_00A4 to B470_00BC	-				
B470_00C0	DMA0_A3 [R/W] B,H,W 00000000 00001111 00000000 00000000				
B470_00C4	DMA0_B3 [R/W] B,H,W 00000000 00000000 00110011 01111111				
B470_00C8	DMA0_SA3 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_00CC	DMA0_DA3 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_00D0	DMA0_C3 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_00D4	DMA0_D3 [R/W] B 00000000 00000000 00000000 00000000				
B470_00D8	DMA0_SASHDW3 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_00DC	DMA0_DASHDW3 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_00E0	DMA0_E3 [R/W] B,H,W 00000000 00000000 00000000 00000000				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B470_00E4 to B470_00FC	-				
B470_0100	DMA0_A4 [R/W] B,H,W 00000000 00001111 00000000 00000000				
B470_0104	DMA0_B4 [R/W] B,H,W 00000000 00000000 00110011 01111111				
B470_0108	DMA0_SA4 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_010C	DMA0_DA4 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_0110	DMA0_C4 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_0114	DMA0_D4 [R/W] B 00000000 00000000 00000000 00000000				
B470_0118	DMA0_SASHDW4 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_011C	DMA0_DASHDW4 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_0120	DMA0_E4 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_0124 to B470_013C	-				
B470_0140	DMA0_A5 [R/W] B,H,W 00000000 00001111 00000000 00000000				
B470_0144	DMA0_B5 [R/W] B,H,W 00000000 00000000 00110011 01111111				
B470_0148	DMA0_SA5 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_014C	DMA0_DA5 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_0150	DMA0_C5 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_0154	DMA0_D5 [R/W] B 00000000 00000000 00000000 00000000				
B470_0158	DMA0_SASHDW5 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_015C	DMA0_DASHDW5 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_0160	DMA0_E5 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_0164 to B470_017C	-				
B470_0180	DMA0_A6 [R/W] B,H,W 00000000 00001111 00000000 00000000				
B470_0184	DMA0_B6 [R/W] B,H,W 00000000 00000000 00110011 01111111				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B470_0188	DMA0_SA6 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_018C	DMA0_DA6 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_0190	DMA0_C6 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_0194	DMA0_D6 [R/W] B 00000000 00000000 00000000 00000000				
B470_0198	DMA0_SASHDW6 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_019C	DMA0_DASHDW6 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_01A0	DMA0_E6 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_01A4 to B470_01BC	-				
B470_01C0	DMA0_A7 [R/W] B,H,W 00000000 00001111 00000000 00000000				
B470_01C4	DMA0_B7 [R/W] B,H,W 00000000 00000000 00110011 01111111				
B470_01C8	DMA0_SA7 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_01CC	DMA0_DA7 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_01D0	DMA0_C7 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_01D4	DMA0_D7 [R/W] B 00000000 00000000 00000000 00000000				
B470_01D8	DMA0_SASHDW7 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_01DC	DMA0_DASHDW7 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_01E0	DMA0_E7 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_01E4 to B470_01FC	-				
B470_0200	DMA0_A8 [R/W] B,H,W 00000000 00001111 00000000 00000000				
B470_0204	DMA0_B8 [R/W] B,H,W 00000000 00000000 00110011 01111111				
B470_0208	DMA0_SA8 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_020C	DMA0_DA8 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_0210	DMA0_C8 [R/W] B,H,W 00000000 00000000 00000000 00000000				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B470_0214	DMA0_D8 [R/W] B 00000000 00000000 00000000 00000000				
B470_0218	DMA0_SASHDW8 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_021C	DMA0_DASHDW8 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_0220	DMA0_E8 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_0224 to B470_023C	-				
B470_0240	DMA0_A9 [R/W] B,H,W 00000000 00001111 00000000 00000000				
B470_0244	DMA0_B9 [R/W] B,H,W 00000000 00000000 00110011 01111111				
B470_0248	DMA0_SA9 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_024C	DMA0_DA9 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_0250	DMA0_C9 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_0254	DMA0_D9 [R/W] B 00000000 00000000 00000000 00000000				
B470_0258	DMA0_SASHDW9 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_025C	DMA0_DASHDW9 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_0260	DMA0_E9 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_0264 to B470_027C	-				
B470_0280	DMA0_A10 [R/W] B,H,W 00000000 00001111 00000000 00000000				
B470_0284	DMA0_B10 [R/W] B,H,W 00000000 00000000 00110011 01111111				
B470_0288	DMA0_SA10 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_028C	DMA0_DA10 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_0290	DMA0_C10 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_0294	DMA0_D10 [R/W] B 00000000 00000000 00000000 00000000				
B470_0298	DMA0_SASHDW10 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_029C	DMA0_DASHDW10 [R/W] B,H,W 00000000 00000000 00000000 00000000				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B470_02A0	DMA0_E10 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_02A4 to B470_02BC	-				
B470_02C0	DMA0_A11 [R/W] B,H,W 00000000 00001111 00000000 00000000				
B470_02C4	DMA0_B11 [R/W] B,H,W 00000000 00000000 00110011 01111111				
B470_02C8	DMA0_SA11 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_02CC	DMA0_DA11 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_02D0	DMA0_C11 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_02D4	DMA0_D11 [R/W] B 00000000 00000000 00000000 00000000				
B470_02D8	DMA0_SASHDW11 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_02DC	DMA0_DASHDW11 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_02E0	DMA0_E11 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_02E4 to B470_02FC	-				
B470_0300	DMA0_A12 [R/W] B,H,W 00000000 00001111 00000000 00000000				
B470_0304	DMA0_B12 [R/W] B,H,W 00000000 00000000 00110011 01111111				
B470_0308	DMA0_SA12 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_030C	DMA0_DA12 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_0310	DMA0_C12 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_0314	DMA0_D12 [R/W] B 00000000 00000000 00000000 00000000				
B470_0318	DMA0_SASHDW12 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_031C	DMA0_DASHDW12 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_0320	DMA0_E12 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_0324 to B470_033C	-				
B470_0340	DMA0_A13 [R/W] B,H,W 00000000 00001111 00000000 00000000				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B470_0344	DMA0_B13 [R/W] B,H,W 00000000 00000000 00110011 01111111				
B470_0348	DMA0_SA13 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_034C	DMA0_DA13 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_0350	DMA0_C13 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_0354	DMA0_D13 [R/W] B 00000000 00000000 00000000 00000000				
B470_0358	DMA0_SASHDW13 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_035C	DMA0_DASHDW13 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_0360	DMA0_E13 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_0364 to B470_037C	-				
B470_0380	DMA0_A14 [R/W] B,H,W 00000000 00001111 00000000 00000000				
B470_0384	DMA0_B14 [R/W] B,H,W 00000000 00000000 00110011 01111111				
B470_0388	DMA0_SA14 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_038C	DMA0_DA14 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_0390	DMA0_C14 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_0394	DMA0_D14 [R/W] B 00000000 00000000 00000000 00000000				
B470_0398	DMA0_SASHDW14 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_039C	DMA0_DASHDW14 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_03A0	DMA0_E14 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_03A4 to B470_03BC	-				
B470_03C0	DMA0_A15 [R/W] B,H,W 00000000 00001111 00000000 00000000				
B470_03C4	DMA0_B15 [R/W] B,H,W 00000000 00000000 00110011 01111111				
B470_03C8	DMA0_SA15 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_03CC	DMA0_DA15 [R/W] B,H,W 00000000 00000000 00000000 00000000				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B470_03D0	DMA0_C15 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_03D4	DMA0_D15 [R/W] B 00000000 00000000 00000000 00000000				
B470_03D8	DMA0_SASHDW15 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_03DC	DMA0_DASHDW15 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_03E0	DMA0_E15 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_03E4 to B470_03FC	-				
B470_0400 to B470_0FFC	-				
B470_1000	DMA0_R [R/W] B,H,W 01000000 00000000 00000000 00000001				
B470_1004	DMA0_DIRQ1 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_1008	DMA0_DIRQ2 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_100C	DMA0_EDIRQ1 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_1010	DMA0_EDIRQ2 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_1014	DMA0_ID [R/W] B,H,W 00000000 00000001 00000101 00000000				
B470_1018 to B470_201C	-				
B4702020 + ((m-8)*0x04)	DMA0__CMICICm(m=8 - 331) [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_2530 to B470_27FC	-				
B470_2800	DMA0_CMCHIC0 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B470_2804	DMA0_CMCHIC1 [R/W] B,H,W 00000000 00000000 00000000 00000001				
B470_2808	DMA0_CMCHIC2 [R/W] B,H,W 00000000 00000000 00000000 00000010				
B470_280C	DMA0_CMCHIC3 [R/W] B,H,W 00000000 00000000 00000000 00000011				
B470_2810	DMA0_CMCHIC4 [R/W] B,H,W 00000000 00000000 00000000 00000100				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B470_2814	DMA0_CMCHIC5 [R/W] B,H,W 00000000 00000000 00000000 00000101				
B470_2818	DMA0_CMCHIC6 [R/W] B,H,W 00000000 00000000 00000000 00000110				
B470_281C	DMA0_CMCHIC7 [R/W] B,H,W 00000000 00000000 00000000 00000111				
B470_2820	DMA0_CMCHIC8 [R/W] B,H,W 00000000 00000000 00000000 00001000				
B470_2824	DMA0_CMCHIC9 [R/W] B,H,W 00000000 00000000 00000000 00001001				
B470_2828	DMA0_CMCHIC10 [R/W] B,H,W 00000000 00000000 00000000 00001010				
B470_282C	DMA0_CMCHIC11 [R/W] B,H,W 00000000 00000000 00000000 00001011				
B470_2830	DMA0_CMCHIC12 [R/W] B,H,W 00000000 00000000 00000000 00001100				
B470_2834	DMA0_CMCHIC13 [R/W] B,H,W 00000000 00000000 00000000 00001101				
B470_2838	DMA0_CMCHIC14 [R/W] B,H,W 00000000 00000000 00000000 00001110				
B470_283C	DMA0_CMCHIC15 [R/W] B,H,W 00000000 00000000 00000000 00001111				
B470_2840 to B470_3FFC	-				
B470_4000 to B470_FFFC	-				Reserved
B471_0000	MPUH0_CTRL0 [R/W] B,H,W 00000000 00000000 00000001 00000000				MPUH0
B471_0004	MPUH0_NMIEN [R/W] B,H,W 00000000 00000000 00000000 00000001				
B471_0008	MPUH0_MERRC [R] B,H,W 00000000 00000000 00000000 00000000				
B471_000C	MPUH0_MERRA [R] B,H,W 00000000 00000000 00000000 00000000				
B471_0010	MPUH0_CTRL1 [R/W] B 00000000 00000000 00000000 00000000				
B471_0014	MPUH0_SADDR1 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B471_0018	MPUH0_EADDR1 [R/W] B,H,W 00000000 00000000 00000000 01111111				
B471_001C	MPUH0_CTRL2 [R/W] B 00000000 00000000 00000000 00000000				
B471_0020	MPUH0_SADDR2 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B471_0024	MPUH0_EADDR2 [R/W] B,H,W 00000000 00000000 00000000 01111111				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B471_0028	MPUH0_CTRL3 [R/W] B 00000000 00000000 00000000 00000000				
B471_002C	MPUH0_SADDR3 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B471_0030	MPUH0_EADDR3 [R/W] B,H,W 00000000 00000000 00000000 01111111				
B471_0034	MPUH0_CTRL4 [R/W] B 00000000 00000000 00000000 00000000				
B471_0038	MPUH0_SADDR4 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B471_003C	MPUH0_EADDR4 [R/W] B,H,W 00000000 00000000 00000000 01111111				
B471_0040	MPUH0_CTRL5 [R/W] B 00000000 00000000 00000000 00000000				
B471_0044	MPUH0_SADDR5 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B471_0048	MPUH0_EADDR5 [R/W] B,H,W 00000000 00000000 00000000 01111111				
B471_004C	MPUH0_CTRL6 [R/W] B 00000000 00000000 00000000 00000000				
B471_0050	MPUH0_SADDR6 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B471_0054	MPUH0_EADDR6 [R/W] B,H,W 00000000 00000000 00000000 01111111				
B471_0058	MPUH0_CTRL7 [R/W] B 00000000 00000000 00000000 00000000				
B471_005C	MPUH0_SADDR7 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B471_0060	MPUH0_EADDR7 [R/W] B,H,W 00000000 00000000 00000000 01111111				
B471_0064	MPUH0_CTRL8 [R/W] B 00000000 00000000 00000000 00000000				
B471_0068	MPUH0_SADDR8 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B471_006C	MPUH0_EADDR8 [R/W] B,H,W 00000000 00000000 00000000 01111111				
B471_0070	MPUH0_UNLOCK [R/W] W 00000000 00000000 00000000 00000000				
B471_0074	MPUH0_MID [R] B,H,W 00000000 00001101 00000000 00000000				
B471_0078 to B471_010C	-				
B471_0110	MPUH0_CTRL9 [R/W] B 00000000 00000000 00000000 00000000				
B471_0114	MPUH0_SADDR9 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B471_0118	MPUH0_EADDR9 [R/W] B,H,W 00000000 00000000 00000000 01111111				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B471_011C	MPUH0_CTRL10 [R/W] B 00000000 00000000 00000000 00000000				
B471_0120	MPUH0_SADDR10 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B471_0124	MPUH0_EADDR10 [R/W] B,H,W 00000000 00000000 00000000 01111111				
B471_0128	MPUH0_CTRL11 [R/W] B 00000000 00000000 00000000 00000000				
B471_012C	MPUH0_SADDR11 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B471_0130	MPUH0_EADDR11 [R/W] B,H,W 00000000 00000000 00000000 01111111				
B471_0134	MPUH0_CTRL12 [R/W] B 00000000 00000000 00000000 00000000				
B471_0138	MPUH0_SADDR12 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B471_013C	MPUH0_EADDR12 [R/W] B,H,W 00000000 00000000 00000000 01111111				
B471_0140	MPUH0_CTRL13 [R/W] B 00000000 00000000 00000000 00000000				
B471_0144	MPUH0_SADDR13 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B471_0148	MPUH0_EADDR13 [R/W] B,H,W 00000000 00000000 00000000 01111111				
B471_014C	MPUH0_CTRL14 [R/W] B 00000000 00000000 00000000 00000000				
B471_0150	MPUH0_SADDR14 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B471_0154	MPUH0_EADDR14 [R/W] B,H,W 00000000 00000000 00000000 01111111				
B471_0158	MPUH0_CTRL15 [R/W] B 00000000 00000000 00000000 00000000				
B471_015C	MPUH0_SADDR15 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B471_0160	MPUH0_EADDR15 [R/W] B,H,W 00000000 00000000 00000000 01111111				
B471_0164	MPUH0_CTRL16 [R/W] B 00000000 00000000 00000000 00000000				
B471_0168	MPUH0_SADDR16 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B471_016C	MPUH0_EADDR16 [R/W] B,H,W 00000000 00000000 00000000 01111111				
B471_0170 to B471_0FFC	-				
B471_1000 to B471_3FFC	-				Reserved

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B471_4000	DMAA0_ASR0 [R] B,H,W 00000000 00000000 00000000 00000000				DMA_COMPLEX_SUBSYSTEM0
B471_4004	DMAA0_ASR1 [R] B,H,W 00000000 00000000 00000000 00000000				
B471_4008	DMAA0_ASR2 [R] B,H,W 00000000 00000000 00000000 00000000				
B471_400C	DMAA0_ASR3 [R] B,H,W 00000000 00000000 00000000 00000000				
B471_4010	DMAA0_ASR4 [R] B,H,W 00000000 00000000 00000000 00000000				
B471_4014	DMAA0_ASR5 [R] B,H,W 00000000 00000000 00000000 00000000				
B471_4018	DMAA0_ASR6 [R] B,H,W 00000000 00000000 00000000 00000000				
B471_401C	DMAA0_ASR7 [R] B,H,W 00000000 00000000 00000000 00000000				
B471_4020	DMAA0_ASR8 [R] B,H,W 00000000 00000000 00000000 00000000				
B471_4024	DMAA0_ASR9 [R] B,H,W 00000000 00000000 00000000 00000000				
B471_4028	DMAA0_ASR10 [R] B,H,W 00000000 00000000 00000000 00000000				
B471_402C	DMAA0_ASR11 [R] B,H,W 00000000 00000000 00000000 00000000				
B471_4030	DMAA0_ASR12 [R] B,H,W 00000000 00000000 00000000 00000000				
B471_4034	DMAA0_ASR13 [R] B,H,W 00000000 00000000 00000000 00000000				
B471_4038	DMAA0_ASR14 [R] B,H,W 00000000 00000000 00000000 00000000				
B471_403C	DMAA0_ASR15 [R] B,H,W 00000000 00000000 00000000 00000000				
B471_4040	DMAA0_ASR16 [R] B,H,W 00000000 00000000 00000000 00000000				
B471_4044	DMAA0_ASR17 [R] B,H,W 00000000 00000000 00000000 00000000				
B471_4048	DMAA0_ASR18 [R] B,H,W 00000000 00000000 00000000 00000000				
B471_404C	DMAA0_ASR19 [R] B,H,W 00000000 00000000 00000000 00000000				
B471_4050 to B471_407C	-				
B471_4080	DMAA0_CMCHICRDB0 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B471_4084	DMAA0_CMCHICRDB1 [R/W] B,H,W 00000000 00000000 00000000 00000000				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B471_4088	DMAA0_CMCHICRDB2 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B471_408C	DMAA0_CMCHICRDB3 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B471_4090	DMAA0_CMCHICRDB4 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B471_4094	DMAA0_CMCHICRDB5 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B471_4098	DMAA0_CMCHICRDB6 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B471_409C	DMAA0_CMCHICRDB7 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B471_40A0 to B471_40FC	-				
B471_4100	DMAA0_RTTSR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B471_4104 to B471_411C	-				
B471_4120	-			DMAA0_RTSSSR [R/W] B,H,W 00000000	
B471_4124 to B471_47FC	-				
B471_4800	-			DMAA0_RLT0_DMACFG [R/W] B,H,W 00000000	
B471_4804	-				
B471_4808	DMAA0_RLT0_TMCSR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B471_480C	-				
B471_4810	DMAA0_RLT0_TMRLR [R/W] W,DW XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
B471_4814	DMAA0_RLT0_TMR [R/W] W,DW XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
B471_4818	-				
B471_481C	-				
B471_4820	-			DMAA0_RLT1_DMACFG [R/W] B,H,W 00000000	
B471_4824	-				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B471_4828	DMAA0_RLT1_TMCSR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B471_482C	-				
B471_4830	DMAA0_RLT1_TMRLR [R/W] W,DW XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
B471_4834	DMAA0_RLT1_TMR [R/W] W,DW XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
B471_4838	-				
B471_483C	-				
B471_4840	-			DMAA0_RLT2_DMACFG [R/W] B,H,W 00000000	
B471_4844	-				
B471_4848	DMAA0_RLT2_TMCSR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B471_484C	-				
B471_4850	DMAA0_RLT2_TMRLR [R/W] W,DW XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
B471_4854	DMAA0_RLT2_TMR [R/W] W,DW XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
B471_4858	-				
B471_485C	-				
B471_4860	-			DMAA0_RLT3_DMACFG [R/W] B,H,W 00000000	
B471_4864	-				
B471_4868	DMAA0_RLT3_TMCSR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B471_486C	-				
B471_4870	DMAA0_RLT3_TMRLR [R/W] W,DW XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
B471_4874	DMAA0_RLT3_TMR [R/W] W,DW XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
B471_4878	-				
B471_487C	-				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B471_4880 to B471_4FFC	-				
B471_5000 to B471_7FFC	-				Reserved
B471_8000	-			CRC00_CRCCR [R/W] B,H,W 00000000	CRC#0
B471_8004	CRC00_CRCINIT [R/W] B,H,W 11111111 11111111 11111111 11111111				
B471_8008	CRC00_CRCIN [R/W] B,H,W 00000000 00000000 00000000 00000000				
B471_800C	CRC00_CRCR [R/W] B,H,W 11111111 11111111 11111111 11111111				
B471_8010 to B471_83FC	-				
B471_8400	-			CRC01_CRCCR [R/W] B,H,W 00000000	CRC#1
B471_8404	CRC01_CRCINIT [R/W] B,H,W 11111111 11111111 11111111 11111111				
B471_8408	CRC01_CRCIN [R/W] B,H,W 00000000 00000000 00000000 00000000				
B471_840C	CRC01_CRCR [R/W] B,H,W 11111111 11111111 11111111 11111111				
B471_8410 to B471_87FC	-				
B471_8800	-			CRC02_CRCCR [R/W] B,H,W 00000000	CRC#2
B471_8804	CRC02_CRCINIT [R/W] B,H,W 11111111 11111111 11111111 11111111				
B471_8808	CRC02_CRCIN [R/W] B,H,W 00000000 00000000 00000000 00000000				
B471_880C	CRC02_CRCR [R/W] B,H,W 11111111 11111111 11111111 11111111				
B471_8810 to B471_8BFC	-				
B471_8C00	-			CRC03_CRCCR [R/W] B,H,W 00000000	CRC#3
B471_8C04	CRC03_CRCINIT [R/W] B,H,W 11111111 11111111 11111111 11111111				
B471_8C08	CRC03_CRCIN [R/W] B,H,W 00000000 00000000 00000000 00000000				
B471_8C0C	CRC03_CRCR [R/W] B,H,W 11111111 11111111 11111111 11111111				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B471_8C10 to B471_8FFC	-				
B471_9000 to B473_7FFC	-				
B473_8000	GPIO_POSR0 [R/W] B,H,W 00000000 00000000 00000000 00000000				GPIO
B473_8004	GPIO_POCR0 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B473_8008	GPIO_DDSR0 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B473_800C	GPIO_DDCR0 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B473_8010	GPIO_POSR1 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B473_8014	GPIO_POCR1 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B473_8018	GPIO_DDSR1 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B473_801C	GPIO_DDCR1 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B473_8020	GPIO_POSR2 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B473_8024	GPIO_POCR2 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B473_8028	GPIO_DDSR2 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B473_802C	GPIO_DDCR2 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B473_8030	GPIO_POSR3 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B473_8034	GPIO_POCR3 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B473_8038	GPIO_DDSR3 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B473_803C	GPIO_DDCR3 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B473_8040	GPIO_POSR4 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B473_8044	GPIO_POCR4 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B473_8048	GPIO_DDSR4 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B473_804C	GPIO_DDCR4 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B473_8050 to B473_81FC	-				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B473_8200	GPIO_PODR0 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B473_8204	GPIO_DDR0 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B473_8208	GPIO_PODR1 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B473_820C	GPIO_DDR1 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B473_8210	GPIO_PODR2 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B473_8214	GPIO_DDR2 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B473_8218	GPIO_PODR3 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B473_821C	GPIO_DDR3 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B473_8220	GPIO_PODR4 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B473_8224	GPIO_DDR4 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B473_8228 to B473_82FC	-				
B473_8300	GPIO_PIDR0 [R/W] B,H,W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
B473_8304	GPIO_PIDR1 [R/W] B,H,W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
B473_8308	GPIO_PIDR2 [R/W] B,H,W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
B473_830C	GPIO_PIDR3 [R/W] B,H,W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
B473_8310	GPIO_PIDR4 [R/W] B,H,W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
B473_8314 to B473_83FC	-				
B473_8400	GPIO_PORTEN [R/W] B,H,W 00000000 00000000 00000000 00000000				
B473_8404	GPIO_KEYCDR [R/W] W 00000000 00000000 00000000 00000000				
B473_8408 to B473_FFFC	-				
B474_0000	PPC_PCFGR001 [R/W] B,H,W 00X00000 00000000		PPC_PCFGR000 [R/W] B,H,W 00X00000 00000000		PPC
B474_0004	PPC_PCFGR003 [R/W] B,H,W 00X00000 00000000		-		
B474_0008	PPC_PCFGR005 [R/W] B,H,W 00X00000 00000000		-		

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B474_000C	PPC_PCFGR007 [R/W] B,H,W 00X00000 00000000		PPC_PCFGR006 [R/W] B,H,W 00X00000 00000000		
B474_0010	PPC_PCFGR009 [R/W] B,H,W 00X00000 00000000		PPC_PCFGR008 [R/W] B,H,W 00X00000 00000000		
B474_0014	-		PPC_PCFGR010 [R/W] B,H,W 00X00000 00000000		
B474_0018	PPC_PCFGR013 [R/W] B,H,W 00X00000 00000000		PPC_PCFGR012 [R/W] B,H,W 00X00000 00000000		
B474_001C	PPC_PCFGR015 [R/W] B,H,W 00X00000 00000000		-		
B474_0020	PPC_PCFGR017 [R/W] B,H,W 00X00000 00000000		PPC_PCFGR016 [R/W] B,H,W 00X00000 00000000		
B474_0024	PPC_PCFGR019 [R/W] B,H,W 00X00000 00000000		PPC_PCFGR018 [R/W] B,H,W 00X00000 00000000		
B474_0028	PPC_PCFGR021 [R/W] B,H,W 00X00000 00000000		PPC_PCFGR020 [R/W] B,H,W 00X00000 00000000		
B474_002C	PPC_PCFGR023 [R/W] B,H,W 00X00000 00000000		PPC_PCFGR022 [R/W] B,H,W 00X00000 00000000		
B474_0030	-		PPC_PCFGR024 [R/W] B,H,W 00X00000 00000000		
B474_0034	PPC_PCFGR027 [R/W] B,H,W 00X00000 00000000		-		
B474_0038	PPC_PCFGR029 [R/W] B,H,W 00X00000 00000000		PPC_PCFGR028 [R/W] B,H,W 00X00000 00000000		
B474_003C	PPC_PCFGR031 [R/W] B,H,W 00X00000 00000000		PPC_PCFGR030 [R/W] B,H,W 00X00000 00000000		
B474_0040	PPC_PCFGR101 [R/W] B,H,W 00X00000 00000000		PPC_PCFGR100 [R/W] B,H,W 00X00000 00000000		
B474_0044	PPC_PCFGR103 [R/W] B,H,W 00X00000 00000000		-		
B474_0048	PPC_PCFGR105 [R/W] B,H,W 00X00000 00000000		-		
B474_004C	PPC_PCFGR107 [R/W] B,H,W 00X00000 00000000		PPC_PCFGR106 [R/W] B,H,W 00X00000 00000000		
B474_0050	PPC_PCFGR109 [R/W] B,H,W 00X00000 00000000		PPC_PCFGR108 [R/W] B,H,W 00X00000 00000000		
B474_0054	-		-		
B474_0058	PPC_PCFGR113 [R/W] B,H,W 00X00000 00000000		PPC_PCFGR112 [R/W] B,H,W 00X00000 00000000		
B474_005C	PPC_PCFGR115 [R/W] B,H,W 00X00000 00000000		PPC_PCFGR114 [R/W] B,H,W 00X00000 00000000		
B474_0060	PPC_PCFGR117 [R/W] B,H,W 00X00000 00000000		-		
B474_0064	PPC_PCFGR119 [R/W] B,H,W 00X00000 00000000		PPC_PCFGR118 [R/W] B,H,W 00X00000 00000000		
B474_0068	-		PPC_PCFGR120 [R/W] B,H,W 00X00000 00000000		

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B474_006C	PPC_PCFGR123 [R/W] B,H,W 00X00000 00000000		PPC_PCFGR122 [R/W] B,H,W 00X00000 00000000		
B474_0070	-		-		
B474_0074	PPC_PCFGR127 [R/W] B,H,W 00X00000 00000000		PPC_PCFGR126 [R/W] B,H,W 00X00000 00000000		
B474_0078	PPC_PCFGR129 [R/W] B,H,W 00X00000 00000000		PPC_PCFGR128 [R/W] B,H,W 00X00000 00000000		
B474_007C	PPC_PCFGR131 [R/W] B,H,W 00X00000 00000000		PPC_PCFGR130 [R/W] B,H,W 00X00000 00000000		
B474_0080	-		-		
B474_0084	PPC_PCFGR203 [R/W] B,H,W 00X00000 00000000		PPC_PCFGR202 [R/W] B,H,W 00X00000 00000000		
B474_0088	PPC_PCFGR205 [R/W] B,H,W 00X00000 00000000		PPC_PCFGR204 [R/W] B,H,W 00X00000 00000000		
B474_008C	PPC_PCFGR207 [R/W] B,H,W 00X00000 00000000		PPC_PCFGR206 [R/W] B,H,W 00X00000 00000000		
B474_0090	PPC_PCFGR209 [R/W] B,H,W 00X00000 00000000		PPC_PCFGR208 [R/W] B,H,W 00X00000 00000000		
B474_0094	PPC_PCFGR211 [R/W] B,H,W 00X00000 00000000		PPC_PCFGR210 [R/W] B,H,W 00X00000 00000000		
B474_0098	PPC_PCFGR213 [R/W] B,H,W 00X00000 00000000		PPC_PCFGR212 [R/W] B,H,W 00X00000 00000000		
B474_009C	PPC_PCFGR215 [R/W] B,H,W 00X00000 00000000		PPC_PCFGR214 [R/W] B,H,W 00X00000 00000000		
B474_00A0	-		-		
B474_00A4	PPC_PCFGR219 [R/W] B,H,W 00X00000 00000000		PPC_PCFGR218 [R/W] B,H,W 00X00000 00000000		
B474_00A8	-		PPC_PCFGR220 [R/W] B,H,W 00X00000 00000000		
B474_00AC	PPC_PCFGR223 [R/W] B,H,W 00X00000 00000000		PPC_PCFGR222 [R/W] B,H,W 00X00000 00000000		
B474_00B0	PPC_PCFGR225 [R/W] B,H,W 00X00000 00000000		PPC_PCFGR224 [R/W] B,H,W 00X00000 00000000		
B474_00B4	-		PPC_PCFGR226 [R/W] B,H,W 00X00000 00000000		
B474_00B8	PPC_PCFGR229 [R/W] B,H,W 00X00000 00000000		-		
B474_00BC	PPC_PCFGR231 [R/W] B,H,W 00X00000 00000000		PPC_PCFGR230 [R/W] B,H,W 00X00000 00000000		
B474_00C0	PPC_PCFGR301 [R/W] B,H,W 00X00000 00000000		PPC_PCFGR300 [R/W] B,H,W 00X00000 00000000		
B474_00C4	-		PPC_PCFGR302 [R/W] B,H,W 00X00000 00000000		
B474_00C8	PPC_PCFGR305 [R/W] B,H,W 00X00000 00000000		PPC_PCFGR304 [R/W] B,H,W 00X00000 00000000		

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B474_00CC	PPC_PCFGR307 [R/W] B,H,W 00X00000 00000000		-		
B474_00D0	PPC_PCFGR309 [R/W] B,H,W 00X00000 00000000		PPC_PCFGR308 [R/W] B,H,W 00X00000 00000000		
B474_00D4	-		-		
B474_00D8	PPC_PCFGR313 [R/W] B,H,W 00X00000 00000000		PPC_PCFGR312 [R/W] B,H,W 00X00000 00000000		
B474_00DC	PPC_PCFGR315 [R/W] B,H,W 00X00000 00000000		PPC_PCFGR314 [R/W] B,H,W 00X00000 00000000		
B474_00E0	PPC_PCFGR317 [R/W] B,H,W 00X00000 00000000		-		
B474_00E4	-		-		
B474_00E8	PPC_PCFGR321 [R/W] B,H,W 00X00000 00000000		-		
B474_00EC	PPC_PCFGR323 [R/W] B,H,W 00X00000 00000000		PPC_PCFGR322 [R/W] B,H,W 00X00000 00000000		
B474_00F0	-		PPC_PCFGR324 [R/W] B,H,W 00X00000 00000000		
B474_00F4	PPC_PCFGR327 [R/W] B,H,W 00X00000 00000000		-		
B474_00F8	-		-		
B474_00FC	PPC_PCFGR331 [R/W] B,H,W 00X00000 00000000		-		
B474_0100	PPC_PCFGR401 [R/W] B,H,W 00X00000 00000000		PPC_PCFGR400 [R/W] B,H,W 00X00000 00000000		
B474_0104	PPC_PCFGR403 [R/W] B,H,W 00X00000 00000000		PPC_PCFGR402 [R/W] B,H,W 00X00000 00000000		
B474_0108	PPC_PCFGR405 [R/W] B,H,W 00X00000 00000000		PPC_PCFGR404 [R/W] B,H,W 00X00000 00000000		
B474_010C	PPC_PCFGR407 [R/W] B,H,W 00X00000 00000000		PPC_PCFGR406 [R/W] B,H,W 00X00000 00000000		
B474_0110	PPC_PCFGR409 [R/W] B,H,W 00X00000 00000000		PPC_PCFGR408 [R/W] B,H,W 00X00000 00000000		
B474_0114	PPC_PCFGR411 [R/W] B,H,W 00X00000 00000000		-		
B474_0118	PPC_PCFGR413 [R/W] B,H,W 00X00000 00000000		-		
B474_011C	-		PPC_PCFGR414 [R/W] B,H,W 00X00000 00000000		
B474_0120	PPC_PCFGR417 [R/W] B,H,W 00X00000 00000000		PPC_PCFGR416 [R/W] B,H,W 00X00000 00000000		
B474_0124	-		PPC_PCFGR418 [R/W] B,H,W 00X00000 00000000		
B474_0128	PPC_PCFGR421 [R/W] B,H,W 00X00000 00000000		PPC_PCFGR420 [R/W] B,H,W 00X00000 00000000		

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B474_012C to B474_03FC	-				
B474_0400	PPC_KEYCDR [R/W] W 00000000 00000000 00000000 00000000				
B474_0404 to B474_7FFC	-				
B474_8000 to B474_8008	-				RIC
B474_800C	RIC_RESIN7 [R/W] B,H,W 00000000 00000000		RIC_RESIN6 [R/W] B,H,W 00000000 00000000		
B474_8010	-		RIC_RESIN8 [R/W] B,H,W 00000000 00000000		
B474_8014	-				
B474_8018	RIC_RESIN13 [R/W] B,H,W 00000000 00000000		RIC_RESIN12 [R/W] B,H,W 00000000 00000000		
B474_801C	RIC_RESIN15 [R/W] B,H,W 00000000 00000000		-		
B474_8020	RIC_RESIN17 [R/W] B,H,W 00000000 00000000		RIC_RESIN16 [R/W] B,H,W 00000000 00000000		
B474_8024	-		-		
B474_8028	RIC_RESIN21 [R/W] B,H,W 00000000 00000000		-		
B474_802C	-		RIC_RESIN22 [R/W] B,H,W 00000000 00000000		
B474_8030 to B474_8068	-				
B474_806C	-		-		
B474_8070	-		-		
B474_8074 to B474_8080	-				
B474_8084	RIC_RESIN67 [R/W] B,H,W 00000000 00000000		RIC_RESIN66 [R/W] B,H,W 00000000 00000000		
B474_8088	-		RIC_RESIN68 [R/W] B,H,W 00000000 00000000		
B474_808C	RIC_RESIN71 [R/W] B,H,W 00000000 00000000		RIC_RESIN70 [R/W] B,H,W 00000000 00000000		
B474_8090	RIC_RESIN73 [R/W] B,H,W 00000000 00000000		RIC_RESIN72 [R/W] B,H,W 00000000 00000000		
B474_8094	RIC_RESIN75 [R/W] B,H,W 00000000 00000000		RIC_RESIN74 [R/W] B,H,W 00000000 00000000		

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B474_8098	RIC_RESIN77 [R/W] B,H,W 00000000 00000000		RIC_RESIN76 [R/W] B,H,W 00000000 00000000		
B474_809C	RIC_RESIN79 [R/W] B,H,W 00000000 00000000		RIC_RESIN78 [R/W] B,H,W 00000000 00000000		
B474_80A0	RIC_RESIN81 [R/W] B,H,W 00000000 00000000		RIC_RESIN80 [R/W] B,H,W 00000000 00000000		
B474_80A4	RIC_RESIN83 [R/W] B,H,W 00000000 00000000		RIC_RESIN82 [R/W] B,H,W 00000000 00000000		
B474_80A8	RIC_RESIN85 [R/W] B,H,W 00000000 00000000		RIC_RESIN84 [R/W] B,H,W 00000000 00000000		
B474_80AC	RIC_RESIN87 [R/W] B,H,W 00000000 00000000		RIC_RESIN86 [R/W] B,H,W 00000000 00000000		
B474_80B0	RIC_RESIN89 [R/W] B,H,W 00000000 00000000		RIC_RESIN88 [R/W] B,H,W 00000000 00000000		
B474_80B4	RIC_RESIN91 [R/W] B,H,W 00000000 00000000		RIC_RESIN90 [R/W] B,H,W 00000000 00000000		
B474_80B8	RIC_RESIN93 [R/W] B,H,W 00000000 00000000		RIC_RESIN92 [R/W] B,H,W 00000000 00000000		
B474_80BC	RIC_RESIN95 [R/W] B,H,W 00000000 00000000		RIC_RESIN94 [R/W] B,H,W 00000000 00000000		
B474_80C0	RIC_RESIN97 [R/W] B,H,W 00000000 00000000		RIC_RESIN96 [R/W] B,H,W 00000000 00000000		
B474_80C4	RIC_RESIN99 [R/W] B,H,W 00000000 00000000		RIC_RESIN98 [R/W] B,H,W 00000000 00000000		
B474_80C8	RIC_RESIN101 [R/W] B,H,W 00000000 00000000		RIC_RESIN100 [R/W] B,H,W 00000000 00000000		
B474_80CC	RIC_RESIN103 [R/W] B,H,W 00000000 00000000		RIC_RESIN102 [R/W] B,H,W 00000000 00000000		
B474_80D0	RIC_RESIN105 [R/W] B,H,W 00000000 00000000		RIC_RESIN104 [R/W] B,H,W 00000000 00000000		
B474_80D4	RIC_RESIN107 [R/W] B,H,W 00000000 00000000		RIC_RESIN106 [R/W] B,H,W 00000000 00000000		
B474_80D8	RIC_RESIN109 [R/W] B,H,W 00000000 00000000		RIC_RESIN108 [R/W] B,H,W 00000000 00000000		
B474_80DC	RIC_RESIN111 [R/W] B,H,W 00000000 00000000		RIC_RESIN110 [R/W] B,H,W 00000000 00000000		
B474_80E0	RIC_RESIN113 [R/W] B,H,W 00000000 00000000		RIC_RESIN112 [R/W] B,H,W 00000000 00000000		
B474_80E4	RIC_RESIN115 [R/W] B,H,W 00000000 00000000		RIC_RESIN114 [R/W] B,H,W 00000000 00000000		
B474_80E8	RIC_RESIN117 [R/W] B,H,W 00000000 00000000		RIC_RESIN116 [R/W] B,H,W 00000000 00000000		
B474_80EC	RIC_RESIN119 [R/W] B,H,W 00000000 00000000		RIC_RESIN118 [R/W] B,H,W 00000000 00000000		
B474_80F0	RIC_RESIN121 [R/W] B,H,W 00000000 00000000		RIC_RESIN120 [R/W] B,H,W 00000000 00000000		
B474_80F4	RIC_RESIN123 [R/W] B,H,W 00000000 00000000		RIC_RESIN122 [R/W] B,H,W 00000000 00000000		

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B474_80F8	RIC_RESIN125 [R/W] B,H,W 00000000 00000000		RIC_RESIN124 [R/W] B,H,W 00000000 00000000		
B474_80FC	-		-		
B474_8100	RIC_RESIN129 [R/W] B,H,W 00000000 00000000		-		
B474_8104	RIC_RESIN131 [R/W] B,H,W 00000000 00000000		-		
B474_8108	-		RIC_RESIN132 [R/W] B,H,W 00000000 00000000		
B474_810C	RIC_RESIN135 [R/W] B,H,W 00000000 00000000		-		
B474_8110	-		RIC_RESIN136 [R/W] B,H,W 00000000 00000000		
B474_8114	RIC_RESIN139 [R/W] B,H,W 00000000 00000000		RIC_RESIN138 [R/W] B,H,W 00000000 00000000		
B474_8118	RIC_RESIN141 [R/W] B,H,W 00000000 00000000		RIC_RESIN140 [R/W] B,H,W 00000000 00000000		
B474_811C	RIC_RESIN143 [R/W] B,H,W 00000000 00000000		-		
B474_8120	RIC_RESIN145 [R/W] B,H,W 00000000 00000000		RIC_RESIN144 [R/W] B,H,W 00000000 00000000		
B474_8124	RIC_RESIN147 [R/W] B,H,W 00000000 00000000		RIC_RESIN146 [R/W] B,H,W 00000000 00000000		
B474_8128	RIC_RESIN149 [R/W] B,H,W 00000000 00000000		RIC_RESIN148 [R/W] B,H,W 00000000 00000000		
B474_812C	-		RIC_RESIN150 [R/W] B,H,W 00000000 00000000		
B474_8130	RIC_RESIN153 [R/W] B,H,W 00000000 00000000		-		
B474_8134	RIC_RESIN155 [R/W] B,H,W 00000000 00000000		RIC_RESIN154 [R/W] B,H,W 00000000 00000000		
B474_8138	RIC_RESIN157 [R/W] B,H,W 00000000 00000000		RIC_RESIN156 [R/W] B,H,W 00000000 00000000		
B474_813C	RIC_RESIN159 [R/W] B,H,W 00000000 00000000		RIC_RESIN158 [R/W] B,H,W 00000000 00000000		
B474_8140	RIC_RESIN161 [R/W] B,H,W 00000000 00000000		RIC_RESIN160 [R/W] B,H,W 00000000 00000000		
B474_8144	RIC_RESIN163 [R/W] B,H,W 00000000 00000000		RIC_RESIN162 [R/W] B,H,W 00000000 00000000		
B474_8148	RIC_RESIN165 [R/W] B,H,W 00000000 00000000		RIC_RESIN164 [R/W] B,H,W 00000000 00000000		
B474_814C	RIC_RESIN167 [R/W] B,H,W 00000000 00000000		RIC_RESIN166 [R/W] B,H,W 00000000 00000000		
B474_8150	RIC_RESIN169 [R/W] B,H,W 00000000 00000000		RIC_RESIN168 [R/W] B,H,W 00000000 00000000		
B474_8154	-		-		

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B474_8158	RIC_RESIN173 [R/W] B,H,W 00000000 00000000		RIC_RESIN172 [R/W] B,H,W 00000000 00000000		
B474_815C	RIC_RESIN175 [R/W] B,H,W 00000000 00000000		RIC_RESIN174 [R/W] B,H,W 00000000 00000000		
B474_8160	RIC_RESIN177 [R/W] B,H,W 00000000 00000000		RIC_RESIN176 [R/W] B,H,W 00000000 00000000		
B474_8164	RIC_RESIN179 [R/W] B,H,W 00000000 00000000		RIC_RESIN178 [R/W] B,H,W 00000000 00000000		
B474_8168	RIC_RESIN181 [R/W] B,H,W 00000000 00000000		-		
B474_816C	RIC_RESIN183 [R/W] B,H,W 00000000 00000000		RIC_RESIN182 [R/W] B,H,W 00000000 00000000		
B474_8170	RIC_RESIN185 [R/W] B,H,W 00000000 00000000		RIC_RESIN184 [R/W] B,H,W 00000000 00000000		
B474_8174	RIC_RESIN187 [R/W] B,H,W 00000000 00000000		RIC_RESIN186 [R/W] B,H,W 00000000 00000000		
B474_8178	-		RIC_RESIN188 [R/W] B,H,W 00000000 00000000		
B474_817C	-		RIC_RESIN190 [R/W] B,H,W 00000000 00000000		
B474_8180	RIC_RESIN193 [R/W] B,H,W 00000000 00000000		-		
B474_8184	RIC_RESIN195 [R/W] B,H,W 00000000 00000000		RIC_RESIN194 [R/W] B,H,W 00000000 00000000		
B474_8188	RIC_RESIN197 [R/W] B,H,W 00000000 00000000		RIC_RESIN196 [R/W] B,H,W 00000000 00000000		
B474_818C	-		-		
B474_8190	-		RIC_RESIN200 [R/W] B,H,W 00000000 00000000		
B474_8194	-		RIC_RESIN202 [R/W] B,H,W 00000000 00000000		
B474_8198	RIC_RESIN205 [R/W] B,H,W 00000000 00000000		-		
B474_819C	RIC_RESIN207 [R/W] B,H,W 00000000 00000000		-		
B474_81A0	RIC_RESIN209 [R/W] B,H,W 00000000 00000000		RIC_RESIN208 [R/W] B,H,W 00000000 00000000		
B474_81A4	RIC_RESIN211 [R/W] B,H,W 00000000 00000000		-		
B474_81A8	RIC_RESIN213 [R/W] B,H,W 00000000 00000000		RIC_RESIN212 [R/W] B,H,W 00000000 00000000		
B474_81AC	RIC_RESIN215 [R/W] B,H,W 00000000 00000000		-		
B474_81B0	RIC_RESIN217 [R/W] B,H,W 00000000 00000000		RIC_RESIN216 [R/W] B,H,W 00000000 00000000		
B474_81B4	RIC_RESIN219 [R/W] B,H,W 00000000 00000000		RIC_RESIN218 [R/W] B,H,W 00000000 00000000		

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B474_81B8	RIC_RESIN221 [R/W] B,H,W 00000000 00000000		RIC_RESIN220 [R/W] B,H,W 00000000 00000000		
B474_81BC to B474_81C4	-				
B474_81C8	RIC_RESIN229 [R/W] B,H,W 00000000 00000000		RIC_RESIN228 [R/W] B,H,W 00000000 00000000		
B474_81CC	-		-		
B474_81D0	RIC_RESIN233 [R/W] B,H,W 00000000 00000000		RIC_RESIN232 [R/W] B,H,W 00000000 00000000		
B474_81D4	RIC_RESIN235 [R/W] B,H,W 00000000 00000000		RIC_RESIN234 [R/W] B,H,W 00000000 00000000		
B474_81D8	-		-		
B474_81DC	-		-		
B474_81E0	RIC_RESIN241 [R/W] B,H,W 00000000 00000000		RIC_RESIN240 [R/W] B,H,W 00000000 00000000		
B474_81E4	RIC_RESIN243 [R/W] B,H,W 00000000 00000000		RIC_RESIN242 [R/W] B,H,W 00000000 00000000		
B474_81E8	-		-		
B474_81EC	RIC_RESIN247 [R/W] B,H,W 00000000 00000000		RIC_RESIN246 [R/W] B,H,W 00000000 00000000		
B474_81F0	RIC_RESIN249 [R/W] B,H,W 00000000 00000000		RIC_RESIN248 [R/W] B,H,W 00000000 00000000		
B474_81F4	RIC_RESIN251 [R/W] B,H,W 00000000 00000000		RIC_RESIN250 [R/W] B,H,W 00000000 00000000		
B474_81F8	RIC_RESIN253 [R/W] B,H,W 00000000 00000000		RIC_RESIN252 [R/W] B,H,W 00000000 00000000		
B474_81FC	-		-		
B474_8200	RIC_RESIN257 [R/W] B,H,W 00000000 00000000		RIC_RESIN256 [R/W] B,H,W 00000000 00000000		
B474_8204	RIC_RESIN259 [R/W] B,H,W 00000000 00000000		RIC_RESIN258 [R/W] B,H,W 00000000 00000000		
B474_8208	RIC_RESIN261 [R/W] B,H,W 00000000 00000000		RIC_RESIN260 [R/W] B,H,W 00000000 00000000		
B474_820C	RIC_RESIN263 [R/W] B,H,W 00000000 00000000		RIC_RESIN262 [R/W] B,H,W 00000000 00000000		
B474_8210	RIC_RESIN265 [R/W] B,H,W 00000000 00000000		RIC_RESIN264 [R/W] B,H,W 00000000 00000000		
B474_8214	RIC_RESIN267 [R/W] B,H,W 00000000 00000000		RIC_RESIN266 [R/W] B,H,W 00000000 00000000		
B474_8218	RIC_RESIN269 [R/W] B,H,W 00000000 00000000		RIC_RESIN268 [R/W] B,H,W 00000000 00000000		
B474_821C	RIC_RESIN271 [R/W] B,H,W 00000000 00000000		RIC_RESIN270 [R/W] B,H,W 00000000 00000000		

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B474_8220	-		-		
B474_8224	-		-		
B474_8228	RIC_RESIN277 [R/W] B,H,W 00000000 00000000		RIC_RESIN276 [R/W] B,H,W 00000000 00000000		
B474_822C	RIC_RESIN279 [R/W] B,H,W 00000000 00000000		RIC_RESIN278 [R/W] B,H,W 00000000 00000000		
B474_8230	RIC_RESIN281 [R/W] B,H,W 00000000 00000000		RIC_RESIN280 [R/W] B,H,W 00000000 00000000		
B474_8234	RIC_RESIN283 [R/W] B,H,W 00000000 00000000		RIC_RESIN282 [R/W] B,H,W 00000000 00000000		
B474_8238	RIC_RESIN285 [R/W] B,H,W 00000000 00000000		RIC_RESIN284 [R/W] B,H,W 00000000 00000000		
B474_823C	RIC_RESIN287 [R/W] B,H,W 00000000 00000000		RIC_RESIN286 [R/W] B,H,W 00000000 00000000		
B474_8240	RIC_RESIN289 [R/W] B,H,W 00000000 00000000		RIC_RESIN288 [R/W] B,H,W 00000000 00000000		
B474_8244	RIC_RESIN291 [R/W] B,H,W 00000000 00000000		RIC_RESIN290 [R/W] B,H,W 00000000 00000000		
B474_8248	RIC_RESIN293 [R/W] B,H,W 00000000 00000000		RIC_RESIN292 [R/W] B,H,W 00000000 00000000		
B474_824C	RIC_RESIN295 [R/W] B,H,W 00000000 00000000		RIC_RESIN294 [R/W] B,H,W 00000000 00000000		
B474_8250	RIC_RESIN297 [R/W] B,H,W 00000000 00000000		RIC_RESIN296 [R/W] B,H,W 00000000 00000000		
B474_8254	RIC_RESIN299 [R/W] B,H,W 00000000 00000000		RIC_RESIN298 [R/W] B,H,W 00000000 00000000		
B474_8258	RIC_RESIN301 [R/W] B,H,W 00000000 00000000		RIC_RESIN300 [R/W] B,H,W 00000000 00000000		
B474_825C	RIC_RESIN303 [R/W] B,H,W 00000000 00000000		RIC_RESIN302 [R/W] B,H,W 00000000 00000000		
B474_8260	RIC_RESIN305 [R/W] B,H,W 00000000 00000000		RIC_RESIN304 [R/W] B,H,W 00000000 00000000		
B474_8264	RIC_RESIN307 [R/W] B,H,W 00000000 00000000		RIC_RESIN306 [R/W] B,H,W 00000000 00000000		
B474_8268	RIC_RESIN309 [R/W] B,H,W 00000000 00000000		RIC_RESIN308 [R/W] B,H,W 00000000 00000000		
B474_826C	-		-		
B474_8270	-		-		
B474_8274	RIC_RESIN315 [R/W] B,H,W 00000000 00000000		RIC_RESIN314 [R/W] B,H,W 00000000 00000000		
B474_8278	RIC_RESIN317 [R/W] B,H,W 00000000 00000000		RIC_RESIN316 [R/W] B,H,W 00000000 00000000		
B474_827C	RIC_RESIN319 [R/W] B,H,W 00000000 00000000		RIC_RESIN318 [R/W] B,H,W 00000000 00000000		

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B474_8280	RIC_RESIN321 [R/W] B,H,W 00000000 00000000		RIC_RESIN320 [R/W] B,H,W 00000000 00000000		
B474_8284	RIC_RESIN323 [R/W] B,H,W 00000000 00000000		RIC_RESIN322 [R/W] B,H,W 00000000 00000000		
B474_8288	RIC_RESIN325 [R/W] B,H,W 00000000 00000000		RIC_RESIN324 [R/W] B,H,W 00000000 00000000		
B474_828C	RIC_RESIN327 [R/W] B,H,W 00000000 00000000		RIC_RESIN326 [R/W] B,H,W 00000000 00000000		
B474_8290	RIC_RESIN329 [R/W] B,H,W 00000000 00000000		RIC_RESIN328 [R/W] B,H,W 00000000 00000000		
B474_8294	-		-		
B474_8298	RIC_RESIN333 [R/W] B,H,W 00000000 00000000		RIC_RESIN332 [R/W] B,H,W 00000000 00000000		
B474_829C	RIC_RESIN335 [R/W] B,H,W 00000000 00000000		RIC_RESIN334 [R/W] B,H,W 00000000 00000000		
B474_82A0	RIC_RESIN337 [R/W] B,H,W 00000000 00000000		RIC_RESIN336 [R/W] B,H,W 00000000 00000000		
B474_82A4	RIC_RESIN339 [R/W] B,H,W 00000000 00000000		RIC_RESIN338 [R/W] B,H,W 00000000 00000000		
B474_82A8	RIC_RESIN341 [R/W] B,H,W 00000000 00000000		RIC_RESIN340 [R/W] B,H,W 00000000 00000000		
B474_82AC	RIC_RESIN343 [R/W] B,H,W 00000000 00000000		RIC_RESIN342 [R/W] B,H,W 00000000 00000000		
B474_82B0	RIC_RESIN345 [R/W] B,H,W 00000000 00000000		RIC_RESIN344 [R/W] B,H,W 00000000 00000000		
B474_82B4	RIC_RESIN347 [R/W] B,H,W 00000000 00000000		RIC_RESIN346 [R/W] B,H,W 00000000 00000000		
B474_82B8	-				
B474_82BC	RIC_RESIN351 [R/W] B,H,W 00000000 00000000		RIC_RESIN350 [R/W] B,H,W 00000000 00000000		
B474_82C0	-				
B474_82C4	-				
B474_82C8	RIC_RESIN357 [R/W] B,H,W 00000000 00000000		RIC_RESIN356 [R/W] B,H,W 00000000 00000000		
B474_82CC	RIC_RESIN359 [R/W] B,H,W 00000000 00000000		RIC_RESIN358 [R/W] B,H,W 00000000 00000000		
B474_82D0	RIC_RESIN361 [R/W] B,H,W 00000000 00000000		RIC_RESIN360 [R/W] B,H,W 00000000 00000000		
B474_82D4	RIC_RESIN363 [R/W] B,H,W 00000000 00000000		RIC_RESIN362 [R/W] B,H,W 00000000 00000000		
B474_82D8	RIC_RESIN365 [R/W] B,H,W 00000000 00000000		RIC_RESIN364 [R/W] B,H,W 00000000 00000000		
B474_82DC	RIC_RESIN367 [R/W] B,H,W 00000000 00000000		RIC_RESIN366 [R/W] B,H,W 00000000 00000000		

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B474_82E0	-				
B474_82E4 to B474_93FC	-				
B474_9400	RIC_KEYCDR [R/W] W 00000000 00000000 00000000 00000000				
B474_9404 to B474_FFFC	-				
B475_0000	PPU0_CR [R/W] B,H,W 00000000 00000000 00000000 00000000				PPU
B475_0004	PPU0_SR [R/W] B,H,W 00000000 00000000 00000000 00000001				
B475_0008	PPU0_UNLOCK [R/W] W 00000000 00000000 00000000 00000000				
B475_000C to B475_007C	-				
B475_0080	PPU0_PPR0 [R/W] B,H,W 11111111 11111111 11111111 11111111				
B475_0084	PPU0_PPR1 [R/W] B,H,W 11111111 11111111 11111111 11111111				
B475_0088	PPU0_PPR2 [R/W] B,H,W 11111111 11111111 11111111 11111111				
B475_008C	PPU0_PPR3 [R/W] B,H,W 11111111 11111111 11111111 11111111				
B475_0090	PPU0_PPR4 [R/W] B,H,W 11111111 11111111 11111111 11111111				
B475_0094	PPU0_PPR5 [R/W] B,H,W 11111111 11111111 11111111 11111111				
B475_0098	PPU0_PPR6 [R/W] B,H,W 11111111 11111111 11111111 11111111				
B475_009C	PPU0_PPR7 [R/W] B,H,W 11111111 11111111 11111111 11111111				
B475_00A0	PPU0_PPR8 [R/W] B,H,W 11111111 11111111 11111111 11111111				
B475_00A4	PPU0_PPR9 [R/W] B,H,W 11111111 11111111 11111111 11111111				
B475_00A8	PPU0_PPR10 [R/W] B,H,W 11111111 11111111 11111111 11111111				
B475_00AC	PPU0_PPR11 [R/W] B,H,W 11111111 11111111 11111111 11111111				
B475_00B0 to B475_00F4	-				
B475_00F8	PPU0_LOCK0 [R/W] W 00000000 00000000 00000000 00000000				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B475_00FC	-				
B475_0100	PPU0_PUR0 [R/W] B,H,W 11111111 11111111 11111111 11111111				
B475_0104	PPU0_PUR1 [R/W] B,H,W 11111111 11111111 11111111 11111111				
B475_0108	PPU0_PUR2 [R/W] B,H,W 11111111 11111111 11111111 11111111				
B475_010C	PPU0_PUR3 [R/W] B,H,W 11111111 11111111 11111111 11111111				
B475_0110	PPU0_PUR4 [R/W] B,H,W 11111111 11111111 11111111 11111111				
B475_0114	PPU0_PUR5 [R/W] B,H,W 11111111 11111111 11111111 11111111				
B475_0118	PPU0_PUR6 [R/W] B,H,W 11111111 11111111 11111111 11111111				
B475_011C	PPU0_PUR7 [R/W] B,H,W 11111111 11111111 11111111 11111111				
B475_0120	PPU0_PUR8 [R/W] B,H,W 11111111 11111111 11111111 11111111				
B475_0124	PPU0_PUR9 [R/W] B,H,W 11111111 11111111 11111111 11111111				
B475_0128	PPU0_PUR10 [R/W] B,H,W 11111111 11111111 11111111 11111111				
B475_012C	PPU0_PUR11 [R/W] B,H,W 11111111 11111111 11111111 11111111				
B475_0130 to B475_0174	-				
B475_0178	PPU0_LOCK1 [R/W] W 00000000 00000000 00000000 00000000				
B475_017C	-				
B475_0180	PPU0_PPWA0 [R/W] B,H,W 11111111 11111111 11111111 11111111				
B475_0184	PPU0_PPWA1 [R/W] B,H,W 11111111 11111111 11111111 11111111				
B475_0188	PPU0_PPWA2 [R/W] B,H,W 11111111 11111111 11111111 11111111				
B475_018C	PPU0_PPWA3 [R/W] B,H,W 11111111 11111111 11111111 11111111				
B475_0190	PPU0_PPWA4 [R/W] B,H,W 11111111 11111111 11111111 11111111				
B475_0194	PPU0_PPWA5 [R/W] B,H,W 11111111 11111111 11111111 11111111				
B475_0198	PPU0_PPWA6 [R/W] B,H,W 11111111 11111111 11111111 11111111				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B475_019C	PPU0_PPWA7 [R/W] B,H,W 11111111 11111111 11111111 11111111				
B475_01A0	PPU0_PPWA8 [R/W] B,H,W 11111111 11111111 11111111 11111111				
B475_01A4	PPU0_PPWA9 [R/W] B,H,W 11111111 11111111 11111111 11111111				
B475_01A8	PPU0_PPWA10 [R/W] B,H,W 11111111 11111111 11111111 11111111				
B475_01AC	PPU0_PPWA11 [R/W] B,H,W 11111111 11111111 11111111 11111111				
B475_01B0 to B475_01F4	-				
B475_01F8	PPU0_LOCK2 [R/W] W 00000000 00000000 00000000 00000000				
B475_01FC	-				
B475_0200	PPU0_PUWA0 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B475_0204	PPU0_PUWA1 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B475_0208	PPU0_PUWA2 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B475_020C	PPU0_PUWA3 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B475_0210	PPU0_PUWA4 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B475_0214	PPU0_PUWA5 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B475_0218	PPU0_PUWA6 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B475_021C	PPU0_PUWA7 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B475_0220	PPU0_PUWA8 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B475_0224	PPU0_PUWA9 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B475_0228	PPU0_PUWA10 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B475_022C	PPU0_PUWA11 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B475_0230 to B475_0274	-				
B475_0278	PPU0_LOCK3 [R/W] W 00000000 00000000 00000000 00000000				
B475_027C to B475_7FFC	-				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B475_8000 To B477_FFFC	-				Reserved
B478_0000 to B478_7FFC	-				Reserved
B478_8000	RLT32_DMACFG [R/W] B,H,W 00000000 00000000 00000000 00000000				Reload Timer ch.32
B478_8004	-				
B478_8008	RLT32_TMCSR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B478_800C	-				
B478_8010	RLT32_TMRLR [R/W] W,DW XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
B478_8014	RLT32_TMR [R/W] W,DW XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
B478_8018 to B478_83FC	-				
B478_8400	RLT33_DMACFG [R/W] B,H,W 00000000 00000000 00000000 00000000				Reload Timer ch.33
B478_8404	-				
B478_8408	RLT33_TMCSR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B478_840C	-				
B478_8410	RLT33_TMRLR [R/W] W,DW XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
B478_8414	RLT33_TMR [R/W] W,DW XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
B478_8418 to B478_87FC	-				
B478_8800 to B478_BFFC	-				Reserved
B478_C000 to B478_FBFC	-				Reserved
B478_FC00	CPG2_RLT_TSEL [R/W] B,H,W 00000000 00000000 00000000 00000000				CPERI #2 Misc_Registers
B478_FC04	CPG2_RLT_SSSR [R/W] B,H,W 00000000 00000000 00000000 00000000				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B478_FC08 to B478_FFFC	-				
B479_0000 to B479_03FC	-				Bus Performance Counter
B479_0400 to B47F_FFFC	-				Reserved

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B480_0000	CPG_MFS00_SSR [R/W] B,H,W 00000011	CPG_MFS00_ESCR/CPG_MFS00_ESCR/CPG_MFS00_ESCR/CPG_MFS00_IBSR [R/W] B,H,W 00000000	CPG_MFS00_SCR/CPG_MFS00_SCR/CPG_MFS00_SCR/CPG_MFS00_IBCR [R/W] B,H,W 00000000	CPG_MFS00_SMR [R/W] B,H,W 00000000	CPG_MFS ch.0
B480_0004	-/CPG_MFS00_RDR(TDR)/-/- [R/W] B,H,W --------(*5)	-/CPG_MFS00_RDR(TDR)/-/- [R/W] B,H,W --------(*5)	CPG_MFS00_RDR(TDR)/CPG_MFS00_RDR(TDR)/-/- [R/W] B,H,W -------0	CPG_MFS00_RDR(TDR) [R/W] B,H,W 00000000	
B480_0008	CPG_MFS00_STMR [R/W] B,H,W 00000000 00000000		CPG_MFS00_SACSR [R/W] B,H,W 00000000 00000000		
B480_000C	-/CPG_MFS00_SCSCR/CPG_MFS00_SFUR/- [R/W] B,H,W --------(*5)(*6)	-/CPG_MFS00_SCSCR/CPG_MFS00_SFUR/- [R/W] B,H,W --------(*5)(*6)	CPG_MFS00_STMCR [R/W] B,H,W 00000000 00000000		
B480_0010	-/CPG_MFS00_SCSTR3/CPG_MFS00_LAMSR/- [R/W] B,H,W --------(*5)(*6)	-/CPG_MFS00_SCSTR2/CPG_MFS00_LAMCR/- [R/W] B,H,W --------(*5)(*6)	-/CPG_MFS00_SCSTR1/CPG_MFS00_SFLR1/CPG_MFS00_EIBCR [R/W] B,H,W --------(*5)(*6)(*7)	-/CPG_MFS00_SCSTR0/CPG_MFS00_SFLR0/CPG_MFS00_NFCR [R/W] B,H,W --------(*5)(*6)(*7)	
B480_0014	-	-/CPG_MFS00_SCSFR2/-/- [R/W] B,H,W --------(*5)	-/CPG_MFS00_SCSFR1/-/- [R/W] B,H,W --------(*5)	-/CPG_MFS00_SCSFR0/-/- [R/W] B,H,W --------(*5)	
B480_0018	-/CPG_MFS00_TBYTE3/CPG_MFS00_LAMESR/- [R/W] B,H,W --------(*5)(*6)	-/CPG_MFS00_TBYTE2/CPG_MFS00_LAMERT/- [R/W] B,H,W --------(*5)(*6)	-/CPG_MFS00_TBYTE1/CPG_MFS00_LAMIER/- [R/W] B,H,W --------(*5)(*6)	CPG_MFS00_TBYTE0/CPG_MFS00_TBYTE0/CPG_MFS00_LAMRID(LAMTID) /- [R/W] B,H,W 00000000	
B480_001C	-/-/-/CPG_MFS00_ISMK [R/W] B,H,W --------(*7)	-/-/-/CPG_MFS00_ISBA [R/W] B,H,W --------(*7)	CPG_MFS00_BGR1 [R/W] H,W 00000000	CPG_MFS00_BGR0 [R/W] H,W 00000000	
B480_0020	CPG_MFS00_FBYTE2 [R/W] B,H,W 00000000	CPG_MFS00_FBYTE1 [R/W] B,H,W 00000000	CPG_MFS00_FCR1 [R/W] B,H,W 00000100	CPG_MFS00_FCR0 [R/W] B,H,W 00000000	
B480_0024	CPG_MFS00_ESR [R/W] B,H,W 00000000	CPG_MFS00_ECR [R/W] B,H,W 00000000	CPG_MFS00_FTICR2 [R/W] B,H,W 00000000	CPG_MFS00_FTICR1 [R/W] B,H,W 00000000	
B480_0028	-			CPG_MFS00_TBSIZE [R/W] B,H,W 00000000	
B480_002C	-/-/CPG_MFS00_SSRC/CPG_MFS00_SSRC [R/W] B,H,W --------(*6)(*7)	-/-/CPG_MFS00_ESCRC/CPG_MFS00_IBSRC [R/W] B,H,W --------(*6)(*7)	-/-/CPG_MFS00_SCRC/CPG_MFS00_IBCRC [R/W] B,H,W --------(*6)(*7)	-/-/CPG_MFS00_SMRC/CPG_MFS00_SMRC [R/W] B,H,W --------(*6)(*7)	
B480_0030	-		CPG_MFS00_SACSRC [R/W] B,H,W 00000000 00000000		
B480_0034	-/-/CPG_MFS00_LAMSRC/- [R/W] B,H,W --------(*6)	-/-/CPG_MFS00_LAMCRC/- [R/W] B,H,W --------(*6)	-		
B480_0038	-/-/CPG_MFS00_LAMESRC/- [R/W] B,H,W --------(*6)	-	-/-/CPG_MFS00_LAMIERC/- [R/W] B,H,W --------(*6)	-	
B480_003C	-		CPG_MFS00_FCR1C [R/W] B,H,W 00000000	CPG_MFS00_FCR0C [R/W] B,H,W 00000000	
B480_0040	CPG_MFS00_ESRC [R/W] B,H,W 00000000	-			

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B480_0044	-/-/CPG_MFS00_SSRS/CPG_MFS00_SSRS [R/W] B,H,W --------(*6)(*7)	-/-/CPG_MFS00_ESCRS/- [R/W] B,H,W --------(*6)	-/-/CPG_MFS00_SCRS/CPG_MFS00_IBCRS [R/W] B,H,W --------(*6)(*7)	-/-/CPG_MFS00_SMRS/CPG_MFS00_SMRS [R/W] B,H,W --------(*6)(*7)	
B480_0048	-		CPG_MFS00_SACSRS [R/W] B,H,W 00000000 00000000		
B480_004C	-	-/-/CPG_MFS00_LAMCRS/- [R/W] B,H,W --------(*6)	-		
B480_0050	-		-/-/CPG_MFS00_LAMIERS/- [R/W] B,H,W --------(*6)	-	
B480_0054	-		CPG_MFS00_FCR1S [R/W] B,H,W 00000000	CPG_MFS00_FCR0S [R/W] B,H,W 00000000	
B480_0058	CPG_MFS00_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS00_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS00_RDR1(TDR1)[R/W] B,H,W	CPG_MFS00_RDR0(TDR0)[R/W] B,H,W	
B480_005C	CPG_MFS00_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS00_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS00_RDR1(TDR1)[R/W] B,H,W	CPG_MFS00_RDR0(TDR0)[R/W] B,H,W	
B480_0060	CPG_MFS00_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS00_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS00_RDR1(TDR1)[R/W] B,H,W	CPG_MFS00_RDR0(TDR0)[R/W] B,H,W	
B480_0064	CPG_MFS00_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS00_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS00_RDR1(TDR1)[R/W] B,H,W	CPG_MFS00_RDR0(TDR0)[R/W] B,H,W	
B480_0068	CPG_MFS00_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS00_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS00_RDR1(TDR1)[R/W] B,H,W	CPG_MFS00_RDR0(TDR0)[R/W] B,H,W	
B480_006C	CPG_MFS00_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS00_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS00_RDR1(TDR1)[R/W] B,H,W	CPG_MFS00_RDR0(TDR0)[R/W] B,H,W	
B480_0070	CPG_MFS00_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS00_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS00_RDR1(TDR1)[R/W] B,H,W	CPG_MFS00_RDR0(TDR0)[R/W] B,H,W	
B480_0074	CPG_MFS00_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS00_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS00_RDR1(TDR1)[R/W] B,H,W	CPG_MFS00_RDR0(TDR0)[R/W] B,H,W	
B480_0078	CPG_MFS00_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS00_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS00_RDR1(TDR1)[R/W] B,H,W	CPG_MFS00_RDR0(TDR0)[R/W] B,H,W	
B480_007C	CPG_MFS00_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS00_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS00_RDR1(TDR1)[R/W] B,H,W	CPG_MFS00_RDR0(TDR0)[R/W] B,H,W	
B480_0080	CPG_MFS00_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS00_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS00_RDR1(TDR1)[R/W] B,H,W	CPG_MFS00_RDR0(TDR0)[R/W] B,H,W	
B480_0084	CPG_MFS00_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS00_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS00_RDR1(TDR1)[R/W] B,H,W	CPG_MFS00_RDR0(TDR0)[R/W] B,H,W	
B480_0088	CPG_MFS00_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS00_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS00_RDR1(TDR1)[R/W] B,H,W	CPG_MFS00_RDR0(TDR0)[R/W] B,H,W	
B480_008C	CPG_MFS00_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS00_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS00_RDR1(TDR1)[R/W] B,H,W	CPG_MFS00_RDR0(TDR0)[R/W] B,H,W	
B480_0090	CPG_MFS00_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS00_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS00_RDR1(TDR1)[R/W] B,H,W	CPG_MFS00_RDR0(TDR0)[R/W] B,H,W	
B480_0094	CPG_MFS00_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS00_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS00_RDR1(TDR1)[R/W] B,H,W	CPG_MFS00_RDR0(TDR0)[R/W] B,H,W	
B480_0098 to B480_03FC	-				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B480_0400	CPG_MFS01_SSR [R/W] B,H,W 00000011	CPG_MFS01_ESCR [R/W] B,H,W 00000000	CPG_MFS01_SCR [R/W] B,H,W 00000000	CPG_MFS01_SMR [R/W] B,H,W 00000000	CPG_MFS ch.1
B480_0404	-/CPG_MFS01_RDR(TDR)/- [R/W] B,H,W --------(*5)	-/CPG_MFS01_RDR(TDR)/- [R/W] B,H,W --------(*5)	CPG_MFS01_RDR(TDR)/CPG_MFS01_RDR(TDR)/- [R/W] B,H,W -------0	CPG_MFS01_RDR(TDR) [R/W] B,H,W 00000000	
B480_0408	CPG_MFS01_STMR [R/W] B,H,W 00000000 00000000		CPG_MFS01_SACSR [R/W] B,H,W 00000000 00000000		
B480_040C	-/CPG_MFS01_SCSCR/CPG_MFS01_SFUR [R/W] B,H,W --------(*5)(*6)	-/CPG_MFS01_SCSCR/CPG_MFS01_SFUR [R/W] B,H,W --------(*5)(*6)	CPG_MFS01_STMCR [R/W] B,H,W 00000000 00000000		
B480_0410	-/CPG_MFS01_SCSTR3/CPG_MFS01_LAMSR [R/W] B,H,W --------(*5)(*6)	-/CPG_MFS01_SCSTR2/CPG_MFS01_LAMCR [R/W] B,H,W --------(*5)(*6)	-/CPG_MFS01_SCSTR1/CPG_MFS01_SFLR1 [R/W] B,H,W --------(*5)(*6)	-/CPG_MFS01_SCSTR0/CPG_MFS01_SFLR0 [R/W] B,H,W --------(*5)(*6)	
B480_0414	-	-/CPG_MFS01_SCSFR2/- [R/W] B,H,W --------(*5)	-/CPG_MFS01_SCSFR1/- [R/W] B,H,W --------(*5)	-/CPG_MFS01_SCSFR0/- [R/W] B,H,W --------(*5)	
B480_0418	-/CPG_MFS01_TBYTE3/CPG_MFS01_LAMESR [R/W] B,H,W --------(*5)(*6)	-/CPG_MFS01_TBYTE2/CPG_MFS01_LAMERT [R/W] B,H,W --------(*5)(*6)	-/CPG_MFS01_TBYTE1/CPG_MFS01_LAMIER [R/W] B,H,W --------(*5)(*6)	CPG_MFS01_TBYTE0/CPG_MFS01_TBYTE0/CPG_MFS01_LAMRID(LAMTID) [R/W] B,H,W 00000000	
B480_041C	-	-	CPG_MFS01_BGR1 [R/W] H,W 00000000	CPG_MFS01_BGR0 [R/W] H,W 00000000	
B480_0420	CPG_MFS01_FBYTE2 [R/W] B,H,W 00000000	CPG_MFS01_FBYTE1 [R/W] B,H,W 00000000	CPG_MFS01_FCR1 [R/W] B,H,W 00000100	CPG_MFS01_FCR0 [R/W] B,H,W 00000000	
B480_0424	CPG_MFS01_ESR [R/W] B,H,W 00000000	CPG_MFS01_ECR [R/W] B,H,W 00000000	CPG_MFS01_FTICR2 [R/W] B,H,W 00000000	CPG_MFS01_FTICR1 [R/W] B,H,W 00000000	
B480_0428	-			CPG_MFS01_TBSIZE [R/W] B,H,W 00000000	
B480_042C	-/-/CPG_MFS01_SSRC [R/W] B,H,W --------(*6)	-/-/CPG_MFS01_ESCRC [R/W] B,H,W --------(*6)	-/-/CPG_MFS01_SCRC [R/W] B,H,W --------(*6)	-/-/CPG_MFS01_SMRC [R/W] B,H,W --------(*6)	
B480_0430	-		CPG_MFS01_SACSRC [R/W] B,H,W 00000000 00000000		
B480_0434	-/-/CPG_MFS01_LAMSRC [R/W] B,H,W --------(*6)	-/-/CPG_MFS01_LAMCRC [R/W] B,H,W --------(*6)	-		
B480_0438	-/-/CPG_MFS01_LAMESRC [R/W] B,H,W --------(*6)	-	-/-/CPG_MFS01_LAMIERC [R/W] B,H,W --------(*6)	-	
B480_043C	-		CPG_MFS01_FCR1C [R/W] B,H,W 00000000	CPG_MFS01_FCR0C [R/W] B,H,W 00000000	
B480_0440	CPG_MFS01_ESRC [R/W] B,H,W 00000000	-			

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B480_0444	-/-/CPG_MFS01_SSRS [R/W] B,H,W --------(*6)	-/-/CPG_MFS01_ESCRS [R/W] B,H,W --------(*6)	-/-/CPG_MFS01_SCRS [R/W] B,H,W --------(*6)	-/-/CPG_MFS01_SMRS [R/W] B,H,W --------(*6)	
B480_0448	-		CPG_MFS01_SACSRS [R/W] B,H,W 00000000 00000000		
B480_044C	-	-/-/CPG_MFS01_LAMCRS [R/W] B,H,W --------(*6)	-		
B480_0450	-		-/-/CPG_MFS01_LAMIERS [R/W] B,H,W --------(*6)	-	
B480_0454	-		CPG_MFS01_FCR1S [R/W] B,H,W 00000000	CPG_MFS01_FCR0S [R/W] B,H,W 00000000	
B480_0458	CPG_MFS01_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS01_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS01_RDR1(TDR1)[R/W] B,H,W	CPG_MFS01_RDR0(TDR0)[R/W] B,H,W	
B480_045C	CPG_MFS01_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS01_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS01_RDR1(TDR1)[R/W] B,H,W	CPG_MFS01_RDR0(TDR0)[R/W] B,H,W	
B480_0460	CPG_MFS01_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS01_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS01_RDR1(TDR1)[R/W] B,H,W	CPG_MFS01_RDR0(TDR0)[R/W] B,H,W	
B480_0464	CPG_MFS01_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS01_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS01_RDR1(TDR1)[R/W] B,H,W	CPG_MFS01_RDR0(TDR0)[R/W] B,H,W	
B480_0468	CPG_MFS01_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS01_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS01_RDR1(TDR1)[R/W] B,H,W	CPG_MFS01_RDR0(TDR0)[R/W] B,H,W	
B480_046C	CPG_MFS01_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS01_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS01_RDR1(TDR1)[R/W] B,H,W	CPG_MFS01_RDR0(TDR0)[R/W] B,H,W	
B480_0470	CPG_MFS01_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS01_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS01_RDR1(TDR1)[R/W] B,H,W	CPG_MFS01_RDR0(TDR0)[R/W] B,H,W	
B480_0474	CPG_MFS01_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS01_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS01_RDR1(TDR1)[R/W] B,H,W	CPG_MFS01_RDR0(TDR0)[R/W] B,H,W	
B480_0478	CPG_MFS01_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS01_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS01_RDR1(TDR1)[R/W] B,H,W	CPG_MFS01_RDR0(TDR0)[R/W] B,H,W	
B480_047C	CPG_MFS01_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS01_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS01_RDR1(TDR1)[R/W] B,H,W	CPG_MFS01_RDR0(TDR0)[R/W] B,H,W	
B480_0480	CPG_MFS01_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS01_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS01_RDR1(TDR1)[R/W] B,H,W	CPG_MFS01_RDR0(TDR0)[R/W] B,H,W	
B480_0484	CPG_MFS01_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS01_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS01_RDR1(TDR1)[R/W] B,H,W	CPG_MFS01_RDR0(TDR0)[R/W] B,H,W	
B480_0488	CPG_MFS01_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS01_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS01_RDR1(TDR1)[R/W] B,H,W	CPG_MFS01_RDR0(TDR0)[R/W] B,H,W	
B480_048C	CPG_MFS01_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS01_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS01_RDR1(TDR1)[R/W] B,H,W	CPG_MFS01_RDR0(TDR0)[R/W] B,H,W	
B480_0490	CPG_MFS01_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS01_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS01_RDR1(TDR1)[R/W] B,H,W	CPG_MFS01_RDR0(TDR0)[R/W] B,H,W	
B480_0494	CPG_MFS01_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS01_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS01_RDR1(TDR1)[R/W] B,H,W	CPG_MFS01_RDR0(TDR0)[R/W] B,H,W	
B480_0498 to B480_07FC	-				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B480_0800	CPG_MFS02_SSR [R/W] B,H,W 00000011	CPG_MFS02_ESCR [R/W] B,H,W 00000000	CPG_MFS02_SCR [R/W] B,H,W 00000000	CPG_MFS02_SMR [R/W] B,H,W 00000000	CPG_MFS ch.2
B480_0804	-/CPG_MFS02_RDR(TDR)/- [R/W] B,H,W --------(*5)	-/CPG_MFS02_RDR(TDR)/- [R/W] B,H,W --------(*5)	CPG_MFS02_RDR(TDR)/CPG_MFS02_RDR(TDR)/- [R/W] B,H,W -------0	CPG_MFS02_RDR(TDR) [R/W] B,H,W 00000000	
B480_0808	CPG_MFS02_STMR [R/W] B,H,W 00000000 00000000		CPG_MFS02_SACSR [R/W] B,H,W 00000000 00000000		
B480_080C	-/CPG_MFS02_SCSCR/CPG_MFS02_SFUR [R/W] B,H,W --------(*5)(*6)	-/CPG_MFS02_SCSCR/CPG_MFS02_SFUR [R/W] B,H,W --------(*5)(*6)	CPG_MFS02_STMCR [R/W] B,H,W 00000000 00000000		
B480_0810	-/CPG_MFS02_SCSTR3/CPG_MFS02_LAMSR [R/W] B,H,W --------(*5)(*6)	-/CPG_MFS02_SCSTR2/CPG_MFS02_LAMCR [R/W] B,H,W --------(*5)(*6)	-/CPG_MFS02_SCSTR1/CPG_MFS02_SFLR1 [R/W] B,H,W --------(*5)(*6)	-/CPG_MFS02_SCSTR0/CPG_MFS02_SFLR0 [R/W] B,H,W --------(*5)(*6)	
B480_0814	-	-/CPG_MFS02_SCSFR2/- [R/W] B,H,W --------(*5)	-/CPG_MFS02_SCSFR1/- [R/W] B,H,W --------(*5)	-/CPG_MFS02_SCSFR0/- [R/W] B,H,W --------(*5)	
B480_0818	-/CPG_MFS02_TBYTE3/CPG_MFS02_LAMESR [R/W] B,H,W --------(*5)(*6)	-/CPG_MFS02_TBYTE2/CPG_MFS02_LAMERT [R/W] B,H,W --------(*5)(*6)	-/CPG_MFS02_TBYTE1/CPG_MFS02_LAMIER [R/W] B,H,W --------(*5)(*6)	CPG_MFS02_TBYTE0/CPG_MFS02_TBYTE0/CPG_MFS02_LAMRID(LAMTID) [R/W] B,H,W 00000000	
B480_081C	-	-	CPG_MFS02_BGR1 [R/W] H,W 00000000	CPG_MFS02_BGR0 [R/W] H,W 00000000	
B480_0820	CPG_MFS02_FBYTE2 [R/W] B,H,W 00000000	CPG_MFS02_FBYTE1 [R/W] B,H,W 00000000	CPG_MFS02_FCR1 [R/W] B,H,W 00000100	CPG_MFS02_FCR0 [R/W] B,H,W 00000000	
B480_0824	CPG_MFS02_ESR [R/W] B,H,W 00000000	CPG_MFS02_ECR [R/W] B,H,W 00000000	CPG_MFS02_FTICR2 [R/W] B,H,W 00000000	CPG_MFS02_FTICR1 [R/W] B,H,W 00000000	
B480_0828	-			CPG_MFS02_TBSIZE [R/W] B,H,W 00000000	
B480_082C	-/-/CPG_MFS02_SSRC [R/W] B,H,W --------(*6)	-/-/CPG_MFS02_ESCRC [R/W] B,H,W --------(*6)	-/-/CPG_MFS02_SCRC [R/W] B,H,W --------(*6)	-/-/CPG_MFS02_SMRC [R/W] B,H,W --------(*6)	
B480_0830	-		CPG_MFS02_SACSRC [R/W] B,H,W 00000000 00000000		
B480_0834	-/-/CPG_MFS02_LAMSRC [R/W] B,H,W --------(*6)	-/-/CPG_MFS02_LAMCRC [R/W] B,H,W --------(*6)	-		
B480_0838	-/-/CPG_MFS02_LAMESRC [R/W] B,H,W --------(*6)	-	-/-/CPG_MFS02_LAMIERC [R/W] B,H,W --------(*6)	-	
B480_083C	-		CPG_MFS02_FCR1C [R/W] B,H,W 00000000	CPG_MFS02_FCR0C [R/W] B,H,W 00000000	
B480_0840	CPG_MFS02_ESRC [R/W] B,H,W 00000000	-			
B480_0844	-/-/CPG_MFS02_SSRS [R/W] B,H,W --------(*6)	-/-/CPG_MFS02_ESCRS [R/W] B,H,W --------(*6)	-/-/CPG_MFS02_SCRS [R/W] B,H,W --------(*6)	-/-/CPG_MFS02_SMRS [R/W] B,H,W --------(*6)	

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B480_0848	-		CPG_MFS02_SACSRS [R/W] B,H,W 00000000 00000000		
B480_084C	-	-/-/CPG_MFS02_LAMCRS [R/W] B,H,W --------(*6)	-		
B480_0850	-		-/-/CPG_MFS02_LAMIERS [R/W] B,H,W --------(*6)	-	
B480_0854	-		CPG_MFS02_FCR1S [R/W] B,H,W 00000000	CPG_MFS02_FCR0S [R/W] B,H,W 00000000	
B480_0858	CPG_MFS02_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS02_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS02_RDR1(TDR1)[R/W] B,H,W	CPG_MFS02_RDR0(TDR0)[R/W] B,H,W	
B480_085C	CPG_MFS02_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS02_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS02_RDR1(TDR1)[R/W] B,H,W	CPG_MFS02_RDR0(TDR0)[R/W] B,H,W	
B480_0860	CPG_MFS02_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS02_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS02_RDR1(TDR1)[R/W] B,H,W	CPG_MFS02_RDR0(TDR0)[R/W] B,H,W	
B480_0864	CPG_MFS02_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS02_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS02_RDR1(TDR1)[R/W] B,H,W	CPG_MFS02_RDR0(TDR0)[R/W] B,H,W	
B480_0868	CPG_MFS02_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS02_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS02_RDR1(TDR1)[R/W] B,H,W	CPG_MFS02_RDR0(TDR0)[R/W] B,H,W	
B480_086C	CPG_MFS02_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS02_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS02_RDR1(TDR1)[R/W] B,H,W	CPG_MFS02_RDR0(TDR0)[R/W] B,H,W	
B480_0870	CPG_MFS02_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS02_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS02_RDR1(TDR1)[R/W] B,H,W	CPG_MFS02_RDR0(TDR0)[R/W] B,H,W	
B480_0874	CPG_MFS02_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS02_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS02_RDR1(TDR1)[R/W] B,H,W	CPG_MFS02_RDR0(TDR0)[R/W] B,H,W	
B480_0878	CPG_MFS02_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS02_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS02_RDR1(TDR1)[R/W] B,H,W	CPG_MFS02_RDR0(TDR0)[R/W] B,H,W	
B480_087C	CPG_MFS02_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS02_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS02_RDR1(TDR1)[R/W] B,H,W	CPG_MFS02_RDR0(TDR0)[R/W] B,H,W	
B480_0880	CPG_MFS02_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS02_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS02_RDR1(TDR1)[R/W] B,H,W	CPG_MFS02_RDR0(TDR0)[R/W] B,H,W	
B480_0884	CPG_MFS02_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS02_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS02_RDR1(TDR1)[R/W] B,H,W	CPG_MFS02_RDR0(TDR0)[R/W] B,H,W	
B480_0888	CPG_MFS02_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS02_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS02_RDR1(TDR1)[R/W] B,H,W	CPG_MFS02_RDR0(TDR0)[R/W] B,H,W	
B480_088C	CPG_MFS02_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS02_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS02_RDR1(TDR1)[R/W] B,H,W	CPG_MFS02_RDR0(TDR0)[R/W] B,H,W	
B480_0890	CPG_MFS02_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS02_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS02_RDR1(TDR1)[R/W] B,H,W	CPG_MFS02_RDR0(TDR0)[R/W] B,H,W	
B480_0894	CPG_MFS02_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS02_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS02_RDR1(TDR1)[R/W] B,H,W	CPG_MFS02_RDR0(TDR0)[R/W] B,H,W	
B480_0898 to B480_0BFC	-				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B480_0C00	CPG_MFS03_SSR [R/W] B,H,W 00000011	CPG_MFS03_ESCR/CPG_MFS03_ESCR/CPG_MFS03_ESCR/CPG_MFS03_IBSR [R/W] B,H,W 00000000	CPG_MFS03_SCR/CPG_MFS03_SCR/CPG_MFS03_SCR/CPG_MFS03_IBCR [R/W] B,H,W 00000000	CPG_MFS03_SMR [R/W] B,H,W 00000000	CPG_MFS ch.3
B480_0C04	-/CPG_MFS03_RDR(TDR)/- /- [R/W] B,H,W --------(*5)	-/CPG_MFS03_RDR(TDR)/- /- [R/W] B,H,W --------(*5)	CPG_MFS03_RDR(TDR)/CPG_MFS03_RDR(TDR)/- /- [R/W] B,H,W -------0	CPG_MFS03_RDR(TDR) [R/W] B,H,W 00000000	
B480_0C08	CPG_MFS03_STMR [R/W] B,H,W 00000000 00000000		CPG_MFS03_SACSR [R/W] B,H,W 00000000 00000000		
B480_0C0C	-/CPG_MFS03_SCSCR/CPG_MFS03_SFUR/- [R/W] B,H,W --------(*5)(*6)	-/CPG_MFS03_SCSCR/CPG_MFS03_SFUR/- [R/W] B,H,W --------(*5)(*6)	CPG_MFS03_STMCR [R/W] B,H,W 00000000 00000000		
B480_0C10	-/CPG_MFS03_SCSTR3/CPG_MFS03_LAMSR/- [R/W] B,H,W --------(*5)(*6)	-/CPG_MFS03_SCSTR2/CPG_MFS03_LAMCR/- [R/W] B,H,W --------(*5)(*6)	-/CPG_MFS03_SCSTR1/CPG_MFS03_SFLR1/CPG_MFS03_EIBCR [R/W] B,H,W --------(*5)(*6)(*7)	-/CPG_MFS03_SCSTR0/CPG_MFS03_SFLR0/CPG_MFS03_NFCR [R/W] B,H,W --------(*5)(*6)(*7)	
B480_0C14	-	-/CPG_MFS03_SCSFR2/-/- [R/W] B,H,W --------(*5)	-/CPG_MFS03_SCSFR1/-/- [R/W] B,H,W --------(*5)	-/CPG_MFS03_SCSFR0/-/- [R/W] B,H,W --------(*5)	
B480_0C18	-/CPG_MFS03_TBYTE3/CPG_MFS03_LAMESR/- [R/W] B,H,W --------(*5)(*6)	-/CPG_MFS03_TBYTE2/CPG_MFS03_LAMERT/- [R/W] B,H,W --------(*5)(*6)	-/CPG_MFS03_TBYTE1/CPG_MFS03_LAMIER/- [R/W] B,H,W --------(*5)(*6)	CPG_MFS03_TBYTE0/CPG_MFS03_TBYTE0/CPG_MFS03_LAMRID(LAMTID) /- [R/W] B,H,W 00000000	
B480_0C1C	-/-/-/CPG_MFS03_ISMK [R/W] B,H,W --------(*7)	-/-/-/CPG_MFS03_ISBA [R/W] B,H,W --------(*7)	CPG_MFS03_BGR1 [R/W] H,W 00000000	CPG_MFS03_BGR0 [R/W] H,W 00000000	
B480_0C20	CPG_MFS03_FBYTE2 [R/W] B,H,W 00000000	CPG_MFS03_FBYTE1 [R/W] B,H,W 00000000	CPG_MFS03_FCR1 [R/W] B,H,W 00000100	CPG_MFS03_FCR0 [R/W] B,H,W 00000000	
B480_0C24	CPG_MFS03_ESR [R/W] B,H,W 00000000	CPG_MFS03_ECR [R/W] B,H,W 00000000	CPG_MFS03_FTICR2 [R/W] B,H,W 00000000	CPG_MFS03_FTICR1 [R/W] B,H,W 00000000	
B480_0C28	-			CPG_MFS03_TBSIZE [R/W] B,H,W 00000000	
B480_0C2C	-/-/CPG_MFS03_SSRC/CPG_MFS03_SSRC [R/W] B,H,W --------(*6)(*7)	-/-/CPG_MFS03_ESCRC/CPG_MFS03_IBSRC [R/W] B,H,W --------(*6)(*7)	-/-/CPG_MFS03_SCRC/CPG_MFS03_IBCRC [R/W] B,H,W --------(*6)(*7)	-/-/CPG_MFS03_SMRC/CPG_MFS03_SMRC [R/W] B,H,W --------(*6)(*7)	
B480_0C30	-		CPG_MFS03_SACSRC [R/W] B,H,W 00000000 00000000		
B480_0C34	-/-/CPG_MFS03_LAMSRC/- [R/W] B,H,W --------(*6)	-/-/CPG_MFS03_LAMCRC/- [R/W] B,H,W --------(*6)	-		
B480_0C38	-/-/CPG_MFS03_LAMESRC/- [R/W] B,H,W --------(*6)	-	-/-/CPG_MFS03_LAMIERC/- [R/W] B,H,W --------(*6)	-	
B480_0C3C	-		CPG_MFS03_FCR1C [R/W] B,H,W 00000000	CPG_MFS03_FCR0C [R/W] B,H,W 00000000	
B480_0C40	CPG_MFS03_ESRC [R/W] B,H,W 00000000	-			
B480_0C44	-/-/CPG_MFS03_SSRS/CPG_MFS03_SSRS [R/W] B,H,W --------(*6)(*7)	-/-/CPG_MFS03_ESCRS/- [R/W] B,H,W --------(*6)	-/-/CPG_MFS03_SCRS/CPG_MFS03_IBCRS [R/W] B,H,W --------(*6)(*7)	-/-/CPG_MFS03_SMRS/CPG_MFS03_SMRS [R/W] B,H,W --------(*6)(*7)	

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B480_0C48	-		CPG_MFS03_SACSRS [R/W] B,H,W 00000000 00000000		
B480_0C4C	-	-/-/CPG_MFS03_LAMCRS/- [R/W] B,H,W --------(*6)	-		
B480_0C50	-		-/-/CPG_MFS03_LAMIERS/- [R/W] B,H,W --------(*6)	-	
B480_0C54	-		CPG_MFS03_FCR1S [R/W] B,H,W 00000000	CPG_MFS03_FCR0S [R/W] B,H,W 00000000	
B480_0C58	CPG_MFS03_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS03_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS03_RDR1(TDR1)[R/W] B,H,W	CPG_MFS03_RDR0(TDR0)[R/W] B,H,W	
B480_0C5C	CPG_MFS03_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS03_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS03_RDR1(TDR1)[R/W] B,H,W	CPG_MFS03_RDR0(TDR0)[R/W] B,H,W	
B480_0C60	CPG_MFS03_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS03_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS03_RDR1(TDR1)[R/W] B,H,W	CPG_MFS03_RDR0(TDR0)[R/W] B,H,W	
B480_0C64	CPG_MFS03_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS03_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS03_RDR1(TDR1)[R/W] B,H,W	CPG_MFS03_RDR0(TDR0)[R/W] B,H,W	
B480_0C68	CPG_MFS03_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS03_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS03_RDR1(TDR1)[R/W] B,H,W	CPG_MFS03_RDR0(TDR0)[R/W] B,H,W	
B480_0C6C	CPG_MFS03_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS03_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS03_RDR1(TDR1)[R/W] B,H,W	CPG_MFS03_RDR0(TDR0)[R/W] B,H,W	
B480_0C70	CPG_MFS03_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS03_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS03_RDR1(TDR1)[R/W] B,H,W	CPG_MFS03_RDR0(TDR0)[R/W] B,H,W	
B480_0C74	CPG_MFS03_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS03_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS03_RDR1(TDR1)[R/W] B,H,W	CPG_MFS03_RDR0(TDR0)[R/W] B,H,W	
B480_0C78	CPG_MFS03_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS03_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS03_RDR1(TDR1)[R/W] B,H,W	CPG_MFS03_RDR0(TDR0)[R/W] B,H,W	
B480_0C7C	CPG_MFS03_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS03_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS03_RDR1(TDR1)[R/W] B,H,W	CPG_MFS03_RDR0(TDR0)[R/W] B,H,W	
B480_0C80	CPG_MFS03_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS03_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS03_RDR1(TDR1)[R/W] B,H,W	CPG_MFS03_RDR0(TDR0)[R/W] B,H,W	
B480_0C84	CPG_MFS03_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS03_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS03_RDR1(TDR1)[R/W] B,H,W	CPG_MFS03_RDR0(TDR0)[R/W] B,H,W	
B480_0C88	CPG_MFS03_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS03_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS03_RDR1(TDR1)[R/W] B,H,W	CPG_MFS03_RDR0(TDR0)[R/W] B,H,W	
B480_0C8C	CPG_MFS03_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS03_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS03_RDR1(TDR1)[R/W] B,H,W	CPG_MFS03_RDR0(TDR0)[R/W] B,H,W	
B480_0C90	CPG_MFS03_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS03_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS03_RDR1(TDR1)[R/W] B,H,W	CPG_MFS03_RDR0(TDR0)[R/W] B,H,W	
B480_0C94	CPG_MFS03_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS03_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS03_RDR1(TDR1)[R/W] B,H,W	CPG_MFS03_RDR0(TDR0)[R/W] B,H,W	
B480_0C98 to B480_0FFC	-				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B480_1000	CPG_MFS04_SSR [R/W] B,H,W 00000011	CPG_MFS04_ESCR/CPG_MFS04_ESCR/CPG_MFS04_ESCR/CPG_MFS04_IBSR [R/W] B,H,W 00000000	CPG_MFS04_SCR/CPG_MFS04_SCR/CPG_MFS04_SCR/CPG_MFS04_IBCR [R/W] B,H,W 00000000	CPG_MFS04_SMR [R/W] B,H,W 00000000	CPG_MFS ch.4
B480_1004	-/CPG_MFS04_RDR(TDR)/- /- [R/W] B,H,W --------(*5)	-/CPG_MFS04_RDR(TDR)/- /- [R/W] B,H,W --------(*5)	CPG_MFS04_RDR(TDR)/CPG_MFS04_RDR(TDR)/- /- [R/W] B,H,W -------0	CPG_MFS04_RDR(TDR) [R/W] B,H,W 00000000	
B480_1008	CPG_MFS04_STMR [R/W] B,H,W 00000000 00000000		CPG_MFS04_SACSR [R/W] B,H,W 00000000 00000000		
B480_100C	-/CPG_MFS04_SCSCR/CPG_MFS04_SFUR/- [R/W] B,H,W --------(*5)(*6)	-/CPG_MFS04_SCSCR/CPG_MFS04_SFUR/- [R/W] B,H,W --------(*5)(*6)	CPG_MFS04_STMCR [R/W] B,H,W 00000000 00000000		
B480_1010	-/CPG_MFS04_SCSTR3/CPG_MFS04_LAMSR/- [R/W] B,H,W --------(*5)(*6)	-/CPG_MFS04_SCSTR2/CPG_MFS04_LAMCR/- [R/W] B,H,W --------(*5)(*6)	-/CPG_MFS04_SCSTR1/CPG_MFS04_SFLR1/CPG_MFS04_EIBCR [R/W] B,H,W --------(*5)(*6)(*7)	-/CPG_MFS04_SCSTR0/CPG_MFS04_SFLR0/CPG_MFS04_NFCR [R/W] B,H,W --------(*5)(*6)(*7)	
B480_1014	-	-/CPG_MFS04_SCSFR2/-/- [R/W] B,H,W --------(*5)	-/CPG_MFS04_SCSFR1/-/- [R/W] B,H,W --------(*5)	-/CPG_MFS04_SCSFR0/-/- [R/W] B,H,W --------(*5)	
B480_1018	-/CPG_MFS04_TBYTE3/CPG_MFS04_LAMESR/- [R/W] B,H,W --------(*5)(*6)	-/CPG_MFS04_TBYTE2/CPG_MFS04_LAMERT/- [R/W] B,H,W --------(*5)(*6)	-/CPG_MFS04_TBYTE1/CPG_MFS04_LAMIER/- [R/W] B,H,W --------(*5)(*6)	CPG_MFS04_TBYTE0/CPG_MFS04_TBYTE0/CPG_MFS04_LAMRID(LAMTID) /- [R/W] B,H,W 00000000	
B480_101C	-/-/-/CPG_MFS04_ISMK [R/W] B,H,W --------(*7)	-/-/-/CPG_MFS04_ISBA [R/W] B,H,W --------(*7)	CPG_MFS04_BGR1 [R/W] H,W 00000000	CPG_MFS04_BGR0 [R/W] H,W 00000000	
B480_1020	CPG_MFS04_FBYTE2 [R/W] B,H,W 00000000	CPG_MFS04_FBYTE1 [R/W] B,H,W 00000000	CPG_MFS04_FCR1 [R/W] B,H,W 00000100	CPG_MFS04_FCR0 [R/W] B,H,W 00000000	
B480_1024	CPG_MFS04_ESR [R/W] B,H,W 00000000	CPG_MFS04_ECR [R/W] B,H,W 00000000	CPG_MFS04_FTICR2 [R/W] B,H,W 00000000	CPG_MFS04_FTICR1 [R/W] B,H,W 00000000	
B480_1028	-			CPG_MFS04_TBSIZE [R/W] B,H,W 00000000	
B480_102C	-/-/CPG_MFS04_SSRC/CPG_MFS04_SSRC [R/W] B,H,W --------(*6)(*7)	-/-/CPG_MFS04_ESCRC/CPG_MFS04_IBSRC [R/W] B,H,W --------(*6)(*7)	-/-/CPG_MFS04_SCRC/CPG_MFS04_IBCRC [R/W] B,H,W --------(*6)(*7)	-/-/CPG_MFS04_SMRC/CPG_MFS04_SMRC [R/W] B,H,W --------(*6)(*7)	
B480_1030	-		CPG_MFS04_SACSRC [R/W] B,H,W 00000000 00000000		
B480_1034	-/-/CPG_MFS04_LAMSRC/- [R/W] B,H,W --------(*6)	-/-/CPG_MFS04_LAMCRC/- [R/W] B,H,W --------(*6)	-		
B480_1038	-/-/CPG_MFS04_LAMESRC/- [R/W] B,H,W --------(*6)	-	-/-/CPG_MFS04_LAMIERC/- [R/W] B,H,W --------(*6)	-	
B480_103C	-		CPG_MFS04_FCR1C [R/W] B,H,W 00000000	CPG_MFS04_FCR0C [R/W] B,H,W 00000000	
B480_1040	CPG_MFS04_ESRC [R/W] B,H,W 00000000	-			
B480_1044	-/-/CPG_MFS04_SSRS/CPG_MFS04_SSRS [R/W] B,H,W --------(*6)(*7)	-/-/CPG_MFS04_ESCRS/- [R/W] B,H,W --------(*6)	-/-/CPG_MFS04_SCRS/CPG_MFS04_IBCRS [R/W] B,H,W --------(*6)(*7)	-/-/CPG_MFS04_SMRS/CPG_MFS04_SMRS [R/W] B,H,W --------(*6)(*7)	

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B480_1048	-		CPG_MFS04_SACSRS [R/W] B,H,W 00000000 00000000		
B480_104C	-	-/-/CPG_MFS04_LAMCRS/- [R/W] B,H,W --------(*6)	-		
B480_1050	-		-/-/CPG_MFS04_LAMIERS/- [R/W] B,H,W --------(*6)	-	
B480_1054	-		CPG_MFS04_FCR1S [R/W] B,H,W 00000000	CPG_MFS04_FCR0S [R/W] B,H,W 00000000	
B480_1058	CPG_MFS04_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS04_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS04_RDR1(TDR1)[R/W] B,H,W	CPG_MFS04_RDR0(TDR0)[R/W] B,H,W	
B480_105C	CPG_MFS04_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS04_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS04_RDR1(TDR1)[R/W] B,H,W	CPG_MFS04_RDR0(TDR0)[R/W] B,H,W	
B480_1060	CPG_MFS04_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS04_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS04_RDR1(TDR1)[R/W] B,H,W	CPG_MFS04_RDR0(TDR0)[R/W] B,H,W	
B480_1064	CPG_MFS04_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS04_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS04_RDR1(TDR1)[R/W] B,H,W	CPG_MFS04_RDR0(TDR0)[R/W] B,H,W	
B480_1068	CPG_MFS04_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS04_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS04_RDR1(TDR1)[R/W] B,H,W	CPG_MFS04_RDR0(TDR0)[R/W] B,H,W	
B480_106C	CPG_MFS04_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS04_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS04_RDR1(TDR1)[R/W] B,H,W	CPG_MFS04_RDR0(TDR0)[R/W] B,H,W	
B480_1070	CPG_MFS04_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS04_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS04_RDR1(TDR1)[R/W] B,H,W	CPG_MFS04_RDR0(TDR0)[R/W] B,H,W	
B480_1074	CPG_MFS04_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS04_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS04_RDR1(TDR1)[R/W] B,H,W	CPG_MFS04_RDR0(TDR0)[R/W] B,H,W	
B480_1078	CPG_MFS04_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS04_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS04_RDR1(TDR1)[R/W] B,H,W	CPG_MFS04_RDR0(TDR0)[R/W] B,H,W	
B480_107C	CPG_MFS04_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS04_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS04_RDR1(TDR1)[R/W] B,H,W	CPG_MFS04_RDR0(TDR0)[R/W] B,H,W	
B480_1080	CPG_MFS04_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS04_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS04_RDR1(TDR1)[R/W] B,H,W	CPG_MFS04_RDR0(TDR0)[R/W] B,H,W	
B480_1084	CPG_MFS04_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS04_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS04_RDR1(TDR1)[R/W] B,H,W	CPG_MFS04_RDR0(TDR0)[R/W] B,H,W	
B480_1088	CPG_MFS04_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS04_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS04_RDR1(TDR1)[R/W] B,H,W	CPG_MFS04_RDR0(TDR0)[R/W] B,H,W	
B480_108C	CPG_MFS04_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS04_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS04_RDR1(TDR1)[R/W] B,H,W	CPG_MFS04_RDR0(TDR0)[R/W] B,H,W	
B480_1090	CPG_MFS04_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS04_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS04_RDR1(TDR1)[R/W] B,H,W	CPG_MFS04_RDR0(TDR0)[R/W] B,H,W	
B480_1094	CPG_MFS04_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS04_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS04_RDR1(TDR1)[R/W] B,H,W	CPG_MFS04_RDR0(TDR0)[R/W] B,H,W	
B480_1098 to B480_13FC	-				
B480_1400 to B480_1FFC	-				Reserved
B480_2000 to B480_7FFC	-				Reserved

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B480_8000			BT00_PCSR/BT00_PCSR/BT00_PRLL/- [R/W] H,W 00000000 00000000		Base Timer ch.0
B480_8004	-		-/BT00_PDUT/ BT00_PRLH/ BT00_DTBF [R/W] H,W 00000000 00000000		
B480_8008	BT00_TMR/ BT00_TMR/ BT00_TMR/- [R/W] H,W 00000000 00000000 00000000 00000000 / (*9) 00000000 00000000 / (*10) XXXXXXXX XXXXXXXX(*4)				
B480_800C	-		BT00_TMCR [R/W] B,H,W 00000000 00000000		
B480_8010	-		BT00_TMCR2 [R/W] B,H,W 00000000	BT00_STC [R/W] B,H,W 00000000	
B480_8014	-		-	BT00_STCC [R/W] B,H,W 00000000	
B480_8018	-		-	BT00_STCS [R/W] B,H,W 00000000	
B480_801C	-		-/BT00_PSDR/-/- [R/W] H,W 00000000 00000000		
B480_8020	-		-/BT00_ADTR/-/- [R/W] H,W 00000000 00000000		
B480_8024	-		BT00_DEBUG [R/W] B,H,W 00000000 00000000		
B480_8028	-		-		
B480_802C	-		BT00_ETCDR [R/W] B,H,W 01111111 11111111		
B480_8030	-			BT_BTSEL01 [R/W] B,H,W 11110000	
B480_8034	-		BT_BTSSSR0 [R/W] B,H,W 11111111 11111111		
B480_8038 to B480_83FC	-				
B480_8400	-		BT01_PCSR/BT01_PCSR/BT01_PRLL/- [R/W] H,W 00000000 00000000		Base Timer ch.1
B480_8404	-		-/BT01_PDUT/ BT01_PRLH/ BT01_DTBF [R/W] H,W 00000000 00000000		
B480_8408	BT01_TMR/ BT01_TMR/ BT01_TMR/- [R/W] H,W 00000000 00000000 00000000 00000000 / (*9) 00000000 00000000 / (*10) XXXXXXXX XXXXXXXX(*4)				
B480_840C	-		BT01_TMCR [R/W] B,H,W 00000000 00000000		
B480_8410	-		BT01_TMCR2 [R/W] B,H,W 00000000	BT01_STC [R/W] B,H,W 00000000	
B480_8414	-		-	BT01_STCC [R/W] B,H,W 00000000	

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B480_8418	-		-	BT01_STCS [R/W] B,H,W 00000000	
B480_841C	-		-/BT01_PSDR/-/- [R/W] H,W 00000000 00000000		
B480_8420	-		-/BT01_ADTR/-/- [R/W] H,W 00000000 00000000		
B480_8424	-		BT01_DEBUG [R/W] B,H,W 00000000 00000000		
B480_8428	-		-		
B480_842C	-		BT01_ETCDR [R/W] B,H,W 01111111 11111111		
B480_8430 to B480_87FC	-				
B480_8800	-		BT02_PCSR/BT02_PCSR/BT02_PRLL/- [R/W] H,W 00000000 00000000		Base Timer ch.2
B480_8804	-		-/BT02_PDUT/ BT02_PRLH/ BT02_DTBF [R/W] H,W 00000000 00000000		
B480_8808	BT02_TMR/ BT02_TMR/ BT02_TMR/- [R/W] H,W 00000000 00000000 00000000 00000000 / (*9) 00000000 00000000 / (*10) XXXXXXXX XXXXXXXX(*4)				
B480_880C	-		BT02_TMCR [R/W] B,H,W 00000000 00000000		
B480_8810	-		BT02_TMCR2 [R/W] B,H,W 00000000	BT02_STC [R/W] B,H,W 00000000	
B480_8814	-		-	BT02_STCC [R/W] B,H,W 00000000	
B480_8818	-		-	BT02_STCS [R/W] B,H,W 00000000	
B480_881C	-		-/BT02_PSDR/-/- [R/W] H,W 00000000 00000000		
B480_8820	-		-/BT02_ADTR/-/- [R/W] H,W 00000000 00000000		
B480_8824	-		BT02_DEBUG [R/W] B,H,W 00000000 00000000		
B480_8828	-		-		
B480_882C	-		BT02_ETCDR [R/W] B,H,W 01111111 11111111		
B480_8830	-			BT_BTSEL23 [R/W] B,H,W 11110000	
B480_8834 to B480_8BFC	-				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B480_8C00	-		BT03_PCSR/BT03_PCSR/BT03_PRLL/- [R/W] H,W 00000000 00000000		Base Timer ch.3
B480_8C04	-		-/BT03_PDUT/ BT03_PRLH/ BT03_DTBF [R/W] H,W 00000000 00000000		
B480_8C08	BT03_TMR/ BT03_TMR/ BT03_TMR/- [R/W] H,W 00000000 00000000 00000000 00000000 / (*9) 00000000 00000000 / (*10) XXXXXXXX XXXXXXXX(*4)				
B480_8C0C	-		BT03_TMCR [R/W] B,H,W 00000000 00000000		
B480_8C10	-		BT03_TMCR2 [R/W] B,H,W 00000000	BT03_STC [R/W] B,H,W 00000000	
B480_8C14	-		-	BT03_STCC [R/W] B,H,W 00000000	
B480_8C18	-		-	BT03_STCS [R/W] B,H,W 00000000	
B480_8C1C	-		-/BT03_PSDR/-/- [R/W] H,W 00000000 00000000		
B480_8C20	-		-/BT03_ADTR/-/- [R/W] H,W 00000000 00000000		
B480_8C24	-		BT03_DEBUG [R/W] B,H,W 00000000 00000000		
B480_8C28	-		-		
B480_8C2C	-		BT03_ETCDR [R/W] B,H,W 01111111 11111111		
B480_8C30 to B480_8FFC	-				
B480_9000	-		BT04_PCSR/BT04_PCSR/BT04_PRLL/- [R/W] H,W 00000000 00000000		Base Timer ch.4
B480_9004	-		-/BT04_PDUT/ BT04_PRLH/ BT04_DTBF [R/W] H,W 00000000 00000000		
B480_9008	BT04_TMR/ BT04_TMR/ BT04_TMR/- [R/W] H,W 00000000 00000000 00000000 00000000 / (*9) 00000000 00000000 / (*10) XXXXXXXX XXXXXXXX(*4)				
B480_900C	-		BT04_TMCR [R/W] B,H,W 00000000 00000000		
B480_9010	-		BT04_TMCR2 [R/W] B,H,W 00000000	BT04_STC [R/W] B,H,W 00000000	
B480_9014	-		-	BT04_STCC [R/W] B,H,W 00000000	
B480_9018	-		-	BT04_STCS [R/W] B,H,W 00000000	
B480_901C	-		-/BT04_PSDR/-/- [R/W] H,W 00000000 00000000		

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B480_9020	-		-/BT04_ADTR/-/- [R/W] H,W 00000000 00000000		
B480_9024	-		BT04_DEBUG [R/W] B,H,W 00000000 00000000		
B480_9028	-		-		
B480_902C	-		BT04_ETCDR [R/W] B,H,W 01111111 11111111		
B480_9030	-			BT_BTSEL45 [R/W] B,H,W 11110000	
B480_9034 to B480_93FC	-				
B480_9400	-		BT05_PCSR/BT05_PCSR/BT05_PRLL/- [R/W] H,W 00000000 00000000		Base Timer ch.5
B480_9404	-		-/BT05_PDUT/ BT05_PRLH/ BT05_DTBF [R/W] H,W 00000000 00000000		
B480_9408	BT05_TMR/ BT05_TMR/ BT05_TMR/- [R/W] H,W 00000000 00000000 00000000 00000000 / (*9) 00000000 00000000 / (*10) XXXXXXXX XXXXXXXX(*4)				
B480_940C	-		BT05_TMCR [R/W] B,H,W 00000000 00000000		
B480_9410	-		BT05_TMCR2 [R/W] B,H,W 00000000	BT05_STC [R/W] B,H,W 00000000	
B480_9414	-		-	BT05_STCC [R/W] B,H,W 00000000	
B480_9418	-		-	BT05_STCS [R/W] B,H,W 00000000	
B480_941C	-		-/BT05_PSDR/-/- [R/W] H,W 00000000 00000000		
B480_9420	-		-/BT05_ADTR/-/- [R/W] H,W 00000000 00000000		
B480_9424	-		BT05_DEBUG [R/W] B,H,W 00000000 00000000		
B480_9428	-		-		
B480_942C	-		BT05_ETCDR [R/W] B,H,W 01111111 11111111		
B480_9430 to B480_97FC	-				
B480_9800	-		BT06_PCSR/BT06_PCSR/BT06_PRLL/- [R/W] H,W 00000000 00000000		Base Timer ch.6
B480_9804	-		-/BT06_PDUT/ BT06_PRLH/ BT06_DTBF [R/W] H,W 00000000 00000000		

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B480_9808	- BT06_TMR/ BT06_TMR/ BT06_TMR/- [R/W] H,W 00000000 00000000 00000000 00000000 / (*9) 00000000 00000000 / (*10) XXXXXXXX XXXXXXXX(*4)				
B480_980C	-		BT06_TMCR [R/W] B,H,W 00000000 00000000		
B480_9810	-		BT06_TMCR2 [R/W] B,H,W 00000000	BT06_STC [R/W] B,H,W 00000000	
B480_9814	-		-	BT06_STCC [R/W] B,H,W 00000000	
B480_9818	-		-	BT06_STCS [R/W] B,H,W 00000000	
B480_981C	-		-/BT06_PSDR/-/- [R/W] H,W 00000000 00000000		
B480_9820	-		-/BT06_ADTR/-/- [R/W] H,W 00000000 00000000		
B480_9824	-		BT06_DEBUG [R/W] B,H,W 00000000 00000000		
B480_9828	-		-		
B480_982C	-		BT06_ETCDR [R/W] B,H,W 01111111 11111111		
B480_9830	-			BT_BTSEL67 [R/W] B,H,W 11110000	
B480_9834 to B480_9BFC	-				
B480_9C00			BT07_PCSR/BT07_PCSR/BT07_PRLL/- [R/W] H,W 00000000 00000000		Base Timer ch.7
B480_9C04	-		-/BT07_PDUT/ BT07_PRLH/ BT07_DTBF [R/W] H,W 00000000 00000000		
B480_9C08	BT07_TMR/ BT07_TMR/ BT07_TMR/- [R/W] H,W 00000000 00000000 00000000 00000000 / (*9) 00000000 00000000 / (*10) XXXXXXXX XXXXXXXX(*4)				
B480_9C0C	-		BT07_TMCR [R/W] B,H,W 00000000 00000000		
B480_9C10	-		BT07_TMCR2 [R/W] B,H,W 00000000	BT07_STC [R/W] B,H,W 00000000	
B480_9C14	-		-	BT07_STCC [R/W] B,H,W 00000000	
B480_9C18	-		-	BT07_STCS [R/W] B,H,W 00000000	
B480_9C1C	-		-/BT07_PSDR/-/- [R/W] H,W 00000000 00000000		
B480_9C20	-		-/BT07_ADTR/-/- [R/W] H,W 00000000 00000000		

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B480_9C24			BT07_DEBUG [R/W] B,H,W 00000000 00000000		
B480_9C28	-		-		
B480_9C2C	-		BT07_ETCDR [R/W] B,H,W 01111111 11111111		
B480_9C30 to B480_9FFC	-				
B480_A000	-		BT08_PCSR/BT08_PCSR/BT08_PRLL/- [R/W] H,W 00000000 00000000		Base Timer ch.8
B480_A004	-		-/BT08_PDUT/ BT08_PRLH/ BT08_DTBF [R/W] H,W 00000000 00000000		
B480_A008	BT08_TMR/ BT08_TMR/ BT08_TMR/- [R/W] H,W 00000000 00000000 00000000 00000000 / (*9) 00000000 00000000 / (*10) XXXXXXXX XXXXXXXX(*4)				
B480_A00C	-		BT08_TMCR [R/W] B,H,W 00000000 00000000		
B480_A010	-		BT08_TMCR2 [R/W] B,H,W 00000000	BT08_STC [R/W] B,H,W 00000000	
B480_A014	-		-	BT08_STCC [R/W] B,H,W 00000000	
B480_A018	-		-	BT08_STCS [R/W] B,H,W 00000000	
B480_A01C	-		-/BT08_PSDR/-/- [R/W] H,W 00000000 00000000		
B480_A020	-		-/BT08_ADTR/-/- [R/W] H,W 00000000 00000000		
B480_A024	-		BT08_DEBUG [R/W] B,H,W 00000000 00000000		
B480_A028	-		-		
B480_A02C	-		BT08_ETCDR [R/W] B,H,W 01111111 11111111		
B480_A030				BT_BTSEL89 [R/W] B,H,W 11110000	
B480_A034 to B480_A3FC	-				
B480_A400	-		BT09_PCSR/BT09_PCSR/BT09_PRLL/- [R/W] H,W 00000000 00000000		Base Timer ch.9
B480_A404	-		-/BT09_PDUT/ BT09_PRLH/ BT09_DTBF [R/W] H,W 00000000 00000000		
B480_A408	BT09_TMR/ BT09_TMR/ BT09_TMR/- [R/W] H,W 00000000 00000000 00000000 00000000 / (*9) 00000000 00000000 / (*10) XXXXXXXX XXXXXXXX(*4)				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B480_A40C	-		BT09_TMCR [R/W] B,H,W 00000000 00000000		
B480_A410	-		BT09_TMCR2 [R/W] B,H,W 00000000	BT09_STC [R/W] B,H,W 00000000	
B480_A414	-		-	BT09_STCC [R/W] B,H,W 00000000	
B480_A418	-		-	BT09_STCS [R/W] B,H,W 00000000	
B480_A41C	-		-/BT09_PSDR/-/- [R/W] H,W 00000000 00000000		
B480_A420	-		-/BT09_ADTR/-/- [R/W] H,W 00000000 00000000		
B480_A424	-		BT09_DEBUG [R/W] B,H,W 00000000 00000000		
B480_A428	-		-		
B480_A42C	-		BT09_ETCDR [R/W] B,H,W 01111111 11111111		
B480_A430 to B480_A7FC	-				
B480_A800	-		BT10_PCSR/BT10_PCSR/BT10_PRLL/- [R/W] H,W 00000000 00000000		Base Timer ch.10
B480_A804	-		-/BT10_PDUT/ BT10_PRLH/ BT10_DTBF [R/W] H,W 00000000 00000000		
B480_A808	BT10_TMR/ BT10_TMR/ BT10_TMR/- [R/W] H,W 00000000 00000000 00000000 00000000 / (*9) 00000000 00000000 / (*10) XXXXXXXX XXXXXXXX(*4)				
B480_A80C	-		BT10_TMCR [R/W] B,H,W 00000000 00000000		
B480_A810	-		BT10_TMCR2 [R/W] B,H,W 00000000	BT10_STC [R/W] B,H,W 00000000	
B480_A814	-		-	BT10_STCC [R/W] B,H,W 00000000	
B480_A818	-		-	BT10_STCS [R/W] B,H,W 00000000	
B480_A81C	-		-/BT10_PSDR/-/- [R/W] H,W 00000000 00000000		
B480_A820	-		-/BT10_ADTR/-/- [R/W] H,W 00000000 00000000		
B480_A824	-		BT10_DEBUG [R/W] B,H,W 00000000 00000000		
B480_A828	-		-		
B480_A82C	-		BT10_ETCDR [R/W] B,H,W 01111111 11111111		

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B480_A830				BT_BTSEL1011 [R/W] B,H,W 11110000	
B480_A834 to B480_ABFC	-				
B480_AC00	-		BT11_PCSR/BT11_PCSR/BT11_PRLL/- [R/W] H,W 00000000 00000000		Base Timer ch.11
B480_AC04	-		-/BT11_PDUT/ BT11_PRLH/ BT11_DTBF [R/W] H,W 00000000 00000000		
B480_AC08	BT11_TMR/ BT11_TMR/ BT11_TMR/- [R/W] H,W 00000000 00000000 00000000 00000000 / (*9) 00000000 00000000 / (*10) XXXXXXXX XXXXXXXX(*4)				
B480_AC0C	-		BT11_TMCR [R/W] B,H,W 00000000 00000000		
B480_AC10	-		BT11_TMCR2 [R/W] B,H,W 00000000	BT11_STC [R/W] B,H,W 00000000	
B480_AC14	-		-	BT11_STCC [R/W] B,H,W 00000000	
B480_AC18	-		-	BT11_STCS [R/W] B,H,W 00000000	
B480_AC1C	-		-/BT11_PSDR/-/- [R/W] H,W 00000000 00000000		
B480_AC20	-		-/BT11_ADTR/-/- [R/W] H,W 00000000 00000000		
B480_AC24	-		BT11_DEBUG [R/W] B,H,W 00000000 00000000		
B480_AC28	-		-		
B480_AC2C	-		BT11_ETCDR [R/W] B,H,W 01111111 11111111		
B480_AC30 to B480_AFFC	-				
B480_B000 to B480_FFFC	-				Reserved
B481_0000	RLT0_DMACFG [R/W] B,H,W 00000000 00000000 00000000 00000000				Reload Timer ch.0
B481_0004	-				
B481_0008	RLT0_TMCSR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B481_000C	-				
B481_0010	RLT0_TMRLR [R/W] W,DW XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B481_0014	RLT0_TMR [R/W] W,DW XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
B481_0018 to B481_03FC	-				
B481_0400	RLT1_DMACFG [R/W] B,H,W 00000000 00000000 00000000 00000000				Reload Timer ch.1
B481_0404	-				
B481_0408	RLT1_TMCSR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B481_040C	-				
B481_0410	RLT1_TMRLR [R/W] W,DW XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
B481_0414	RLT1_TMR [R/W] W,DW XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
B481_0418 to B481_07FC	-				
B481_0800	RLT2_DMACFG [R/W] B,H,W 00000000 00000000 00000000 00000000				Reload Timer ch.2
B481_0804	-				
B481_0808	RLT2_TMCSR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B481_080C	-				
B481_0810	RLT2_TMRLR [R/W] W,DW XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
B481_0814	RLT2_TMR [R/W] W,DW XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
B481_0818 to B481_0BFC	-				
B481_0C00	RLT3_DMACFG [R/W] B,H,W 00000000 00000000 00000000 00000000				Reload Timer ch.3
B481_0C04	-				
B481_0C08	RLT3_TMCSR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B481_0C0C	-				
B481_0C10	RLT3_TMRLR [R/W] W,DW XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B481_0C14	RLT3_TMR [R/W] W,DW XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
B481_0C18 to B481_0FFC	-				
B481_1000 to B481_FFFC	-				Reserved
B482_0000	FRT00_CPCLRB/FRT00_CPCLR [R/W] W 11111111 11111111 11111111 11111111				FRT ch.0
B482_0004	FRT00_TCDT [R/W] W 00000000 00000000 00000000 00000000				
B482_0008	FRT00_TCCS [R/W] B,H,W 11111111 11111111 00000000 01000000				
B482_000C	FRT00_TECCS [R/W] B,H,W 11111111 11111111 00000000 11111111				
B482_0010	FRT00_TCCSC [R/W] B,H,W 00000000 00000000 00000000 00000000				
B482_0014	FRT00_TCCSS [R/W] B,H,W 00000000 00000000 00000000 00000000				
B482_0018	-		FRT00_DEBUG [R/W] B,H,W 00000000 00000000		
B482_001C to B482_03FC	-				
B482_0400	FRT01_CPCLRB/FRT01_CPCLR [R/W] W 11111111 11111111 11111111 11111111				FRT ch.1
B482_0404	FRT01_TCDT [R/W] W 00000000 00000000 00000000 00000000				
B482_0408	FRT01_TCCS [R/W] B,H,W 11111111 11111111 00000000 01000000				
B482_040C	FRT01_TECCS [R/W] B,H,W 11111111 11111111 00000000 11111111				
B482_0410	FRT01_TCCSC [R/W] B,H,W 00000000 00000000 00000000 00000000				
B482_0414	FRT01_TCCSS [R/W] B,H,W 00000000 00000000 00000000 00000000				
B482_0418	-		FRT01_DEBUG [R/W] B,H,W 00000000 00000000		
B482_041C to B482_07FC	-				
B482_0800	FRT02_CPCLRB/FRT02_CPCLR [R/W] W 11111111 11111111 11111111 11111111				FRT ch.2
B482_0804	FRT02_TCDT [R/W] W 00000000 00000000 00000000 00000000				
B482_0808	FRT02_TCCS [R/W] B,H,W 11111111 11111111 00000000 01000000				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B482_080C	FRT02_TECCS [R/W] B,H,W 11111111 11111111 00000000 11111111				
B482_0810	FRT02_TCCSC [R/W] B,H,W 00000000 00000000 00000000 00000000				
B482_0814	FRT02_TCCSS [R/W] B,H,W 00000000 00000000 00000000 00000000				
B482_0818	-		FRT02_DEBUG [R/W] B,H,W 00000000 00000000		
B482_081C to B482_0BFC	-				
B482_0C00	FRT03_CPCLRB/FRT03_CPCLR [R/W] W 11111111 11111111 11111111 11111111				FRT ch.3
B482_0C04	FRT03_TCDT [R/W] W 00000000 00000000 00000000 00000000				
B482_0C08	FRT03_TCCS [R/W] B,H,W 11111111 11111111 00000000 01000000				
B482_0C0C	FRT03_TECCS [R/W] B,H,W 11111111 11111111 00000000 11111111				
B482_0C10	FRT03_TCCSC [R/W] B,H,W 00000000 00000000 00000000 00000000				
B482_0C14	FRT03_TCCSS [R/W] B,H,W 00000000 00000000 00000000 00000000				
B482_0C18	-		FRT03_DEBUG [R/W] B,H,W 00000000 00000000		
B482_0C1C to B482_0FFC	-				
B482_1000	FRT04_CPCLRB/FRT04_CPCLR [R/W] W 11111111 11111111 11111111 11111111				FRT ch.4
B482_1004	FRT04_TCDT [R/W] W 00000000 00000000 00000000 00000000				
B482_1008	FRT04_TCCS [R/W] B,H,W 11111111 11111111 00000000 01000000				
B482_100C	FRT04_TECCS [R/W] B,H,W 11111111 11111111 00000000 11111111				
B482_1010	FRT04_TCCSC [R/W] B,H,W 00000000 00000000 00000000 00000000				
B482_1014	FRT04_TCCSS [R/W] B,H,W 00000000 00000000 00000000 00000000				
B482_1018	-		FRT04_DEBUG [R/W] B,H,W 00000000 00000000		
B482_101C to B482_13FC	-				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B482_1400	FRT05_CPCLRB/FRT05_CPCLR [R/W] W 11111111 11111111 11111111 11111111				FRT ch.5
B482_1404	FRT05_TCDT [R/W] W 00000000 00000000 00000000 00000000				
B482_1408	FRT05_TCCS [R/W] B,H,W 11111111 11111111 00000000 01000000				
B482_140C	FRT05_TECCS [R/W] B,H,W 11111111 11111111 00000000 11111111				
B482_1410	FRT05_TCCSC [R/W] B,H,W 00000000 00000000 00000000 00000000				
B482_1414	FRT05_TCCSS [R/W] B,H,W 00000000 00000000 00000000 00000000				
B482_1418	-		FRT05_DEBUG [R/W] B,H,W 00000000 00000000		
B482_141C to B482_17FC					
B482_1800 to B482_7FFC	-				Reserved
B482_8000	ICU00_IPCP0 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				ICU ch.0, 1
B482_8004	ICU00_IPCP1 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
B482_8008	ICU00_ICS [R/W] B,H,W 11111111 11111111 11111100 00000000				
B482_800C	ICU00_ICSC [R/W] B,H,W 00000000 00000000 00000000 00000000				
B482_8010	ICU00_ICSS [R/W] B,H,W 00000000 00000000 00000000 00000000				
B482_8014 to B482_83FC	-				
B482_8400	ICU01_IPCP0 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				ICU ch.2, 3
B482_8404	ICU01_IPCP1 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
B482_8408	ICU01_ICS [R/W] B,H,W 11111111 11111111 11111100 00000000				
B482_840C	ICU01_ICSC [R/W] B,H,W 00000000 00000000 00000000 00000000				
B482_8410	ICU01_ICSS [R/W] B,H,W 00000000 00000000 00000000 00000000				
B482_8414 to B482_87FC	-				
B482_8800	ICU02_IPCP0 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				ICU ch.4, 5

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B482_8804	ICU02_IPCP1 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
B482_8808	ICU02_ICS [R/W] B,H,W 11111111 11111111 11111100 00000000				
B482_880C	ICU02_ICSC [R/W] B,H,W 00000000 00000000 00000000 00000000				
B482_8810	ICU02_ICSS [R/W] B,H,W 00000000 00000000 00000000 00000000				
B482_8814 to B482_8BFC	-				
B482_8C00	ICU03_IPCP0 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				ICU ch.6, 7
B482_8C04	ICU03_IPCP1 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
B482_8C08	ICU03_ICS [R/W] B,H,W 11111111 11111111 11111100 00000000				
B482_8C0C	ICU03_ICSC [R/W] B,H,W 00000000 00000000 00000000 00000000				
B482_8C10	ICU03_ICSS [R/W] B,H,W 00000000 00000000 00000000 00000000				
B482_8C14 to B482_8FFC	-				
B482_9000	ICU04_IPCP0 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				ICU ch.8, 9
B482_9004	ICU04_IPCP1 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
B482_9008	ICU04_ICS [R/W] B,H,W 11111111 11111111 11111100 00000000				
B482_900C	ICU04_ICSC [R/W] B,H,W 00000000 00000000 00000000 00000000				
B482_9010	ICU04_ICSS [R/W] B,H,W 00000000 00000000 00000000 00000000				
B482_9014 to B482_93FC	-				
B482_9400	ICU05_IPCP0 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				ICU ch.10, 11
B482_9404	ICU05_IPCP1 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
B482_9408	ICU05_ICS [R/W] B,H,W 11111111 11111111 11111100 00000000				
B482_940C	ICU05_ICSC [R/W] B,H,W 00000000 00000000 00000000 00000000				
B482_9410	ICU05_ICSS [R/W] B,H,W 00000000 00000000 00000000 00000000				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B482_9414 to B482_97FC	-				
B482_9800 to B482_FFFC	-				Reserved
B483_0000	OCU00_OCCPB0/OCU00_OCCP0 [R/W] W 00000000 00000000 00000000 00000000				OCU ch.0, 1
B483_0004	OCU00_OCCPB1/OCU00_OCCP1 [R/W] W 00000000 00000000 00000000 00000000				
B483_0008	OCU00_OCS [R/W] B,H,W 11111111 11111111 11100000 00001100				
B483_000C	OCU00_OCSC [R/W] B,H,W 00000000 00000000 00000000 00000000				
B483_0010	OCU00_OCSS [R/W] B,H,W 00000000 00000000 00000000 00000000				
B483_0014 to B483_03FC	-				
B483_0400	OCU01_OCCPB0/OCU01_OCCP0 [R/W] W 00000000 00000000 00000000 00000000				OCU ch.2, 3
B483_0404	OCU01_OCCPB1/OCU01_OCCP1 [R/W] W 00000000 00000000 00000000 00000000				
B483_0408	OCU01_OCS [R/W] B,H,W 11111111 11111111 11100000 00001100				
B483_040C	OCU01_OCSC [R/W] B,H,W 00000000 00000000 00000000 00000000				
B483_0410	OCU01_OCSS [R/W] B,H,W 00000000 00000000 00000000 00000000				
B483_0414 to B483_07FC	-				
B483_0800	OCU02_OCCPB0/OCU02_OCCP0 [R/W] W 00000000 00000000 00000000 00000000				OCU ch.4, 5
B483_0804	OCU02_OCCPB1/OCU02_OCCP1 [R/W] W 00000000 00000000 00000000 00000000				
B483_0808	OCU02_OCS [R/W] B,H,W 11111111 11111111 11100000 00001100				
B483_080C	OCU02_OCSC [R/W] B,H,W 00000000 00000000 00000000 00000000				
B483_0810	OCU02_OCSS [R/W] B,H,W 00000000 00000000 00000000 00000000				
B483_0814 to B483_0BFC	-				
B483_0C00	OCU03_OCCPB0/OCU03_OCCP0 [R/W] W 00000000 00000000 00000000 00000000				OCU ch.6, 7

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B483_0C04	OCU03_OCCPB1/OCU03_OCCP1 [R/W] W 00000000 00000000 00000000 00000000				
B483_0C08	OCU03_OCS [R/W] B,H,W 11111111 11111111 11100000 00001100				
B483_0C0C	OCU03_OCSC [R/W] B,H,W 00000000 00000000 00000000 00000000				
B483_0C10	OCU03_OCSS [R/W] B,H,W 00000000 00000000 00000000 00000000				
B483_0C14 to B483_0FFC	-				
B483_1000	OCU04_OCCPB0/OCU04_OCCP0 [R/W] W 00000000 00000000 00000000 00000000				OCU ch.8, 9
B483_1004	OCU04_OCCPB1/OCU04_OCCP1 [R/W] W 00000000 00000000 00000000 00000000				
B483_1008	OCU04_OCS [R/W] B,H,W 11111111 11111111 11100000 00001100				
B483_100C	OCU04_OCSC [R/W] B,H,W 00000000 00000000 00000000 00000000				
B483_1010	OCU04_OCSS [R/W] B,H,W 00000000 00000000 00000000 00000000				
B483_1014 to B483_13FC	-				
B483_1400	OCU05_OCCPB0/OCU05_OCCP0 [R/W] W 00000000 00000000 00000000 00000000				OCU ch.10, 11
B483_1404	OCU05_OCCPB1/OCU05_OCCP1 [R/W] W 00000000 00000000 00000000 00000000				
B483_1408	OCU05_OCS [R/W] B,H,W 11111111 11111111 11100000 00001100				
B483_140C	OCU05_OCSC [R/W] B,H,W 00000000 00000000 00000000 00000000				
B483_1410	OCU05_OCSS [R/W] B,H,W 00000000 00000000 00000000 00000000				
B483_1414 to B483_17FC	-				
B483_1800 to B483_FBFC	-				Reserved
B483_FC00	CPG0_RLT_TSEL [R/W] B,H,W 00000000 00000000 00000000 00000000				CPERI #0 Misc_Registers
B483_FC04	CPG0_RLT_SSSR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B483_FC08 to B483_FFFC	-				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B484_0000 to B484_FFFC	-				Apps#5 Area
B485_0000 to B487_FFFC	-				Reserved
B488_0000	CPG_MFS08_SSR [R/W] B,H,W 00000011	CPG_MFS08_ESCR/CPG_MFS08_ESCR/CPG_MFS08_ESCR/CPG_MFS08_IBSR [R/W] B,H,W 00000000	CPG_MFS08_SCR/CPG_MFS08_SCR/CPG_MFS08_SCR/CPG_MFS08_IBCR [R/W] B,H,W 00000000	CPG_MFS08_SMR [R/W] B,H,W 00000000	CPG_MFS ch.8
B488_0004	-/CPG_MFS08_RDR(TDR)/-/- [R/W] B,H,W --------(*5)	-/CPG_MFS08_RDR(TDR)/-/- [R/W] B,H,W --------(*5)	CPG_MFS08_RDR(TDR)/CPG_MFS08_RDR(TDR)/-/- [R/W] B,H,W -------0	CPG_MFS08_RDR(TDR) [R/W] B,H,W 00000000	
B488_0008	CPG_MFS08_STMR [R/W] B,H,W 00000000 00000000		CPG_MFS08_SACSR [R/W] B,H,W 00000000 00000000		
B488_000C	-/CPG_MFS08_SCSCR/CPG_MFS08_SFUR/- [R/W] B,H,W --------(*5)(*6)	-/CPG_MFS08_SCSCR/CPG_MFS08_SFUR/- [R/W] B,H,W --------(*5)(*6)	CPG_MFS08_STMCR [R/W] B,H,W 00000000 00000000		
B488_0010	-/CPG_MFS08_SCSTR3/CPG_MFS08_LAMSR/- [R/W] B,H,W --------(*5)(*6)	-/CPG_MFS08_SCSTR2/CPG_MFS08_LAMCR/- [R/W] B,H,W --------(*5)(*6)	-/CPG_MFS08_SCSTR1/CPG_MFS08_SFLR1/CPG_MFS08_EIBCR [R/W] B,H,W --------(*5)(*6)(*7)	-/CPG_MFS08_SCSTR0/CPG_MFS08_SFLR0/CPG_MFS08_NFCR [R/W] B,H,W --------(*5)(*6)(*7)	
B488_0014	-	-/CPG_MFS08_SCSFR2/-/- [R/W] B,H,W --------(*5)	-/CPG_MFS08_SCSFR1/-/- [R/W] B,H,W --------(*5)	-/CPG_MFS08_SCSFR0/-/- [R/W] B,H,W --------(*5)	
B488_0018	-/CPG_MFS08_TBYTE3/CPG_MFS08_LAMESR/- [R/W] B,H,W --------(*5)(*6)	-/CPG_MFS08_TBYTE2/CPG_MFS08_LAMERT/- [R/W] B,H,W --------(*5)(*6)	-/CPG_MFS08_TBYTE1/CPG_MFS08_LAMIER/- [R/W] B,H,W --------(*5)(*6)	CPG_MFS08_TBYTE0/CPG_MFS08_TBYTE0/CPG_MFS08_LAMRID(LAMTID) /- [R/W] B,H,W 00000000	
B488_001C	-/-/-/CPG_MFS08_ISMK [R/W] B,H,W --------(*7)	-/-/-/CPG_MFS08_ISBA [R/W] B,H,W --------(*7)	CPG_MFS08_BGR1 [R/W] H,W 00000000	CPG_MFS08_BGR0 [R/W] H,W 00000000	
B488_0020	CPG_MFS08_FBYTE2 [R/W] B,H,W 00000000	CPG_MFS08_FBYTE1 [R/W] B,H,W 00000000	CPG_MFS08_FCR1 [R/W] B,H,W 00000100	CPG_MFS08_FCR0 [R/W] B,H,W 00000000	
B488_0024	CPG_MFS08_ESR [R/W] B,H,W 00000000	CPG_MFS08_ECR [R/W] B,H,W 00000000	CPG_MFS08_FTICR2 [R/W] B,H,W 00000000	CPG_MFS08_FTICR1 [R/W] B,H,W 00000000	
B488_0028	-			CPG_MFS08_TBSIZE [R/W] B,H,W 00000000	
B488_002C	-/-/CPG_MFS08_SSRC/CPG_MFS08_SSRC [R/W] B,H,W --------(*6)(*7)	-/-/CPG_MFS08_ESCRC/CPG_MFS08_IBSRC [R/W] B,H,W --------(*6)(*7)	-/-/CPG_MFS08_SCRC/CPG_MFS08_IBCRC [R/W] B,H,W --------(*6)(*7)	-/-/CPG_MFS08_SMRC/CPG_MFS08_SMRC [R/W] B,H,W --------(*6)(*7)	
B488_0030	-		CPG_MFS08_SACSRC [R/W] B,H,W 00000000 00000000		
B488_0034	-/-/CPG_MFS08_LAMSRC/- [R/W] B,H,W --------(*6)	-/-/CPG_MFS08_LAMCRC/- [R/W] B,H,W --------(*6)	-		
B488_0038	-/-/CPG_MFS08_LAMESRC/- [R/W] B,H,W --------(*6)	-	-/-/CPG_MFS08_LAMIERC/- [R/W] B,H,W --------(*6)	-	

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B488_003C	-		CPG_MFS08_FCR1C [R/W] B,H,W 00000000	CPG_MFS08_FCR0C [R/W] B,H,W 00000000	
B488_0040	CPG_MFS08_ESRC [R/W] B,H,W 00000000	-			
B488_0044	-/-/CPG_MFS08_SSRS/CPG_MFS08_SSRS [R/W] B,H,W --------(*6)(*7)	-/-/CPG_MFS08_ESCRS/- [R/W] B,H,W --------(*6)	-/-/CPG_MFS08_SCRS/CPG_MFS08_IBCRS [R/W] B,H,W --------(*6)(*7)	-/-/CPG_MFS08_SMRS/CPG_MFS08_SMRS [R/W] B,H,W --------(*6)(*7)	
B488_0048	-		CPG_MFS08_SACSRS [R/W] B,H,W 00000000 00000000		
B488_004C		-/-/CPG_MFS08_LAMCRS/- [R/W] B,H,W --------(*6)			
B488_0050	-		-/-/CPG_MFS08_LAMIERS/- [R/W] B,H,W --------(*6)	-	
B488_0054	-		CPG_MFS08_FCR1S [R/W] B,H,W 00000000	CPG_MFS08_FCR0S [R/W] B,H,W 00000000	
B488_0058	CPG_MFS08_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS08_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS08_RDR1(TDR1)[R/W] B,H,W	CPG_MFS08_RDR0(TDR0)[R/W] B,H,W	
B488_005C	CPG_MFS08_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS08_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS08_RDR1(TDR1)[R/W] B,H,W	CPG_MFS08_RDR0(TDR0)[R/W] B,H,W	
B488_0060	CPG_MFS08_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS08_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS08_RDR1(TDR1)[R/W] B,H,W	CPG_MFS08_RDR0(TDR0)[R/W] B,H,W	
B488_0064	CPG_MFS08_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS08_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS08_RDR1(TDR1)[R/W] B,H,W	CPG_MFS08_RDR0(TDR0)[R/W] B,H,W	
B488_0068	CPG_MFS08_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS08_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS08_RDR1(TDR1)[R/W] B,H,W	CPG_MFS08_RDR0(TDR0)[R/W] B,H,W	
B488_006C	CPG_MFS08_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS08_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS08_RDR1(TDR1)[R/W] B,H,W	CPG_MFS08_RDR0(TDR0)[R/W] B,H,W	
B488_0070	CPG_MFS08_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS08_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS08_RDR1(TDR1)[R/W] B,H,W	CPG_MFS08_RDR0(TDR0)[R/W] B,H,W	
B488_0074	CPG_MFS08_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS08_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS08_RDR1(TDR1)[R/W] B,H,W	CPG_MFS08_RDR0(TDR0)[R/W] B,H,W	
B488_0078	CPG_MFS08_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS08_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS08_RDR1(TDR1)[R/W] B,H,W	CPG_MFS08_RDR0(TDR0)[R/W] B,H,W	
B488_007C	CPG_MFS08_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS08_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS08_RDR1(TDR1)[R/W] B,H,W	CPG_MFS08_RDR0(TDR0)[R/W] B,H,W	
B488_0080	CPG_MFS08_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS08_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS08_RDR1(TDR1)[R/W] B,H,W	CPG_MFS08_RDR0(TDR0)[R/W] B,H,W	
B488_0084	CPG_MFS08_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS08_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS08_RDR1(TDR1)[R/W] B,H,W	CPG_MFS08_RDR0(TDR0)[R/W] B,H,W	
B488_0088	CPG_MFS08_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS08_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS08_RDR1(TDR1)[R/W] B,H,W	CPG_MFS08_RDR0(TDR0)[R/W] B,H,W	
B488_008C	CPG_MFS08_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS08_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS08_RDR1(TDR1)[R/W] B,H,W	CPG_MFS08_RDR0(TDR0)[R/W] B,H,W	
B488_0090	CPG_MFS08_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS08_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS08_RDR1(TDR1)[R/W] B,H,W	CPG_MFS08_RDR0(TDR0)[R/W] B,H,W	
B488_0094	CPG_MFS08_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS08_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS08_RDR1(TDR1)[R/W] B,H,W	CPG_MFS08_RDR0(TDR0)[R/W] B,H,W	
B488_0098 to B488_03FC	-				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B488_0400	CPG_MFS09_SSR [R/W] B,H,W 00000011	CPG_MFS09_ESCR/CPG_MFS09_ESCR/CPG_MFS09_ESCR/CPG_MFS09_IBSR [R/W] B,H,W 00000000	CPG_MFS09_SCR/CPG_MFS09_SCR/CPG_MFS09_SCR/CPG_MFS09_IBCR [R/W] B,H,W 00000000	CPG_MFS09_SMR [R/W] B,H,W 00000000	CPG_MFS ch.9
B488_0404	-/CPG_MFS09_RDR(TDR)/-/- [R/W] B,H,W --------(*5)	-/CPG_MFS09_RDR(TDR)/-/- [R/W] B,H,W --------(*5)	CPG_MFS09_RDR(TDR)/CPG_MFS09_RDR(TDR)/-/- [R/W] B,H,W -------0	CPG_MFS09_RDR(TDR) [R/W] B,H,W 00000000	
B488_0408	CPG_MFS09_STMR [R/W] B,H,W 00000000 00000000		CPG_MFS09_SACSR [R/W] B,H,W 00000000 00000000		
B488_040C	-/CPG_MFS09_SCSCR/CPG_MFS09_SFUR/- [R/W] B,H,W --------(*5)(*6)	-/CPG_MFS09_SCSCR/CPG_MFS09_SFUR/- [R/W] B,H,W --------(*5)(*6)	CPG_MFS09_STMCR [R/W] B,H,W 00000000 00000000		
B484_0410	-/CPG_MFS09_SCSTR3/CPG_MFS09_LAMSR/- [R/W] B,H,W --------(*5)(*6)	-/CPG_MFS09_SCSTR2/CPG_MFS09_LAMCR/- [R/W] B,H,W --------(*5)(*6)	-/CPG_MFS09_SCSTR1/CPG_MFS09_SFLR1/CPG_MFS09_EIBCR [R/W] B,H,W --------(*5)(*6)(*7)	-/CPG_MFS09_SCSTR0/CPG_MFS09_SFLR0/CPG_MFS09_NFCR [R/W] B,H,W --------(*5)(*6)(*7)	
B488_0414	-	-/CPG_MFS09_SCSFR2/-/- [R/W] B,H,W --------(*5)	-/CPG_MFS09_SCSFR1/-/- [R/W] B,H,W --------(*5)	-/CPG_MFS09_SCSFR0/-/- [R/W] B,H,W --------(*5)	
B488_0418	-/CPG_MFS09_TBYTE3/CPG_MFS09_LAMESR/- [R/W] B,H,W --------(*5)(*6)	-/CPG_MFS09_TBYTE2/CPG_MFS09_LAMERT/- [R/W] B,H,W --------(*5)(*6)	-/CPG_MFS09_TBYTE1/CPG_MFS09_LAMIER/- [R/W] B,H,W --------(*5)(*6)	CPG_MFS09_TBYTE0/CPG_MFS09_TBYTE0/CPG_MFS09_LAMRID(LAMTID)/- [R/W] B,H,W 00000000	
B488_041C	-/-/-/CPG_MFS09_ISMK [R/W] B,H,W --------(*7)	-/-/-/CPG_MFS09_ISBA [R/W] B,H,W --------(*7)	CPG_MFS09_BGR1 [R/W] H,W 00000000	CPG_MFS09_BGR0 [R/W] H,W 00000000	
B488_0420	CPG_MFS09_FBYTE2 [R/W] B,H,W 00000000	CPG_MFS09_FBYTE1 [R/W] B,H,W 00000000	CPG_MFS09_FCR1 [R/W] B,H,W 00000100	CPG_MFS09_FCR0 [R/W] B,H,W 00000000	
B488_0424	CPG_MFS09_ESR [R/W] B,H,W 00000000	CPG_MFS09_ECR [R/W] B,H,W 00000000	CPG_MFS09_FTICR2 [R/W] B,H,W 00000000	CPG_MFS09_FTICR1 [R/W] B,H,W 00000000	
B488_0428	-			CPG_MFS09_TBSIZE [R/W] B,H,W 00000000	
B488_042C	-/-/CPG_MFS09_SSRC/CPG_MFS09_SSRC [R/W] B,H,W --------(*6)(*7)	-/-/CPG_MFS09_ESCRC/CPG_MFS09_IBSRC [R/W] B,H,W --------(*6)(*7)	-/-/CPG_MFS09_SCRC/CPG_MFS09_IBCRC [R/W] B,H,W --------(*6)(*7)	-/-/CPG_MFS09_SMRC/CPG_MFS09_SMRC [R/W] B,H,W --------(*6)(*7)	
B488_0430	-		CPG_MFS09_SACSRC [R/W] B,H,W 00000000 00000000		
B488_0434	-/-/CPG_MFS09_LAMSRC/- [R/W] B,H,W --------(*6)	-/-/CPG_MFS09_LAMCRC/- [R/W] B,H,W --------(*6)	-		
B488_0438	-/-/CPG_MFS09_LAMESRC/- [R/W] B,H,W --------(*6)	-	-/-/CPG_MFS09_LAMIERC/- [R/W] B,H,W --------(*6)	-	
B488_043C	-		CPG_MFS09_FCR1C [R/W] B,H,W 00000000	CPG_MFS09_FCR0C [R/W] B,H,W 00000000	
B488_0440	CPG_MFS09_ESRC [R/W] B,H,W 00000000	-			

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B488_0444	-/-/CPG_MFS09_SSRS/CPG_MFS09_SSRS [R/W] B,H,W --------(*6)(*7)	-/-/CPG_MFS09_ESCRS/- [R/W] B,H,W --------(*6)	-/-/CPG_MFS09_SCRS/CPG_MFS09_IBCRS [R/W] B,H,W --------(*6)(*7)	-/-/CPG_MFS09_SMRS/CPG_MFS09_SMRS [R/W] B,H,W --------(*6)(*7)	
B488_0448	-		CPG_MFS09_SACSRS [R/W] B,H,W 00000000 00000000		
B488_044C		-/-/CPG_MFS09_LAMCRS/- [R/W] B,H,W --------(*6)	-		
B488_0450	-		-/-/CPG_MFS09_LAMIERS/- [R/W] B,H,W --------(*6)	-	
B488_0454	-		CPG_MFS09_FCR1S [R/W] B,H,W 00000000	CPG_MFS09_FCR0S [R/W] B,H,W 00000000	
B488_0458	CPG_MFS09_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS09_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS09_RDR1(TDR1)[R/W] B,H,W	CPG_MFS09_RDR0(TDR0)[R/W] B,H,W	
B488_045C	CPG_MFS09_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS09_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS09_RDR1(TDR1)[R/W] B,H,W	CPG_MFS09_RDR0(TDR0)[R/W] B,H,W	
B488_0460	CPG_MFS09_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS09_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS09_RDR1(TDR1)[R/W] B,H,W	CPG_MFS09_RDR0(TDR0)[R/W] B,H,W	
B488_0464	CPG_MFS09_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS09_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS09_RDR1(TDR1)[R/W] B,H,W	CPG_MFS09_RDR0(TDR0)[R/W] B,H,W	
B488_0468	CPG_MFS09_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS09_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS09_RDR1(TDR1)[R/W] B,H,W	CPG_MFS09_RDR0(TDR0)[R/W] B,H,W	
B488_046C	CPG_MFS09_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS09_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS09_RDR1(TDR1)[R/W] B,H,W	CPG_MFS09_RDR0(TDR0)[R/W] B,H,W	
B488_0470	CPG_MFS09_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS09_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS09_RDR1(TDR1)[R/W] B,H,W	CPG_MFS09_RDR0(TDR0)[R/W] B,H,W	
B488_0474	CPG_MFS09_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS09_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS09_RDR1(TDR1)[R/W] B,H,W	CPG_MFS09_RDR0(TDR0)[R/W] B,H,W	
B488_0478	CPG_MFS09_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS09_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS09_RDR1(TDR1)[R/W] B,H,W	CPG_MFS09_RDR0(TDR0)[R/W] B,H,W	
B488_047C	CPG_MFS09_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS09_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS09_RDR1(TDR1)[R/W] B,H,W	CPG_MFS09_RDR0(TDR0)[R/W] B,H,W	
B488_0480	CPG_MFS09_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS09_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS09_RDR1(TDR1)[R/W] B,H,W	CPG_MFS09_RDR0(TDR0)[R/W] B,H,W	
B488_0484	CPG_MFS09_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS09_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS09_RDR1(TDR1)[R/W] B,H,W	CPG_MFS09_RDR0(TDR0)[R/W] B,H,W	
B488_0488	CPG_MFS09_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS09_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS09_RDR1(TDR1)[R/W] B,H,W	CPG_MFS09_RDR0(TDR0)[R/W] B,H,W	
B488_048C	CPG_MFS09_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS09_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS09_RDR1(TDR1)[R/W] B,H,W	CPG_MFS09_RDR0(TDR0)[R/W] B,H,W	
B488_0490	CPG_MFS09_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS09_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS09_RDR1(TDR1)[R/W] B,H,W	CPG_MFS09_RDR0(TDR0)[R/W] B,H,W	
B488_0494	CPG_MFS09_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS09_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS09_RDR1(TDR1)[R/W] B,H,W	CPG_MFS09_RDR0(TDR0)[R/W] B,H,W	
B488_0498 to B488_07FC	-				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B488_0800	CPG_MFS10_SSR [R/W] B,H,W 00000011	CPG_MFS10_ESCR/CPG_MFS10_ESCR/CPG_MFS10_ESCR/CPG_MFS10_IBSR [R/W] B,H,W 00000000	CPG_MFS10_SCR/CPG_MFS10_SCR/CPG_MFS10_SCR/CPG_MFS10_IBCR [R/W] B,H,W 00000000	CPG_MFS10_SMR [R/W] B,H,W 00000000	CPG_MFS ch.10
B488_0804	-/CPG_MFS10_RDR(TDR)/-/- [R/W] B,H,W --------(*5)	-/CPG_MFS10_RDR(TDR)/-/- [R/W] B,H,W --------(*5)	CPG_MFS10_RDR(TDR)/CPG_MFS10_RDR(TDR)/-/- [R/W] B,H,W -------0	CPG_MFS10_RDR(TDR) [R/W] B,H,W 00000000	
B488_0808	CPG_MFS10_STMR [R/W] B,H,W 00000000 00000000		CPG_MFS10_SACSR [R/W] B,H,W 00000000 00000000		
B488_080C	-/CPG_MFS10_SCSCR/CPG_MFS10_SFUR/- [R/W] B,H,W --------(*5)(*6)	-/CPG_MFS10_SCSCR/CPG_MFS10_SFUR/- [R/W] B,H,W --------(*5)(*6)	CPG_MFS10_STMCR [R/W] B,H,W 00000000 00000000		
B488_0810	-/CPG_MFS10_SCSTR3/CPG_MFS10_LAMSR/- [R/W] B,H,W --------(*5)(*6)	-/CPG_MFS10_SCSTR2/CPG_MFS10_LAMCR/- [R/W] B,H,W --------(*5)(*6)	-/CPG_MFS10_SCSTR1/CPG_MFS10_SFLR1/CPG_MFS10_EIBCR [R/W] B,H,W --------(*5)(*6)(*7)	-/CPG_MFS10_SCSTR0/CPG_MFS10_SFLR0/CPG_MFS10_NFCR [R/W] B,H,W --------(*5)(*6)(*7)	
B488_0814	-	-/CPG_MFS10_SCSFR2/-/- [R/W] B,H,W --------(*5)	-/CPG_MFS10_SCSFR1/-/- [R/W] B,H,W --------(*5)	-/CPG_MFS10_SCSFR0/-/- [R/W] B,H,W --------(*5)	
B488_0818	-/CPG_MFS10_TBYTE3/CPG_MFS10_LAMESR/- [R/W] B,H,W --------(*5)(*6)	-/CPG_MFS10_TBYTE2/CPG_MFS10_LAMERT/- [R/W] B,H,W --------(*5)(*6)	-/CPG_MFS10_TBYTE1/CPG_MFS10_LAMIER/- [R/W] B,H,W --------(*5)(*6)	CPG_MFS10_TBYTE0/CPG_MFS10_TBYTE0/CPG_MFS10_LAMRID(LAMTID)/- [R/W] B,H,W 00000000	
B488_081C	-/-/-/CPG_MFS10_ISMK [R/W] B,H,W --------(*7)	-/-/-/CPG_MFS10_ISBA [R/W] B,H,W --------(*7)	CPG_MFS10_BGR1 [R/W] H,W 00000000	CPG_MFS10_BGR0 [R/W] H,W 00000000	
B488_0820	CPG_MFS10_FBYTE2 [R/W] B,H,W 00000000	CPG_MFS10_FBYTE1 [R/W] B,H,W 00000000	CPG_MFS10_FCR1 [R/W] B,H,W 00000100	CPG_MFS10_FCR0 [R/W] B,H,W 00000000	
B488_0824	CPG_MFS10_ESR [R/W] B,H,W 00000000	CPG_MFS10_ECR [R/W] B,H,W 00000000	CPG_MFS10_FTICR2 [R/W] B,H,W 00000000	CPG_MFS10_FTICR1 [R/W] B,H,W 00000000	
B488_0828	-			CPG_MFS10_TBSIZE [R/W] B,H,W 00000000	
B488_082C	-/-/CPG_MFS10_SSRC/CPG_MFS10_SSRC [R/W] B,H,W --------(*6)(*7)	-/-/CPG_MFS10_ESCRC/CPG_MFS10_IBSRC [R/W] B,H,W --------(*6)(*7)	-/-/CPG_MFS10_SCRC/CPG_MFS10_IBCRC [R/W] B,H,W --------(*6)(*7)	-/-/CPG_MFS10_SMRC/CPG_MFS10_SMRC [R/W] B,H,W --------(*6)(*7)	
B488_0830	-		CPG_MFS10_SACSRC [R/W] B,H,W 00000000 00000000		
B488_0834	-/-/CPG_MFS10_LAMSRC/- [R/W] B,H,W --------(*6)	-/-/CPG_MFS10_LAMCRC/- [R/W] B,H,W --------(*6)	-		
B488_0838	-/-/CPG_MFS10_LAMESRC/- [R/W] B,H,W --------(*6)	-	-/-/CPG_MFS10_LAMIERC/- [R/W] B,H,W --------(*6)	-	
B488_083C	-		CPG_MFS10_FCR1C [R/W] B,H,W 00000000	CPG_MFS10_FCR0C [R/W] B,H,W 00000000	
B488_0840	CPG_MFS10_ESRC [R/W] B,H,W 00000000	-			

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B488_0844	-/-/CPG_MFS10_SSRS/CPG_MFS10_SSRS [R/W] B,H,W --------(*6)(*7)	-/-/CPG_MFS10_ESCRS/- [R/W] B,H,W --------(*6)	-/-/CPG_MFS10_SCRS/CPG_MFS10_IBCRS [R/W] B,H,W --------(*6)(*7)	-/-/CPG_MFS10_SMRS/CPG_MFS10_SMRS [R/W] B,H,W --------(*6)(*7)	
B488_0848	-		CPG_MFS10_SACSRS [R/W] B,H,W 00000000 00000000		
B488_084C	-	-/-/CPG_MFS10_LAMCRS/- [R/W] B,H,W --------(*6)	-		
B488_0850	-		-/-/CPG_MFS10_LAMIERS/- [R/W] B,H,W --------(*6)	-	
B488_0854	-		CPG_MFS10_FCR1S [R/W] B,H,W 00000000	CPG_MFS10_FCR0S [R/W] B,H,W 00000000	
B488_0858	CPG_MFS10_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS10_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS10_RDR1(TDR1)[R/W] B,H,W	CPG_MFS10_RDR0(TDR0)[R/W] B,H,W	
B488_085C	CPG_MFS10_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS10_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS10_RDR1(TDR1)[R/W] B,H,W	CPG_MFS10_RDR0(TDR0)[R/W] B,H,W	
B488_0860	CPG_MFS10_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS10_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS10_RDR1(TDR1)[R/W] B,H,W	CPG_MFS10_RDR0(TDR0)[R/W] B,H,W	
B488_0864	CPG_MFS10_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS10_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS10_RDR1(TDR1)[R/W] B,H,W	CPG_MFS10_RDR0(TDR0)[R/W] B,H,W	
B488_0868	CPG_MFS10_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS10_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS10_RDR1(TDR1)[R/W] B,H,W	CPG_MFS10_RDR0(TDR0)[R/W] B,H,W	
B488_086C	CPG_MFS10_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS10_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS10_RDR1(TDR1)[R/W] B,H,W	CPG_MFS10_RDR0(TDR0)[R/W] B,H,W	
B488_0870	CPG_MFS10_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS10_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS10_RDR1(TDR1)[R/W] B,H,W	CPG_MFS10_RDR0(TDR0)[R/W] B,H,W	
B488_0874	CPG_MFS10_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS10_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS10_RDR1(TDR1)[R/W] B,H,W	CPG_MFS10_RDR0(TDR0)[R/W] B,H,W	
B488_0878	CPG_MFS10_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS10_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS10_RDR1(TDR1)[R/W] B,H,W	CPG_MFS10_RDR0(TDR0)[R/W] B,H,W	
B488_087C	CPG_MFS10_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS10_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS10_RDR1(TDR1)[R/W] B,H,W	CPG_MFS10_RDR0(TDR0)[R/W] B,H,W	
B488_0880	CPG_MFS10_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS10_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS10_RDR1(TDR1)[R/W] B,H,W	CPG_MFS10_RDR0(TDR0)[R/W] B,H,W	
B488_0884	CPG_MFS10_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS10_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS10_RDR1(TDR1)[R/W] B,H,W	CPG_MFS10_RDR0(TDR0)[R/W] B,H,W	
B488_0888	CPG_MFS10_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS10_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS10_RDR1(TDR1)[R/W] B,H,W	CPG_MFS10_RDR0(TDR0)[R/W] B,H,W	
B488_088C	CPG_MFS10_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS10_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS10_RDR1(TDR1)[R/W] B,H,W	CPG_MFS10_RDR0(TDR0)[R/W] B,H,W	
B488_0890	CPG_MFS10_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS10_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS10_RDR1(TDR1)[R/W] B,H,W	CPG_MFS10_RDR0(TDR0)[R/W] B,H,W	
B488_0894	CPG_MFS10_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS10_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS10_RDR1(TDR1)[R/W] B,H,W	CPG_MFS10_RDR0(TDR0)[R/W] B,H,W	
B488_0898 to B488_0BFC	-				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B488_0C00	CPG_MFS11_SSR [R/W] B,H,W 00000011	CPG_MFS11_ESCR/CPG_MFS11_ESCR/CPG_MFS11_ESCR/CPG_MFS11_IBSR [R/W] B,H,W 00000000	CPG_MFS11_SCR/CPG_MFS11_SCR/CPG_MFS11_SCR/CPG_MFS11_IBCR [R/W] B,H,W 00000000	CPG_MFS11_SMR [R/W] B,H,W 00000000	CPG_MFS ch.11
B488_0C04	-/CPG_MFS11_RDR(TDR)/-/- [R/W] B,H,W --------(*5)	-/CPG_MFS11_RDR(TDR)/-/- [R/W] B,H,W --------(*5)	CPG_MFS11_RDR(TDR)/CPG_MFS11_RDR(TDR)/-/- [R/W] B,H,W -------0	CPG_MFS11_RDR(TDR) [R/W] B,H,W 00000000	
B488_0C08	CPG_MFS11_STMR [R/W] B,H,W 00000000 00000000		CPG_MFS11_SACSR [R/W] B,H,W 00000000 00000000		
B488_0C0C	-/CPG_MFS11_SCSCR/CPG_MFS11_SFUR/- [R/W] B,H,W --------(*5)(*6)	-/CPG_MFS11_SCSCR/CPG_MFS11_SFUR/- [R/W] B,H,W --------(*5)(*6)	CPG_MFS11_STMCR [R/W] B,H,W 00000000 00000000		
B488_0C10	-/CPG_MFS11_SCSTR3/CPG_MFS11_LAMSR/- [R/W] B,H,W --------(*5)(*6)	-/CPG_MFS11_SCSTR2/CPG_MFS11_LAMCR/- [R/W] B,H,W --------(*5)(*6)	-/CPG_MFS11_SCSTR1/CPG_MFS11_SFLR1/CPG_MFS11_EIBCR [R/W] B,H,W --------(*5)(*6)(*7)	-/CPG_MFS11_SCSTR0/CPG_MFS11_SFLR0/CPG_MFS11_NFCR [R/W] B,H,W --------(*5)(*6)(*7)	
B488_0C14	-	-/CPG_MFS11_SCSFR2/-/- [R/W] B,H,W --------(*5)	-/CPG_MFS11_SCSFR1/-/- [R/W] B,H,W --------(*5)	-/CPG_MFS11_SCSFR0/-/- [R/W] B,H,W --------(*5)	
B488_0C18	-/CPG_MFS11_TBYTE3/CPG_MFS11_LAMESR/- [R/W] B,H,W --------(*5)(*6)	-/CPG_MFS11_TBYTE2/CPG_MFS11_LAMERT/- [R/W] B,H,W --------(*5)(*6)	-/CPG_MFS11_TBYTE1/CPG_MFS11_LAMIER/- [R/W] B,H,W --------(*5)(*6)	CPG_MFS11_TBYTE0/CPG_MFS11_TBYTE0/CPG_MFS11_LAMRID(LAMTID)/- [R/W] B,H,W 00000000	
B488_0C1C	-/-/-/CPG_MFS11_ISMK [R/W] B,H,W --------(*7)	-/-/-/CPG_MFS11_ISBA [R/W] B,H,W --------(*7)	CPG_MFS11_BGR1 [R/W] H,W 00000000	CPG_MFS11_BGR0 [R/W] H,W 00000000	
B488_0C20	CPG_MFS11_FBYTE2 [R/W] B,H,W 00000000	CPG_MFS11_FBYTE1 [R/W] B,H,W 00000000	CPG_MFS11_FCR1 [R/W] B,H,W 00000100	CPG_MFS11_FCR0 [R/W] B,H,W 00000000	
B488_0C24	CPG_MFS11_ESR [R/W] B,H,W 00000000	CPG_MFS11_ECR [R/W] B,H,W 00000000	CPG_MFS11_FTICR2 [R/W] B,H,W 00000000	CPG_MFS11_FTICR1 [R/W] B,H,W 00000000	
B488_0C28	-			CPG_MFS11_TBSIZE [R/W] B,H,W 00000000	
B488_0C2C	-/-/CPG_MFS11_SSRC/CPG_MFS11_SSRC [R/W] B,H,W --------(*6)(*7)	-/-/CPG_MFS11_ESCRC/CPG_MFS11_IBSRC [R/W] B,H,W --------(*6)(*7)	-/-/CPG_MFS11_SCRC/CPG_MFS11_IBCRC [R/W] B,H,W --------(*6)(*7)	-/-/CPG_MFS11_SMRC/CPG_MFS11_SMRC [R/W] B,H,W --------(*6)(*7)	
B488_0C30	-		CPG_MFS11_SACSRC [R/W] B,H,W 00000000 00000000		
B488_0C34	-/-/CPG_MFS11_LAMSRC/- [R/W] B,H,W --------(*6)	-/-/CPG_MFS11_LAMCRC/- [R/W] B,H,W --------(*6)	-		
B488_0C38	-/-/CPG_MFS11_LAMESRC/- [R/W] B,H,W --------(*6)	-	-/-/CPG_MFS11_LAMIERC/- [R/W] B,H,W --------(*6)	-	
B488_0C3C	-		CPG_MFS11_FCR1C [R/W] B,H,W 00000000	CPG_MFS11_FCR0C [R/W] B,H,W 00000000	
B488_0C40	CPG_MFS11_ESRC [R/W] B,H,W 00000000	-			

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B488_0C44	-/-/CPG_MFS11_SSRS/CPG_MFS11_SSRS [R/W] B,H,W --------(*6)(*7)	-/-/CPG_MFS11_ESCRS/- [R/W] B,H,W --------(*6)	-/-/CPG_MFS11_SCRS/CPG_MFS11_IBCRS [R/W] B,H,W --------(*6)(*7)	-/-/CPG_MFS11_SMRS/CPG_MFS11SMRS [R/W] B,H,W --------(*6)(*7)	
B488_0C48	-		CPG_MFS11_SACSRS [R/W] B,H,W 00000000 00000000		
B488_0C4C	-	-/-/CPG_MFS11_LAMCRS/- [R/W] B,H,W --------(*6)	-		
B488_0C50	-		-/-/CPG_MFS11_LAMIERS/- [R/W] B,H,W --------(*6)	-	
B488_0C54	- -		CPG_MFS11_FCR1S [R/W] B,H,W 00000000	CPG_MFS11_FCR0S [R/W] B,H,W 00000000	
B488_0C58	CPG_MFS11_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS11_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS11_RDR1(TDR1)[R/W] B,H,W	CPG_MFS11_RDR0(TDR0)[R/W] B,H,W	
B488_0C5C	CPG_MFS11_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS11_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS11_RDR1(TDR1)[R/W] B,H,W	CPG_MFS11_RDR0(TDR0)[R/W] B,H,W	
B488_0C60	CPG_MFS11_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS11_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS11_RDR1(TDR1)[R/W] B,H,W	CPG_MFS11_RDR0(TDR0)[R/W] B,H,W	
B488_0C64	CPG_MFS11_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS11_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS11_RDR1(TDR1)[R/W] B,H,W	CPG_MFS11_RDR0(TDR0)[R/W] B,H,W	
B488_0C68	CPG_MFS11_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS11_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS11_RDR1(TDR1)[R/W] B,H,W	CPG_MFS11_RDR0(TDR0)[R/W] B,H,W	
B488_0C6C	CPG_MFS11_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS11_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS11_RDR1(TDR1)[R/W] B,H,W	CPG_MFS11_RDR0(TDR0)[R/W] B,H,W	
B488_0C70	CPG_MFS11_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS11_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS11_RDR1(TDR1)[R/W] B,H,W	CPG_MFS11_RDR0(TDR0)[R/W] B,H,W	
B488_0C74	CPG_MFS11_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS11_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS11_RDR1(TDR1)[R/W] B,H,W	CPG_MFS11_RDR0(TDR0)[R/W] B,H,W	
B488_0C78	CPG_MFS11_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS11_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS11_RDR1(TDR1)[R/W] B,H,W	CPG_MFS11_RDR0(TDR0)[R/W] B,H,W	
B488_0C7C	CPG_MFS11_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS11_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS11_RDR1(TDR1)[R/W] B,H,W	CPG_MFS11_RDR0(TDR0)[R/W] B,H,W	
B488_0C80	CPG_MFS11_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS11_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS11_RDR1(TDR1)[R/W] B,H,W	CPG_MFS11_RDR0(TDR0)[R/W] B,H,W	
B488_0C84	CPG_MFS11_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS11_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS11_RDR1(TDR1)[R/W] B,H,W	CPG_MFS11_RDR0(TDR0)[R/W] B,H,W	
B488_0C88	CPG_MFS11_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS11_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS11_RDR1(TDR1)[R/W] B,H,W	CPG_MFS11_RDR0(TDR0)[R/W] B,H,W	
B488_0C8C	CPG_MFS11_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS11_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS11_RDR1(TDR1)[R/W] B,H,W	CPG_MFS11_RDR0(TDR0)[R/W] B,H,W	
B488_0C90	CPG_MFS11_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS11_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS11_RDR1(TDR1)[R/W] B,H,W	CPG_MFS11_RDR0(TDR0)[R/W] B,H,W	
B488_0C94	CPG_MFS11_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS11_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS11_RDR1(TDR1)[R/W] B,H,W	CPG_MFS11_RDR0(TDR0)[R/W] B,H,W	
B488_0C98 to B488_0FFC	-				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B488_1000	CPG_MFS12_SSR [R/W] B,H,W 00000011	CPG_MFS12_ESCR [R/W] B,H,W 00000000	CPG_MFS12_SCR [R/W] B,H,W 00000000	CPG_MFS12_SMR [R/W] B,H,W 00000000	CPG_MFS ch.12
B488_1004	-/CPG_MFS12_RDR(TDR)/- [R/W] B,H,W --------(*5)	-/CPG_MFS12_RDR(TDR)/- [R/W] B,H,W --------(*5)	CPG_MFS12_RDR(TDR)/CPG_MFS12_RDR(TDR)/- [R/W] B,H,W -------0	CPG_MFS12_RDR(TDR) [R/W] B,H,W 00000000	
B488_1008	CPG_MFS12_STMR [R/W] B,H,W 00000000 00000000		CPG_MFS12_SACSR [R/W] B,H,W 00000000 00000000		
B488_100C	-/CPG_MFS12_SCSCR/CPG_MFS12_SFUR [R/W] B,H,W --------(*5)(*6)	-/CPG_MFS12_SCSCR/CPG_MFS12_SFUR [R/W] B,H,W --------(*5)(*6)	CPG_MFS12_STMCR [R/W] B,H,W 00000000 00000000		
B488_1010	-/CPG_MFS12_SCSTR3/CPG_MFS12_LAMSR [R/W] B,H,W --------(*5)(*6)	-/CPG_MFS12_SCSTR2/CPG_MFS12_LAMCR [R/W] B,H,W --------(*5)(*6)	-/CPG_MFS12_SCSTR1/CPG_MFS12_SFLR1 [R/W] B,H,W --------(*5)(*6)	-/CPG_MFS12_SCSTR0/CPG_MFS12_SFLR0 [R/W] B,H,W --------(*5)(*6)	
B488_1014	-	-/CPG_MFS12_SCSFR2/- [R/W] B,H,W --------(*5)	-/CPG_MFS12_SCSFR1/- [R/W] B,H,W --------(*5)	-/CPG_MFS12_SCSFR0/- [R/W] B,H,W --------(*5)	
B488_1018	-/CPG_MFS12_TBYTE3/CPG_MFS12_LAMESR [R/W] B,H,W --------(*5)(*6)	-/CPG_MFS12_TBYTE2/CPG_MFS12_LAMERT [R/W] B,H,W --------(*5)(*6)	-/CPG_MFS12_TBYTE1/CPG_MFS12_LAMIER [R/W] B,H,W --------(*5)(*6)	CPG_MFS12_TBYTE0/CPG_MFS12_TBYTE0/CPG_MFS12_LAMRID(LAMTID) [R/W] B,H,W 00000000	
B488_101C	-	-	CPG_MFS12_BGR1 [R/W] H,W 00000000	CPG_MFS12_BGR0 [R/W] H,W 00000000	
B488_1020	CPG_MFS12_FBYTE2 [R/W] B,H,W 00000000	CPG_MFS12_FBYTE1 [R/W] B,H,W 00000000	CPG_MFS12_FCR1 [R/W] B,H,W 00000100	CPG_MFS12_FCR0 [R/W] B,H,W 00000000	
B488_1024	CPG_MFS12_ESR [R/W] B,H,W 00000000	CPG_MFS12_ECR [R/W] B,H,W 00000000	CPG_MFS12_FTICR2 [R/W] B,H,W 00000000	CPG_MFS12_FTICR1 [R/W] B,H,W 00000000	
B488_1028	-			CPG_MFS12_TBSIZE [R/W] B,H,W 00000000	
B488_102C	-/-/CPG_MFS12_SSRC [R/W] B,H,W --------(*6)	-/-/CPG_MFS12_ESCRC [R/W] B,H,W --------(*6)	-/-/CPG_MFS12_SCRC [R/W] B,H,W --------(*6)	-/-/CPG_MFS12_SMRC [R/W] B,H,W --------(*6)	
B488_1030	-		CPG_MFS12_SACSRC [R/W] B,H,W 00000000 00000000		
B488_1034	-/-/CPG_MFS12_LAMSRC [R/W] B,H,W --------(*6)	-/-/CPG_MFS12_LAMCRC [R/W] B,H,W --------(*6)	-		
B488_1038	-/-/CPG_MFS12_LAMESRC [R/W] B,H,W --------(*6)	-	-/-/CPG_MFS12_LAMIERC [R/W] B,H,W --------(*6)	-	
B488_103C	-		CPG_MFS12_FCR1C [R/W] B,H,W 00000000	CPG_MFS12_FCR0C [R/W] B,H,W 00000000	
B488_1040	CPG_MFS12_ESRC [R/W] B,H,W 00000000	-			

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B488_1044	-/-/CPG_MFS12_SSRS [R/W] B,H,W --------(*6)	-/-/CPG_MFS12_ESCRS [R/W] B,H,W --------(*6)	-/-/CPG_MFS12_SCRS [R/W] B,H,W --------(*6)	-/-/CPG_MFS12_SMRS [R/W] B,H,W --------(*6)	
B488_1048	-		CPG_MFS12_SACSRS [R/W] B,H,W 00000000 00000000		
B488_104C	-	-/-/CPG_MFS12_LAMCRS [R/W] B,H,W --------(*6)	-		
B488_1050	-		-/-/CPG_MFS12_LAMIERS [R/W] B,H,W --------(*6)	-	
B488_1054			CPG_MFS12_FCR1S [R/W] B,H,W 00000000	CPG_MFS12_FCR0S [R/W] B,H,W 00000000	
B488_1058	CPG_MFS12_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS12_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS12_RDR1(TDR1)[R/W] B,H,W	CPG_MFS12_RDR0(TDR0)[R/W] B,H,W	
B488_105C	CPG_MFS12_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS12_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS12_RDR1(TDR1)[R/W] B,H,W	CPG_MFS12_RDR0(TDR0)[R/W] B,H,W	
B488_1060	CPG_MFS12_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS12_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS12_RDR1(TDR1)[R/W] B,H,W	CPG_MFS12_RDR0(TDR0)[R/W] B,H,W	
B488_1064	CPG_MFS12_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS12_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS12_RDR1(TDR1)[R/W] B,H,W	CPG_MFS12_RDR0(TDR0)[R/W] B,H,W	
B488_1068	CPG_MFS12_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS12_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS12_RDR1(TDR1)[R/W] B,H,W	CPG_MFS12_RDR0(TDR0)[R/W] B,H,W	
B488_106C	CPG_MFS12_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS12_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS12_RDR1(TDR1)[R/W] B,H,W	CPG_MFS12_RDR0(TDR0)[R/W] B,H,W	
B488_1070	CPG_MFS12_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS12_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS12_RDR1(TDR1)[R/W] B,H,W	CPG_MFS12_RDR0(TDR0)[R/W] B,H,W	
B488_1074	CPG_MFS12_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS12_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS12_RDR1(TDR1)[R/W] B,H,W	CPG_MFS12_RDR0(TDR0)[R/W] B,H,W	
B488_1078	CPG_MFS12_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS12_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS12_RDR1(TDR1)[R/W] B,H,W	CPG_MFS12_RDR0(TDR0)[R/W] B,H,W	
B488_107C	CPG_MFS12_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS12_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS12_RDR1(TDR1)[R/W] B,H,W	CPG_MFS12_RDR0(TDR0)[R/W] B,H,W	
B488_1080	CPG_MFS12_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS12_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS12_RDR1(TDR1)[R/W] B,H,W	CPG_MFS12_RDR0(TDR0)[R/W] B,H,W	
B488_1084	CPG_MFS12_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS12_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS12_RDR1(TDR1)[R/W] B,H,W	CPG_MFS12_RDR0(TDR0)[R/W] B,H,W	
B488_1088	CPG_MFS12_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS12_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS12_RDR1(TDR1)[R/W] B,H,W	CPG_MFS12_RDR0(TDR0)[R/W] B,H,W	
B488_108C	CPG_MFS12_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS12_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS12_RDR1(TDR1)[R/W] B,H,W	CPG_MFS12_RDR0(TDR0)[R/W] B,H,W	
B488_1090	CPG_MFS12_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS12_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS12_RDR1(TDR1)[R/W] B,H,W	CPG_MFS12_RDR0(TDR0)[R/W] B,H,W	
B488_1094	CPG_MFS12_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS12_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS12_RDR1(TDR1)[R/W] B,H,W	CPG_MFS12_RDR0(TDR0)[R/W] B,H,W	
B488_1098 to B488_13FC	-				
B488_1400 to B488_FFFF	-				Reserved

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B489_0000	RLT16_DMACFG [R/W] B,H,W 00000000 00000000 00000000 00000000				Reload Timer ch.16
B489_0004	-				
B489_0008	RLT16_TMCSR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B489_000C	-				
B489_0010	RLT16_TMRLR [R/W] W,DW XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
B489_0014	RLT16_TMR [R/W] W,DW XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
B489_0018 to B489_03FC	-				
B489_0400	RLT17_DMACFG [R/W] B,H,W 00000000 00000000 00000000 00000000				Reload Timer ch.17
B489_0404	-				
B489_0408	RLT17_TMCSR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B489_040C	-				
B489_0410	RLT17_TMRLR [R/W] W,DW XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
B489_0414	RLT17_TMR [R/W] W,DW XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
B489_0418 to B489_07FC	-				
B489_0800	RLT18_DMACFG [R/W] B,H,W 00000000 00000000 00000000 00000000				Reload Timer ch.18
B489_0804	-				
B489_0808	RLT18_TMCSR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B489_080C	-				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B489_0810	RLT18_TMRLR [R/W] W,DW XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
B489_0814	RLT18_TMR [R/W] W,DW XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
B489_0818 to B489_0BFC	-				
B489_0C00	RLT19_DMACFG [R/W] B,H,W 00000000 00000000 00000000 00000000				Reload Timer ch.19
B489_0C04	-				
B489_0C08	RLT19_TMCSR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B489_0C0C	-				
B489_0C10	RLT19_TMRLR [R/W] W,DW XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
B489_0C14	RLT19_TMR [R/W] W,DW XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
B489_0C18 to B489_0FFC	-				
B489_1000 to B489_3FFC	-				Reserved
B489_4000 to B489_7FFC	-				Reserved
B489_8000	QC08_QRCR [R/W] H,W 00000000 00000000		QC08_QPCR [R/W] H,W 00000000 00000000		QPRC ch.8
B489_8004	QC08_QPRCR [R/W] H,W 00000000 00000000		QC08_QPCCR [R/W] H,W 00000000 00000000		
B489_8008	QC08_QICRH [R/W] B,H,W 00000000	QC08_QICRL [R/W] B,H,W 00000000	QC08_QMPR [R/W] H,W 11111111 11111111		
B489_800C	QC08_QECR [R/W] B,H,W 00000000 00000000		QC08_QCR [R/W] B,H,W 00000000 00000000		
B489_8010	-				
B489_8014	-				
B489_8018	-				
B489_801C	-				
B489_8020 to B489_83FC	-				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B489_8400	QC09_QRCR [R/W] H,W 00000000 00000000		QC09_QPCR [R/W] H,W 00000000 00000000		QPRC ch.9
B489_8404	QC09_QPRCR [R/W] H,W 00000000 00000000		QC09_QPCCR [R/W] H,W 00000000 00000000		
B489_8408	QC09_QICRH [R/W] B,H,W 00000000	QC09_QICRL [R/W] B,H,W 00000000	QC09_QMPR [R/W] H,W 11111111 11111111		
B489_840C	QC09_QECR [R/W] B,H,W 00000000 00000000		QC09_QCR [R/W] B,H,W 00000000 00000000		
B489_8410					
B489_8414					
B489_8418					
B489_841C					
B489_8420 to B489_87FC					
B489_8800 to B489_FFFC	-				
B48A_0000 to B48A_FFFC	-				Reserved
B48B_0000 to B48B_0FFC	-				Reserved
B48B_1000 to B48B_FBFC	-				Reserved
B48B_FC00	CPG1_RLT_TSEL [R/W] B,H,W 00000000 00000000 00000000 00000000				CPERI #1 Misc_Registers
B48B_FC04	CPG1_RLT_SSSR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B48B_FC08 to B48B_FFFC	-				Reserved
B48C_0000 to B48C_3FFC	-				Reserved
B48C_4000	SMC0_PWC1 [R/W] B,H,W [R/W] B,H,W ------XX XXXXXXXX		SMC0_PWC [R/W] B,H,W ------0- -00000--		SMC ch.0
B48C_4004	SMC0_PWS [R/W] B,H,W -0000000 --000000		SMC0_PWC2 [R/W] B,H,W ------XX XXXXXXXX		
B48C_4008	SMC0_PTRGDL [R/W] B,H,W [R/W] B,H,W -------- 00000000		SMC0_PWSS [R/W] B,H,W -0------ --------		

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B48C_400C	-		-		
B48C_4010 to B48C_43FC	-				
B48C_4400	SMC1_PWC1 [R/W] B,H,W ------XX XXXXXXXX		SMC1_PWC [R/W] B,H,W ------0- -00000--		SMC ch.1
B48C_4404	SMC1_PWS [R/W] B,H,W -0000000 --000000		SMC1_PWC2 [R/W] B,H,W ------XX XXXXXXXX		
B48C_4408	SMC1_PTRGDL [R/W] B,H,W -------- 00000000		SMC1_PWSS [R/W] B,H,W -0------ --------		
B48C_440C	-		-		
B48C_4410 to B48C_47FC	-				
B48C_4800	SMC2_PWC1 [R/W] B,H,W ------XX XXXXXXXX		SMC2_PWC [R/W] B,H,W ------0- -00000--		SMC ch.2
B48C_4804	SMC2_PWS [R/W] B,H,W -0000000 --000000		SMC2_PWC2 [R/W] B,H,W ------XX XXXXXXXX		
B48C_4808	SMC2_PTRGDL [R/W] B,H,W -------- 00000000		SMC2_PWSS [R/W] B,H,W -0------ --------		
B48C_480C	-		-		
B48C_4810 to B48C_4BFC	-				
B48C_4C00	SMC3_PWC1 [R/W] B,H,W ------XX XXXXXXXX		SMC3_PWC [R/W] B,H,W ------0- -00000--		SMC ch.3
B48C_4C04	SMC3_PWS [R/W] B,H,W -0000000 --000000		SMC3_PWC2 [R/W] B,H,W ------XX XXXXXXXX		
B48C_4C08	SMC3_PTRGDL [R/W] B,H,W -------- 00000000		SMC3_PWSS [R/W] B,H,W -0------ --------		
B48C_4C0C	-		-		
B48C_4C10 to B48C_4FFC	-				
B48C_5000 to B48C_57FC	-				Reserved
B48C_5800	SMCTGg_PTRG [R/W] B,H,W -------- ------00		SMCTGg_PTRGS [R/W] B,H,W --000000 --000000		SMCTRG0
B48C_5804 to B48C_5BFC	-				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B48C_5C00	LCDCMR [R/W] B,H,W 00000000	LCRS [R/W] B,H,W 00000000	LCR0 [R/W] B,H,W 00010000	LCR1 [R/W] B,H,W 00000000	LCDC
B48C_5C04	VRAM0 [R/W] B,H,W 00000000	VRAM1 [R/W] B,H,W 00000000	VRAM2 [R/W] B,H,W 00000000	VRAM3 [R/W] B,H,W 00000000	
B48C_5C08	VRAM4 [R/W] B,H,W 00000000	VRAM5 [R/W] B,H,W 00000000	VRAM6 [R/W] B,H,W 00000000	VRAM7 [R/W] B,H,W 00000000	
B48C_5C0C	VRAM8 [R/W] B,H,W 00000000	VRAM9 [R/W] B,H,W 00000000	VRAM10 [R/W] B,H,W 00000000	VRAM11 [R/W] B,H,W 00000000	
B48C_5C10	VRAM12 [R/W] B,H,W 00000000	VRAM13 [R/W] B,H,W 00000000	VRAM14 [R/W] B,H,W 00000000	VRAM15 [R/W] B,H,W 00000000	
B48C_5C14	LDR [R/W] B,H,W -------0 00000000		-		
B48C_5C18 to B48C_5FFC	-				
B48C_6000	LCD_KEYCDR [R/W] W 0000---- -------- ----0000 00000000				LCDE
B48C_6004	SEGER [R/W] B,H,W 00000000 00000000 10000000 11111111				
B48C_6008	COMVER [R/W] B,H,W 11111111 11111111 11111111 11111111				
B48C_600C to B48C_63FC	-				
B48C_6400 to B48C_FFFC	-				Reserved
B48D_0000 to B48F_FFFC	-				Reserved

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B490_0000	CPG_CANFD0_CREL [R] B,H,W 00110010 00000100 00010010 00011000				CAN_FD ch.0
B490_0004	CPG_CANFD0_ENDN [R] B,H,W 10000111 01100101 01000011 00100001				
B490_0008	-				
B490_000C	CPG_CANFD0_DBTP [R/W] B,H,W 00000000 00000000 00001010 00110011				
B490_0010	CPG_CANFD0_TEST [R/W] B,H,W 00000000 00000000 00000000 *0000000				
B490_0014	CPG_CANFD0_RWD [R/W] B,H,W 00000000 00000000 00000000 00000000				
B490_0018	CPG_CANFD0_CCCR [R/W] B,H,W 00000000 00000000 00000000 00000001				
B490_001C	CPG_CANFD0_NBTP [R/W] B,H,W 00001100 00000000 00001010 00000011				
B490_0020	CPG_CANFD0_TSCC [R/W] B,H,W 00000000 00000000 00000000 00000000				
B490_0024	CPG_CANFD0_TSCV [R/W] B,H,W 00000000 00000000 00000000 00000000				
B490_0028	CPG_CANFD0_TOCC [R/W] B,H,W 11111111 11111111 00000000 00000000				
B490_002C	CPG_CANFD0_TOCV [R/W] B,H,W 00000000 00000000 11111111 11111111				
B490_0030 to B490_003C	-				
B490_0040	CPG_CANFD0_ECR [R] B,H,W XXXXXXXX 00000000 00000000 00000000				
B490_0044	CPG_CANFD0_PSR [R] H,W XXXXXXXX XXXXXXXX X0000111 00000111				
B490_0048	CPG_CANFD0_TDCR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B490_004C	-				
B490_0050	CPG_CANFD0_IR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B490_0054	CPG_CANFD0_IE [R/W] B,H,W 00000000 00000000 00000000 00000000				
B490_0058	CPG_CANFD0_ILS [R/W] B,H,W 00000000 00000000 00000000 00000000				
B490_005C	CPG_CANFD0_ILE [R/W] B,H,W 00000000 00000000 00000000 00000000				
B490_0060 to B490_007C	-				
B490_0080	CPG_CANFD0_GFC [R/W] B,H,W 00000000 00000000 00000000 00000000				
B490_0084	CPG_CANFD0_SIDFC [R/W] B,H,W 00000000 00000000 00000000 00000000				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B490_0088	CPG_CANFD0_XIDFC [R/W] B,H,W 00000000 00000000 00000000 00000000				
B490_008C	-				
B490_0090	CPG_CANFD0_XIDAM [R/W] B,H,W 00011111 11111111 11111111 11111111				
B490_0094	CPG_CANFD0_HPMS [R] B,H,W XXXXXXXX XXXXXXXX 00000000 00000000				
B490_0098	CPG_CANFD0_NDAT1 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B490_009C	CPG_CANFD0_NDAT2 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B490_00A0	CPG_CANFD0_RXF0C [R/W] B,H,W 00000000 00000000 00000000 00000000				
B490_00A4	CPG_CANFD0_RXF0S [R] B,H,W XXXXXX00 XX000000 XX000000 X0000000				
B490_00A8	CPG_CANFD0_RXF0A [R/W] B,H,W 00000000 00000000 00000000 00000000				
B490_00AC	CPG_CANFD0_RXBC [R/W] B,H,W 00000000 00000000 00000000 00000000				
B490_00B0	CPG_CANFD0_RXF1C [R/W] B,H,W 00000000 00000000 00000000 00000000				
B490_00B4	CPG_CANFD0_RXF1S [R] B,H,W 00XXXX00 XX000000 XX000000 X0000000				
B490_00B8	CPG_CANFD0_RXF1A [R/W] B,H,W 00000000 00000000 00000000 00000000				
B490_00BC	CPG_CANFD0_RXESC [R/W] B,H,W 00000000 00000000 00000000 00000000				
B490_00C0	CPG_CANFD0_TXBC [R/W] B,H,W 00000000 00000000 00000000 00000000				
B490_00C4	CPG_CANFD0_TXFQS [R] B,H,W XXXXXXXX XX000000 XXX00000 XX000000				
B490_00C8	CPG_CANFD0_TXESC [R/W] B,H,W 00000000 00000000 00000000 00000000				
B490_00CC	CPG_CANFD0_TXBRP [R] B,H,W 00000000 00000000 00000000 00000000				
B490_00D0	CPG_CANFD0_TXBAR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B490_00D4	CPG_CANFD0_TXBCR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B490_00D8	CPG_CANFD0_TXBTO [R] B,H,W 00000000 00000000 00000000 00000000				
B490_00DC	CPG_CANFD0_TXBCF [R] B,H,W 00000000 00000000 00000000 00000000				
B490_00E0	CPG_CANFD0_TXBTIE [R/W] B,H,W 00000000 00000000 00000000 00000000				
B490_00E4	CPG_CANFD0_TXBCIE [R/W] B,H,W 00000000 00000000 00000000 00000000				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B490_00E8 to B490_00EC	-				
B490_00F0	CPG_CANFD0_TXEFC [R/W] B,H,W 00000000 00000000 00000000 00000000				
B490_00F4	CPG_CANFD0_TXEFS [R] B,H,W XXXXXX00 XXX00000 XXX00000 XX000000				
B490_00F8	CPG_CANFD0_TXEFA [R/W] B,H,W 00000000 00000000 00000000 00000000				
B490_00FC to B490_01FC	-				
B490_0200	CPG_CANFD0_FDSEAR [R] B,H,W 00000000 00000000		CPG_CANFD0_FDESR [R] B,H,W 00000000	CPG_CANFD0_FDECR [R/W] B,H,W 00000000	
B490_0204	CPG_CANFD0_FDDEAR [R] B,H,W 00000000 00000000		CPG_CANFD0_FDESCR [W] B,H,W 00000000	-	
B490_0208 to B490_02FC	-				
B490_0300 to B490_7FFC	-				Reserved
B490_8000 to B490_BFFC	CAN_RAM ch.0				CAN RAM ch.0
B490_C000 to B490_FFFC	-				Reserved
B491_0000	CPG_CANFD1_CREL [R] B,H,W 00110010 00000100 00010010 00011000				CAN_FD ch.1
B491_0004	CPG_CANFD1_ENDN [R] B,H,W 10000111 01100101 01000011 00100001				
B491_0008	-				
B491_000C	CPG_CANFD1_DBTP [R/W] B,H,W 00000000 00000000 00001010 00110011				
B491_0010	CPG_CANFD1_TEST [R/W] B,H,W 00000000 00000000 00000000 *0000000				
B491_0014	CPG_CANFD1_RWD [R/W] B,H,W 00000000 00000000 00000000 00000000				
B491_0018	CPG_CANFD1_CCCR [R/W] B,H,W 00000000 00000000 00000000 00000001				
B491_001C	CPG_CANFD1_NBTP [R/W] B,H,W 00001100 00000000 00001010 00000011				
B491_0020	CPG_CANFD1_TSCC [R/W] B,H,W 00000000 00000000 00000000 00000000				
B491_0024	CPG_CANFD1_TSCV [R/W] B,H,W 00000000 00000000 00000000 00000000				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B491_0028	CPG_CANFD1_TOCC [R/W] B,H,W 11111111 11111111 00000000 00000000				
B491_002C	CPG_CANFD1_TOCV [R/W] B,H,W 00000000 00000000 11111111 11111111				
B491_0030 to B491_003C	-				
B491_0040	CPG_CANFD1_ECR [R] B,H,W XXXXXXXX 00000000 00000000 00000000				
B491_0044	CPG_CANFD1_PSR [R] H,W XXXXXXXX XXXXXXXX X0000111 00000111				
B491_0048	CPG_CANFD1_TDCR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B491_004C	-				
B491_0050	CPG_CANFD1_IR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B491_0054	CPG_CANFD1_IE [R/W] B,H,W 00000000 00000000 00000000 00000000				
B491_0058	CPG_CANFD1_ILS [R/W] B,H,W 00000000 00000000 00000000 00000000				
B491_005C	CPG_CANFD1_ILE [R/W] B,H,W 00000000 00000000 00000000 00000000				
B491_0060 to B491_007C	-				
B491_0080	CPG_CANFD1_GFC [R/W] B,H,W 00000000 00000000 00000000 00000000				
B491_0084	CPG_CANFD1_SIDFC [R/W] B,H,W 00000000 00000000 00000000 00000000				
B491_0088	CPG_CANFD1_XIDFC [R/W] B,H,W 00000000 00000000 00000000 00000000				
B491_008C	-				
B491_0090	CPG_CANFD1_XIDAM [R/W] B,H,W 00011111 11111111 11111111 11111111				
B491_0094	CPG_CANFD1_HPMS [R] B,H,W XXXXXXXX XXXXXXXX 00000000 00000000				
B491_0098	CPG_CANFD1_NDAT1 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B491_009C	CPG_CANFD1_NDAT2 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B491_00A0	CPG_CANFD1_RXF0C [R/W] B,H,W 00000000 00000000 00000000 00000000				
B491_00A4	CPG_CANFD1_RXF0S [R] B,H,W XXXXXX00 XX000000 XX000000 X0000000				
B491_00A8	CPG_CANFD1_RXF0A [R/W] B,H,W 00000000 00000000 00000000 00000000				
B491_00AC	CPG_CANFD1_RXBC [R/W] B,H,W 00000000 00000000 00000000 00000000				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B491_00B0	CPG_CANFD1_RXF1C [R/W] B,H,W 00000000 00000000 00000000 00000000				
B491_00B4	CPG_CANFD1_RXF1S [R] B,H,W 00XXXX00 XX000000 XX000000 X0000000				
B491_00B8	CPG_CANFD1_RXF1A [R/W] B,H,W 00000000 00000000 00000000 00000000				
B491_00BC	CPG_CANFD1_RXESC [R/W] B,H,W 00000000 00000000 00000000 00000000				
B491_00C0	CPG_CANFD1_TXBC [R/W] B,H,W 00000000 00000000 00000000 00000000				
B491_00C4	CPG_CANFD1_TXFQS [R] B,H,W XXXXXXXX XX000000 XXX00000 XX000000				
B491_00C8	CPG_CANFD1_TXESC [R/W] B,H,W 00000000 00000000 00000000 00000000				
B491_00CC	CPG_CANFD1_TXBRP [R] B,H,W 00000000 00000000 00000000 00000000				
B491_00D0	CPG_CANFD1_TXBAR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B491_00D4	CPG_CANFD1_TXBCR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B491_00D8	CPG_CANFD1_TXBTO [R] B,H,W 00000000 00000000 00000000 00000000				
B491_00DC	CPG_CANFD1_TXBCF [R] B,H,W 00000000 00000000 00000000 00000000				
B491_00E0	CPG_CANFD1_TXBTIE [R/W] B,H,W 00000000 00000000 00000000 00000000				
B491_00E4	CPG_CANFD1_TXBCIE [R/W] B,H,W 00000000 00000000 00000000 00000000				
B491_00E8 to B491_00EC	-				
B491_00F0	CPG_CANFD1_TXEFC [R/W] B,H,W 00000000 00000000 00000000 00000000				
B491_00F4	CPG_CANFD1_TXEFS [R] B,H,W XXXXXX00 XXX00000 XXX00000 XX000000				
B491_00F8	CPG_CANFD1_TXEFA [R/W] B,H,W 00000000 00000000 00000000 00000000				
B491_00FC to B491_01FC	-				
B491_0200	CPG_CANFD1_FDSEAR [R] B,H,W 00000000 00000000		CPG_CANFD1_FDESR [R] B,H,W 00000000	CPG_CANFD1_FDECR [R/W] B,H,W 00000000	
B491_0204	CPG_CANFD1_FDDEAR [R] B,H,W 00000000 00000000		CPG_CANFD1_FDESCR [W] B,H,W 00000000	-	
B491_0208 to B491_02FC	-				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B491_0300 to B491_7FFC	-				Reserved
B491_8000 to B491_BFFC	CAN_RAM ch.1				CAN RAM ch.1
B491_C000 to B491_FFFC	-				Reserved
B492_0000	CPG_CANFD2_CREL [R] B,H,W 00110010 00000100 00010010 00011000				CAN_FD ch.2
B492_0004	CPG_CANFD2_ENDN [R] B,H,W 10000111 01100101 01000011 00100001				
B492_0008	-				
B492_000C	CPG_CANFD2_DBTP [R/W] B,H,W 00000000 00000000 00001010 00110011				
B492_0010	CPG_CANFD2_TEST [R/W] B,H,W 00000000 00000000 00000000 *0000000				
B492_0014	CPG_CANFD2_RWD [R/W] B,H,W 00000000 00000000 00000000 00000000				
B492_0018	CPG_CANFD2_CCCR [R/W] B,H,W 00000000 00000000 00000000 00000001				
B492_001C	CPG_CANFD2_ NBTP [R/W] B,H,W 00001100 00000000 00001010 00000011				
B492_0020	CPG_CANFD2_TSCC [R/W] B,H,W 00000000 00000000 00000000 00000000				
B492_0024	CPG_CANFD2_TSCV [R/W] B,H,W 00000000 00000000 00000000 00000000				
B492_0028	CPG_CANFD2_TOCC [R/W] B,H,W 11111111 11111111 00000000 00000000				
B492_002C	CPG_CANFD2_TOCV [R/W] B,H,W 00000000 00000000 11111111 11111111				
B492_0030 to B492_003C	-				
B492_0040	CPG_CANFD2_ECR [R] B,H,W XXXXXXXX 00000000 00000000 00000000				
B492_0044	CPG_CANFD2_PSR [R] H,W XXXXXXXX XXXXXXXX X0000111 00000111				
B492_0048	CPG_CANFD2_TDCR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B492_004C	-				
B492_0050	CPG_CANFD2_IR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B492_0054	CPG_CANFD2_IE [R/W] B,H,W 00000000 00000000 00000000 00000000				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B492_0058	CPG_CANFD2_ILS [R/W] B,H,W 00000000 00000000 00000000 00000000				
B492_005C	CPG_CANFD2_ILE [R/W] B,H,W 00000000 00000000 00000000 00000000				
B492_0060 to B492_007C	-				
B492_0080	CPG_CANFD2_GFC [R/W] B,H,W 00000000 00000000 00000000 00000000				
B492_0084	CPG_CANFD2_SIDFC [R/W] B,H,W 00000000 00000000 00000000 00000000				
B492_0088	CPG_CANFD2_XIDFC [R/W] B,H,W 00000000 00000000 00000000 00000000				
B492_008C	-				
B492_0090	CPG_CANFD2_XIDAM [R/W] B,H,W 00011111 11111111 11111111 11111111				
B492_0094	CPG_CANFD2_HPMS [R] B,H,W XXXXXXXX XXXXXXXX 00000000 00000000				
B492_0098	CPG_CANFD2_NDAT1 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B492_009C	CPG_CANFD2_NDAT2 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B492_00A0	CPG_CANFD2_RXF0C [R/W] B,H,W 00000000 00000000 00000000 00000000				
B492_00A4	CPG_CANFD2_RXF0S [R] B,H,W XXXXXX00 XX000000 XX000000 X0000000				
B492_00A8	CPG_CANFD2_RXF0A [R/W] B,H,W 00000000 00000000 00000000 00000000				
B492_00AC	CPG_CANFD2_RXBC [R/W] B,H,W 00000000 00000000 00000000 00000000				
B492_00B0	CPG_CANFD2_RXF1C [R/W] B,H,W 00000000 00000000 00000000 00000000				
B492_00B4	CPG_CANFD2_RXF1S [R] B,H,W 00XXXX00 XX000000 XX000000 X0000000				
B492_00B8	CPG_CANFD2_RXF1A [R/W] B,H,W 00000000 00000000 00000000 00000000				
B492_00BC	CPG_CANFD2_RXESC [R/W] B,H,W 00000000 00000000 00000000 00000000				
B492_00C0	CPG_CANFD2_TXBC [R/W] B,H,W 00000000 00000000 00000000 00000000				
B492_00C4	CPG_CANFD2_TXFQS [R] B,H,W XXXXXXXX XX000000 XXX00000 XX000000				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B492_00C8	CPG_CANFD2_TXESC [R/W] B,H,W 00000000 00000000 00000000 00000000				
B492_00CC	CPG_CANFD2_TXBRP [R] B,H,W 00000000 00000000 00000000 00000000				
B492_00D0	CPG_CANFD2_TXBAR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B492_00D4	CPG_CANFD2_TXBCR [R/W] B,H,W 00000000 00000000 00000000 00000000				
B492_00D8	CPG_CANFD2_TXBTO [R] B,H,W 00000000 00000000 00000000 00000000				
B492_00DC	CPG_CANFD2_TXBCF [R] B,H,W 00000000 00000000 00000000 00000000				
B492_00E0	CPG_CANFD2_TXBTIE [R/W] B,H,W 00000000 00000000 00000000 00000000				
B492_00E4	CPG_CANFD2_TXBCIE [R/W] B,H,W 00000000 00000000 00000000 00000000				
B492_00E8 to B492_00EC	-				
B492_00F0	CPG_CANFD2_TXEFC [R/W] B,H,W 00000000 00000000 00000000 00000000				
B492_00F4	CPG_CANFD2_TXEFS [R] B,H,W XXXXXX00 XXX00000 XXX00000 XX000000				
B492_00F8	CPG_CANFD2_TXEFA [R/W] B,H,W 00000000 00000000 00000000 00000000				
B492_00FC to B492_01FC	-				
B492_0200	CPG_CANFD2_FDSEAR [R] B,H,W 00000000 00000000		CPG_CANFD2_FDESR [R] B,H,W 00000000	CPG_CANFD2_FDECR [R/W] B,H,W 00000000	
B492_0204	CPG_CANFD2_FDDEAR [R] B,H,W 00000000 00000000		CPG_CANFD2_FDESCR [W] B,H,W 00000000	-	
B492_0208 to B492_02FC	-				
B492_0300 to B492_7FFC	-				Reserved

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B492_8000 to B492_BFFC	CAN RAM ch.2				CAN RAM ch.2
B492_C000 to B492_FFFC	-				Reserved
B493_0000 to B4BF_FFFC	-				Reserved
B4C0_0000 to B4FF_FFFC	-				BIT RMW alias for CPERI#0 (Covers B490_0000 - B497_FFFF)
B500_0000 to B5FF_FFFC	-				Reserved
B600_0000 to B6FF_FFFC					Reserved
B700_0000 to B77F_FFFC	-				BIT RMW alias for CPERI#2 (Covers B470_0000 - B47F_FFFF)
B780_0000 to B7BF_FFFC	-				BIT RMW alias for CPERI#0 (Covers B480_0000 - B487_FFFF)
B7C0_0000 to B7FF_FFFC	-				BIT RMW alias for CPERI#1 (Covers B488_0000 - B48F_FFFF)

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B800_0000 to FFFE_DFFC	-				Reserved
FFFE_E000 to FFFE_E3F4	-				IRC
FFFE_E3F8	-				
FFFE_E3FC	IRC0_NMIVASBR [R/W] B,H,W 00000000 00000000 00000000 00000000				
FFFE_E400 to FFFE_F7FC	-				
FFFE_F800 to FFFE_FBF4	-				
FFFE_FBF8	-				
FFFE_FBFC	IRC_NMIVASBR [R/W] B,H,W 00000000 00000000 00000000 00000000				
FFFE_FC00 to FFFE_FF54					BOOT ROM HW I/F
FFFE_FF58	EXCFG_UNLOCK [R/W] W 00000000 00000000 00000000 00000000				
FFFE_FF5C	-				
FFFE_FF60	EXCFG_CNFG [R/W] B,H,W 00000000 00000000 00000000 00000001				
FFFE_FF64 to FFFE_FF80	-				
FFFE_FF84	EXCFG_UNDEFINACT [R/W] B,H,W 11111111 11111111 00000000 00100100				
FFFE_FF88	EXCFG_SVCINACT [R/W] B,H,W 11111111 11111111 00000000 00101000				
FFFE_FF8C	EXCFG_PABORTINACT [R/W] B,H,W 11111111 11111111 00000000 00101100				
FFFE_FF90	EXCFG_DABORTINACT [R/W] B,H,W 11111111 11111111 00000000 00110000				
FFFE_FF94 to FFFE_FFC0	-				
FFFE_FFC4	EXCFG_UNDEFACT [R/W] B,H,W 11111111 11111111 00000000 00100100				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
FFFE_FFC8	EXCFG_SVCACT [R/W] B,H,W 11111111 11111111 00000000 00101000				
FFFE_FFCC	EXCFG_PABORTACT [R/W] B,H,W 11111111 11111111 00000000 00101100				
FFFE_FFD0	EXCFG_DABORTACT [R/W] B,H,W 11111111 11111111 00000000 00110000				
FFFE_FFD4	-				
FFFE_FFD8	-				
FFFE_FFDC	-				

*1: The access size depends on the value of FWIDTH 17:16 of DDRHSSPIn_DMFIFOCFGS.

*2: An initial value depends on a set value. It sets it on the kind side.

*3: 00000111_11111111 8KB

*4: An initial value is XXXXXXXX_XXXXXXXX only at the reload timer.

*5: Because it is not CSIO mode, it reserves immediately after reset.

*6: Because it is not LIN mode, it reserves immediately after reset.

*7:Because it is not I^2C mode, it reserves immediately after reset.

*9: An initial value is 00000000 00000000 00000000 00000000 only at the PWM.

*10: An initial value is 00000000 00000000 only at the PPG.

*11:see 1.4.1.1 in "HWM of SHE"

*12:see "HWM of Lowpower"

*13:see "HWM of Hardware WatchdogTimer"

3. APPS#5

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B484_0000	-	SGDER0 [R/W] B,H,W 00000000	SGCR0 [R/W] B,H,W -0000-0- 00000000		SG ch.0
B484_0004	SGAR0 [R/W] B,H,W 00000000 00000000		SGFR0 [R/W] B,H,W 00000000	SGNR0 [R/W] B,H,W 00000000	
B484_0008	SGTCR0 [R/W] B,H,W 00000000	SGIDR0 [R/W] B,H,W 00000000	SGPCR0 [R/W] B,H,W 00000000 11111111		
B484_000C	SGDMAR0 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B484_0010	-		SGCCR0 [R/W] H,W -------- ------0-		
B484_0014 to B484_03FC	-				
B484_0400	-	SGDER1 [R/W] B,H,W 00000000	SGCR1 [R/W] B,H,W -0000-0- 00000000		SG ch.1
B484_0404	SGAR1 [R/W] B,H,W 00000000 00000000		SGFR1 [R/W] B,H,W 00000000	SGNR1 [R/W] B,H,W 00000000	
B484_0408	SGTCR1 [R/W] B,H,W 00000000	SGIDR1 [R/W] B,H,W 00000000	SGPCR1 [R/W] B,H,W 00000000 11111111		
B484_040C	SGDMAR1 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B484_0410	-		SGCCR1 [R/W] H,W -------- ------0-		
B484_0414 to B484_07FC	-				
B484_0800	-	SGDER2 [R/W] B,H,W 00000000	SGCR2 [R/W] B,H,W -0000-0- 00000000		SG ch.2
B484_0804	SGAR2 [R/W] B,H,W 00000000 00000000		SGFR2 [R/W] B,H,W 00000000	SGNR2 [R/W] B,H,W 00000000	
B484_0808	SGTCR2 [R/W] B,H,W 00000000	SGIDR2 [R/W] B,H,W 00000000	SGPCR2 [R/W] B,H,W 00000000 11111111		
B484_080C	SGDMAR2 [R/W] B,H,W 00000000 00000000 00000000 00000000				
B484_0810	-		SGCCR2 [R/W] H,W -------- ------0-		
B484_0814 to B484_0BFC	-				
B484_0C00 to B484_37FC	-				Reserved
B484_3800	-		BT12_PCSR/BT12_PCSR/BT12_PRLL/- [R/W] H,W 00000000 00000000		Base Timer ch.12
B484_3804	-		-/BT12_PDUT/ BT12_PRLH/ BT12_DTBF [R/W] H,W 00000000 00000000		

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B484_3808	BT12_TMR/ BT12_TMR/ BT12_TMR/- [R/W] H,W 00000000 00000000 00000000 00000000 / (*9) 00000000 00000000 / (*10) XXXXXXXX XXXXXXXX(*4)				
B484_380C	-		BT12_TMCR [R/W] B,H,W 00000000 00000000		
B484_3810	-		BT12_TMCR2 [R/W] B,H,W 00000000	BT12_STC [R/W] B,H,W 00000000	
B484_3814	-		-	BT12_STCC [R/W] B,H,W 00000000	
B484_3818	-		-	BT12_STCS [R/W] B,H,W 00000000	
B484_381C	-		-/BT12_PSDR/-/- [R/W] H,W 00000000 00000000		
B484_3820	-		-/BT12_ADTR/-/- [R/W] H,W 00000000 00000000		
B484_3824	-		BT12_DEBUG [R/W] B,H,W 00000000 00000000		
B484_3828	-		-		
B484_382C	-		BT12_ETCDR [R/W] B,H,W 01111111 11111111		
B484_3830	-			BT_BTSEL1213 [R/W] B,H,W 11110000	
B484_3834	-		BT_BTSSSR12 [R/W] B,H,W 11111111 11111111		
B484_3838 to B484_3BFC	-				
B484_3C00	-		BT13_PCSR/BT13_PCSR/BT13_PRLL/- [R/W] H,W 00000000 00000000		Base Timer ch.13
B484_3C04	-		-/BT13_PDUT/ BT13_PRLH/ BT13_DTBF [R/W] H,W 00000000 00000000		
B484_3C08	BT13_TMR/ BT13_TMR/ BT13_TMR/- [R/W] H,W 00000000 00000000 00000000 00000000 / (*9) 00000000 00000000 / (*10) XXXXXXXX XXXXXXXX(*4)				
B484_3C0C	-		BT13_TMCR [R/W] B,H,W 00000000 00000000		
B484_3C10	-		BT13_TMCR2 [R/W] B,H,W 00000000	BT13_STC [R/W] B,H,W 00000000	
B484_3C14	-		-	BT13_STCC [R/W] B,H,W 00000000	
B484_3C18	-		-	BT13_STCS [R/W] B,H,W 00000000	
B484_3C1C	-		-/BT13_PSDR/-/- [R/W] H,W 00000000 00000000		
B484_3C20	-		-/BT13_ADTR/-/- [R/W] H,W 00000000 00000000		

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B484_3C24	-		BT13_DEBUG [R/W] B,H,W 00000000 00000000		
B484_3C28	-		-		
B484_3C2C	-		BT13_ETCDR [R/W] B,H,W 01111111 11111111		
B484_3C30 to B484_3FFC					
B484_4000	-		BT14_PCSR/BT14_PCSR/BT14_PRLL/- [R/W] H,W 00000000 00000000		Base Timer ch.14
B484_4004	-		-/BT14_PDUT/ BT14_PRLH/ BT14_DTBF [R/W] H,W 00000000 00000000		
B484_4008	BT14_TMR/ BT14_TMR/ BT14_TMR/- [R/W] H,W 00000000 00000000 00000000 00000000 / (*9) 00000000 00000000 / (*10) XXXXXXXX XXXXXXXX(*4)				
B484_400C	-		BT14_TMCR [R/W] B,H,W 00000000 00000000		
B484_4010	-		BT14_TMCR2 [R/W] B,H,W 00000000	BT14_STC [R/W] B,H,W 00000000	
B484_4014	-		-	BT14_STCC [R/W] B,H,W 00000000	
B484_4018	-		-	BT14_STCS [R/W] B,H,W 00000000	
B484_401C	-		-/BT14_PSDR/-/- [R/W] H,W 00000000 00000000		
B484_4020	-		-/BT14_ADTR/-/- [R/W] H,W 00000000 00000000		
B484_4024	-		BT14_DEBUG [R/W] B,H,W 00000000 00000000		
B484_4028	-		-		
B484_402C	-		BT14_ETCDR [R/W] B,H,W 01111111 11111111		
B484_4030	-			BT_BTSEL1415 [R/W] B,H,W 11110000	
B484_4034 to B484_43FC	-				
B484_4400	-		BT15_PCSR/BT15_PCSR/BT15_PRLL/- [R/W] H,W 00000000 00000000		Base Timer ch.15
B484_4404	-		-/BT15_PDUT/ BT15_PRLH/ BT15_DTBF [R/W] H,W 00000000 00000000		

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B484_4408	- BT15_TMR/ BT15_TMR/ BT15_TMR/- [R/W] H,W 00000000 00000000 00000000 00000000 / (*9) 00000000 00000000 / (*10) XXXXXXXX XXXXXXXX(*4)				
B484_440C	-		BT15_TMCR [R/W] B,H,W 00000000 00000000		
B484_4410	-		BT15_TMCR2 [R/W] B,H,W 00000000	BT15_STC [R/W] B,H,W 00000000	
B484_4414	-		-	BT15_STCC [R/W] B,H,W 00000000	
B484_4418	-		-	BT15_STCS [R/W] B,H,W 00000000	
B484_441C	-		-/BT15_PSDR/-/- [R/W] H,W 00000000 00000000		
B484_4420	-		-/BT15_ADTR/-/- [R/W] H,W 00000000 00000000		
B484_4424	-		BT15_DEBUG [R/W] B,H,W 00000000 00000000		
B484_4428	-		-		
B484_442C	-		BT15_ETCDR [R/W] B,H,W 01111111 11111111		
B484_4430 to B484_47FC	-				
B484_4800	-		BT16_PCSR/BT16_PCSR/BT16_PRLL/- [R/W] H,W 00000000 00000000		Base Timer ch.16
B484_4804	-		-/BT16_PDUT/ BT16_PRLH/ BT16_DTBF [R/W] H,W 00000000 00000000		
B484_4808	BT16_TMR/ BT16_TMR/ BT16_TMR/- [R/W] H,W 00000000 00000000 00000000 00000000 / (*9) 00000000 00000000 / (*10) XXXXXXXX XXXXXXXX(*4)				
B484_480C	-		BT16_TMCR [R/W] B,H,W 00000000 00000000		
B484_4810	-		BT16_TMCR2 [R/W] B,H,W 00000000	BT16_STC [R/W] B,H,W 00000000	
B484_4814	-		-	BT16_STCC [R/W] B,H,W 00000000	
B484_4818	-		-	BT16_STCS [R/W] B,H,W 00000000	
B484_481C	-		-/BT16_PSDR/-/- [R/W] H,W 00000000 00000000		
B484_4820	-		-/BT16_ADTR/-/- [R/W] H,W 00000000 00000000		
B484_4824	-		BT16_DEBUG [R/W] B,H,W 00000000 00000000		

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B484_4828	-		-		
B484_482C	-		BT16_ETCDR [R/W] B,H,W 01111111 11111111		
B484_4830	-			BT_BTSEL1617 [R/W] B,H,W 11110000	
B484_4834 to B484_4BFC	-				
B484_4C00	-		BT17_PCSR/BT17_PCSR/BT17_PRLL/- [R/W] H,W 00000000 00000000		Base Timer ch.17
B484_4C04	-		-/BT17_PDUT/ BT17_PRLH/ BT17_DTBF [R/W] H,W 00000000 00000000		
B484_4C08	BT17_TMR/ BT17_TMR/ BT17_TMR/- [R/W] H,W 00000000 00000000 00000000 00000000 / (*9) 00000000 00000000 / (*10) XXXXXXXX XXXXXXXX(*4)				
B484_4C0C	-		BT17_TMCR [R/W] B,H,W 00000000 00000000		
B484_4C10	-		BT17_TMCR2 [R/W] B,H,W 00000000	BT17_STC [R/W] B,H,W 00000000	
B484_4C14	-		-	BT17_STCC [R/W] B,H,W 00000000	
B484_4C18	-		-	BT17_STCS [R/W] B,H,W 00000000	
B484_4C1C	-		-/BT17_PSDR/-/- [R/W] H,W 00000000 00000000		
B484_4C20	-		-/BT17_ADTR/-/- [R/W] H,W 00000000 00000000		
B484_4C24	-		BT17_DEBUG [R/W] B,H,W 00000000 00000000		
B484_4C28	-		-		
B484_4C2C	-		BT17_ETCDR [R/W] B,H,W 01111111 11111111		
B484_4C30 to B484_4FFC	-				
B484_5000	-		BT18_PCSR/BT18_PCSR/BT18_PRLL/- [R/W] H,W 00000000 00000000		Base Timer ch.18
B484_5004	-		-/BT18_PDUT/ BT18_PRLH/ BT18_DTBF [R/W] H,W 00000000 00000000		
B484_5008	BT18_TMR/ BT18_TMR/ BT18_TMR/- [R/W] H,W 00000000 00000000 00000000 00000000 / (*9) 00000000 00000000 / (*10) XXXXXXXX XXXXXXXX(*4)				
B484_500C	-		BT18_TMCR [R/W] B,H,W 00000000 00000000		

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B484_5010	-		BT18_TMCR2 [R/W] B,H,W 00000000	BT18_STC [R/W] B,H,W 00000000	
B484_5014	-		-	BT18_STCC [R/W] B,H,W 00000000	
B484_5018	-		-	BT18_STCS [R/W] B,H,W 00000000	
B484_501C	-		-/BT18_PSDR/-/- [R/W] H,W 00000000 00000000		
B484_5020	-		-/BT18_ADTR/-/- [R/W] H,W 00000000 00000000		
B484_5024	-		BT18_DEBUG [R/W] B,H,W 00000000 00000000		
B484_5028	-		-		
B484_502C	-		BT18_ETCDR [R/W] B,H,W 01111111 11111111		
B484_5030	-			BT_BTSEL1819 [R/W] B,H,W 11110000	
B484_5034 to B484_53FC	-				
B484_5400	-		BT19_PCSR/BT19_PCSR/BT19_PRLL/- [R/W] H,W 00000000 00000000		Base Timer ch.19
B484_5404	-		-/BT19_PDUT/ BT19_PRLH/ BT19_DTBF [R/W] H,W 00000000 00000000		
B484_5408	BT19_TMR/ BT19_TMR/ BT19_TMR/- [R/W] H,W 00000000 00000000 00000000 00000000 / (*9) 00000000 00000000 / (*10) XXXXXXXX XXXXXXXX(*4)				
B484_540C	-		BT19_TMCR [R/W] B,H,W 00000000 00000000		
B484_5410	-		BT19_TMCR2 [R/W] B,H,W 00000000	BT19_STC [R/W] B,H,W 00000000	
B484_5414	-		-	BT19_STCC [R/W] B,H,W 00000000	
B484_5418	-		-	BT19_STCS [R/W] B,H,W 00000000	
B484_541C	-		-/BT19_PSDR/-/- [R/W] H,W 00000000 00000000		
B484_5420	-		-/BT19_ADTR/-/- [R/W] H,W 00000000 00000000		
B484_5424	-		BT19_DEBUG [R/W] B,H,W 00000000 00000000		
B484_5428	-		-		
B484_542C	-		BT19_ETCDR [R/W] B,H,W 01111111 11111111		

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B484_5430 to B484_57FC	-				
B484_5800	-		BT20_PCSR/BT20_PCSR/BT20_PRLL/- [R/W] H,W 00000000 00000000		Base Timer ch.20
B484_5804	-		-/BT20_PDUT/ BT20_PRLH/ BT20_DTBF [R/W] H,W 00000000 00000000		
B484_5808	BT20_TMR/ BT20_TMR/ BT20_TMR/- [R/W] H,W 00000000 00000000 00000000 00000000 / (*9) 00000000 00000000 / (*10) XXXXXXXX XXXXXXXX(*4)				
B484_580C	-		BT20_TMCR [R/W] B,H,W 00000000 00000000		
B484_5810	-		BT20_TMCR2 [R/W] B,H,W 00000000	BT20_STC [R/W] B,H,W 00000000	
B484_5814	-		-	BT20_STCC [R/W] B,H,W 00000000	
B484_5818	-		-	BT20_STCS [R/W] B,H,W 00000000	
B484_581C	-		-/BT20_PSDR/-/- [R/W] H,W 00000000 00000000		
B484_5820	-		-/BT20_ADTR/-/- [R/W] H,W 00000000 00000000		
B484_5824	-		BT20_DEBUG [R/W] B,H,W 00000000 00000000		
B484_5828	-		-		
B484_582C	-		BT20_ETCDR [R/W] B,H,W 01111111 11111111		
B484_5830	-			BT_BTSEL2021 [R/W] B,H,W 11110000	
B484_5834 to B484_5BFC	-				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B484_5C00	-		BT21_PCSR/BT21_PCSR/BT21_PRLL/- [R/W] H,W 00000000 00000000		Base Timer ch.21
B484_5C04	-		-/BT21_PDUT/ BT21_PRLH/ BT21_DTBF [R/W] H,W 00000000 00000000		
B484_5C08	BT21_TMR/ BT21_TMR/ BT21_TMR/- [R/W] H,W 00000000 00000000 00000000 00000000 / (*9) 00000000 00000000 / (*10) XXXXXXXX XXXXXXXX(*4)				
B484_5C0C	-		BT21_TMCR [R/W] B,H,W 00000000 00000000		
B484_5C10	-		BT21_TMCR2 [R/W] B,H,W 00000000	BT21_STC [R/W] B,H,W 00000000	
B484_5C14	-		-	BT21_STCC [R/W] B,H,W 00000000	
B484_5C18	-		-	BT21_STCS [R/W] B,H,W 00000000	
B484_5C1C	-		-/BT21_PSDR/-/- [R/W] H,W 00000000 00000000		
B484_5C20	-		-/BT21_ADTR/-/- [R/W] H,W 00000000 00000000		
B484_5C24	-		BT21_DEBUG [R/W] B,H,W 00000000 00000000		
B484_5C28	-		-		
B484_5C2C	-		BT21_ETCDR [R/W] B,H,W 01111111 11111111		
B484_5C30 to B484_5FFC	-				
B484_6000	-		BT22_PCSR/BT22_PCSR/BT22_PRLL/- [R/W] H,W 00000000 00000000		Base Timer ch.22
B484_6004	-		-/BT22_PDUT/ BT22_PRLH/ BT22_DTBF [R/W] H,W 00000000 00000000		
B484_6008	BT22_TMR/ BT22_TMR/ BT22_TMR/- [R/W] H,W 00000000 00000000 00000000 00000000 / (*9) 00000000 00000000 / (*10) XXXXXXXX XXXXXXXX(*4)				
B484_600C	-		BT22_TMCR [R/W] B,H,W 00000000 00000000		
B484_6010	-		BT22_TMCR2 [R/W] B,H,W 00000000	BT22_STC [R/W] B,H,W 00000000	
B484_6014	-		-	BT22_STCC [R/W] B,H,W 00000000	
B484_6018	-		-	BT22_STCS [R/W] B,H,W 00000000	
B484_601C	-		-/BT22_PSDR/-/- [R/W] H,W 00000000 00000000		

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B484_6020	-		-/BT22_ADTR/-/- [R/W] H,W 00000000 00000000		
B484_6024	-		BT22_DEBUG [R/W] B,H,W 00000000 00000000		
B484_6028	-		-		
B484_602C	-		BT22_ETCDR [R/W] B,H,W 01111111 11111111		
B484_6030	-			BT_BTSEL2223 [R/W] B,H,W 11110000	
B484_6034 to B484_63FC	-				
B484_6400	-		BT23_PCSR/BT23_PCSR/BT23_PRLL/- [R/W] H,W 00000000 00000000		Base Timer ch.23
B484_6404	-		-/BT23_PDUT/ BT23_PRLH/ BT23_DTBF [R/W] H,W 00000000 00000000		
B484_6408	BT23_TMR/ BT23_TMR/ BT23_TMR/- [R/W] H,W 00000000 00000000 00000000 00000000 / (*9) 00000000 00000000 / (*10) XXXXXXXX XXXXXXXX(*4)				
B484_640C	-		BT23_TMCR [R/W] B,H,W 00000000 00000000		
B484_6410	-		BT23_TMCR2 [R/W] B,H,W 00000000	BT23_STC [R/W] B,H,W 00000000	
B484_6414	-		-	BT23_STCC [R/W] B,H,W 00000000	
B484_6418	-		-	BT23_STCS [R/W] B,H,W 00000000	
B484_641C	-		-/BT23_PSDR/-/- [R/W] H,W 00000000 00000000		
B484_6420	-		-/BT23_ADTR/-/- [R/W] H,W 00000000 00000000		
B484_6424	-		BT23_DEBUG [R/W] B,H,W 00000000 00000000		
B484_6428	-		-		
B484_642C	-		BT23_ETCDR [R/W] B,H,W 01111111 11111111		
B484_6430 to B484_67FC	-				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B484_6800	-		BT24_PCSR/BT24_PCSR/BT24_PRLL/- [R/W] H,W 00000000 00000000		Base Timer ch.24
B484_6804	-		-/BT24_PDUT/ BT24_PRLH/ BT24_DTBF [R/W] H,W 00000000 00000000		
B484_6808	BT24_TMR/ BT24_TMR/ BT24_TMR/- [R/W] H,W 00000000 00000000 00000000 00000000 / (*9) 00000000 00000000 / (*10) XXXXXXXX XXXXXXXX(*4)				
B484_680C	-		BT24_TMCR [R/W] B,H,W 00000000 00000000		
B484_6810	-		BT24_TMCR2 [R/W] B,H,W 00000000	BT24_STC [R/W] B,H,W 00000000	
B484_6814	-		-	BT24_STCC [R/W] B,H,W 00000000	
B484_6818	-		-	BT24_STCS [R/W] B,H,W 00000000	
B484_681C	-		-/BT24_PSDR/-/- [R/W] H,W 00000000 00000000		
B484_6820	-		-/BT24_ADTR/-/- [R/W] H,W 00000000 00000000		
B484_6824	-		BT24_DEBUG [R/W] B,H,W 00000000 00000000		
B484_6828	-		-		
B484_682C	-		BT24_ETCDR [R/W] B,H,W 01111111 11111111		
B484_6830	-			BT_BTSEL2425 [R/W] B,H,W 11110000	
B484_6834	-		BT_BTSSSR24 [R/W] B,H,W 11111111 11111111		
B484_6838 to B484_6BFC	-				
B484_6C00	-		BT25_PCSR/BT25_PCSR/BT25_PRLL/- [R/W] H,W 00000000 00000000		Base Timer ch.25
B484_6C04	-		-/BT25_PDUT/ BT25_PRLH/ BT25_DTBF [R/W] H,W 00000000 00000000		
B484_6C08	BT25_TMR/ BT25_TMR/ BT25_TMR/- [R/W] H,W 00000000 00000000 00000000 00000000 / (*9) 00000000 00000000 / (*10) XXXXXXXX XXXXXXXX(*4)				
B484_6C0C	-		BT25_TMCR [R/W] B,H,W 00000000 00000000		
B484_6C10	-		BT25_TMCR2 [R/W] B,H,W 00000000	BT25_STC [R/W] B,H,W 00000000	
B484_6C14	-		-	BT25_STCC [R/W] B,H,W 00000000	

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B484_6C18	-		-	BT25_STCS [R/W] B,H,W 00000000	
B484_6C1C	-		-/BT25_PSDR/-/- [R/W] H,W 00000000 00000000		
B484_6C20	-		-/BT25_ADTR/-/- [R/W] H,W 00000000 00000000		
B484_6C28	-		BT25_DEBUG [R/W] B,H,W 00000000 00000000		
B484_6C24	-		-		
B484_6C2C	-		BT25_ETCDR [R/W] B,H,W 01111111 11111111		
B484_6C30 to B484_6FFC	-				
B484_7000	-		BT26_PCSR/BT26_PCSR/BT26_PRLL/- [R/W] H,W 00000000 00000000		Base Timer ch.26
B484_7004	-		-/BT26_PDUT/ BT26_PRLH/ BT26_DTBF [R/W] H,W 00000000 00000000		
B484_7008	BT26_TMR/ BT26_TMR/ BT26_TMR/- [R/W] H,W 00000000 00000000 00000000 00000000 / (*9) 00000000 00000000 / (*10) XXXXXXXX XXXXXXXX(*4)				
B484_700C	-		BT26_TMCR [R/W] B,H,W 00000000 00000000		
B484_7010	-		BT26_TMCR2 [R/W] B,H,W 00000000	BT26_STC [R/W] B,H,W 00000000	
B484_7014	-		-	BT26_STCC [R/W] B,H,W 00000000	
B484_7018	-		-	BT26_STCS [R/W] B,H,W 00000000	
B484_701C	-		-/BT26_PSDR/-/- [R/W] H,W 00000000 00000000		
B484_7020	-		-/BT26_ADTR/-/- [R/W] H,W 00000000 00000000		
B484_7024	-		BT26_DEBUG [R/W] B,H,W 00000000 00000000		
B484_7028	-		-		
B484_702C	-		BT26_ETCDR [R/W] B,H,W 01111111 11111111		
B484_7030	-			BT_BTSEL2627 [R/W] B,H,W 11110000	
B484_7034 to B484_73FC	-				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B484_7400	-		BT27_PCSR/BT27_PCSR/BT27_PRLL/- [R/W] H,W 00000000 00000000		Base Timer ch.27
B484_7404	-		-/BT27_PDUT/ BT27_PRLH/ BT27_DTBF [R/W] H,W 00000000 00000000		
B484_7408	BT27_TMR/ BT27_TMR/ BT27_TMR/- [R/W] H,W 00000000 00000000 00000000 00000000 / (*9) 00000000 00000000 / (*10) XXXXXXXX XXXXXXXX(*4)				
B484_740C	-		BT27_TMCR [R/W] B,H,W 00000000 00000000		
B484_7410	-		BT27_TMCR2 [R/W] B,H,W 00000000	BT27_STC [R/W] B,H,W 00000000	
B484_7414	-		-	BT27_STCC [R/W] B,H,W 00000000	
B484_7418	-		-	BT27_STCS [R/W] B,H,W 00000000	
B484_741C	-		-/BT27_PSDR/-/- [R/W] H,W 00000000 00000000		
B484_7420	-		-/BT27_ADTR/-/- [R/W] H,W 00000000 00000000		
B484_7424	-		BT27_DEBUG [R/W] B,H,W 00000000 00000000		
B484_7428	-		-		
B484_742C	-		BT27_ETCDR [R/W] B,H,W 01111111 11111111		
B484_7430 to B484_77FC	-				
B484_7800	-		BT28_PCSR/BT28_PCSR/BT28_PRLL/- [R/W] H,W 00000000 00000000		Base Timer ch.28
B484_7804	-		-/BT28_PDUT/ BT28_PRLH/ BT28_DTBF [R/W] H,W 00000000 00000000		
B484_7808	BT28_TMR/ BT28_TMR/ BT28_TMR/- [R/W] H,W 00000000 00000000 00000000 00000000 / (*9) 00000000 00000000 / (*10) XXXXXXXX XXXXXXXX(*4)				
B484_780C	-		BT28_TMCR [R/W] B,H,W 00000000 00000000		
B484_7810	-		BT28_TMCR2 [R/W] B,H,W 00000000	BT28_STC [R/W] B,H,W 00000000	
B484_7814	-		-	BT28_STCC [R/W] B,H,W 00000000	
B484_7818	-		-	BT28_STCS [R/W] B,H,W 00000000	
B484_781C	-		-/BT28_PSDR/-/- [R/W] H,W 00000000 00000000		

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B484_7820	-		-/BT28_ADTR/-/- [R/W] H,W 00000000 00000000		
B484_7824	-		BT28_DEBUG [R/W] B,H,W 00000000 00000000		
B484_7828	-		-		
B484_782C	-		BT28_ETCDR [R/W] B,H,W 01111111 11111111		
B484_7830	-			BT_BTSEL2829 [R/W] B,H,W 11110000	
B484_7834 to B484_7BFC	-				
B484_7C00	-		BT29_PCSR/BT29_PCSR/BT29_PRLL/- [R/W] H,W 00000000 00000000		Base Timer ch.29
B484_7C04	-		-/BT29_PDUT/ BT29_PRLH/ BT29_DTBF [R/W] H,W 00000000 00000000		
B484_7C08	BT29_TMR/ BT29_TMR/ BT29_TMR/- [R/W] H,W 00000000 00000000 00000000 00000000 / (*9) 00000000 00000000 / (*10) XXXXXXXX XXXXXXXX(*4)				
B484_7C0C	-		BT29_TMCR [R/W] B,H,W 00000000 00000000		
B484_7C10	-		BT29_TMCR2 [R/W] B,H,W 00000000	BT29_STC [R/W] B,H,W 00000000	
B484_7C14	-		-	BT29_STCC [R/W] B,H,W 00000000	
B484_7C18	-		-	BT29_STCS [R/W] B,H,W 00000000	
B484_7C1C	-		-/BT29_PSDR/-/- [R/W] H,W 00000000 00000000		
B484_7C20	-		-/BT29_ADTR/-/- [R/W] H,W 00000000 00000000		
B484_7C24	-		BT29_DEBUG [R/W] B,H,W 00000000 00000000		
B484_7C28	-		-		
B484_7C2C	-		BT29_ETCDR [R/W] B,H,W 01111111 11111111		
B484_7C30 to B484_7FFC	-				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B484_8000	–				ADC0
B484_8004	–	–	–	ADTSS0[R/W] B,H,W -------0	
B484_8008	ADTSE0[R/W] B,H,W 00000000 00000000 00000000 00000000				
B484_800C	–		–		
B484_8010	ADCOMP3/ADCOMPB3[R/W] H,W 00000000 00000000		–		
B484_8014	ADCOMP5/ADCOMPB5[R/W] H,W 00000000 00000000		–		
B484_8018	–		ADCOMP6/ADCOMPB6[R/W] H,W 00000000 00000000		
B484_801C	ADCOMP9/ADCOMPB9[R/W] H,W 00000000 00000000		–		
B484_8020	–		ADCOMP10/ADCOMPB10[R/W] H,W 00000000 00000000		
B484_8024	ADCOMP13/ADCOMPB13[R/W] H,W 00000000 00000000		ADCOMP12/ADCOMPB12[R/W] H,W 00000000 00000000		
B484_8028	ADCOMP15/ADCOMPB15[R/W] H,W 00000000 00000000		ADCOMP14/ADCOMPB14[R/W] H,W 00000000 00000000		
B484_802C	ADCOMP17/ADCOMPB17[R/W] H,W 00000000 00000000		–		
B484_8030	ADCOMP19/ADCOMPB19[R/W] H,W 00000000 00000000		ADCOMP18/ADCOMPB18[R/W] H,W 00000000 00000000		
B484_8034	ADCOMP21/ADCOMPB21[R/W] H,W 00000000 00000000		ADCOMP20/ADCOMPB20[R/W] H,W 00000000 00000000		
B484_8038	ADCOMP23/ADCOMPB23[R/W] H,W 00000000 00000000		ADCOMP22/ADCOMPB22[R/W] H,W 00000000 00000000		
B484_803C	–		ADCOMP24/ADCOMPB24[R/W] H,W 00000000 00000000		
B484_8040	ADCOMP27/ADCOMPB27[R/W] H,W 00000000 00000000		–		
B484_8044	ADCOMP29/ADCOMPB29[R/W] H,W 00000000 00000000		ADCOMP28/ADCOMPB28[R/W] H,W 00000000 00000000		
B484_8048	ADCOMP31/ADCOMPB31[R/W] H,W 00000000 00000000		ADCOMP30/ADCOMPB30[R/W] H,W 00000000 00000000		
B484_804C	–		–		
B484_8050	ADTCS3[R/W] B,H,W 00000000 0010----		–		
B484_8054	ADTCS5[R/W] B,H,W 00000000 0010----		–		
B484_8058	–		ADTCS6[R/W] B,H,W 00000000 0010----		
B484_805C	ADTCS9[R/W] B,H,W 00000000 0010----		–		

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B484_8060	–		ADTCS10[R/W] B,H,W 00000000 0010----		
B484_8064	ADTCS13[R/W] B,H,W 00000000 0010----		ADTCS12[R/W] B,H,W 00000000 0010----		
B484_8068	ADTCS15[R/W] B,H,W 00000000 0010----		ADTCS14[R/W] B,H,W 00000000 0010----		
B484_806C	ADTCS17[R/W] B,H,W 00000000 0010----		–		
B484_8070	ADTCS19[R/W] B,H,W 00000000 0010----		ADTCS18[R/W] B,H,W 00000000 0010----		
B484_8074	ADTCS21[R/W] B,H,W 00000000 0010----		ADTCS20[R/W] B,H,W 00000000 0010----		
B484_8078	ADTCS23[R/W] B,H,W 00000000 0010----		ADTCS22[R/W] B,H,W 00000000 0010----		
B484_807C	–		ADTCS24[R/W] B,H,W 00000000 0010----		
B484_8080	ADTCS27[R/W] B,H,W 00000000 0010----		–		
B484_8084	ADTCS29[R/W] B,H,W 00000000 0010----		ADTCS28[R/W] B,H,W 00000000 0010----		
B484_8088	ADTCS31[R/W] B,H,W 00000000 0010----		ADTCS30[R/W] B,H,W 00000000 0010----		
B484_808C	–		–		
B484_8090	ADTCD3[R] B,H,W 10--0000 00000000		–		
B484_8094	ADTCD5[R] B,H,W 10--0000 00000000		–		
B484_8098	–		ADTCD6[R] B,H,W 10--0000 00000000		
B484_809C	ADTCD9[R] B,H,W 10--0000 00000000		–		
B484_80A0	–		ADTCD10[R] B,H,W 10--0000 00000000		
B484_80A4	ADTCD13[R] B,H,W 10--0000 00000000		ADTCD12[R] B,H,W 10--0000 00000000		
B484_80A8	ADTCD15[R] B,H,W 10--0000 00000000		ADTCD14[R] B,H,W 10--0000 00000000		
B484_80AC	ADTCD17[R] B,H,W 10--0000 00000000		–		
B484_80B0	ADTCD19[R] B,H,W 10--0000 00000000		ADTCD18[R] B,H,W 10--0000 00000000		
B484_80B4	ADTCD21[R] B,H,W 10--0000 00000000		ADTCD20[R] B,H,W 10--0000 00000000		
B484_80B8	ADTCD23[R] B,H,W 10--0000 00000000		ADTCD22[R] B,H,W 10--0000 00000000		
B484_80BC	–		ADTCD24[R] B,H,W 10--0000 00000000		

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B484_80C0	ADTCD27[R] B,H,W 10--0000 00000000		–		
B484_80C4	ADTCD29[R] B,H,W 10--0000 00000000		ADTCD28[R] B,H,W 10--0000 00000000		
B484_80C8	ADTCD31[R] B,H,W 10--0000 00000000		ADTCD30[R] B,H,W 10--0000 00000000		
B484_80CC	–		–		
B484_80D0	ADTECS3[R/W] B,H,W -------0 ---00000		–		
B484_80D4	ADTECS5[R/W] B,H,W -------0 ---00000		–		
B484_80D8	–		ADTECS6[R/W] B,H,W -------0 ---00000		
B484_80DC	ADTECS9[R/W] B,H,W -------0 ---00000		–		
B484_80E0	–		ADTECS10[R/W] B,H,W -------0 ---00000		
B484_80E4	ADTECS13[R/W] B,H,W -------0 ---00000		ADTECS12[R/W] B,H,W -------0 ---00000		
B484_80E8	ADTECS15[R/W] B,H,W -------0 ---00000		ADTECS14[R/W] B,H,W -------0 ---00000		
B484_80EC	ADTECS17[R/W] B,H,W -------0 ---00000		–		
B484_80F0	ADTECS19[R/W] B,H,W -------0 ---00000		ADTECS18[R/W] B,H,W -------0 ---00000		
B484_80F4	ADTECS21[R/W] B,H,W -------0 ---00000		ADTECS20[R/W] B,H,W -------0 ---00000		
B484_80F8	ADTECS23[R/W] B,H,W -------0 ---00000		ADTECS22[R/W] B,H,W -------0 ---00000		
B484_80FC	–		ADTECS24[R/W] B,H,W -------0 ---00000		
B484_8100	ADTECS27[R/W] B,H,W -------0 ---00000		–		
B484_8104	ADTECS29[R/W] B,H,W -------0 ---00000		ADTECS28[R/W] B,H,W -------0 ---00000		
B484_8108	ADTECS31[R/W] B,H,W -------0 ---00000		ADTECS30[R/W] B,H,W -------0 ---00000		
B484_810C	ADRCLT0[R/W] B,H,W ----0000 00000000		ADRCUT0[R/W] B,H,W ----0000 00000000		
B484_8110	ADRCLT1[R/W] B,H,W ----0000 00000000		ADRCUT1[R/W] B,H,W ----0000 00000000		
B484_8114	ADRCLT2[R/W] B,H,W ----0000 00000000		ADRCUT2[R/W] B,H,W ----0000 00000000		
B484_8118	ADRCLT3[R/W] B,H,W ----0000 00000000		ADRCUT3[R/W] B,H,W ----0000 00000000		
B484_811C	ADRCCS3[R/W] B,H,W 00100000	–	–	–	

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B484_8120	–	ADRCCS6[R/W] B,H,W 00100000	ADRCCS5[R/W] B,H,W 00100000	–	
B484_8124	–	ADRCCS10[R/W] B,H,W 00100000	ADRCCS9[R/W] B,H,W 00100000	–	
B484_8128	ADRCCS15[R/W] B,H,W 00100000	ADRCCS14[R/W] B,H,W 00100000	ADRCCS13[R/W] B,H,W 00100000	ADRCCS12[R/W] B,H,W 00100000	
B484_812C	ADRCCS19[R/W] B,H,W 00100000	ADRCCS18[R/W] B,H,W 00100000	ADRCCS17[R/W] B,H,W 00100000	–	
B484_8130	ADRCCS23[R/W] B,H,W 00100000	ADRCCS22[R/W] B,H,W 00100000	ADRCCS21[R/W] B,H,W 00100000	ADRCCS20[R/W] B,H,W 00100000	
B484_8134	ADRCCS27[R/W] B,H,W 00100000	–	–	ADRCCS24[R/W] B,H,W 00100000	
B484_8138	ADRCCS31[R/W] B,H,W 00100000	ADRCCS30[R/W] B,H,W 00100000	ADRCCS29[R/W] B,H,W 00100000	ADRCCS28[R/W] B,H,W 00100000	
B484_813C	ADRCOT0[R] B,H,W 00000000 00000000 00000000 00000000				
B484_8140	ADRCIF0[R] B,H,W 00000000 00000000 00000000 00000000				
B484_8144	–	–	–	ADSCANS0[R/W] B,H,W 000-----	
B484_8148	ADNCS3[R/W] B,H,W 0-000-00	ADNCS2[R/W] B,H,W 0-000-00	ADNCS1[R/W] B,H,W 0-000-00	ADNCS0[R/W] B,H,W 0-000-00	
B484_814C	ADNCS7[R/W] B,H,W 0-000-00	ADNCS6[R/W] B,H,W 0-000-00	ADNCS5[R/W] B,H,W 0-000-00	ADNCS4[R/W] B,H,W 0-000-00	
B484_8150	ADNCS11[R/W] B,H,W 0-000-00	ADNCS10[R/W] B,H,W 0-000-00	ADNCS9[R/W] B,H,W 0-000-00	ADNCS8[R/W] B,H,W 0-000-00	
B484_8154	ADNCS15[R/W] B,H,W 0-000-00	ADNCS14[R/W] B,H,W 0-000-00	ADNCS13[R/W] B,H,W 0-000-00	ADNCS12[R/W] B,H,W 0-000-00	
B484_8158	ADPRTF0[R] B,H,W 00000000 00000000 00000000 00000000				
B484_815C	ADEOCF0[R] B,H,W 11111111 11111111 11111111 11111111				
B484_8160	–		–		
B484_8164	ADTCSC3[W] B,H,W 00------ --------		–		
B484_8168	ADTCSC5[W] B,H,W 00------ --------		–		
B484_816C	–		ADTCSC6[W] B,H,W 00------ --------		
B484_8170	ADTCSC9[W] B,H,W 00------ --------		–		
B484_8174	–		ADTCSC10[W] B,H,W 00------ --------		
B484_8178	ADTCSC13[W] B,H,W 00------ --------		ADTCSC12[W] B,H,W 00------ --------		
B484_817C	ADTCSC15[W] B,H,W 00------ --------		ADTCSC14[W] B,H,W 00------ --------		

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B484_8180	ADTCSC17[W] B,H,W 00------ --------		–		
B484_8184	ADTCSC19[W] B,H,W 00------ --------		ADTCSC18[W] B,H,W 00------ --------		
B484_8188	ADTCSC21[W] B,H,W 00------ --------		ADTCSC20[W] B,H,W 00------ --------		
B484_818C	ADTCSC23[W] B,H,W 00------ --------		ADTCSC22[W] B,H,W 00------ --------		
B484_8190	–		ADTCSC24[W] B,H,W 00------ --------		
B484_8194	ADTCSC27[W] B,H,W 00------ --------		–		
B484_8198	ADTCSC29[W] B,H,W 00------ --------		ADTCSC28[W] B,H,W 00------ --------		
B484_819C	ADTCSC31[W] B,H,W 00------ --------		ADTCSC30[W] B,H,W 00------ --------		
B484_81A0	ADRCS3[R] B,H,W 00000000	–	–	–	
B484_81A4	–	ADRCS6[R] B,H,W 00000000	ADRCS5[R] B,H,W 00000000	–	
B484_81A8	–	ADRCS10[R] B,H,W 00000000	ADRCS9[R] B,H,W 00000000	–	
B484_81AC	ADRCS15[R] B,H,W 00000000	ADRCS14[R] B,H,W 00000000	ADRCS13[R] B,H,W 00000000	ADRCS12[R] B,H,W 00000000	
B484_81B0	ADRCS19[R] B,H,W 00000000	ADRCS18[R] B,H,W 00000000	ADRCS17[R] B,H,W 00000000	–	
B484_81B4	ADRCS23[R] B,H,W 00000000	ADRCS22[R] B,H,W 00000000	ADRCS21[R] B,H,W 00000000	ADRCS20[R] B,H,W 00000000	
B484_81B8	ADRCS27[R] B,H,W 00000000	–	–	ADRCS24[R] B,H,W 00000000	
B484_81BC	ADRCS31[R] B,H,W 00000000	ADRCS30[R] B,H,W 00000000	ADRCS29[R] B,H,W 00000000	ADRCS28[R] B,H,W 00000000	
B484_81C0	ADRCIFC0[W] B,H,W 00000000 00000000 00000000 00000000				
B484_81C4	–	–	–	ADSCANSC0[W] B,H,W 0-------	
B484_81C8	ADMD0 [R/W] B,H,W 0---0000	ADCH0 [R] B,H,W ---00000	ADCS0 [R] B,H,W 0------- --------		
B484_81CC	ADSTPCS3 [R/W] B,H,W 00000000	ADSTPCS2 [R/W] B,H,W 00000000	ADSTPCS1 [R/W] B,H,W 00000000	ADSTPCS0 [R/W] B,H,W 00000000	
B484_81D0	ADSTPCS7 [R/W] B,H,W 00000000	ADSTPCS6 [R/W] B,H,W 00000000	ADSTPCS5 [R/W] B,H,W 00000000	ADSTPCS4 [R/W] B,H,W 00000000	
B484_81D4	ADFDCTR0 [R/W] B,H,W 11111111 11111111 11111111 00000000				
B484_81D8 to B484_83FC	–				

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B484_8400	–				ADC1/PWU
B484_8404	–	–	–	ADTSS1[R/W] B,H,W -------0	
B484_8408	ADTSE1[R/W] B,H,W 00000000 00000000 00000000 00000000				
B484_840C	ADCOMP33/ADCOMPB33[R/W] H,W 00000000 00000000		ADCOMP32/ADCOMPB32[R/W] H,W 00000000 00000000		
B484_8410	ADCOMP35/ADCOMPB35[R/W] H,W 00000000 00000000		ADCOMP34/ADCOMPB34[R/W] H,W 00000000 00000000		
B484_8414	ADCOMP37/ADCOMPB37[R/W] H,W 00000000 00000000		ADCOMP36/ADCOMPB36[R/W] H,W 00000000 00000000		
B484_8418	ADCOMP39/ADCOMPB39[R/W] H,W 00000000 00000000		ADCOMP38/ADCOMPB38[R/W] H,W 00000000 00000000		
B484_841C	ADCOMP41/ADCOMPB41[R/W] H,W 00000000 00000000		ADCOMP40/ADCOMPB40[R/W] H,W 00000000 00000000		
B484_8420	ADCOMP43/ADCOMPB43[R/W] H,W 00000000 00000000		ADCOMP42/ADCOMPB42[R/W] H,W 00000000 00000000		
B484_8424	–		–		
B484_8428	ADCOMP47/ADCOMPB47[R/W] H,W 00000000 00000000		ADCOMP46/ADCOMPB46[R/W] H,W 00000000 00000000		
B484_842C	ADCOMP49/ADCOMPB49[R/W] H,W 00000000 00000000		ADCOMP48/ADCOMPB48[R/W] H,W 00000000 00000000		
B484_8430	ADCOMP51/ADCOMPB51[R/W] H,W 00000000 00000000		ADCOMP50/ADCOMPB50[R/W] H,W 00000000 00000000		
B484_8434	ADCOMP53/ADCOMPB53[R/W] H,W 00000000 00000000		ADCOMP52/ADCOMPB52[R/W] H,W 00000000 00000000		
B484_8438	ADCOMP55/ADCOMPB55[R/W] H,W 00000000 00000000		–		
B484_843C	ADCOMP57/ADCOMPB57[R/W] H,W 00000000 00000000		ADCOMP56/ADCOMPB56[R/W] H,W 00000000 00000000		
B484_8440	ADCOMP59/ADCOMPB59[R/W] H,W 00000000 00000000		ADCOMP58/ADCOMPB58[R/W] H,W 00000000 00000000		
B484_8444	ADCOMP61/ADCOMPB61[R/W] H,W 00000000 00000000		ADCOMP60/ADCOMPB60[R/W] H,W 00000000 00000000		
B484_8448	–		ADCOMP62/ADCOMPB62[R/W] H,W 00000000 00000000		
B484_844C	ADTCS33[R/W] B,H,W 00000000 0010----		ADTCS32[R/W] B,H,W 00000000 0010----		
B484_8450	ADTCS35[R/W] B,H,W 00000000 0010----		ADTCS34[R/W] B,H,W 00000000 0010----		
B484_8454	ADTCS37[R/W] B,H,W 00000000 0010----		ADTCS36[R/W] B,H,W 00000000 0010----		
B484_8458	ADTCS39[R/W] B,H,W 00000000 0010----		ADTCS38[R/W] B,H,W 00000000 0010----		
B484_845C	ADTCS41[R/W] B,H,W 00000000 0010----		ADTCS40[R/W] B,H,W 00000000 0010----		

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B484_8460	ADTCS43[R/W] B,H,W 00000000 0010----		ADTCS42[R/W] B,H,W 00000000 0010----		
B484_8464	–		–		
B484_8468	ADTCS47[R/W] B,H,W 00000000 0010----		ADTCS46[R/W] B,H,W 00000000 0010----		
B484_846C	ADTCS49[R/W] B,H,W 00000000 0010----		ADTCS48[R/W] B,H,W 00000000 0010----		
B484_8470	ADTCS51[R/W] B,H,W 00000000 0010----		ADTCS50[R/W] B,H,W 00000000 0010----		
B484_8474	ADTCS53[R/W] B,H,W 00000000 0010----		ADTCS52[R/W] B,H,W 00000000 0010----		
B484_8478	ADTCS55[R/W] B,H,W 00000000 0010----		–		
B484_847C	ADTCS57[R/W] B,H,W 00000000 0010----		ADTCS56[R/W] B,H,W 00000000 0010----		
B484_8480	ADTCS59[R/W] B,H,W 00000000 0010----		ADTCS58[R/W] B,H,W 00000000 0010----		
B484_8484	ADTCS61[R/W] B,H,W 00000000 0010----		ADTCS60[R/W] B,H,W 00000000 0010----		
B484_8488	–		ADTCS62[R/W] B,H,W 00000000 0010----		
B484_848C	ADTCD33[R] B,H,W 10--0000 00000000		ADTCD32[R] B,H,W 10--0000 00000000		
B484_8490	ADTCD35[R] B,H,W 10--0000 00000000		ADTCD34[R] B,H,W 10--0000 00000000		
B484_8494	ADTCD37[R] B,H,W 10--0000 00000000		ADTCD36[R] B,H,W 10--0000 00000000		
B484_8498	ADTCD39[R] B,H,W 10--0000 00000000		ADTCD38[R] B,H,W 10--0000 00000000		
B484_849C	ADTCD41[R] B,H,W 10--0000 00000000		ADTCD40[R] B,H,W 10--0000 00000000		
B484_84A0	ADTCD43[R] B,H,W 10--0000 00000000		ADTCD42[R] B,H,W 10--0000 00000000		
B484_84A4	–		–		
B484_84A8	ADTCD47[R] B,H,W 10--0000 00000000		ADTCD46[R] B,H,W 10--0000 00000000		
B484_84AC	ADTCD49[R] B,H,W 10--0000 00000000		ADTCD48[R] B,H,W 10--0000 00000000		
B484_84B0	ADTCD51[R] B,H,W 10--0000 00000000		ADTCD50[R] B,H,W 10--0000 00000000		
B484_84B4	ADTCD53[R] B,H,W 10--0000 00000000		ADTCD52[R] B,H,W 10--0000 00000000		
B484_84B8	ADTCD55[R] B,H,W 10--0000 00000000		–		
B484_84BC	ADTCD57[R] B,H,W 10--0000 00000000		ADTCD56[R] B,H,W 10--0000 00000000		

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B484_84C0	ADTCD59[R] B,H,W 10--0000 00000000		ADTCD58[R] B,H,W 10--0000 00000000		
B484_84C4	ADTCD61[R] B,H,W 10--0000 00000000		ADTCD60[R] B,H,W 10--0000 00000000		
B484_84C8	–		ADTCD62[R] B,H,W 10--0000 00000000		
B484_84CC	ADTECS33[R/W] B,H,W -------0 ---00000		ADTECS32[R/W] B,H,W -------0 ---00000		
B484_84D0	ADTECS35[R/W] B,H,W -------0 ---00000		ADTECS34[R/W] B,H,W -------0 ---00000		
B484_84D4	ADTECS37[R/W] B,H,W -------0 ---00000		ADTECS36[R/W] B,H,W -------0 ---00000		
B484_84D8	ADTECS39[R/W] B,H,W -------0 ---00000		ADTECS38[R/W] B,H,W -------0 ---00000		
B484_84DC	ADTECS41[R/W] B,H,W -------0 ---00000		ADTECS40[R/W] B,H,W -------0 ---00000		
B484_84E0	ADTECS43[R/W] B,H,W -------0 ---00000		ADTECS42[R/W] B,H,W -------0 ---00000		
B484_84E4	–		–		
B484_84E8	ADTECS47[R/W] B,H,W -------0 ---00000		ADTECS46[R/W] B,H,W -------0 ---00000		
B484_84EC	ADTECS49[R/W] B,H,W -------0 ---00000		ADTECS48[R/W] B,H,W -------0 ---00000		
B484_84F0	ADTECS51[R/W] B,H,W -------0 ---00000		ADTECS50[R/W] B,H,W -------0 ---00000		
B484_84F4	ADTECS53[R/W] B,H,W -------0 ---00000		ADTECS52[R/W] B,H,W -------0 ---00000		
B484_84F8	ADTECS55[R/W] B,H,W -------0 ---00000		–		
B484_84FC	ADTECS57[R/W] B,H,W -------0 ---00000		ADTECS56[R/W] B,H,W -------0 ---00000		
B484_8500	ADTECS59[R/W] B,H,W -------0 ---00000		ADTECS58[R/W] B,H,W -------0 ---00000		
B484_8504	ADTECS61[R/W] B,H,W -------0 ---00000		ADTECS60[R/W] B,H,W -------0 ---00000		
B484_8508	–		ADTECS62[R/W] B,H,W -------0 ---00000		
B484_850C	ADRCLT4[R/W] B,H,W ----0000 00000000		ADRCUT4[R/W] B,H,W ----0000 00000000		
B484_8510	ADRCLT5[R/W] B,H,W ----0000 00000000		ADRCUT5[R/W] B,H,W ----0000 00000000		
B484_8514	ADRCLT6[R/W] B,H,W ----0000 00000000		ADRCUT6[R/W] B,H,W ----0000 00000000		
B484_8518	ADRCLT7[R/W] B,H,W ----0000 00000000		ADRCUT7[R/W] B,H,W ----0000 00000000		
B484_851C	ADRCCS35[R/W] B,H,W 00100000	ADRCCS34[R/W] B,H,W 00100000	ADRCCS33[R/W] B,H,W 00100000	ADRCCS32[R/W] B,H,W 00100000	

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B484_8520	ADRCCS39[R/W] B,H,W 00100000	ADRCCS38[R/W] B,H,W 00100000	ADRCCS37[R/W] B,H,W 00100000	ADRCCS36[R/W] B,H,W 00100000	
B484_8524	ADRCCS43[R/W] B,H,W 00100000	ADRCCS42[R/W] B,H,W 00100000	ADRCCS41[R/W] B,H,W 00100000	ADRCCS40[R/W] B,H,W 00100000	
B484_8528	ADRCCS47[R/W] B,H,W 00100000	ADRCCS46[R/W] B,H,W 00100000	–	–	
B484_852C	ADRCCS51[R/W] B,H,W 00100000	ADRCCS50[R/W] B,H,W 00100000	ADRCCS49[R/W] B,H,W 00100000	ADRCCS48[R/W] B,H,W 00100000	
B484_8530	ADRCCS55[R/W] B,H,W 00100000	–	ADRCCS53[R/W] B,H,W 00100000	ADRCCS52[R/W] B,H,W 00100000	
B484_8534	ADRCCS59[R/W] B,H,W 00100000	ADRCCS58[R/W] B,H,W 00100000	ADRCCS57[R/W] B,H,W 00100000	ADRCCS56[R/W] B,H,W 00100000	
B484_8538	–	ADRCCS62[R/W] B,H,W 00100000	ADRCCS61[R/W] B,H,W 00100000	ADRCCS60[R/W] B,H,W 00100000	
B484_853C	ADRCOT1[R] B,H,W 00000000 00000000 00000000 00000000				
B484_8540	ADRCIF1[R] B,H,W 00000000 00000000 00000000 00000000				
B484_8544	–	–	–	ADSCANS1[R/W] B,H,W 000-----	
B484_8548	ADNCS19[R/W] B,H,W 0-000-00	ADNCS18[R/W] B,H,W 0-000-00	ADNCS17[R/W] B,H,W 0-000-00	ADNCS16[R/W] B,H,W 0-000-00	
B484_854C	ADNCS23[R/W] B,H,W 0-000-00	–	ADNCS21[R/W] B,H,W 0-000-00	ADNCS20[R/W] B,H,W 0-000-00	
B484_8550	ADNCS27[R/W] B,H,W 0-000-00	ADNCS26[R/W] B,H,W 0-000-00	ADNCS25[R/W] B,H,W 0-000-00	ADNCS24[R/W] B,H,W 0-000-00	
B484_8554	ADNCS31[R/W] B,H,W 0-000-00	ADNCS30[R/W] B,H,W 0-000-00	ADNCS29[R/W] B,H,W 0-000-00	ADNCS28[R/W] B,H,W 0-000-00	
B484_8558	ADPRTF1[R] B,H,W 00000000 00000000 00000000 00000000				
B484_855C	ADEOCF1[R] B,H,W 11111111 11111111 11111111 11111111				
B484_8560	ADTCSC33[W] B,H,W 00------ --------		ADTCSC32[W] B,H,W 00------ --------		
B484_8564	ADTCSC35[W] B,H,W 00------ --------		ADTCSC34[W] B,H,W 00------ --------		
B484_8568	ADTCSC37[W] B,H,W 00------ --------		ADTCSC36[W] B,H,W 00------ --------		
B484_856C	ADTCSC39[W] B,H,W 00------ --------		ADTCSC38[W] B,H,W 00------ --------		
B484_8570	ADTCSC41[W] B,H,W 00------ --------		ADTCSC40[W] B,H,W 00------ --------		
B484_8574	ADTCSC43[W] B,H,W 00------ --------		ADTCSC42[W] B,H,W 00------ --------		
B484_8578	–		–		
B484_857C	ADTCSC47[W] B,H,W 00------ --------		ADTCSC46[W] B,H,W 00------ --------		

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B484_8580	ADTCSC49[W] B,H,W 00------ --------		ADTCSC48[W] B,H,W 00------ --------		
B484_8584	ADTCSC51[W] B,H,W 00------ --------		ADTCSC50[W] B,H,W 00------ --------		
B484_8588	ADTCSC53[W] B,H,W 00------ --------		ADTCSC52[W] B,H,W 00------ --------		
B484_858C	ADTCSC55[W] B,H,W 00------ --------		–		
B484_8590	ADTCSC57[W] B,H,W 00------ --------		ADTCSC56[W] B,H,W 00------ --------		
B484_8594	ADTCSC59[W] B,H,W 00------ --------		ADTCSC58[W] B,H,W 00------ --------		
B484_8598	ADTCSC61[W] B,H,W 00------ --------		ADTCSC60[W] B,H,W 00------ --------		
B484_859C	–		ADTCSC62[W] B,H,W 00------ --------		
B484_85A0	ADRCS35[R] B,H,W 00000000	ADRCS34[R] B,H,W 00000000	ADRCS33[R] B,H,W 00000000	ADRCS32[R] B,H,W 00000000	
B484_85A4	ADRCS39[R] B,H,W 00000000	ADRCS38[R] B,H,W 00000000	ADRCS37[R] B,H,W 00000000	ADRCS36[R] B,H,W 00000000	
B484_85A8	ADRCS43[R] B,H,W 00000000	ADRCS42[R] B,H,W 00000000	ADRCS41[R] B,H,W 00000000	ADRCS40[R] B,H,W 00000000	
B484_85AC	ADRCS47[R] B,H,W 00000000	ADRCS46[R] B,H,W 00000000	–	–	
B484_85B0	ADRCS51[R] B,H,W 00000000	ADRCS50[R] B,H,W 00000000	ADRCS49[R] B,H,W 00000000	ADRCS48[R] B,H,W 00000000	
B484_85B4	ADRCS55[R] B,H,W 00000000	–	ADRCS53[R] B,H,W 00000000	ADRCS52[R] B,H,W 00000000	
B484_85B8	ADRCS59[R] B,H,W 00000000	ADRCS58[R] B,H,W 00000000	ADRCS57[R] B,H,W 00000000	ADRCS56[R] B,H,W 00000000	
B484_85BC	–	ADRCS62[R] B,H,W 00000000	ADRCS61[R] B,H,W 00000000	ADRCS60[R] B,H,W 00000000	
B484_85C0	ADRCIFC1[W] B,H,W 00000000 00000000 00000000 00000000				
B484_85C4	–	–	–	ADSCANSC1[W] B,H,W 0-------	
B484_85C8	ADMD1 [R/W] B,H,W 0---0000	ADCH1 [R] B,H,W ---00000	ADCS1 [R] B,H,W 0------- --------		
B484_85CC	ADSTPCS11 [R/W] B,H,W 00000000	ADSTPCS10 [R/W] B,H,W 00000000	ADSTPCS9 [R/W] B,H,W 00000000	ADSTPCS8 [R/W] B,H,W 00000000	
B484_85D0	ADSTPCS15 [R/W] B,H,W 00000000	ADSTPCS14 [R/W] B,H,W 00000000	ADSTPCS13 [R/W] B,H,W 00000000	ADSTPCS12 [R/W] B,H,W 00000000	
B484_85D4	ADFDCTR1 [R/W] B,H,W 11111111 11111111 11111111 00000000				
B484_85D8 to B484_85FC	–				Reserved

Address	address offset value / register name				Block name
	+3	+2	+1	+0	
B484_8600	–	–	–	PWUC[R/W] B,H,W 00000000	Reserved
B484_8604 to B484_87FC	–				Reserved
B484_8800	KEYCDR[W] W 0000---- -------- ----0000 00000000				ADER
B484_8804	ADER0[R/W] B,H,W 11111111 11111111 11111111 11111111				
B484_8808	ADER1[R/W] B,H,W 11111111 11111111 11111111 11111111				
B484_880C to B484_8BFC	–				Reserved
B484_8C00 to B484_8FFC	–				Reserved
B484_9000	–		GTCPCLR[R/W] H, 11111111 11111111		Global Timer (16bit)
B484_9004	–		GT_TCDT,[R/W]H 00000000 00000000		
B484_9008	–		GT_TCCS[R/,W]B,H ----00000 01-00000		
B484_900C	–		-		
B484_9010	–		GT_TCCSC [R/W] B,H, ------0 -0-0----		
B484_9414	–		GT_TCCSS [R/W] B,H, ------0 -0-0----		
B484_9018	–		GT_DEBUG [R/W] B,H, ---- ---0		
B484_901C	–				
B484_9020	–			GT_MSCTR [R/W] B,H, -----000	
B484_9024	GT_MSCHB[R/W] W -0000000 00000000 00000000 00000000				
B484_9028	-				
B484_902C	GT_MSCH[R/W] W -0000000 00000000 00000000 00000000				
B484_9030 to B484_FFFC	–				

CHAPTER 65: APPENDIX: DMA Channel Activation Factors

This appendix shows DMA channel Activation Factors.

1. DMA Channel Activation Factors (S6J312xHzC*)

1. DMA Channel Activation Factors (S6J312xHzC*)

This section shows DMA channel activation factors in S6J312xHzC*.

DMA sources from the peripherals in S6J312xHzC*.

Number of Channels	Peripheral Functions	Type
0 to 7	Reserved	-
8	Ext-IRQ	1
9	Work Flash DREQ	1
10	FRT ch.0 Compare clear	2
11	FRT ch.0 Zero detection	2
12	FRT ch.1 Compare clear	2
13	FRT ch.1 Zero detection	2
14	FRT ch.2 Compare clear	2
15	FRT ch.2 Zero detection	2
16	FRT ch.3 Compare clear	2
17	FRT ch.3 Zero detection	2
18	FRT ch.4 Compare clear	2
19	FRT ch.4 Zero detection	2
20	FRT ch.5 Compare clear	2
21	FRT ch.5 Zero detection	2
22	ICU ch.0 DREQ	2
23	ICU ch.1 DREQ	2
24	ICU ch.2 DREQ	2
25	ICU ch.3 DREQ	2
26	ICU ch.4 DREQ	2
27	ICU ch.5 DREQ	2
28	ICU ch.6 DREQ	2
29	ICU ch.7 DREQ	2
30	ICU ch.8 DREQ	2
31	ICU ch.9 DREQ	2
32	ICU ch.10 DREQ	2
33	ICU ch.11 DREQ	2
34	OCU ch.0 DREQ	2
35	OCU ch.1 DREQ	2
36	OCU ch.2 DREQ	2
37	OCU ch.3 DREQ	2
38	OCU ch.4 DREQ	2
39	OCU ch.5 DREQ	2
40	OCU ch.6 DREQ	2
41	OCU ch.7 DREQ	2
42	OCU ch.8 DREQ	2
43	OCU ch.9 DREQ	2
44	OCU ch.10 DREQ	2
45	OCU ch.11 DREQ	2
46	Base Timer ch.0-0 DREQ	2
47	Base Timer ch.0-1 DREQ	2

Number of Channels	Peripheral Functions	Type
48	Base Timer ch.1-0 DREQ	2
49	Base Timer ch.1-1 DREQ	2
50	Base Timer ch.2-0 DREQ	2
51	Base Timer ch.2-1 DREQ	2
52	Base Timer ch.3-0 DREQ	2
53	Base Timer ch.3-1 DREQ	2
54	Base Timer ch.4-0 DREQ	2
55	Base Timer ch.4-1 DREQ	2
56	Base Timer ch.5-0 DREQ	2
57	Base Timer ch.5-1 DREQ	2
58	Base Timer ch.6-0 DREQ	2
59	Base Timer ch.6-1 DREQ	2
60	Base Timer ch.7-0 DREQ	2
61	Base Timer ch.7-1 DREQ	2
62	Base Timer ch.8-0 DREQ	2
63	Base Timer ch.8-1 DREQ	2
64	Base Timer ch.9-0 DREQ	2
65	Base Timer ch.9-1 DREQ	2
66	Base Timer ch.10-0 DREQ	2
67	Base Timer ch.10-1 DREQ	2
68	Base Timer ch.11-0 DREQ	2
69	Base Timer ch.11-1 DREQ	2
70	MFS ch.0 Tx DREQ	3
71	MFS ch.0 Rx DREQ	3
72	MFS ch.1 Tx DREQ	3
73	MFS ch.1 Rx DREQ	3
74	MFS ch.2 Tx DREQ	3
75	MFS ch.2 Rx DREQ	3
76	MFS ch.3 Tx DREQ	3
77	MFS ch.3 Rx DREQ	3
78	MFS ch.4 Tx DREQ	3
79	MFS ch.4 Rx DREQ	3
80 to 85	Reserved	-
86	A/D unit.0 DREQ	2
87	A/D unit.1 DREQ	2
88	Reserved	-
89	Reserved	-
90	DMAC RLT_DREQ ch.0	1
91	DMAC RLT_DREQ ch.1	1
92	DMAC RLT_DREQ ch.2	1
93	DMAC RLT_DREQ ch.3	1
94 to 119	Reserved	-
120	Reserved	-
121 to 123	Reserved	-
124	HSSPI TX-FIFO DREQ	1
125	HSSPI RX-FIFO DREQ	1
126 to 135	Reserved	-

Number of Channels	Peripheral Functions	Type
136	RLT ch.0 DREQ	1
137	RLT ch.1 DREQ	1
138	RLT ch.2 DREQ	1
139	RLT ch.3 DREQ	1
140 to 151	Reserved	-
152	RLT ch.16 DREQ	1
153	RLT ch.17 DREQ	1
154	RLT ch.18 DREQ	1
155	RLT ch.19 DREQ	1
156 to 167	Reserved	-
168	RLT ch.32 DREQ	1
169	RLT ch.33 DREQ	1
170 to 199	Reserved	-
200	MFS ch.8 RX DREQ	3
201	MFS ch.8 TX DREQ	3
202	MFS ch.9 RX DREQ	3
203	MFS ch.9 TX DREQ	3
204	MFS ch.10 RX DREQ	3
205	MFS ch.10 TX DREQ	3
206	MFS ch.11 RX DREQ	3
207	MFS ch.11 TX DREQ	3
208	MFS ch.12 RX DREQ	3
209	MFS ch.12 TX DREQ	3
210 to 328	Reserved	-
329	Sound Generator ch.0	2
330	Sound Generator ch.1	2
331	Sound Generator ch.2	2
332 to 511	Reserved	-

* x: A/9, z:A/B

Type	Automatically cleared the interrupt factor flag of peripheral function by acceptance of DMA transfer request	Interrupt Enable
1	Not cleared.	Unnecessary(*1)
2	Cleared	Necessary
3(*2)	Not cleared	Necessary

*

*1: Necessary to enable DMA.

*2: Only Multi Function Serial function. The interrupt flag which is the factor of DMA transfer request is cleared when data transfer is started.

CHAPTER 66: APPENDIX: List of Interrupt Vector and NMI Map

This appendix shows List of Interrupt Vector and NMI Map

1. List of Interrupt Vector (S6J312xHzC*)
2. NMI Map (S6J312xHzC*)

LISTINT-T01P00R01L01-E1-GBM02

1. List of Interrupt Vector (S6J312xHzC*)

This section shows list of interrupt vector in S6J312xHzC*.
This list shows the assignments of interrupt factors and interrupt vectors/interrupt control registers.

*x: A/9, z: A/B

IRQ No.	Detail	IRQ Priority Register	Vector Address Register
0	Reserved		
1	System Control Status Interrupt Request	IRCn_IRQPL0:IRQPL1[4:0]	IRCn_IRQVA1
2	HW-WDT Pre-warning Interrupt Request	IRCn_IRQPL0:IRQPL2[4:0]	IRCn_IRQVA2
3	SW-WDT Pre-warning Interrupt Request	IRCn_IRQPL0:IRQPL3[4:0]	IRCn_IRQVA3
4 to 7	Reserved		
8	TCFLASH Interrupt Request	IRCn_IRQPL2:IRQPL8[4:0]	IRCn_IRQVA8
9	Reserved		
10	Work FLASH HANG Interrupt Request	IRCn_IRQPL2:IRQPL10[4:0]	IRCn_IRQVA10
11 to 13	Reserved		
14	Scratch Pad RAM Single Bit Error Interrupt Request	IRCn_IRQPL3:IRQPL14[4:0]	IRCn_IRQVA14
15	Backup RAM / CAN FD RAMs Single Bit Error Interrupt Request	IRCn_IRQPL3:IRQPL15[4:0]	IRCn_IRQVA15
16	IRC0 Vector Address RAM Single Bit Error Interrupt Request	IRCn_IRQPL4:IRQPL16[4:0]	IRCn_IRQVA16
17 to 18	Reserved		
19	Debugger Interrupt Request	IRCn_IRQPL4:IRQPL19[4:0]	IRCn_IRQVA19
20	Work FLASH RDY Interrupt Request/Single Bit Error Interrupt Request/WE Release Interrupt Request	IRCn_IRQPL5:IRQPL20[4:0]	IRCn_IRQVA20
21 to 22	Reserved		
23	EICU	IRCn_IRQPL5:IRQPL23[4:0]	IRCn_IRQVA23
24	External Interrupt Request ch.0	IRCn_IRQPL6:IRQPL24[4:0]	IRCn_IRQVA24
25	External Interrupt Request ch.1	IRCn_IRQPL6:IRQPL25[4:0]	IRCn_IRQVA25
26	External Interrupt Request ch.2	IRCn_IRQPL6:IRQPL26[4:0]	IRCn_IRQVA26
27	External Interrupt Request ch.3	IRCn_IRQPL6:IRQPL27[4:0]	IRCn_IRQVA27
28	External Interrupt Request ch.4	IRCn_IRQPL7:IRQPL28[4:0]	IRCn_IRQVA28
29	External Interrupt Request ch.5	IRCn_IRQPL7:IRQPL29[4:0]	IRCn_IRQVA29
30	External Interrupt Request ch.6	IRCn_IRQPL7:IRQPL30[4:0]	IRCn_IRQVA30
31	External Interrupt Request ch.7	IRCn_IRQPL7:IRQPL31[4:0]	IRCn_IRQVA31

IRQ No.	Detail	IRQ Priority Register	Vector Address Register
32	External Interrupt Request ch.8	IRCn_IRQPL8:IRQPL32[4:0]	IRCn_IRQVA32
33	External Interrupt Request ch.9	IRCn_IRQPL8:IRQPL33[4:0]	IRCn_IRQVA33
34	External Interrupt Request ch.10	IRCn_IRQPL8:IRQPL34[4:0]	IRCn_IRQVA34
35	External Interrupt Request ch.11	IRCn_IRQPL8:IRQPL35[4:0]	IRCn_IRQVA35
36	External Interrupt Request ch.12	IRCn_IRQPL9:IRQPL36[4:0]	IRCn_IRQVA36
37	External Interrupt Request ch.13	IRCn_IRQPL9:IRQPL37[4:0]	IRCn_IRQVA37
38	External Interrupt Request ch.14	IRCn_IRQPL9:IRQPL38[4:0]	IRCn_IRQVA38
39	External Interrupt Request ch.15	IRCn_IRQPL9:IRQPL39[4:0]	IRCn_IRQVA39
40 to 55	Reserved		
56	CAN-FD ch.0 (OR-ed of all factors)	IRCn_IRQPL14:IRQPL56[4:0]	IRCn_IRQVA56
57	CAN-FD ch.1 (OR-ed of all factors)	IRCn_IRQPL14:IRQPL57[4:0]	IRCn_IRQVA57
58	CAN-FD ch.2 (OR-ed of all factors)	IRCn_IRQPL14:IRQPL58[4:0]	IRCn_IRQVA58
59 to 63	Reserved		
64	M.F.S RX ch.0	IRCn_IRQPL16:IRQPL64[4:0]	IRCn_IRQVA64
65	M.F.S TX ch.0	IRCn_IRQPL16:IRQPL65[4:0]	IRCn_IRQVA65
66	M.F.S Rx ch.1	IRCn_IRQPL16:IRQPL66[4:0]	IRCn_IRQVA66
67	M.F.S TX ch.1	IRCn_IRQPL16:IRQPL67[4:0]	IRCn_IRQVA67
68	M.F.S Rx ch.2	IRCn_IRQPL17:IRQPL68[4:0]	IRCn_IRQVA68
69	M.F.S TX ch.2	IRCn_IRQPL17:IRQPL69[4:0]	IRCn_IRQVA69
70	M.F.S Rx ch.3	IRCn_IRQPL17:IRQPL70[4:0]	IRCn_IRQVA70
71	M.F.S TX ch.3	IRCn_IRQPL17:IRQPL71[4:0]	IRCn_IRQVA71
72	M.F.S Rx ch.4	IRCn_IRQPL18:IRQPL72[4:0]	IRCn_IRQVA72
73	M.F.S TX ch.4	IRCn_IRQPL18:IRQPL73[4:0]	IRCn_IRQVA73
74 to 79	Reserved		
80	M.F.S Rx ch.8	IRCn_IRQPL20:IRQPL80[4:0]	IRCn_IRQVA80
81	M.F.S TX ch.8	IRCn_IRQPL20:IRQPL81[4:0]	IRCn_IRQVA81
82	M.F.S Rx ch.9	IRCn_IRQPL20:IRQPL82[4:0]	IRCn_IRQVA82
83	M.F.S TX ch.9	IRCn_IRQPL20:IRQPL83[4:0]	IRCn_IRQVA83
84	M.F.S Rx ch.10	IRCn_IRQPL21:IRQPL84[4:0]	IRCn_IRQVA84
85	M.F.S TX ch.10	IRCn_IRQPL21:IRQPL85[4:0]	IRCn_IRQVA85
86	M.F.S Rx ch.11	IRCn_IRQPL21:IRQPL86[4:0]	IRCn_IRQVA86

IRQ No.	Detail	IRQ Priority Register	Vector Address Register
87	M.F.S TX ch.11	IRCn_IRQPL21:IRQPL87[4:0]	IRCn_IRQVA87
88	M.F.S Rx ch.12	IRCn_IRQPL22:IRQPL88[4:0]	IRCn_IRQVA88
89	M.F.S TX ch.12	IRCn_IRQPL22:IRQPL89[4:0]	IRCn_IRQVA89
90 to 99	Reserved		
100	SHE Err IRQ	IRCn_IRQPL25:IRQPL100[4:0]	IRCn_IRQVA100
101	SHE IRQ	IRCn_IRQPL25:IRQPL101[4:0]	IRCn_IRQVA101
102	DDR_HSSPI ch.0 RX	IRCn_IRQPL25:IRQPL102[4:0]	IRCn_IRQVA102
103	DDR_HSSPI ch.0 TX	IRCn_IRQPL25:IRQPL103[4:0]	IRCn_IRQVA103
104 to 109	Reserved		
110	TCRAM diag	IRCn_IRQPL27:IRQPL110[4:0]	IRCn_IRQVA110
111	Reserved		
112	Backup RAM diag	IRCn_IRQPL28:IRQPL112[4:0]	IRCn_IRQVA112
113 to 115	Reserved		
116	RTC+Calibration IRQ	IRCn_IRQPL29:IRQPL116[4:0]	IRCn_IRQVA116
117	CR calibration IRQ	IRCn_IRQPL29:IRQPL117[4:0]	IRCn_IRQVA117
118 to 127	Reserved		
128	Base Timer ch.0/8/9/10/11	IRCn_IRQPL32:IRQPL128[4:0]	IRCn_IRQVA128
129	Base Timer ch.1	IRCn_IRQPL32:IRQPL129[4:0]	IRCn_IRQVA129
130	Base Timer ch.2	IRCn_IRQPL32:IRQPL130[4:0]	IRCn_IRQVA130
131	Base Timer ch.3	IRCn_IRQPL32:IRQPL131[4:0]	IRCn_IRQVA131
132	Base Timer ch.4	IRCn_IRQPL33:IRQPL132[4:0]	IRCn_IRQVA132
133	Base Timer ch.5	IRCn_IRQPL33:IRQPL133[4:0]	IRCn_IRQVA133
134	Base Timer ch.6	IRCn_IRQPL33:IRQPL134[4:0]	IRCn_IRQVA134
135	Base Timer ch.7	IRCn_IRQPL33:IRQPL135[4:0]	IRCn_IRQVA135
136 to 151	Reserved		
152	ReloadTimer ch.0	IRCn_IRQPL38:IRQPL152[4:0]	IRCn_IRQVA152
153	ReloadTimer ch.1	IRCn_IRQPL38:IRQPL153[4:0]	IRCn_IRQVA153
154	ReloadTimer ch.2	IRCn_IRQPL38:IRQPL154[4:0]	IRCn_IRQVA154
155	ReloadTimer ch.3	IRCn_IRQPL38:IRQPL155[4:0]	IRCn_IRQVA155
156 to 159	Reserved		
160	ReloadTimer ch.16	IRCn_IRQPL40:IRQPL160[4:0]	IRCn_IRQVA160
161	ReloadTimer ch.17	IRCn_IRQPL40:IRQPL161[4:0]	IRCn_IRQVA161
162	ReloadTimer ch.18	IRCn_IRQPL40:IRQPL162[4:0]	IRCn_IRQVA162

IRQ No.	Detail	IRQ Priority Register	Vector Address Register
163	ReloadTimer ch.19	IRCn_IRQPL40:IRQPL163[4:0]	IRCn_IRQVA163
164 to 167	Reserved		
168	ReloadTimer ch.32	IRCn_IRQPL42:IRQPL168[4:0]	IRCn_IRQVA168
169	ReloadTimer ch.33	IRCn_IRQPL42:IRQPL169[4:0]	IRCn_IRQVA169
170 to 175	Reserved		
176	FRT ch.0	IRCn_IRQPL44:IRQPL176[4:0]	IRCn_IRQVA176
177	FRT ch.1	IRCn_IRQPL44:IRQPL177[4:0]	IRCn_IRQVA177
178	FRT ch.2	IRCn_IRQPL44:IRQPL178[4:0]	IRCn_IRQVA178
179	FRT ch.3	IRCn_IRQPL44:IRQPL179[4:0]	IRCn_IRQVA179
180	FRT ch.4	IRCn_IRQPL45:IRQPL180[4:0]	IRCn_IRQVA180
181	FRT ch.5	IRCn_IRQPL45:IRQPL181[4:0]	IRCn_IRQVA181
182 to 191	Reserved		
192	ICU ch.0	IRCn_IRQPL48:IRQPL192[4:0]	IRCn_IRQVA192
193	ICU ch.2	IRCn_IRQPL48:IRQPL193[4:0]	IRCn_IRQVA193
194	ICU ch.4	IRCn_IRQPL48:IRQPL194[4:0]	IRCn_IRQVA194
195	ICU ch.6	IRCn_IRQPL48:IRQPL195[4:0]	IRCn_IRQVA195
196	ICU ch.8	IRCn_IRQPL49:IRQPL196[4:0]	IRCn_IRQVA196
197	ICU ch.10	IRCn_IRQPL49:IRQPL197[4:0]	IRCn_IRQVA197
198 to 207	Reserved		
208	OCU ch.0	IRCn_IRQPL52:IRQPL208[4:0]	IRCn_IRQVA208
209	OCU ch.2	IRCn_IRQPL52:IRQPL209[4:0]	IRCn_IRQVA209
210	OCU ch.4	IRCn_IRQPL52:IRQPL210[4:0]	IRCn_IRQVA210
211	OCU ch.6	IRCn_IRQPL52:IRQPL211[4:0]	IRCn_IRQVA211
212	OCU ch.8	IRCn_IRQPL53:IRQPL212[4:0]	IRCn_IRQVA212
213	OCU ch.10	IRCn_IRQPL53:IRQPL213[4:0]	IRCn_IRQVA213
214 to 231	Reserved		
232	QPRC ch.8	IRCn_IRQPL58:IRQPL232[4:0]	IRCn_IRQVA232
233	QPRC ch.9	IRCn_IRQPL58:IRQPL233[4:0]	IRCn_IRQVA233
234 to 239	Reserved		
240	ICU ch.1	IRCn_IRQPL60:IRQPL240[4:0]	IRCn_IRQVA240
241	ICU ch.3	IRCn_IRQPL60:IRQPL241[4:0]	IRCn_IRQVA241
242	ICU ch.5	IRCn_IRQPL60:IRQPL242[4:0]	IRCn_IRQVA242

IRQ No.	Detail	IRQ Priority Register	Vector Address Register
243	ICU ch.7	IRCn_IRQPL60:IRQPL243[4:0]	IRCn_IRQVA243
244	ICU ch.9	IRCn_IRQPL61:IRQPL244[4:0]	IRCn_IRQVA244
245	ICU ch.11	IRCn_IRQPL61:IRQPL245[4:0]	IRCn_IRQVA245
246 to 255	Reserved		
256	OCU ch.1	IRCn_IRQPL64:IRQPL256[4:0]	IRCn_IRQVA256
257	OCU ch.3	IRCn_IRQPL64:IRQPL257[4:0]	IRCn_IRQVA257
258	OCU ch.5	IRCn_IRQPL64:IRQPL258[4:0]	IRCn_IRQVA258
259	OCU ch.7	IRCn_IRQPL64:IRQPL259[4:0]	IRCn_IRQVA259
260	OCU ch.9	IRCn_IRQPL65:IRQPL260[4:0]	IRCn_IRQVA260
261	OCU ch.11	IRCn_IRQPL65:IRQPL261[4:0]	IRCn_IRQVA261
262 to 271	Reserved		
272	DMA Error Interrupt Request	IRCn_IRQPL68:IRQPL272[4:0]	IRCn_IRQVA272
273	DMA Completion Interrupt Request ch.0	IRCn_IRQPL68:IRQPL273[4:0]	IRCn_IRQVA273
274	DMA Completion Interrupt Request ch.1	IRCn_IRQPL68:IRQPL274[4:0]	IRCn_IRQVA274
275	DMA Completion Interrupt Request ch.2	IRCn_IRQPL68:IRQPL275[4:0]	IRCn_IRQVA275
276	DMA Completion Interrupt Request ch.3	IRCn_IRQPL69:IRQPL276[4:0]	IRCn_IRQVA276
277	DMA Completion Interrupt Request ch.4	IRCn_IRQPL69:IRQPL277[4:0]	IRCn_IRQVA277
278	DMA Completion Interrupt Request ch.5	IRCn_IRQPL69:IRQPL278[4:0]	IRCn_IRQVA278
279	DMA Completion Interrupt Request ch.6	IRCn_IRQPL69:IRQPL279[4:0]	IRCn_IRQVA279
280	DMA Completion Interrupt Request ch.7	IRCn_IRQPL70:IRQPL280[4:0]	IRCn_IRQVA280
281	Reserved		
282	DMAC RLT ch.0, 1, 2, 3	IRCn_IRQPL70:IRQPL282[4:0]	IRCn_IRQVA282
283 to 284	Reserved		
285	DMA Completion Interrupt Request ch.8	IRCn_IRQPL71:IRQPL285[4:0]	IRCn_IRQVA285
286	DMA Completion Interrupt Request ch.9	IRCn_IRQPL71:IRQPL286[4:0]	IRCn_IRQVA286
287	DMA Completion Interrupt Request ch.10	IRCn_IRQPL71:IRQPL287[4:0]	IRCn_IRQVA287
288	DMA Completion Interrupt Request ch.11	IRCn_IRQPL72:IRQPL288[4:0]	IRCn_IRQVA288
289	DMA Completion Interrupt Request ch.12	IRCn_IRQPL72:IRQPL289[4:0]	IRCn_IRQVA289
290	DMA Completion Interrupt Request ch.13	IRCn_IRQPL72:IRQPL290[4:0]	IRCn_IRQVA290
291	DMA Completion Interrupt Request ch.14	IRCn_IRQPL72:IRQPL291[4:0]	IRCn_IRQVA291
292	DMA Completion Interrupt Request ch.15	IRCn_IRQPL73:IRQPL292[4:0]	IRCn_IRQVA292
293 to 307	Reserved		
308	SCT RC IRQ	IRCn_IRQPL77:IRQPL308[4:0]	IRCn_IRQVA308

IRQ No.	Detail	IRQ Priority Register	Vector Address Register
309	SCT SRC IRQ	IRCn_IRQPL77:IRQPL309[4:0]	IRCn_IRQVA309
310	SCT Main OSC IRQ	IRCn_IRQPL77:IRQPL310[4:0]	IRCn_IRQVA310
311	Reserved		
312	CR5 Performance Monitor Unit IRQ	IRCn_IRQPL78:IRQPL312[4:0]	IRCn_IRQVA312
313	Reserved		
314 to 319	Reserved		
320	M.F.S Error ch.0 (TX error, RX error OR-ed)	IRCn_IRQPL80:IRQPL320[4:0]	IRCn_IRQVA320
321	M.F.S Error ch.1 (TX error, RX error OR-ed)	IRCn_IRQPL80:IRQPL321[4:0]	IRCn_IRQVA321
322	M.F.S Error ch.2 (TX error, RX error OR-ed)	IRCn_IRQPL80:IRQPL322[4:0]	IRCn_IRQVA322
323	M.F.S Error ch.3 (TX error, RX error OR-ed)	IRCn_IRQPL80:IRQPL323[4:0]	IRCn_IRQVA323
324	M.F.S Error ch.4 (TX error, RX error OR-ed)	IRCn_IRQPL81:IRQPL324[4:0]	IRCn_IRQVA324
325 to 327	Reserved		
328	M.F.S Error ch.8 (TX error, RX error OR-ed)	IRCn_IRQPL82:IRQPL328[4:0]	IRCn_IRQVA328
329	M.F.S Error ch.9 (TX error, RX error OR-ed)	IRCn_IRQPL82:IRQPL329[4:0]	IRCn_IRQVA329
330	M.F.S Error ch.10 (TX error, RX error OR-ed)	IRCn_IRQPL82:IRQPL330[4:0]	IRCn_IRQVA330
331	M.F.S Error ch.11 (TX error, RX error OR-ed)	IRCn_IRQPL82:IRQPL331[4:0]	IRCn_IRQVA331
332	M.F.S Error ch.12 (TX error, RX error OR-ed)	IRCn_IRQPL83:IRQPL332[4:0]	IRCn_IRQVA332
333 to 380	Reserved		
381	Base Timer ch.12/20/21/22/23	IRCn_IRQPL95:IRQPL381[4:0]	IRCn_IRQVA381
382	Base Timer ch.13	IRCn_IRQPL95:IRQPL382[4:0]	IRCn_IRQVA382
383	Base Timer ch.14	IRCn_IRQPL95:IRQPL383[4:0]	IRCn_IRQVA383
384	Base Timer ch.15	IRCn_IRQPL96:IRQPL384[4:0]	IRCn_IRQVA384
385	Base Timer ch.16	IRCn_IRQPL96:IRQPL385[4:0]	IRCn_IRQVA385
386	Base Timer ch.17	IRCn_IRQPL96:IRQPL386[4:0]	IRCn_IRQVA386
387	Base Timer ch.18	IRCn_IRQPL96:IRQPL387[4:0]	IRCn_IRQVA387
388	Base Timer ch.19	IRCn_IRQPL97:IRQPL388[4:0]	IRCn_IRQVA388

IRQ No.	Detail	IRQ Priority Register	Vector Address Register
389	Base Timer ch.24	IRCn_IRQPL97:IRQPL389[4:0]	IRCn_IRQVA389
390	Base Timer ch.25	IRCn_IRQPL97:IRQPL390[4:0]	IRCn_IRQVA390
391	Base Timer ch.26	IRCn_IRQPL97:IRQPL391[4:0]	IRCn_IRQVA391
392	Base Timer ch.27	IRCn_IRQPL98:IRQPL392[4:0]	IRCn_IRQVA392
393	Base Timer ch.28	IRCn_IRQPL98:IRQPL393[4:0]	IRCn_IRQVA393
394	Base Timer ch.29	IRCn_IRQPL98:IRQPL394[4:0]	IRCn_IRQVA394
395	A/D unit.0 Scan Conversion End	IRCn_IRQPL98:IRQPL395[4:0]	IRCn_IRQVA395
396	A/D unit.0 A/D Conversion End	IRCn_IRQPL99:IRQPL396[4:0]	IRCn_IRQVA396
397	A/D unit.0 Range Compare	IRCn_IRQPL99:IRQPL397[4:0]	IRCn_IRQVA397
398	A/D unit.1 Scan Conversion End	IRCn_IRQPL99:IRQPL398[4:0]	IRCn_IRQVA398
399	A/D unit.1 A/D Conversion End	IRCn_IRQPL99:IRQPL399[4:0]	IRCn_IRQVA399
400	A/D unit.1 Range Compare	IRCn_IRQPL100:IRQPL400 [4:0]	IRCn_IRQVA400
401	Global Timer Compare Clear	IRCn_IRQPL100:IRQPL401 [4:0]	IRCn_IRQVA401
402 to 403	Reserved		
404	Sound Generator ch.0	IRCn_IRQPL101:IRQPL404 [4:0]	IRCn_IRQVA404
IRQ No.	**Detail**	**IRQ Priority Register**	**Vector Address Register**
405	Sound Generator ch.1	IRCn_IRQPL101:IRQPL405 [4:0]	IRCn_IRQVA405
406	Sound Generator ch.2	IRCn_IRQPL101:IRQPL406 [4:0]	IRCn_IRQVA406
407 to 511	Reserved		

Notes:

- *Please refer to 'chapter INTERRUPT CONTROLLER' for more details of IRQ Priority Register and Vector Address Register.*
- *SHE OFF product is prohibited to use Interrupt for SHE function.*

2. NMI Map (S6J312xHzC*)

This section shows NMI map in S6J312xHzC*.

*x: A/9, z: A/B

No	Detail	NMI Priority Register	Vector Address Register
0	NMIX pin (Ext-IRC)	IRCn_NMIPL0:NMIPL0[3:0]	IRCn_NMIVA0
1	Reserved		
2	Reserved		
3	Reserved		
4	LVDs IRQ	IRCn_NMIPL1:NMIPL4[3:0]	IRCn_NMIVA4
5	CSV, Profile	IRCn_NMIPL1:NMIPL5[3:0]	IRCn_NMIVA5
6	HW-WDT	IRCn_NMIPL1:NMIPL6[3:0]	IRCn_NMIVA6
7	SW-WDT	IRCn_NMIPL1:NMIPL7[3:0]	IRCn_NMIVA7
8	IRC 2-bit ECC error detection	IRCn_NMIPL2:NMIPL8[3:0]	IRCn_NMIVA8
9	Reserved		
10	Reserved		
11	Backup RAM 2-bit ECC error detection	IRCn_NMIPL2:NMIPL11[3:0]	IRCn_NMIVA11
12	CAN-FD RAMs 2-bit ECC error detection	IRCn_NMIPL3:NMIPL12[3:0]	IRCn_NMIVA12
13	DMAC MPU #0 protection violation	IRCn_NMIPL3:NMIPL13[3:0]	IRCn_NMIVA13
14	Reserved		
15	SHE MPU	IRCn_NMIPL3:NMIPL15[3:0]	IRCn_NMIVA15
16	Reserved		
17	Reserved		
18	TPU(s) protection violation	IRCn_NMIPL4:NMIPL18[3:0]	IRCn_NMIVA18
19 to 31	Reserved		

Notes:

- *Please refer to ‘chapter INTERRUPT CONTROLLER' for more details of NMI Priority Register and Vector Address Register.*
- *SHE OFF product is prohibited to use NMI for SHE function.*

CHAPTER 67: APPENDIX: Major Changes

Page	Section	Change Results
-	Cover	Added the full production names including SHE-OFF series as follows (Correct) S6J3120 Series 32-bit Microcontroller Spansion® TraveoTM Family S6J312AHAB/S6J3129HAB/S6J3128HAB S6J312AHBB/S6J3129HBB/S6J3128HBB HARDWARE MANUAL
-	Cover	Corrected information about Spansion to Cypress.
47	CHAPTER 0:Introduction	The item should be added as indicated by the shading below. (Correct) 30.About the 12-Bit A/D Converter 31.About the Secure Hardware Extension (SHE) 32.About the Partial Wakeup Control 33.Common items
49	CHAPTER 0:Introduction 2. About Clock System Specifications	Features of "For details" should be corrected as indicated by the shading below 2. About Clock System Specifications (Error) About PLL0 and SSCG PLL0 settings at the time of main clock 4 MHz input. The input frequency supplied to PLL must be divided by 1 (4 MHz). When CPUs operate at 112 MHz, usage by multiplying by 112 (PLLout=448 MHz) is recommended. When CPUs operate at 80 MHz, usage by multiplying by 120 (PLLout=480 MHz) is recommended. (Correct) About PLL0 and SSCG PLL0 settings at the time of main clock 4 MHz input. The input frequency supplied to PLL must be divided by 1 (4 MHz). When CPUs operate at 128 MHz, usage by multiplying by 128 (PLLout=512 MHz) is recommended. When CPUs operate at 80 MHz, usage by multiplying by 120 (PLLout=480 MHz) is recommended.

Page	Section	Change Results
49	CHAPTER 0:Introduction 2. About Clock System Specifications	Added item as follows: 5.1.3 PLL/SSCG stabilization time control register Setting the value 1000 to the PLLSTABS and SSCGSTABS bits is prohibited.
50	CHAPTER 0:Introduction 3. About Low-Power Consumption Specifications	Features of "For details" should be deleted as indicated by the shading below. −3 V I/O −Partial Wakeup
50	CHAPTER 0:Introduction 3. About Low-Power Consumption Specifications	Added the following description in features of “For details”. This model does not support IRQ_PW bit in MCG_IRSR0 register. This bit is always read to 0.
50	CHAPTER 0:Introduction 4.About Low-Voltage Detection Specifications	Features of "For details" should be deleted as indicated by the shading below (Error) The low-voltage detection voltage and the detection error are desired values.
51	CHAPTER 0:Introduction 7.About the Real Time Clock	Features of "For details" should be deleted as indicated by the shading below This model does not support Partial Wakeup.
52	CHAPTER 0:Introduction 9. About the Multi-function Serial Interface	Features of "For details" should be corrected as indicated by the shading below. (Error) This model does not support the I^2C communication function. It also does not support registers related to I^2C. (Correct) This model supports the I^2C communication function. - This model does not support high-speed mode. (Max. 400kbps) - This model does not support Min. IOL = 3mA with VOL = 0.4V. - At Multi-function serial ch.1, ch.2, and ch.12, I2C-UART function is not supported.

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>52</td><td>CHAPTER 0:Introduction
9. About the Multi-function Serial Interface</td><td>Features of "For details" should be corrected as indicated by the shading below.
(Correct)

■The maximum frequency of a peripheral clock is 32 MHz in UART (asynchronous serial interface) operation mode.
■Hardware flow control is not supported in UART (asynchronous serial interface) operation mode.
■The maximum frequency of a peripheral clock is 32 MHz in CSIO (clock synchronous serial interface) operation mode.
■The maximum frequency of a peripheral clock is 32 MHz in LIN interface (v2.1) (LIN communication control interface (v2.1)) operation mode.</td></tr>
<tr><td>52</td><td>CHAPTER 0:Introduction
9. About the Multi-function Serial Interface</td><td>Added the following description.

When reset is issued, the following ports become "L" output as the initial state.
- SCS0_0 /SIN4_1 (assigned to P023)
- SIN1_0 (assigned to P028)
- SOT1_0 (assigned to P029)
- SCS1_0 (assigned to P031)
- SCK1_0 (assigned to P100)
- SIN2_0 (assigned to P408)
- SOT2_0 (assigned to P409)
- SOT4_1 (assigned to P024)
- SCS42_1 (assigned to P027)
- SCS43_1 (assigned to P030)
- SOT12_0 (assigned to P406)
- SCK12_0 /SCK10_1 (assigned to P407)
Take care of the above port while used as UART/CSIO/LIN.
When reset is issued, the communication may be disturbed by "L" output.
Regarding the initial state for each ports, refer to the data sheet "11.PIN STATUSES IN CPU STATUS".</td></tr>
<tr><td>55</td><td>CHAPTER 0:Introduction
16. About TCFLASH</td><td>Added the table of full production as follows:

■ This model supports TCFLASH memory sizes are as follows.

<table><tr><th>Part number</th><th>TCFlash size (Byte)</th></tr><tr><td>S6J312AHzB*</td><td>1024K + 64K</td></tr><tr><td>S6J3129HzB*</td><td>768K + 64K</td></tr><tr><td>S6J3128HzB*</td><td>512K + 64K</td></tr></table>* z: A / B</td></tr>
<tr><td>58</td><td>CHAPTER 0:Introduction
28. About the LCD Controller</td><td>Added the table of full production as follows:
(Correct)
■When reset is issued, the following ports become "L" output as the initial state.
−P019, P023, P024, P027, P028, P029, P030, P031, P100, P406, P407, P408, P409
Therefore, don't add the Pull-up registers outside of this product to the above ports.
In case that the above ports are used as Pull-up, use the Pull-up function implemented in this product.</td></tr>
</table>

Page	Section	Change Results
58	CHAPTER 0:Introduction 30.About the 12-Bit A/D Converter	The item should be deleted as indicated by the shading below. (Correct) ■The following numbers are not supported as activation channels in this model. Channel No. 0, 1, 2, 4, 7, 8, 11, 16, 25, 26, 44, 45, 54, 63 ■This model does not support Partial Wakeup.
58	CHAPTER 0:Introduction 31.About the Secure Hardware Extension (SHE)	Features of "For details" should be added as indicated by the shading below. 31.About the Secure Hardware Extension (SHE)23.1. How to use Reset factor register This section explains restrictions on the operation specifications for the Secure Hardware Extension (SHE) Regarding S6J312xHBB*, The models are not supported SHE function. * x: A/9/8
58	CHAPTER 0:Introduction 32.About the Partial Wakeup Control	Features of "For details" should be added as indicated by the shading below. 32.About the Partial Wakeup Control This section explains restrictions on the operation specifications for Partial Wakeup Control This model does not support the following analog channels for partial wakeup. AN44(PWU_AN4), AN45(PWU_AN5)
64	CHAPTER 1:Overview 2.1. Cortex-R5 Core	Features of "For details" should be added as indicated by the shading below (Correct) ECC support for the TCM ports for RAM
65	CHAPTER 1:Overview 2. Features 2.2. Peripheral Functions	Features of "For details" should be corrected as indicated by the shading below. Built-in Flash memory size −Program: 1024 K + 64 KB (S6J312AHzB*)/768 K + 64 KB (S6J3129HzB*)/512 K + 64 KB (S6J3128HzB*) *z: A/B −Work: 112 KB (S6J312AHzB*)/ 112 KB (S6J3129HzB*)/112 KB (S6J3128HzB*) *z: A/B
65	CHAPTER 1:Overview 2. Features 2.2. Peripheral Functions	Features of "For details" should be corrected as indicated by the shading below. Built-in RAM size −TCRAM 64 KB(S6J312AHzB*)/ 48 KB(S6J3129HzB*)/32 KB(S6J3128HzB*) −System SRAM 16 KB (S6J312AHzB*)/ 16 KB (S6J3129HzB*)/ 16 KB (S6J3128HzB*) −Backup RAM 8 KB (S6J312AHzB*)/ 8 KB (S6J3129HzB*)/8 KB (S6J3128HzB*) *z: A/B
65	CHAPTER 1:Overview 2. Features 2.2. Peripheral Functions	Features of "For details" should be corrected as indicated by the shading below. General-purpose ports: 112 channels (S6J312AHzB*)/ 112 channels (S6J3129HzB*)/ 112 channels (S6J3128HzB*)

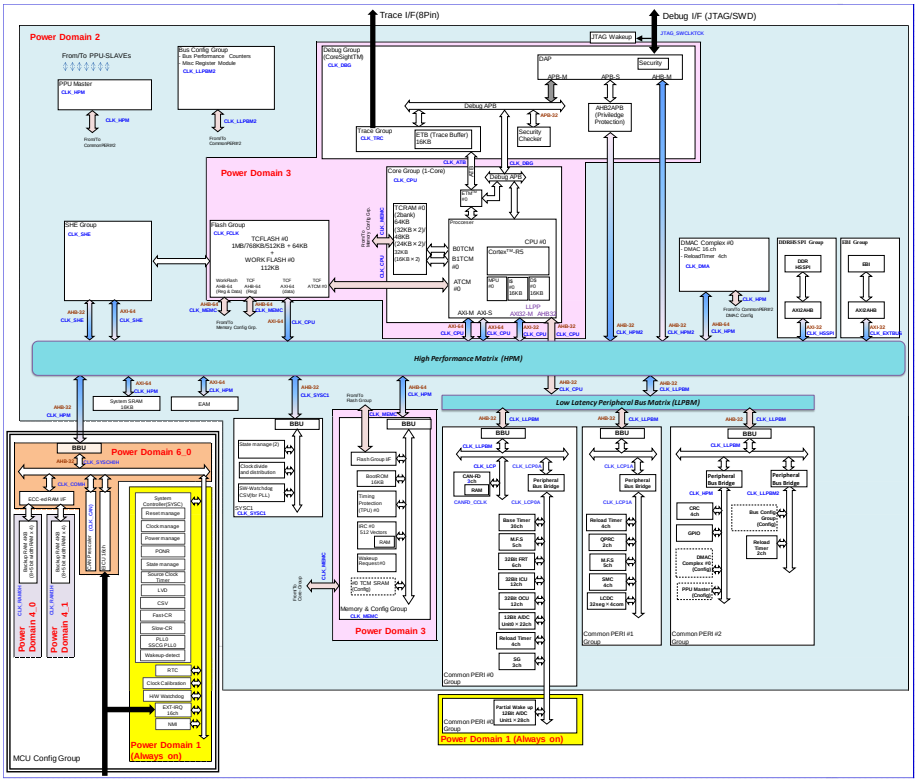
Page	Section	Change Results
65	CHAPTER 1:Overview 2. Features 2.2. Peripheral Functions	Features of "For details" should be corrected as indicated by the shading below. A/D converter (successive approximation type) −12-bit resolution, 2 units mounted: Max 50 channels (22 channels + 28 channels)(S6J312AHzB*)/ Max 50 channels (22 channels + 28 channels)(S6J3129HzB*)/ Max 50 channels (22 channels + 28 channels)(S6J3128HzB*) *z: A/B
65	CHAPTER 1:Overview 2. Features 2.2. Peripheral Functions	Features of "For details" should be corrected as indicated by the shading below. Multi-function serial (transmission and reception FIFOs mounted) :Max 10 channels(S6J312AHzB*)/ Max 10 channels(S6J3129HzB*)/ Max 10 channels(S6J3128HzB*) *z: A/B
65	CHAPTER 1:Overview 2. Features 2.2. Peripheral Functions	Added <I^2C>as follows: <I^2C> −Full duplex, double buffering system; 64-byte transmission FIFO, 64-byte reception FIFO −Standard mode (Max. 100kbps) is supported only. −DMA transfer is supported.
66	CHAPTER 1:Overview 2. Features 2.2. Peripheral Functions	Features of "For details" should be corrected as indicated by the shading below. (Error) −CAN transfer speed :1Mbps (Correct) −CAN transfer speed :5Mbps
67	CHAPTER 1:Overview 2.2. Peripheral Functions	Features of "For details" should be added as indicated by the shading below (Error) Low-power consumption control − Standby function − Power-off function (Correct) Low-power consumption control − Standby function − Power-off function − Partial wakeup function
68	CHAPTER 1:Overview 2.2. Peripheral Functions	Revised the full production as follows: - Package: LEU144 (S6J312xHzB*) - *x:A/9/8, z: A/B

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>69,</td><td>CHAPTER 1:Overview 3.Product Lineup</td><td>
Features of "For details" should be corrected as indicated by the shading below.
Table 3 1 Memory Size
<table>
<tr><th colspan="2"></th><th>S6J312AHzB*</th><th>S6J3129HzB*</th><th>S6J3128HzB*[1]</th></tr>
<tr><td rowspan="2">Flash</td><td>Program</td><td>1024K bytes
+
Small sector (8 KB x 8)</td><td>768K bytes
+
Small sector (8 KB x 8)</td><td>512K bytes
+
Small sector (8 KB x 8)</td></tr>
<tr><td>Work</td><td>112K bytes</td><td>112K bytes</td><td>112K bytes</td></tr>
<tr><td rowspan="3">RAM</td><td>TCRAM</td><td>64K bytes</td><td>48K bytes</td><td>32K bytes</td></tr>
<tr><td>System SRAM</td><td>16K bytes</td><td>16K bytes</td><td>16K bytes</td></tr>
<tr><td>Backup RAM</td><td>8K bytes</td><td>8K bytes</td><td>8K bytes</td></tr>
</table>
*z: A/B
</td></tr>
<tr><td>69</td><td>CHAPTER 1:Overview 3.Product Lineup</td><td>
Features of "For details" should be added as indicated by the shading below.
Table 3-2 SHE Option
<table>
<tr><th></th><th>S6J312xHAB*</th><th>S6J312xHBB*</th></tr>
<tr><td>Security (SHE)</td><td>ON</td><td>OFF</td></tr>
</table>
* x: A/9/8
</td></tr>
<tr><td>69, 70</td><td>CHAPTER 1:Overview 3.Product Lineup</td><td>
Features of "For details" should be corrected as indicated by the shading below.

Table 3-3: Product Lineup
<table>
<tr><th></th><th>S6J312xHzB*</th></tr>
<tr><td>Maximum CPU operating frequency</td><td>128 MHz</td></tr>
<tr><td>Multi-function serial</td><td>10 channels*2</td></tr>
<tr><td>Low-power consumption mode</td><td>Standby function
Power-off function
Partial wakeup function</td></tr>
</table>
*1: x: A/9/8, z: A/B
*2: I2C-UART function is not supported at Multi-function serial ch.1, ch.2, and ch.12.
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>71,
72</td><td>CHAPTER 1:Overview
4. Function Overview</td><td>
Features of "For details" should be corrected as indicated by the shading below.(Only correct items)
Table 4-1: Functional Overview
(error)
<table>
<tr><td>Function</td><td>Features</td></tr>
<tr><td>Flash interface</td><td>1 wait (target) at operating frequency of 112 MHz</td></tr>
<tr><td>Multi-function serial</td><td>3 functions are available: UART, CSIO, and LIN-UART.
…</td></tr>
<tr><td>CAN-FD Interface</td><td>A communication rate supportsb1 Mbps</td></tr>
<tr><td>Low-power consumption control</td><td>Power-off function is mounted.
Standby function is mounted.</td></tr>
</table>
(correct)
<table>
<tr><td>Function</td><td>Features</td></tr>
<tr><td>Flash interface</td><td>1 wait (target) at operating frequency of 128 MHz</td></tr>
<tr><td>Multi-function serial</td><td>4 functions are available: UART, CSIO, LIN-UART and I²C.
…

Support for 100kbps I²C only</td></tr>
<tr><td>CAN-FD Interface</td><td>A communication rate supports 5 Mbps</td></tr>
<tr><td>Low-power consumption control</td><td>Power-off function is mounted.
Standby function is mounted.
Partial wakeup function is mounted.</td></tr>
</table>
</td></tr>
</table>

Page	Section	Change Results
74	CHAPTER 1:Overview 5.Block Diagrams	Revised the tables as follows for full production. (Correct) Table 5-1 S6J312xHzB* Pin Functions  * x: A/9/8, z: A/B

Page	Section	Change Results
75	CHAPTER 1:Overview 6. Memory Map	Revised "Figure 6-1 Memory Map" as follows (Correct)

Figure 6-1 Memory Map

ADDRESS START	ADDRESS END	group	S6J312AHzB* part	S6J3129HzB* part	S6J3128HzB* part
0x0000_0000	0x0000_7FFF	Internal area for CR5 Complex	TCRAM (Main 64KByte)	TCRAM (Main 48KByte)	TCRAM (Main 32KByte)
0x0000_8000	0x0000_BFFF				Reserved
0x0000_C000	0x0000_FFFF			Reserved	Reserved
0x0001_0000	0x007F_FFFF		Reserved	Reserved	Reserved
0x0080_0000	0x009E_FFFF		Reserved	Reserved	Reserved
0x009F_0000	0x009F_FFFF		TCM_FLASH (Small Sector 8KByte×8)	TCM_FLASH (Small Sector 8KByte×8)	TCM_FLASH (Small Sector 8KByte×8)
0x00A0_0000	0x00A7_FFFF		TCM_FLASH (Code 1MByte)	TCM_FLASH (Code 768KByte)	TCM_FLASH (Code 512KByte)
0x00A8_0000	0x00AB_FFFF				Reserved
0x00AC_0000	0x00AF_FFFF			Reserved	Reserved
0x00B0_0000	0x00DF_FFFF		Reserved	Reserved	Reserved
0x00E0_0000	0x00FF_FFFF		Reserved	Reserved	Reserved
0x0100_0000	0x019E_FFFF		Reserved	Reserved	Reserved
0x019F_0000	0x019F_FFFF		AXI_FLASH_MEMORY (Small Sector 8KByte×8 *Mirror)	AXI_FLASH_MEMORY (Small Sector 8KByte×8 *Mirror)	AXI_FLASH_MEMORY (Small Sector 8KByte×8 *Mirror)
0x01A0_0000	0x01A7_FFFF		AXI_FLASH_MEMORY (Code 1MByte *Mirror)	AXI_FLASH_MEMORY (Code 768KByte *Mirror)	AXI_FLASH_MEMORY (Code 512KByte *Mirror)
0x01A8_0000	0x01AB_FFFF				Reserved
0x01AC_0000	0x01AF_FFFF			Reserved	Reserved
0x01B0_0000	0x01DF_FFFF		Reserved	Reserved	Reserved
0x01E0_0000	0x01FF_FFFF		Reserved	Reserved	Reserved
0x0200_0000	0x0200_3FFF	Shared Flash and memory area	SYSTEM SRAM (16KByte)	SYSTEM SRAM (16KByte)	SYSTEM SRAM (16KByte)
0x0200_4000	0x0203_FFFF		Reserved	Reserved	Reserved
0x0204_0000	0x027F_FFFF		Reserved	Reserved	Reserved
0x0280_0000	0x0280_0FFF		Exclusive Access Memory	Exclusive Access Memory	Exclusive Access Memory
0x0280_1000	0x03FF_FFFF		Reserved	Reserved	Reserved
0x0400_0000	0x05FF_FFFF		AXI_SLAVE_CORE0	AXI_SLAVE_CORE0	AXI_SLAVE_CORE0
0x0600_0000	0x0DFF_FFFF		Reserved	Reserved	Reserved
0x0E00_0000	0x0E01_BFFF		WORK_FLASH (112KByte mirror area 1)	WORK_FLASH (112KByte mirror area 1)	WORK_FLASH (112KByte mirror area 1)
0x0E01_C000	0x0E0F_FFFF		Reserved	Reserved	Reserved
0x0E10_0000	0x0E1F_FFFF		Reserved	Reserved	Reserved
0x0E20_0000	0x0E21_BFFF		WORK_FLASH (112KByte mirror area 3)	WORK_FLASH (112KByte mirror area 3)	WORK_FLASH (112KByte mirror area 3)
0x0E21_C000	0x0E2F_FFFF		Reserved	Reserved	Reserved
0x0E30_0000	0x0E31_BFFF		WORK_FLASH (112KByte mirror area 4)	WORK_FLASH (112KByte mirror area 4)	WORK_FLASH (112KByte mirror area 4)
0x0E31_C000	0x0E3F_FFFF		Reserved	Reserved	Reserved
0x0E40_0000	0x0E7F_FFFF		Reserved	Reserved	Reserved
0x0E80_0000	0x0E80_1FFF		Backup RAM 8KByte	Backup RAM 8KByte	Backup RAM 8KByte
0x0E80_2000	0x0E80_FFFF		Reserved	Reserved	Reserved
0x0E81_0000	0x0E87_FFFF		Reserved	Reserved	Reserved
0x0E88_0000	0x0FFF_FFFF		Reserved	Reserved	Reserved
0x1000_0000	0x1FFF_FFFF	External bus area	EBI_MEMORY(SRAM/FLASH)	EBI_MEMORY(SRAM/FLASH)	EBI_MEMORY(SRAM/FLASH)
0x2000_0000	0x7FFF_FFFF	Reserved	Reserved	Reserved	Reserved
0x8000_0000	0x8FFF_FFFF	HSSPI memory area	HSSPI0_MEMORY	HSSPI0_MEMORY	HSSPI0_MEMORY
0x9000_0000	0xAFFF_FFFF	Reserved	Reserved	Reserved	Reserved
0xB000_0000	0xB483_FFFF	Peri area	Peri_area	Peri_area	Peri_area
0xB484_0000	0xB484_FFFF		APPS#5	APPS#5	APPS#5
0xB485_0000	0xB48B_FFFF		Peri_area	Peri_area	Peri_area
0xB48C_0000	0xB48C_FFFF		APPS#7	APPS#7	APPS#7
0xB48D_0000	0xB7FF_FFFF		Peri_area	Peri_area	Peri_area
0xB800_0000	0xBFFF_FFFF				
0xC000_0000	0xFFFE_DFFF	Reserved	Reserved	Reserved	Reserved
0xFFFE_E000	0xFFFE_FFFF	Error Config	ERRCFG	ERRCFG	ERRCFG
0xFFFF_0000	0xFFFF_3FFF	BootROM	BootRom	BootRom	BootRom
0xFFFF_4000	0xFFFF_FFFF		Reserved	Reserved	Reserved

*z: A/B

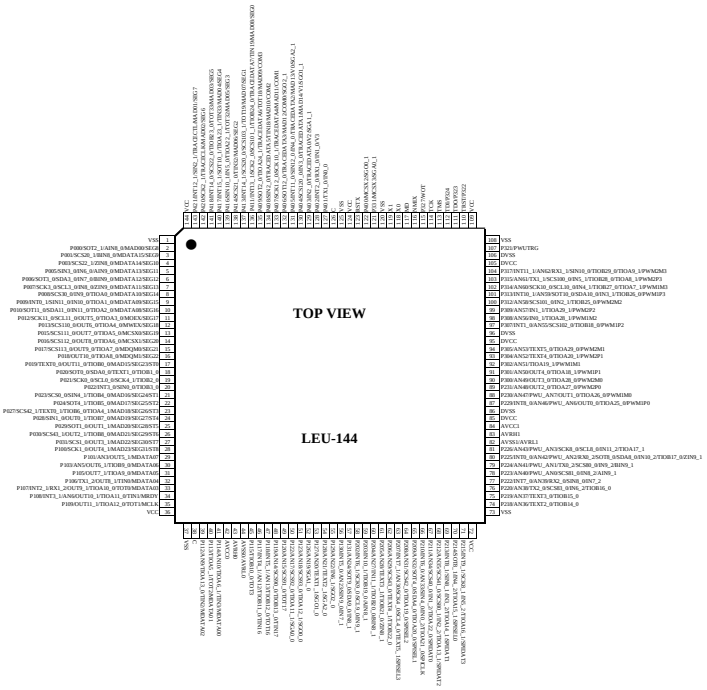
<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>76</td><td>CHAPTER 1:Overview 6. Memory Map</td><td>Features of "For details" should be added as indicated by the shading below.

(error)
3.WORKFLASH (0x0E00_0000 -), WORKFLASH (0x0E20_0000 -), and WORKFLASH (0x0E30_0000 -)

(correct)
3.WORKFLASH (0x0E00_0000 -), WORKFLASH (0x0E20_0000 -), and WORKFLASH (0x0E30_0000 -)

The ECC movement in TCM port is based on ECC setting inside the CPU.</td></tr>
<tr><td>78</td><td>CHAPTER 1:Overview 6. Memory Map</td><td>Features of "For details" should be corrected as indicated by the shading below.

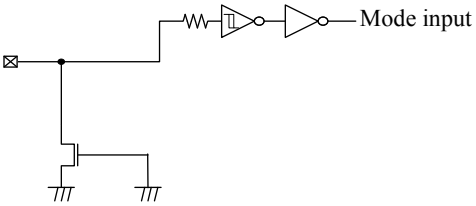
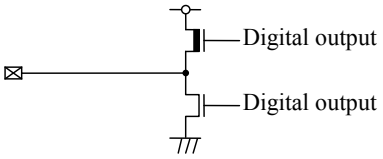
(error)
Figure 6-2 S6J312AHAA Peripheral Map
(correct)
Figure 6-2 S6J312xHzB* Peripheral Map *x: A/9/8 z: A/B</td></tr>
<tr><td>81</td><td>CHAPTER 1:Overview 6.Memory Map</td><td>Added “Partial Wake Up” to address of “B484_8400 to B484_87FF”.
(Correct)<table><tr><td>B484_8400</td><td>B484_87FF</td><td>APPS #5</td><td>A/D unit1 , Partial Wake Up</td><td>297</td></tr></table></td></tr>
<tr><td>81</td><td>CHAPTER 1:Overview 6.Memory Map</td><td>Features of "For details" should be added as indicated by the shading below
(Correct)<table><tr><td>B484_8C00</td><td>B484_8FFF</td><td></td><td>Reserved</td><td>-</td></tr><tr><td>B484_9000</td><td>B484_93FF</td><td>APPS #5</td><td>Global Timer</td><td>300</td></tr><tr><td>B484_9400</td><td>B484_FFFF</td><td></td><td>Reserved</td><td>-</td></tr></table></td></tr>
<tr><td>81</td><td>CHAPTER 1:Overview 6.Memory Map</td><td>Revised the memory map of APPS#7 as follows.
(Correct)<table><tr><td>START Address</td><td>END Address</td><td></td><td>Function</td><td>PPU No</td></tr><tr><td>B48C_0000</td><td>B48C_3FFF</td><td></td><td>Reserved</td><td>-</td></tr><tr><td>B48C_4000</td><td>B48C_43FF</td><td>APPS #7</td><td>Stepper Motor Control ch.0</td><td>317</td></tr><tr><td>B48C_4400</td><td>B48C_47FF</td><td>APPS #7</td><td>Stepper Motor Control ch.1</td><td>318</td></tr><tr><td>B48C_4800</td><td>B48C_4BFF</td><td>APPS #7</td><td>Stepper Motor Control ch.2</td><td>319</td></tr><tr><td>B48C_4C00</td><td>B48C_4FFF</td><td>APPS #7</td><td>Stepper Motor Control ch.3</td><td>320</td></tr><tr><td>B48C_5000</td><td>B48C_57FF</td><td></td><td>Reserved</td><td>-</td></tr><tr><td>B48C_5800</td><td>B48C_5BFF</td><td>APPS #7</td><td>SMC Trigger Generator</td><td>323</td></tr><tr><td>B48C_5C00</td><td>B48C_5FFF</td><td>APPS #7</td><td>Liquid Crystal Display Controller</td><td>324</td></tr><tr><td>B48C_6000</td><td>B48C_63FF</td><td>APPS #7</td><td>Liquid Crystal Display input/output control</td><td>325</td></tr><tr><td>B48C_6400</td><td>B48C_FFFF</td><td></td><td>Reserved</td><td>-</td></tr></table></td></tr>
</table>

Page	Section	Change Results
81, 82	CHAPTER 1:Overview 6.Memory Map	The following comments and restriction should be added under the table of “APPS#5 area” (Correct) When MPU attribute of Cortex-R5 is configured as "Normal", store buffer inside Cortex-R5 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]* MPU attribute "Device" or "Strongly Ordered" is required for accesses to areas below, in particular situation. – *FLASH Memory (when writing commands)* SHE OFF product is prohibited to access SHE area (B200_0000 to B20F_FFFF)
86	CHAPTER 1:Overview 8. Pin Assignment	Features of "For details" should be corrected as indicated by the shading below. (Correct) Figure 8-1 Pin Assignment for S6J312xHzB* * x: A/9/8, z: A/B 
88	CHAPTER 1:Overview 10. List of Pin Functions	Revised the tables as follows for full production. (Correct) Table 10-1 S6J311xJzB* Pin Functions * x: E/D/C/B, z: A/B

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>86
to
91,
93
to
97,
99,
102</td><td>CHAPTER 1:Overview
10. List of Pin Functions</td><td>Revised the “I/O pin” to ”output pin” as follows
(Correct)
Pin 13 Multi-function serial ch.11 serial chip select 1 output pin (0)
Pin14 Multi-function serial ch.11 serial chip select 2 output pin (0)
Pin 15 Multi-function serial ch.11 serial chip select 3 output pin (0)
Pin 23 Multi-function serial ch.4 serial chip select 2 output pin (1)
Pin 26 Multi-function serial ch.4 serial chip select 3 output pin (1)
Pin 49 Multi-function serial ch.9 serial chip select 1 output pin (0)
Pin 50 Multi-function serial ch.9 serial chip select 2 output pin (0)
Pin 51 Multi-function serial ch.9 serial chip select 3 output pin (0)
Pin 62 Multi-function serial ch.4 serial chip select 3 output pin (0)
Pin 64 Multi-function serial ch.4 serial chip select 2 output pin (0)
Pin 68 Multi-function serial ch.4 serial chip select 1 output pin (0)
Pin 76 Multi-function serial ch.8 serial chip select 3 output pin (0)
Pin 78 Multi-function serial ch.8 serial chip select 1 output pin (0)
Pin 97 Multi-function serial ch.10 serial chip select 2 output pin (0)
Pin 100 Multi-function serial ch.10 serial chip select 1 output pin (0)
Pin 136 Multi-function serial ch.10 serial chip select 1 output pin (1)
Pin 137 Multi-function serial ch.10 serial chip select 3 output pin (1)</td></tr>
<tr><td>98</td><td>CHAPTER 1:Overview
10. List of Pin Functions</td><td>Deleted Pin87 RX0_1 from “Table 5-1 S6J312xHzB* Pin Functions”
(Error)
<table>
<tr><td rowspan="8">87</td><td>P229</td><td>-</td><td rowspan="8">M</td><td>General-purpose I/O port</td></tr>
<tr><td>INT8_0</td><td>-</td><td>INT8 external interrupt input pin (0)</td></tr>
<tr><td>AN46</td><td>-</td><td>ADC analog 46 input pin</td></tr>
<tr><td>PWU_AN6</td><td>-</td><td>Partial wakeup ADC analog 6 input pin</td></tr>
<tr><td>RX0_1</td><td>-</td><td>CAN reception data 0 input pin (1)</td></tr>
<tr><td>OUT0_0</td><td>-</td><td>Output compare ch.0 output pin (0)</td></tr>
<tr><td>TIOA25_0</td><td>-</td><td>Base timer ch.25 TIOA I/O pin (0)</td></tr>
<tr><td>PWM1P0</td><td>-</td><td>SMC ch.0 (P1) output pin</td></tr>
</table>
(Correct)
<table>
<tr><td rowspan="8">87</td><td>P229</td><td>-</td><td rowspan="8">M</td><td>General-purpose I/O port</td></tr>
<tr><td>INT8_0</td><td>-</td><td>INT8 external interrupt input pin (0)</td></tr>
<tr><td>AN46</td><td>-</td><td>ADC analog 46 input pin</td></tr>
<tr><td>PWU_AN6</td><td>-</td><td>Partial wakeup ADC analog 6 input pin</td></tr>
<tr><td>OUT0_0</td><td>-</td><td>Output compare ch.0 output pin (0)</td></tr>
<tr><td>TIOA25_0</td><td>-</td><td>Base timer ch.25 TIOA I/O pin (0)</td></tr>
<tr><td>PWM1P0</td><td>-</td><td>SMC ch.0 (P1) output pin</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>99</td><td>CHAPTER 1:Overview 10. List of Pin Functions</td><td>Deleted Pin97 RX0_0 from “Table 5-1 S6J312xHzB* Pin Functions”
(Error)
<table><tr><td>97</td><td>P307
INT1_0
AN55
RX0_0
SCS102_0
TIOB18_0
PWM1P2</td><td>-
-
-
-
-
-
-</td><td>M</td><td>General-purpose I/O port
INT1 external interrupt input pin (0)
ADC analog 55 input pin
CAN reception data 0 input pin (0)
Multi-function serial ch.10 serial chip select 2 I/O pin (0)
Base timer ch.18 TIOB input pin (0)
SMC ch.2 (P1) output pin</td></tr></table>
(Correct)
<table><tr><td>97</td><td>P307
INT1_0
AN55
SCS102_0
TIOB18_0
PWM1P2</td><td>-
-
-
-
-
-</td><td>M</td><td>General-purpose I/O port
INT1 external interrupt input pin (0)
ADC analog 55 input pin
Multi-function serial ch.10 serial chip select 2 I/O pin (0)
Base timer ch.18 TIOB input pin (0)
SMC ch.2 (P1) output pin</td></tr></table></td></tr>
<tr><td>88 to 90, 94, 95, 97, 99</td><td>CHAPTER 1:Overview 10. List of Pin Functions</td><td>Added the I2C function to Pin 6,7,10,11,18,19,57,58,63,65,80,81,101,102

Pin 6 SDA3_0 I2C bus ch.3 serial data I/O pin
Pin 7 SCL3_0 I2C bus ch.3 serial clock I/O pin
Pin 10 SDA11_0 I2C bus ch.11 serial data I/O pin
Pin 11 SCL11_0 I2C bus ch.11 serial clock I/O pin
Pin 18 SDA0_0 I2C bus ch.0 serial data I/O pin
Pin 19 SCL0_0 I2C bus ch.0 serial clock I/O pin
Pin 57 SDA9_0 I2C bus ch.9 serial data I/O pin
Pin 58 SCL9_0 I2C bus ch.9 serial clock I/O pin
Pin 63 SDA4_0 I2C bus ch.4 serial data I/O pin
Pin 65 SCL4_0 I2C bus ch.4 serial clock I/O pin
Pin 80 SDA8_0 I2C bus ch.8 serial data I/O pin
Pin 81 SCL8_0 I2C bus ch.8 serial clock I/O pin
Pin 101 SDA10_0 I2C bus ch.10 serial data I/O pin
Pin 102 SCL10_0 I2C bus ch.10 serial clock I/O pin</td></tr>
<tr><td>97, 98</td><td>CHAPTER 1:Overview 10. List of Pin Functions</td><td>Added the Partial wakeup ADC analog input to Pin 78,79,80,81,87,88.
Pin78 PWU_AN0 Partial wakeup ADC analog 0 input pin
Pin79 PWU_AN1 Partial wakeup ADC analog 1 input pin
Pin80 PWU_AN2 Partial wakeup ADC analog 2 input pin
Pin81 PWU_AN3 Partial wakeup ADC analog 3 input pin
Pin87 PWU_AN6 Partial wakeup ADC analog 6 input pin
Pin88 PWU_AN7 Partial wakeup ADC analog 7 input pin</td></tr>
</table>

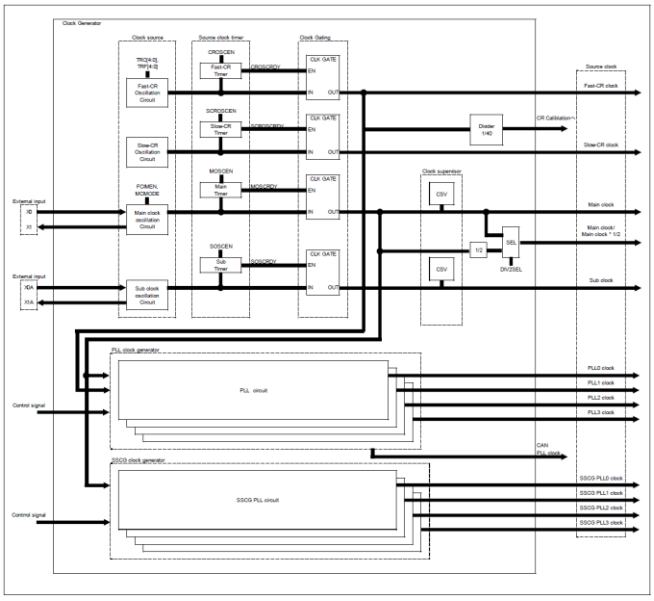
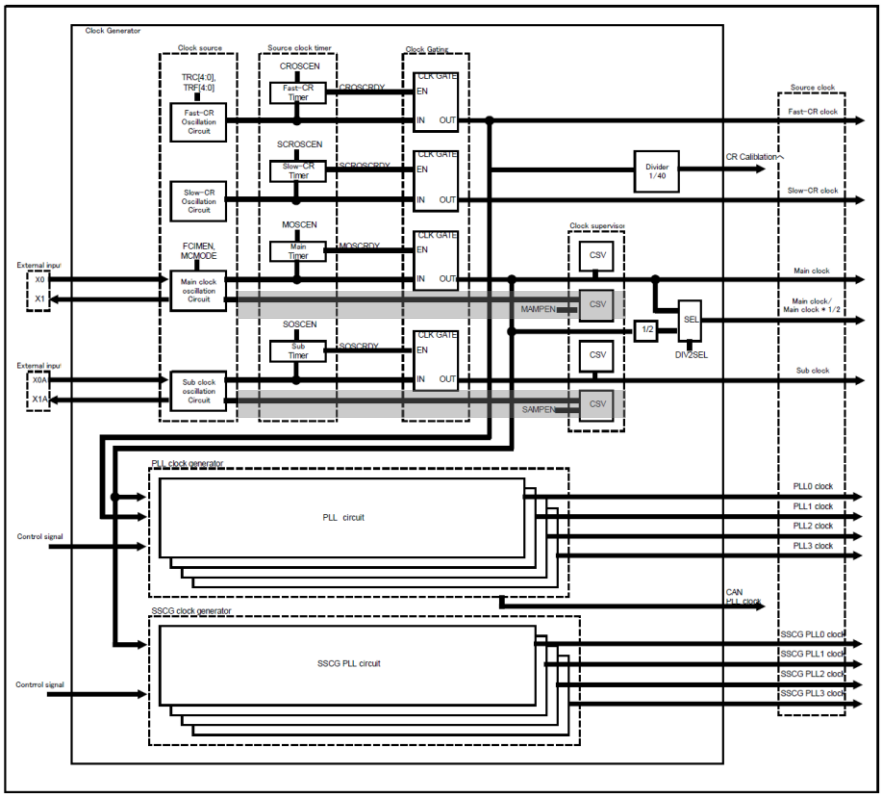
Page	Section	Change Results
100	CHAPTER 1:Overview 10. List of Pin Functions	Added the Partial wakeup trigger output to Pin 107. Pin107 PWUTRG Partial wakeup trigger output pin
101	CHAPTER 1:Overview 10. List of Pin Functions	Deleted the following function on each pins. Pin131 RX2_1 CAN reception data 2 output pin (1) Pin132 TX2_1 CAN transmission data 2 output pin (1)
105	CHAPTER 1:Overview 11. I/O Circuit Type	Revised Type C of “I/O Circuit Type” as follows: (Correct) 
108	CHAPTER 1:Overview 11. I/O Circuit Type	Added Type R to “11. I/O Circuit Type” R  Output of 2 mA

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>111</td><td>CHAPTER 2:CPU
2. Notes</td><td>Features of "For details" should be added as indicated by the shading below.

(Correct)
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.</td></tr>
<tr><td>115</td><td>CHAPTER 3:Operation Mode
2. Configuration</td><td>Revised the sentence as follows
(Error)
The following figure is a block diagram of the clock supervisor.
(Correct)
The following figure is a block diagram of the mode controller.</td></tr>
<tr><td>117</td><td>CHAPTER 3:Operation Mode
4. Register</td><td>Features of "For details" should be added as indicated by the shading below.

(Correct)
Table 4-2 List of register mirror area
<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></td></tr>
</table>

Page	Section	Change Results
200	CHAPTER 5:Clock System 1. Overview	Features of "For details" should be corrected as indicated by the shading below. (Error) ■PLL0 clock ■PLL1 clock (Correct) ■PLL0 clock ■SSCG PLL0 clock
200	CHAPTER 5: Clock System 1. Overview	Features of "For details" should be corrected as indicated by the shading below (Error) The clock domain are the following in the clock system. (Correct) The clock domains are the following in the clock system.
200	CHAPTER 5:Clock System 1.Overview	Features of "For details" should be corrected as indicated by the shading below (Error) The only 1 source clock is selected for each clock domain. Each clock domain use the selected source clock and divide to generate internal clock. (Correct) 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.
203	CHAPTER 5: Clock System 2.1. Clock System Configuration and Block Diagram	Features of "For details" should be modified as indicated by the shading below. (Error) ■Main clock/Main clock * 1/2 ■PLLx clock (x=0-1) (Correct) ■Main clock/Main clock * 1/2 ■PLLx clock (x=0) ■SSCG PLLx clock (x=0)

Page	Section	Change Results
204	CHAPTER 5:Clock System 2.2. Clock Generator Configuration and Block Diagram	Features of "For details" should be corrected as indicated by the shading below. Figure 2-2 Clock generator block diagram (Error)  (Correct) 

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>211</td><td>CHAPTER 5: Clock
System 3.1. Source Clock Generation</td><td>Features of "For details" should be deleted as indicated by the shading below.

(Error)
Source clock enable/disable
After each hard reset, the Fast-CR oscillator, Slow-CR oscillator, Main oscillator, PLL multiplier circuits are disabled
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

(Correct)
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.</td></tr>
<tr><td>211</td><td>CHAPTER 5: Clock
System 3.1. Source Clock Generation</td><td>Features of "For details" should be added as indicated by the shading below.

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>Fast-CR clock</td><td>Enable</td><td>Enable</td><td>Programmable</td></tr>
<tr><td>Slow-CR clock</td><td>Enable</td><td>Enable</td><td>Programmable</td></tr>
<tr><td>Main clock/Main clock * 1/2</td><td>Enable</td><td>Programmable</td><td>Programmable</td></tr>
<tr><td>PLL0 clock</td><td>Disable</td><td>Programmable</td><td>Programmable</td></tr>
<tr><td>SSCG PLL0 clock</td><td>Disable</td><td>Programmable</td><td>Programmable</td></tr>
</table></td></tr>
<tr><td>212</td><td>CHAPTER 5:Clock System
3.2. PLL/SSCG PLL Clock</td><td>Features of "For details" should be modified as indicated by the shading below.

(Error)
3.2. PLL clock
This section explains PLL clock.
Clock system has several PLL multiplier circuits which is used to generate the PLL clock out of the Fast-CR clock / Main clock.

(Correct)
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.</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>212</td><td>CHAPTER 5:Clock System 3.2. PLL/SSCG PLL Clock</td><td>Features of "For details" should be added as indicated by the shading below.

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.(SYSC_RUNSSCGxCNTR0 and SYSC_RUNSSCGxCNTR1/SYSC_PSSSSCGxCNTR and SYSC_PSSSSCGxCNTR)
3. Set the RUN/PSS profile register for SSCG PLLxEN (SYSC_RUNCKSRER/SYSC_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
6. stabilization time and then switches the clock source domain to SSCG PLL source.</td></tr>
<tr><td>212</td><td>CHAPTER 5:Clock System 3.2. PLL/SSCG PLL Clock</td><td>Features of "For details" should be modified as indicated by the shading below.

(Error)
Configuration of PLL clock stabilization time
For details refer to 'PLL stabilization time control register (SYSC_PLLSSCGSTCNTR)'.

(Correct)
Configuration of PLL/SSCG PLL clock stabilization time
For details refer to '5.1.3 PLL/SSCG stabilization time control register (SYSC_PLLSSCGSTCNTR)'.</td></tr>
<tr><td>213, 214</td><td>CHAPTER 5:Clock System 3.2. PLL/SSCG PLL Clock</td><td>Added Notes for Table 3 2 Example of PLL clock configuration and Table 3 3 Example of SSCG PLL clock configuration.</td></tr>
<tr><td>214</td><td>CHAPTER 5:Clock System 3.2. PLL/SSCG PLL Clock</td><td>Features of "For details" should be corrected as indicated by the shading below.
Table 3-2 Example of PLL Clock Configuration
<table>
<tr><th>Input Clock</th><th>Input Clock Divider Configuration</th><th>PLLin</th><th>PLL Multiplier Configuration</th><th>PLLout</th><th>Output Divider Configuration</th><th>PLL Clock</th></tr>
<tr><td>4MHz</td><td>1</td><td>4MHz</td><td>128</td><td>512MHz</td><td>4</td><td>128MHz</td></tr>
<tr><td>4MHz</td><td>1</td><td>4MHz</td><td>120</td><td>480MHz</td><td>6</td><td>80MHz</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>214</td><td>CHAPTER 5:Clock System 3.2. PLL/SSCG PLL Clock</td><td>
Features of "For details" should be corrected as indicated by the shading below.

Multiplying configuration of SSCG PLL clock
Table 3 3 Example of SSCG PLL clock configuration
<table>
<tr><th>Input clock</th><th>Input clock divider configuration</th><th>SSCG PLLin</th><th>SSCG PLL Multiplier configuration</th><th>SSCG PLLout</th><th>Output divider configuration</th><th>SSCG PLL clock</th></tr>
<tr><td>4MHz</td><td>1</td><td>4MHz</td><td>128</td><td>512MHz</td><td>4</td><td>128MHz</td></tr>
<tr><td>4MHz</td><td>1</td><td>4MHz</td><td>120</td><td>480MHz</td><td>6</td><td>80MHz</td></tr>
</table>

Note:
− Make the input clock frequency division setting so that the SSCG PLL input clock (SSCG PLLin) is at 4 MHz. Make the multiplication setting so that the SSCG PLL output clock (SSCG PLLout) is at 400 MHz to 512 MHz.
− The maximum SSCG PLL clock frequency must be 128MHz.
Output divider configuration can be set by the following.
- SSCGDIVM bit in SYSC0_RUNSSCG0CNTR0 register
- SSCGDIVM bit in SYSC0_PSSSSCG0CNTR0 register
(e.g. If SSCG PLLout is 512MHz, these settings must be configured as "multiply by 4" and over multiplication setting.)
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>216</td><td>CHAPTER 5: Clock System 3.3 Clock Distributer</td><td>
Features of "For details" should be corrected as indicated by the shading below
(a part of table 3-5)
(Error)
Table 3-5 Clock divider configuration and maximum frequency
<table>
<tr><th>Maximum Frequency</th></tr>
<tr><td>112MHz</td></tr>
<tr><td>Same as CLK_CPU</td></tr>
<tr><td>56MHz</td></tr>
<tr><td>56MHz</td></tr>
<tr><td>56MHz</td></tr>
<tr><td>28MHz</td></tr>
<tr><td>14MHz</td></tr>
<tr><td>28MHz</td></tr>
<tr><td>28MHz</td></tr>
<tr><td>25MHz</td></tr>
<tr><td>28MHz</td></tr>
</table>
(Correct)
Table 3-5 Clock Divider Configuration and Maximum Frequency
<table>
<tr><th>S6J312xHzB* Frequency</th></tr>
<tr><td>128MHz</td></tr>
<tr><td>Same as CLK_CPU</td></tr>
<tr><td>64MHz</td></tr>
<tr><td>64MHz</td></tr>
<tr><td>64MHz</td></tr>
<tr><td>32MHz</td></tr>
<tr><td>16MHz</td></tr>
<tr><td>32MHz</td></tr>
<tr><td>32MHz</td></tr>
<tr><td>25MHz</td></tr>
<tr><td>32MHz</td></tr>
</table>
* x: A/9/8, z: A/B
Note:
– The appendix reference for CLK_EXTBUS.
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>219</td><td>CHAPTER 5:Clock System
3.4. Clock Gear</td><td>Features of "For details" should be modified as indicated by the shading below.

(Error)
This section explains clock gear.
■Due to drastic frequency difference, switching from/to the main clock to/from PLL clock causes huge fluctuation of power supply current.

(Correct)
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.</td></tr>
<tr><td>226</td><td>CHAPTER 5:Clock System
4.1. RUN Configuration</td><td>The Note of "Figure 4-2" should be modified as indicated by the shading below.

(Error)
Note:
To use the 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.

(Correct)
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.</td></tr>
<tr><td>230</td><td>CHAPTER 5: Clock System
5.1.2. Main Oscillator Control Register(SYSC_MOSCCNTR)</td><td>"Data Attribute" should be corrected as indicated by the shading below.

(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>0</td><td colspan="3">000</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>232</td><td>CHAPTER 5:Clock System 5.1.3. PLL/SSCG Stabilization Time Control Register (SYSC_PLLSSCGSTCNTR)</td><td>
Features of "For details" should be added as indicated by the shading below.

(Error)
5.1.3. PLL stabilization time control register (SYSC_PLLSSCGSTCNTR)
PLL/SSCG stabilization time control register (SYSC_STCNTR) is used to control clock stabilization time.
<table>
<tr><td>bit</td><td colspan="8">bit 31 bit 8</td></tr>
<tr><td>Field</td><td colspan="8">Reserved</td></tr>
<tr><td>Data Attribute</td><td colspan="8">R0,WX</td></tr>
<tr><td>Prot_Attr</td><td colspan="8">WPS</td></tr>
<tr><td>Initial Value</td><td colspan="8">00000000 00000000 0000000</td></tr>
<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="4">Reserved</td><td colspan="4">PLLSTABS[3:0]</td></tr>
<tr><td>Data Attribute</td><td>R1,WX</td><td colspan="3">R/W</td><td>R1,WX</td><td colspan="3">R/W</td></tr>
<tr><td>Prot_Attr</td><td colspan="8">WPS</td></tr>
<tr><td>Initial Value</td><td colspan="4">1111</td><td colspan="4">1111</td></tr>
</table>

(Correct)
5.1.3. PLL/ SSCG stabilization time control register (SYSC_PLLSSCGSTCNTR)
PLL/SSCG stabilization time control register (SYSC_STCNTR) is used to control clock stabilization time.
<table>
<tr><td>bit</td><td colspan="8">bit 31 bit 8</td></tr>
<tr><td>Field</td><td colspan="8">Reserved</td></tr>
<tr><td>Data Attribute</td><td colspan="8">R0,WX</td></tr>
<tr><td>Prot_Attr</td><td colspan="8">WPS</td></tr>
<tr><td>Initial Value</td><td colspan="8">00000000 00000000 0000000</td></tr>
<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="4">SSCGSTABS[3:0]</td><td colspan="4">PLLSTABS[3:0]</td></tr>
<tr><td>Data Attribute</td><td>R1,WX</td><td colspan="3">R/W</td><td>R1,WX</td><td colspan="3">R/W</td></tr>
<tr><td>Prot_Attr</td><td colspan="8">WPS</td></tr>
<tr><td>Initial Value</td><td colspan="4">1111</td><td colspan="4">1111</td></tr>
</table>
</td></tr>
<tr><td>232</td><td>CHAPTER 5:Clock System 5.1.3. PLL/SSCG Stabilization Time Control Register (SYSC_PLLSSCGSTCNTR)</td><td>
Features of "For details" should be modified as indicated by the shading below.

(Error)
[bit31:4] Reserved

(Correct)
[bit31:8] Reserved
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>232</td><td>CHAPTER 5: Clock System
5.1.3. PLL/SSCG Stabilization Time Control Register (SYSC_PLLSSCGSTCNTR)</td><td>""5.1.3. PLL stabilization time control register (SYSC_PLLSSCGSTCNTR)"" should be corrected as indicated by the shading below.

(Error)
5.1.3. PLL stabilization time control register (SYSC_PLLSSCGSTCNTR)
[bit3:0] PLLSTABS[3:0] :
These bits select the stabilization time for PLL0/1/2/3 clock.
<table>
<tr><th>bit 7:4</th><th>Description</th></tr>
<tr><td>1000</td><td>Stabilization time : Main clock period[s] * 2^9 [cycle]</td></tr>
<tr><td>1001</td><td>Stabilization time : Main clock period[s] * 2^10 [cycle]</td></tr>
<tr><td>1010</td><td>Stabilization time : Main clock period[s] * 2^11 [cycle]</td></tr>
<tr><td>1011</td><td>Stabilization time : Main clock period[s] * 2^12 [cycle]</td></tr>
<tr><td>1100</td><td>Stabilization time : Main clock period[s] * 2^13 [cycle]</td></tr>
<tr><td>1101</td><td>Stabilization time : Main clock period[s] * 2^14 [cycle]</td></tr>
<tr><td>1110</td><td>Stabilization time : Main clock period[s] * 2^15 [cycle]</td></tr>
<tr><td>1111</td><td>Stabilization time : Main clock period[s] * 2^16 [cycle]</td></tr>
</table>
(Correct)
5.1.3.PLL/ SSCG stabilization time control register (SYSC_PLLSSCGSTCNTR)
[bit7:4] SSCGSTABS[3:0]
These bits select the stabilization time for SSCG PLL0/1/2/3 clock
<table>
<tr><th>bit 7:4</th><th>Description</th></tr>
<tr><td>1000</td><td>Stabilization time : Main clock period[s] * 2^9 [cycle]</td></tr>
<tr><td>1001</td><td>Stabilization time : Main clock period[s] * 2^10 [cycle]</td></tr>
<tr><td>1010</td><td>Stabilization time : Main clock period[s] * 2^11 [cycle]</td></tr>
<tr><td>1011</td><td>Stabilization time : Main clock period[s] * 2^12 [cycle]</td></tr>
<tr><td>1100</td><td>Stabilization time : Main clock period[s] * 2^13 [cycle]</td></tr>
<tr><td>1101</td><td>Stabilization time : Main clock period[s] * 2^14 [cycle]</td></tr>
<tr><td>1110</td><td>Stabilization time : Main clock period[s] * 2^15 [cycle]</td></tr>
<tr><td>1111</td><td>Stabilization time : Main clock period[s] * 2^16 [cycle]</td></tr>
</table>
Reading SSCGSTABS[3] bit always returns '1'
Note:
− These bits have to be set before PLL clock enable setting.
− Changing these bits are prohibited after PLL clock enable setting.
[bit3:0] PLLSTABS[3:0] :
These bits select the stabilization time for PLL0/1/2/3 clock.
<table>
<tr><th>bit 3:0</th><th>Description</th></tr>
<tr><td>1000</td><td>Stabilization time : Main clock period[s] * 2^9 [cycle]</td></tr>
<tr><td>1001</td><td>Stabilization time : Main clock period[s] * 2^10 [cycle]</td></tr>
<tr><td>1010</td><td>Stabilization time : Main clock period[s] * 2^11 [cycle]</td></tr>
<tr><td>1011</td><td>Stabilization time : Main clock period[s] * 2^12 [cycle]</td></tr>
<tr><td>1100</td><td>Stabilization time : Main clock period[s] * 2^13 [cycle]</td></tr>
<tr><td>1101</td><td>Stabilization time : Main clock period[s] * 2^14 [cycle]</td></tr>
<tr><td>1110</td><td>Stabilization time : Main clock period[s] * 2^15 [cycle]</td></tr>
<tr><td>1111</td><td>Stabilization time : Main clock period[s] * 2^16 [cycle]</td></tr>
</table></td></tr>
</table>

Page	Section	Change Results
242	CHAPTER 5:Clock System 6. Usage Precautions	Features of "For details" should be modified as indicated by the shading below. (before) ■The change of the PLL configuration The PLL configurations have to be set before PLL clock enable setting. Changing the PLL configurations are prohibited after PLL clock enable setting. Set the same configurations to RUN and PSS profiles if PLL clocks are enabled in PSS. ■The change of the clock gear configuration The clock gear configurations have to be set before PLL clock enable setting. Changing the clock gear configurations are prohibited after PLL clock enable setting (after) ■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
251	CHAPTER 6:Low Power Consumption 3.2. Power Supply Domain	Features of "For details" should be added as indicated by the shading below. (Error) ※PDxx = PD2/PD5_0/PD5_1/PD5_2/PD5_3/PD6_0/PD6_1 (Correct) **Note:** – PDxx = PD2/PD5_0/PD5_1/PD5_2/PD5_3/PD6_0/PD6_1

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>259</td><td>CHAPTER 6:Low Power Consumption 3.4. Profile</td><td>"Table 3-3" should be modified as indicated by the shading below.

(Error)
<table>
<tr><th>Classification</th><th>Error Description</th><th>Target State RUN/PSS</th><th>Error Flag</th></tr>
<tr><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td>The selected source clock of clock domain 0/MCUC clock domain is a clock for which oscillation is disabled.</td><td>RUN/PSS</td><td>SYSC_SYSRUNPEFR:PEF1
SYSC_SYSPSSPEFR:PEF1</td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td>The PLL0/SSCG0 setting was changed in the PLL0/SSCG0 oscillation state.</td><td>RUN/PSS</td><td>SYSC_SYSRUNPEFR:PEF3
SYSC_SYSPSSPEFR:PEF3</td></tr>
<tr><td>CSV</td><td>CSV operation was enabled while monitoring/reference clock oscillation was disabled.</td><td>RUN/PSS</td><td>SYSC_SYSRUNPEFR:PEF4
SYSC_SYSPSSPEFR:PEF4</td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
<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>SYSC_SYSPSSPEFR:PEF9</td></tr>
<tr><td>Power</td><td>Power domain restrictions (hierarchical relationship/combination) were ignored.</td><td>PSS</td><td>SYSC_SYSPSSPEFR:PEF10</td></tr>
</table>
(Correct)
<table>
<tr><th>Classification</th><th>Error Description</th><th>Target State RUN/PSS</th><th>Error Flag</th></tr>
<tr><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td>The selected source clock of clock domain 0/MCUC clock domain is a clock for which oscillation is disabled.</td><td>RUN/PSS</td><td>SYSC0_SYSRUNPEFR:PEF1
SYSC0_SYSPSSPEFR:PEF1</td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td>The PLL0/SSCG0 setting was changed in the PLL0/SSCG0 oscillation state.</td><td>RUN/PSS</td><td>SYSC0_SYSRUNPEFR:PEF3
SYSC0_SYSPSSPEFR:PEF3</td></tr>
<tr><td>CSV</td><td>CSV operation was enabled while monitoring/reference clock oscillation was disabled.</td><td>RUN/PSS</td><td>SYSC0_SYSRUNPEFR:PEF4
SYSC0_SYSPSSPEFR:PEF4</td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
<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>
<tr><td>Power</td><td>Power domain restrictions (hierarchical relationship/combination) were ignored.</td><td>PSS</td><td>SYSC0_SYSPSSPEFR:PEF10</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>259</td><td>CHAPTER 6:Low Power Consumption 3.4. Profile</td><td>The Note should be modified as indicated by the shading below.

(Error)
* If sub/PLL/SSCG is not implemented, those items are inapplicable for profile error.

(Correct)
Notes:
- If sub/PLL/SSCG is not implemented, those items are inapplicable for profile error.
- X : Don't care
- 0/1 : active value
- C : Object be to compare</td></tr>
<tr><td>260</td><td>CHAPTER 6:Low Power Consumption 3.4. Profile</td><td>"RUN" table should be modified as indicated by the shading below.

(Error)
<table>
<tr><td>SYSC0_RUNSSCG0CNTR0</td><td>SYSC0_RUNPLL0CNTR</td><td colspan="4">SYSC0_RUNCKSRER</td></tr>
<tr><td>SSCG0INSEL</td><td>PLL0INSEL</td><td>SSCG0EN</td><td>PLL0EN</td><td>MOSCEN</td><td>CROSCEN</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)
<table>
<tr><td>SYSC0_RUNSSCG0CNTR0</td><td>SYSC0_RUNPLL0CNTR</td><td colspan="4">SYSC0_RUNCKSRER</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>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>260</td><td>CHAPTER 6:Low Power Consumption 3.4. Profile</td><td>"PSS" table should be modified as indicated by the shading below.

(Error)
<table>
<tr><td>SYSC0_PSSSSCGC0NTR0</td><td>SYSC0_PSSPLL0CNTR</td><td colspan="4">SYSC0_PSSCKSRER</td></tr>
<tr><td>SSCG0INSEL</td><td>PLL0INSEL</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)
<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></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>261</td><td>CHAPTER 6:Low Power Consumption 3.4. Profile</td><td>"RUN" table should be modified as indicated by the shading below.

(Error)
<table><tr><td colspan="3">SYSC0_ RUNCKSELR</td><td colspan="3">SYSC1_RUNCKSELR0</td></tr><tr><td colspan="3">MCUCGCSL[2:0]</td><td colspan="3">CD0CSL[2:0]</td></tr><tr><td>0</td><td>1</td><td>1</td><td>X</td><td>X</td><td>X</td></tr></table>
(Correct)
<table><tr><td colspan="3">SYSC0_ RUNCKSELR</td><td colspan="3">SYSC1_RUNCKSELR0</td></tr><tr><td colspan="3">CDMCUCDCSL[2:0]</td><td colspan="3">CD0CSL[2:0]</td></tr><tr><td>0</td><td>1</td><td>1</td><td>X</td><td>X</td><td>X</td></tr></table></td></tr>
<tr><td>261</td><td>CHAPTER 6:Low Power Consumption 3.4. Profile</td><td>"RUN (PLL0)" table should be modified as indicated by the shading below.

(Error)
<table><tr><td>SYSC0_RUNCKSRER</td><td>SYSC0_PSSCKSRER</td><td>SYSC0_STSCKSRER</td><td>SYSC0_RUNPLL0CNTR</td><td>SYSC0_PSSPLL0CNTR</td><td>SYSC0_STSPLL0CNTR</td></tr><tr><td>PLL0EN</td><td>PLL0EN</td><td>PLL0EN</td><td>PLLINSEL
PLLDIVN[7:0]
PLLDIVM[3:0]
PLLDIVL[1:0]</td><td>PLLINSEL
PLLDIVN[7:0]
PLLDIVM[3:0]
PLLDIVL[1:0]</td><td>PLLINSEL
PLLDIVN[7:0]
PLLDIVM[3:0]
PLLDIVL[1:0]</td></tr><tr><td>0</td><td>X</td><td>1</td><td>C</td><td>X</td><td>C</td></tr></table>
(Correct)
<table><tr><td>SYSC0_RUNCKSRER</td><td>SYSC0_PSSCKSRER</td><td>SYSC0_STSCKSRER</td><td>SYSC0_RUNPLL0CNTR</td><td>SYSC0_PSSPLL0CNTR</td><td>SYSC0_STSPLL0CNTR</td></tr><tr><td>PLL0EN</td><td>PLL0EN</td><td>PLL0EN</td><td>PLL0ISEL
PLL0DIVN[7:0]
PLL0DIVM[3:0]
PLL0DIVL[1:0]</td><td>PLL0ISEL
PLL0DIVN[7:0]
PLL0DIVM[3:0]
PLL0DIVL[1:0]</td><td>PLL0ISEL
PLL0DIVN[7:0]
PLL0DIVM[3:0]
PLL0DIVL[1:0]</td></tr><tr><td>0</td><td>X</td><td>1</td><td>C</td><td>X</td><td>C</td></tr></table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>262</td><td>CHAPTER 6:Low Power Consumption 3.4. Profile</td><td>"RUN (SSCG0)" table should be modified as indicated by the shading below.

(Error)
<table>
<tr><td>SYSC0_RUNCKSRER</td><td>SYSC0_PSSCKSRER</td><td>SYSC0_STSCKSRER</td><td>SYSC0_RUNSSCG0CNTR1</td><td>SYSC0_PSSSSCG0CNTR0/1</td><td>SYSC0_STSSSCG0CNTR0/1</td></tr>
<tr><td>SSCG0EN</td><td>SSCG0EN</td><td>SSCG0EN</td><td>SSCGINSEL
SSCGDIVN[7:0]
SSCGDIVM[3:0]
SSCGDIVL[1:0]
SSCGRATE[9:0]
SSCGMODE
SSCGFREQ[1:0]
SSCGSSEN</td><td>SSCGINSEL
SSCGDIVN[7:0]
SSCGDIVM[3:0]
SSCGDIVL[1:0]
SSCGRATE[9:0]
SSCGMODE
SSCGFREQ[1:0]
SSCGSSEN</td><td>SSCGINSEL
SSCGDIVN[7:0]
SSCGDIVM[3:0]
SSCGDIVL[1:0]
SSCGRATE[9:0]
SSCGMODE
SSCGFREQ[1:0]
SSCGSSEN</td></tr>
<tr><td>0</td><td>X</td><td>1</td><td>C</td><td>X</td><td>C</td></tr>
</table>
(Correct)
<table>
<tr><td>SYSC0_RUNCKSRER</td><td>SYSC0_PSSCKSRER</td><td>SYSC0_STSCKSRER</td><td>SYSC0_RUNSSCG0CNTR1</td><td>SYSC0_PSSSSCG0CNTR0/1</td><td>SYSC0_STSSSCG0CNTR0/1</td></tr>
<tr><td>SSCG0EN</td><td>SSCG0EN</td><td>SSCG0EN</td><td>SSCG0ISEL
SSCG0DIVN[7:0]
SSCG0DIVM[3:0]
SSCG0DIVL[1:0]
SSCG0RATE[9:0]
SSCG0MODE
SSCG0FREQ[1:0]
SSCG0SSEN</td><td>SSCG0ISEL
SSCG0DIVN[7:0]
SSCG0DIVM[3:0]
SSCG0DIVL[1:0]
SSCG0RATE[9:0]
SSCG0MODE
SSCG0FREQ[1:0]
SSCG0SSEN</td><td>SSCG0ISEL
SSCG0DIVN[7:0]
SSCG0DIVM[3:0]
SSCG0DIVL[1:0]
SSCG0RATE[9:0]
SSCG0MODE
SSCG0FREQ[1:0]
SSCG0SSEN</td></tr>
<tr><td>0</td><td>X</td><td>1</td><td>C</td><td>X</td><td>C</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>262</td><td>CHAPTER 6:Low Power Consumption 3.4. Profile</td><td>"PSS (PLL0)" table should be modified as indicated by the shading below.

(Error)
<table>
<tr><td>SYSC_RUN
CKSRER</td><td>SYSC_PSS
CKSRER</td><td>SYSC_STS
CKSRER</td><td>SYSC0_RUN
PLL0CNTR</td><td>SYSC0_PSS
PLL0CNTR</td><td>SYSC0_STS
PLL0CNTR</td></tr>
<tr><td>PLL0EN</td><td>PLL0EN</td><td>PLL0EN</td><td>PLLINSEL
PLLDIVN[7:0]
PLLDIVM[3:0]
PLLDIVL[1:0]</td><td>PLLINSEL
PLLDIVN[7:0]
PLLDIVM[3:0]
PLLDIVL[1:0]</td><td>PLLINSEL
PLLDIVN[7:0]
PLLDIVM[3:0]
PLLDIVL[1:0]</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>C</td><td>C</td><td>X</td></tr>
</table>
(Correct)
<table>
<tr><td>SYSC0_RUN
CKSRER</td><td>SYSC0_PSS
CKSRER</td><td>SYSC0_STS
CKSRER</td><td>SYSC0_RUN
PLL0CNTR</td><td>SYSC0_PSS
PLL0CNTR</td><td>SYSC0_STS
PLL0CNTR</td></tr>
<tr><td>PLL0EN</td><td>PLL0EN</td><td>PLL0EN</td><td>PLL0ISEL
PLL0DIVN[7:0]
PLL0DIVM[3:0]
PLL0DIVL[1:0]</td><td>PLL0ISEL
PLL0DIVN[7:0]
PLL0DIVM[3:0]
PLL0DIVL[1:0]</td><td>PLL0ISEL
PLL0DIVN[7:0]
PLL0DIVM[3:0]
PLL0DIVL[1:0]</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>C</td><td>C</td><td>X</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>262</td><td>CHAPTER 6:Low Power Consumption 3.4. Profile</td><td>"PSS (SSCG0)" table should be modified as indicated by the shading below.

(Error)
<table>
<tr><td>SYSC0_RUNCKSRER</td><td>SYSC0_PSSCKSRER</td><td>SYSC0_STSCKSRER</td><td>SYSC0_RUNSSCG0CNTR0/1</td><td>SYSC0_PSSSSCG0CNTR0/1</td><td>SYSC0_STSSSCG0CNTR0/1</td></tr>
<tr><td>SSCG0EN</td><td>SSCG0EN</td><td>SSCG0EN</td><td>SSCGINSEL
SSCGDIVN[7:0]
SSCGDIVM[3:0]
SSCGDIVL[1:0]
SSCGRATE[9:0]
SSCGMODE
SSCGFREQ[1:0]
SSCGSSEN</td><td>SSCGINSEL
SSCGDIVN[7:0]
SSCGDIVM[3:0]
SSCGDIVL[1:0]
SSCGRATE[9:0]
SSCGMODE
SSCGFREQ[1:0]
SSCGSSEN</td><td>SSCGINSEL
SSCGDIVN[7:0]
SSCGDIVM[3:0]
SSCGDIVL[1:0]
SSCGRATE[9:0]
SSCGMODE
SSCGFREQ[1:0]
SSCGSSEN</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>C</td><td>C</td><td>X</td></tr>
</table>

(Correct)
<table>
<tr><td>SYSC0_RUNCKSRER</td><td>SYSC0_PSSCKSRER</td><td>SYSC0_STSCKSRER</td><td>SYSC0_RUNSSCG0CNTR1</td><td>SYSC0_PSSSSCG0CNTR0/1</td><td>SYSC0_STSSSCG0CNTR0/1</td></tr>
<tr><td>SSCG0EN</td><td>SSCG0EN</td><td>SSCG0EN</td><td>SSCG0ISEL
SSCG0DIVN[7:0]
SSCG0DIVM[3:0]
SSCG0DIVL[1:0]
SSCG0RATE[9:0]
SSCG0MODE
SSCG0FREQ[1:0]
SSCG0SSEN</td><td>SSCG0ISEL
SSCG0DIVN[7:0]
SSCG0DIVM[3:0]
SSCG0DIVL[1:0]
SSCG0RATE[9:0]
SSCG0MODE
SSCG0FREQ[1:0]
SSCG0SSEN</td><td>SSCG0ISEL
SSCG0DIVN[7:0]
SSCG0DIVM[3:0]
SSCG0DIVL[1:0]
SSCG0RATE[9:0]
SSCG0MODE
SSCG0FREQ[1:0]
SSCG0SSEN</td></tr>
<tr><td>0</td><td>X</td><td>1</td><td>C</td><td>X</td><td>C</td></tr>
</table></td></tr>
<tr><td>265</td><td>CHAPTER 6:Low Power Consumption 3.4. Profile</td><td>"RUN" table should be modified as indicated by the shading below.

(Error)
<table>
<tr><td colspan="4">SYSC0_RUNCKSRER</td><td colspan="4">SWDG0_CFG</td></tr>
<tr><td>SOSCEN</td><td>MOSCEN</td><td>SCROSCEN</td><td>CROSCEN</td><td>WDENRUN</td><td>WDENPSS</td><td>ALLOWCLKSTOP</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>
</table>

(Correct)
<table>
<tr><td colspan="4">SYSC0_RUNCKSRER</td><td colspan="4">SWDG0_CFG</td></tr>
<tr><td>SOSCEN</td><td>MOSCEN</td><td>SCROSCEN</td><td>CROSCEN</td><td>WDENRUN</td><td>WDENPSS</td><td>ALLOWSTOPCLK</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>
</table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>265</td><td>CHAPTER 6:Low Power Consumption 3.4. Profile</td><td>"PSS" table should be modified as indicated by the shading below.

(Error)
<table>
<tr><td colspan="4">SYSC0_PSSCKSRER</td><td colspan="5">SWDG0_CFG</td></tr>
<tr><td>SOSCEN</td><td>MOSCEN</td><td>SCROSCEN</td><td>CROSCEN</td><td>WDENRUN</td><td>WDENPSS</td><td>ALLOWCLKSTOP</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>
</table>

(Correct)
<table>
<tr><td colspan="4">SYSC0_PSSCKSRER</td><td colspan="5">SWDG0_CFG</td></tr>
<tr><td>SOSCEN</td><td>MOSCEN</td><td>SCROSCEN</td><td>CROSCEN</td><td>WDENRUN</td><td>WDENPSS</td><td>ALLOWSTOPCLK</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>
</table></td></tr>
<tr><td>266</td><td>CHAPTER 6:Low Power Consumption 3.4. Profile</td><td>"PSS" table should be modified as indicated by the shading below.

(Error)
<table>
<tr><td colspan="2">SYSC0_PSSCKSRER</td><td colspan="5">HWDG_CFG</td></tr>
<tr><td>SCROSCEN</td><td>CROSCEN</td><td>WDENRUN</td><td>WDENPSS</td><td>ALLOWCLKSTOP</td><td colspan="2">CLKSEL[1:0]</td></tr>
<tr><td>0</td><td>X</td><td>X</td><td>1</td><td>0</td><td>X</td><td>1</td></tr>
</table>

(Correct)
<table>
<tr><td colspan="2">SYSC0_PSSCKSRER</td><td colspan="5">HWDG_CFG</td></tr>
<tr><td>SCROSCEN</td><td>CROSCEN</td><td>WDENRUN</td><td>WDENPSS</td><td>ALLOWSTOPCLK</td><td colspan="2">CLKSEL[1:0]</td></tr>
<tr><td>0</td><td>X</td><td>X</td><td>1</td><td>0</td><td>X</td><td>1</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>268</td><td>CHAPTER 6:Low Power Consumption 3.5. Interrupts</td><td>"Table 3-4" table should be modified as indicated by the shading below.

(Error)
<table>
<tr><th>Generation Condition</th><th>Factor Register</th><th>Enabling Condition</th><th>Clearing Condition</th></tr>
<tr><td>When a RUN profile update is completed</td><td>SYSC_SYSSTSR: RUNDF0</td><td>Writing "1" to the SYSC_SYSINTER:RUNDIE0 bit</td><td>Writing "1" to the SYSC_SYSICLR:RUNDFCLR0 bit</td></tr>
<tr><td>When the RUN profile register contents have an error and the RUN profile update trigger is enabled</td><td>SYSC_SYSERRIR1: RUNERRIF0</td><td>Always enabled because of the NMI level</td><td>Writing "1" to the SYSC_SYSERRICLR1:RUNERRICLR0 bit</td></tr>
<tr><td>When the PSS profile register contents have an error during a transition to PSS (execution of the WFI instruction)</td><td>SYSC_SYSERRIR1: PSSERRIF0</td><td>Always enabled because of the NMI level</td><td>Writing "1" to the SYSC_SYSERRICLR1:PSSERRICLR0 bit</td></tr>
<tr><td>When the value written to the RUN update enable register is invalid</td><td>SYSC_SYSERRIR1: RUNTRGERRIF</td><td>Always enabled because of the NMI level</td><td>Writing "1" to the SYSC_SYSERRICLR1:RUNTRGERRICLR bit</td></tr>
<tr><td>When the RUN profile is updated again during a RUN profile update</td><td>SYSC_SYSERRIR1: TRGERRIF</td><td>Always enabled because of the NMI level</td><td>Writing "1" to the SYSC_SYSERRICLR1:TRGERRICLR bit</td></tr>
</table>
(Correct)
<table>
<tr><th>Generation Condition</th><th>Factor Register</th><th>Enabling Condition</th><th>Clearing Condition</th></tr>
<tr><td>When a RUN profile update is completed</td><td>SYSC0_SYSSTSR: RUNDF0</td><td>Writing "1" to the SYSC0_SYSINTER:RUNDIE0 bit</td><td>Writing "1" to the SYSC0_SYSICLR:RUNDFCLR0 bit</td></tr>
<tr><td>When the RUN profile register contents have an error and the RUN profile update trigger is enabled</td><td>SYSC0_SYSERRIR1: RUNERRIF0</td><td>Always enabled because of the NMI level</td><td>Writing "1" to the SYSC0_SYSERRICLR1: RUNERRICLR0 bit</td></tr>
<tr><td>When the PSS profile register contents have an error during a transition to PSS (execution of the WFI instruction)</td><td>SYSC0_SYSERRIR1: PSSERRIF0</td><td>Always enabled because of the NMI level</td><td>Writing "1" to the SYSC0_SYSERRICLR1: PSSERRICLR0 bit</td></tr>
<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_SYSERRICLR1: RUNTRGERRICLR bit</td></tr>
<tr><td>When the RUN profile is updated again during a RUN profile update</td><td>SYSC0_SYSERRIR1: TRGERRIF</td><td>Always enabled because of the NMI level</td><td>Writing "1" to the SYSC0_SYSERRICLR1: TRGERRICLR bit</td></tr>
</table></td></tr>
</table>

Page	Section	Change Results
271	CHAPTER 6:Low Power Consumption 4.1. RUN Profile Update	"Example explanations" should be modified as indicated by the shading below. (Error) 3. Start a profile update by writing 0xAB to the RUN profile update trigger register. (SYSC_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 (SYSC_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 (SYSC_SYSSTSR:RUNSTS0). 2. The circuit copies the RUN profile contents to the APPLIED 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. (SYSC_SYSSTSR:RUNSTS0) to "0" and sets "1" in the system status register (SYSC_SYSSTSR:RUNDF0). Also, if the system status interrupt enable register (SYSC_SYSINTER:RUNDIE0) is enabled, an interrupt is generated. (Correct) 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 APPLIED 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.

Page	Section	Change Results
271	CHAPTER 6:Low Power Consumption 4.1. RUN Profile Update	The Note should be modified as indicated by the shading below. (Error) 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 (SYSC_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 (SYSC_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. (Correct) 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.

Page	Section	Change Results
273	CHAPTER 6:Low Power Consumption 4.2. Transition from RUN to PSS	"Example explanations" should be modified as indicated by the shading below. (Error) 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. (SYSC_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 (SYSC_SYSSTSR:PSSSTS0) to "1". 2. The circuit copies the PSS profile contents to the APPLIED 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 (SYSC_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 of each CPU (SYSC_SYSSTSR:PSSSTS0) to "0" and the PSS profile completion flag of each CPU (SYSC_SYSSTSR:PSSDF0) "1". (Correct) 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 APPLIED 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 of each CPU (SYSC0_SYSSTSR:PSSSTS0) to "0" and the PSS profile completion flag of each CPU (SYSC0_SYSSTSR:PSSDF0) "1".
273	CHAPTER 6:Low Power Consumption 4.2. Transition from RUN to PSS	Revised the word “WIF” to “WFI” in the “Notes”

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>273</td><td>CHAPTER 6:Low Power Consumption 4.2. Transition from RUN to PSS</td><td>The Note should be added as indicated by the shading below.

(Correct)
Notes:
− When transferring from a RUN mode to PSS mode, please be sure to make Main_OSC enable.</td></tr>
<tr><td>279</td><td>CHAPTER 6:Low Power Consumption 5. Registers</td><td>The Note should be added as indicated by the shading below.

− 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.
− Protection range of SYSC0_PROTKEYR is 0xB060_0080-0xB060_06FC.
− Protection range of SYSC1_PROTKEYR is 0xB030_0080-0xB030_027C.</td></tr>
<tr><td>285</td><td>CHAPTER 6:Low Power Consumption 5.1.1. Protection Key Setting Register (SYSC0_PROTKEYR)</td><td>The Note should be deleted as indicated by the shading below.

Notes:
- After the register in MCUconfig area is written irrestespective of the master, the protection key is locked again.
- This register permits word access only. Any other access is invalid.</td></tr>
<tr><td>290</td><td>CHAPTER 6:Low Power Consumption 5.2.2. RUN Clock Source Enable Register (SYSC0_RUNCKSRER)</td><td>"R/W Attribute" should be modified 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 colspan="4">Reserved</td><td>SOSCEN</td><td>MOSCEN</td><td>SCROSCEN</td><td>CROSCEN</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>

(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>MOSCEN</td><td>SCROSCEN</td><td>CROSCEN</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>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>314</td><td>CHAPTER 6:Low Power Consumption 5.2.12. RUN Profile Update Trigger Register (SYSC0_TRGRUNCNTR)</td><td>The Note should be modified as indicated by the shading below.

(Error)
Notes:
- If the value written in this register is invalid, an error flag (SYSC_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 (SYSC_SYSERRIR1:TRGERRIF is "1") is enabled.

(Correct)
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.</td></tr>
<tr><td>342</td><td>CHAPTER 6:Low Power Consumption 5.3.11. PSS Regulator Setting Register (SYSC0_PSSREGCFGR)</td><td>"R/W Attribute" should be modified 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>RMSEL</td><td colspan="6">Reserved</td><td>RVSEL</td></tr>
<tr><td>R/W Attribute</td><td>R/W</td><td colspan="6">R0,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 colspan="6">000000</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>RMSEL</td><td colspan="6">Reserved</td><td>RVSEL</td></tr>
<tr><td>R/W Attribute</td><td>R/W</td><td colspan="6">R0,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 colspan="6">000000</td><td>0</td></tr>
</table>
</td></tr>
<tr><td>343</td><td>CHAPTER 6:Low Power Consumption 5.3.12. PSS Profile Update Enable Register (SYSC0_PSSENR)</td><td>The Note should be modified as indicated by the shading below.

(Error)
Notes:
If the value written in this register invalid, an error flag (SYSC_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.
(Correct)
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.</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>343</td><td>CHAPTER 6:Low Power Consumption 5.3.12. PSS Profile Update Enable Register (SYSC0_PSSENR)</td><td>"[bit3] SOSCEN" should be deleted as indicated by the shading below.

[bit3] SOSCEN: Sub clock oscillation enable bit
This bit sets the sub clock oscillation which is being updated.
<table>
<tr><th>bit</th><th>Description</th></tr>
<tr><td>0</td><td>Disable sub clock oscillation</td></tr>
<tr><td>1</td><td>Enable sub clock oscillation</td></tr>
</table></td></tr>
<tr><td>343</td><td>CHAPTER 6:Low Power Consumption 5.3.12. PSS Profile Update Enable Register (SYSC0_PSSENR)</td><td>"[bit2] MOSCEN" should be deleted as indicated by the shading below.

[bit2] MOSCEN: Main clock oscillation enable bit
This bit sets the main clock oscillation which is being updated.
<table>
<tr><th>bit</th><th>Description</th></tr>
<tr><td>0</td><td>Disable main clock oscillation</td></tr>
<tr><td>1</td><td>Enable main clock oscillation</td></tr>
</table></td></tr>
<tr><td>343</td><td>CHAPTER 6:Low Power Consumption 5.3.12. PSS Profile Update Enable Register (SYSC0_PSSENR)</td><td>"[bit1] SCROSCEN" should be deleted as indicated by the shading below.

[bit1] SCROSCEN: Low-speed CR oscillation enable bit
This bit sets the low-speed CR oscillation which is being updated.
<table>
<tr><th>bit</th><th>Description</th></tr>
<tr><td>0</td><td>Disable low-speed CR oscillation</td></tr>
<tr><td>1</td><td>Enable low-speed CR oscillation</td></tr>
</table></td></tr>
<tr><td>343</td><td>CHAPTER 6:Low Power Consumption 5.3.12. PSS Profile Update Enable Register (SYSC0_PSSENR)</td><td>"[bit0] CROSCEN" should be deleted as indicated by the shading below.

[bit0] CROSCEN: High-speed CR oscillation enable bit
This bit sets the high-speed CR oscillation which is being updated.
<table>
<tr><th>bit</th><th>Description</th></tr>
<tr><td>0</td><td>Disable high-speed CR oscillation</td></tr>
<tr><td>1</td><td>Enable high-speed CR oscillation</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>374</td><td>CHAPTER 6:Low Power Consumption 5.5.1. STS Power Domain Setting Register (SYSC0_STSPDCFGR)</td><td>"Initial Value" should be modified as indicated by the shading below.

(Error)
<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>PD5_3 EN</td><td>PD5_2 EN</td><td>PD5_1 EN</td><td>PD5_0 EN</td><td colspan="2">Reserved</td><td>PD4_1 EN</td><td>PD4_0 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 colspan="2">R0,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>1</td><td>1</td><td>1</td><td>1</td><td colspan="2">00</td><td>1*</td><td>1*</td></tr>
</table>
(Correct)
<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>PD5_3 EN</td><td>PD5_2 EN</td><td>PD5_1 EN</td><td>PD5_0 EN</td><td colspan="2">Reserved</td><td>PD4_1 EN</td><td>PD4_0 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 colspan="2">R0,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>1</td><td>1</td><td>1</td><td>1</td><td colspan="2">00</td><td>1</td><td>1</td></tr>
</table></td></tr>
<tr><td>381</td><td>CHAPTER 6:Low Power Consumption 5.5.2. STS Clock Source Enable Register (SYSC0_STSCKSRER)</td><td>The Note of "[bit6] MOSCRDY" should be modified as indicated by the shading below.

(Error)
Note:
- This bit is read as "1" if stabilization is completed and sub clock is stable.

(Correct)
Note:
- This bit is read as "1" if stabilization is completed and main clock is stable.</td></tr>
<tr><td>381</td><td>CHAPTER 6:Low Power Consumption 5.5.2. STS Clock Source Enable Register (SYSC0_STSCKSRER)</td><td>The Note of "[bit5] SCROSCRDY" should be modified as indicated by the shading below.

(Error)
Note:
- This bit is read as "1" if stabilization is completed and sub clock is stable.

(Correct)
Note:
- This bit is read as "1" if stabilization is completed and Low-speed CR clock is stable.</td></tr>
</table>

Page	Section	Change Results
381	CHAPTER 6:Low Power Consumption 5.5.2. STS Clock Source Enable Register (SYSC0_STSCKSRER)	The Note of "[bit4] CROSCRDY" should be modified as indicated by the shading below. (Error) Note: - This bit is read as "1" if stabilization is completed and sub clock is stable. (Correct) Note: - This bit is read as "1" if stabilization is completed and High-speed CR clock is stable.
406	CHAPTER 6:Low Power Consumption 5.6.1. System Status Register (SYSC0_SYSSTSR)	The Note of "[bit5] PSSDF0" should be modified as indicated by the shading below. (Error) Notes: - To clear this register, write "1" in SYSC_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. (Correct) 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.
406	CHAPTER 6:Low Power Consumption 5.6.1. System Status Register (SYSC0_SYSSTSR)	The Note of "[bit4] RUNDF0" should be modified as indicated by the shading below. (Error) Notes: - To clear this register, write "1" in SYSC_SYSICLR:RUNDFCLR0. - To use this register as an interrupt, write "1" in SYSC_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. (Correct) 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.

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>409</td><td>CHAPTER 6:Low Power Consumption 5.6.4. System Error Interrupt Factor Register 0 (SYSC0_SYSERRIR0)</td><td>"SOAIF bit" and "MOSCIF bit" should be deleted 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 colspan="4">Reserved</td><td>SOAIF</td><td>MOAIF</td><td>SOSCIF</td><td>MOSCIF</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>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 colspan="6">Reserved</td><td>SOSCIF</td><td>MOSCIF</td></tr>
<tr><td>R/W Attribute</td><td colspan="6">R0,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="6">000000</td><td>0</td><td>0</td></tr>
</table></td></tr>
<tr><td>412</td><td>CHAPTER 6:Low Power Consumption 5.6.4. System Error Interrupt Factor Register 0 (SYSC0_SYSERRIR0)</td><td>The bit width of "Reserved" should be modified as indicated by the shading below.

(Error) [bit7:4] Reserved: Reserved bits

(Correct) [bit7:2] Reserved: Reserved bits</td></tr>
<tr><td>412</td><td>CHAPTER 6:Low Power Consumption 5.6.4. System Error Interrupt Factor Register 0 (SYSC0_SYSERRIR0)</td><td>"[bit3] SOAIF" should be deleted as indicated by the shading below.

[bit3] SOAIF: Sub oscillation abnormal monitor detection error interrupt request bit
This bit retains the fact that an interrupt request was generated for a sub oscillation abnormal monitor detection error.
<table>
<tr><th>bit</th><th>Description</th></tr>
<tr><td>0</td><td>No interrupt request is detected.</td></tr>
<tr><td>1</td><td>An interrupt request is detected.</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>412</td><td>CHAPTER 6:Low Power Consumption 5.6.4. System Error Interrupt Factor Register 0 (SYSC0_SYSERRIR0)</td><td>"[bit2] MOAIF" should be deleted as indicated by the shading below.

[bit2] MOAIF: Main oscillation abnormal monitor detection error interrupt request bit
This bit retains the fact that an interrupt request was generated for a main oscillation abnormal monitor detection error.
<table>
<tr><th>bit</th><th>Description</th></tr>
<tr><td>0</td><td>No interrupt request is detected.</td></tr>
<tr><td>1</td><td>An interrupt request is detected.</td></tr>
</table></td></tr>
<tr><td>414</td><td>CHAPTER 6:Low Power Consumption 5.6.5. System Error Interrupt Factor Register 1 (SYSC0_SYSERRIR1)</td><td>The description of "[bit3] PSSENERRIF0" should be modified as indicated by the shading below.

(Error)
[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 (SYSC_PSSENR:PSSEN0).

(Correct)
[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).</td></tr>
<tr><td>415</td><td>CHAPTER 6:Low Power Consumption 5.6.6. System Error Interrupt Factor Clear Register 0 (SYSC0_SYSERRICLR0)</td><td>"SOAIF bit" and "MOSCIF bit" should be deleted as indicated by the shading below.

(Error)
<table>
<tr><th>bit</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>Field</td><td colspan="4">Reserved</td><td>SOAICLR</td><td>MOAICLR</td><td>SOSCICLR</td><td>MOSCICLR</td></tr>
<tr><td>R/W Attribute</td><td colspan="4">R0,WX</td><td>R0,W</td><td>R0,W</td><td>R0,W</td><td>R0,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>0</td><td>0</td><td>0</td></tr>
</table>
(Correct)
<table>
<tr><th>bit</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>Field</td><td colspan="6">Reserved</td><td>SOSCICLR</td><td>MOSCICLR</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">WPS</td></tr>
<tr><td>Initial Value</td><td colspan="6">000000</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>418</td><td>CHAPTER 6:Low Power Consumption 5.6.6. System Error Interrupt Factor Clear Register 0 (SYSC0_SYSERRICLR0)</td><td>The bit width of "Reserved" should be modified as indicated by the shading below.

(Error) [bit7:4] Reserved: Reserved bits

(Correct) [bit7:2] Reserved: Reserved bits</td></tr>
<tr><td>418</td><td>CHAPTER 6:Low Power Consumption 5.6.6. System Error Interrupt Factor Clear Register 0 (SYSC0_SYSERRICLR0)</td><td>"[bit3] SOAICLR" should be deleted as indicated by the shading below.

[bit3] SOAICLR: Sub oscillation abnormal monitor detection error interrupt factor clear bit
This bit clears the interrupt factor of a sub oscillation abnormal monitor detection error.
<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: Clear the interrupt factor.</td></tr><tr><td>During read operation</td><td>The read value is always "0".</td></tr></table></td></tr>
<tr><td>418</td><td>CHAPTER 6:Low Power Consumption 5.6.6. System Error Interrupt Factor Clear Register 0 (SYSC0_SYSERRICLR0)</td><td>"[bit2] MOAICLR" should be deleted as indicated by the shading below.

[bit2] MOAICLR: Main oscillation abnormal monitor detection error interrupt factor clear bit
This bit clears the interrupt factor of a main oscillation abnormal monitor detection error.
<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: Clear the interrupt factor.</td></tr><tr><td>During read operation</td><td>The read value is always "0".</td></tr></table></td></tr>
<tr><td>418</td><td>CHAPTER 6:Low Power Consumption 5.6.6. System Error Interrupt Factor Clear Register 0 (SYSC0_SYSERRICLR0)</td><td>"[bit1] SOSCCLR" should be modified as indicated by the shading below.

(Error)
[bit1] SOSCCLR: Sub oscillation abnormality detection error interrupt factor clear bit

(Correct)
[bit1] SOSCICLR: Sub oscillation abnormality detection error interrupt factor clear bit</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>418</td><td>CHAPTER 6:Low Power Consumption 5.6.6. System Error Interrupt Factor Clear Register 0 (SYSC0_SYSERRICLR0)</td><td>"[bit0] MOSCCLR" should be modified as indicated by the shading below.

(Error)
[bit0] MOSCCLR: Main oscillation abnormality detection error interrupt factor clear bit

(Correct)
[bit0] MOSCICLR: Main oscillation abnormality detection error interrupt factor clear bit</td></tr>
<tr><td>435</td><td>CHAPTER 6:Low Power Consumption 5.9.1. MCU_Config_Group Interrupt Request Status Register0 (MCG_IRSR0)</td><td>"R/W Attribute" should be modified 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="7">Reserved</td><td>IRQ_PW</td></tr>
<tr><td>R/W Attribute</td><td colspan="7">R,WX</td><td>R,WX</td></tr>
<tr><td>Protection Attribute</td><td colspan="8">-</td></tr>
<tr><td>Initial Value</td><td colspan="7">0000000</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="7">Reserved</td><td>IRQ_PW</td></tr>
<tr><td>R/W Attribute</td><td colspan="7">R0,WX</td><td>R,WX</td></tr>
<tr><td>Protection Attribute</td><td colspan="8">-</td></tr>
<tr><td>Initial Value</td><td colspan="7">0000000</td><td>0</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>435</td><td>CHAPTER 6:Low Power Consumption 5.9.1. MCU_Config_Group Interrupt Request Status Register0 (MCG_IRSR0)</td><td>"R/W Attribute" should be modified as indicated by the shading below.

(Error)
<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="4">Reserved</td><td>IRQ_SCT_SUB</td><td>IRQ_SCT_MAIN</td><td>IRQ_SCT_SCR</td><td>IRQ_SCT_CR</td></tr>
<tr><td>R/W Attribute</td><td colspan="4">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">-</td></tr>
<tr><td>Initial Value</td><td colspan="4">0000</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
(Correct)
<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="4">Reserved</td><td>IRQ_SCT_SUB</td><td>IRQ_SCT_MAIN</td><td>IRQ_SCT_SCR</td><td>IRQ_SCT_CR</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">-</td></tr>
<tr><td>Initial Value</td><td colspan="4">0000</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table></td></tr>
<tr><td>435</td><td>CHAPTER 6:Low Power Consumption 5.9.1. MCU_Config_Group Interrupt Request Status Register0 (MCG_IRSR0)</td><td>"R/W Attribute" should be modified 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="4">Reserved</td><td>Reserved</td><td>IRQ_RTC</td><td>IRQ_SCU</td><td>IRQ_HWDG</td></tr>
<tr><td>R/W Attribute</td><td colspan="4">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">-</td></tr>
<tr><td>Initial Value</td><td colspan="4">0000</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="4">Reserved</td><td>Reserved</td><td>IRQ_RTC</td><td>IRQ_SCU</td><td>IRQ_HWDG</td></tr>
<tr><td>R/W Attribute</td><td colspan="4">R0,WX</td><td>R0,WX</td><td>R,WX</td><td>R,WX</td><td>R,WX</td></tr>
<tr><td>Protection Attribute</td><td colspan="8">-</td></tr>
<tr><td>Initial Value</td><td colspan="4">0000</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>435</td><td>CHAPTER 6:Low Power Consumption 5.9.1. MCU_Config_Group Interrupt Request Status Register0 (MCG_IRSR0)</td><td>"R/W Attribute" and "Field" should be modified 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 colspan="3">Reserved</td><td>NMI_WAKE</td><td>NMI_EXTINT</td><td>NMI_HWDG</td><td>NMI_SCU</td><td>NMI_LVD</td></tr>
<tr><td>R/W Attribute</td><td colspan="3">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">-</td></tr>
<tr><td>Initial Value</td><td colspan="3">000</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 colspan="3">Reserved</td><td>Reserved</td><td>NMI_EXTINT</td><td>NMI_HWDG</td><td>NMI_SCU</td><td>NMI_LVD</td></tr>
<tr><td>R/W Attribute</td><td colspan="3">R0,WX</td><td>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">-</td></tr>
<tr><td>Initial Value</td><td colspan="3">000</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table></td></tr>
<tr><td>436</td><td>CHAPTER 6:Low Power Consumption 5.9.1. MCU_Config_Group Interrupt Request Status Register0 (MCG_IRSR0)</td><td>The bit width of "Reserved" should be modified as indicated by the shading below.

(Error) [bit7:5] Reserved: Reserved bits

(Correct) [bit7:4] Reserved: Reserved bits</td></tr>
<tr><td>436</td><td>CHAPTER 6:Low Power Consumption 5.9.1. MCU_Config_Group Interrupt Request Status Register0 (MCG_IRSR0)</td><td>"[bit4] NMI_WAKE" should be deleted as indicated by the shading below.

[bit4] NMI_WAKE: WAKE detection interrupt status (NMI level)
<table>
<tr><th>bit</th><th>Description</th></tr>
<tr><td>0</td><td>The interrupt does not occur.</td></tr>
<tr><td>1</td><td>The interrupt occurs.</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>439</td><td>CHAPTER 6:Low Power Consumption 5.9.3. MCU_Config_Group Interrupt Request Status Register2 (MCG_IRSR2)</td><td>"R/W Attribute" should be modified 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 colspan="8">R,WX</td></tr>
<tr><td>Protection Attribute</td><td colspan="8">-</td></tr>
<tr><td>Initial Value</td><td colspan="8">00000000</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,WX</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>
<tr><td>439</td><td>CHAPTER 6:Low Power Consumption 5.9.3. MCU_Config_Group Interrupt Request Status Register2 (MCG_IRSR2)</td><td>"R/W Attribute" should be modified as indicated by the shading below.

(Error)
<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>IRQ_RLT2</td><td>IRQ_RLT1</td><td>IRQ_RLT0</td></tr>
<tr><td>R/W Attribute</td><td colspan="5">R,WX</td><td>R,WX</td><td>R,WX</td><td>R,WX</td></tr>
<tr><td>Protection Attribute</td><td colspan="8">-</td></tr>
<tr><td>Initial Value</td><td colspan="5">00000</td><td>0</td><td>0</td><td>0</td></tr>
</table>
(Correct)
<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>IRQ_RLT2</td><td>IRQ_RLT1</td><td>IRQ_RLT0</td></tr>
<tr><td>R/W Attribute</td><td colspan="5">R0,WX</td><td>R,WX</td><td>R,WX</td><td>R,WX</td></tr>
<tr><td>Protection Attribute</td><td colspan="8">-</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>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>439</td><td>CHAPTER 6:Low Power Consumption 5.9.3. MCU_Config_Group Interrupt Request Status Register2 (MCG_IRSR2)</td><td>"R/W Attribute" should be modified 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="6">Reserved</td><td>IRQ_CRCAL</td><td>IRQ_EICU</td></tr>
<tr><td>R/W Attribute</td><td colspan="6">R,WX</td><td>R,WX</td><td>R,WX</td></tr>
<tr><td>Protection Attribute</td><td colspan="8">-</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>IRQ_CRCAL</td><td>IRQ_EICU</td></tr>
<tr><td>R/W Attribute</td><td colspan="6">R0,WX</td><td>R,WX</td><td>R,WX</td></tr>
<tr><td>Protection Attribute</td><td colspan="8">-</td></tr>
<tr><td>Initial Value</td><td colspan="6">000000</td><td>0</td><td>0</td></tr>
</table></td></tr>
<tr><td>441</td><td>CHAPTER 6:Low Power Consumption 5.9.3. MCU_Config_Group Interrupt Request Status Register2 (MCG_IRSR2)</td><td>"[bit0] IRQ_RAMDE" should be modified as indicated by the shading below.

(Error)
[bit0] IRQ_RAMDE: BackUP-RAM double-bit error interrupt status

(Correct)
[bit0] NMI_RAMDE: BackUP-RAM double-bit error interrupt status (NMI level)</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>442</td><td>CHAPTER 6:Low Power Consumption 5.9.4. MCU_Config_Group Interrupt Request Status Register3 (MCG_IRSR3)</td><td>"R/W Attribute" should be modified 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>Reserved</td><td>IRQ_MCAN2_INT1</td><td>IRQ_MCAN1_INT1</td><td>IRQ_MCAN0_INT1</td><td>Reserved</td><td>IRQ_MCAN2_INT0</td><td>IRQ_MCAN1_INT0</td><td>IRQ_MCAN0_INT0</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">-</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>Reserved</td><td>IRQ_MCAN2_INT1</td><td>IRQ_MCAN1_INT1</td><td>IRQ_MCAN0_INT1</td><td>Reserved</td><td>IRQ_MCAN2_INT0</td><td>IRQ_MCAN1_INT0</td><td>IRQ_MCAN0_INT0</td></tr>
<tr><td>R/W Attribute</td><td>R0,WX</td><td>R,WX</td><td>R,WX</td><td>R,WX</td><td>R0,WX</td><td>R,WX</td><td>R,WX</td><td>R,WX</td></tr>
<tr><td>Protection Attribute</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>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>442</td><td>CHAPTER 6:Low Power Consumption 5.9.4. MCU_Config_Group Interrupt Request Status Register3 (MCG_IRSR3)</td><td>"R/W Attribute" and "Field" should be modified as indicated by the shading below.

(Error)
<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>IRQ_MCAN2_SE</td><td>IRQ_MCAN1_SE</td><td>IRQ_MCAN0_SE</td><td>Reserved</td><td>IRQ_MCAN2_DE</td><td>IRQ_MCAN1_DE</td><td>IRQ_MCAN0_DE</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">-</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>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>IRQ_MCAN2_SE</td><td>IRQ_MCAN1_SE</td><td>IRQ_MCAN0_SE</td><td>Reserved</td><td>NMI_MCAN2_DE</td><td>NMI_MCAN1_DE</td><td>NMI_MCAN0_DE</td></tr>
<tr><td>R/W Attribute</td><td>R0,WX</td><td>R,WX</td><td>R,WX</td><td>R,WX</td><td>R0,WX</td><td>R,WX</td><td>R,WX</td><td>R,WX</td></tr>
<tr><td>Protection Attribute</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>442</td><td>CHAPTER 6:Low Power Consumption 5.9.4. MCU_Config_Group Interrupt Request Status Register3 (MCG_IRSR3)</td><td>"R/W Attribute" should be modified 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="5">Reserved</td><td>IRQ_MFS2_SIRQ</td><td>IRQ_MFS1_SIRQ</td><td>IRQ_MFS0_SIRQ</td></tr>
<tr><td>R/W Attribute</td><td colspan="5">R,WX</td><td>R,WX</td><td>R,WX</td><td>R,WX</td></tr>
<tr><td>Protection Attribute</td><td colspan="8">-</td></tr>
<tr><td>Initial Value</td><td colspan="5">00000</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="5">Reserved</td><td>IRQ_MFS2_SIRQ</td><td>IRQ_MFS1_SIRQ</td><td>IRQ_MFS0_SIRQ</td></tr>
<tr><td>R/W Attribute</td><td colspan="5">R0,WX</td><td>R,WX</td><td>R,WX</td><td>R,WX</td></tr>
<tr><td>Protection Attribute</td><td colspan="8">-</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>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>442</td><td>CHAPTER 6:Low Power Consumption 5.9.4. MCU_Config_Group Interrupt Request Status Register3 (MCG_IRSR3)</td><td>"R/W Attribute" should be modified 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>Reserved</td><td>IRQ_MFS2_RIRQ</td><td>IRQ_MFS1_RIRQ</td><td>IRQ_MFS0_RIRQ</td><td>Reserved</td><td>IRQ_MFS2_TIRQ</td><td>IRQ_MFS1_TIRQ</td><td>IRQ_MFS0_TIRQ</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">-</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>Reserved</td><td>IRQ_MFS2_RIRQ</td><td>IRQ_MFS1_RIRQ</td><td>IRQ_MFS0_RIRQ</td><td>Reserved</td><td>IRQ_MFS2_TIRQ</td><td>IRQ_MFS1_TIRQ</td><td>IRQ_MFS0_TIRQ</td></tr>
<tr><td>R/W Attribute</td><td>R0,WX</td><td>R,WX</td><td>R,WX</td><td>R,WX</td><td>R0,WX</td><td>R,WX</td><td>R,WX</td><td>R,WX</td></tr>
<tr><td>Protection Attribute</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>444</td><td>CHAPTER 6:Low Power Consumption 5.9.4. MCU_Config_Group Interrupt Request Status Register3 (MCG_IRSR3)</td><td>"[bit18] IRQ_MCAN2_DE" should be modified as indicated by the shading below.

(Error)
[bit18] IRQ_MCAN2_DE: CAN-FD ch2 RAM ECC double-bit error detection interrupt status

(Correct)
[bit18] NMI_MCAN2_DE: CAN-FD ch2 RAM ECC double-bit error detection interrupt status (NMI level)</td></tr>
<tr><td>444</td><td>CHAPTER 6:Low Power Consumption 5.9.4. MCU_Config_Group Interrupt Request Status Register3 (MCG_IRSR3)</td><td>"[bit17] IRQ_MCAN1_DE" should be modified as indicated by the shading below.

(Error)
[bit17] IRQ_MCAN1_DE: CAN-FD ch1 RAM ECC double-bit error detection interrupt status

(Correct)
[bit17] NMI_MCAN1_DE: CAN-FD ch1 RAM ECC double-bit error detection interrupt status (NMI level)</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>444</td><td>CHAPTER 6:Low Power Consumption 5.9.4. MCU_Config_Group Interrupt Request Status Register3 (MCG_IRSR3)</td><td>"[bit16] IRQ_MCAN0_DE" should be modified as indicated by the shading below.

(Error)
[bit16] IRQ_MCAN0_DE: CAN-FD ch0 RAM ECC double-bit error detection interrupt status

(Correct)
[bit16] NMI_MCAN0_DE: CAN-FD ch0 RAM ECC double-bit error detection interrupt status (NMI level)</td></tr>
<tr><td>447</td><td>CHAPTER 6:Low Power Consumption 5.9.5. MCU_Config_Group Interrupt Request Status Register4 (MCG_IRSR4)</td><td>"R/W Attribute" should be modified 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 colspan="8">R,WX</td></tr>
<tr><td>Protection Attribute</td><td colspan="8">-</td></tr>
<tr><td>Initial Value</td><td colspan="8">00000000</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,WX</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>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>447</td><td>CHAPTER 6:Low Power Consumption 5.9.5. MCU_Config_Group Interrupt Request Status Register4 (MCG_IRSR4)</td><td>"R/W Attribute" should be modified as indicated by the shading below.

(Error)
<table>
<tr><td>bit</td><td>23</td><td>22</td><td>22</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">R,WX</td></tr>
<tr><td>Protection Attribute</td><td colspan="8">-</td></tr>
<tr><td>Initial Value</td><td colspan="8">00000000</td></tr>
</table>
(Correct)
<table>
<tr><td>bit</td><td>23</td><td>33</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,WX</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>
<tr><td>447</td><td>CHAPTER 6:Low Power Consumption 5.9.5. MCU_Config_Group Interrupt Request Status Register4 (MCG_IRSR4)</td><td>"R/W Attribute" should be modified 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 colspan="8">R,WX</td></tr>
<tr><td>Protection Attribute</td><td colspan="8">-</td></tr>
<tr><td>Initial Value</td><td colspan="8">00000000</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,WX</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>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>447</td><td>CHAPTER 6:Low Power Consumption 5.9.5. MCU_Config_Group Interrupt Request Status Register4 (MCG_IRSR4)</td><td>"R/W Attribute" should be modified 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>Reserved</td><td>IRQ_MFS2_REIRQ</td><td>IRQ_MFS1_REIRQ</td><td>IRQ_MFS0_REIRQ</td><td>Reserved</td><td>IRQ_MFS2_TEIRQ</td><td>IRQ_MFS1_TEIRQ</td><td>IRQ_MFS0_TEIRQ</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">-</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>Reserved</td><td>IRQ_MFS2_REIRQ</td><td>IRQ_MFS1_REIRQ</td><td>IRQ_MFS0_REIRQ</td><td>Reserved</td><td>IRQ_MFS2_TEIRQ</td><td>IRQ_MFS1_TEIRQ</td><td>IRQ_MFS0_TEIRQ</td></tr>
<tr><td>R/W Attribute</td><td>R0,WX</td><td>R,WX</td><td>R,WX</td><td>R,WX</td><td>R0,WX</td><td>R,WX</td><td>R,WX</td><td>R,WX</td></tr>
<tr><td>Protection Attribute</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>450</td><td>CHAPTER 6:Low Power Consumption 5.10.1. Protection Key Setting Register (SYSC1_PROTKEYR)</td><td>The Note should be deleted as indicated by the shading below.

(Error)
＜Notes＞
− After the SYSC1 area is written, the protection key is locked again regardless of the master.
− Only word access is valid to this register. Any other access is invalid.

(Correct)
− Only word access is valid to this register. Any other access is invalid.</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>474</td><td>CHAPTER 6:Low Power Consumption 5.11.7. RUN Clock Divider Register 0 (SYSC1_RUNCKDIVR0)</td><td>"[bit20:16] TRCDIV " should be modified as indicated by the shading below.

(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>0_0000</td><td>No division</td></tr>
<tr><td>0_0001</td><td>Divided by 2</td></tr>
<tr><td>0_0010</td><td>Divided by 3</td></tr>
<tr><td>0_0011</td><td>Divided by 4</td></tr>
<tr><td>・・・</td><td>・・・</td></tr>
<tr><td>1_1101</td><td>Divided by 30</td></tr>
<tr><td>1_1110</td><td>Divided by 31</td></tr>
<tr><td>1_1111</td><td>Divided by 32</td></tr>
</table></td></tr>
<tr><td>488</td><td>CHAPTER 6:Low Power Consumption 5.11.12. RUN Clock Divider Register 5 (SYSC1_RUNCKDIVR5)</td><td>"[bit11:8] CD1B0DIV" should be modified as indicated by the shading below.

(Error) [bit11:8] CD1B0DIV: CD1A0 clock divider setting bits

(Correct) [bit11:8] CD1A0DIV: CD1A0 clock divider setting bits</td></tr>
<tr><td>539</td><td>CHAPTER 6:Low Power Consumption 5.12.12. PSS Clock Divider Register 5 (SYSC1_PSSCKDIVR5)</td><td>"[bit11:8] CD1B0DIV" should be modified as indicated by the shading below.

(Error) [bit11:8] CD1B0DIV: CD1A0 clock divider setting bits

(Correct) [bit11:8] CD1A0DIV: CD1A0 clock divider setting bits</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>587</td><td>CHAPTER 6:Low Power Consumption
5.13.11. APP Clock Divider Register 4 (SYSC1_APPCKDIVR4)</td><td>"Initial Value" should be modified as indicated by the shading below.

(Error)
<table>
<tr><td>bit</td><td>31</td><td>8</td></tr>
<tr><td>Field</td><td colspan="2">Reserved</td></tr>
<tr><td>R/W Attribute</td><td colspan="2">R0,WX</td></tr>
<tr><td>Protection Attribute</td><td colspan="2">WPS</td></tr>
<tr><td>Initial Value</td><td colspan="2">00000000</td></tr>
</table>

(Correct)
<table>
<tr><td>bit</td><td>31</td><td>8</td></tr>
<tr><td>Field</td><td colspan="2">Reserved</td></tr>
<tr><td>R/W Attribute</td><td colspan="2">R0,WX</td></tr>
<tr><td>Protection Attribute</td><td colspan="2">WPS</td></tr>
<tr><td>Initial Value</td><td colspan="2">00000000_00000000_00000000</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>631</td><td>CHAPTER 6:Low Power Consumption 5.14.7. STS Clock Divider Register 0 (SYSC1_STSCKDIVR0)</td><td>"[bit20:16] TRCDIV: " should be modified as indicated by the shading below.

(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>0_0000</td><td>No division</td></tr>
<tr><td>0_0001</td><td>Divided by 2</td></tr>
<tr><td>0_0010</td><td>Divided by 3</td></tr>
<tr><td>0_0011</td><td>Divided by 4</td></tr>
<tr><td>・・・</td><td>・・・</td></tr>
<tr><td>1_1101</td><td>Divided by 30</td></tr>
<tr><td>1_1110</td><td>Divided by 31</td></tr>
<tr><td>1_1111</td><td>Divided by 32</td></tr>
</table>
</td></tr>
</table>

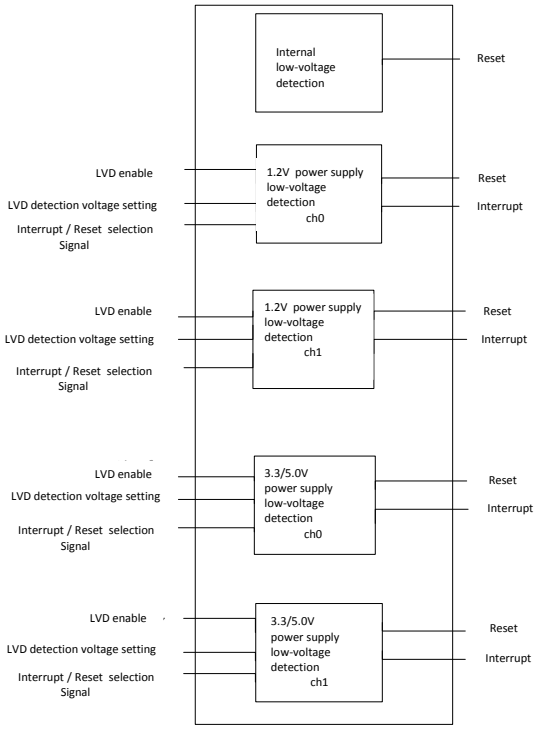
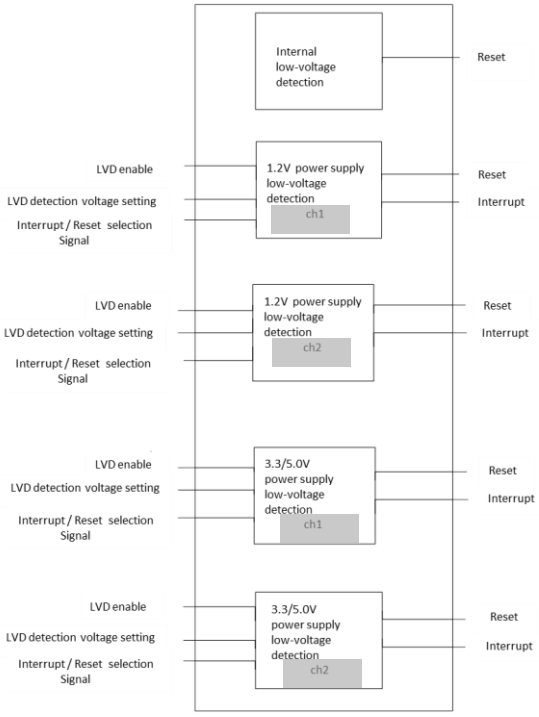
<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>664</td><td>CHAPTER 7:Low-Voltage Detection
1. Overview</td><td>“Table 7-3 LVD operation select bits " should be corrected as indicated by the shading below.

(Error)
<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)
<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>
<tr><td>664</td><td>CHAPTER 7:Low-Voltage Detection
1. Overview</td><td>“Table 7-4 LVD ready bits " should be corrected as indicated by the shading below.

(Error)
<table><tr><td>LVDL0RDY</td><td>1.2 V power supply low-voltage detection ready bit ch0</td></tr><tr><td>LVDL1RDY</td><td>1.2 V power supply low-voltage detection ready bit ch1</td></tr><tr><td>LVDH0RDY</td><td>3.3/5.0 V power supply low-voltage detection ready bit ch0</td></tr><tr><td>LVDH1RDY</td><td>3.3/5.0 V power supply low-voltage detection ready bit ch1</td></tr></table>
(Correct)
<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></td></tr>
<tr><td>664</td><td>CHAPTER 7:Low-Voltage Detection
1. Overview</td><td>“Table 7-5 LVD reset source bits " should be corrected as indicated by the shading below.

(Error)
<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>LVDH1E</td><td>3.3/5.0 V power supply low-voltage detection operation enable bit ch1</td></tr></table>
(Correct)
<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>
</table>

Page	Section	Change Results
664	CHAPTER 7:Low-Voltage Detection 1. Overview	Features of "For details" should be corrected as indicated by the shading below. (Error) Notes: - In the S6J3110 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 - In the S6J3110 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. - Ch.1 is not implemented - LVD specification is a target value (Correct) Notes: - In the S6J3110 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 - In the S6J3110 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. -Ch.2 is not implemented

Page	Section	Change Results
665	CHAPTER 7:Low-Voltage Detection 2. Configuration and Block Diagrams	"Figure2-1" should be corrected as indicated by the shading below. (Error)  (Correct) 

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>673</td><td>CHAPTER 7:Low-Voltage Detection
5. Operation Examples</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)

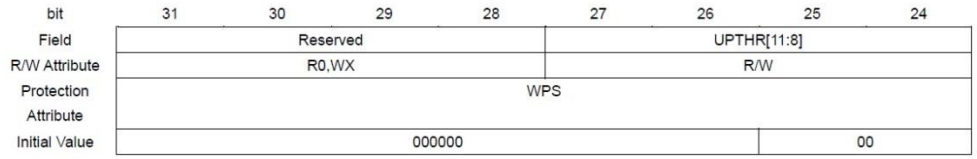
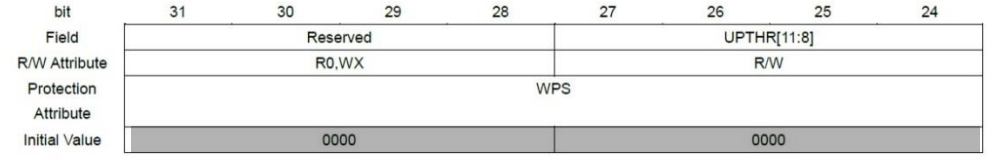
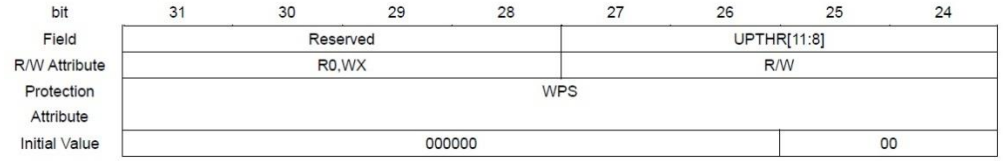
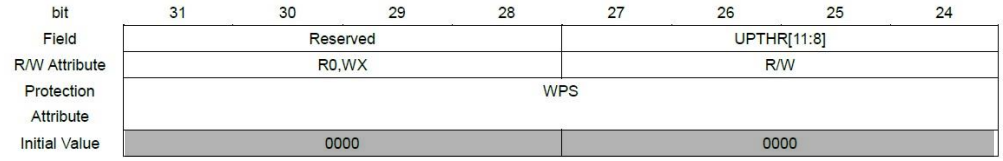
<table>
<tr><td>LVDL0E : low voltage detect enable for 1.2V power supply, ch0</td></tr>
<tr><td>LVDL1E : low voltage detect enable for 1.2V power supply, ch1</td></tr>
<tr><td>LVDH0E : low voltage detect enable for 3.3V or 5.5V power supply, ch0</td></tr>
<tr><td>LVDH1E : low voltage detect enable for 3.3V or 5.5V power supply, ch1</td></tr>
</table>

(Correct)
<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>
</td></tr>
<tr><td>702</td><td>CHAPTER 8:Clock Supervisor
5.9. PLL2 Clock Supervisor Setting Register 0 (SYSC_CSVPLL2CFG R0)</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="4">Reserved</td><td colspan="4">UPTHR[11:8]</td></tr>
<tr><td>R/W Attribute</td><td colspan="4">R0,WX</td><td colspan="4">R/W</td></tr>
<tr><td>Protection Attribute</td><td colspan="8">WPS</td></tr>
<tr><td>Initial Value</td><td colspan="6">000000</td><td colspan="2">00</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="4">Reserved</td><td colspan="4">UPTHR[11:8]</td></tr>
<tr><td>R/W Attribute</td><td colspan="4">R0,WX</td><td colspan="4">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 colspan="4">0000</td></tr>
</table>
</td></tr>
</table>

Page	Section	Change Results
714	CHAPTER 8:Clock Supervisor 5.15. SSCG PLL1 Clock Supervisor Setting Register 0 (SYSC_CSVSP1CFGR0)	Features of "For details" should be corrected as indicated by the shading below. (Error)  (Correct) 
718	CHAPTER 8:Clock Supervisor 5.17. SSCG PLL2 Clock Supervisor Setting Register 0 (SYSC_CSVSP2CFGR0)	Features of "For details" should be corrected as indicated by the shading below. (Error)  (Correct) 

Page	Section	Change Results
727	CHAPTER 8:Clock Supervisor 5.22. Clock Supervisor Test Register (SYSC_CSVTESTR)	Features of "For details" should be corrected as indicated by the shading below. (Error) 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: 000 | 0 | 0 | 0 | 0 | 0 (Correct) 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

Page	Section	Change Results
730	CHAPTER 8:Clock Supervisor 6. Precautions for Using This Device	Features of "For details" should be corrected as indicated by the shading below. (Error) 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 monitoring clock and reference clock. - To stop the clock supervisor, simultaneously stop oscillation of the monitoring clock. - After a reset is generated by the detection of a clock abnormality, the main clock supervisor reference clock returns to the high-speed CR clock. The clock of the detected abnormality must not be selected again. - 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. (Correct) 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 monitoring clock and reference clock. - To stop the clock supervisor, simultaneously stop oscillation of the monitoring clock. - After a reset is generated by the detection of a clock abnormality, MCU will operate with FastCR clock. At this time, CSV cause clock must not be selected again - 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.
732	CHAPTER 9: Source Clock Timer 1. Overview	Revised oscillation stabilization wait time from “0.48ms” to “0.88ms”.

Page	Section	Change Results
778	CHAPTER 10: Real Time Clock 3. Operation of the Real Time Clock	Features of "For details" should be corrected as indicated by the shading below. (Error) **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 countermeasures full clock cycles of CLKCAL and returns the value in RTC_CNTCAL. (Correct) **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.

Page	Section	Change Results
781	CHAPTER 10: Real Time Clock 3. Operation of the Real Time Clock	Features of "For details" should be added as indicated by the shading below. (Error) 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) desserts 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_CALTRIG) value defines the duration between two Calibrations. (Correct) 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) desserts 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_CALTRIG) value defines the duration between two Calibrations.
783	CHAPTER 10: Real Time Clock 4.1. Timer Control Register (RTC_WTCR)	Revised ACCESS_TYPE of bit 10 form “R/W” to “R0, W”

Page	Section	Change Results
784	CHAPTER 10: Real Time Clock 4.1. Timer Control Register (RTC_WTCR)	Features of "For details" should be deleted as indicated by the shading below. (Error) 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. ----------SNIP ---------- [bit9] MTRG : Manual Trigger for Calibration This bit is used to trigger a software calibration cycle. (Correct) 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. [bit9] MTRG : Manual Trigger for Calibration This bit is used to trigger a software calibration cycle.
785	CHAPTER 10: Real Time Clock 4.1. Timer Control Register (RTC_WTCR)	Features of "For details" should be added as indicated by the shading below. (Correct) **[bit5:4] RCKSEL[1:0]: Clock Select for RTC** ***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.*
786	CHAPTER 10: Real Time Clock 4.1. Timer Control Register (RTC_WTCR)	Features of "For details" should be added as indicated by the shading below. (Correct) **[bit3] CSM: Clock Switching Mode** ***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.*

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>787</td><td>CHAPTER 10: Real Time Clock
4.1. Timer Control Register (RTC_WTCR)</td><td>
Features of "For details" should be added as indicated by the shading below.

(Error)
[bit0] ST : Start
This bit is used to start/stop the RTC.
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>The Real Time Clock module stops operating, and the Sub-second counter and the Hour/Minute/Second counters are stopped</td></tr>
<tr><td>1</td><td>The settings of the Hour/Minute/Second registers are loaded to the Hour/Minute/Second counters, and the RTC module starts to operate</td></tr>
</table>

(Correct)
[bit0] ST : Start
This bit is used to start/stop the RTC.
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>The Real Time Clock module stops operating, and the Sub-second counter and the Hour/Minute/Second counters are stopped</td></tr>
<tr><td>1</td><td>The settings of the Hour/Minute/Second registers are loaded to the Hour/Minute/Second counters, and the RTC module starts to operate</td></tr>
</table>
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.
</td></tr>
<tr><td>788</td><td>CHAPTER 10:Real Time Clock
4.2. Timer Status Register (RTC_WTSR)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
4.2.Timer Status Register (RTC_WTSR)
<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 colspan="6">Reserved</td><td>CSF</td><td>RUN</td></tr>
<tr><td>ACCESS_TYPE</td><td colspan="6">R0,WX</td><td>R,W</td><td>R,WX</td></tr>
<tr><td>PROT_TYPE</td><td colspan="8">-</td></tr>
<tr><td>INITIAL_VALUE</td><td colspan="6">000000</td><td>0</td><td>0</td></tr>
</table>

(Correct)
4.2.Timer Status Register (RTC_WTSR)
<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>CLK_STS</td><td colspan="5">Reserved</td><td>CSF</td><td>RUN</td></tr>
<tr><td>ACCESS_TYPE</td><td>R,WX</td><td colspan="5">R0,WX</td><td>R,W</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 colspan="5">00000</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>788,
789</td><td>CHAPTER 10:Real Time Clock
4.2. Timer Status Register (RTC_WTSR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)

[bit7:2] Reserved

(Correct)

[bit7] CLK_STS: Clock Switching Status
This bit indicates the status of the clock switching.
<table><tr><th>Bit</th><th>Description</th></tr><tr><td>0</td><td>The clock switching is inactive</td></tr><tr><td>1</td><td>The clock switching is in progress</td></tr></table>
[bit6:2] Reserved</td></tr>
<tr><td>798</td><td>CHAPTER 10:Real Time Clock
4.6 Sub-Second Register (RTC_WTBR)</td><td>Features of "For details" should be added as indicated by the shading below.

(Error)
Note:
The required value depends on the RTC clock that is used (CLKRTC) and can be calculated with the formula on page 418

(Correct)
Note:
The required value depends on the RTC clock that is used (CLKRTC) and can be calculated with the formula on the following page.</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>802</td><td>CHAPTER 10:Real Time Clock
4.8 Real time clock real time register1 (RTC_RTR1)</td><td>Features of "For details" should be added 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 colspan="8">WTDR[15:8]</td></tr>
<tr><td>ACCESS_TYPE</td><td colspan="8">R/W</td></tr>
<tr><td>PROT_TYPE</td><td colspan="8">-</td></tr>
<tr><td>INITIAL_VALUE</td><td colspan="8">00000000</td></tr>
</table>

<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 colspan="8">WTDR[7:0]</td></tr>
<tr><td>ACCESS_TYPE</td><td colspan="8">R/W</td></tr>
<tr><td>PROT_TYPE</td><td colspan="8">-</td></tr>
<tr><td>INITIAL_VALUE</td><td colspan="8">00000000</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 colspan="8">WTDR[15:8]</td></tr>
<tr><td>ACCESS_TYPE</td><td colspan="8">R,W</td></tr>
<tr><td>PROT_TYPE</td><td colspan="8">-</td></tr>
<tr><td>INITIAL_VALUE</td><td colspan="8">00000000</td></tr>
</table>

<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 colspan="8">WTDR[7:0]</td></tr>
<tr><td>ACCESS_TYPE</td><td colspan="8">R,W</td></tr>
<tr><td>PROT_TYPE</td><td colspan="8">-</td></tr>
<tr><td>INITIAL_VALUE</td><td colspan="8">00000000</td></tr>
</table>
</td></tr>
<tr><td>802</td><td>CHAPTER 10:Real Time Clock
4.8 Real time clock real time register1 (RTC_RTR1)</td><td>Features of "For details" should be added as indicated by the shading below.

(Error)
Notes:
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.

(Correct)
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.</td></tr>
</table>

Page	Section	Change Results
810	CHAPTER 10:Real Time Clock 4.13 Debug Register (RTC_DEBUG)	Features of "For details" should be added as indicated by the shading below. [bit0] DBGEN : Debug Enable (Error) 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™-R4 Technical Reference Manual. (Correct) 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.
814	CHAPTER 10:Real Time Clock 5. Usage Precaution	Features of "For details" should be corrected as indicated by the shading below. (Error) 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 inoperation (Correct) 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
814	CHAPTER 10:Real Time Clock 5. Usage Precaution	Features of "For details" should be deleted as indicated by the shading below. (Error) ■ RTC cannot use the calibration function under following condition: RTC clock*2 < Bus clock

Page	Section	Change Results
822	CHAPTER 11:CR Calibration 3.2 CR Clock Frequency Correction	Features of "For details" should be added as indicated by the shading below. (Error) 11. 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 the frequency of FB with FA and then repeat the step of 11. 12. 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 the frequency of FB with FA and then repeat the step of 12. (Correct) 11. 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. 12. 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.
837	CHAPTER 12: Backup RAM Interface 3.1. RAMECC Function	"Test mode" 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).

Page	Section	Change Results
839	CHAPTER 12: Backup RAM Interface 3.2. RAM Diagnosis	“RAM diagnosis procedure " should be corrected as indicated by the shading below. (Error) 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_TICR.TCI. (Correct) 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.
841	CHAPTER 12: Backup RAM Interface 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. (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.
841	CHAPTER 12: Backup RAM Interface 3.3. RAM Initialization	Features of "For details" should be corrected as indicated by the shading below. (Error) 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". (Correct) 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".

Page	Section	Change Results
841	CHAPTER 12: Backup RAM Interface 3.3. RAM Initialization	Features of "For details" should be corrected as indicated by the shading below. (Error) 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. (Correct) 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.
841	CHAPTER 12: Backup RAM Interface 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. (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.

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>847</td><td>CHAPTER 12: Backup RAM Interface
5. Registers</td><td>Features of "For details" should be added as indicated by the shading below.
(Error)

Table 5-1 BACKUP-RAM Interface Registers
<table><tr><th>Abbreviated Register Name</th><th>Register Name</th><th>Reference</th></tr><tr><td>BURIF_UNLOCK</td><td>BURAMIF unlock register</td><td>5.1</td></tr><tr><td>BURIF_STATUS</td><td>BURAMIF status register</td><td>5.2</td></tr><tr><td>BURIF_DEEAR</td><td>BURAMIF double-bit ECC error address register</td><td>5.3</td></tr><tr><td>BURIF_SEEAR</td><td>BURAMIF single-bit ECC error address register</td><td>5.4</td></tr><tr><td>BURIF_EFEAR</td><td>BURAMIF ECC pseudo-error generation address register</td><td>5.5</td></tr><tr><td>BURIF_EECSR</td><td>BURAMIF ECC error control register</td><td>5.6</td></tr><tr><td>BURIF_EFECR</td><td>BURAMIF ECC pseudo-error generation control register</td><td>5.7</td></tr><tr><td>BURIF_TEAR0</td><td>BURAMIF TEST error address register 0</td><td>5.8</td></tr><tr><td>BURIF_TEAR1</td><td>BURAMIF TEST error address register 1</td><td>5.9</td></tr><tr><td>BURIF_TEAR2</td><td>BURAMIF TEST error address register 2</td><td>5.10</td></tr><tr><td>BURIF_TASAR</td><td>BURAMIF TEST start address register</td><td>5.11</td></tr><tr><td>BURIF_TAEAR</td><td>BURAMIF TEST end address register</td><td>5.12</td></tr><tr><td>BURIF_TTCR</td><td>BURAMIF TEST diagnosis function register</td><td>5.13</td></tr><tr><td>BURIF_TICR</td><td>BURAMIF TEST initialization function register</td><td>5.14</td></tr><tr><td>BURIF_TFECR</td><td>BURAMIF TEST pseudo-error generation control register</td><td>5.15</td></tr><tr><td>BURIF_TKCCR</td><td>BURAMIF TEST key code control register</td><td>5.16</td></tr><tr><td>BURIF_TSRCR</td><td>BURAMIF TEST software reset generation control register</td><td>5.17</td></tr><tr><td>BURIF_EVENTCLR</td><td>BURAMIF event clear register</td><td>5.18</td></tr></table>
(Correct)

Table 5-1 BACKUP-RAM Interface Registers
<table><tr><th>Abbreviated Register Name</th><th>Register Name</th><th>Reference</th></tr><tr><td>BURIF_UNLOCK</td><td>BURAMIF unlock register</td><td>5.1</td></tr><tr><td>BURIF_STATUS</td><td>BURAMIF status register</td><td>5.2</td></tr><tr><td>BURIF_DEEAR</td><td>BURAMIF double-bit ECC error address register</td><td>5.3</td></tr><tr><td>BURIF_SEEAR</td><td>BURAMIF single-bit ECC error address register</td><td>5.4</td></tr><tr><td>BURIF_EFEAR</td><td>BURAMIF ECC pseudo-error generation address register</td><td>5.5</td></tr><tr><td>BURIF_EECSR</td><td>BURAMIF ECC error control register</td><td>5.6</td></tr><tr><td>BURIF_EFECR</td><td>BURAMIF ECC pseudo-error generation control register</td><td>5.7</td></tr><tr><td>BURIF_TEAR0</td><td>BURAMIF TEST error address register 0</td><td>5.8</td></tr><tr><td>BURIF_TEAR1</td><td>BURAMIF TEST error address register 1</td><td>5.9</td></tr><tr><td>BURIF_TEAR2</td><td>BURAMIF TEST error address register 2</td><td>5.10</td></tr><tr><td>BURIF_TASAR</td><td>BURAMIF TEST start address register</td><td>5.11</td></tr><tr><td>BURIF_TAEAR</td><td>BURAMIF TEST end address register</td><td>5.12</td></tr><tr><td>BURIF_TTCR</td><td>BURAMIF TEST diagnosis function register</td><td>5.13</td></tr><tr><td>BURIF_TICR</td><td>BURAMIF TEST initialization function register</td><td>5.14</td></tr><tr><td>BURIF_TFECR</td><td>BURAMIF TEST pseudo-error generation control register</td><td>5.16</td></tr><tr><td>BURIF_TKCCR</td><td>BURAMIF TEST key code control register</td><td>5.17</td></tr><tr><td>BURIF_TSRCR</td><td>BURAMIF TEST software reset generation control register</td><td>5.15</td></tr><tr><td>BURIF_EVENTCLR</td><td>BURAMIF event clear register</td><td>5.18</td></tr></table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>848</td><td>CHAPTER 12: Backup RAM Interface
5.1 BURAMIF Unlock Register
(BURIF_UNLOCK)</td><td>Features of "For details" should be deleted as indicated by the shading below.

Delete the "Register configuration" and "Register functions".

(Error)

Register configurations
<table><tr><td>Bit</td><td>31</td><td>0</td></tr><tr><td>Field</td><td colspan="2">UNLOCK[31:0]</td></tr><tr><td>R/W Attribute</td><td colspan="2">R0, W</td></tr><tr><td>Protection Attribute</td><td colspan="2">WP</td></tr><tr><td>Initial Value</td><td colspan="2">00000000_00000000_00000000_00000000</td></tr></table>
Register functions

(Correct)

<table><tr><td>Bit</td><td>31</td><td>0</td></tr><tr><td>Field</td><td colspan="2">UNLOCK[31:0]</td></tr><tr><td>R/W Attribute</td><td colspan="2">R0,W</td></tr><tr><td>Protection Attribute</td><td colspan="2">WP</td></tr><tr><td>Initial Value</td><td colspan="2">00000000_00000000_00000000_00000000</td></tr></table></td></tr>
<tr><td>849</td><td>CHAPTER 12: Backup RAM Interface
5.2 BURAMIF Status Register
(BURIF_STATUS)</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 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">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 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>

Page	Section	Change Results
849	CHAPTER 12:BACKUP RAM 5.2 BURAMIF Status Register (BURIF_STATUS)	Features of "For details" should be deleted as indicated by the shading below. Delete the "Register configuration" and "Register functions". Same 5.1 BURAMIF Unlock Register (BURIF_UNLOCK) (Error) Register configuration Register functions".
850	CHAPTER 12: Backup RAM Interface 5.3 BURAMIF Double-Bit ECC Error Address Register (BURIF_DEEAR)	Features of "For details" should be deleted as indicated by the shading below. Delete the "Register configuration "and "Register functions". Same 5.1 BURAMIF Unlock Register (BURIF_UNLOCK) (Error) Register configuration Register functions".
851	CHAPTER 12: Backup RAM Interface 5.4 BURAMIF Single-bit ECC Error Address Register (BURIF_SEEAR)	Features of "For details" should be deleted as indicated by the shading below. Delete the "Register configuration "and "Register functions". Same 5.1 BURAMIF Unlock Register (BURIF_UNLOCK) (Error) Register configuration Register functions".
852	CHAPTER 12: Backup RAM Interface 5.5 BURAMIF ECC Pseudo-error Generation Address Register (BURIF_EFEAR)	Features of "For details" should be deleted as indicated by the shading below. Delete the "Register configuration "and "Register functions". Same 5.1 BURAMIF Unlock Register (BURIF_UNLOCK) (Error) Register configuration Register functions".
852	CHAPTER 12: Backup RAM Interface 5.5 BURAMIF ECC Pseudo-error Generation Address Register (BURIF_EFEAR)	Features of "For details" should be added 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

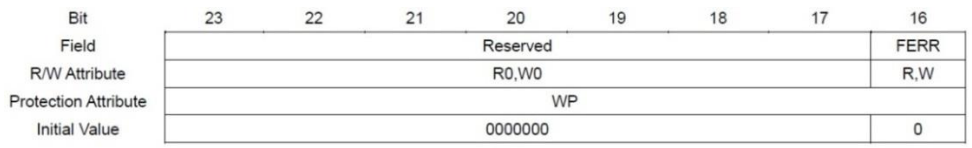
<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>853</td><td>CHAPTER 12: Backup RAM Interface
5.6 BURAMIF ECC Error Control Register (BURIF_EECSR)</td><td>Features of "For details" should be deleted as indicated by the shading below.

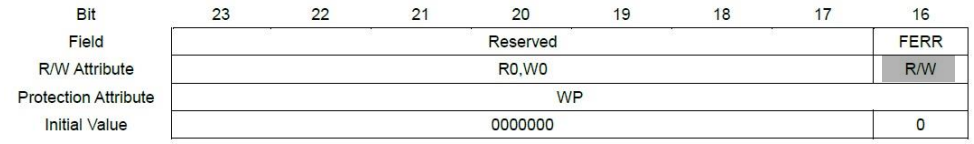
Delete the "Register configuration "and "Register functions".
Same 5.1 BURAMIF Unlock Register (BURIF_UNLOCK)

(Error)
Register configuration
Register functions".</td></tr>
<tr><td>854</td><td>CHAPTER 12: Backup RAM Interface
5.7 BURAMIF ECC Pseudo-error Generation Control Register (BURIF_EFECR)</td><td>Features of "For details" should be deleted as indicated by the shading below.

Delete the "Register configuration "and "Register functions".
Same 5.1 BURAMIF Unlock Register (BURIF_UNLOCK)

(Error)
Register configuration
Register functions".</td></tr>
<tr><td>854</td><td>CHAPTER 12: Backup RAM Interface
5.7 BURAMIF ECC Pseudo-error Generation Control Register (BURIF_EFECR)</td><td>Features of "For details" should be deleted as indicated by the shading below.

(Error)
 
<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="7">Reserved</td><td>FERR</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="8">WP</td></tr>
<tr><td>Initial Value</td><td colspan="7">0000000</td><td>0</td></tr>
</table>

(Correct)
 
<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="7">Reserved</td><td>FERR</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="8">WP</td></tr>
<tr><td>Initial Value</td><td colspan="7">0000000</td><td>0</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>854</td><td>CHAPTER 12: Backup RAM Interface
5.7 BURAMIF ECC Pseudo-error Generation Control Register (BURIF_EFECR)</td><td>"[bit16] FERR" 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>856</td><td>CHAPTER 12: Backup RAM Interface
5.8 BURAMIF TEST Error Address Register 0 (BURIF_TEAR0)</td><td>Features of "For details" should be deleted as indicated by the shading below.

Delete the "Register configuration "and "Register functions".
Same 5.1 BURAMIF Unlock Register (BURIF_UNLOCK)

(Error)
Register configuration
Register functions".</td></tr>
<tr><td>856</td><td>CHAPTER 12: Backup RAM Interface
5.9 BURAMIF TEST Error Address Register 1 (BURIF_TEAR1)</td><td>Features of "For details" should be deleted as indicated by the shading below.

Delete the "Register configuration "and "Register functions".
Same 5.1 BURAMIF Unlock Register (BURIF_UNLOCK)

(Error)
Register configuration
Register functions".</td></tr>
</table>

Page	Section	Change Results
860	CHAPTER 12: Backup RAM Interface 5.10 BURAMIF TEST Error Address Register 2 (BURIF_TEAR2)	Features of "For details" should be deleted as indicated by the shading below. Delete the "Register configuration" and "Register functions". Same 5.1 BURAMIF Unlock Register (BURIF_UNLOCK) (Error) Register configuration Register functions".
862	CHAPTER 12: Backup RAM Interface 5.11 BURAMIF TEST Start Address Register (BURIF_TASAR)	Features of "For details" should be deleted as indicated by the shading below. Delete the "Register configuration" and "Register functions". Same 5.1 BURAMIF Unlock Register (BURIF_UNLOCK) (Error) Register configuration Register functions".
863	CHAPTER 12: Backup RAM Interface 5.12 BURAMIF TEST End Address Register (BURIF_TAEAR)	Features of "For details" should be deleted as indicated by the shading below. Delete the "Register configuration" and "Register functions". Same 5.1 BURAMIF Unlock Register (BURIF_UNLOCK) (Error) Register configuration Register functions".
863	CHAPTER 12: Backup RAM Interface 5.12 BURAMIF TEST End Address Register (BURIF_TAEAR)	"[bit14:0] END_ADDR" should be corrected as indicated by the shading below. (Error) [bit14:0] END_ADDR: RAM diagnosis end address (Correct) [bit14:0] EADDR: RAM diagnosis end address

Page	Section	Change Results
863	CHAPTER 12: Backup RAM Interface 5.12 BURAMIF TEST End Address Register (BURIF_TAEAR)	Features of "For details" should be deleted as indicated by the shading 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_TASARX.SADDR > BURIF_TAEAR.EADDR cannot be set. - The effective address of EADDR changes according to RAM size. - 8KB Effective address: EADDR[10:0] - 64KB Effective address: EADDR[13:0] - EADDR initial Value changes according to RAM size. - 00000111_11111111 8KB - 00111111_11111111 64KB (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_TASARX.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. - 8KB Effective address: EADDR[10:0] - 64KB Effective address: EADDR[13:0] - EADDR initial Value changes according to RAM size. - 00000111_11111111 8KB - 00111111_11111111 64KB

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>864</td><td>CHAPTER 12: Backup RAM Interface
5.13 BURAMIF TEST Diagnosis Function Register
(BURIF_TTCR)</td><td>Features of "For details" should be deleted 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="6">Reserved</td><td>TSTAT</td><td>OVFLW</td></tr>
<tr><td>R/W Attribute</td><td colspan="6">R0, W0</td><td>R, WX</td><td>R, WX</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>TEIE</td><td>TEI</td><td>TCIE</td><td>TCI</td><td colspan="3">TTYP</td><td>TRUN</td></tr>
<tr><td>R/W Attribute</td><td>R/W</td><td>R, W0</td><td>R/W</td><td>R, W0</td><td colspan="3">R/W</td><td>R, WX</td></tr>
<tr><td>Protection Attribute</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 colspan="3">110</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>TSTAT</td><td>OVFLW</td></tr>
<tr><td>R/W Attribute</td><td colspan="6">R0,W0</td><td>R,WX</td><td>R,WX</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>TEIE</td><td>TEI</td><td>TCIE</td><td>TCI</td><td colspan="3">TTYP</td><td>TRUN</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">R/W</td><td>R,WX</td></tr>
<tr><td>Protection Attribute</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 colspan="3">110</td><td>0</td></tr>
</table>
</td></tr>
<tr><td>864</td><td>CHAPTER 12: Backup RAM Interface
5.13 BURAMIF TEST Diagnosis Function Register
(BURIF_TTCR)</td><td>Features of "For details" should be deleted as indicated by the shading below.

Delete the "Register configuration" and "Register functions".
Same 5.1 BURAMIF Unlock Register (BURIF_UNLOCK)

(Error)
Register configuration
Register functions".</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>867</td><td>CHAPTER 12: Backup RAM Interface
5.14 BURAMIF TEST Initialization Function Register (BURIF_TICR)</td><td>Features of "For details" should be deleted 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 colspan="4">Reserved</td><td>ICIE</td><td>ICI</td><td>ITYP</td><td>IRUN</td></tr>
<tr><td>R/W Attribute</td><td colspan="4">R0, W0</td><td>R/W</td><td>R, W0</td><td>R, W0</td><td>R, WX</td></tr>
<tr><td>Protection Attribute</td><td colspan="8">WP</td></tr>
<tr><td>Initial Value</td><td colspan="4">0000</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 colspan="4">Reserved</td><td>ICIE</td><td>ICI</td><td>ITYP</td><td>IRUN</td></tr>
<tr><td>R/W Attribute</td><td colspan="4">R0,W0</td><td>R/W</td><td>R,W</td><td>R/W</td><td>R,WX</td></tr>
<tr><td>Protection Attribute</td><td colspan="8">WP</td></tr>
<tr><td>Initial Value</td><td colspan="4">0000</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
</td></tr>
<tr><td>867</td><td>CHAPTER 12: Backup RAM Interface
5.14 BURAMIF TEST Initialization Function Register (BURIF_TICR)</td><td>Features of "For details" should be deleted as indicated by the shading below.

Delete the "Register configuration" and "Register functions".
Same 5.1 BURAMIF Unlock Register (BURIF_UNLOCK)

(Error)
Register configuration
Register functions".</td></tr>
<tr><td>868</td><td>CHAPTER 12: Backup RAM Interface
5.15 BURAMIF TEST Software Reset Generation Control Register (BURIF_TSRCR)</td><td>Features of "For details" should be deleted 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>SRST</td><td colspan="7">Reserved</td></tr>
<tr><td>R/W Attribute</td><td>R0, W</td><td colspan="7">R0, W0</td></tr>
<tr><td>Protection Attribute</td><td colspan="8">-</td></tr>
<tr><td>Initial Value</td><td>0</td><td colspan="7">0000000</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>SRST</td><td colspan="7">Reserved</td></tr>
<tr><td>R/W Attribute</td><td>R0,W</td><td colspan="7">R0,W0</td></tr>
<tr><td>Protection Attribute</td><td colspan="8">WP</td></tr>
<tr><td>Initial Value</td><td>0</td><td colspan="7">0000000</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>868</td><td>CHAPTER 12: Backup RAM Interface
5.15 BURAMIF TEST Software Reset Generation Control Register (BURIF_TSRCR)</td><td>Features of "For details" should be deleted as indicated by the shading below.

Delete the "Register configuration" and "Register functions".
Same 5.1 BURAMIF Unlock Register (BURIF_UNLOCK)

(Error)
Register configuration
Register functions".</td></tr>
<tr><td>869</td><td>CHAPTER 12: Backup RAM Interface
5.16 BURAMIF TEST Pseudo-error Generation Control Register (BURIF_TFECR</td><td>Features of "For details" should be deleted as indicated by the shading below.

Delete the "Register configuration" and "Register functions".
Same 5.1 BURAMIF Unlock Register (BURIF_UNLOCK)

(Error)
Register configuration
Register functions".</td></tr>
<tr><td>869</td><td>CHAPTER 12: Backup RAM Interface
5.16 BURAMIF TEST Pseudo-error Generation Control Register (BURIF_TFECR</td><td>"[bit2:0] ETYP" should be corrected as indicated by the shading below.

(Error)

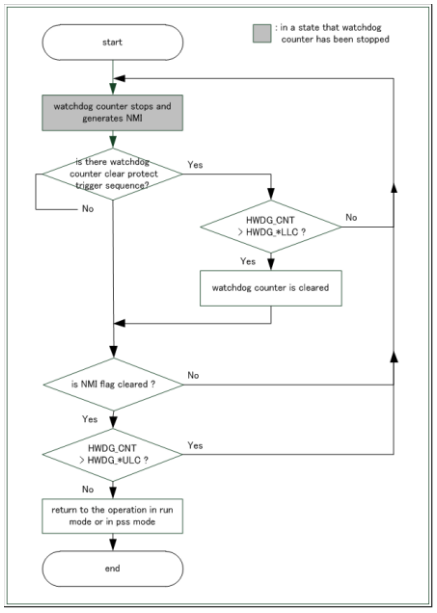
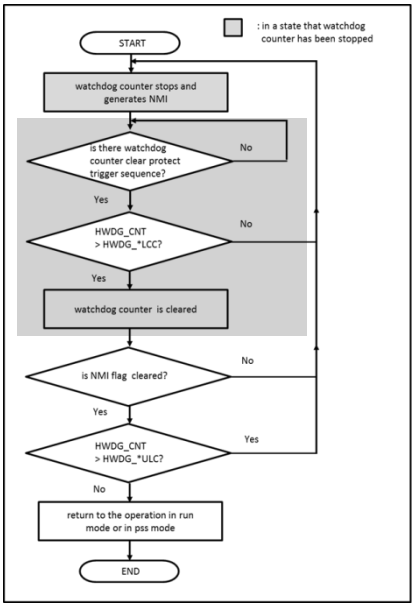
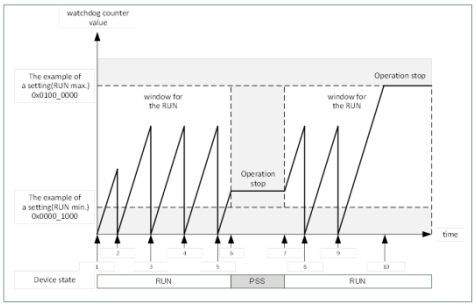
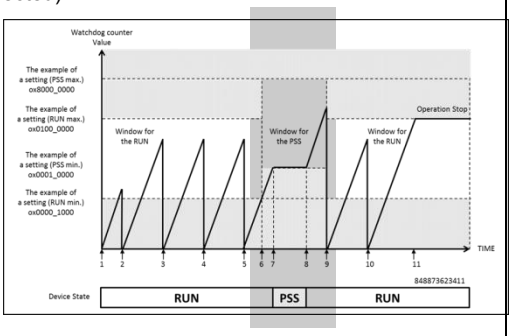
[bit2:0] ETYP: Pseudo-error generation processing specification
<table>
<tr><th>ETYP</th><th>Pseudo-error Generation Processing Specification Bit</th></tr>
<tr><td>During Write Operation</td><td>0bxx1: Pseudo-errors are generated during march diagnosis.
0bx1x: Pseudo-errors are generated during checker 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>
</table>

Page	Section	Change Results
870	CHAPTER 12: Backup RAM Interface 5.17 BURAMIF TEST Key Code Control Register (BURIF_TKCCR)	Features of "For details" should be deleted as indicated by the shading below. Delete the "Register configuration" and "Register functions". Same 5.1 BURAMIF Unlock Register (BURIF_UNLOCK) (Error) Register configuration Register functions".
871	CHAPTER 12: Backup RAM Interface 5.18 BURAMIF Event Clear Register (BURIF_EVENTCLR)	Features of "For details" should be deleted as indicated by the shading below. Delete the "Register configuration" and "Register functions". Same 5.1 BURAMIF Unlock Register (BURIF_UNLOCK) (Error) Register configuration Register functions".
872	CHAPTER 12: Backup RAM Interface 5.18 BURAMIF Event Clear Register (BURIF_EVENTCLR)	"[bit6] TEICLR" 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
872	CHAPTER 12: Backup RAM Interface 5.18 BURAMIF Event Clear Register (BURIF_EVENTCLR)	"[bit4] TCICLR" should be corrected as indicated by the shading below. (Error) [bit4] TCICLR: Diagnosis end (Correct) [bit4] TCICLR: Diagnosis End Clear Bit

Page	Section	Change Results
873	CHAPTER 12: Backup RAM Interface 6. Precautions for Using Device	Features of "For details" should be corrected as indicated by the shading below. (Error) Power domain control for BackupRAM depends on profile setting. To detail profile setting information, please refer to the Chapter "Lowpower". When BackupRAM power restores, the completion for the profile setting needs to be confirmed through profile update status register, or start to access BackupRAM after confirmed profile update completion interrupt register. BackupRAM can configure SLEEP setting during transfer to PSS. To detail SLEEP control information, please refer to the Chapter "Lowpower". When BackupRAM returns from SLEEP setting, start to access BackupRAM after confirmed that PSS had transferred to RUN status through the device status register of the system status register. (Correct) 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.
905	CHAPTER 14: Hardware Watchdog Timer 3. Explanation of Operation	"PSS operation setting" should be deleted as indicated by the shading below. (Error) - 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. For example, when the WDENPSS bit in the Hardware watchdog configuration register (HWDG_CFG) is set to "0" and the WDENRUN bit in the same register is set to "1", operation is as follows: Use of the window setting for PSS continues until the watchdog counter clear protection trigger sequence is executed. (Correct) - 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.

Page	Section	Change Results
905	CHAPTER 14: Hardware Watchdog Timer 3. Explanation of Operation	The Notes of "PSS operation setting" should be added as indicated by the shading below. 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.
906	CHAPTER 14: Hardware Watchdog Timer 3. Explanation of Operation	The Note of "Watchdog register write protection sequence" should be added as indicated by the shading below. (Error) Note: For details on the watchdog register write protection sequence, see (Correct) Note: For details on the watchdog register write protection sequence, see Figure 3-1 Setting Procedure for Watchdog Register Write Protection Sequence.
906	CHAPTER 14: Hardware Watchdog Timer 3. Explanation of Operation	The Note of “Figure 3-2 Setting Procedure for Watchdog Counter Clear Protection Trigger Sequence " should be corrected as indicated by the shading below. (Error) The notation * in is replaced with device state RUN or PSS. (Correct) The notation * in Figure 3-2 Setting Procedure for Watchdog Counter Clear Protection Trigger Sequence is replaced with device state RUN or PSS.

Page	Section	Change Results
917	CHAPTER 14: Hardware Watchdog Timer 3. Explanation of Operation	"Figure 3-6" should be modified as indicated by the shading below. (Error)  (Correct) 
922	CHAPTER 14: Hardware Watchdog Timer 5. Operation Examples	"Figure 5-2" should be modified as indicated by the shading below. (Error)  (Corrected) 

Page	Section	Change Results
922	CHAPTER 14: Hardware Watchdog Timer 5. Operation Examples	"Example of Operating Hardware Watchdog Timer (When Operation in PSS Is Disabled)" should be modified as indicated by the shading below. (Error) 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, the Hardware watchdog timer stops operation. 7. The device state returns from PSS to RUN. At this time, the Hardware watchdog timer immediately starts operation. 8. The watchdog counter clear protection trigger sequence is executed, and the watchdog counter is cleared. 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. (Correct) 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.

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>923</td><td>CHAPTER 14: Hardware Watchdog Timer
5. Operation Examples</td><td>The Notes of "Example of Operating Hardware Watchdog Timer (When Operation in PSS Is Disabled)" should be added as indicated by the shading below.

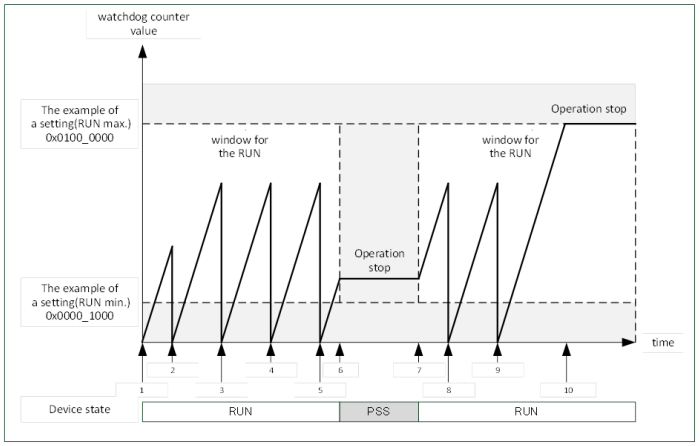
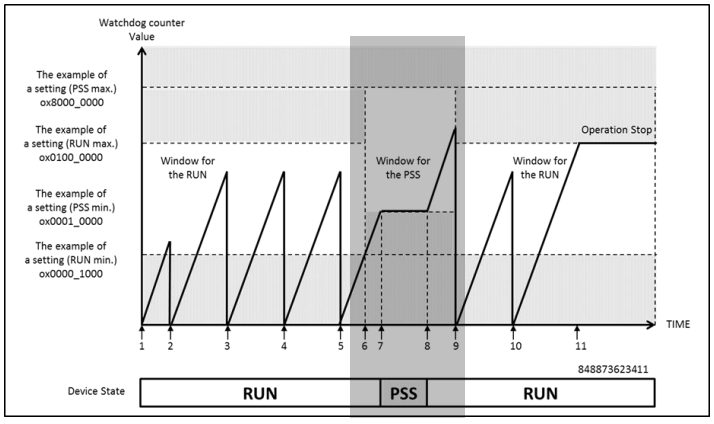
(Correct)
Figure 5 2 Example of Operating Hardware Watchdog Timer (When Operation in PSS Is Disabled)
...
Notes:
- The window setting is not immediately switched to the window setting for RUN when the device state is returned from PSS to RUN.
- After the device state returned from PSS to RUN, the watchdog 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 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)
- 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.</td></tr>
<tr><td>924</td><td>CHAPTER 14: Hardware Watchdog Timer
6. Register List</td><td>"Table 6-1" should be added as indicated by the shading below.

(Error)
<table><tr><th>Abbreviated Register Name</th><th>Register Name</th><th>Reference</th></tr><tr><td>HWDG_INTCLR</td><td>Hardware watchdog interrupt clear register</td><td>6.7</td></tr><tr><td>HWDG_TRG0CFG</td><td>Hardware watchdog trigger 0 configuration register</td><td></td></tr><tr><td>HWDG_TRG1CFG</td><td>Hardware watchdog trigger 1 configuration register</td><td>6.9</td></tr></table>
(Correct)
<table><tr><th>Abbreviated Register Name</th><th>Register Name</th><th>Reference</th></tr><tr><td>HWDG_INTCLR</td><td>Hardware watchdog interrupt clear register</td><td>6.7</td></tr><tr><td>HWDG_TRG0CFG</td><td>Hardware watchdog trigger 0 configuration register</td><td>6.8</td></tr><tr><td>HWDG_TRG1CFG</td><td>Hardware watchdog trigger 1 configuration register</td><td>6.9</td></tr></table></td></tr>
</table>

Page	Section	Change Results
929	CHAPTER 14: Hardware Watchdog Timer 6.4. Hardware Watchdog Trigger 0 Register (HWDG_TRG0)	The Note of "[bit7:0] WDGTRG0[7:0]" should be modified as indicated by the shading below. (Error) Note: エラー! 参照元が見つかりません。 (Correct) Note: Figure 3-2 shows the flow of the watchdog counter clear protection trigger sequence.
930	CHAPTER 14: Hardware Watchdog Timer 6.5. Hardware Watchdog Trigger 1 Register (HWDG_TRG1)	"[bit7:0] WDGTRG0[7:0]" should be modified as indicated by the shading below. (Error) [bit7:0] WDGTRG0[7:0]: Watchdog trigger 1 bits (Corrected) [bit7:0] WDGTRG1[7:0]: Watchdog Trigger 1 Bits
930	CHAPTER 14: Hardware Watchdog Timer 6.5. Hardware Watchdog Trigger 1 Register (HWDG_TRG1)	The Note of "[bit7:0] WDGTRG1[7:0]" should be modified as indicated by the shading below. (Error) Note: エラー! 参照元が見つかりません。 (Correct) Note: Figure 3-2 shows the flow of the watchdog counter clear protection trigger sequence.
931	CHAPTER 14: Hardware Watchdog Timer 6.6. Hardware Watchdog Interrupt Configuration Register (HWDG_INT)	"[bit17] RSTEN " should be added as indicated by the shading below. (Error) Prohibiting to set this bit to "0" except for the purpose of using it as a test function. (Correct) Prohibiting setting this bit to "0" except for the purpose of using it as a test function.
942	CHAPTER 14: Hardware Watchdog Timer 6.15. Hardware Watchdog Configuration Register (HWDG_CFG)	The Note of "[bit20:16] OBSSEL[4:0]" should be modified as indicated by the shading below. (Error) Note: Output of watchdog counter monitor bit for MCU output pin is unsupported. For the product, Any data written to these bits does not affect operation. (Correct) 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.

Page	Section	Change Results
942	CHAPTER 14: Hardware Watchdog Timer 6.15. Hardware Watchdog Configuration Register (HWDG_CFG)	" [bit9:8] CLKSEL[1:0]: Clock selection bits" should be modified as indicated by the shading below. (Error) For details on clock switching, see "7 Precautions for Using This Device エラー! 参照元が見つかりません。." (Correct) For details on clock switching, see "7 Precautions for Using This Device Switching of watchdog counter source clock.
943	CHAPTER 14: Hardware Watchdog Timer 6.15. Hardware Watchdog Configuration Register (HWDG_CFG)	"[bit2] ALLOWSTOPCLK" should be added as indicated by the shading below. (Error) [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, "1" and are always read.) (Correct) [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, "1" is always read.)
948	CHAPTER 14: Hardware Watchdog Timer 7. Precautions for Using This Device	The Notes of "Watchdog window setting" should be added as indicated by the shading below. (Correct) 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)

Page	Section	Change Results
957	CHAPTER 15:Software Watchdog Timer 3 Explanation of Operation	“PSS operation setting” should be deleted as indicated by the shading below. 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. For example, when the WDENPSS bit in the Software watchdog configuration register (SWDG_CFG) is set to "0" and the WDENRUN bit in the same register is set to "1", operation is as follows: Use of the window setting for PSS continues until the watchdog counter clear protection trigger sequence is executed.
957	CHAPTER 15:Software Watchdog Timer 3 Explanation of Operation	The Notes of “PSS operation setting” should be added as indicated by the shading below. 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.
957	CHAPTER 15:Software Watchdog Timer 3 Explanation of Operation	The description for "Figure 3-4" should be modified as indicated by the shading below. (Error) Figure 3-4 is shown about the hardware watchdog operation in RUN. Figure 3-5 is shown about the hardware watchdog operation in PSS. (Correct) Figure 3-4 is shown about the software watchdog operation in RUN. Figure 3-5 is shown about the software watchdog operation in PSS.
958	CHAPTER 15:Software Watchdog Timer 3 Explanation of Operation	"Watchdog register write protection sequence" should be modified as indicated by the shading below. (Error) Note: For details on the watchdog register write protection sequence, see (Correct) Note: For details on the watchdog register write protection sequence, see .Figure 3-1 Setting Procedure for Watchdog Register Write Protection Sequence.

Page	Section	Change Results
964	CHAPTER 15:Software Watchdog Timer 3 Explanation of Operation	The Note of " Figure 3-2 Setting Procedure for Watchdog Counter Clear Protection Trigger Sequence " should be corrected as indicated by the shading below. (Error) The notation * in エラー! 参照元が見つかりません。 is replaced with device state RUN or PSS. (Correct) The notation * in Figure 3-2 Setting Procedure for Watchdog Counter Clear Protection Trigger Sequence is replaced with device state RUN or PSS.
973	CHAPTER 15:Software Watchdog Timer 5 Operation Examples	"Figure 5-2" should be modified as indicated by the shading below. (Error)  (Correct) 

Page	Section	Change Results
973	CHAPTER 15:Software Watchdog Timer 5 Operation Examples	“Figure 5-2 Example of Operating Software Watchdog Timer (When Operation in PSS Is Disabled)” should be modified as indicated by the shading below. (Error) 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, the Software watchdog timer stops operation. 7.The device state returns from PSS to RUN. At this time, the Software watchdog timer immediately starts operation. 8.The watchdog counter clear protection trigger sequence is executed, and the watchdog counter is cleared. 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. (Correct) 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.

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>974</td><td>CHAPTER 15:Software Watchdog Timer
5 Operation Examples</td><td>The Note of “Figure 5-2 Example of Operating Software Watchdog Timer (When Operation in PSS Is Disabled)” should be added as indicated by the shading below.

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 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 PSS to RUN. Because the window setting don’t change immediately. When device state change from RUN to PSS, clear the watchdog counter.</td></tr>
<tr><td>975</td><td>CHAPTER 15:Software Watchdog Timer
6. Register List</td><td>"Table 6-1" should be added as indicated by the shading below.

(Error)
<table>
<tr><th>Abbreviated Register Name</th><th>Register Name</th><th>Reference</th></tr>
<tr><td>SWDG_INTCLR</td><td>Software watchdog interrupt clear register</td><td>6.7</td></tr>
<tr><td>SWDG_TRG0CFG</td><td>Software watchdog trigger 0 configuration register</td><td></td></tr>
<tr><td>SWDG_TRG1CFG</td><td>Software watchdog trigger 1 configuration register</td><td>6.9</td></tr>
</table>
(Correct)
<table>
<tr><th>Abbreviated Register Name</th><th>Register Name</th><th>Reference</th></tr>
<tr><td>SWDG_INTCLR</td><td>Software watchdog interrupt clear register</td><td>6.7</td></tr>
<tr><td>SWDG_TRG0CFG</td><td>Software watchdog trigger 0 configuration register</td><td>6.8</td></tr>
<tr><td>SWDG_TRG1CFG</td><td>Software watchdog trigger 1 configuration register</td><td>6.9</td></tr>
</table></td></tr>
<tr><td>976</td><td>CHAPTER 15:Software Watchdog Timer
6.1. Software Watchdog Protection Register (SWDG_PROT)</td><td>The title should be modified as indicated by the shading below.

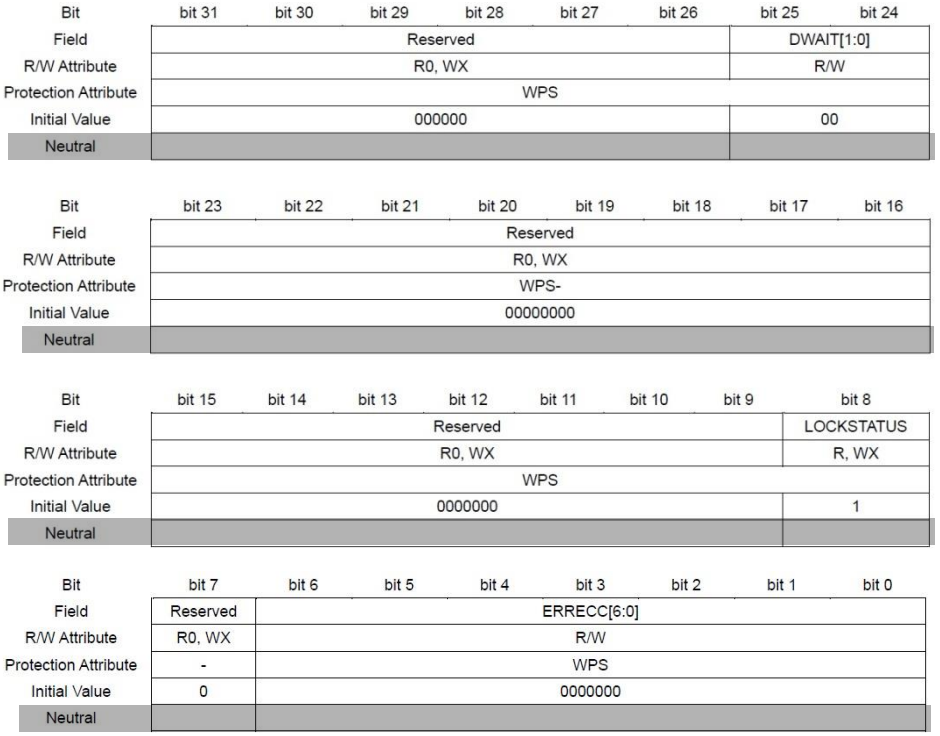
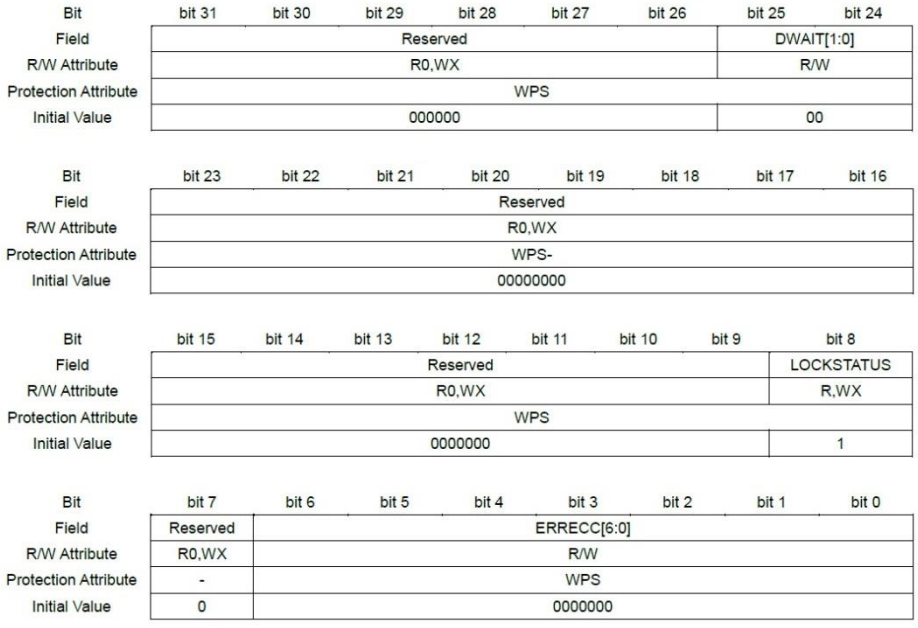
(Error)
Software Watchdog Protection Register (SWDG_PROT)

(Correct)
Software Watchdog Protection Register (SWDT_PROT)</td></tr>
</table>

Page	Section	Change Results
976	CHAPTER 15:Software Watchdog Timer 6.1. Software Watchdog Protection Register (SWDG_PROT)	Features of "For details" should be deleted as indicated by the shading below. (Error) **[bit31:0] KEY[31:0]: Protection bits** ***Notes:*** – *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.*
980	CHAPTER 15:Software Watchdog Timer 6.4 Software Watchdog Trigger 0 Register (SWDG_TRG0)	The Note of "[bit7:0] WDGTRG0[7:0]" should be added as indicated by the shading below. (Error) Note: エラー! 参照元が見つかりません。 (Correct) Note: Figure 3-2 shows the flow of the watchdog counter clear protection trigger sequence.
981	CHAPTER 15:Software Watchdog Timer 6.5 Software Watchdog Trigger 1 Register (SWDG_TRG1)	"[bit7:0] WDGTRG0[7:0]" should be modified as indicated by the shading below. (Error) [bit7:0] WDGTRG0[7:0]: Watchdog trigger 1 bits (Corrected) [bit7:0] WDGTRG1[7:0]: Watchdog trigger 1 bits
981	CHAPTER 15:Software Watchdog Timer 6.5 Software Watchdog Trigger 1 Register (SWDG_TRG1)	The Note of "[bit7:0] WDGTRG1[7:0]" should be added as indicated by the shading below. (Error) Note: エラー! 参照元が見つかりません。 (Correct) Note: Figure 3-2 shows the flow of the watchdog counter clear protection trigger sequence.
993	CHAPTER 15:Software Watchdog Timer 6.15 Software Watchdog Configuration Register (SWDG_CFG)	"Reserved" should be modified as indicated by the shading below. Register configurations (Error) [bit23:16] Reserved: Reserved bits (Correct) [bit23:10] Reserved: Reserved bits

Page	Section	Change Results
993	CHAPTER 15:Software Watchdog Timer 6.15 Software Watchdog Configuration Register (SWDG_CFG)	"[bit15:10] Reserved " should be deleted as indicated by the shading below. (Error) [bit15:10] Reserved: Reserved bits
993	CHAPTER 15:Software Watchdog Timer 6.15 Software Watchdog Configuration Register (SWDG_CFG)	" [bit9:8] CLKSEL[1:0]" should be modified as indicated by the shading below. (Error) 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 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. Switching of watchdog counter source clock." (Correct) 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."
999	CHAPTER 15:Software Watchdog Timer 7. Precautions for Using This Device	The Notes of “Watchdog window setting” should be added as indicated by the shading below. 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 (HWDG_PSSULS) - Software watchdog upper limit RUN setting register (HWDG_RUNULS)
1001	CHAPTER 16:TCRAM Interface	Revised the word “Cortex-R5F” to “Cortex-R5”
1002	CHAPTER 16:TCRAM Interface 1. Overview	Revised the word “Cortex-R5F” to “Cortex-R5”

Page	Section	Change Results
1003 to 1005	CHAPTER 16:TCRAM Interface 2. Configuration 3. Explanation of Operation	Revised the word “Cortex-R5F” to “Cortex-R5”

Page	Section	Change Results
1012	CHAPTER 16:TCRAM Interface 5.1 TCRAM IF Configuration Register 0 (TRCFGn_TCMCFG0)	Features of "For details" should be corrected as indicated by the shading below. (Error)  (Correct) 

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	
Neutral								

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							
Neutral								

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
Neutral								

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						
Neutral								

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						

Page	Section	Change Results
1013	CHAPTER 16:TCRAM Interface 5.1. TCRAM IF Configuration Register 0 (TRCFGn_TCMCFG0)	Revised the word “Cortex-R5F” to “Cortex-R5”
1014	CHAPTER 16:TCRAM Interface 5.2. TCRAM IF Configuration Register 1 (TRCFGn_TCMCFG1)	Revised the word “Cortex-R5F” to “Cortex-R5”
1014	CHAPTER 16:TCRAM Interface 5.2 TCRAM IF Configuration Register 1 (TRCFGn_TCMCFG1)	Features of "For details" should be corrected as indicated by the shading below. (Error) (Correct)

(Error)

Bit	bit 31	bit 0
Field	ERRBIT[31:0]	
R/W Attribute	R/W	
Protection Attribute	WPS	
Initial Value	00000000_00000000_00000000_00000000	
Neutral		

(Correct)

Bit	bit 31	bit 0
Field	ERRBIT[31:0]	
R/W Attribute	R/W	
Protection Attribute	WPS	
Initial Value	00000000_00000000_00000000_00000000	

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1015</td><td>CHAPTER 16:TCRAM Interface
5.3 TCRAM IF Unlock Register (TRCFGn_TCMUNLOCK)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)

<table>
<tr><td>Bit</td><td>bit 31</td><td>bit 0</td></tr>
<tr><td>Field</td><td colspan="2">UNLOCK[31:0]</td></tr>
<tr><td>R/W Attribute</td><td colspan="2">R0, W</td></tr>
<tr><td>Protection Attribute</td><td colspan="2">WPS</td></tr>
<tr><td>Initial Value</td><td colspan="2">00000000_00000000_00000000_00000000</td></tr>
<tr><td>Neutral</td><td colspan="2"></td></tr>
</table>

(Correct)

<table>
<tr><td>Bit</td><td>bit 31</td><td>bit 0</td></tr>
<tr><td>Field</td><td colspan="2">UNLOCK[31:0]</td></tr>
<tr><td>R/W Attribute</td><td colspan="2">R0,W</td></tr>
<tr><td>Protection Attribute</td><td colspan="2">WPS</td></tr>
<tr><td>Initial Value</td><td colspan="2">00000000_00000000_00000000_00000000</td></tr>
</table>
</td></tr>
</table>

Page	Section	Change Results
1016	CHAPTER 16:TCRAM Interface 5.4 TCRAM IF TEST Error Address Register 0 (TRCFGn_TEAR0)	Features of "For details" should be corrected as indicated by the shading below. (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, W / R0, W Protection Attribute: - Initial Value: 000 / 00000 Neutral Bit / bit 23 / bit 22 / bit 21 / bit 20 / bit 19 / bit 18 / bit 17 / bit 16 Field: Reserved R/W Attribute: R0, W Protection Attribute: - Initial Value: 00000000 Neutral Bit / bit 15 / bit 14 / bit 13 / bit 12 / bit 11 / bit 10 / bit 9 / bit 8 Field: Reserved / ERR_ADDR[13:8] R/W Attribute: R0, W / R, W Protection Attribute: - Initial Value: 00 / 000000 Neutral 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, W Protection Attribute: - Initial Value: 00000000 Neutral (Correct) 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[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: ERR_ADDR[7:0] R/W Attribute: R,WX Protection Attribute: - Initial Value: 00000000

Page	Section	Change Results
1018	CHAPTER 16:TCRAM Interface 5.5 TCRAM IF TEST Error Address Register 1 (TRCFGn_TEAR1)	Features of "For details" should be corrected as indicated by the shading below. (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, W			R0, W				
Protection Attribute	-							
Initial Value	000			00000				
Neutral								

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

Bit	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
Field	Reserved		ERR_ADDR[13:8]					
R/W Attribute	R0, W		R, W					
Protection Attribute	-							
Initial Value	00		000000					
Neutral								

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, W							
Protection Attribute	-							
Initial Value	00000000							
Neutral								

(Correct)

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[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	ERR_ADDR[7:0]							
R/W Attribute	R,WX							
Protection Attribute	-							
Initial Value	00000000							

Page	Section	Change Results
1020	CHAPTER 16:TCRAM Interface 5.6 TCRAM IF TEST Error Address Register 2 (TRCFGn_TEAR2)	Features of "For details" should be corrected as indicated by the shading below. (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, W | R0, W Protection Attribute: - Initial Value: 000 | 00000 Neutral Bit / bit 23 / bit 22 / bit 21 / bit 20 / bit 19 / bit 18 / bit 17 / bit 16 Field: Reserved R/W Attribute: R0, W Protection Attribute: - Initial Value: 00000000 Neutral Bit / bit 15 / bit 14 / bit 13 / bit 12 / bit 11 / bit 10 / bit 9 / bit 8 Field: Reserved | ERR_ADDR[13:8] R/W Attribute: R0, W | R, W Protection Attribute: - Initial Value: 00 | 000000 Neutral 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, W Protection Attribute: - Initial Value: 00000000 Neutral (Correct) 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[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: ERR_ADDR[7:0] R/W Attribute: R,WX Protection Attribute: - Initial Value: 00000000

Page	Section	Change Results
1022	CHAPTER 16:TCRAM Interface 5.7 TCRAM IF TEST End Address Register (TRCFGn_TAEAR)	Features of "For details" should be corrected as indicated by the shading below. (Error) Bit: bit 15 | bit 14 | bit 13 | bit 12 | bit 11 | bit 10 | bit 9 | bit 8 Field: Reserved | EADDR[13:8] R/W Attribute: R0, WX | R/W Protection Attribute: WPS Initial Value: 00 | 111111 Neutral 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 Neutral (Correct) Bit: bit 15 | bit 14 | bit 13 | bit 12 | bit 11 | bit 10 | bit 9 | bit 8 Field: Reserved | EADDR[13:8] R/W Attribute: R0,WX | R/W Protection Attribute: WPS Initial Value: 00 | 111111 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

Page	Section	Change Results
1023	CHAPTER 16:TCRAM Interface 5.8 TCRAM IF TEST Start Address Register (TRCFGn_TASAR)	Features of "For details" should be corrected as indicated by the shading below. (Error) Bit / bit 15 / bit 14 / bit 13 / bit 12 / bit 11 / bit 10 / bit 9 / bit 8 Field: Reserved / SADDR[13:8] R/W Attribute: R0, WX / R/W Protection Attribute: WPS Initial Value: 00 / 000000 Neutral 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 Neutral (Correct) Bit / bit 15 / bit 14 / bit 13 / bit 12 / bit 11 / bit 10 / bit 9 / bit 8 Field: Reserved / SADDR[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: SADDR[7:0] R/W Attribute: R/W Protection Attribute: WPS Initial Value: 00000000

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1024</td><td>CHAPTER 16:TCRAM Interface
5.9 TCRAM IF TEST Pseudo Error Generation Control Register (TRCFGn_TFECR)</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="4">Reserved</td><td>FERR</td><td colspan="3">ETYP[2:0]</td></tr>
<tr><td>R/W Attribute</td><td colspan="4">R0, WX</td><td>R/W</td><td colspan="3">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 colspan="3">000</td></tr>
<tr><td>Neutral</td><td colspan="4"></td><td></td><td colspan="3"></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="4">Reserved</td><td>FERR</td><td colspan="3">ETYP[2:0]</td></tr>
<tr><td>R/W Attribute</td><td colspan="4">R0,WX</td><td>R/W</td><td colspan="3">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 colspan="3">000</td></tr>
</table>
</td></tr>
<tr><td>1025</td><td>CHAPTER 16:TCRAM Interface
5.10 TCRAM IF TEST Initialization Function Register (TRCFGn_TICR)</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="4">Reserved</td><td>ICIE</td><td>ICI</td><td>ITYP</td><td>IRUN</td></tr>
<tr><td>R/W Attribute</td><td colspan="4">R0, WX</td><td>R/W</td><td>R, WX</td><td>R/W</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>0</td><td>0</td><td>0</td></tr>
<tr><td>Neutral</td><td colspan="4"></td><td></td><td></td><td></td><td></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="4">Reserved</td><td>ICIE</td><td>ICI</td><td>ITYP</td><td>IRUN</td></tr>
<tr><td>R/W Attribute</td><td colspan="4">R0,WX</td><td>R/W</td><td>R,WX</td><td>R/W</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>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>1026</td><td>CHAPTER 16:TCRAM Interface
5.11 TCRAM IF TEST Diagnosis Function Register (TRCFGn_TTCR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)

<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="6">Reserved</td><td>TSTAT</td><td>OVFLW</td></tr>
<tr><td>R/W Attribute</td><td colspan="6">R0, 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="6">000000</td><td>0</td><td>0</td></tr>
<tr><td>Neutral</td><td colspan="6"></td><td></td><td></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>TEIE</td><td>TEI</td><td>TCIE</td><td>TCI</td><td colspan="3">TTYP</td><td>TRUN</td></tr>
<tr><td>R/W Attribute</td><td>R/W</td><td>R, WX</td><td>R/W</td><td>R, WX</td><td colspan="3">R/W</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 colspan="3">110</td><td>0</td></tr>
<tr><td>Neutral</td><td></td><td></td><td></td><td></td><td colspan="3"></td><td></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="6">Reserved</td><td>TSTAT</td><td>OVFLW</td></tr>
<tr><td>R/W Attribute</td><td colspan="6">R0,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="6">000000</td><td>0</td><td>0</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>TEIE</td><td>TEI</td><td>TCIE</td><td>TCI</td><td colspan="3">TTYP</td><td>TRUN</td></tr>
<tr><td>R/W Attribute</td><td>R/W</td><td>R,WX</td><td>R/W</td><td>R,WX</td><td colspan="3">R/W</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 colspan="3">110</td><td>0</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1029</td><td>CHAPTER 16:TCRAM Interface
5.12 TCRAM IF TEST Soft Reset Generation Control Register (TRCFGn_TSRCR)</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>SRST</td><td colspan="7">Reserved</td></tr>
<tr><td>R/W Attribute</td><td>R0, W</td><td colspan="7">R0, WX</td></tr>
<tr><td>Protection Attribute</td><td colspan="8">WPS</td></tr>
<tr><td>Initial Value</td><td>0</td><td colspan="7">0000000</td></tr>
<tr><td>Neutral</td><td></td><td colspan="7"></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>SRST</td><td colspan="7">Reserved</td></tr>
<tr><td>R/W Attribute</td><td>R0,W</td><td colspan="7">R0,WX</td></tr>
<tr><td>Protection Attribute</td><td colspan="8">WPS</td></tr>
<tr><td>Initial Value</td><td>0</td><td colspan="7">0000000</td></tr>
</table>
</td></tr>
<tr><td>1030</td><td>CHAPTER 16:TCRAM Interface
5.13 TCRAM IF TEST Key Code Control Register (TRCFGn_TKCCR)</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="2">KEY[1:0]</td><td colspan="4">Reserved</td><td colspan="2">CODE[1:0]</td></tr>
<tr><td>R/W Attribute</td><td colspan="2">R0, W</td><td colspan="4">R0, WX</td><td colspan="2">R/W</td></tr>
<tr><td>Protection Attribute</td><td colspan="8">WPS</td></tr>
<tr><td>Initial Value</td><td colspan="2">00</td><td colspan="4">0000</td><td colspan="2">00</td></tr>
<tr><td>Neutral</td><td colspan="2"></td><td colspan="4"></td><td colspan="2"></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="2">KEY[1:0]</td><td colspan="4">Reserved</td><td colspan="2">CODE[1:0]</td></tr>
<tr><td>R/W Attribute</td><td colspan="2">R0,W</td><td colspan="4">R0,WX</td><td colspan="2">R/W</td></tr>
<tr><td>Protection Attribute</td><td colspan="8">WPS</td></tr>
<tr><td>Initial Value</td><td colspan="2">00</td><td colspan="4">0000</td><td colspan="2">00</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1031</td><td>CHAPTER 16:TCRAM Interface
5.14 TCRAM IF TEST Status Clear Register (TRCFGn_TSCR)</td><td>Features of "For details" should be corrected as indicated by the shading below.

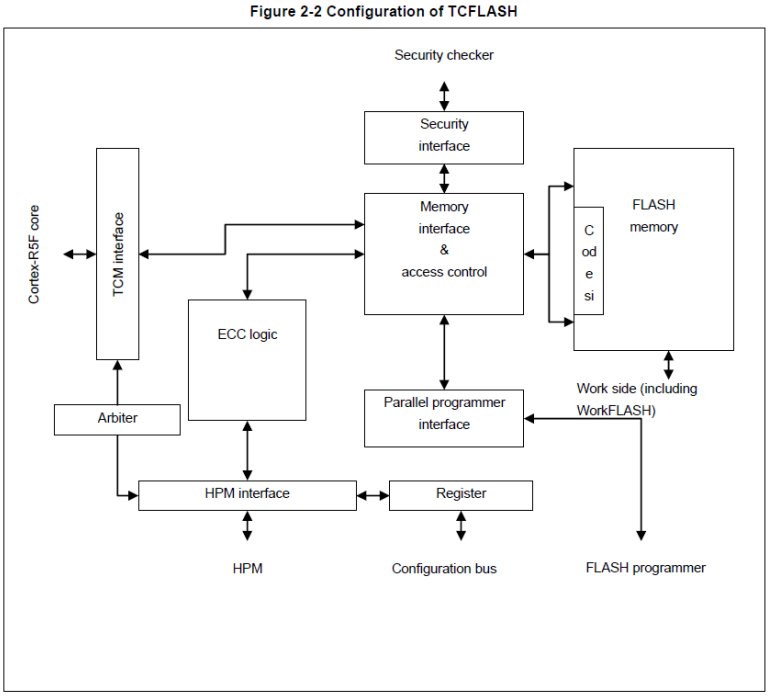
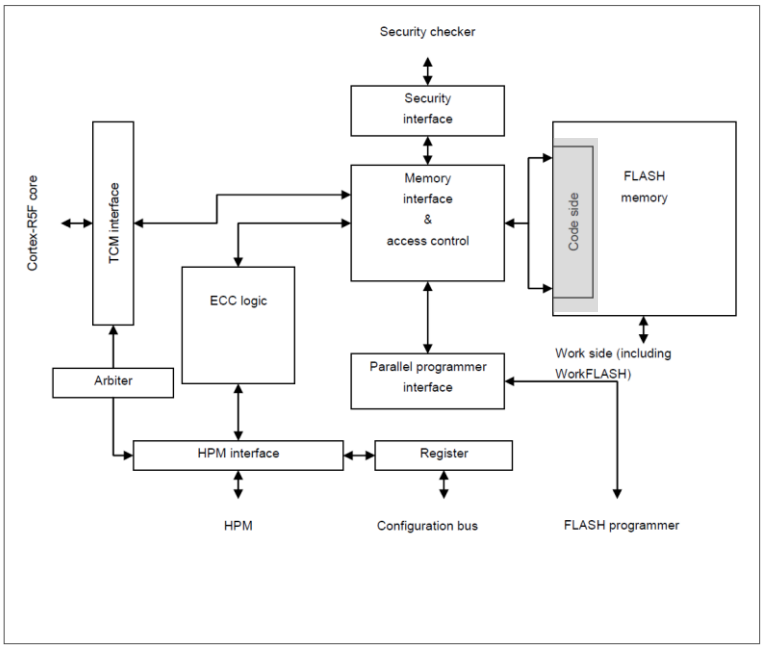
(Error)

<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 colspan="2">bit 1 bit 0</td></tr>
<tr><td>Field</td><td>Reserved</td><td>TEIC</td><td>Reserved</td><td>TCIC</td><td>Reserved</td><td>ICIC</td><td colspan="2">Reserved</td></tr>
<tr><td>R/W Attribute</td><td>R0, WX</td><td>R0, W</td><td>R0, WX</td><td>R0, W</td><td>R0, WX</td><td>R0, W</td><td colspan="2">R0, 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>0</td><td colspan="2">00</td></tr>
<tr><td>Neutral</td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="2"></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 colspan="2">bit 1 bit 0</td></tr>
<tr><td>Field</td><td>Reserved</td><td>TEIC</td><td>Reserved</td><td>TCIC</td><td>Reserved</td><td>ICIC</td><td colspan="2">Reserved</td></tr>
<tr><td>R/W Attribute</td><td>R0,WX</td><td>R0,W</td><td>R0,WX</td><td>R0,W</td><td>R0,WX</td><td>R0,W</td><td colspan="2">R0,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>0</td><td colspan="2">00</td></tr>
</table>
</td></tr>
</table>

Page	Section	Change Results
1034	CHAPTER 17:TCFLASH 1. Overview	Revised the word “Cortex-R5F” to “Cortex-R5”
1035	CHAPTER 17:TCFLASH 2. Configuration	Revised the word “Cortex-R5F” to “Cortex-R5”

Page	Section	Change Results
1036	CHAPTER 17:TCFLASH 2.Configuration 2.1.Block Diagrams	Features of "For details" should be corrected as indicated by the shading below (Error)  Figure 2-2 Configuration of TCFLASH (Correct)  Figure 2-2 Configuration of TCFLASH

Page	Section	Change Results
1038	CHAPTER 17:TCFLASH 2. Configuration 2.2. Address/Sector Map of TCFLASH	"Figure 2-4 " should be corrected as indicated by the shading below (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.
1041	CHAPTER 17:TCFLASH 3. Explanation of Operation	Revised the word “Cortex-R5F” to “Cortex-R5”
1047	CHAPTER 17:TCFLASH 3.Explanation of Operation 3.5.1.Command Sequence	Features of "For details" should be corrected as indicated by the shading below (Error) ■The 4 MB model has 2 FLASH macros (SA0 to 39 and SA40 to 71), and care should be taken regarding the following. (Correct) ■The 4 MB model has 2 FLASHmacros (SA0 to 39 and SA40 to 71), and care should be taken regarding the following.
1047	CHAPTER 17:TCFLASH 3.Explanation of Operation 3.5.1.Command Sequence	Features of "For details" should be corrected as indicated by the shading below (Error) Multiple sector erasure cannot be executed over 2 FLASH macros (SA39 and SA40). (Correct) Multiple sector erasure cannot be executed over 2 FLASH macro (SA39 and SA40).

Page	Section	Change Results
1051	CHAPTER 17:TCFLASH 3.8. ECC Generation and Check	Features of "For details" should be added as indicated by the shading below. (Correct) 3.8.ECC Generation and Check The ECC movement in TCM port is based on ECC setting inside the CPU.
1051	CHAPTER 17:TCFLASH 3.8. ECC Generation and Check	Revised the word “Cortex-R5F” to “Cortex-R5”
1055	CHAPTER 17:TCFLASH 3.Explanation of Operation 3.9.Interrupt	Features of "For details" should be corrected as indicated by the shading below (Error) When a timeout is detected during the execution of an automatic algorithm, etc., the hangup state is detected, causing the HANGINT bit in the TCFCFG_FSTAT register to show "1". (Correct) 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".
1057	CHAPTER 17:TCFLASH 4. Setting Procedure Examples	Revised product names in the explanation as follows In the case of S6J312xHzB*, the maximum operating frequency of flash memory built in this LSI is 64 MHz. Therefore, if the system operating frequency is 128 MHz, for example, the setting value for FAWC[1:0] should be ""01"" as calculated using the above formula." *: x: A/9/8, z: A/B
1067	CHAPTER 17:TCFLASH 4.Setting Procedure Examples 4.6.1.Sector Erase Suspend	Features of "For details" should be corrected as indicated by the shading below (Error) Do not write a command sequence other than the read/reset or sector erase resume command in the flash memory in the sector erase suspends state. Figure 4-5 shows a flow example of sector erases suspend. (Correct) 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 flow example of sector erase suspend.

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1073</td><td>CHAPTER 17:TCFLASH
5.3. TCFLASH ECC Control Register: TCFCFG_FECCCTRL (TCFLASH ECC Control Register)</td><td>Revised the word “Cortex-R5F” to “Cortex-R5”</td></tr>
<tr><td>1079</td><td>CHAPTER 17:TCFLASH
5.Registers
5.7.TCFLASH Status Register: TCFCFG_FSTAT (TCFLASH Status Register)</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>16</td></tr>
<tr><td>Field</td><td colspan="2">Reserved</td></tr>
<tr><td>R/W Attribute</td><td colspan="2">R0, WX</td></tr>
<tr><td>Protection Attribute</td><td colspan="2">-</td></tr>
<tr><td>Initial Value</td><td colspan="2">00000000_00000000</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="6">Reserved</td><td>HANGINT</td><td>RDYINT</td></tr>
<tr><td>R/W Attribute</td><td colspan="6">R0, WX</td><td>R, WX</td><td>R, WX</td></tr>
<tr><td>Protection Attribute</td><td colspan="8">-</td></tr>
<tr><td>Initial Value</td><td colspan="6">000000</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>CERS</td><td>PGMS</td><td>ESPS</td><td>ERSEC</td><td>SERS</td><td>READ</td><td>HANG</td><td>RDY</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">-</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>0</td><td>1</td></tr>
</table>

(Correct)
<table>
<tr><td>bit</td><td>31</td><td>16</td></tr>
<tr><td>Field</td><td colspan="2">Reserved</td></tr>
<tr><td>R/W attribute</td><td colspan="2">R0, WX</td></tr>
<tr><td>Protection Attribute</td><td colspan="2">-</td></tr>
<tr><td>Initial Value</td><td colspan="2">00000000_00000000</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="6">Reserved</td><td>HANGINT</td><td>RDYINT</td></tr>
<tr><td>R/W Attribute</td><td colspan="6">R0, WX</td><td>R, WX</td><td>R, WX</td></tr>
<tr><td>Protection Attribute</td><td colspan="8">-</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>CERS</td><td>PGMS</td><td>ESPS</td><td>ERSEC</td><td>SERS</td><td>READ</td><td>HANG</td><td>RDY</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">-</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
1079	CHAPTER 17:TCFLASH 5.Registers 5.7.TCFLASH Status Register: TCFCFG_FSTAT (TCFLASH Status Register)	Features of "For details" should be corrected as indicated by the shading below The sentence below is added to "[bit 8] RDYINT: Ready interrupt". Note: − Initial value depends on the boot configuration.
1081	CHAPTER 17:TCFLASH 5.Registers 5.7.TCFLASH Status Register: TCFCFG_FSTAT (TCFLASH Status Register)	Features of "For details" should be corrected as indicated by the shading below The sentence below is added to "[bit 2] READ: Reading ready". Note: − Initial value depends on the boot configuration.
1081	CHAPTER 17:TCFLASH 5.Registers 5.7.TCFLASH Status Register: TCFCFG_FSTAT (TCFLASH Status Register)	Features of "For details" should be corrected as indicated by the shading below The sentence below is added at the end of this section. Note: − Initial value depends on the boot configuration.

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1091</td><td>CHAPTER 17:TCFLASH
5.Registers
5.14.TCFLASH TCM Buffer Register: TCFCFG_TCMBUF0 to 15 (TCFLASH Buffer0 to 15)</td><td>Features of "For details" should be corrected as indicated by the shading below

(Error)
<table>
<tr><td>bit</td><td colspan="2">239</td><td>160</td><td>159</td><td>152</td><td>151</td><td>132</td><td>131</td><td>128</td></tr>
<tr><td>Field</td><td colspan="2">Reserved</td><td>BVALID</td><td colspan="2">Reserved</td><td colspan="2">BUFADDH[23:4]</td><td colspan="2">BUFADDL[3:0]</td></tr>
<tr><td>R/W Attribute</td><td colspan="2">R0, WX</td><td>R,W</td><td colspan="2">R0, WX</td><td colspan="2">R,W</td><td colspan="2">R0, WX</td></tr>
<tr><td>Protection Attribute</td><td colspan="5">WPS</td><td colspan="4">WPS</td></tr>
<tr><td>Initial Value</td><td colspan="5">ALL0</td><td colspan="4">800000</td></tr>
</table>
<table>
<tr><td>bit</td><td>255</td><td>240</td></tr>
<tr><td>Field</td><td colspan="2">ECC_PARITY[15:0]</td></tr>
<tr><td>R/W Attribute</td><td colspan="2">R,W</td></tr>
<tr><td>Protection Attribute</td><td colspan="2">WP</td></tr>
<tr><td>Initial Value</td><td colspan="2">ALL0</td></tr>
</table>
(Correct)
<table>
<tr><td>bit</td><td colspan="2">239</td><td>161</td><td>160</td><td>159</td><td>152</td><td>151</td><td>132</td><td>131</td><td>128</td></tr>
<tr><td>Field</td><td colspan="3">Reserved</td><td>BVALID</td><td colspan="2">Reserved</td><td colspan="2">BUFADDH[23:4]</td><td colspan="2">BUFADDL[3:0]</td></tr>
<tr><td>R/W Attribute</td><td colspan="3">R0, WX</td><td>R,W</td><td colspan="2">R0, WX</td><td colspan="2">R,W</td><td colspan="2">R0, WX</td></tr>
<tr><td>Protection Attribute</td><td colspan="6">WPS</td><td colspan="4">WPS</td></tr>
<tr><td>Initial Value</td><td colspan="6">ALL0</td><td colspan="4">0x800000</td></tr>
</table>
<table>
<tr><td>bit</td><td>255</td><td>240</td></tr>
<tr><td>Field</td><td colspan="2">ECC_PARITY[15:0]</td></tr>
<tr><td>R/W Attribute</td><td colspan="2">R,W</td></tr>
<tr><td>Protection Attribute</td><td colspan="2">WPS</td></tr>
<tr><td>Initial Value</td><td colspan="2">ALL0</td></tr>
</table></td></tr>
<tr><td>1091</td><td>CHAPTER 17:TCFLASH
5.Registers
5.14.TCFLASH TCM Buffer Register: TCFCFG_TCMBUF0 to 15 (TCFLASH Buffer0 to 15)</td><td>Features of "For details" should be added as indicated by the shading below:

Note:
See the following table for the initial value of BUFFADDH[23:4] and BUFFADDL[3:0].
<table>
<tr><th>TCFLASH memory size</th><th>Initial value of BUFFADDH[23:4] and BUFFADDL[3:0]</th></tr>
<tr><td>1MB</td><td>0x900000</td></tr>
<tr><td>2MB</td><td>0x800000</td></tr>
<tr><td>4MB</td><td>0x800000</td></tr>
</table></td></tr>
<tr><td>1098</td><td>CHAPTER 18:WorkFLASH
2.Configuration
2.2.Address/Sector Maps in User Mode</td><td>Features of "For details" should be corrected as indicated by the shading below

(Error)
Mirror area 4 is a read-only area that is set up for reading with memory attribute access.

(Correct)
Mirror area 4 is a read-only area that is set up for reading with Normal memory attribute access.</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1098</td><td>CHAPTER 18:WorkFLASH
2.Configuration
2.2.Address/Sector Maps in User Mode</td><td>Features of "For details" should be corrected as indicated by the shading below

(Error)
The following table shows the purpose for each area.
<table>
<tr><td>Area number</td><td>Read operation</td><td>Write operation</td></tr>
<tr><td>Mirror area 1</td><td>Normal read</td><td>Write the value with ECC calculated</td></tr>
<tr><td>Mirror area 2</td><td>Reserved area (access prohibited)</td><td>Reserved area (access prohibited)</td></tr>
<tr><td>Mirror area 3</td><td>Normal read</td><td>Write the value without ECC</td></tr>
<tr><td>Mirror area 4</td><td>Memory attribute read</td><td>Access prohibited</td></tr>
</table>
(Correct)
The following table shows the purpose for each area.
<table>
<tr><th>Area number</th><th>Read operation</th><th>Write operation</th></tr>
<tr><td>Mirror area 1</td><td>Normal read</td><td>Write the value with ECC calculated</td></tr>
<tr><td>Mirror area 2</td><td>Reserved area (access prohibited)</td><td>Reserved area (access prohibited)</td></tr>
<tr><td>Mirror area 3</td><td>Normal read</td><td>Write the value without ECC</td></tr>
<tr><td>Mirror area 4</td><td>Normal memory attribute read</td><td>Access prohibited</td></tr>
</table></td></tr>
<tr><td>1099</td><td>CHAPTER 18:WorkFLASH
2.Configuration
2.2.Address/Sector Maps in User Mode</td><td>"Figure 2-2" should be corrected as indicated by the shading below

(Error)
Reserved area (*1)
Memory attribute access(*1)
(*1) Accessing this region results in a bus error, but the reserved area access bit (WFCFG_BERR:RESA) is not set.

(Correct)
Reserved area (*2)
Normal memory attribute access (*1)
(*2) Accessing this region results in a bus error, but the reserved area access bit (WFCFG_BERR:RESA) is not set.</td></tr>
<tr><td>1106</td><td>CHAPTER 18:WorkFLASH
3.5. ECC Generation and Check</td><td>Revised the word “Cortex(TM)-R5F” to “Cortex(TM)-R5”</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1111</td><td>CHAPTER 18:WorkFLASH
3. Explanation of Operation
3.7. Bus Error Response</td><td>Features of "For details" should be corrected as indicated by the shading below

(Error)
<table><tr><td>- Writing to the registers other than below with unpreviledge mode.
- WFCFG_WCR
- WFCFG_SEQCM
- WFCFG_EMENR
- WFCFG_ARBCLR
- WFCFG_BERRCLR
- FCFG_WARBR</td><td>6</td><td>UNACC</td></tr></table>
(Correct)
<table><tr><td>- Writing to the registers other than below with unprivileged mode.
- WFCFG_WCR
- WFCFG_SEQCM
- WFCFG_EMENR
- WFCFG_ARBCLR
- WFCFG_BERRCLR
- FCFG_WARBR</td><td>6</td><td>UNACC</td></tr></table></td></tr>
<tr><td>1114</td><td>CHAPTER 18:WorkFLASH
4.2. Enabling Writing</td><td>Revised an arrow to “Figure 4-1 Procedure for Setting Flash Write Enable”.</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1120</td><td>CHAPTER 18:WorkFLASH
5.Registers</td><td>
Features of "For details" should be corrected as indicated by the shading below

Table 5-1 Register Map
(Error)

Table 5-1 Register Map
<table>
<tr><th>Abbreviated Register Name</th><th>Description</th><th>Reference</th></tr>
<tr><td>WFCFG_CPR</td><td>WorkFLASH configuration protection key register</td><td>5.1</td></tr>
<tr><td>WFCFG_CR</td><td>WorkFLASH configuration register</td><td>5.2</td></tr>
<tr><td>WFCFG_ECR</td><td>WorkFLASH ECC control register</td><td>5.3</td></tr>
<tr><td>WFCFG_WCR</td><td>WorkFLASH write command sequencer configuration register</td><td>5.4</td></tr>
<tr><td>WFCFG_WSR</td><td>WorkFLASH write command sequencer status register</td><td>5.5</td></tr>
<tr><td>WFCFG_DBEIR</td><td>WorkFLASH data bit error injection register</td><td>5.6</td></tr>
<tr><td>WFCFG_EEIR</td><td>WorkFLASH ECC bit error injection register</td><td>5.7</td></tr>
<tr><td>WFCFG_ICR</td><td>WorkFLASH interrupt control register</td><td>5.8</td></tr>
<tr><td>WFCFG_SR</td><td>WorkFLASH status register</td><td>5.9</td></tr>
<tr><td>WFCFG_SECIR</td><td>WorkFLASH SEC interrupt register</td><td>5.10</td></tr>
<tr><td>WFCFG_EEAR</td><td>WorkFLASH ECC error address register</td><td>5.11</td></tr>
<tr><td>WFCFG_EMENR</td><td>WorkFLASH extra mode enable register</td><td>5.12</td></tr>
<tr><td>WFCFG_SEQCM</td><td>WorkFLASH sequencer command register</td><td>5.13</td></tr>
<tr><td>WFCFG_ARBERR</td><td>WorkFLASH arbitration error register</td><td>5.14</td></tr>
<tr><td>WFCFG_ABRCLR</td><td>WorkFLASH arbitration error clear register</td><td>5.15</td></tr>
<tr><td>WFCFG_BERR</td><td>WorkFLASH bus error response status register</td><td>5.16</td></tr>
<tr><td>WFCFG_BERRCLR</td><td>WorkFLASH bus error response status clear register</td><td>5.17</td></tr>
<tr><td>WFCFG_UCESR</td><td>WorkFLASH uncorrectable error status register</td><td>5.18</td></tr>
<tr><td>WFCFG_UCEAR</td><td>WorkFLASH uncorrectable error address register</td><td>5.19</td></tr>
<tr><td>FCFG_WARBR</td><td>FLASH write arbitration register</td><td>5.20</td></tr>
</table>

(Correct)
Table 5-1 Register Map
<table>
<tr><th>Abbreviated Register Name</th><th>Description</th><th>Reference</th></tr>
<tr><td>WFCFG_CPR</td><td>WorkFLASH configuration protection key register</td><td>5.1</td></tr>
<tr><td>WFCFG_CR</td><td>WorkFLASH configuration register</td><td>5.2</td></tr>
<tr><td>WFCFG_ECR</td><td>WorkFLASH ECC control register</td><td>5.3</td></tr>
<tr><td>WFCFG_WCR</td><td>WorkFLASH write command sequencer configuration register</td><td>5.4</td></tr>
<tr><td>WFCFG_WSR</td><td>WorkFLASH write command sequencer status register</td><td>5.5</td></tr>
<tr><td>WFCFG_DBEIR</td><td>WorkFLASH data bit error injection register</td><td>5.6</td></tr>
<tr><td>WFCFG_EEIR</td><td>WorkFLASH ECC bit error injection register</td><td>5.7</td></tr>
<tr><td>WFCFG_ICR</td><td>WorkFLASH interrupt control register</td><td>5.8</td></tr>
<tr><td>WFCFG_SR</td><td>WorkFLASH status register</td><td>5.9</td></tr>
<tr><td>WFCFG_SECIR</td><td>WorkFLASH SEC interrupt register</td><td>5.10</td></tr>
<tr><td>WFCFG_EEAR</td><td>WorkFLASH ECC error address register</td><td>5.11</td></tr>
<tr><td>WFCFG_EMENR</td><td>WorkFLASH extra mode enable register</td><td>5.12</td></tr>
<tr><td>WFCFG_SEQCM</td><td>WorkFLASH sequencer command register</td><td>5.13</td></tr>
<tr><td>WFCFG_ARBERR</td><td>WorkFLASH arbitration error register</td><td>5.14</td></tr>
<tr><td>WFCFG_ARBCLR</td><td>WorkFLASH arbitration error clear register</td><td>5.15</td></tr>
<tr><td>WFCFG_BERR</td><td>WorkFLASH bus error response status register</td><td>5.16</td></tr>
<tr><td>WFCFG_BERRCLR</td><td>WorkFLASH bus error response status clear register</td><td>5.17</td></tr>
<tr><td>WFCFG_UCESR</td><td>WorkFLASH uncorrectable error status register</td><td>5.18</td></tr>
<tr><td>WFCFG_UCEAR</td><td>WorkFLASH uncorrectable error address register</td><td>5.19</td></tr>
<tr><td>FCFG_WARBR</td><td>FLASH write arbitration register</td><td>5.20</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1124</td><td>CHAPTER 18:WorkFLASH
5.Registers
5.3. WorkFLASH ECC Control Register: WFCFG_ECR</td><td>Features of "For details" should be corrected as indicated by the shading below

(Error)
You can change the setting value of this register only once.

(Correct)
You can write this register only once.</td></tr>
<tr><td>1152</td><td>CHAPTER 18:WorkFLASH
6.Other</td><td>Features of "For details" should be corrected as indicated by the shading below

(Error)
To generate a software reset, the write enable (TCFCFG_FCFGR:WE) bit must be "1".

(Correct)
To generate a software reset, the write enable (WFCFG_CR:WE) bit must be "1".</td></tr>
<tr><td>1152</td><td>CHAPTER 18:WorkFLASH
6.Other</td><td>Features of "For details" should be added as indicated by the shading below

MPU attribute/Write access type
The MPU attribute is supported the below table when each master writes to the Work Flash.
<table>
<tr><th></th><th colspan="2">CPU</th><th>DMA</th><th>DAP</th></tr>
<tr><th></th><th>Normal</th><th>Device</th><th></th><th></th></tr>
<tr><td>Dword</td><td>Support</td><td>Non-support</td><td>Support</td><td>Non-support</td></tr>
<tr><td>Word</td><td>Not recommend</td><td>Support</td><td>Support</td><td>Support</td></tr>
<tr><td>Hword</td><td>Not recommend</td><td>Support</td><td>Support</td><td>Support</td></tr>
<tr><td>Byte</td><td>Not recommend</td><td>Support</td><td>Support</td><td>Support</td></tr>
</table></td></tr>
<tr><td>1166</td><td>CHAPTER 19:BootROM Hardware Interface
5.Registers
5.3.EXCFG Inactive Set - Undefined Instruction Vector Register (EXCFG_UNDEFINACT)</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></td><td></td><td></td><td></td><td></td><td></td><td>0</td></tr>
<tr><td>Field</td><td colspan="8">UNDEFVEC</td></tr>
<tr><td>R/W Attribute</td><td colspan="8">R,W</td></tr>
<tr><td>Protection Attribute</td><td colspan="8">WPL</td></tr>
<tr><td>Initial Value</td><td colspan="8">11111111_11111111_00000000_00100100</td></tr>
</table>
(Correct)
<table>
<tr><td>Bit</td><td>31</td><td></td><td></td><td></td><td></td><td></td><td></td><td>0</td></tr>
<tr><td>Field</td><td colspan="8">UNDEFVEC</td></tr>
<tr><td>R/W Attribute</td><td colspan="8">R,W</td></tr>
<tr><td>Protection Attribute</td><td colspan="8">WPS</td></tr>
<tr><td>Initial Value</td><td colspan="8">11111111_11111111_00000000_00100100</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1167</td><td>CHAPTER 19:BootROM Hardware Interface
5.4.EXCFG Inactive Set - Supervisor Call Vector Register (EXCFG_SVCINACT)</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></td><td></td><td></td><td></td><td></td><td></td><td>0</td></tr><tr><td>Field</td><td colspan="8">SVCVEC</td></tr><tr><td>R/W Attribute</td><td colspan="8">R,W</td></tr><tr><td>Protection Attribute</td><td colspan="8">WPL</td></tr><tr><td>Initial Value</td><td colspan="8">11111111_11111111_00000000_00101000</td></tr></table>
(Correct)
<table><tr><td>Bit</td><td>31</td><td></td><td></td><td></td><td></td><td></td><td></td><td>0</td></tr><tr><td>Field</td><td colspan="8">SVCVEC</td></tr><tr><td>R/W Attribute</td><td colspan="8">R,W</td></tr><tr><td>Protection Attribute</td><td colspan="8">WPS</td></tr><tr><td>Initial Value</td><td colspan="8">11111111_11111111_00000000_00101000</td></tr></table></td></tr>
<tr><td>1168</td><td>CHAPTER 19:BootROM Hardware Interface
5.5.EXCFG Inactive Set - Prefetch Abort Vector Register (EXCFG_PABORTINACT)</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></td><td></td><td></td><td></td><td></td><td></td><td>0</td></tr><tr><td>Field</td><td colspan="8">PABORTVEC</td></tr><tr><td>R/W Attribute</td><td colspan="8">R,W</td></tr><tr><td>Protection Attribute</td><td colspan="8">WPL</td></tr><tr><td>Initial Value</td><td colspan="8">11111111_11111111_00000000_00101100</td></tr></table>
(Correct)
<table><tr><td>Bit</td><td>31</td><td></td><td></td><td></td><td></td><td></td><td></td><td>0</td></tr><tr><td>Field</td><td colspan="8">PABORTVEC</td></tr><tr><td>R/W Attribute</td><td colspan="8">R,W</td></tr><tr><td>Protection Attribute</td><td colspan="8">WPS</td></tr><tr><td>Initial Value</td><td colspan="8">11111111_11111111_00000000_00101100</td></tr></table></td></tr>
<tr><td>1169</td><td>CHAPTER 19:BootROM Hardware Interface
5.6.EXCFG Inactive Set - Data Abort Vector Register (EXCFG_DABORTINACT)</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></td><td></td><td></td><td></td><td></td><td></td><td>0</td></tr><tr><td>Field</td><td colspan="8">DABORTVEC</td></tr><tr><td>R/W Attribute</td><td colspan="8">R,W</td></tr><tr><td>Protection Attribute</td><td colspan="8">WPL</td></tr><tr><td>Initial Value</td><td colspan="8">11111111_11111111_00000000_00110000</td></tr></table>
(Correct)
<table><tr><td>Bit</td><td>31</td><td></td><td></td><td></td><td></td><td></td><td></td><td>0</td></tr><tr><td>Field</td><td colspan="8">DABORTVEC</td></tr><tr><td>R/W Attribute</td><td colspan="8">R,W</td></tr><tr><td>Protection Attribute</td><td colspan="8">WPS</td></tr><tr><td>Initial Value</td><td colspan="8">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>1170</td><td>CHAPTER 19:BootROM Hardware Interface
5.7.EXCFG Active Set - Undefined Instruction Vector Register (EXCFG_UNDEFACT)</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></td><td></td><td></td><td></td><td></td><td></td><td>0</td></tr><tr><td>Field</td><td colspan="8">UNDEFVEC</td></tr><tr><td>R/W Attribute</td><td colspan="8">R,W</td></tr><tr><td>Protection Attribute</td><td colspan="8">WPL</td></tr><tr><td>Initial Value</td><td colspan="8">11111111_11111111_00000000_00100100</td></tr></table>
(Correct)
<table><tr><td>Bit</td><td>31</td><td></td><td></td><td></td><td></td><td></td><td></td><td>0</td></tr><tr><td>Field</td><td colspan="8">UNDEFVEC</td></tr><tr><td>R/W Attribute</td><td colspan="8">R,W</td></tr><tr><td>Protection Attribute</td><td colspan="8">WPS</td></tr><tr><td>Initial Value</td><td colspan="8">11111111_11111111_00000000_00100100</td></tr></table></td></tr>
<tr><td>1171</td><td>CHAPTER 19:BootROM Hardware Interface
5.8.EXCFG Active Set - Supervisor Call Vector Register (EXCFG_SVCACT)</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></td><td></td><td></td><td></td><td></td><td></td><td>0</td></tr><tr><td>Field</td><td colspan="8">SVCVEC</td></tr><tr><td>R/W Attribute</td><td colspan="8">R,W</td></tr><tr><td>Protection Attribute</td><td colspan="8">WPL</td></tr><tr><td>Initial Value</td><td colspan="8">11111111_11111111_00000000_00101000</td></tr></table>
(Correct)
<table><tr><td>Bit</td><td>31</td><td></td><td></td><td></td><td></td><td></td><td></td><td>0</td></tr><tr><td>Field</td><td colspan="8">SVCVEC</td></tr><tr><td>R/W Attribute</td><td colspan="8">R,W</td></tr><tr><td>Protection Attribute</td><td colspan="8">WPS</td></tr><tr><td>Initial Value</td><td colspan="8">11111111_11111111_00000000_00101000</td></tr></table></td></tr>
<tr><td>1172</td><td>CHAPTER 19:BootROM Hardware Interface
5.9.EXCFG Active Set - Prefetch Abort Vector Register (EXCFG_PABORTACT)</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></td><td></td><td></td><td></td><td></td><td></td><td>0</td></tr><tr><td>Field</td><td colspan="8">PABORTVEC</td></tr><tr><td>R/W Attribute</td><td colspan="8">R,W</td></tr><tr><td>Protection Attribute</td><td colspan="8">WPL</td></tr><tr><td>Initial Value</td><td colspan="8">11111111_11111111_00000000_00101100</td></tr></table>
(Correct)
<table><tr><td>Bit</td><td>31</td><td></td><td></td><td></td><td></td><td></td><td></td><td>0</td></tr><tr><td>Field</td><td colspan="8">PABORTVEC</td></tr><tr><td>R/W Attribute</td><td colspan="8">R,W</td></tr><tr><td>Protection Attribute</td><td colspan="8">WPS</td></tr><tr><td>Initial Value</td><td colspan="8">11111111_11111111_00000000_00101100</td></tr></table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1173</td><td>CHAPTER 19:BootROM Hardware Interface
5.10.EXCFG Active Set - Data Abort Vector Register (EXCFG_DABORTACT)</td><td>Features of "For details" should be corrected as indicated by the shading below

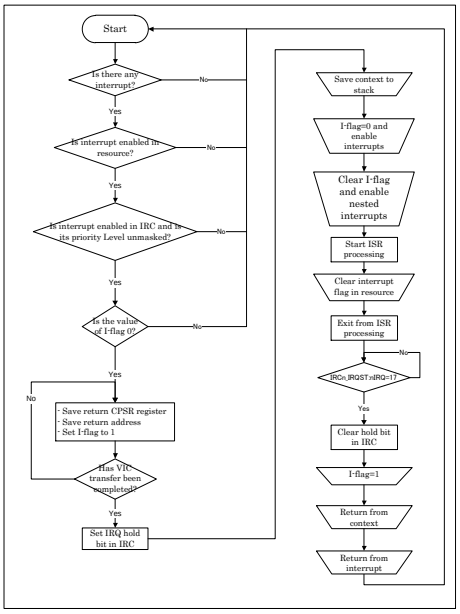
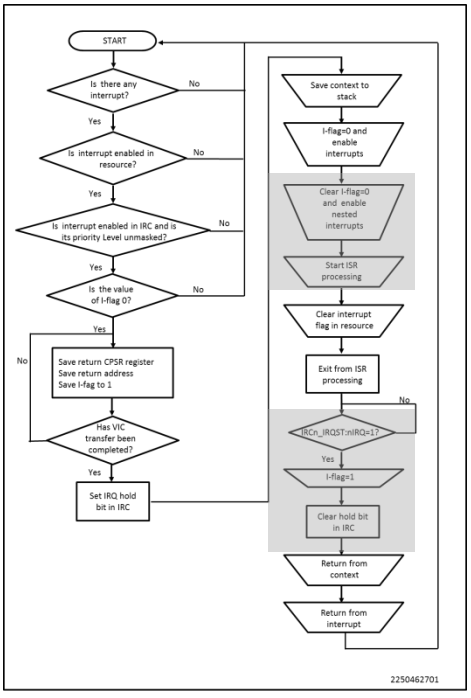
(Error)
<table><tr><td>Bit</td><td>31</td><td></td><td></td><td></td><td></td><td></td><td></td><td>0</td></tr><tr><td>Field</td><td colspan="8">DABORTVEC</td></tr><tr><td>R/W Attribute</td><td colspan="8">R,W</td></tr><tr><td>Protection Attribute</td><td colspan="8">WPL</td></tr><tr><td>Initial Value</td><td colspan="8">11111111_11111111_00000000_00110000</td></tr></table>
(Correct)
<table><tr><td>Bit</td><td>31</td><td></td><td></td><td></td><td></td><td></td><td></td><td>0</td></tr><tr><td>Field</td><td colspan="8">DABORTVEC</td></tr><tr><td>R/W Attribute</td><td colspan="8">R,W</td></tr><tr><td>Protection Attribute</td><td colspan="8">WPS</td></tr><tr><td>Initial Value</td><td colspan="8">11111111_11111111_00000000_00110000</td></tr></table></td></tr>
<tr><td>1179</td><td>CHAPTER 20:BootROM Software Interface
2.BootROM Markers
2.2.Register Lists</td><td>Features of "For details" should be corrected as indicated by the shading below

Table 2-1.
(Error)
<table><tr><th>Abbreviated Register Name</th><th>Register Name</th><th>Reference</th></tr><tr><td>MK_SER</td><td>Security Enable Marker</td><td>2.3</td></tr><tr><td>MK_SSR</td><td>Security Scope Marker</td><td>2.3</td></tr><tr><td>MK_CEER</td><td>Chip Erase Enable Marker</td><td>2.3</td></tr><tr><td>MK_SOER</td><td>Security Overw rite Enable Marker</td><td>2.3</td></tr><tr><td>MK_SWPOER</td><td>Sector Write Permission Overw rite Enable Marker</td><td>2.3</td></tr><tr><td>MK_WSWPR</td><td>Work Flash Sector Write Permissions Marker</td><td>2.3</td></tr><tr><td>MK_C0SWPR</td><td>Code Flash Sector Write Permissions of the small sectors Marker</td><td>2.3</td></tr><tr><td>MK_C1SWPR</td><td>Code Flash Sector Write Permissions of the large sectors Marker</td><td>2.3</td></tr></table>
(Correct)
<table><tr><th>Abbreviated Register Name</th><th>Register Name</th><th>Reference</th></tr><tr><td>MK_SER</td><td>Security Enable Marker</td><td>2.3</td></tr><tr><td>MK_SSR</td><td>Security Scope Marker</td><td>2.3</td></tr><tr><td>MK_CEER</td><td>Chip Erase Enable Marker</td><td>2.3</td></tr><tr><td>MK_SOER</td><td>Security Overw rite Enable Marker</td><td>2.3</td></tr><tr><td>MK_SWPOER</td><td>Sector Write Permission Overw rite Enable Marker</td><td>2.3</td></tr><tr><td>MK_WSWPR</td><td>Work Flash Sector Write Permissions Marker</td><td>2.3</td></tr><tr><td>MK_C0SWPR</td><td>Code Flash 0 Sector Write Permissions of the small sectors Marker</td><td>2.3</td></tr><tr><td>MK_C1SWPR</td><td>Code Flash 0 Sector Write Permissions of the large sectors Marker</td><td>2.3</td></tr><tr><td>MK_C2SWPR</td><td>Code Flash 1 Sector Write Permissions of the large sectors Marker</td><td>2.3</td></tr><tr><td>MK_C3SWPR</td><td>Code Flash 2 Sector Write Permissions of the large sectors Marker</td><td>2.3</td></tr></table></td></tr>
</table>

Page	Section	Change Results
1216	CHAPTER 21:Exclusive Access Memory (EAM) 1. Overview	Revised the word “Cortex-R5F” to “Cortex-R5”
1219	CHAPTER 21:Exclusive Access Memory (EAM) 3.Explanation of Operation	Features of "For details" should be corrected as indicated by the shading below (Error) 3.Explanation of Operation 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) 3.Explanation of Operation 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.)

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1229</td><td>CHAPTER 21:Exclusive Access Memory (EAM)
4. Operation Examples</td><td>"Table 4-4" should be deleted as indicated by the shading below
(Error)
Table 4-4 When ID and Address Range of Exclusive Load Instruction (LDREXx) Do Not Match Those of Exclusive Store Instruction (STREXx)
<table>
<tr><th colspan="2">Monitoring Register 1</th><th colspan="2">Monitoring Register 2</th><th colspan="2">Monitoring Register 3</th><th colspan="2">Monitoring Register 4</th></tr>
<tr><td>ID</td><td>ADD</td><td>ID</td><td>ADD</td><td>ID</td><td>ADD</td><td>ID</td><td>ADD</td></tr>
<tr><td>0</td><td>A</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>0</td><td>A</td><td>1</td><td>B</td><td></td><td></td><td></td><td></td></tr>
<tr><td>0</td><td>A</td><td>1</td><td>B</td><td>2</td><td>C</td><td></td><td></td></tr>
<tr><td>0</td><td>A</td><td>1</td><td>B</td><td>2</td><td>C</td><td>3</td><td>D</td></tr>
<tr><td>0</td><td>A</td><td>1</td><td>B</td><td>2</td><td>C</td><td>3</td><td>D</td></tr>
<tr><td>0</td><td>A</td><td>1</td><td>B</td><td>2</td><td>C</td><td>3</td><td>D</td></tr>
<tr><td>0</td><td>A</td><td>1</td><td>B</td><td>2</td><td>C</td><td>3</td><td>D</td></tr>
<tr><td>0</td><td>A</td><td>1</td><td>B</td><td>2</td><td>C</td><td>3</td><td>D</td></tr>
</table>
(Correct)
Table 4-4 When ID and Address Range of Exclusive Load Instruction (LDREXx) Do Not Match Those of Exclusive Store Instruction (STREXx)
<table>
<tr><th colspan="2">Monitoring Register 1</th><th colspan="2">Monitoring Register 2</th><th colspan="2">Monitoring Register 3</th><th colspan="2">Monitoring Register 4</th></tr>
<tr><td>ID</td><td>ADD</td><td>ID</td><td>ADD</td><td>ID</td><td>ADD</td><td>ID</td><td>ADD</td></tr>
<tr><td>0</td><td>A</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>0</td><td>A</td><td>1</td><td>B</td><td></td><td></td><td></td><td></td></tr>
<tr><td>0</td><td>A</td><td>1</td><td>B</td><td>2</td><td>C</td><td></td><td></td></tr>
<tr><td>0</td><td>A</td><td>1</td><td>B</td><td>2</td><td>C</td><td>3</td><td>D</td></tr>
<tr><td></td><td></td><td>1</td><td>B</td><td>2</td><td>C</td><td>3</td><td>D</td></tr>
<tr><td></td><td></td><td></td><td></td><td>2</td><td>C</td><td>3</td><td>D</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td>3</td><td>D</td></tr>
<tr><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>1230</td><td>CHAPTER 21:Exclusive Access Memory (EAM)
4. Operation Examples</td><td>"Table 4-6" should be deleted as indicated by the shading below
(Error)
Table 4-6 When 5 Exclusive Load Instructions (LDREXx) Are Received
<table>
<tr><th rowspan="2">Response</th><th colspan="2">Monitoring Register 1</th><th colspan="2">Monitoring Register 2</th><th colspan="2">Monitoring Register 3</th><th colspan="2">Monitoring Register 4</th></tr>
<tr><td>ID</td><td>ADD</td><td>ID</td><td>ADD</td><td>ID</td><td>ADD</td><td>ID</td><td>ADD</td></tr>
<tr><td>EXOKAY</td><td>0</td><td>A</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>EXOKAY</td><td>0</td><td>A</td><td>1</td><td>B</td><td></td><td></td><td></td><td></td></tr>
<tr><td>EXOKAY</td><td>0</td><td>A</td><td>1</td><td>B</td><td>2</td><td>C</td><td></td><td></td></tr>
<tr><td>EXOKAY</td><td>0</td><td>A</td><td>1</td><td>B</td><td>2</td><td>C</td><td>3</td><td>D</td></tr>
<tr><td>EXOKAY</td><td>0</td><td>E</td><td>1</td><td>B</td><td>2</td><td>C</td><td>3</td><td>D</td></tr>
<tr><td>OKAY</td><td>0</td><td>E</td><td>1</td><td>B</td><td>2</td><td>C</td><td>3</td><td>D</td></tr>
<tr><td>EXOKAY</td><td>0</td><td>E</td><td></td><td></td><td>2</td><td>C</td><td>3</td><td>D</td></tr>
<tr><td>EXOKAY</td><td>0</td><td>E</td><td></td><td></td><td></td><td></td><td>3</td><td>D</td></tr>
<tr><td>EXOKAY</td><td>0</td><td>E</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>EXOKAY</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>
(Correct)
Table 4-6 When 5 Exclusive Load Instructions (LDREXx) Are Received
<table>
<tr><th rowspan="2">Response</th><th colspan="2">Monitoring Register 1</th><th colspan="2">Monitoring Register 2</th><th colspan="2">Monitoring Register 3</th><th colspan="2">Monitoring Register 4</th></tr>
<tr><td>ID</td><td>ADD</td><td>ID</td><td>ADD</td><td>ID</td><td>ADD</td><td>ID</td><td>ADD</td></tr>
<tr><td>EXOKAY</td><td>0</td><td>A</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>EXOKAY</td><td>0</td><td>A</td><td>1</td><td>B</td><td></td><td></td><td></td><td></td></tr>
<tr><td>EXOKAY</td><td>0</td><td>A</td><td>1</td><td>B</td><td>2</td><td>C</td><td></td><td></td></tr>
<tr><td>EXOKAY</td><td>0</td><td>A</td><td>1</td><td>B</td><td>2</td><td>C</td><td>3</td><td>D</td></tr>
<tr><td>EXOKAY</td><td>0</td><td>E</td><td>1</td><td>B</td><td>2</td><td>C</td><td>3</td><td>D</td></tr>
<tr><td>OKAY</td><td></td><td></td><td>1</td><td>B</td><td>2</td><td>C</td><td>3</td><td>D</td></tr>
<tr><td>EXOKAY</td><td></td><td></td><td></td><td></td><td>2</td><td>C</td><td>3</td><td>D</td></tr>
<tr><td>EXOKAY</td><td></td><td></td><td></td><td></td><td></td><td></td><td>3</td><td>D</td></tr>
<tr><td>EXOKAY</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>OKAY</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table></td></tr>
</table>

Page	Section	Change Results
1247	CHAPTER 22:Interrupt Controller 3.Explanation of Operation	Features of "For details" should be corrected as indicated by the shading below Figure 3 4 Nest supported IRQ processing Flow. (Error)  (Correct) 

Page	Section	Change Results
1252	CHAPTER 22:Interrupt Controller 3.Explanation of Operation	Features of "For details" should be deleted as indicated by the shading below (Error) 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 Disabling interrupts for I-FLAG After temporarily disabling all interrupts for CPU I-FLAG using the I-FLAG set instruction, do not write data to each of IRCn_CSR:IRQEN, IRCn_IRQPLM, or IRCn_IRQHC. 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. (Correct) 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.

1253	CHAPTER 22:Interrupt Controller
	3.Explanation of Operation
		Features of "For details" should be corrected as indicated by the shading below
		
		(Error)
		Changing an interrupt level using IRC_IRQPLM with interrupts enabled
		Before rewriting IRCn_IRQPLM, 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
		
		(Correct)
		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

Page	Section	Change Results
1253	CHAPTER 22:Interrupt Controller 3.Explanation of Operation	Features of "For details" should be corrected as indicated by the shading below (Error) 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_IRQPL1-7 - IRCn_IRQS0-15 - IRCn_IRQR0-15 - IRCn_IRQCES0-15 - IRCn_IRQCEC0-15 - IRCn_IRQCE0-15 - IRCn_IRQHC - IRCn_IRQPLM - IRCn_CSR (Correct) 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

Page	Section	Change Results
1260	CHAPTER 22:Interrupt Controller 5.4.IRC IRQ Status Register (IRCn_IRQST)	Features of "For details" should be corrected as indicated by the shading below (Error) [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_IRQPL1-7 - IRCn_IRQS0-15 - IRCn_IRQR0-15 - IRCn_IRQCES0-15 - IRCn_IRQCEC0-15 - IRCn_IRQCE0-15 - IRCn_IRQHC - IRCn_IRQPLM - IRCn_CSR (Correct) [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

Page	Section	Change Results
1268	CHAPTER 22:Interrupt Controller 5.9.IRC IRQ Priority Level Register (IRCn_IRQPL0 to IRCn_IRQPL127)	Features of "For details" should be corrected as indicated by the shading below (Error) [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. - The "r" in bit field name IRQPLr corresponds to the IRQ channel number, r (0 to 511). (Correct) [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.).

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1282</td><td>CHAPTER 22:Interrupt Controller
5.18.IRC IRQ Channel Enable Setting Register (IRCn_IRQCE0 to IRCn_IRQCE15)</td><td>Features of "For details" should be corrected as indicated by the shading below

(Error)

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.

<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>IRQCE31</td><td>IRQCE30</td><td>IRQCE29</td><td>IRQCE28</td><td>IRQCE27</td><td>IRQCE26</td><td>IRQCE25</td><td>IRQCE24</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>IRQCE23</td><td>IRQCE22</td><td>IRQCE21</td><td>IRQCE20</td><td>IRQCE19</td><td>IRQCE18</td><td>IRQCE17</td><td>IRQCE16</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>IRQCE15</td><td>IRQCE14</td><td>IRQCE13</td><td>IRQCE12</td><td>IRQCE11</td><td>IRQCE10</td><td>IRQCE9</td><td>IRQCE8</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>IRQCE7</td><td>IRQCE6</td><td>IRQCE5</td><td>IRQCE4</td><td>IRQCE3</td><td>IRQCE2</td><td>IRQCE1</td><td>IRQCE0</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>

(Correct)

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.

<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>IRQCE31</td><td>IRQCE30</td><td>IRQCE29</td><td>IRQCE28</td><td>IRQCE27</td><td>IRQCE26</td><td>IRQCE25</td><td>IRQCE24</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>IRQCE23</td><td>IRQCE22</td><td>IRQCE21</td><td>IRQCE20</td><td>IRQCE19</td><td>IRQCE18</td><td>IRQCE17</td><td>IRQCE16</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>IRQCE15</td><td>IRQCE14</td><td>IRQCE13</td><td>IRQCE12</td><td>IRQCE11</td><td>IRQCE10</td><td>IRQCE9</td><td>IRQCE8</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>IRQCE7</td><td>IRQCE6</td><td>IRQCE5</td><td>IRQCE4</td><td>IRQCE3</td><td>IRQCE2</td><td>IRQCE1</td><td>IRQCE0</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>
</td></tr>
</table>

Page	Section	Change Results
1286	CHAPTER 22:Interrupt Controller 5.21.IRC IRQ Hold Clear Register (IRCn_IRQHC)	Features of "For details" should be added as indicated by the shading below (Error) [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. Note: - This register is forbidded 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. (Correct) [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 forbidded 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.).
1289	CHAPTER 22:Interrupt Controller 5.23.IRC IRQ Priority Level Mask Register (IRCn_IRQPLM)	Features of "For details" should be added as indicated by the shading below (Error) [bit5:0] IRQPLM: IRQ priority level mask bits These bits set the priority level mask value. (Correct) [bit5:0] IRQPLM: IRQ priority level mask bits These bits set the priority level mask value. Notes: - 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.).

Page	Section	Change Results
1291	CHAPTER 22:Interrupt Controller 5.24.IRC Control/Status Register (IRCn_CSR)	Features of "For details" should be added as indicated by the shading below (Error) [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. (Correct) [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. Notes: - 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.).
1315	CHAPTER 23:Time Protection 4. Setting Procedure Examples	"Figure 4-1" should be corrected as indicated by the shading below (Error) TPUn (Correct) TPU0
1316	CHAPTER 23:Time Protection 4. Setting Procedure Examples	"Figure 4-2" should be corrected as indicated by the shading below (Error) TPUn (Correct) TPU0

Page	Section	Change Results
1317	CHAPTER 23:Time Protection 5.Registers	Features of "For details" should be deleted as indicated by the shading below (Error) 5.Registers This section explains the registers used by the time protection unit. Base address is 0xB040_8000. (Correct) 5.Registers This section explains the registers used by the time protection unit.

Page	Section	Change Results
1324	CHAPTER 23:Time Protection 5.Registers 5.6.TPU Timer Interrupt Enable Register (TPU0_TIE)	Features of "For details" should be corrected as indicated by the shading below (Error) **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 (Correct) **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

Page	Section	Change Results
1333	CHAPTER 24:Security 2.1.Security Scope	Features of "For details" should be deleted as indicated by the shading below (Error) 2.1 Security Scope This section explains scopes of the security supported in FCR4. Security Scope This product has following "security scope" to protect the device from unauthorized access. (Correct) 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.
1333	CHAPTER 24:Security 2. Security Scope and Access Restriction by the Security 2.1.Security Scope	Features of "For details" should be corrected as indicated by the shading below (Error) Device Protection The ability to execute “chip erase” always exists. The marker MK_CEER does not have any functionality. (Correct) Device Protection 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.

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1343</td><td>CHAPTER 24:Security
3. Using and Configuring the Security
3.3.3.Chip Erase Enable (MK_CEER)</td><td>Features of "For details" should be corrected as indicated by the shading below
(Error)
3.3.3.Chip Erase Enable (MK_CEER)
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.

(Correct)
3.3.3.Chip Erase Enable (MK_CEER)
[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.</td></tr>
<tr><td>1349</td><td>CHAPTER 24:Security
3. Using and Configuring the Security
3.4.Gain Access in USERMODE</td><td>Features of "For details" should be added as indicated by the shading below

(Correct)
In non-USERMODE Flash access can only be forbidden by the activation of security.
Full-chip erase can be forbidden using the MK_CEER marker also for non-USERMODE. It is required to enable it together with Security.</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1352</td><td>CHAPTER 24:Security
4.Registers
4.1.Security Status Register (TCFCFG0_SECSTAT)</td><td>Features of "For details" should be corrected as indicated by the shading below

(Error)
4.1. Security Status Register (TCFCFG0_SECSTAT)
This section describes the function of the Security Status Register.
<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">-</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><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">-</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><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>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</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>
<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>1</td><td>0 or 1</td><td>0 or 1</td></tr>
</table>

(Correct)
4.1. Security Status Register (TCFCFG0_SECSTAT)
This section describes the function of the Security Status Register.
<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>
</table>
<table>
<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>
</table>
<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 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>
<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>1</td><td>0 or 1</td><td>0 or 1</td></tr>
</table>
</td></tr>
</table>

Page	Section	Change Results
1353	CHAPTER 24: Security 4. Registers 4.1.Security Status Register (TCFCFG0_SECSTAT)	Features of "For details" should be deleted as indicated by the shading below (Error) [2] CEEN: Chip Erase Enable Register In this product CEEN is always ‘1’ (fixed, because CEEM marker is not supported).

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1355</td><td>CHAPTER 24:Security
4.Registers
4.2.Security Enable Register
(TCFCFG0_SER)</td><td>
Features of "For details" should be corrected as indicated by the shading below:

(Error)
4.2. Security Enable Register (TCFCFG0_SER)
This section describes the function of the Security Enable Register.
<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">SER[31:24]</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 or 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><td>0 or 1</td><td>0 or 1</td></tr>
</table>
<table>
<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">SER[23:16]</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 or 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><td>0 or 1</td><td>0 or 1</td></tr>
</table>
<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 colspan="8">SER[15:8]</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 or 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><td>0 or 1</td><td>0 or 1</td></tr>
</table>
<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 colspan="8">SER[7:0]</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 or 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><td>0 or 1</td><td>0 or 1</td></tr>
</table>

(Correct)
4.2. Security Enable Register (TCFCFG0_SER)
This section describes the function of the Security Enable Register.
<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">SER[31:24]</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">WP</td></tr>
<tr><td>INITIAL_VALUE</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><td>0 or 1</td><td>0 or 1</td><td>0 or 1</td></tr>
</table>
<table>
<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">SER[23:16]</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">WP</td></tr>
<tr><td>INITIAL_VALUE</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><td>0 or 1</td><td>0 or 1</td><td>0 or 1</td></tr>
</table>
<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 colspan="8">SER[15:8]</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">WP</td></tr>
<tr><td>INITIAL_VALUE</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><td>0 or 1</td><td>0 or 1</td><td>0 or 1</td></tr>
</table>
<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 colspan="8">SER[7:0]</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">WP</td></tr>
<tr><td>INITIAL_VALUE</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><td>0 or 1</td><td>0 or 1</td><td>0 or 1</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1356</td><td>CHAPTER 24:Security
4.Registers
4.3.Security Scope Register
(TCFCFG0_SSR)</td><td>
Features of "For details" should be corrected as indicated by the shading below:

(Error)
4.3. Security Scope Register (TCFCFG0_SSR)
This section describes the function of the Security Scope Register.
<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">SSR[31:24]</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 or 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><td>0 or 1</td><td>0 or 1</td></tr>
</table>
<table>
<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">SSR[23:16]</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 or 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><td>0 or 1</td><td>0 or 1</td></tr>
</table>
<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 colspan="8">SSR[15:8]</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 or 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><td>0 or 1</td><td>0 or 1</td></tr>
</table>
<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 colspan="8">SSR[7:0]</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 or 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><td>0 or 1</td><td>0 or 1</td></tr>
</table>

(Correct)
4.3. Security Scope Register (TCFCFG0_SSR)
This section describes the function of the Security Scope Register.
<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">SSR[31:24]</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">WP</td></tr>
<tr><td>INITIAL_VALUE</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><td>0 or 1</td><td>0 or 1</td><td>0 or 1</td></tr>
</table>
<table>
<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">SSR[23:16]</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">WP</td></tr>
<tr><td>INITIAL_VALUE</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><td>0 or 1</td><td>0 or 1</td><td>0 or 1</td></tr>
</table>
<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 colspan="8">SSR[15:8]</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">WP</td></tr>
<tr><td>INITIAL_VALUE</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><td>0 or 1</td><td>0 or 1</td><td>0 or 1</td></tr>
</table>
<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 colspan="8">SSR[7:0]</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">WP</td></tr>
<tr><td>INITIAL_VALUE</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><td>0 or 1</td><td>0 or 1</td><td>0 or 1</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 24:Security
4.Registers
4.4.Chip Erase Enable Register
(TCFCFG0_CEER)</td><td>
Features of "For details" should be corrected as indicated by the shading below:

(Error)
4.4. Chip Erase Enable Register (TCFCFG0_CEER)
This section describes the function of the Chip Erase Enable Register.

<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">CEER[31:24]</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 or 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><td>0 or 1</td><td>0 or 1</td></tr>
</table>
<table>
<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">CEER[23:16]</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 or 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><td>0 or 1</td><td>0 or 1</td></tr>
</table>
<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 colspan="8">CEER[15:8]</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 or 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><td>0 or 1</td><td>0 or 1</td></tr>
</table>
<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 colspan="8">CEER[7:0]</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 or 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><td>0 or 1</td><td>0 or 1</td></tr>
</table>

(Correct)
4.4. Chip Erase Enable Register (TCFCFG0_CEER)
This section describes the function of the Chip Erase Enable Register.

<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">CEER[31:24]</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">WP</td></tr>
<tr><td>INITIAL_VALUE</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><td>0 or 1</td><td>0 or 1</td><td>0 or 1</td></tr>
</table>
<table>
<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">CEER[23:16]</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">WP</td></tr>
<tr><td>INITIAL_VALUE</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><td>0 or 1</td><td>0 or 1</td><td>0 or 1</td></tr>
</table>
<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 colspan="8">CEER[15:8]</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">WP</td></tr>
<tr><td>INITIAL_VALUE</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><td>0 or 1</td><td>0 or 1</td><td>0 or 1</td></tr>
</table>
<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 colspan="8">CEER[7:0]</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">WP</td></tr>
<tr><td>INITIAL_VALUE</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><td>0 or 1</td><td>0 or 1</td><td>0 or 1</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1359</td><td>CHAPTER 24: Security
4. Registers
4.4.Chip Erase Enable Register (TCFCFG0_CEER)</td><td>Features of "For details" should be corrected as indicated by the shading below
(Error)
[31:0] CEER: Chip Erase Enable Register
Disabling chip erase is not possible by this product. Writing to that register has no effect.

<table><tr><th>Value</th><th>Description</th></tr><tr><td>Any</td><td>No effect</td></tr></table>
(Correct)
[31:0] CEER: Chip Erase Enable Register
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.

The chip erase can be triggered by JTAG equipment or 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.</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1360</td><td>CHAPTER 24:Security
4.Registers
4.5.Security Overwrite Enable Register (TCFCFG0_SOER)</td><td>
Features of "For details" should be corrected as indicated by the shading below:
(Error)
4.5. Security Overwrite Enable Register (TCFCFG0_SOER)
This section describes the function of the Security Overwrite Enable Register.
<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">SOER[31:24]</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 or 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><td>0 or 1</td><td>0 or 1</td></tr>
</table>
<table>
<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">SOER[23:16]</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 or 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><td>0 or 1</td><td>0 or 1</td></tr>
</table>
<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 colspan="8">SOER[15:8]</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 or 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><td>0 or 1</td><td>0 or 1</td></tr>
</table>
<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 colspan="8">SOER[7:0]</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 or 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><td>0 or 1</td><td>0 or 1</td></tr>
</table>
(Correct)
4.5. Security Overwrite Enable Register (TCFCFG0_SOER)
This section describes the function of the Security Overwrite Enable Register.
<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">SOER[31:24]</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">WP</td></tr>
<tr><td>INITIAL_VALUE</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><td>0 or 1</td><td>0 or 1</td><td>0 or 1</td></tr>
</table>
<table>
<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">SOER[23:16]</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">WP</td></tr>
<tr><td>INITIAL_VALUE</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><td>0 or 1</td><td>0 or 1</td><td>0 or 1</td></tr>
</table>
<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 colspan="8">SOER[15:8]</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">WP</td></tr>
<tr><td>INITIAL_VALUE</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><td>0 or 1</td><td>0 or 1</td><td>0 or 1</td></tr>
</table>
<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 colspan="8">SOER[7:0]</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">WP</td></tr>
<tr><td>INITIAL_VALUE</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><td>0 or 1</td><td>0 or 1</td><td>0 or 1</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1362</td><td>CHAPTER 24:Security
4.Registers
4.6.Sector Write Permission Overwrite Enable Register (TCFCFG0_SWPOER)</td><td>
Features of "For details" should be corrected as indicated by the shading below:

(Error)
4.6. Sector Write Permission Overwrite Enable Register (TCFCFG0_SWPOER)
This section describes the function of the Sector Write Permission Overwrite Enable Register.
<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">SWPOER[31:24]</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 or 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><td>0 or 1</td><td>0 or 1</td></tr>
</table>
<table>
<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">SWPOER[23:16]</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 or 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><td>0 or 1</td><td>0 or 1</td></tr>
</table>
<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 colspan="8">SWPOER[15:8]</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 or 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><td>0 or 1</td><td>0 or 1</td></tr>
</table>
<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 colspan="8">SWPOER[7:0]</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 or 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><td>0 or 1</td><td>0 or 1</td></tr>
</table>

(Correct)
4.6. Sector Write Permission Overwrite Enable Register (TCFCFG0_SWPOER)
This section describes the function of the Sector Write Permission Overwrite Enable Register.
<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">SWPOER[31:24]</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">WP</td></tr>
<tr><td>INITIAL_VALUE</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><td>0 or 1</td><td>0 or 1</td><td>0 or 1</td></tr>
</table>
<table>
<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">SWPOER[23:16]</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">WP</td></tr>
<tr><td>INITIAL_VALUE</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><td>0 or 1</td><td>0 or 1</td><td>0 or 1</td></tr>
</table>
<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 colspan="8">SWPOER[15:8]</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">WP</td></tr>
<tr><td>INITIAL_VALUE</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><td>0 or 1</td><td>0 or 1</td><td>0 or 1</td></tr>
</table>
<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 colspan="8">SWPOER[7:0]</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">WP</td></tr>
<tr><td>INITIAL_VALUE</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><td>0 or 1</td><td>0 or 1</td><td>0 or 1</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1364</td><td>CHAPTER 24:Security
4.Registers
4.7.Work Flash Sector Write Permissions Register (TCFCFG0_WSWP)</td><td>
Features of "For details" should be corrected as indicated by the shading below:
(Error)
4.7. Work Flash Sector Write Permissions Register (TCFCFG0_WSWP)
This section describes the function of the Work Flash Sector Write Permission Register.
<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/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 or 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><td>0 or 1</td><td>0 or 1</td></tr>
</table>
<table>
<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/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 or 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><td>0 or 1</td><td>0 or 1</td></tr>
</table>
<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>Reserved</td><td>Reserved</td><td>WSWP13</td><td>WSWP12</td><td>WSWP11</td><td>WSWP10</td><td>WSWP9</td><td>WSWP8</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 or 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><td>0 or 1</td><td>0 or 1</td></tr>
</table>
<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>WSWP7</td><td>WSWP6</td><td>WSWP5</td><td>WSWP4</td><td>WSWP3</td><td>WSWP2</td><td>WSWP1</td><td>WSWP0</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 or 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><td>0 or 1</td><td>0 or 1</td></tr>
</table>

(Correct)
4.7. Work Flash Sector Write Permissions Register (TCFCFG0_WSWP)
This section describes the function of the Work Flash Sector Write Permission Register.
<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/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">WP</td></tr>
<tr><td>INITIAL_VALUE</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><td>0 or 1</td><td>0 or 1</td><td>0 or 1</td></tr>
</table>
<table>
<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/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">WP</td></tr>
<tr><td>INITIAL_VALUE</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><td>0 or 1</td><td>0 or 1</td><td>0 or 1</td></tr>
</table>
<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>Reserved</td><td>Reserved</td><td>WSWP13</td><td>WSWP12</td><td>WSWP11</td><td>WSWP10</td><td>WSWP9</td><td>WSWP8</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">WP</td></tr>
<tr><td>INITIAL_VALUE</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><td>0 or 1</td><td>0 or 1</td><td>0 or 1</td></tr>
</table>
<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>WSWP7</td><td>WSWP6</td><td>WSWP5</td><td>WSWP4</td><td>WSWP3</td><td>WSWP2</td><td>WSWP1</td><td>WSWP0</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">WP</td></tr>
<tr><td>INITIAL_VALUE</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><td>0 or 1</td><td>0 or 1</td><td>0 or 1</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1367</td><td>CHAPTER 24:Security
4.Registers
4.8.Code Flash 0 Sector Write Permissions of the small sectors Register (TCFCFG0_C0SWP)</td><td>
Features of "For details" should be corrected as indicated by the shading below:

(Error)
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.
<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/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 or 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><td>0 or 1</td><td>0 or 1</td></tr>
</table>
<table>
<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/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 or 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><td>0 or 1</td><td>0 or 1</td></tr>
</table>
<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 colspan="8">Reserved</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 or 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><td>0 or 1</td><td>0 or 1</td></tr>
</table>
<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>C0SWP7</td><td>C0SWP6</td><td>C0SWP5</td><td>C0SWP4</td><td>C0SWP3</td><td>C0SWP2</td><td>C0SWP1</td><td>C0SWP0</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 or 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><td>0 or 1</td><td>0 or 1</td></tr>
</table>

(Correct)
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.
<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/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">WP</td></tr>
<tr><td>INITIAL_VALUE</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><td>0 or 1</td><td>0 or 1</td><td>0 or 1</td></tr>
</table>
<table>
<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/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">WP</td></tr>
<tr><td>INITIAL_VALUE</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><td>0 or 1</td><td>0 or 1</td><td>0 or 1</td></tr>
</table>
<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 colspan="8">Reserved</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">WP</td></tr>
<tr><td>INITIAL_VALUE</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><td>0 or 1</td><td>0 or 1</td><td>0 or 1</td></tr>
</table>
<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>C0SWP7</td><td>C0SWP6</td><td>C0SWP5</td><td>C0SWP4</td><td>C0SWP3</td><td>C0SWP2</td><td>C0SWP1</td><td>C0SWP0</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">WP</td></tr>
<tr><td>INITIAL_VALUE</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><td>0 or 1</td><td>0 or 1</td><td>0 or 1</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1369</td><td>CHAPTER 24:Security
4.Registers
4.9.Code Flash 0 Sector Write Permissions of the large sectors Register (TCFCFG0_C1SWP)</td><td>
Features of "For details" should be corrected as indicated by the shading below:

(Error)
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.

<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>C1SWP39</td><td>C1SWP38</td><td>C1SWP37</td><td>C1SWP36</td><td>C1SWP35</td><td>C1SWP34</td><td>C1SWP33</td><td>C1SWP32</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 or 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><td>0 or 1</td><td>0 or 1</td></tr>
</table>
<table>
<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>C1SWP31</td><td>C1SWP30</td><td>C1SWP29</td><td>C1SWP28</td><td>C1SWP27</td><td>C1SWP26</td><td>C1SWP25</td><td>C1SWP24</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 or 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><td>0 or 1</td><td>0 or 1</td></tr>
</table>
<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>C1SWP23</td><td>C1SWP22</td><td>C1SWP21</td><td>C1SWP20</td><td>C1SWP19</td><td>C1SWP18</td><td>C1SWP17</td><td>C1SWP16</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 or 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><td>0 or 1</td><td>0 or 1</td></tr>
</table>
<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>C1SWP15</td><td>C1SWP14</td><td>C1SWP13</td><td>C1SWP12</td><td>C1SWP11</td><td>C1SWP10</td><td>C1SWP9</td><td>C1SWP8</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 or 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><td>0 or 1</td><td>0 or 1</td></tr>
</table>

(Correct)
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.

<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>C1SWP39</td><td>C1SWP38</td><td>C1SWP37</td><td>C1SWP36</td><td>C1SWP35</td><td>C1SWP34</td><td>C1SWP33</td><td>C1SWP32</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">WP</td></tr>
<tr><td>INITIAL_VALUE</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><td>0 or 1</td><td>0 or 1</td><td>0 or 1</td></tr>
</table>
<table>
<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>C1SWP31</td><td>C1SWP30</td><td>C1SWP29</td><td>C1SWP28</td><td>C1SWP27</td><td>C1SWP26</td><td>C1SWP25</td><td>C1SWP24</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">WP</td></tr>
<tr><td>INITIAL_VALUE</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><td>0 or 1</td><td>0 or 1</td><td>0 or 1</td></tr>
</table>
<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>C1SWP23</td><td>C1SWP22</td><td>C1SWP21</td><td>C1SWP20</td><td>C1SWP19</td><td>C1SWP18</td><td>C1SWP17</td><td>C1SWP16</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">WP</td></tr>
<tr><td>INITIAL_VALUE</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><td>0 or 1</td><td>0 or 1</td><td>0 or 1</td></tr>
</table>
<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>C1SWP15</td><td>C1SWP14</td><td>C1SWP13</td><td>C1SWP12</td><td>C1SWP11</td><td>C1SWP10</td><td>C1SWP9</td><td>C1SWP8</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">WP</td></tr>
<tr><td>INITIAL_VALUE</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><td>0 or 1</td><td>0 or 1</td><td>0 or 1</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1375</td><td>CHAPTER 24:Security
4.Registers
4.10.Code Flash 1 Sector Write Permissions of the large sectors Register (TCFCFG0_C2SWP)</td><td>
Features of "For details" should be corrected as indicated by the shading below:

(Error)
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.
<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>C2SWP39</td><td>C2SWP38</td><td>C2SWP37</td><td>C2SWP36</td><td>C2SWP35</td><td>C2SWP34</td><td>C2SWP33</td><td>C2SWP32</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 or 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><td>0 or 1</td><td>0 or 1</td></tr>
</table>
<table>
<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>C2SWP31</td><td>C2SWP30</td><td>C2SWP29</td><td>C2SWP28</td><td>C2SWP27</td><td>C2SWP26</td><td>C2SWP25</td><td>C2SWP24</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 or 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><td>0 or 1</td><td>0 or 1</td></tr>
</table>
<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>C2SWP23</td><td>C2SWP22</td><td>C2SWP21</td><td>C2SWP20</td><td>C2SWP19</td><td>C2SWP18</td><td>C2SWP17</td><td>C2SWP16</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 or 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><td>0 or 1</td><td>0 or 1</td></tr>
</table>
<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>C2SWP15</td><td>C2SWP14</td><td>C2SWP13</td><td>C2SWP12</td><td>C2SWP11</td><td>C2SWP10</td><td>C2SWP9</td><td>C2SWP8</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 or 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><td>0 or 1</td><td>0 or 1</td></tr>
</table>

(Correct)
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.
<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>C2SWP39</td><td>C2SWP38</td><td>C2SWP37</td><td>C2SWP36</td><td>C2SWP35</td><td>C2SWP34</td><td>C2SWP33</td><td>C2SWP32</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">WP</td></tr>
<tr><td>INITIAL_VALUE</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><td>0 or 1</td><td>0 or 1</td><td>0 or 1</td></tr>
</table>
<table>
<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>C2SWP31</td><td>C2SWP30</td><td>C2SWP29</td><td>C2SWP28</td><td>C2SWP27</td><td>C2SWP26</td><td>C2SWP25</td><td>C2SWP24</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">WP</td></tr>
<tr><td>INITIAL_VALUE</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><td>0 or 1</td><td>0 or 1</td><td>0 or 1</td></tr>
</table>
<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>C2SWP23</td><td>C2SWP22</td><td>C2SWP21</td><td>C2SWP20</td><td>C2SWP19</td><td>C2SWP18</td><td>C2SWP17</td><td>C2SWP16</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">WP</td></tr>
<tr><td>INITIAL_VALUE</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><td>0 or 1</td><td>0 or 1</td><td>0 or 1</td></tr>
</table>
<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>C2SWP15</td><td>C2SWP14</td><td>C2SWP13</td><td>C2SWP12</td><td>C2SWP11</td><td>C2SWP10</td><td>C2SWP9</td><td>C2SWP8</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">WP</td></tr>
<tr><td>INITIAL_VALUE</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><td>0 or 1</td><td>0 or 1</td><td>0 or 1</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1380</td><td>CHAPTER 24:Security
4.Registers
4.11.Code Flash 2 Sector Write Permissions of the large sectors Register (TCFCFG0_C3SWP)</td><td>
Features of "For details" should be corrected as indicated by the shading below:
(Error)
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.
<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>C3SWP39</td><td>C3SWP38</td><td>C3SWP37</td><td>C3SWP36</td><td>C3SWP35</td><td>C3SWP34</td><td>C3SWP33</td><td>C3SWP32</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 or 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><td>0 or 1</td><td>0 or 1</td></tr>
</table>
<table>
<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>C3SWP31</td><td>C3SWP30</td><td>C3SWP29</td><td>C3SWP28</td><td>C3SWP27</td><td>C3SWP26</td><td>C3SWP25</td><td>C3SWP24</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 or 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><td>0 or 1</td><td>0 or 1</td></tr>
</table>
<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>C3SWP23</td><td>C3SWP22</td><td>C3SWP21</td><td>C3SWP20</td><td>C3SWP19</td><td>C3SWP18</td><td>C3SWP17</td><td>C3SWP16</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 or 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><td>0 or 1</td><td>0 or 1</td></tr>
</table>
<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>C3SWP15</td><td>C3SWP14</td><td>C3SWP13</td><td>C3SWP12</td><td>C3SWP11</td><td>C3SWP10</td><td>C3SWP9</td><td>C3SWP8</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 or 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><td>0 or 1</td><td>0 or 1</td></tr>
</table>

(Correct)
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.
<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>C3SWP39</td><td>C3SWP38</td><td>C3SWP37</td><td>C3SWP36</td><td>C3SWP35</td><td>C3SWP34</td><td>C3SWP33</td><td>C3SWP32</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">WP</td></tr>
<tr><td>INITIAL_VALUE</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><td>0 or 1</td><td>0 or 1</td><td>0 or 1</td></tr>
</table>
<table>
<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>C3SWP31</td><td>C3SWP30</td><td>C3SWP29</td><td>C3SWP28</td><td>C3SWP27</td><td>C3SWP26</td><td>C3SWP25</td><td>C3SWP24</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">WP</td></tr>
<tr><td>INITIAL_VALUE</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><td>0 or 1</td><td>0 or 1</td><td>0 or 1</td></tr>
</table>
<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>C3SWP23</td><td>C3SWP22</td><td>C3SWP21</td><td>C3SWP20</td><td>C3SWP19</td><td>C3SWP18</td><td>C3SWP17</td><td>C3SWP16</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">WP</td></tr>
<tr><td>INITIAL_VALUE</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><td>0 or 1</td><td>0 or 1</td><td>0 or 1</td></tr>
</table>
<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>C3SWP15</td><td>C3SWP14</td><td>C3SWP13</td><td>C3SWP12</td><td>C3SWP11</td><td>C3SWP10</td><td>C3SWP9</td><td>C3SWP8</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">WP</td></tr>
<tr><td>INITIAL_VALUE</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><td>0 or 1</td><td>0 or 1</td><td>0 or 1</td></tr>
</table>
</td></tr>
</table>

Page	Section	Change Results
1388	CHAPTER 25: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) Secure key storage implemented in the WFLASH (Work Flash, refer to chapter TODO for more information). In this implementation: (Correct) Secure key storage implemented in the WFLASH (Work Flash, refer to chapter WORKFLASH for more information). In this implementation:
1389	CHAPTER 25: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

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1391</td><td>CHAPTER 25:Secure Hardware Extension(SHE)
2.1.Configuration registers for the Command Interface

2.1.1.SHE Command Register (SHE_CMD)</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>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,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>Protection Attribute</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><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,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>Protection Attribute</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><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,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>Protection Attribute</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><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>CMD[7]</td><td>CMD[6]</td><td>CMD[5]</td><td>CMD[4]</td><td>CMD[3]</td><td>CMD[2]</td><td>CMD[1]</td><td>CMD[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 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</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,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>Protection Attribute</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><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,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>Protection Attribute</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><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,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>Protection Attribute</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><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>CMD[7]</td><td>CMD[6]</td><td>CMD[5]</td><td>CMD[4]</td><td>CMD[3]</td><td>CMD[2]</td><td>CMD[1]</td><td>CMD[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 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>

Page	Section	Change Results
1392	CHAPTER 25:Secure Hardware Extension(SHE) 2.Secure Hardware Extension registers 2.1.1.SHE Command Register (SHE_CMD)	Features of “For details” should be added as indicated by the shading below Note: − Initial value depends on the boot configuration.

Page	Section	Change Results
1393	CHAPTER 25:Secure Hardware Extension(SHE) 2.Secure Hardware Extension registers 2.1.2.SHE Command Cancel Register (SHE_CMDCANCEL)	Features of "For details" should be corrected as indicated by the shading below (Error) 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 (Correct) 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>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1395</td><td>CHAPTER 25:Secure Hardware Extension(SHE)
2.Secure Hardware Extension registers
2.1.3.SHE Clock Control Register (SHE_CLKCTRL)</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>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,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>Protection Attribute</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><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>DISREQ</td></tr>
<tr><td>R/W Attribute</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>R,W</td></tr>
<tr><td>Protection Attribute</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><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,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>Protection Attribute</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><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>ENREQ</td></tr>
<tr><td>R/W Attribute</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>R,W</td></tr>
<tr><td>Protection Attribute</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</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,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>Protection Attribute</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><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>DISREQ</td></tr>
<tr><td>R/W Attribute</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>R,W</td></tr>
<tr><td>Protection Attribute</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><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,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>Protection Attribute</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><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>ENREQ</td></tr>
<tr><td>R/W Attribute</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>R,W</td></tr>
<tr><td>Protection Attribute</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>

Page	Section	Change Results
1397	CHAPTER 25:Secure Hardware Extension (SHE) 2.Secure Hardware Extension registers 2.2.1.SHE Status Register (SHE_STATUS)	Features of "For details" should be corrected as indicated by the shading below

(Error)

Bit	31	30	29	28	27	26	25	24
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	FATALERR
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	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

(Correct)

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>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1440</td><td>CHAPTER 25:Secure Hardware Extension (SHE)
2.Secure Hardware Extension registers
2.2.1.SHE Status Register (SHE_STATUS)</td><td>Features of "For details" should be added as indicated by the shading below

Note:
− Initial value depends on the boot configuration.</td></tr>
<tr><td>1401</td><td>CHAPTER 25:Secure Hardware Extension (SHE)
2.Secure Hardware Extension registers
2.2.2.SHE Error Code Register (SHE_ERC)</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>CANCELEXTD[7]</td><td>CANCELEXTD[6]</td><td>CANCELEXTD[5]</td><td>CANCELEXTD[4]</td><td>CANCELEXTD[3]</td><td>CANCELEXTD[2]</td><td>CANCELEXTD[1]</td><td>CANCELEXTD[0]</td></tr><tr><td>R/W Attribute</td><td>R,WX</td><td>R,WX</td><td>R0,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"></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>CANCELEXTD[7]</td><td>CANCELEXTD[6]</td><td>CANCELEXTD[5]</td><td>CANCELEXTD[4]</td><td>CANCELEXTD[3]</td><td>CANCELEXTD[2]</td><td>CANCELEXTD[1]</td><td>CANCELEXTD[0]</td></tr><tr><td>R/W Attribute</td><td>R,WX</td><td>R,WX</td><td>R0,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">-</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>1401</td><td>CHAPTER 25:Secure Hardware Extension (SHE)
2.Secure Hardware Extension registers
2.2.2.SHE Error Code Register (SHE_ERC)</td><td>Features of "For details" should be corrected as indicated by the shading below

(Error)
<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>CANCELMAIN[7]</td><td>CANCELMAIN[6]</td><td>CANCELMAIN[5]</td><td>CANCELMAIN[4]</td><td>CANCELMAIN[3]</td><td>CANCELMAIN[2]</td><td>CANCELMAIN[1]</td><td>CANCELMAIN[0]</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"></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>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>CANCELMAIN[7]</td><td>CANCELMAIN[6]</td><td>CANCELMAIN[5]</td><td>CANCELMAIN[4]</td><td>CANCELMAIN[3]</td><td>CANCELMAIN[2]</td><td>CANCELMAIN[1]</td><td>CANCELMAIN[0]</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">-</td></tr><tr><td>Initial Value</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table></td></tr>
<tr><td>1401</td><td>CHAPTER 25:Secure Hardware Extension (SHE)
2.Secure Hardware Extension registers
2.2.2.SHE Error Code Register (SHE_ERC)</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>CMDEXTD[7]</td><td>CMDEXTD[6]</td><td>CMDEXTD[5]</td><td>CMDEXTD[4]</td><td>CMDEXTD[3]</td><td>CMDEXTD[2]</td><td>CMDEXTD[1]</td><td>CMDEXTD[0]</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"></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>CMDEXTD[7]</td><td>CMDEXTD[6]</td><td>CMDEXTD[5]</td><td>CMDEXTD[4]</td><td>CMDEXTD[3]</td><td>CMDEXTD[2]</td><td>CMDEXTD[1]</td><td>CMDEXTD[0]</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">-</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>1404</td><td>CHAPTER 25:Secure Hardware Extension (SHE)
2.Secure Hardware Extension registers
2.2.3.SHE Clock Status Register (SHE_CLKSTAT)</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>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,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>Protection Attribute</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><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,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>Protection Attribute</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><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,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>Protection Attribute</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><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>CLKOFF</td></tr>
<tr><td>R/W Attribute</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>R,WX</td></tr>
<tr><td>Protection Attribute</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</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,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>Protection Attribute</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><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,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>Protection Attribute</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><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,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>Protection Attribute</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><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>CLKOFF</td></tr>
<tr><td>R/W Attribute</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>R,WX</td></tr>
<tr><td>Protection Attribute</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>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1405</td><td>CHAPTER 25:Secure Hardware Extension (SHE)
2.Secure Hardware Extension registers
2.2.4.SHE Module ID Register (SHE_MID)</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>MID[31]</td><td>MID[30]</td><td>MID[29]</td><td>MID[28]</td><td>MID[27]</td><td>MID[26]</td><td>MID[25]</td><td>MID[24]</td></tr>
<tr><td>R/W Attribute</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>Protection Attribute</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><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>MID[23]</td><td>MID[22]</td><td>MID[21]</td><td>MID[20]</td><td>MID[19]</td><td>MID[18]</td><td>MID[17]</td><td>MID[16]</td></tr>
<tr><td>R/W Attribute</td><td>R0,WX</td><td>R0,WX</td><td>R0,WX</td><td>R0,WX</td><td>R1,WX</td><td>R1,WX</td><td>R1,WX</td><td>R1,WX</td></tr>
<tr><td>Protection Attribute</td><td colspan="8"></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>
<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>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>R/W Attribute</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>R1,WX</td><td>R0,WX</td></tr>
<tr><td>Protection Attribute</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>1</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>MID[7]</td><td>MID[6]</td><td>MID[5]</td><td>MID[4]</td><td>MID[3]</td><td>MID[2]</td><td>MID[1]</td><td>MID[0]</td></tr>
<tr><td>R/W Attribute</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>Protection Attribute</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</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>MID[31]</td><td>MID[30]</td><td>MID[29]</td><td>MID[28]</td><td>MID[27]</td><td>MID[26]</td><td>MID[25]</td><td>MID[24]</td></tr>
<tr><td>R/W Attribute</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>Protection Attribute</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><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>MID[23]</td><td>MID[22]</td><td>MID[21]</td><td>MID[20]</td><td>MID[19]</td><td>MID[18]</td><td>MID[17]</td><td>MID[16]</td></tr>
<tr><td>R/W Attribute</td><td>R0,WX</td><td>R0,WX</td><td>R0,WX</td><td>R0,WX</td><td>R1,WX</td><td>R1,WX</td><td>R1,WX</td><td>R1,WX</td></tr>
<tr><td>Protection Attribute</td><td colspan="8">-</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>
<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>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>R/W Attribute</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>R1,WX</td><td>R0,WX</td></tr>
<tr><td>Protection Attribute</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>1</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>MID[7]</td><td>MID[6]</td><td>MID[5]</td><td>MID[4]</td><td>MID[3]</td><td>MID[2]</td><td>MID[1]</td><td>MID[0]</td></tr>
<tr><td>R/W Attribute</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>Protection Attribute</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>

Page	Section	Change Results
1406	CHAPTER 25:Secure Hardware Extension (SHE) 2.Secure Hardware Extension registers 2.3.1.Interrupt Request Register (SHE_IRQ)	Features of "For details" should be corrected as indicated by the shading below

(Error)

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	FATALER R	IFIFOLOCKER R	OMSTIFSELER R	IMSTIFSELER R	OMSTER R	IMSTER R	OFIFORDER R	IFIFOWRER R
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

(Correct)

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	FATALER R	IFIFOLOCKER R	OMSTIFSELER R	IMSTIFSELER R	OMSTER R	IMSTER R	OFIFORDER R	IFIFOWRER R
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

Page	Section	Change Results
1408	CHAPTER 25:Secure Hardware Extension (SHE) 2.Secure Hardware Extension registers 2.3.1.Interrupt Request Register (SHE_IRQ)	Features of “For details” should be added as indicated by the shading below Note: −Initial value depends on the boot configuration.

Page	Section	Change Results
1409	CHAPTER 25:Secure Hardware Extension (SHE) 2.Secure Hardware Extension registers 2.3.2.Interrupt Request Enable Register (SHE_IRQEN)	Features of "For details" should be corrected as indicated by the shading below

(Error)

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	FATALER R	IFIFOLOCKER R	OMSTIFSELER R	IMSTIFSELER R	OMSTER R	IMSTER R	OFIFORDER R	IFIFOWRER R
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

(Correct)

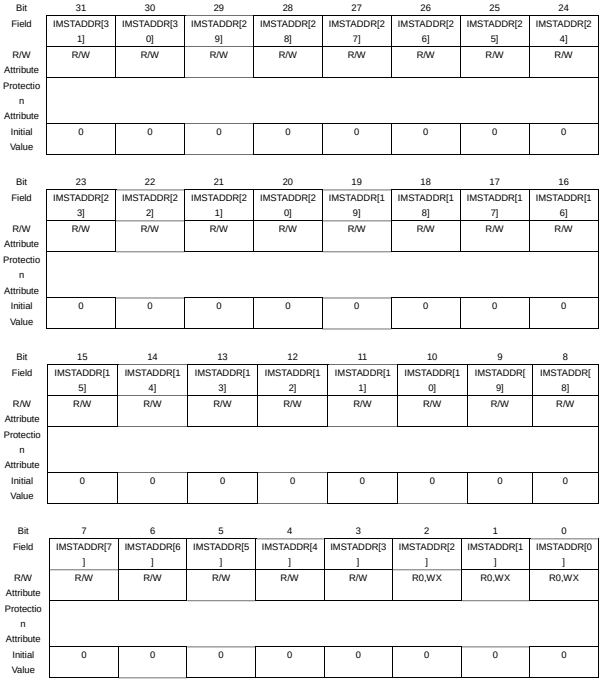
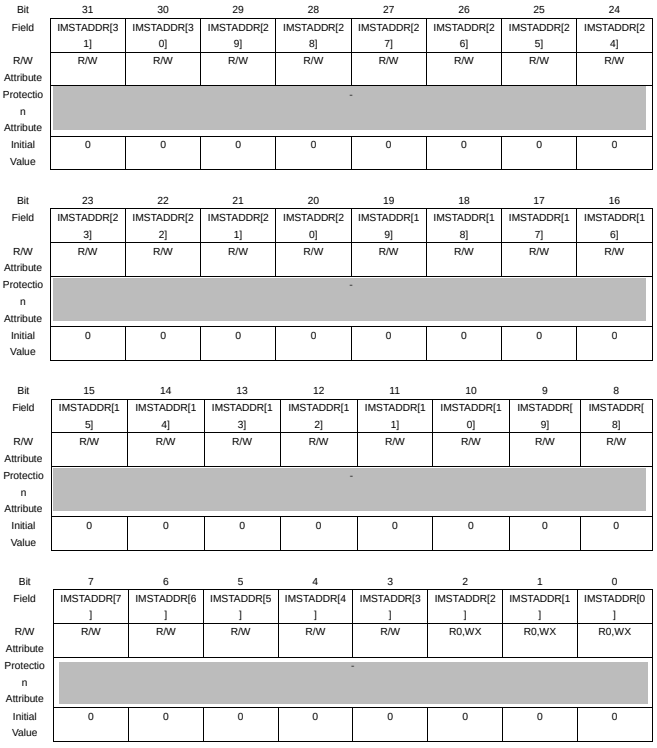
Bit	31	30	29	28	27	26	25	24
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

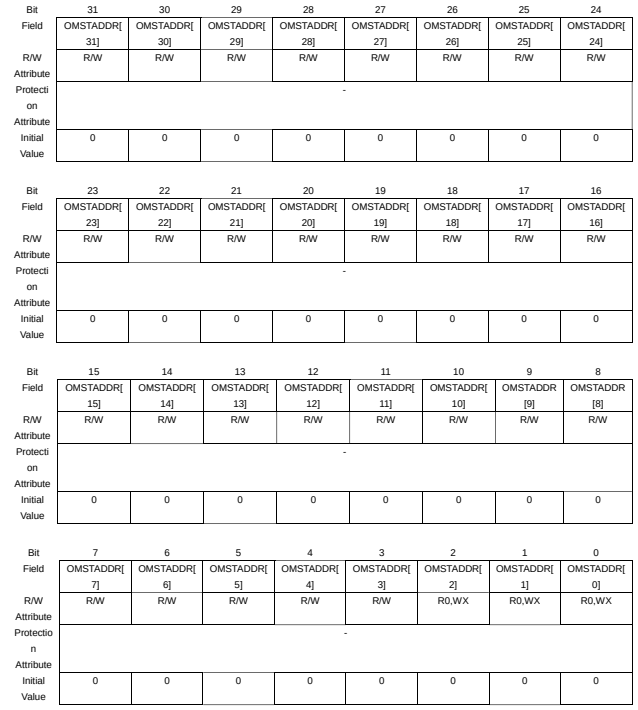
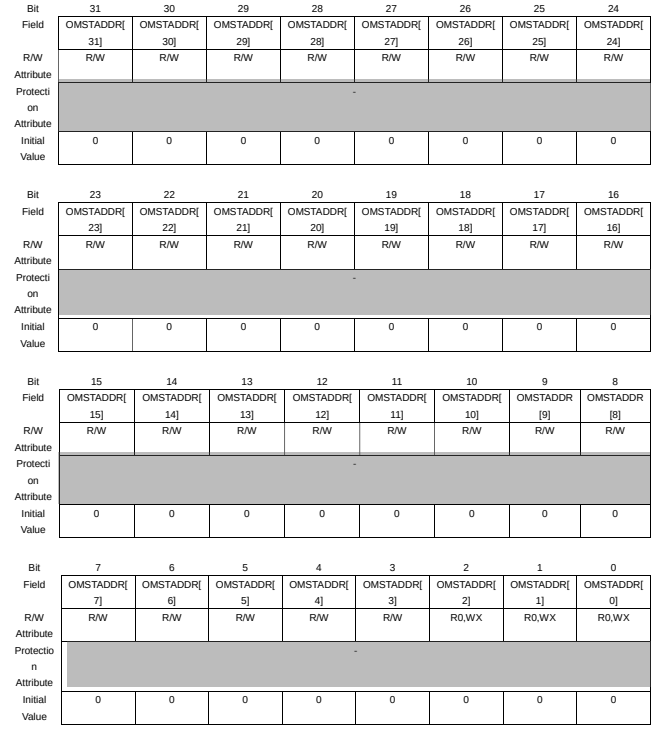
Bit	23	22	21	20	19	18	17	16
Field	FATALER R	IFIFOLOCKER R	OMSTIFSELER R	IMSTIFSELER R	OMSTER R	IMSTER R	OFIFORDER R	IFIFOWRER R
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

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

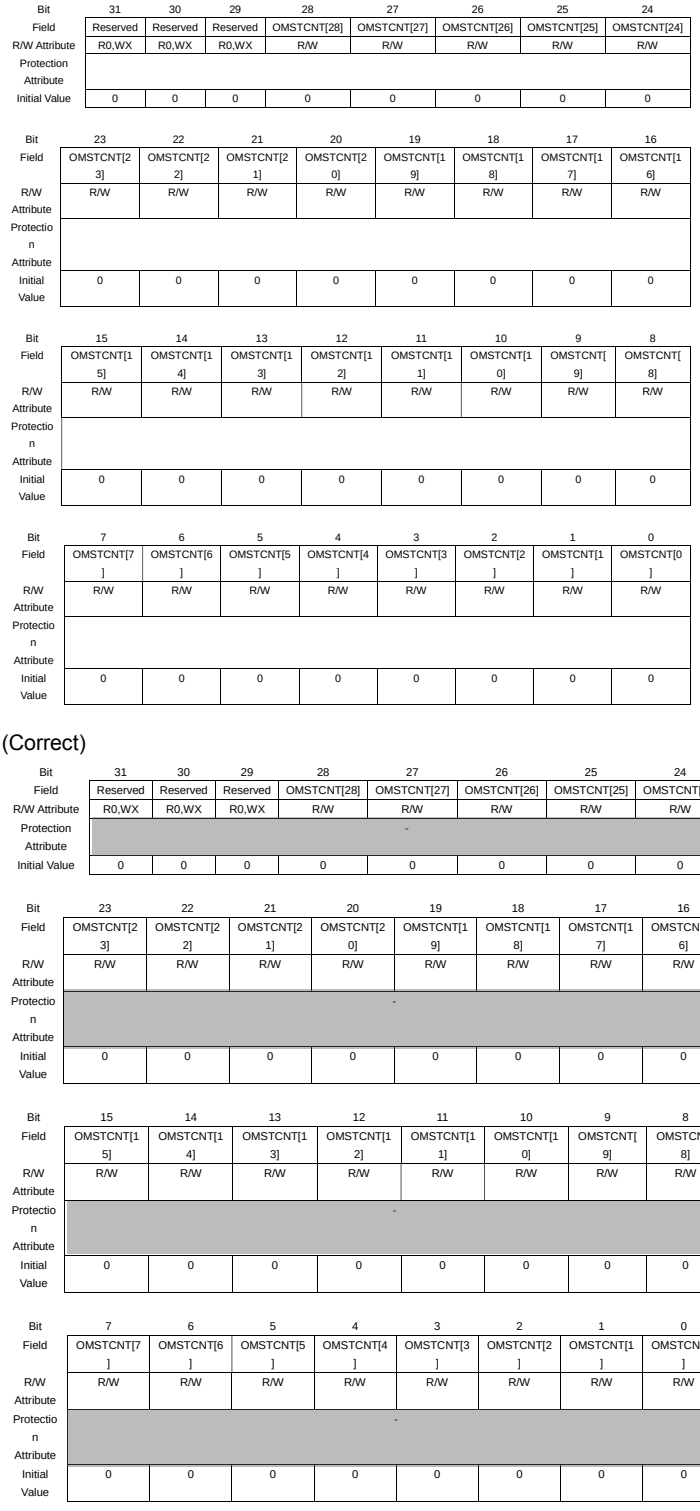
Bit	7	6	5	4	3	2	1	0
Field	Reserved	Reserved	OFIFORDTH	IFIFOWRTH	OMSTIDLE	IMSTIDLE	DONE	COMPAREMATCH
R/W Attribute	R0,WX	R0,WX	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

Page	Section	Change Results
1411	CHAPTER 25:Secure Hardware Extension (SHE) 2.Secure Hardware Extension registers 2.3.3.Interrupt Request Clear Register (SHE_IRQCLR)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit 31 30 29 28 27 26 25 24 Field: Reserved ×8 R/W Attribute: R0,WX ×8 Protection Attribute: Initial Value: 0 ×8 Bit 23 22 21 20 19 18 17 16 Field: FATALERR / IFIFOLOCKERR / OMSTIFSELERR / IMSTIFSELERR / OMSTERR / IMSTERR / OFIFORDERR / IFIFOWRERR R/W Attribute: R0,W ×8 Protection Attribute: Initial Value: 0 ×8 Bit 15 14 13 12 11 10 9 8 Field: Reserved ×8 R/W Attribute: R0,WX ×8 Protection Attribute: Initial Value: 0 ×8 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 ×5 Protection Attribute: Initial Value: 0 ×8 (Correct) Bit 31 30 29 28 27 26 25 24 Field: Reserved ×8 R/W Attribute: R0,WX ×8 Protection Attribute: - Initial Value: 0 ×8 Bit 23 22 21 20 19 18 17 16 Field: FATALERR / IFIFOLOCKERR / OMSTIFSELERR / IMSTIFSELERR / OMSTERR / IMSTERR / OFIFORDERR / IFIFOWRERR R/W Attribute: R0,W ×8 Protection Attribute: - Initial Value: 0 ×8 Bit 15 14 13 12 11 10 9 8 Field: Reserved ×8 R/W Attribute: R0,WX ×8 Protection Attribute: - Initial Value: 0 ×8 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 ×5 Protection Attribute: - Initial Value: 0 ×8

Page	Section	Change Results
1413	CHAPTER 25:Secure Hardware Extension (SHE) 2.Secure Hardware Extension registers 2.4.1.Input Channel Master Start Address Register (SHE_IMSTADDR)	Features of "For details" should be corrected as indicated by the shading below (Error)  (Correct) 

Page	Section	Change Results
1415, 1416	CHAPTER 25:Secure Hardware Extension (SHE) 2.Secure Hardware Extension registers 2.4.2.Output Channel Master Start Address Register (SHE_OMSTADDR)	Features of "For details" should be corrected as indicated by the shading below (Error)  (Correct) 

Page	Section	Change Results
1417, 1418	CHAPTER 25:Secure Hardware Extension (SHE) 2.Secure Hardware Extension registers 2.4.3.Input Channel Master Data Transfer Counter (SHE_IMSTCNT)	Features of "For details" should be corrected as indicated by the shading below (Error) 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 (Correct) 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

Page	Section	Change Results
1419	CHAPTER 25:Secure Hardware Extension (SHE) 2.Secure Hardware Extension registers 2.4.4.Output Channel Master Data Transfer Counter (SHE_OMSTCNT)	Features of "For details" should be corrected as indicated by the shading below (Error)  (Correct)

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1421</td><td>CHAPTER 25:Secure Hardware Extension (SHE)
2.Secure Hardware Extension registers
2.4.5.Input Channel Master Start Trigger (SHE_IMSTSTART)</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>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,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>Protection Attribute</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><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,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>Protection Attribute</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><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,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>Protection Attribute</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><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>IMSTSTART</td></tr>
<tr><td>R/W Attribute</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>R,W</td></tr>
<tr><td>Protection Attribute</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</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,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>Protection Attribute</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><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,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>Protection Attribute</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><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,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>Protection Attribute</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><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>IMSTSTART</td></tr>
<tr><td>R/W Attribute</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>R,W</td></tr>
<tr><td>Protection Attribute</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>

Page	Section	Change Results
1422	CHAPTER 25:Secure Hardware Extension (SHE) 2.Secure Hardware Extension registers 2.4.6.Output Channel Master Start Trigger (SHE_OMSTSTART)	Features of "For details" should be corrected as indicated by the shading below

(Error)

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

(Correct)

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>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1402</td><td>CHAPTER 25:Secure Hardware Extension (SHE)
2.Secure Hardware Extension registers
2.2.2.SHE Error Code Register (SHE_ERC)</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>CMDMAIN[7]</td><td>CMDMAIN[6]</td><td>CMDMAIN[5]</td><td>CMDMAIN[4]</td><td>CMDMAIN[3]</td><td>CMDMAIN[2]</td><td>CMDMAIN[1]</td><td>CMDMAIN[0]</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"></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>CMDMAIN[7]</td><td>CMDMAIN[6]</td><td>CMDMAIN[5]</td><td>CMDMAIN[4]</td><td>CMDMAIN[3]</td><td>CMDMAIN[2]</td><td>CMDMAIN[1]</td><td>CMDMAIN[0]</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">-</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
1423	CHAPTER 25:Secure Hardware Extension (SHE) 2.Secure Hardware Extension registers 2.4.7.Input FIFO Configuration Register (SHE_IFIFOCFG)	Features of "For details" should be corrected as indicated by the shading below

(Error)

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

(Correct)

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>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1425</td><td>CHAPTER 25:Secure Hardware Extension (SHE)
2.Secure Hardware Extension registers
2.4.8.Output FIFO Configuration Register (SHE_OFIFOCFG)</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>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,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>Protection Attribute</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><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>IFSEL</td></tr>
<tr><td>R/W Attribute</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>R/W</td></tr>
<tr><td>Protection Attribute</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>
<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,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>Protection Attribute</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><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>RDTHRESHOLD[5]</td><td>RDTHRESHOLD[4]</td><td>RDTHRESHOLD[3]</td><td>RDTHRESHOLD[2]</td><td>RDTHRESHOLD[1]</td><td>RDTHRESHOLD[0]</td></tr>
<tr><td>R/W Attribute</td><td>R0,WX</td><td>R0,WX</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 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</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,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>Protection Attribute</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><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>IFSEL</td></tr>
<tr><td>R/W Attribute</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>R/W</td></tr>
<tr><td>Protection Attribute</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>
<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,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>Protection Attribute</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><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>RDTHRESHOLD[5]</td><td>RDTHRESHOLD[4]</td><td>RDTHRESHOLD[3]</td><td>RDTHRESHOLD[2]</td><td>RDTHRESHOLD[1]</td><td>RDTHRESHOLD[0]</td></tr>
<tr><td>R/W Attribute</td><td>R0,WX</td><td>R0,WX</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 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>

Page	Section	Change Results
1428	CHAPTER 25:Secure Hardware Extension (SHE) 2.Secure Hardware Extension registers 2.4.9.Input FIFO Compare Value Register (SHE_COMPARE0~1)	Features of "For details" should be corrected as indicated by the shading below (Error)

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

(Correct)

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

Page	Section	Change Results
1430	CHAPTER 25:Secure Hardware Extension (SHE) 2.Secure Hardware Extension registers 2.4.9.Input FIFO Compare Value Register (SHE_COMPARE0~1)	Features of "For details" should be corrected as indicated by the shading below (Error)

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

(Correct)

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>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1432</td><td>CHAPTER 25:Secure Hardware Extension (SHE)
2.Secure Hardware Extension registers
2.5.1.Compare Register Access Status Register (SHE_COMPACC)</td><td>
Features of "For details" should be corrected as indicated by the shading below

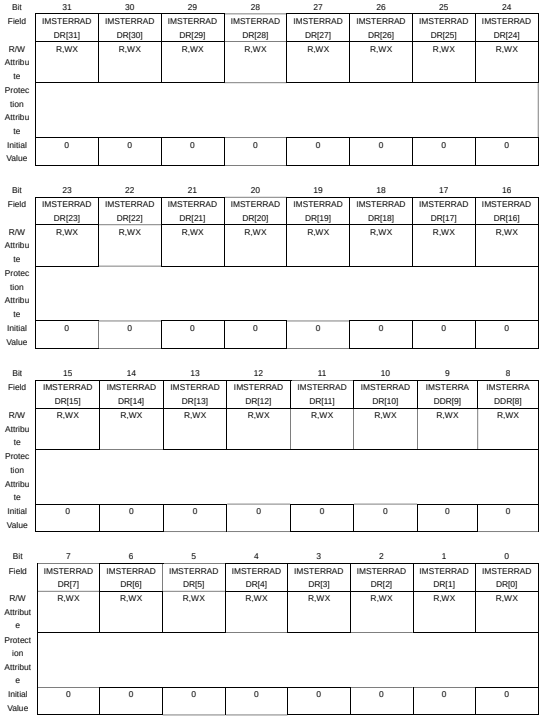
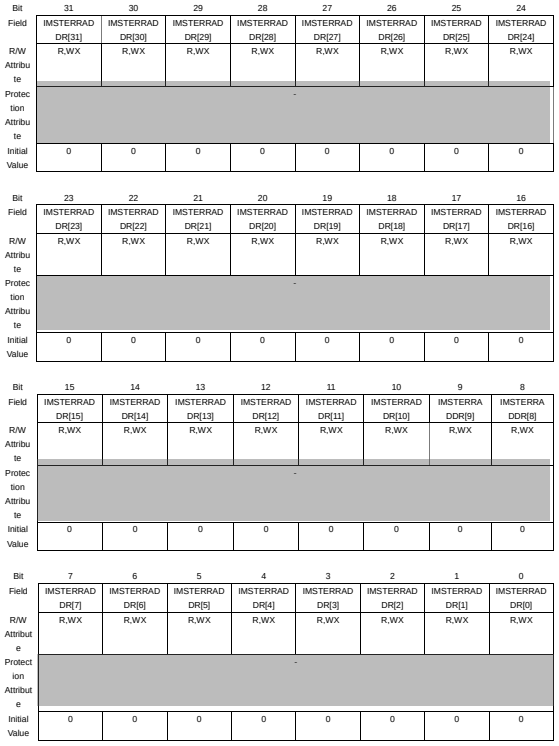
(Error)
<table>
<tr><td>Bit</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td>Field</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>R/W Attribute</td><td>R0,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>Protection Attribute</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><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,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>Protection Attribute</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><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,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>Protection Attribute</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><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>CPUEN</td></tr>
<tr><td>R/W Attribute</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>R,WX</td></tr>
<tr><td>Protection Attribute</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>

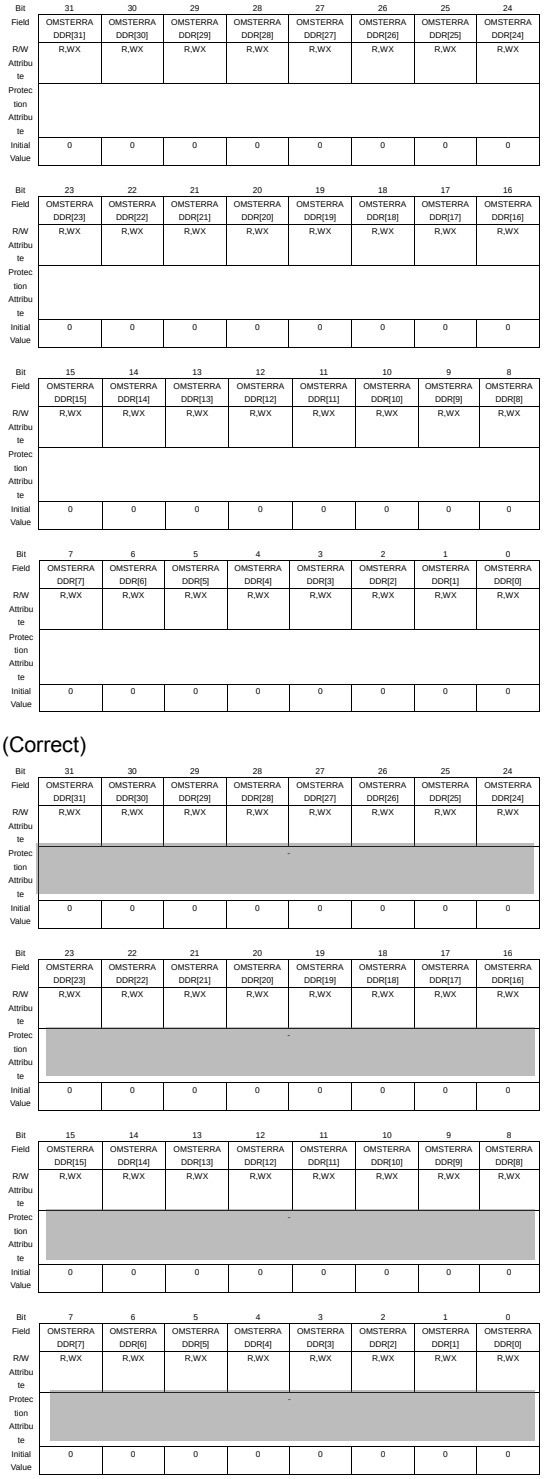
(Correct)
<table>
<tr><td>Bit</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td>Field</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>R/W Attribute</td><td>R0,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>Protection Attribute</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><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,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>Protection Attribute</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><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,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>Protection Attribute</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><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>CPUEN</td></tr>
<tr><td>R/W Attribute</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>R,WX</td></tr>
<tr><td>Protection Attribute</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>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1433</td><td>CHAPTER 25:Secure Hardware Extension (SHE)
2.Secure Hardware Extension registers
2.5.2.Data Master Status Register (SHE_MSTSTATUS)</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>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>OMSTERRRESP[1]</td><td>OMSTERRRESP[0]</td><td>OMSTERR</td></tr>
<tr><td>R/W Attribute</td><td>R0,WX</td><td>R0,WX</td><td>R0,WX</td><td>R0,WX</td><td>R0,WX</td><td>R,WX</td><td>R,WX</td><td>R,WX</td></tr>
<tr><td>Protection Attribute</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><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>OMSTLOCK</td><td>OMSTIDLE</td></tr>
<tr><td>R/W Attribute</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>R,WX</td><td>R,WX</td></tr>
<tr><td>Protection Attribute</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>
<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>IMSTERRRESP[1]</td><td>IMSTERRRESP[0]</td><td>IMSTERR</td></tr>
<tr><td>R/W Attribute</td><td>R0,WX</td><td>R0,WX</td><td>R0,WX</td><td>R0,WX</td><td>R0,WX</td><td>R,WX</td><td>R,WX</td><td>R,WX</td></tr>
<tr><td>Protection Attribute</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><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>IMSTLOCK</td><td>IMSTIDLE</td></tr>
<tr><td>R/W Attribute</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>R,WX</td><td>R,WX</td></tr>
<tr><td>Protection Attribute</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</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>OMSTERRRESP[1]</td><td>OMSTERRRESP[0]</td><td>OMSTERR</td></tr>
<tr><td>R/W Attribute</td><td>R0,WX</td><td>R0,WX</td><td>R0,WX</td><td>R0,WX</td><td>R0,WX</td><td>R,WX</td><td>R,WX</td><td>R,WX</td></tr>
<tr><td>Protection Attribute</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><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>OMSTLOCK</td><td>OMSTIDLE</td></tr>
<tr><td>R/W Attribute</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>R,WX</td><td>R,WX</td></tr>
<tr><td>Protection Attribute</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>
<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>IMSTERRRESP[1]</td><td>IMSTERRRESP[0]</td><td>IMSTERR</td></tr>
<tr><td>R/W Attribute</td><td>R0,WX</td><td>R0,WX</td><td>R0,WX</td><td>R0,WX</td><td>R0,WX</td><td>R,WX</td><td>R,WX</td><td>R,WX</td></tr>
<tr><td>Protection Attribute</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><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>IMSTLOCK</td><td>IMSTIDLE</td></tr>
<tr><td>R/W Attribute</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>R,WX</td><td>R,WX</td></tr>
<tr><td>Protection Attribute</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>

Page	Section	Change Results
1436	CHAPTER 25:Secure Hardware Extension (SHE) 2.Secure Hardware Extension registers 2.5.3.Input Channel Master Error Response Address Register (SHE_IMSTERRADDR)	Features of "For details" should be corrected as indicated by the shading below (Error)  (Correct) 

Page	Section	Change Results
1438	CHAPTER 25:Secure Hardware Extension (SHE) 2.Secure Hardware Extension registers 2.5.4.Output Channel Master Error Response Address Register (SHE_OMSTERRADDR)	Features of "For details" should be corrected as indicated by the shading below (Error)  (Correct)

Page	Section	Change Results
1440	CHAPTER 25:Secure Hardware Extension (SHE) 2.Secure Hardware Extension registers 2.5.5.FIFO Status Register (SHE_FIFOSTATUS)	Features of "For details" should be corrected as indicated by the shading below (Error) 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 (Correct) 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>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1440</td><td>CHAPTER 25:Secure Hardware Extension (SHE)
2.Secure Hardware Extension registers
2.5.5.FIFO Status Register (SHE_FIFOSTATUS)</td><td>Features of "For details" should be corrected as indicated by the shading below
Modify the word in "Table 2 26 FIFO Status Register (SHE_FIFOSTATUS) bits".

(Error)
<table><tr><td>[16]</td><td>COMPAREMATCH</td><td>This bit signalises that the required amount of data has been transferred through the input FIFO and its write port is locked.</td></tr></table>
(Correct)
<table><tr><td>[16]</td><td>COMPAREMATCH</td><td>This bit signalizes that the required amount of data has been transferred through the input FIFO and its write port is locked.</td></tr></table></td></tr>
</table>

Page	Section	Change Results
1442	CHAPTER 25:Secure Hardware Extension (SHE) 2.Secure Hardware Extension registers 2.5.6.FIFO Load Register (SHE_FIFOLOAD)	Features of "For details" should be corrected as indicated by the shading below (Error)

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

(Correct)

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

Page	Section	Change Results
1444	CHAPTER 25:Secure Hardware Extension (SHE) 2.Secure Hardware Extension registers 2.5.7.Input FIFO Data Counter Register (SHE_DATACNT0~1)	Features of "For details" should be corrected as indicated by the shading below (Error) (Correct)

(Error)

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

(Correct)

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

Page	Section	Change Results
1446	CHAPTER 25:Secure Hardware Extension (SHE) 2.Secure Hardware Extension registers 2.5.7.Input FIFO Data Counter Register (SHE_DATACNT0~1)	Features of "For details" should be corrected as indicated by the shading below (Error)

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

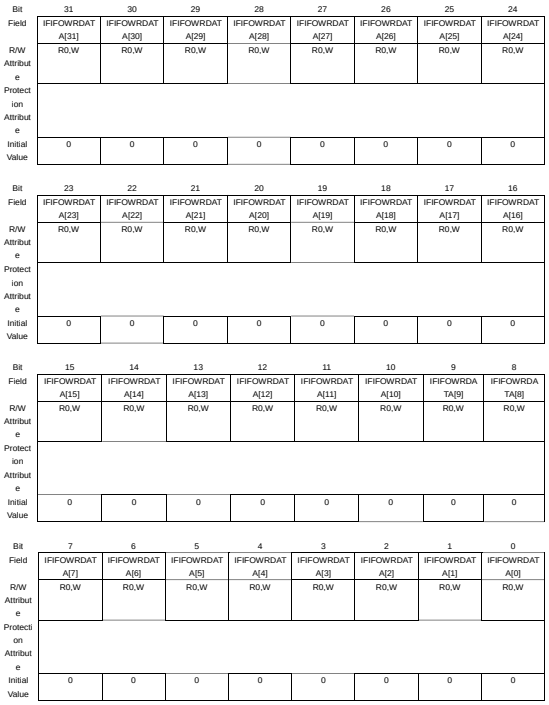
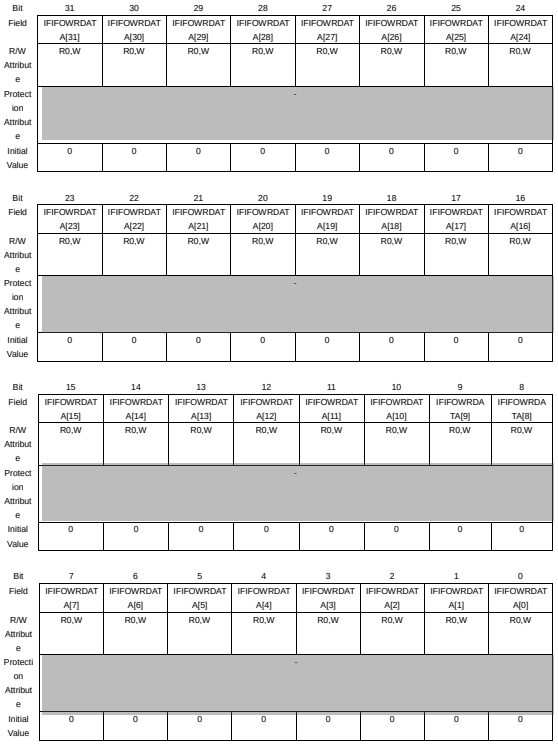
(Correct)

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

Page	Section	Change Results
1447, 1448	CHAPTER 25:Secure Hardware Extension (SHE) 2.Secure Hardware Extension registers 2.6.1.Input FIFO Write Data Register (SHE_IFIFOWRDATA 0~31)	Features of "For details" should be corrected as indicated by the shading below (Error)  (Correct) 

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

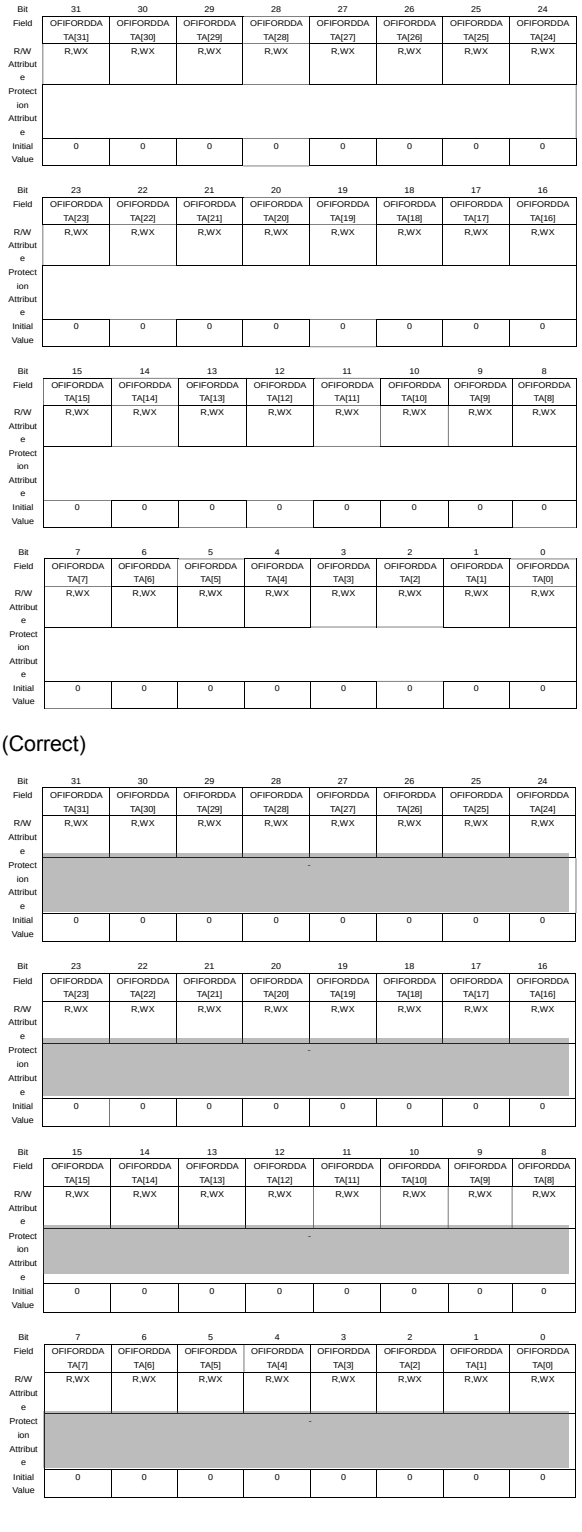
Bit	7	6	5	4	3	2	1	0
Field	IFIFOWRDATA[7]	IFIFOWRDATA[6]	IFIFOWRDATA[5]	IFIFOWRDATA[4]	IFIFOWRDATA[3]	IFIFOWRDATA[2]	IFIFOWRDATA[1]	IFIFOWRDATA[0]
R/W Attribute	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
Protection Attribute								
Initial Value	0	0	0	0	0	0	0	0

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

Page	Section	Change Results
1449, 1450	CHAPTER 25:Secure Hardware Extension (SHE) 2.Secure Hardware Extension registers 2.6.2.Output FIFO Read Data Register (SHE_OFIFORDDATA0~31)	Features of "For details" should be corrected as indicated by the shading below (Error)  (Correct)

Page	Section	Change Results
1452	CHAPTER 25: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) The processing of a sequential SHE command is presented in Table 3-1 and is described in the following. (Correct) The processing of a sequential SHE command is presented in Figure 1-1 and is described in the following.
1452	CHAPTER 25: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 3-1 Flow chart for the execution of a sequential SHE command (Correct) Figure 3-1 Flow chart for the execution of a sequential SHE command
1456	CHAPTER 25: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
1458	CHAPTER 25: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 Execution of CMD_CANCEL Command (Correct) Figure 3-3 Execution of CMD_CANCEL Command
1459	CHAPTER 25: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

Page	Section	Change Results
1460	CHAPTER 25: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 issued after the previous command has completed (Correct) Figure 3-5 Execution of CMD_CANCEL command when no command execution is ongoing.
1461	CHAPTER 25: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.
1462	CHAPTER 25:Secure Hardware Extension(SHE) 3.Operation of the Secure Hardware Extension 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
1462	CHAPTER 25:Secure Hardware Extension(SHE) 3.Operation of the Secure Hardware Extension 3.1.1.Power saving functionality	Features of "For details" should be added as indicated by the shading below (Error) System Wide Clock/Power Gating This means SHE is notified if any clock is gated for at least one of the following modules (the appropriate clock name is given in parentheses): (Correct) System Wide Clock/Power Gating This means for a Traveo 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):
1464	CHAPTER 25:Secure Hardware Extension(SHE) 3.1.2.Command interface access issues	Features of "For details" should be deleted as indicated by the shading below Access to the SHE command Interface is protected through a PPU. In other words reading and writing in user mode are only possible if the PPU is appropriately configured. Privileged accesses are always allowed.

Page	Section	Change Results
1465	CHAPTER 25:Secure Hardware Extension(SHE) 3.2.Operation of the data interface	Features of "For details" should be corrected as indicated by the shading below (Error) Table 3-10 SHE data flow diagram (Correct) Figure 3-8 SHE data flow diagram
1468	CHAPTER 25:Secure Hardware Extension(SHE) 3.2.Operation of the data interface	Features of "For details" should be corrected as indicated by the shading below (Error) Error Handling 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 cancelled. (Correct) Error Handling 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.
1468	CHAPTER 25:Secure Hardware Extension(SHE) 3.2.Operation of the data interface	Features of "For details" should be added as indicated by the shading below (Error) **Memory protection** For safety reasons both masters have a Memory Protection Unit (MPU) in order to protect the memory areas of the MCU. (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 26 (Memory Protection Unit for the AXI) of the Traveo Cluster Series Hardware Manual.
1470	CHAPTER 25:Secure Hardware Extension(SHE) 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

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1471</td><td>CHAPTER 25:Secure Hardware Extension(SHE)
3.3.Operation of the FIFO unit</td><td>Features of "For details" should be corrected as indicated by the shading below

(Error)
Data ordering for the command parameters
1. 64-bit wide parameters must be represented in the native little endian byte order of host system and, as described in the Section 3.1, the least significant part of the parameter has to be transferedtransferred first.

(Correct)
Data ordering for the command parameters
1. 64-bit wide parameters must be represented in the native little endian byte order of Traveo host system and, as described in the Section 3.1, the least significant part of the parameter has to be transferred first.</td></tr>
<tr><td>1472</td><td>CHAPTER 25:Secure Hardware Extension(SHE)
3.3.Operation of the FIFO unit</td><td>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</td></tr>
<tr><td>1476</td><td>CHAPTER 25:Secure Hardware Extension(SHE)
4.1.1.Notes on using SHE commands</td><td>Features of "For details" should be corrected as indicated by the shading below

(Error)
Secure Boot Settings
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' =ciel (size from BDR' / 16) * 2.

(Correct)
Secure Boot Settings
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.</td></tr>
<tr><td>1477</td><td>CHAPTER 25:Secure Hardware Extension(SHE)
4.1.2.SHE Used Error Codes/Used Cancel Codes</td><td>Features of "For details" should be corrected as indicated by the shading below

(Error)
Table 4-1 SHE Used Error Codes
<table><tr><td>0x0010</td><td>0x010D</td><td>ERC_GENERAL_ERROR</td><td>AES engine could not be stopped after a proceeding error.</td></tr></table>
(Correct)
Table 4-1 SHE Used Error Codes
<table><tr><td>0x0010</td><td>0x010D</td><td>ERC_GENERAL_ERROR</td><td>AES engine could not be stopped after a preceding error.</td></tr></table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1480</td><td>CHAPTER 25:Secure Hardware Extension(SHE)
4.1.2.SHE Used Error Codes/Used Cancel Codes</td><td>Features of "For details" should be corrected as indicated by the shading below

(Error)
Table 4-1 SHE Used Error Codes
<table><tr><td>0x0010</td><td>0x240D</td><td>ERC_GENERAL_ERROR</td><td>TRNG returned a "Von Neuman" error.</td></tr></table>
(Correct)
Table 4-1 SHE Used Error Codes
<table><tr><td>0x0010</td><td>0x240D</td><td>ERC_GENERAL_ERROR</td><td>TRNG returned a "Von Neumann" error.</td></tr></table></td></tr>
<tr><td>1487</td><td>CHAPTER 25:Secure Hardware Extension(SHE)
6. References</td><td>Features of "For details" should be added as indicated by the shading below

(Error)
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

(Correct)
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</td></tr>
<tr><td>1489
to
1514</td><td>CHAPTER 26: Memory Protection Unit for AXI of SHE</td><td>Added new Chapter named “Memory Protection Unit for AXI of SHE”.</td></tr>
<tr><td>1519</td><td>CHAPTER 27: DMA Complex Subsystem
3.Operation of the DMA COMPLEX SUBSYSTEM</td><td>Features of "For details" should be added as indicated by the shading below

(Correct)
When stopping DMA_CLK and accessing a register, an error reply is returned.</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1533</td><td>CHAPTER 27: DMA Complex Subsystem 4.6. DMA Additional Control CMCHIC Reload Data Bank Register i (DMAAn_CMCHICRDBi) (i=0~7)</td><td>Features of "For details" should be added 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>CI7</td><td>CI6</td><td>CI5</td><td>CI4</td><td>CI3</td><td>CI2</td><td>CI1</td><td>CI0</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"></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>CI7</td><td>CI6</td><td>CI5</td><td>CI4</td><td>CI3</td><td>CI2</td><td>CI1</td><td>CI0</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">-</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>1549</td><td>CHAPTER 27: 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)
[bit18] TRG: Trigger
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' to load the reload register content before counting starts.

(Correct)
[bit18] TRG: Trigger
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.</td></tr>
<tr><td>1553</td><td>CHAPTER 27: DMA Complex Subsystem 8.3.32-bit Reload Register (DMAAn_RLTm_TMRLR)</td><td>Features of "For details" should be added as indicated by the shading below

Note:
– This register is not initialized by Hard Reset (i.e. "X").</td></tr>
</table>

Page	Section	Change Results
1554	CHAPTER 27: DMA Complex Subsystem 8.4.32-bit Timer Register (DMAAn_RLTm_TMR)	Features of "For details" should be added as indicated by the shading below Note: – This register is not initialized by Hard Reset (i.e. "X").
1558	CHAPTER 28: DMA Controller 2. Configuration	Features of "For details" should be corrected as indicated by the shading below (Error) (Note) – Please refer "CHAPTER:Overview" for the maximum numbers. (Correct) Note: – Please refer ""APPENDIX:DMA channel Activation Factors"" for the maximum numbers.
1582	CHAPTER 28: DMA Controller 4.Registers	Features of "For details" should be corrected as indicated by the shading below (Error) Note: – Please refer "CHAPTER:Overview" for the maximum numbers. (Correct) Notes: – Please refer "CHAPTER: Overview" for the maximum numbers. – *1: Initial value of DMAi_CMCHICn:CI[7:0] is 'n'.
1593	CHAPTER 28: DMA Controller 4.Registers	Features of "For details" should be corrected as indicated by the shading below (Error) 01 4-beat incrementing burst (INCR4) 10 8-bit incrementing burst (INCR8) 11 16-bit incrementing burst (INCR16) (Correct) 01 4-beat incrementing burst (INCR4) 10 8-beat incrementing burst (INCR8) 11 16-beat incrementing burst (INCR16)

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1594</td><td>CHAPTER 28: 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)
<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>DQ</td><td>EQ</td><td>MS[1]</td><td>MS[0]</td><td>TW[1]</td><td>TW[0]</td><td>SR</td><td>Reserved</td></tr><tr><td>Attribute</td><td>R,WX</td><td>R,WX</td><td>R0,W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R,WX</td><td>R,WX</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>0</td><td>0</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>DQ</td><td>EQ</td><td>MS[1]</td><td>MS[0]</td><td>TW[1]</td><td>TW[0]</td><td>SR</td><td>Reserved</td></tr><tr><td>Attribute</td><td>R,WX</td><td>R,WX</td><td>R0,W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R,WX</td><td>R0,WX</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>0</td><td>0</td><td>0</td><td>0</td></tr></table></td></tr>
<tr><td>1611</td><td>CHAPTER 28: 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)
This register controls internal client interface 'm'.

(Correct)
This register controls internal channel interface 'n'.</td></tr>
<tr><td>1611</td><td>CHAPTER 28:DMA Controller
4.17. DMA Controller Client Matrix Channel Interface Configuration Register 'n' (DMAi_CMCHICn)</td><td>The attributes should be corrected as indicated by the shading below.

(Correct)
<table><tr><td>bit</td><td>bit 23</td><td>bit 22</td><td>bit 21</td><td>bit 20</td><td>bit 19</td><td>bit 18</td><td>bit 17</td><td>bit 16</td></tr><tr><td>Field</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>RLSLOT[2]</td><td>RLSLOT[1]</td><td>RLSLOT[0]</td></tr><tr><td>Attribute</td><td>R0,WX</td><td>R0,WX</td><td>R0,WX</td><td>R0,WX</td><td>R0,WX</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>0</td><td>0</td><td>0</td><td>0</td></tr><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>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>Attribute</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>R,W0</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>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>1619</td><td>CHAPTER 29: MPU16 AHB
2.MPU16 AHB registers</td><td>Features of "For details" should be added as indicated by the shading below
(Correct)
Figure 2- 1 List of register mirror area
<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_0400 to 0000_07FF
0000_0800 to 0000_0BFF
0000_0C00 to 0000_0FFF</td><td>MPU for DMAC#0</td><td>Accessing this region results to the access to MPU for DMAC#0 register area with following offset.
Offset: (addr & 0000_03FF)</td><td>This area is covered by PPU #66 as well as the mirrored peripheral</td></tr>
<tr><td>0000_1400 to 0000_17FF
0000_1800 to 0000_1BFF
0000_1C00 to 0000_1FFF</td><td>MPU for DMAC#1</td><td>Accessing this region results to the access to MPU for DMAC#1 register area with following offset.
Offset: (addr & 0000_03FF)</td><td>This area is covered by PPU #67 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>1623</td><td>CHAPTER 29: MPU16 AHB
2.3.MPU16 AHB Memory Error Control Register (MPUHn_MERRC)</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>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,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>Protection Attribute</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><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,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>Protection Attribute</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><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,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>Protection Attribute</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><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>HPROT</td><td>HWRITE</td></tr>
<tr><td>R/W Attribute</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>R,WX</td><td>R,WX</td></tr>
<tr><td>Protection Attribute</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</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,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>Protection Attribute</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><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,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>Protection Attribute</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><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,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>Protection Attribute</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><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>HPROT</td><td>HWRITE</td></tr>
<tr><td>R/W Attribute</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>R,WX</td><td>R,WX</td></tr>
<tr><td>Protection Attribute</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>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1624</td><td>CHAPTER 29: MPU16 AHB
2.MPU16 AHB registers
2.4.MPU AHB Memory Error Address Register (MPUHn_MERRA)</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>HADDR[31]</td><td>HADDR[30]</td><td>HADDR[29]</td><td>HADDR[28]</td><td>HADDR[27]</td><td>HADDR[26]</td><td>HADDR[25]</td><td>HADDR[24]</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"></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>HADDR[23]</td><td>HADDR[22]</td><td>HADDR[21]</td><td>HADDR[20]</td><td>HADDR[19]</td><td>HADDR[18]</td><td>HADDR[17]</td><td>HADDR[16]</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"></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>HADDR[15]</td><td>HADDR[14]</td><td>HADDR[13]</td><td>HADDR[12]</td><td>HADDR[11]</td><td>HADDR[10]</td><td>HADDR[9]</td><td>HADDR[8]</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"></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>HADDR[7]</td><td>HADDR[6]</td><td>HADDR[5]</td><td>HADDR[4]</td><td>HADDR[3]</td><td>HADDR[2]</td><td>HADDR[1]</td><td>HADDR[0]</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"></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>HADDR[31]</td><td>HADDR[30]</td><td>HADDR[29]</td><td>HADDR[28]</td><td>HADDR[27]</td><td>HADDR[26]</td><td>HADDR[25]</td><td>HADDR[24]</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">-</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>HADDR[23]</td><td>HADDR[22]</td><td>HADDR[21]</td><td>HADDR[20]</td><td>HADDR[19]</td><td>HADDR[18]</td><td>HADDR[17]</td><td>HADDR[16]</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">-</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>HADDR[15]</td><td>HADDR[14]</td><td>HADDR[13]</td><td>HADDR[12]</td><td>HADDR[11]</td><td>HADDR[10]</td><td>HADDR[9]</td><td>HADDR[8]</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">-</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>HADDR[7]</td><td>HADDR[6]</td><td>HADDR[5]</td><td>HADDR[4]</td><td>HADDR[3]</td><td>HADDR[2]</td><td>HADDR[1]</td><td>HADDR[0]</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">-</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>1627</td><td>CHAPTER 29: MPU16 AHB
2.MPU16 AHB registers
2.6.MPU16 AHB Start Address Registers (MPUHn_SADDR1~16)</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>SADDR[7]</td><td>SADDR[6]</td><td>SADDR[5]</td><td>SADDR[4]</td><td>SADDR[3]</td><td>SADDR[2]</td><td>SADDR[1]</td><td>SADDR[0]</td></tr>
<tr><td>R/W Attribute</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>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>0</td><td>0</td><td>0</td></tr>
</table>

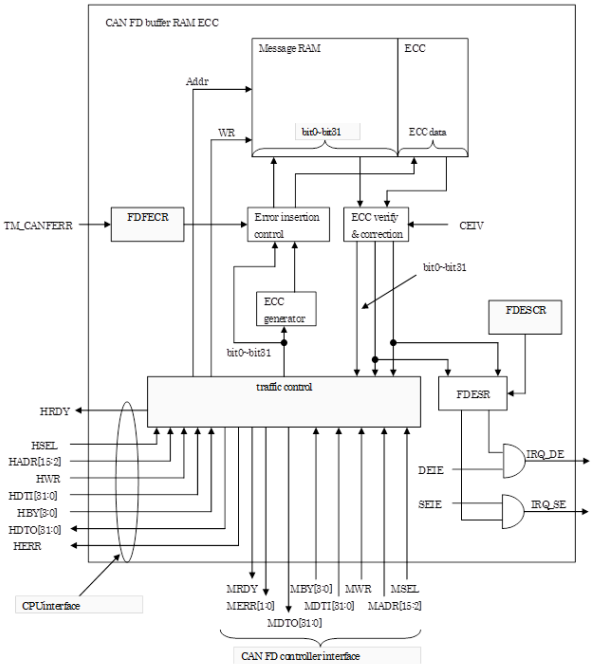
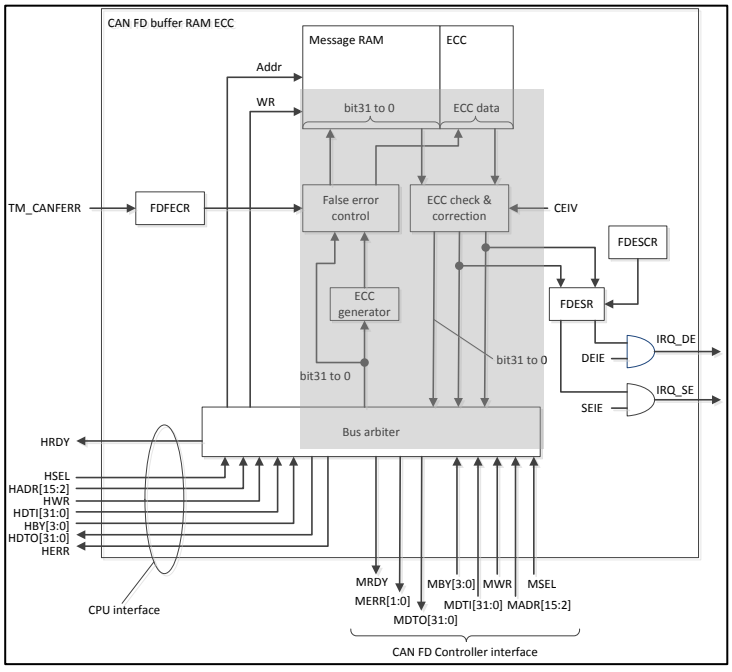
(Correct)
<table>
<tr><td>Bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Field</td><td>SADDR[7]</td><td>SADDR[6]</td><td>SADDR[5]</td><td>SADDR[4]</td><td>SADDR[3]</td><td>SADDR[2]</td><td>SADDR[1]</td><td>SADDR[0]</td></tr>
<tr><td>R/W Attribute</td><td>R/W</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>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>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>1631</td><td>CHAPTER 29: MPU16 AHB 2.MPU16 AHB registers 2.9.MPU16 AHB Module ID Register (MPUHn_MID)</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>MID[31]</td><td>MID[30]</td><td>MID[29]</td><td>MID[28]</td><td>MID[27]</td><td>MID[26]</td><td>MID[25]</td><td>MID[24]</td></tr>
<tr><td>R/W Attribute</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>Protection Attribute</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><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>MID[23]</td><td>MID[22]</td><td>MID[21]</td><td>MID[20]</td><td>MID[19]</td><td>MID[18]</td><td>MID[17]</td><td>MID[16]</td></tr>
<tr><td>R/W Attribute</td><td>R0,WX</td><td>R0,WX</td><td>R0,WX</td><td>R0,WX</td><td>R1,WX</td><td>R1,WX</td><td>R0,WX</td><td>R1,WX</td></tr>
<tr><td>Protection Attribute</td><td colspan="8"></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>0</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>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>R/W Attribute</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>Protection Attribute</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><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>MID[7]</td><td>MID[6]</td><td>MID[5]</td><td>MID[4]</td><td>MID[3]</td><td>MID[2]</td><td>MID[1]</td><td>MID[0]</td></tr>
<tr><td>R/W Attribute</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>Protection Attribute</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</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>MID[31]</td><td>MID[30]</td><td>MID[29]</td><td>MID[28]</td><td>MID[27]</td><td>MID[26]</td><td>MID[25]</td><td>MID[24]</td></tr>
<tr><td>R/W Attribute</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>Protection Attribute</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><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>MID[23]</td><td>MID[22]</td><td>MID[21]</td><td>MID[20]</td><td>MID[19]</td><td>MID[18]</td><td>MID[17]</td><td>MID[16]</td></tr>
<tr><td>R/W Attribute</td><td>R0,WX</td><td>R0,WX</td><td>R0,WX</td><td>R0,WX</td><td>R1,WX</td><td>R1,WX</td><td>R0,WX</td><td>R1,WX</td></tr>
<tr><td>Protection Attribute</td><td colspan="8">-</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>0</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>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>R/W Attribute</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>Protection Attribute</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><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>MID[7]</td><td>MID[6]</td><td>MID[5]</td><td>MID[4]</td><td>MID[3]</td><td>MID[2]</td><td>MID[1]</td><td>MID[0]</td></tr>
<tr><td>R/W Attribute</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>Protection Attribute</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>

Page	Section	Change Results
1637	CHAPTER 30: CAN FD Message RAM ECC Function Cover	Comment should be deleted as indicated by the shading below. (Error) This specification describes only the block of the CAN FD message RAM ECC function. For the external specifications of the CAN FD controller, see "Chapter: CAN FD Controller". * Furthermore, "Chapter: CAN FD Controller" describes TSS[1:0] register setting as TSS[1:0] : “10” = External timestamp counter value used , but this setting is forbidden for this product. If set, operation will be as same as “TSS[1:0] : “00”= Timestamp counter value always 0x0000”.
1638	CHAPTER 30: CAN FD Message RAM ECC Function 1.Overview	"1.Overview" should be corrected as indicated by the shading below. (Error) 1. Overview 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) 1.Overview 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.

Page	Section	Change Results
1639	CHAPTER 30: CAN FD Message RAM ECC Function 2.Configuration	"Figure 2-1" should be corrected as indicated by the shading below. (Error)  (Correct) 

Page	Section	Change Results
1640	CHAPTER 30: CAN FD Message RAM ECC Function 2.Configuration	"2. Configuration" should be corrected as indicated by the shading below. (Correct) 2. Configuration Figure 2-1 is a 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
1641	CHAPTER 30: CAN FD Message RAM ECC Function 3. Interrupts	"3. Interrupts" should be corrected as indicated by the shading below. (Error) 3. Interrupts Interrupts of the CAN FD message RAM ECC function The message RAM ECC supports the following interrupts. (Correct) 3. Interrupts Interrupts of the CAN FD Message RAM ECC Function The Message RAM ECC Function supports the following interrupts.

Page	Section	Change Results
1642	CHAPTER 30: CAN FD Message RAM ECC Function 4. Explanation of Operation	"4. Explanation of Operation" should be corrected as indicated by the shading below. (Error) 4. Explanation of Operation This section explains the operation and function of the CAN FD message RAM ECC. This section explains the following functions. ■ Message RAM ECC generation ■ Message RAM ECC test ■ ECC error insertion function of the message RAM (Correct) 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
1643	CHAPTER 30: CAN FD Message RAM ECC Function 4. Explanation of Operation 4.1. Message RAM ECC Generation	"4.1. Message RAM ECC Generation" should be corrected as indicated by the shading below. (Error) 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. (Correct) 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.

Page	Section	Change Results
1644	CHAPTER 30: CAN FD Message RAM ECC Function 4. Explanation of Operation 4.2. Message RAM ECC Check	"4.2. Message RAM ECC Test" should be corrected as indicated by the shading below. (Error) 4.2. Message RAM ECC Test This section explains the ECC test of the message RAM. ECC test When data is read from the message RAM, the ECC data is also read. In the ECC test, calculation is performed based on the syndrome to determine whether the data in the message RAM is correct. If it is determined from the test result that the data in the message RAM is incorrect due to a RAM defect or another problem, 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", "1" is set in the error bit and an interrupt occurs at this time. **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. 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. (Correct) 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.

Page	Section	Change Results
1648	CHAPTER 30: CAN FD Message RAM ECC Function 6. Registers 6.1.CAN FD ECC Error Control Register (FDECR)	Features of "For details" should be added as indicated by the shading below. (Correct) [bit7:4] Reserved: Reserved bits When writing, always write "0". When reading, "0" is always read.

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1648</td><td>CHAPTER 30: CAN FD Message RAM ECC Function
6. Registers
6.1.CAN FD ECC Error Control Register (FDECR)</td><td>"6.1. CAN FD ECC Error Control Register (FDECR)" should be corrected as indicated by the shading below.

(Error)

[bit3] CEIV: ECC test disable bit
<table>
<tr><th>CEIV</th><th>Function</th></tr>
<tr><td>0</td><td>Enable ECC test. [Initial value]</td></tr>
<tr><td>1</td><td>Disable ECC test.</td></tr>
</table>
■ This bit can be changed when "1" is set in the CCE bit in the CC control register (CCCR) of the CAN FD controller.
■ When CEIV is "0", the ECC test is enabled.
■ When CEIV is "1", ECC test is disabled.

[bit2] CEREN: ECC error response enable bit
<table>
<tr><th>CEREN</th><th>Function</th></tr>
<tr><td>0</td><td>Disable error response for double-bit error detection. [Initial value]</td></tr>
<tr><td>1</td><td>Enable error response for double-bit error detection.</td></tr>
</table>
■ This bit can be changed when "1" is set in the CCE bit in the CC control register (CCCR) of the CAN FD controller.
■ When CEREN 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 access.
■ When CEREN 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 access.

(Correct)

[bit3] CEIV: ECC check disable bit
<table>
<tr><th>bit</th><th>Description</th></tr>
<tr><td>0</td><td>Enable ECC check. [Initial value]</td></tr>
<tr><td>1</td><td>Disable ECC check.</td></tr>
</table>
■ 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
<table>
<tr><th>bit</th><th>Description</th></tr>
<tr><td>0</td><td>Disable error response for double-bit error detection. [Initial value]</td></tr>
<tr><td>1</td><td>Enable error response for double-bit error detection.</td></tr>
</table>
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.</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1649</td><td>CHAPTER 30: CAN FD Message RAM ECC Function
6. Registers
6.1.CAN FD ECC Error Control Register (FDECR)</td><td>"6.1. CAN FD ECC Error Control Register (FDECR)" should be corrected as indicated by the shading below.

(Error)

[bit1] DEIE: Double-bit error factor interrupt enable bit
<table>
<tr><th>DEIE</th><th>Function</th></tr>
<tr><td>0</td><td>Disable the interrupt caused by the double-bit error (FDESR.DEI).
[Initial value]</td></tr>
<tr><td>1</td><td>Enable the interrupt caused by the double-bit error (FDESR.DEI).</td></tr>
</table>
■ When DEIE is "0", even if a double-bit error is detected (FDESR.DEI = "1"), the IRQ_DE signal remains "L".
■ When DEIE 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
<table>
<tr><th>SEIE</th><th>Function</th></tr>
<tr><td>0</td><td>Disable the interrupt caused by the single-bit error (FDESR.SEI).
[Initial value]</td></tr>
<tr><td>1</td><td>Enable the interrupt caused by the single-bit error (FDESR.SEI).</td></tr>
</table>
■ When SEIE is "0", even if a single-bit error is detected (FDESR.SEI = "1"), the IRQ_SE signal remains "L".
■ When SEIE is "1", if a single-bit error is detected (FDESR.SEI = "1"), the IRQ_SE signal becomes "H" to request an interrupt.

(Correct)

[bit1] DEIE: Double-bit error factor interrupt enable bit
<table>
<tr><th>bit</th><th>Description</th></tr>
<tr><td>0</td><td>Disable the interrupt caused by the double-bit error (FDESR.DEI).
[Initial value]</td></tr>
<tr><td>1</td><td>Enable the interrupt caused by the double-bit error (FDESR.DEI).</td></tr>
</table>
■ 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
<table>
<tr><th>bit</th><th>Description</th></tr>
<tr><td>0</td><td>Disable the interrupt caused by the single-bit error (FDESR.SEI).
[Initial value]</td></tr>
<tr><td>1</td><td>Enable the interrupt caused by the single-bit error (FDESR.SEI).</td></tr>
</table>
■ 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.</td></tr>
</table>

Page	Section	Change Results
1650	CHAPTER 30: CAN FD Message RAM ECC Function 6. Registers 6.2. CAN FD ECC Error Status Register (FDESR)	Features of "For details" should be added as indicated by the shading below. (Correct) [bit7:2] Reserved: Reserved bits When writing, always write "0". When reading, "0" is always read.

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1650</td><td>CHAPTER 30: CAN FD Message RAM ECC Function
6. Registers
6.2. CAN FD ECC Error Status Register (FDESR)</td><td>"6.2. CAN FD ECC Error Status Register (FDESR)" should be corrected as indicated by the shading below.

(Error)
6.2. CAN FD ECC Error Status Register (FDESR)

<table>
<tr><th>DEI</th><th>Function</th></tr>
<tr><td>0</td><td>Indicate that a double-bit error has not been detected. [Initial value]</td></tr>
<tr><td>1</td><td>Indicate that a double-bit error has been detected.</td></tr>
</table>
Any value written in the DEI bit is ignored.
■ When the DEI bit is "1", to clear this bit to "0", write "1" in 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, "1" is set.

(Correct)
6.2. CAN FD ECC Error Status Register (FDESR)

<table>
<tr><th>bit</th><th>Description</th></tr>
<tr><td>0</td><td>Indicate that a double-bit error has not been detected. [Initial value]</td></tr>
<tr><td>1</td><td>Indicate that a double-bit error has been detected.</td></tr>
</table>
■ 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".</td></tr>
<tr><td>1652</td><td>CHAPTER 30: CAN FD Message RAM ECC Function
6. Registers
6.3. CAN FD ECC Error Status Clear Register (FDESCR)</td><td>Features of "For details" should be added as indicated by the shading below.

(Correct)
[bit7:2] Reserved: Reserved bits
When writing, always write "0". When reading, "0" is always read.</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1652</td><td>CHAPTER 30: CAN FD Message RAM ECC Function
6. Registers
6.3. CAN FD ECC Error Status Clear Register (FDESCR)</td><td>"6.3. CAN FD ECC Error Status Clear Register (FDESCR)" should be corrected as indicated by the shading below.

(Error)
6.3. CAN FD ECC Error Status Clear Register (FDESCR)
[bit1] DEIC: Double-bit error clear bit
<table>
<tr><td rowspan="2">DEIC</td><td>Function</td></tr>
<tr><td>During Write Operation</td></tr>
<tr><td>0</td><td>No effect</td></tr>
<tr><td>1</td><td>Change the double-bit error occurrence bit (FDESR.DEI) to "0".</td></tr>
</table>
"0" is always read.

[bit0] SEIC: Single-bit error clear bit
<table>
<tr><td rowspan="2">SEIC</td><td>Function</td></tr>
<tr><td>During Write Operation</td></tr>
<tr><td>0</td><td>No effect</td></tr>
<tr><td>1</td><td>Change the single-bit error occurrence bit (FDESR.SEI) to "0".</td></tr>
</table>
"0" is always read.

(Correct)
6.3. CAN FD ECC Error Status Clear Register (FDESCR)
bit1] DEIC: Double-bit error clear bit
<table>
<tr><td rowspan="2">bit</td><td>Description</td></tr>
<tr><td>During Write Operation</td></tr>
<tr><td>0</td><td>No effect</td></tr>
<tr><td>1</td><td>Change the double-bit error occurrence bit (FDESR.DEI) to "0".</td></tr>
</table>
"0" is always read.

[bit0] SEIC: Single-bit error clear bit
<table>
<tr><td rowspan="2">bit</td><td>Description</td></tr>
<tr><td>During Write Operation</td></tr>
<tr><td>0</td><td>No effect</td></tr>
<tr><td>1</td><td>Change the single-bit error occurrence bit (FDESR.SEI) to "0".</td></tr>
</table>
"0" is always read.</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1653</td><td>CHAPTER 30: CAN FD Message RAM ECC Function
6. Registers
6.4. CAN FD ECC Double-bit Error Address Register (FDDEAR)</td><td>"6.4. CAN FD ECC Double-bit Error Address Register (FDDEAR)" should be corrected as indicated by the shading below.

(Error)
6.4. CAN FD ECC Double-bit Error Address Register (FDDEAR)
[bit15:0] DRA15 to 0: Double-bit error message RAM address bits
<table>
<tr><th>DRA15 to 0 Function</th><th>Function</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>
</table>
■ Writing data in DRA15 to 0 is invalid.
■ DRA15 to 0 are valid when the DEI bit in the CAN FD ECC status register (FDESR) is "1".
■ When the DEI bit in the CAN FD ECC status register (FDESR) changes from "0" to "1", the message RAM address is written in 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 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 status register (FDESR) is cleared to "0".

(Correct)
6.4. CAN FD ECC Double-bit Error Address Register (FDDEAR)
[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>
</table>
■ 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".</td></tr>
</table>

Page	Section	Change Results
1654	CHAPTER 30: CAN FD Message RAM ECC Function 6. Registers 6.5. CAN FD ECC Single-bit Error Address Register (FDSEAR)	"6.5. CAN FD ECC Single-bit Error Address Register (FDSEAR)" should be corrected as indicated by the shading below. (Error) 6..5. CAN FD ECC Single-bit Error Address Register (FDSEAR) The CAN FD ECC single-bit error address register (FDSEAR) retains the address in the message RAM at which the relevant error occurs during single-bit error detection in the ECC test. This register is valid when the SEI bit in the CAN FD ECC status register (FDESR) is "1". (Correct) 6..5. CAN FD ECC Single-bit Error Address Register (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.

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1654</td><td>CHAPTER 30: CAN FD Message RAM ECC Function
6. Registers
6.5. CAN FD ECC Single-bit Error Address Register (FDSEAR)</td><td>"6.5. CAN FD ECC Single-bit Error Address Register (FDSEAR)" should be corrected as indicated by the shading below.

(Error)
6.5. CAN FD ECC Single-bit Error Address Register (FDSEAR)
[bit15:0] SRA15 to 0: Double-bit error message RAM address bits
<table>
<tr><th>SRA15 to 0 Function</th><th>Function</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>
</table>
■ Writing data in SRA15 to 0 is invalid.
■ SRA15 to 0 are valid when the SEI bit in the CAN FD ECC status register (FDESR) is "1".
■ When the SEI bit in the CAN FD ECC status register (FDESR) changes from "0" to "1", the message RAM address is written in 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 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 status register (FDESR) is cleared to "0".

(Correct)
6.5. CAN FD ECC Single-bit Error Address Register (FDSEAR)
[bit15:0] SRA15 to SRA0: Double-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 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>
</table>
■ 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"</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1655</td><td>CHAPTER 30: CAN FD Message RAM ECC Function
6. Registers
6.6. CAN FD ECC False Error Control Register (FDFECR)</td><td>"6.6. CAN FD ECC Error Insertion Control Register (FDFECR)" should be corrected as indicated by the shading below.

(Error)
6.6. CAN FD ECC Error Insertion Control Register (FDFECR)
The CAN FD ECC error insertion 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)
6.6.CAN FD ECC False Error Control Register (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".</td></tr>
<tr><td>1655</td><td>CHAPTER 30: CAN FD Message RAM ECC Function
6. Registers
6.6. CAN FD ECC False Error Control Register (FDFECR)</td><td>"6.6. CAN FD ECC Error Insertion Control Register (FDFECR)" should be corrected as indicated by the shading below.

(Error)
6.6. CAN FD ECC Error Insertion Control Register (FDFECR)
[bit31] FERR: Error insertion enable bit
<table><tr><th>FERR</th><th>Function</th></tr><tr><td>0</td><td>Disable writing of error data. [Initial value]</td></tr><tr><td>1</td><td>Enable writing of error data.</td></tr></table>■ 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 0 and EI15 to 0.
[bit30:19] Reserved: Reserved bits

(Correct)
6.6. CAN FD ECC Error Insertion Control Register (FDFECR)
[bit31] FERR: Error insertion enable bit
<table><tr><th>bit</th><th>Description</th></tr><tr><td>0</td><td>Disable writing of error data. [Initial value]</td></tr><tr><td>1</td><td>Enable writing of error data.</td></tr></table>■ 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.</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1656</td><td>CHAPTER 30: CAN FD Message RAM ECC Function
6. Registers
6.6. CAN FD ECC False Error Control Register (FDFECR)</td><td>"6.6. CAN FD ECC Error Insertion Control Register (FDFECR)" should be corrected as indicated by the shading below.

(Error)
6.6. CAN FD ECC Error Insertion Control Register (FDFECR)
[bit18:16] EY2 to 0: Error byte specification bits
<table>
<tr><th>EY2 to 0</th><th>Function</th></tr>
<tr><td>RY0</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>
■ 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 2 = "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 0 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 0 to specify bits.
■ When the TM_CANFERR signal is "L", the EY2 to 0 bits are all fixed to "0" and writing to these bits is ignored.
■ When the TM_CANFERR signal is "H", the EY2 to 0 bits are writable.

(Correct)
6.6. CAN FD ECC Error Insertion Control Register (FDFECR)
[bit18:16] EY2 to 0: 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>
■ 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.</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1657</td><td>CHAPTER 30: CAN FD Message RAM ECC Function
6. Registers
6.6. CAN FD ECC False Error Control Register (FDFECR)</td><td>"6.6. CAN FD ECC Error Insertion Control Register (FDFECR)" should be corrected as indicated by the shading below.

(Error)

[bit15:0] EI15 to 0: Error bit specification bits
<table>
<tr><th>EI15 to 0</th><th>Function</th></tr>
<tr><td>EI0</td><td>In the 2 bytes specified by EY2 to 0, bit0 is an error target.</td></tr>
<tr><td>EI1</td><td>In the 2 bytes specified by EY2 to 0, bit1 is an error target.</td></tr>
<tr><td>EI2</td><td>In the 2 bytes specified by EY2 to 0, bit2 is an error target.</td></tr>
<tr><td>EI3</td><td>In the 2 bytes specified by EY2 to 0, bit3 is an error target.</td></tr>
<tr><td>EI4</td><td>In the 2 bytes specified by EY2 to 0, bit4 is an error target.</td></tr>
<tr><td>EI5</td><td>In the 2 bytes specified by EY2 to 0, bit5 is an error target.</td></tr>
<tr><td>RI6</td><td>In the 2 bytes specified by EY2 to 0, bit6 is an error target.</td></tr>
<tr><td>EI7</td><td>In the 2 bytes specified by EY1 to 0, bit7 is an error target.</td></tr>
<tr><td>EI8</td><td>In the 2 bytes specified by EY1 to 0, bit8 is an error target.</td></tr>
<tr><td>EI9</td><td>In the 2 bytes specified by EY1 to 0, bit9 is an error target.</td></tr>
<tr><td>EI10</td><td>In the 2 bytes specified by EY1 to 0, bit10 is an error target.</td></tr>
<tr><td>EI11</td><td>In the 2 bytes specified by EY1 to 0, bit11 is an error target.</td></tr>
<tr><td>EI12</td><td>In the 2 bytes specified by EY1 to 0, bit12 is an error target.</td></tr>
<tr><td>EI13</td><td>In the 2 bytes specified by EY1 to 0, bit13 is an error target.</td></tr>
<tr><td>EI14</td><td>In the 2 bytes specified by EY1 to 0, bit14 is an error target.</td></tr>
<tr><td>EI15</td><td>In the 2 bytes specified by EY1 to 0, 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>

Page	Section	Change Results
1657	CHAPTER 30: CAN FD Message RAM ECC Function 6. Registers 6.6. CAN FD ECC False Error Control Register (FDFECR)	"6.6. CAN FD ECC Error Insertion Control Register (FDFECR)" should be corrected as indicated by the shading below. (Error) 6.6. CAN FD ECC Error Insertion Control Register (FDFECR) [bit15:0] EI15 to 0: Error bit specification bits ■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 2 = EI4 to 15 = "0", the bit data for 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 2 = EI4 to 15 = "0", the bit data for bit3 and bit19 in the message RAM is inverted before it is written. The other bits are written as they are. ■ When EI15 to 0 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 0 to specify bits. ■ When the TM_CANFERR signal is "L", the EI15 to 0 bits are all fixed to "0" and writing to these bits is ignored. ■ When the TM_CANFERR signal is "H", the EI15 to 0 bits are writable. (Correct) 6.6. CAN FD ECC Error Insertion Control Register (FDFECR) [bit15:0] EI15 to 0: Error bit specification bits ■ 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.

Page	Section	Change Results
1658	CHAPTER 30: CAN FD Message RAM ECC Function 7. Precautions	"7. Precautions for Using This Device" should be corrected as indicated by the shading below. (Error) 7. Precautions for Using This Device 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 test** ■ 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 test 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. (Correct) 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.

Page	Section	Change Results
1658	CHAPTER 30: CAN FD Message RAM ECC Function 7. Precautions	"7. Precautions for Using This Device" should be corrected as indicated by the shading below. (Error) **ECC error insertion function** ■ Even if the TM_CANFERR signal is set to "L" after error data is written to the message RAM by using the ECC error insertion 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 error insertion 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 error insertion function, error data can be written to the message RAM by the CAN FD controller or CPU. (Correct) **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.

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1842</td><td>CHAPTER 32: CAN Prescaler
5.1. CAN Prescaler Control Register (CANP_CTR)</td><td>"5.1. CAN Prescaler Control Register (CANP_CTR)" should be added as indicated by the shading below.

(Error)
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 clock, confirm that the PLL clock is selected by reading CANP_STR.SCKM in the CAN prescaler status register before starting communication with CAN.

(Correct)
− 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 clock, confirm that the PLL 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 clock is selected.</td></tr>
<tr><td>1843</td><td>CHAPTER 32: CAN Prescaler
5.2. CAN Prescaler Status Register (CANP_STR)</td><td>"5.2. CAN Prescaler Status Register (CANP_STR)" 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 colspan="5">Reserved</td><td colspan="2">SCKM[1:0]</td><td>BUSY</td></tr>
<tr><td>R/W Attribute</td><td colspan="5">R0,WX</td><td colspan="2">R,WX</td><td>R,WX</td></tr>
<tr><td>Protection Attribute</td><td colspan="8"></td></tr>
<tr><td>Initial Value</td><td colspan="5">00000</td><td colspan="2">00</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 colspan="5">Reserved</td><td colspan="2">SCKM[1:0]</td><td>BUSY</td></tr>
<tr><td>R/W Attribute</td><td colspan="5">R0,WX</td><td colspan="2">R,WX</td><td>R,WX</td></tr>
<tr><td>Protection Attribute</td><td colspan="8">-</td></tr>
<tr><td>Initial Value</td><td colspan="5">00000</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>1843</td><td>CHAPTER 32: CAN Prescaler
5.2. CAN Prescaler Status Register (CANP_STR)</td><td>"5.2. CAN Prescaler Status Register (CANP_STR)" should be corrected as indicated by the shading below.

(Error)
[bit2:1] SCKM: Source clock display
These bits indicate the currently selected source clock.
<table>
<tr><th>Bit:2</th><th>Bit:1</th><th>Description</th></tr>
<tr><td>0</td><td>1</td><td>PLL clock</td></tr>
<tr><td>1</td><td>0</td><td>Main clock</td></tr>
<tr><td colspan="2">Other than above</td><td>On-chip bus clock</td></tr>
</table>
(Correct)
[bit2:1] SCKM: Source clock display
These bits indicate the currently selected source clock.
<table>
<tr><th>Bit:2</th><th>Bit:1</th><th>Description</th></tr>
<tr><td>0</td><td>0</td><td>Clock stop</td></tr>
<tr><td>0</td><td>1</td><td>PLL clock</td></tr>
<tr><td>1</td><td>0</td><td>Main clock</td></tr>
<tr><td>1</td><td>1</td><td>State in the middle of the source clock selection</td></tr>
</table>
Note:
− In case of the middle of the source clock selection, On-chip bus clock is selected.</td></tr>
</table>

Page	Section	Change Results
1844	CHAPTER 32: CAN Prescaler 6. Restriction	"6. Restriction" should be corrected as indicated by the shading below. (Error) 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 (Correct) 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

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1846</td><td>CHAPTER 33: 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)
Functions of the UART (asynchronous serial interface)
<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 (64 bytes each) (when FIFOs are used)</td></tr>
</table>
(Correct)
Functions of the UART (Asynchronous Serial Interface)
<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>1848</td><td>CHAPTER 33: 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)
Functions of the CSIO (clock synchronous serial interface)
<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 (64 bytes each) (when FIFOs are used)</td></tr>
</table>
(Correct)
Functions of the CSIO (Clock Synchronous Serial Interface)
<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>1849</td><td>CHAPTER 33: 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)
3.1.Manual Mode
<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 (64 bytes each) (when FIFOs are used)</td></tr>
</table>
(Correct)
3.1.Manual Mode
<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>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1851,
1852</td><td>CHAPTER 33: 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)
3.2.Assist Mode
<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 (64 bytes each) (when FIFOs are used)</td></tr></table>
(Correct)
3.2.Assist Mode
<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>1853</td><td>CHAPTER 33: 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)
Functions of the I²C interface (I²C communication control interface)
<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 (64 bytes each) (when FIFOs are used)</td></tr></table>
(Correct)
Functions of the I²C Interface (I²C Communication Control Interface)
<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>1855</td><td>CHAPTER 33: Overview of Multi-Function Serial Interface
5. Block Diagram</td><td>"5. Block Diagram" should be corrected as indicated by the shading below.

(Error)
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 waits for the I2C bus.

(Correct)
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.</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1864</td><td>CHAPTER 34
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)
UART (Asynchronous serial interface) functions
<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)
UART (Asynchronous Serial Interface) Functions
<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>1942</td><td>CHAPTER 34
UART
(Asynchronous
Serial Interface)
10.9.Transfer Byte
Register (TBYTE0)</td><td>Features of "For details" should be added as indicated by the shading below.

(Correct)
[bit7:0] TBYTE0: Transfer Byte Resister</td></tr>
<tr><td>1953</td><td>CHAPTER 34
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

(Correct)
[bit15:8] FBYTE2: FIFO2 Data Count Indication Byte
[bit7:0] FBYTE1: FIFO1 Data Count Indication Byte</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1956</td><td>CHAPTER 34: UART (Asynchronous Serial Interface) 10.14.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 generates an interrupt.</td></tr>
<tr><td>Read</td><td>Read valid data count.</td></tr>
</table>

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.
– If block transfer is performed during DMA transfer, only block size 1 can be set. To perform block transfer with ECR:TXBLKEN = 1, set the block size value in this register.

(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>
<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>

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.
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1962</td><td>CHAPTER 34 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

(Correct)
[bit7:0] TBSIZE: Transmission Block Size Byte</td></tr>
<tr><td>1974</td><td>CHAPTER 35:CSIO (Clock Synchronous Serial 1.Overview of the CSIO (Clock Synchronous Serial Interface)</td><td>Features of "For details" should be corrected as indicated by the shading below.

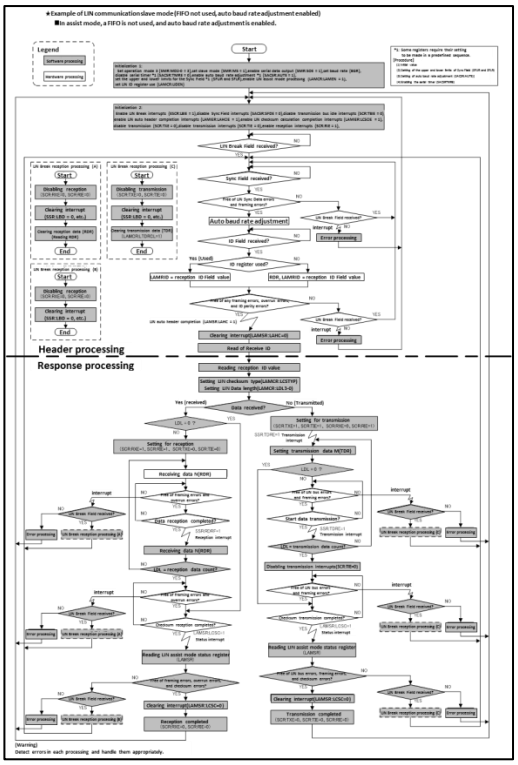
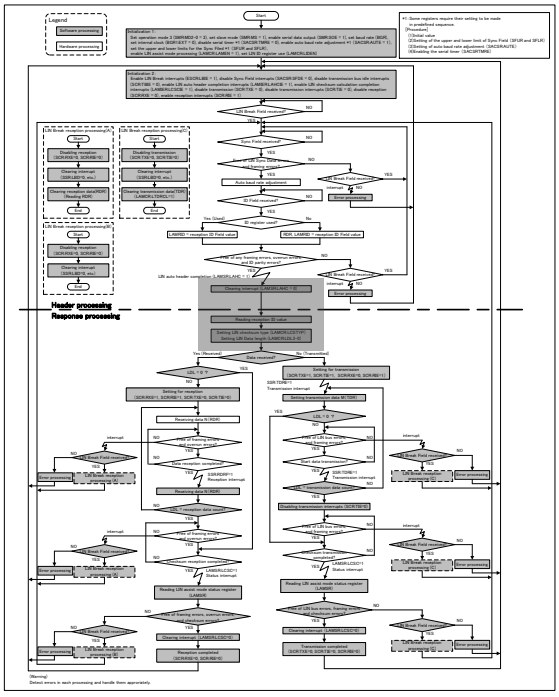
(Error)
Functions of the CSIO (Clock Synchronous Serial Interface)
<table>
<tr><th></th><th></th><th>Function</th></tr>
<tr><td>1</td><td>Data buffer</td><td>− Full-duplex double buffer (when FIFO is not used)
− Transmission and reception FIFOs (64 bytes each)*1 (when FIFO is used)</td></tr>
</table>
(Correct)
Functions of the CSIO (Clock Synchronous Serial Interface)
<table>
<tr><th></th><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>2118</td><td>CHAPTER 35: CSIO (Clock Synchronous Serial 9.12.Transfer Byte Registers (TBYTE3 to TBYTE0)</td><td>Features of "For details" should be added as indicated by the shading below.

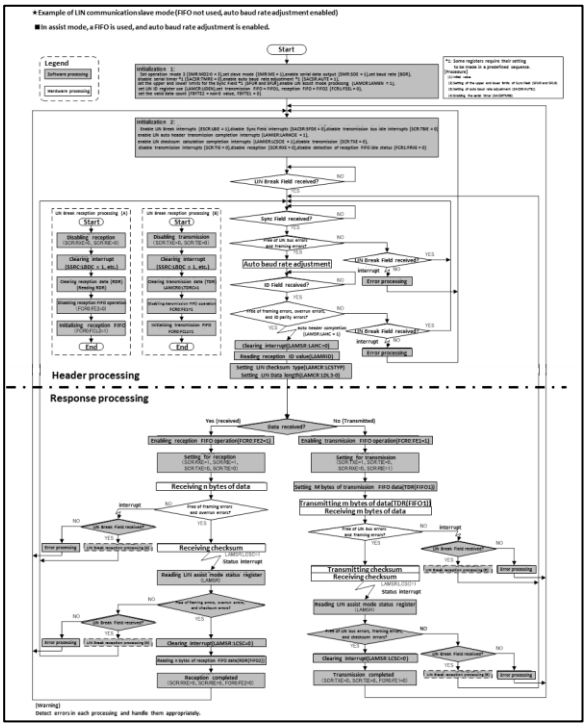
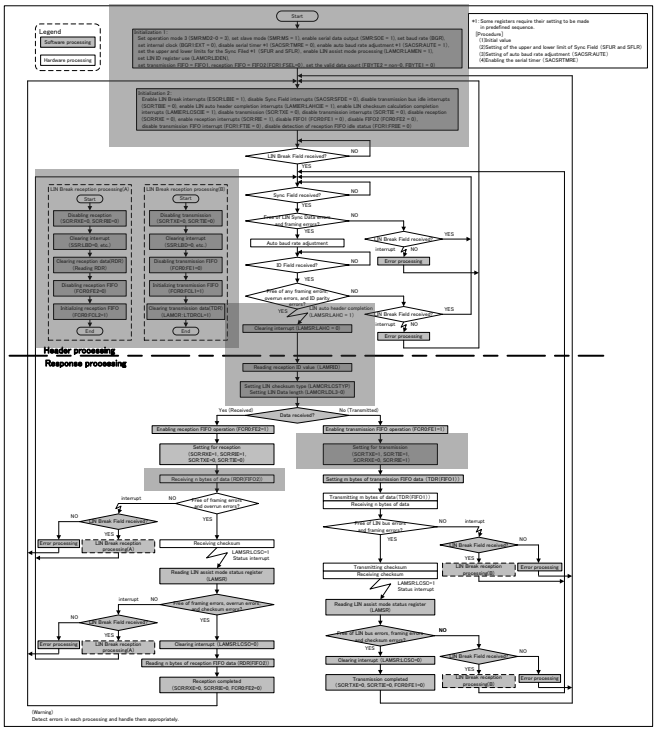
(Correct)
[bit15] TBYTE: Transfer Byte</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>2148,
2149</td><td>CHAPTER 36:LIN Interface (V2.1) (LIN Communication Control Interface (V2.1))
1.1.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 FIFO is not used)
− Transmission and reception FIFOs (64 bytes 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 Buffer</td><td>Full-duplex double buffering (when FIFO is not used)
− Transmission and reception FIFOs (64 bytes each) (when FIFO is used)</td></tr></table></td></tr>
<tr><td>2150,
2151</td><td>CHAPTER 36:LIN Interface (V2.1) (LIN Communication Control Interface (V2.1))
1.2.Assist Mode</td><td>Features of "For details" should be corrected as indicated by the shading below.

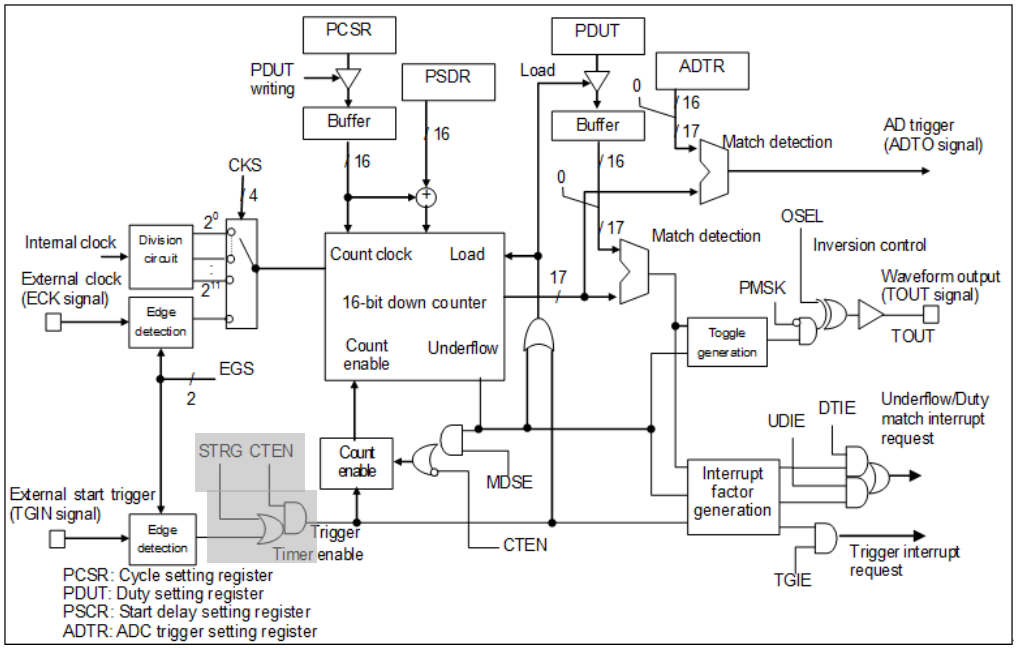
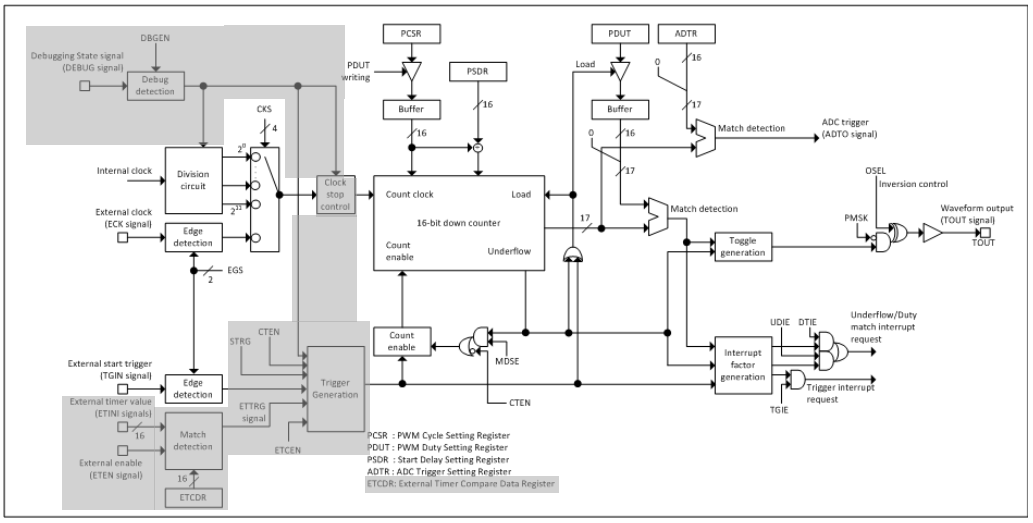
(Error)
LIN interface (v2.1) (LIN communication control interface (v2.1)) functions (assist mode)
<table><tr><th></th><th></th><th>Function</th></tr><tr><td>1</td><td>Data Buffer</td><td>− Full-duplex double buffering (when FIFO is not used)
− Transmission and reception FIFOs (64 bytes each)(when a FIFO is used)</td></tr></table>
(Correct)
LIN Interface (v2.1) (LIN Communication Control Interface (v2.1)) Functions (Assist Mode)
<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 FIFO is not used)
− Transmission and reception FIFOs (64 bytes each)(when a FIFO is used)</td></tr></table></td></tr>
</table>

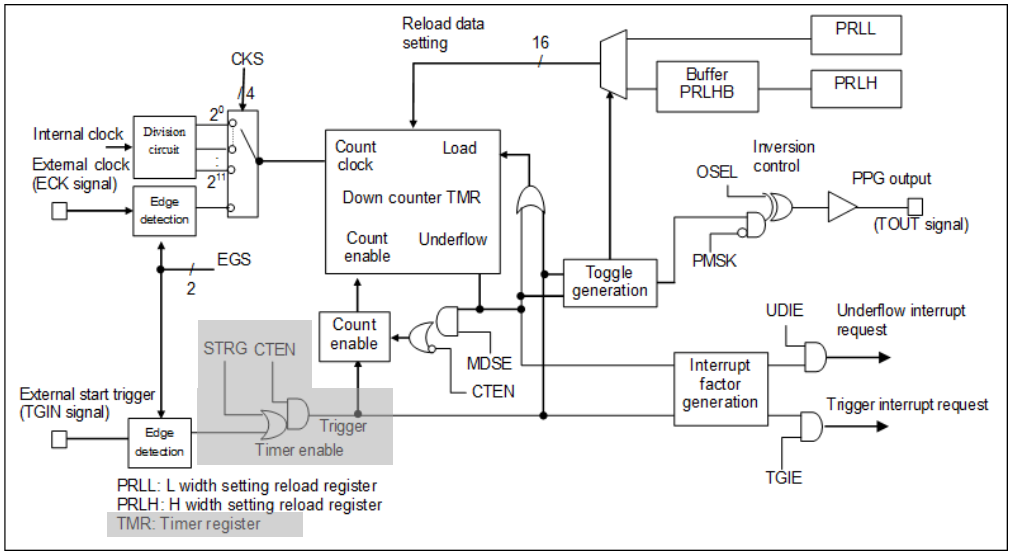
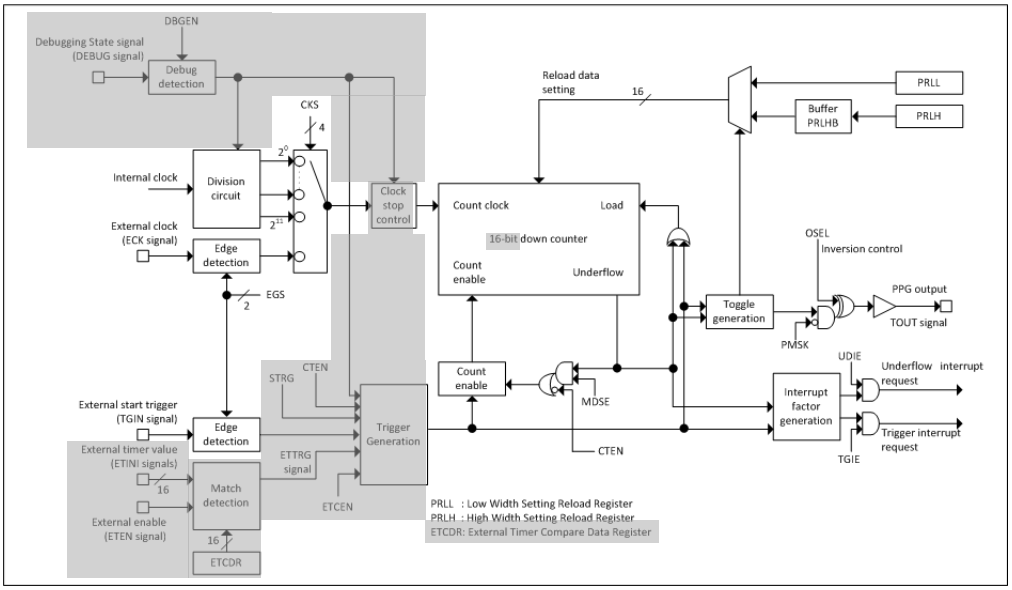
Page	Section	Change Results
2256	CHAPTER 36: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 8.2. Assist Mode	"Figure 8-8 " should be corrected as indicated by the shading below (Error)  (Correct) 

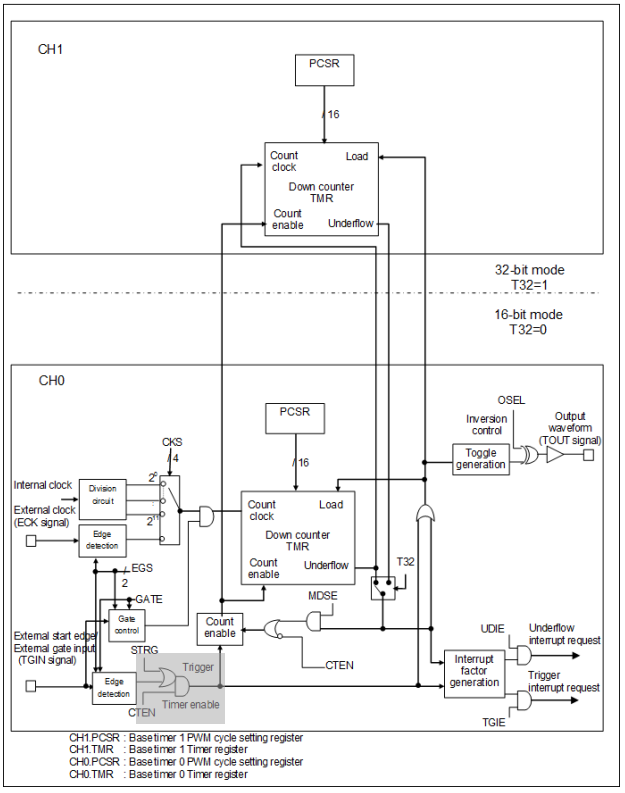
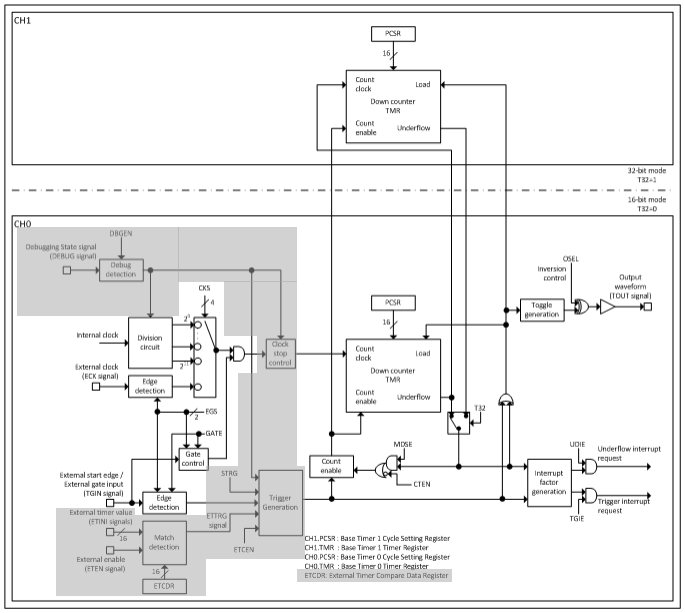
Page	Section	Change Results
2257	CHAPTER 36: LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 8.2. Assist Mode	"Figure 8-9 " should be corrected as indicated by the shading below (Error)  (Correct) 

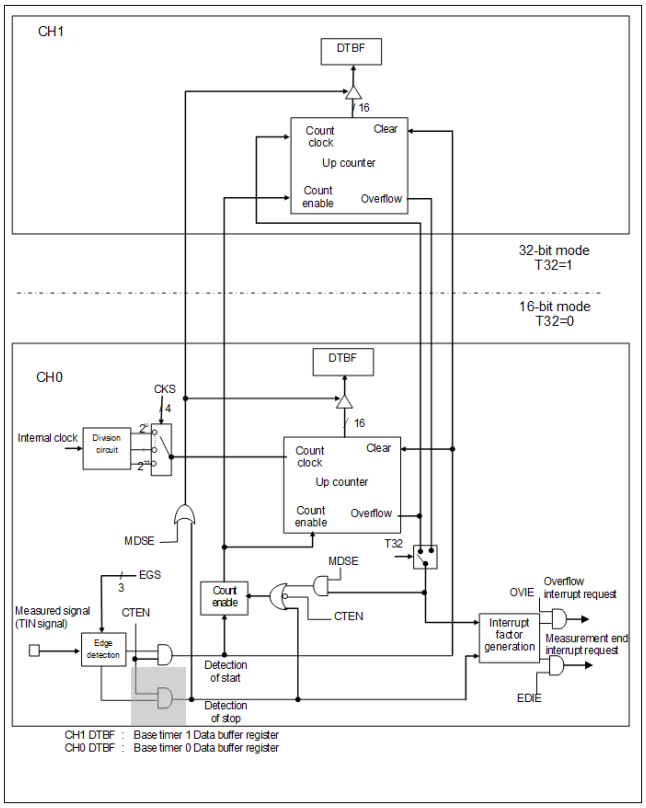
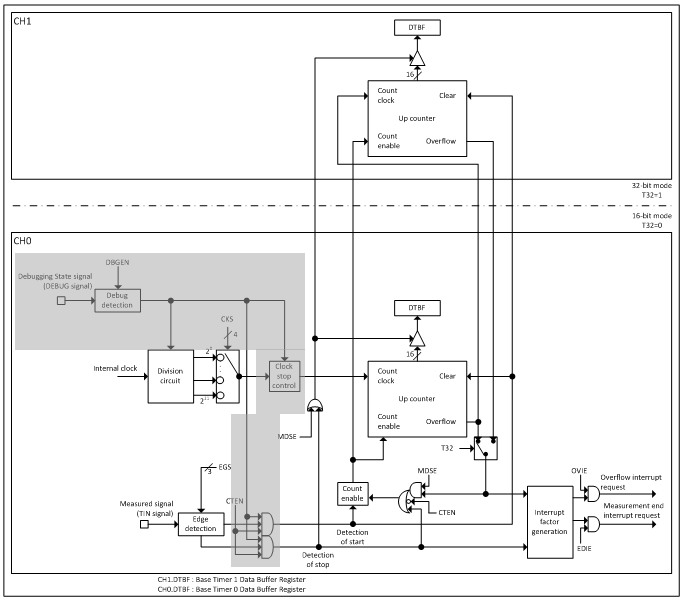
Page	Section	Change Results
2258	HAPTER 36:LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.20.IIFO Byte Register (FBYTE)	Features of "For details" should be corrected as indicated by the shading below. (Error) 9. Interface (V2.1) Registers (Correct) 9. LIN Interface (V2.1) Registers
2312	HAPTER 36:LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9. LIN Interface (V2.1) Registers	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
2314	HAPTER 36:LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.21.Transmission FIFO Interrupt Control Register (FTICR)	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
2320	HAPTER 36:LIN Interface (V2.1) (LIN Communication Control Interface (V2.1)) 9.24.Transmission Block Size Register (TBSIZE)	Features of "For details" should be corrected as indicated by the shading below. (Error) TBSIZE bit (Correct) [bit7:0]TBSIZE
2349 to 2510	CHAPTER 37: I2C Interface(I2C Communication Control Interface)	Added a new CHAPTER named “I2C Interface”.

Page	Section	Change Results
2512	CHAPTER 38:Base Timer 1.Overview of the Base Timer	Features of "For details" should be corrected as indicated by the shading below. (Error) 16-bit PWM timer The 16-bit PWM timer operates with the cycle of 16-bit. The 16-bit PWM timer is described to be 16-bit down counter where irrelevant to the start delay control. (Correct) 16-Bit PWM Timer The 16-bit PWM Timer operates in 16-bit periods. The down counter will be described in 16 bit formats when the 16-bit PWM Timer is irrelevant to the start delay control. The external timer match starting can start a timer, when a match of the value set as the External Timer Compare Data Register (ETCDR) and an external timer value is detected.
2513	CHAPTER 38:Base Timer 1.Overview of the Base Timer	Features of "For details" should be added as indicated by the shading below. (Correct) 16-Bit PPG Timer The external timer match starting can start a timer, when a match of the value set as the External Timer Compare Data Register (ETCDR) and an external timer value is detected.
2513	CHAPTER 38:Base Timer 1.Overview of the Base Timer	Features of "For details" should be added as indicated by the shading below. (Correct) 16/32-Bit Reload Timer The external timer match starting can start a timer, when a match of the value set as the External Timer Compare Data Register (ETCDR) and an external timer value is detected.
2513	CHAPTER 38:Base Timer 1.Overview of the Base Timer	Features of "For details" should be corrected as indicated by the shading below. (Error) 16/32-bit PWC timer Each measurement mode: (Correct) 16/32-Bit PWC Timer The supported measurement modes:

Page	Section	Change Results
2515	CHAPTER 38:Base Timer 2.Block Diagrams of the Base Timer	Features of "For details" should be corrected as indicated by the shading below. (Error) Figure 2-1 Block Diagram of the 16-bit PWM Timer  (Correct) Figure 2-1 Block Diagram of the 16-Bit PWM Timer 

Page	Section	Change Results
2516	CHAPTER 38:Base Timer 2.Block Diagrams of the Base Timer	Features of "For details" should be corrected as indicated by the shading below. (Error) Figure 2-2 Block Diagram of the 16-bit PPG Timer  (Correct) Figure 2-2 Block Diagram of the 16-Bit PPG Timer 

Page	Section	Change Results
2517	CHAPTER 38:Base Timer 2.Block Diagrams of the Base Timer	Features of "For details" should be corrected as indicated by the shading below. (Error) Figure 2-3 Block Diagram of the 16/32-bit Reload Timer (ch1, ch0)  (Correct) Figure 2-3 Block Diagram of the 16/32-Bit Reload Timer (ch1, ch0) 

Page	Section	Change Results
2518	CHAPTER 38:Base Timer 2.Block Diagrams of the Base Timer	Features of "For details" should be corrected as indicated by the shading below. (Error) Figure 2-4 Block Diagram of the 16/32-bit PWC Timer (ch1, ch0)  (Correct) Figure 2-4 Block Diagram of the 16/32-Bit PWC Timer (ch1, ch0) 

Page	Section	Change Results
2519	CHAPTER 38:Base Timer 3.Operations of the Base Timer	Features of "For details" should be corrected as indicated by the shading below. (Error) 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. 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. 16-bit reload timer 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.. 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. (Correct) 16-Bit PWM Timer The 16-bit PWM Timer starts counting down from the set cycle value upon trigger activation. In the beginning the output level is "L". 16-Bit PPG Timer The 16-bit PPG Timer starts counting down from the set value in the Low Width Setting Reload Register (PRLL) upon trigger activation. In the beginning the output level is "L". 16-Bit Reload Timer The output level is defined by the MDSE bit setting - the polarity of the output is toggled each time an underflow is detected, or, the output is set to "H" when counting starts and switches to "L" when an underflow is detected. 16-Bit PWC Timer The PWC Timer starts the 16-bit up counter upon detection of the measurement start edge, and stops the counter upon the detection of a measurement end edge. The resulting count value will be stored as the measured pulse width in the Data Buffer Register (DTBF).

Page	Section	Change Results
2520	CHAPTER 38:Base Timer 4.32-bit Mode Operation	Features of "For details" should be corrected as indicated by the shading below. (Error) 32-bit mode settings There is no need to set reset mode. 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. 32-bit mode operation Timer start, waveform output, and interrupt signals for the even-numbered channel are valid. (The odd-numbered channel is fixed at "L" and masked.) (Correct) 32-Bit Mode Settings There is no need to set the odd-numbered channel to Reset mode. After the writing of the T32 bit, the change to 32-bit operation mode is reflected immediately, so change the settings for both channels while the counter is in a stopped state. 32-Bit Mode Operation The timer start, waveform output, and interrupt signals of the even-numbered channel are valid. (The corresponding signals for the odd-numbered channel will be masked to "L".)
2523	CHAPTER 38:Base Timer 5.Debug Mode	Features of "For details" should be added as indicated by the shading below. (Correct) 5.Debug Mode If it goes into a Debug mode, operation of the Base Timer can be stopped and debugged. If all the following conditions are satisfied, it will go into the Debug mode. −The debug enable bit (DBGEN) of the Base Timer Debug Register (BT_DEBUG) is "1". −The debug signal (DEBUG) is active. It becomes the following operation when it goes into the Debug mode. −The prescaler stop. −The timer stop. −The output waveform stop. −It holds the output value at the time of the stop. −The trigger is ignored in PWM Timer, PPG Timer or Reload timer. −The measurement edge is ignored in PWC Timer. It becomes the following operation when it restores from the Debug mode. −The prescaler is resumed from the stopped value. −The timer is resumed from the stopped value. −The output waveform resumes the output from the stopped state. Note: −Register access is possible in the Debug mode.

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>2526</td><td>CHAPTER 38:Base Timer
8.Registers of the Base Timer</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
Table 7-1 List of Registers When the 16-bit PWM Timer Is Selected
<table>
<tr><th>Abbreviated Register Name</th><th>Register Name</th><th>Reference</th></tr>
<tr><td>TMCR</td><td>Timer control register</td><td>9.1.8</td></tr>
<tr><td>TMCR2</td><td>Timer control register 2</td><td>9.1.8</td></tr>
<tr><td>STC</td><td>Status control register</td><td>9.1.8</td></tr>
<tr><td>STCC</td><td>Status Control Clear Register</td><td>9.1.8</td></tr>
<tr><td>STCS</td><td>Status Control Set Register</td><td>9.1.8</td></tr>
<tr><td>PCSR</td><td>PWM cycle setting register</td><td>9.1.9</td></tr>
<tr><td>PDUT</td><td>PWM duty setting register</td><td>9.1.10</td></tr>
<tr><td>TMR</td><td>Timer register</td><td>9.1.11</td></tr>
<tr><td>PSDR</td><td>Start delay value setting register</td><td>9.1.12</td></tr>
<tr><td>ADTR</td><td>ADC trigger value setting register</td><td>9.1.13</td></tr>
</table>
(Correct)
Table 8-1 List of Registers When the 16-Bit PWM Timer Is Selected
<table>
<tr><th>Abbreviated Register Name</th><th>Register Name</th><th>Reference</th></tr>
<tr><td>TMCR</td><td>Timer Control Register</td><td>10.1.9</td></tr>
<tr><td>TMCR2</td><td>Timer Control Register 2</td><td>10.1.9</td></tr>
<tr><td>STC</td><td>Status Control Register</td><td>10.1.9</td></tr>
<tr><td>STCC</td><td>Status Control Clear Register</td><td>10.1.9</td></tr>
<tr><td>STCS</td><td>Status Control Set Register</td><td>10.1.9</td></tr>
<tr><td>PCSR</td><td>PWM Cycle Setting Register</td><td>10.1.10</td></tr>
<tr><td>PDUT</td><td>PWM Duty Setting Register</td><td>10.1.11</td></tr>
<tr><td>TMR</td><td>Timer Register</td><td>10.1.12</td></tr>
<tr><td>PSDR</td><td>Start Delay Value Setting Register</td><td>10.1.13</td></tr>
<tr><td>ADTR</td><td>ADC Trigger Value Setting Register</td><td>10.1.14</td></tr>
<tr><td>BT_DEBUG</td><td>Base Timer Debug Register</td><td>10.1.15</td></tr>
<tr><td>ETCDR</td><td>External Timer Compare Data Register</td><td>10.1.16</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>2526</td><td>CHAPTER 38:Base Timer
8.Registers of the Base Timer</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
Table 7-2 List of Registers When the 16-bit PPG Timer Is Selected
<table>
<tr><th>Abbreviated Register Name</th><th>Register Name</th><th>Reference</th></tr>
<tr><td>TMCR</td><td>Timer control register</td><td>9.2.6</td></tr>
<tr><td>TMCR2</td><td>Timer control register 2</td><td>9.2.6</td></tr>
<tr><td>STC</td><td>Status control register</td><td>9.2.6</td></tr>
<tr><td>STCC</td><td>Status Control Clear Register</td><td>9.2.6</td></tr>
<tr><td>STCS</td><td>Status Control Set Register</td><td>9.2.6</td></tr>
<tr><td>PRLL</td><td>L width setting reload register</td><td>9.2.7</td></tr>
<tr><td>PRLH</td><td>H width setting reload register</td><td>9.2.8</td></tr>
<tr><td>TMR</td><td>Timer register</td><td>9.2.9</td></tr>
</table>

(Correct)
Table 8-2 List of Registers When the 16-Bit PPG Timer Is Selected
<table>
<tr><th>Abbreviated Register Name</th><th>Register Name</th><th>Reference</th></tr>
<tr><td>TMCR</td><td>Timer Control Register</td><td>10.2.7</td></tr>
<tr><td>TMCR2</td><td>Timer Control Register 2</td><td>10.2.7</td></tr>
<tr><td>STC</td><td>Status Control Register</td><td>10.2.7</td></tr>
<tr><td>STCC</td><td>Status Control Clear Register</td><td>10.2.7</td></tr>
<tr><td>STCS</td><td>Status Control Set Register</td><td>10.2.7</td></tr>
<tr><td>PRLL</td><td>Low Width Setting Reload Register</td><td>10.2.8</td></tr>
<tr><td>PRLH</td><td>High Width Setting Reload Register</td><td>10.2.9</td></tr>
<tr><td>TMR</td><td>Timer Register</td><td>10.2.10</td></tr>
<tr><td>BT_DEBUG</td><td>Base Timer Debug Register</td><td>10.2.11</td></tr>
<tr><td>ETCDR</td><td>External Timer Compare Data Register</td><td>10.2.12</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>2527</td><td>CHAPTER 38:Base Timer
8.Registers of the Base Timer</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
Table 7-3 List of Registers When the Reload Timer Is Selected
<table>
<tr><th>Abbreviated Register Name</th><th>Register Name</th><th>Reference</th></tr>
<tr><td>TMCR</td><td>Timer control register</td><td>9.3.3</td></tr>
<tr><td>TMCR2</td><td>Timer control register 2</td><td>9.3.3</td></tr>
<tr><td>STC</td><td>Status control register</td><td>9.3.3</td></tr>
<tr><td>STCC</td><td>Status Control Clear Register</td><td>9.3.3</td></tr>
<tr><td>STCS</td><td>Status Control Set Register</td><td>9.3.3</td></tr>
<tr><td>PCSR</td><td>Cycle setting register</td><td>9.3.4</td></tr>
<tr><td>TMR</td><td>Timer register</td><td>9.3.5</td></tr>
</table>

(Correct)
Table 8-3 List of Registers When the Reload Timer Is Selected
<table>
<tr><th>Abbreviated Register Name</th><th>Register Name</th><th>Reference</th></tr>
<tr><td>TMCR</td><td>Timer Control Register</td><td>10.3.4</td></tr>
<tr><td>TMCR2</td><td>Timer Control Register 2</td><td>10.3.4</td></tr>
<tr><td>STC</td><td>Status Control Register</td><td>10.3.4</td></tr>
<tr><td>STCC</td><td>Status Control Clear Register</td><td>10.3.4</td></tr>
<tr><td>STCS</td><td>Status Control Set Register</td><td>10.3.4</td></tr>
<tr><td>PCSR</td><td>Cycle Setting Register</td><td>10.3.5</td></tr>
<tr><td>TMR</td><td>Timer Register</td><td>10.3.6</td></tr>
<tr><td>BT_DEBUG</td><td>Base Timer Debug Register</td><td>10.2.11</td></tr>
<tr><td>ETCDR</td><td>External Timer Compare Data Register</td><td>10.2.12</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>2527</td><td>CHAPTER 38:Base Timer
8.Registers of the Base Timer</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
Table 7-4 List of Registers When the PWC Timer Is Selected
<table>
<tr><th>Abbreviated Register Name</th><th>Register Name</th><th>Reference</th></tr>
<tr><td>TMCR</td><td>Timer control register</td><td>9.4.2</td></tr>
<tr><td>TMCR2</td><td>Timer control register 2</td><td>9.4.2</td></tr>
<tr><td>STC</td><td>Status control register</td><td>9.4.2</td></tr>
<tr><td>STCC</td><td>Status Control Clear Register</td><td>9.4.2</td></tr>
<tr><td>STCS</td><td>Status Control Set Register</td><td>9.4.2</td></tr>
<tr><td>DTBF</td><td>Data buffer register</td><td>9.4.3</td></tr>
</table>

(Correct)
Table 8-4 List of Registers When the PWC Timer Is Selected
<table>
<tr><th>Abbreviated Register Name</th><th>Register Name</th><th>Reference</th></tr>
<tr><td>TMCR</td><td>Timer Control Register</td><td>10.4.2</td></tr>
<tr><td>TMCR2</td><td>Timer Control Register 2</td><td>10.4.2</td></tr>
<tr><td>STC</td><td>Status Control Register</td><td>10.4.2</td></tr>
<tr><td>STCC</td><td>Status Control Clear Register</td><td>10.4.2</td></tr>
<tr><td>STCS</td><td>Status Control Set Register</td><td>10.4.2</td></tr>
<tr><td>DTBF</td><td>Data Buffer Register</td><td>10.4.3</td></tr>
<tr><td>BT_DEBUG</td><td>Base Timer Debug Register</td><td>10.2.11</td></tr>
</table></td></tr>
<tr><td>2528</td><td>CHAPTER 38:Base Timer
9.Notes on Using the Base Timer</td><td>Features of "For details" should be deleted as indicated by the shading below.

(Error)
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." [x]</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>2528</td><td>CHAPTER 38:Base Timer
9.Notes on Using the Base Timer</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
Notes to observe on configuration using program, common to use of timers
[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

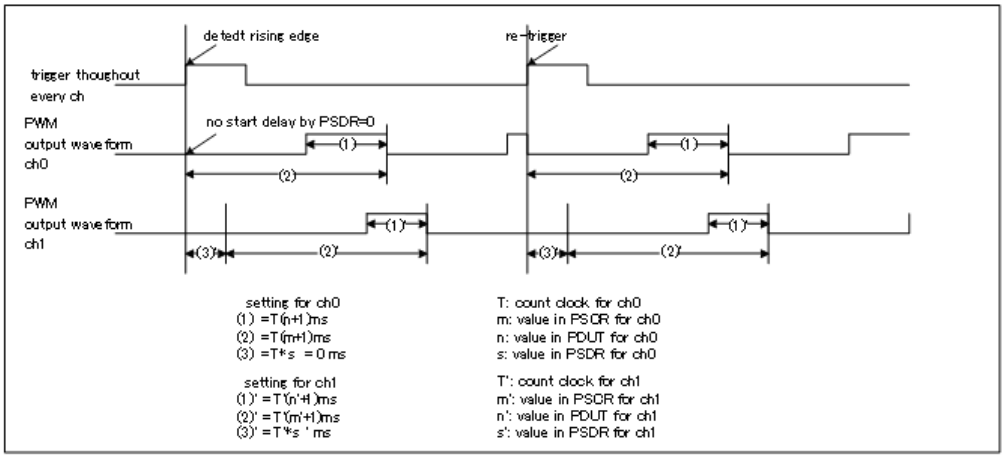
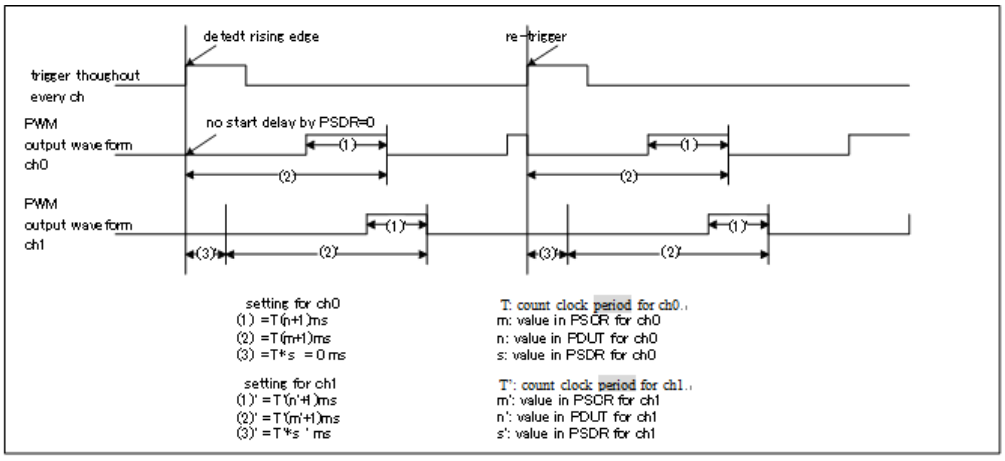
(Correct)
Notes to observe on configuration using program, common to use of timers
<table>
<tr><th>Bit</th><th>Bit Name</th><th>Description</th></tr>
<tr><td>[TMCR2 bit0], [TMCR bit14,13,12]</td><td>CKS3 to CKS0</td><td>Clock selection bits</td></tr>
<tr><td>[bit10,9,8]</td><td>EGS2, EGS1, EGS0</td><td>Measurement edge selection bits</td></tr>
<tr><td>[bit7]</td><td>T32</td><td>32-bit timer selection bit (when the Reload Timer and PWC function are selected)</td></tr>
<tr><td>[bit6,5,4]</td><td>FMD2 to FMD0</td><td>Timer function selection bits</td></tr>
<tr><td>[bit2]</td><td>MDSE</td><td>Measurement mode (single/continuous) selection bit</td></tr>
</table>
</td></tr>
<tr><td>2528</td><td>CHAPTER 38:Base Timer
9.Notes on Using the Base Timer</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
Notes on using the 16-bit PWM/PPG/reload timer
If a restart is detected at the count end time in one-shot mode, the count value is reloaded. Then, the restart begins.

(Correct)
Notes on using the 16-bit PWM/PPG/Reload Timer
If the Base Timer is configured in one-shot mode and is restarted when count end is detected, the counter will reload the count value and restart.
</td></tr>
</table>

Page	Section	Change Results
2528	CHAPTER 38:Base Timer 9.Notes on Using the Base Timer	Features of "For details" should be corrected as indicated by the shading below. (Error) Notes on using the PWC timer 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". If a restart is performed after a count operation begins, operations such as the following may occur, depending on the timing. (Correct) Notes on using the PWC Timer If the PWC mode setting (FMD2 to FMD0 = "100") and the measurement start setting (CTEN="1") are set simultaneously after a system reset assertion or upon leaving Reset mode, the PWC Timer operation may vary due to the state of the immediately preceding measurement signal. When operating in continuous measurement mode, if a measurement start edge is detected at the same time that the Base Timer has been restarted, 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 the restart has been performed.

Page	Section	Change Results
2531	CHAPTER38: Base Timer 10.1.PWM Timer Function	Features of "For details" should be corrected as indicated by the shading below. (Error) 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) (Correct) 10.1.1 16-bit PWM Timer Operation 10.1.2 One-shot Operation 10.1.3 Start Delay Function 10.1.4 ADC Trigger Output Timing 10.1.5 External Timer Match Starting 10.1.6 Interrupt Factors and Timing Chart 10.1.7 Output Waveform 10.1.8.PWM Timer Operation Flow 10.1.9 Timer Control Registers (TMCR, TMCR2), Status Control Register (STC), Status Control Clear Register (STCC), and Status Control Set Register (STCS) 10.1.10 PWM Cycle Setting Register (PCSR) 10.1.11 PWM Duty Setting Register (PDUT) 10.1.12 Timer Register (TMR) 10.1.13 Start Delay Value Setting Register (PSDR) 10.1.14 ADC Trigger Value Setting Register (ADTR) 10.1.15 Base Timer Debug Register (BT_DEBUG) 10.1.16 External Timer Compare Data Register (ETCDR)
2535	CHAPTER38: Base Timer 10.1.3.Start Delay Function	Features of "For details" should be corrected as indicated by the shading below. (Error) ***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. (Correct) ***Start Delay Operation with Multiple Channels interlocked*** The trigger can be controlled to interlock several channels in a specific phase relationship.

Page	Section	Change Results
2536	CHAPTER38: Base Timer 10.1.3.Start Delay Function	Features of "For details" should be corrected as indicated by the shading below. (Error) Figure 9-10 Start Delay Operation in Continuous Operation with Multiple Channels Interlocked (Trigger Restart Enabled)  (Correct) Figure 10-10 Start Delay Operation in Continuous Operation with Multiple Channels Interlocked (Trigger Restart Enabled) 
2537	CHAPTER38: Base Timer 10.1.5.External Timer Match Starting	Added the section "10.1.5.External Timer Match Starting "

Page	Section	Change Results
2538	CHAPTER38: Base Timer 10.1.6.Interrupt Factors and Timing Chart	Features of "For details" should be corrected as indicated by the shading below. (Error) **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). (Correct) **Interrupt Factors and Timing Chart (PWM Output: Normal Polarity)** The time required for a counter value to be loaded following the input of a trigger is, for software triggers T, and for external triggers 2T to 3T (T: internal clock cycle).
2538, 2539	CHAPTER38: Base Timer 10.1.7.Output Waveform	Features of "For details" should be corrected as indicated by the shading below. (Error) Figure 9-13 Example Where PWM Output Is All "L" Level Figure 9-14 Example Where PWM Output Is All "H" Level (Correct) Figure 10-14 Example Where PWM Output Is All "L" Level（TMCR.OSEL="0"） Figure 10-15 Example Where PWM Output Is All "H" Level（TMCR.OSEL="0"）
2541	CHAPTER38: Base Timer 10.1.9.1 Timer Control Register (Upper Byte of TMCR)	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 (Correct) [TMCR2:bit0,bit14:12] CKS3 to CKS0: Count Clock Selection Bits

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>2545</td><td>CHAPTER38: Base Timer
10.1.9.3 Timer Control Register 2 (TMCR2)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
Timer control register 2 (TMCR2)
<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="7">Reserved</td><td>CKS3</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>

(Correct)
10.1.9.3 Timer Control Register 2 (TMCR2)
<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="5">Reserved</td><td>ETCEN</td><td>CKS3</td></tr>
<tr><td>R/W Attribute</td><td colspan="5">R0,W0</td><td>R/W</td><td>R/W</td></tr>
<tr><td>Protection Attribute</td><td colspan="7">-</td></tr>
<tr><td>Initial Value</td><td colspan="5">000000</td><td>0</td><td>0</td></tr>
</table>
</td></tr>
<tr><td>2545</td><td>CHAPTER38: Base Timer
10.1.9.3 Timer Control Register 2 (TMCR2)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit15:9] Reserved: Reserved bits
[bit8] CKS3: Count Clock Selection Bit

(Correct)
[bit7:2] Reserved: Reserved Bits

[bit1] ETCEN: External Timer Compare Enable Bit
It is the bit which enables match detection of an external timer value and the ETCDR register value.
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Match detection disable</td></tr>
<tr><td>1</td><td>Match detection enable</td></tr>
</table>

[bit0] CKS3: Count Clock Selection Bit</td></tr>
<tr><td>2546</td><td>CHAPTER38: Base Timer
10.1.9.4 Status Control Register (STC)</td><td>Features of "For details" should be deleted as indicated by the shading below.

(Error)
10.1.9.4 Status Control Register (STC)
Note: The TMCR2 register is located in the upper byte of this register.</td></tr>
</table>

Page	Section	Change Results
2548	CHAPTER38: Base Timer 10.1.9.5 Status Control Clear Register (STCC)	Features of "For details" should be deleted as indicated by the shading below. (Error) 10.1.9.5 Status Control Clear Register (STCC) Note: Reserved bits are located in the upper byte of this register.
2550	CHAPTER38: Base Timer 10.1.9.6 Status Control Set Register (STCS)	Features of "For details" should be deleted as indicated by the shading below. (Error) 10.1.9.6 Status Control Set Register (STCS) Note: Reserved bits are located in the upper byte of this register.
2553	CHAPTER38: Base Timer 10.1.12.Timer Register (TMR)	Features of "For details" should be deleted as indicated by the shading below. (Error) [bit31:17] Reserved: Reserved Bits If writing to this bit, write "0".
2556	CHAPTER38: Base Timer 10.1.15.Base Timer Debug Register (BT_DEBUG)	Added the section ” 10.1.15.Base Timer Debug Register (BT_DEBUG) "
2557	CHAPTER38: Base Timer 10.1.16.External Timer Compare Data Register (ETCDR)	Added the section ” 10.1.16.External Timer Compare Data Register (ETCDR)”

Page	Section	Change Results
2558	CHAPTER38: Base Timer 10.2.PPG Timer Function	Features of "For details" should be corrected as indicated by the shading below. (Error) 9.2.PPG Timer Function 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) (Correct) 10.2.PPG Timer Function 10.2.1 16-bit PPG Timer Operation 10.2.2 Continuous Operation 10.2.3 One-shot Operation 10.2.4 External Timer Match Starting 10.2.5 Interrupt Factors and Timing Chart 10.2.6 PPG Timer Operation Flow 10.2.7 Timer Control Registers (TMCR, TMCR2), Status Control Register (STC), Status Control Clear Register (STCC), and Status Control Set Register (STCS) 10.2.8 Low Width Setting Reload Register (PRLL) 10.2.9 High Width Setting Reload Register (PRLH) 10.2.10 Timer Register (TMR) 10.2.11 Base Timer Debug Register (BT_DEBUG) 10.2.12 External Timer Compare Data Register (ETCDR)
2559	CHAPTER38: Base Timer 10.2.1.16-Bit PPG Timer Operation	Features of "For details" should be corrected as indicated by the shading below. (Error) ***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. (Correct) ***Timing of Writing to Reload Registers*** Data is written to the reload registers PRLL and PRLH when an activation trigger is detected, and also before the next cycle change occurs following an underflow interrupt factor (UDIR) being asserted to "1".

Page	Section	Change Results
2562	CHAPTER38: Base Timer 10.2.3.One-Shot Operation	Features of "For details" should be corrected as indicated by the shading below. (Error) ***Relationship between reload value and pulse width*** 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". (Correct) ***Relationship Between Reload Value and Pulse Width*** Therefore, when the value of the reload register is 0x0000, the pulse width is equal to 1 count clock cycle. Similarly, when the value of the reload register is 0xFFFF, the pulse width is equal to 65536 count clock cycles.
2563	CHAPTER38: Base Timer 10.2.4.External Timer Match Starting	Added the section ” 10.2.4.External Timer Match Starting”
2566	CHAPTER38: Base Timer 10.2.7.Timer Control Registers (TMCR, TMCR2), Status Control Register (STC), Status Control Clear Register (STCC), and Status Control Set Register (STCS)	Features of "For details" should be deleted as indicated by the shading below. (Error) 10.2.7.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
2566	CHAPTER38: Base Timer 10.2.7.1 Timer Control Register (Upper Byte of TMCR)	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 (Correct) [TMCR2:bit0,bit14:12] CKS3 to CKS0: Count Clock Selection Bits

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>2570</td><td>CHAPTER38: Base Timer
10.2.7.3 Timer Control Register 2 (TMCR2)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
Timer control register 2 (TMCR2)
<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="7">Reserved</td><td>CKS3</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>

(Correct)
10.2.7.3 Timer Control Register 2 (TMCR2)
<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>ETCEN</td><td>CKS3</td></tr>
<tr><td>R/W Attribute</td><td colspan="6">R0,W0</td><td>R/W</td><td>R/W</td></tr>
<tr><td>Protection Attribute</td><td colspan="8">-</td></tr>
<tr><td>Initial Value</td><td colspan="6">000000</td><td>0</td><td>0</td></tr>
</table>
</td></tr>
<tr><td>2570</td><td>CHAPTER38: Base Timer
10.2.7.3 Timer Control Register 2 (TMCR2)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit15:9] Reserved: Reserved bits
[bit8] CKS3: Count clock selection bit

(Correct)
[bit7:2] Reserved: Reserved Bits
[bit1] ETCEN: External Timer Compare Enable Bit
It is the bit which enables match detection of an external timer value and the ETCDR register value.
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Match detection disable</td></tr>
<tr><td>1</td><td>Match detection enable</td></tr>
</table>

[bit0] CKS3: Count Clock Selection Bit</td></tr>
<tr><td>2573</td><td>CHAPTER38: Base Timer
10.2.7.5 Status Control Clear Register (STCC)</td><td>Features of "For details" should be deleted as indicated by the shading below.

(Error)
10.2.7.5 Status Control Clear Register (STCC)
Note: Reserved bits are located in the upper byte of this register.</td></tr>
<tr><td>2575</td><td>CHAPTER38: Base Timer
10.2.7.6 Status Control Set Register (STCS)</td><td>Features of "For details" should be deleted as indicated by the shading below.

(Error)
10.2.7.6 Status Control Set Register (STCS)
Note: Reserved bits are located in the upper byte of this register.</td></tr>
</table>

Page	Section	Change Results
2579	CHAPTER38: Base Timer 10.2.11.Base Timer Debug Register (BT_DEBUG)	Added the section ” 10.2.11.Base Timer Debug Register (BT_DEBUG)”
2580	CHAPTER38: Base Timer 10.2.12.External Timer Compare Data Register (ETCDR)	Added the section ” 10.2.12.External Timer Compare Data Register (ETCDR)”
2581	CHAPTER38: Base Timer 10.3.Reload Timer Function	Features of "For details" should be corrected as indicated by the shading below. (Error) 9.3.Reload Timer Function 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) (Correct) 10.3.Reload Timer Function 10.3.1 Operations of 16-bit Reload Timer 10.3.2 External Timer Match Starting 10.3.3 Reload Timer Operation Flow 10.3.4 Timer Control Registers (TMCR, TMCR2), Status Control Register (STC), Status Control Clear Register (STCC), and Status Control Set Register (STCS) 10.3.5 Cycle Setting Register (PCSR) 10.3.6 Timer Register (TMR) 10.3.7 Base Timer Debug Register (BT_DEBUG) 10.3.8 External Timer Compare Data Register (ETCDR)

Page	Section	Change Results
2583	CHAPTER38: Base Timer 10.3.1.Operations of 16-Bit Reload Timer	Features of "For details" should be corrected as indicated by the shading below. (Error) Operation of input pin function 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). (Correct) Operation of Input Pin Function In order to synchronize the signal that is input from the TGIN pin, the time required from the input of a valid level until count enable becomes effective is 2T to 3T at the start and end of the count operation(T: internal clock cycle).
2586	CHAPTER38: Base Timer 10.3.2.External Timer Match Starting	Added the section ” 10.3.2.External Timer Match Starting”
2589	CHAPTER38: Base Timer 10.3.4.Timer Control Registers (TMCR, TMCR2), Status Control Register (STC), Status Control Clear Register (STCC), and Status Control Set Register (STCS)	Features of "For details" should be deleted as indicated by the shading below. (Error) 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
2589	CHAPTER38: Base Timer 10.3.4.Timer Control Registers (TMCR, TMCR2), Status Control Register (STC), Status Control Clear Register (STCC), and Status Control Set Register (STCS)	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 (Correct) [TMCR2:bit0,bit14:12] CKS3 to CKS0: Count Clock Selection Bits

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>2591</td><td>CHAPTER38: Base Timer
10.3.4.3 Timer Control Register 2 (TMCR2)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
Timer control register 2 (TMCR2)
<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>GATE</td><td colspan="6">Reserved</td><td>CKS3</td></tr>
<tr><td>R/W Attribute</td><td>R/W</td><td colspan="6">R0,W0</td><td>R/W</td></tr>
<tr><td>Protection Attribute</td><td>-</td><td colspan="6">-</td><td>-</td></tr>
<tr><td>Initial Value</td><td>0</td><td colspan="6">000000</td><td>0</td></tr>
</table>

(Correct)
10.3.4.3 Timer Control Register 2 (TMCR2)
<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>GATE</td><td colspan="5">Reserved</td><td>ETCEN</td><td>CKS3</td></tr>
<tr><td>R/W Attribute</td><td>R/W</td><td colspan="5">R0,W0</td><td>R/W</td><td>R/W</td></tr>
<tr><td>Protection Attribute</td><td colspan="7">-</td></tr>
<tr><td>Initial Value</td><td>0</td><td colspan="5">00000</td><td>0</td><td>0</td></tr>
</table>
</td></tr>
<tr><td>2591</td><td>CHAPTER38: Base Timer
10.3.4.3 Timer Control Register 2 (TMCR2)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit15] GATE: Gate input enable bit
[bit14:9] Reserved: Reserved bits
[bit8] CKS3: Count clock selection bit

(Correct)
[bit7] GATE: Gate Input Enable Bit
[bit6:2] Reserved: Reserved Bits
[bit1] ETCEN: External Timer Compare Enable Bit
It is the bit which enables match detection of an external timer value and the ETCDR register value.
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Match detection disable</td></tr>
<tr><td>1</td><td>Match detection enable</td></tr>
</table>
[bit0] CKS3: Count Clock Selection Bit</td></tr>
<tr><td>2594</td><td>CHAPTER38: Base Timer
10.3.4.4 Status Control Register (STC)</td><td>Features of "For details" should be deleted as indicated by the shading below.

(Error)
10.3.4.4 Status Control Register (STC)
Note: The TMCR2 register is located in the upper byte of this register.</td></tr>
<tr><td>2596</td><td>CHAPTER38: Base Timer
10.3.4.5 Status Control Clear Register (STCC)</td><td>Features of "For details" should be deleted as indicated by the shading below.

(Error)
10.3.4.5 Status Control Clear Register (STCC)
Note: Reserved bits are located in the upper byte of this register.</td></tr>
</table>

Page	Section	Change Results
2598	CHAPTER38: Base Timer 10.3.4.6 Status Control Set Register (STCS)	Features of "For details" should be deleted as indicated by the shading below. (Error) 10.3.4.6 Status Control Set Register (STCS) Note: Reserved bits are located in the upper byte of this register.
2601	CHAPTER38: Base Timer 10.3.7.Base Timer Debug Register (BT_DEBUG)	Added the section ” 10.3.7.Base Timer Debug Register (BT_DEBUG)”
2602	CHAPTER38: Base Timer 10.3.8.External Timer Compare Data Register (ETCDR)	Added the section ” 10.3.8.External Timer Compare Data Register (ETCDR)”
2603	CHAPTER38: Base Timer 10.4.PWC Timer Function	Features of "For details" should be corrected as indicated by the shading below. (Error) 9.4.PWC Timer Function 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) (Correct) 10.4.PWC Timer Function 10.4.1 Operations of PWC Timer 10.4.2 Timer Control Registers (TMCR, TMCR2), Status Control Register (STC), Status Control Clear Register (STCC), and Status Control Set Register (STCS) 10.4.3 Data Buffer Register (DTBF) 10.4.4 Base Timer Debug Register (BT_DEBUG)
2613	CHAPTER38: Base Timer 10.4.2.Timer Control Registers (TMCR, TMCR2), Status Control Register (STC), Status Control Clear Register (STCC), and Status Control Set Register (STCS)	Features of "For details" should be deleted as indicated by the shading below. (Error) 10.4.2.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

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>2613</td><td>CHAPTER38: Base Timer
10.4.2.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

(Correct)
[TMCR2:bit0,bit14:12] CKS3 to CKS0: Count Clock Selection Bits</td></tr>
<tr><td>2615</td><td>CHAPTER 36: Base Timer
10.4.2.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)
・Timer control register (lower byte of TMCR)
<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>T32</td><td>FMD2</td><td>FMD1</td><td>FMD0</td><td>Reserved</td><td>MDSE</td><td>CTEN</td><td>Reserved</td></tr>
<tr><td>Attribute</td><td>R0,W0</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R0,W0</td><td>R/W</td><td>R/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></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)
10.4.2.2 Timer Control Register (Lower Byte of TMCR)
<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>T32</td><td>FMD2</td><td>FMD1</td><td>FMD0</td><td>OSEL</td><td>MDSE</td><td>CTEN</td><td>Reserved</td></tr>
<tr><td>Attribute</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>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></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>2616</td><td>CHAPTER 36: Base Timer
10.4.2.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 added as indicated by the shading below

(Error)
<table><tr><th>Bit</th><th>Description</th></tr><tr><td>0</td><td>Stop operation.</td></tr><tr><td>1</td><td>Enable operation.</td></tr></table>
(Correct)
<table><tr><th>Bit</th><th>Description</th></tr><tr><td>0</td><td>Stop operation.</td></tr><tr><td>1</td><td>Enable operation.</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".</td></tr>
<tr><td>2617</td><td>CHAPTER38: Base Timer
10.4.2.3 Timer Control Register 2 (TMCR2)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
Timer control register 2 (TMCR2)
<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="7">Reserved</td><td>CKS3</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>Note: This register is located in the upper byte of the STC register.
[bit15:9] Reserved: Reserved bits
[bit8] CKS3: Count clock selection bit

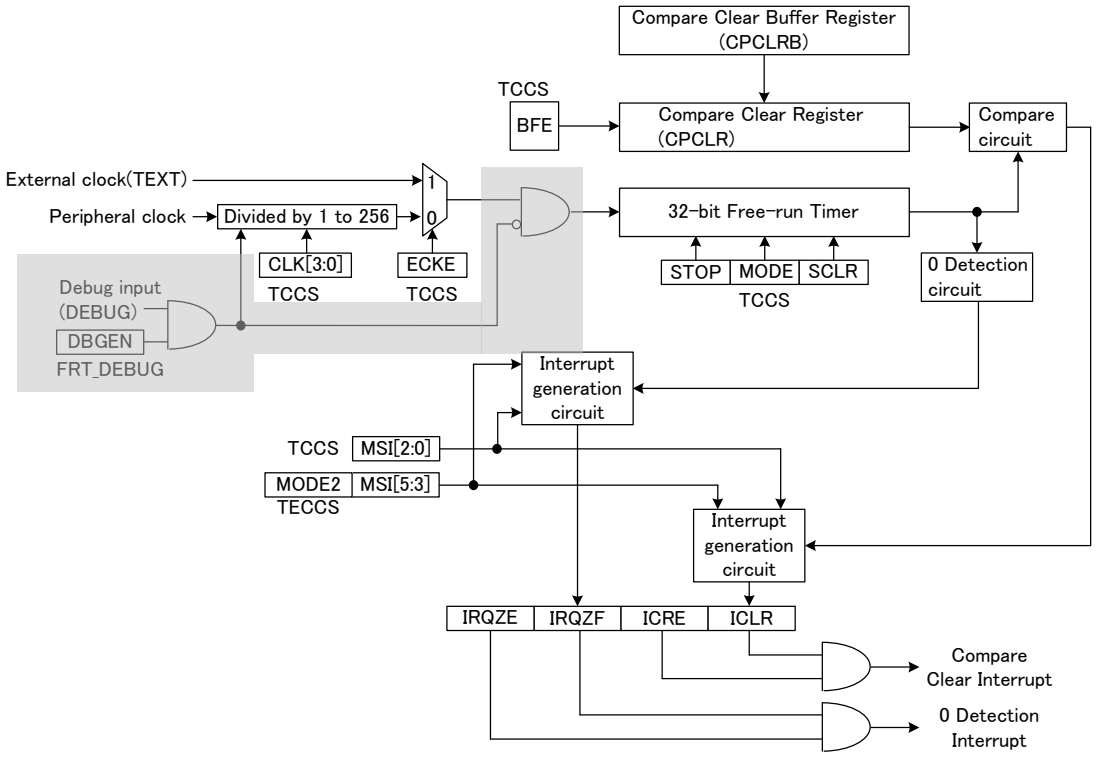
(Correct)
10.4.2.3 Timer Control Register 2 (TMCR2)
<table><tr><td>bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr><tr><td>Field</td><td colspan="7">Reserved</td><td>CKS3</td></tr><tr><td>R/W Attribute</td><td colspan="7">R0,W0</td><td>R/W</td></tr><tr><td>Protection Attribute</td><td colspan="8">-</td></tr><tr><td>Initial Value</td><td colspan="7">0000000</td><td>0</td></tr></table>[bit7:1] Reserved: Reserved Bits
[bit0] CKS3: Count Clock Selection Bit</td></tr>
<tr><td>2618</td><td>CHAPTER38: Base Timer
10.4.2.4 Status Control Register (STC)</td><td>Features of "For details" should be deleted as indicated by the shading below.

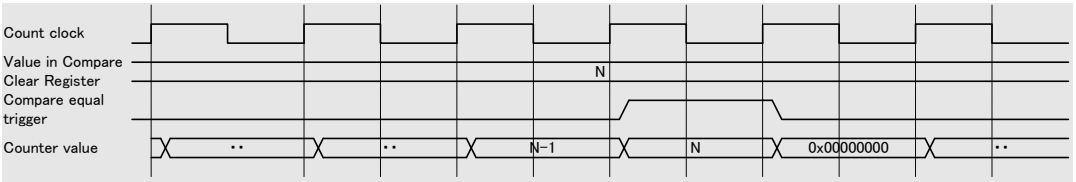
(Error)
10.4.2.4 Status Control Register (STC)
Note: The TMCR2 register is located in the upper byte of this register.</td></tr>
</table>

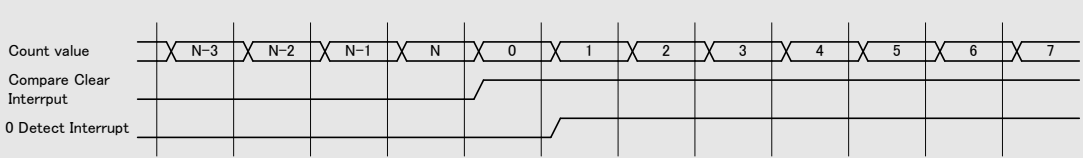
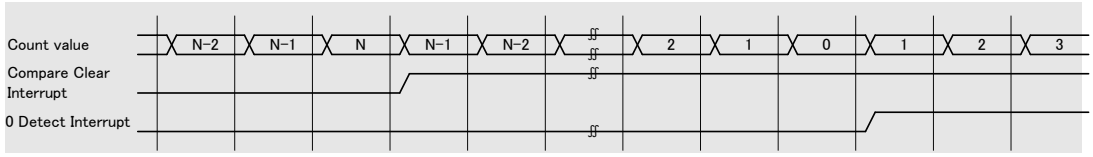
Page	Section	Change Results
2620	CHAPTER38: Base Timer 10.4.2.5 Status Control Clear Register (STCC)	Features of "For details" should be deleted as indicated by the shading below. (Error) 10.4.2.5 Status Control Clear Register (STCC) Note: Reserved bits are located in the upper byte of this register.
2621	CHAPTER38: Base Timer 10.4.2.6 Status Control Set Register (STCS)	Features of "For details" should be deleted as indicated by the shading below. (Error) 10.4.2.6 Status Control Set Register (STCS) Note: Reserved bits are located in the upper byte of this register.
2622	CHAPTER38: Base Timer 10.4.3.Data Buffer Register (DTBF)	Features of "For details" should be corrected as indicated by the shading below. (Error) 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. (Correct) 10.4.3.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, if this is an even-numbered channel the lower 16 bits will be read similarly, for odd number channels, the upper 16 bits will be read.
2623	CHAPTER38: Base Timer 10.4.4.Base Timer Debug Register (BT_DEBUG)	Added the section ” 10.4.4.Base Timer Debug Register (BT_DEBUG)”
2628	CHAPTER 39: Base Timer I/O SELECTION FUNCTION 3. Explanation of Operation	"I/O mode 2" should be corrected as indicated by the shading below (Error) I/O mode 2: External trigger 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. (Correct) 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.

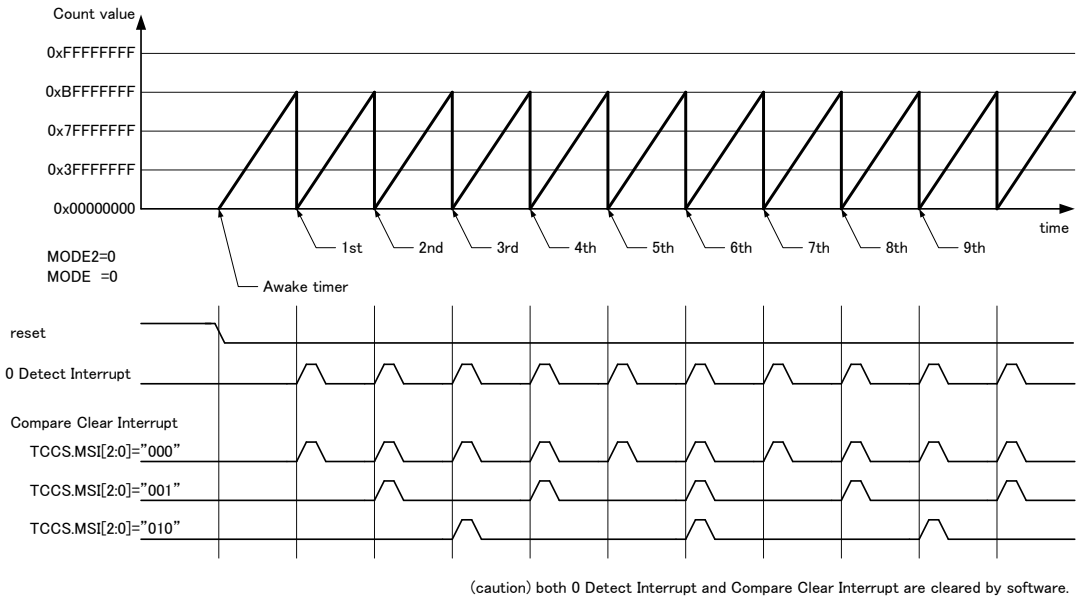
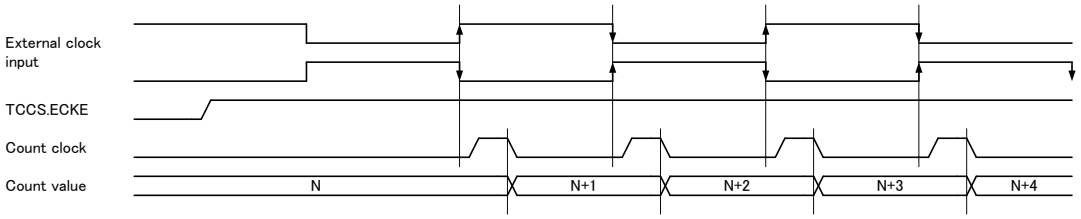
Page	Section	Change Results
2636	CHAPTER 39: Base Timer I/O SELECTION FUNCTION 4.4 Simultaneous Soft Start Register (BT_BTSSSR0)	"4.4 Simultaneous Soft Start Register (BT_BTSSSR0)" should be corrected as indicated by the shading below (Error) 4.4. Simultaneous Soft Start Register (BT_BTSSSR0) [bit11:0] 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." Writing "1" starts the corresponding channel, and writing "0" is invalid Up to 12 channels with channel numbers 0 to 11 can be started simultaneously. (Correct) 4.2. Simultaneous Soft Start Register (BT_BTSSSR0) [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." 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.

Page	Section	Change Results
2639 to 2680	CHAPTER40 Base Timer Simultaneous Operation	Added a new chapter named “CHAPTER40 Base Timer Simultaneous Operation”
2682	CHAPTER 41: 32-Bit Free-Run Timer 1. Overview of the 32-Bit Free-Run Timer	Features of "For details" should be corrected as indicated by the shading below. (Error) 1. Overview of the 32-bit Free-run Timer 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. 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. The count value is reset to "00000000H" 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. (Correct) 1. Overview of the 32-bit Free-run Timer This timer can be used with the 32-bit Input Capture and the 32-bit Output Compare. The 9 types of count 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. The count value is reset to 0x00000000 when a hardware reset occurs, 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 Free-run Timer can be used as a clock count of the 32-bit Output Compare and the 32-bit Input Capture.

Page	Section	Change Results
2683	CHAPTER 41: 32-Bit Free-Run Timer 2.Block Diagram of the 32-Bit Free-Run Timer	"Figure 2 1 Block Diagram of 32-Bit Free-Run Timer" should be modified as follows. (Correct) 

Page	Section	Change Results
2683	CHAPTER 41: 32-Bit Free-Run Timer 3. Operation of the 32-Bit Free-Run Timer	"Program Example of the 32-bit Free-run Timer." should be deleted as indicated by the shading below. (Error) 2.2. Program Example of the 32-bit Free-run Timer. Program Example of the 32-bit Free-run Timer. This section provides the program example of the 32-bit free-run timer. Program example of the free-run timer (32-bit operation) -- Processing - The compare clear interrupt is generated when the free-run timer counts every 4 ms. - This timer is used to regenerate the compare clear timer in the up count mode. - When a peripheral clock is running at 16 MHz, the count clock counts every 62.5 ns. -- Coding example ICR33 .EQU 000461H ;Compare clear of the 32-bit free-run timer 0 ;Interrupt control register TCCS .EQU 0000B8H ;Timer state control register CPCLRB .EQU 0000B0H ;Compare clear buffer register ;---------- Main program ---------- ORG C0000H START: … ;---------- Interrupt program ---------- WARI LDI #FDFFH,r1 ANDH r1,@r3 ;Clear an interrupt request flag. ; : ; User processing ; : RETI :Return from the interrupt. ;---------- Vector setting ---------- VECT .ORG FFFF8H .DATA.W WARI ; Set the interrupt routine. .ORG FFFF8H .DATA.W 0x07000000 ; Set single chip mode. .DATA.W START ; Set the reset vector. .END
2684	CHAPTER 41: 32-Bit Free-Run Timer 3.Operation of the 32-Bit Free-Run Timer	“Figure 3 1 Clear Timing of 32-Bit Free-Run Timer” should be modified as follows. (Correct) 

Page	Section	Change Results
2687	CHAPTER 41: 32-Bit Free-Run Timer 3.Operation of the 32-Bit Free-Run Timer	Features of "For details" should be corrected as indicated by the shading below. (Error) Timer interrupt Note: The 0 detection interrupt is not generated when the timer is cleared (SCLR: bit4=1 of the timer state control register (TCCS)). (Correct) Timer interrupt For Zero Detection and Compare Clear Detection, detection begins when counting is enabled on the 32-bit Free-run Timer and the count value is updated from the current value to the next value. Note: The 0 detection interrupt is not generated when the timer is cleared (the Timer Clear bit(SCLR) in the Timer State Control Register (TCCS) ="1").
2687	CHAPTER 41: 32-Bit Free-Run Timer 3.Operation of the 32-Bit Free-Run Timer	“Figure 3 6 Interrupt Generated in Up Count Mode (the Timer Count Mode Bit(MODE) in the Timer State Control Register (TCCS) ="0")” should be modified as follows. (Correct) 
2688	CHAPTER 41: 32-Bit Free-Run Timer 3.Operation of the 32-Bit Free-Run Timer	“Figure 3 7 Interrupt Generated in Up/Down Count Mode (the Timer Count Mode Bit(MODE) in the Timer State Control Register (TCCS) ="1")” should be modified as follows. (Correct) 

Page	Section	Change Results
2690	CHAPTER 41: 32-Bit Free-Run Timer 3.Operation of the 32-Bit Free-Run Timer	“Figure 3-8 Compare Clear Interrupts to Be Masked in Up Count Mode” should be modified as follows. (Correct)  (caution) both 0 Detect Interrupt and Compare Clear Interrupt are cleared by software.
2691	CHAPTER 41: 32-Bit Free-Run Timer 3.Operation of the 32-Bit Free-Run Timer	“Figure 3 11 Count Timing of 32-Bit Free-Run Timer (in Up Count Mode)” should be modified as follows. (Correct) 
2696	CHAPTER 41: 32-Bit Free-Run Timer 3.Operation of the 32-Bit Free-Run Timer	Section “3.3.Debug Mode” should be added.

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>2697</td><td>CHAPTER 41: 32-Bit Free-Run Timer
4.Registers of the 32-Bit Free-Run Timer</td><td>Features of "For details" should be added as indicated by the shading below.

(Correct)
Table 4-1 Register List of 32-Bit Free-Run Timer
<table>
<tr><th>Abbreviated Register Name</th><th>Register Name</th><th>Reference</th></tr>
<tr><td>CPCLRB/CPCLR</td><td>Compare Clear Buffer Register, Compare Clear Register</td><td>4.1</td></tr>
<tr><td>TCDT</td><td>Timer Data Register</td><td>4.2</td></tr>
<tr><td>TCCS</td><td>Timer State Control Register</td><td>4.3</td></tr>
<tr><td>TECCS</td><td>Timer Extended State Control Register</td><td>4.4</td></tr>
<tr><td>TCCSC</td><td>Timer State Control Clear Register</td><td>4.5</td></tr>
<tr><td>TCCSS</td><td>Timer State Control Set Register</td><td>4.6</td></tr>
<tr><td>FRT_DEBUG</td><td>Free-run Timer Debug Register</td><td>4.7</td></tr>
</table></td></tr>
<tr><td>2697</td><td>CHAPTER 41: 32-Bit Free-Run Timer
4.Registers of the 32-Bit Free-Run Timer</td><td>This table should be deleted as indicated by the shading below.

(Error)
Register bit locations of the 32-bit free-run timer
Table 3-2 Register Bit Locations of the 32-bit Free-run Timer
<table>
<tr><td>▪</td><td>bit31</td><td>bit30</td><td>bit29</td><td>bit28</td><td>bit27</td><td>bit26</td><td>bit25</td><td>bit24</td></tr>
<tr><td></td><td>bit23</td><td>bit22</td><td>bit21</td><td>bit20</td><td>bit19</td><td>bit18</td><td>bit17</td><td>bit16</td></tr>
<tr><td></td><td>bit15</td><td>bit14</td><td>bit13</td><td>bit12</td><td>bit11</td><td>bit10</td><td>bit9</td><td>bit8</td></tr>
<tr><td></td><td>bit7</td><td>bit6</td><td>bit5</td><td>bit4</td><td>bit3</td><td>bit2</td><td>bit1</td><td>bit0</td></tr>
<tr><td>▪CPCLRB/CPCLR</td><td>CL31</td><td>CL30</td><td>CL29</td><td>CL28</td><td>CL27</td><td>CL26</td><td>CL25</td><td>CL24</td></tr>
<tr><td></td><td>CL23</td><td>CL22</td><td>CL21</td><td>CL20</td><td>CL19</td><td>CL18</td><td>CL17</td><td>CL16</td></tr>
<tr><td></td><td>CL15</td><td>CL14</td><td>CL13</td><td>CL12</td><td>CL11</td><td>CL10</td><td>CL09</td><td>CL08</td></tr>
<tr><td></td><td>CL07</td><td>CL06</td><td>CL05</td><td>CL04</td><td>CL03</td><td>CL02</td><td>CL01</td><td>CL00</td></tr>
<tr><td>▪TCDT</td><td>T31</td><td>T30</td><td>T29</td><td>T28</td><td>T27</td><td>T26</td><td>T25</td><td>T24</td></tr>
<tr><td></td><td>T23</td><td>T22</td><td>T21</td><td>T20</td><td>T19</td><td>T18</td><td>T17</td><td>T16</td></tr>
<tr><td></td><td>T15</td><td>T14</td><td>T13</td><td>T12</td><td>T11</td><td>T10</td><td>T09</td><td>T08</td></tr>
<tr><td></td><td>T07</td><td>T06</td><td>T05</td><td>T04</td><td>T03</td><td>T02</td><td>T01</td><td>T00</td></tr>
<tr><td>▪TCCS</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></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></td><td>ECKE</td><td>IRQZF</td><td>IRQZE</td><td>MSI2</td><td>MSI1</td><td>MSI0</td><td>ICLR</td><td>ICRE</td></tr>
<tr><td></td><td>BFE</td><td>STOP</td><td>MODE</td><td>SCLR</td><td>CLK3</td><td>CLK2</td><td>CLK1</td><td>CLK0</td></tr>
<tr><td>▪TECCS</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></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></td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>MODE2</td><td>MSI5</td><td>MSI4</td><td>MSI3</td></tr>
<tr><td></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>▪TCCSC</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></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></td><td>ECKEC</td><td>IRQZFC</td><td>IRQZEC</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>ICLRC</td><td>ICREC</td></tr>
<tr><td></td><td>BFEC</td><td>STOPC</td><td>MODEC</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>▪TCCSS</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></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></td><td>ECKES</td><td>Reserved</td><td>IRQZES</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>ICRES</td></tr>
<tr><td></td><td>BFES</td><td>STOPS</td><td>MODES</td><td>SCLRS</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>2700</td><td>CHAPTER 41: 32-Bit Free-Run Timer
4.2.Timer Data Register (TCDT)</td><td>Features of "For details" should be added as indicated by the shading below.

(Correct)
[bit31:0] T[31:0]: Timer Data Value Bits
The Timer Data Register (TCDT) is set to 0x00000000 while the 32-bit Free-run Timer is stopped (the Timer Enable bit(STOP) in the Timer State Control Register (TCCS) ="1").

If the Timer Clear bit (SCLR) in the Timer State Control Register (TCCS) is set to "1" or the value of the Compare Clear Register (CPCLR) matches the timer count value, the counter is cleared in synchronization with the count timing.</td></tr>
<tr><td>2702</td><td>CHAPTER 41: 32-Bit Free-Run Timer
4.3.Timer State Control Register (TCCS)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
3.3.Timer State Control Register (TCCS)
<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">R1,WX</td></tr><tr><td>Protection Attribute</td><td colspan="8">-</td></tr><tr><td>Initial Value</td><td colspan="8">11111111</td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><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">R1,WX</td></tr><tr><td>Protection Attribute</td><td colspan="8">-</td></tr><tr><td>Initial Value</td><td colspan="8">11111111</td></tr></table>[bit31:16] Reserved bits
"1" is always read from this bit..

(Correct)
4.3.Timer State Control Register (TCCS)
<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,WX</td></tr><tr><td>Protection Attribute</td><td colspan="8">-</td></tr><tr><td>Initial Value</td><td colspan="8">XXXXXXXX</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,WX</td></tr><tr><td>Protection Attribute</td><td colspan="8">-</td></tr><tr><td>Initial Value</td><td colspan="8">XXXXXXXX</td></tr></table>[bit31:16] Reserved Bits
Reading these bits returns an undefined value.</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>2702</td><td>CHAPTER 41: 32-Bit Free-Run Timer
4. Registers of the 32-bit Free-run Timer
4.3. Timer State Control Register (TCCS)</td><td>"4.3. Timer State Control Register (TCCS)" should be corrected as indicated by the shading below.

(Error)
4.3. Timer State Control Register (TCCS)
<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>ECKE</td><td>IRQZF</td><td>IRQZE</td><td>MSI2</td><td>MSI1</td><td>MSI0</td><td>ICLR</td><td>ICRE</td></tr>
<tr><td>Attribute</td><td>R/W</td><td>R</td><td>R/W</td><td>R,W</td><td>R,W</td><td>R,W</td><td>R</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)
4.3. Timer State Control Register (TCCS)
<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>ECKE</td><td>IRQZF</td><td>IRQZE</td><td>MSI2</td><td>MSI1</td><td>MSI0</td><td>ICLR</td><td>ICRE</td></tr>
<tr><td>Attribute</td><td>R/W</td><td>R,WX</td><td>R/W</td><td>R,W</td><td>R,W</td><td>R,W</td><td>R,WX</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>
<tr><td>2706, 2707</td><td>CHAPTER 41: 32-Bit Free-Run Timer
4.3.Timer State Control Register (TCCS)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
[bit4] SCLR: Timer clear bit
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.

(Correct)
[bit4] SCLR: Timer Clear Bit
If this bit is set to "1" while the 32-bit Free-run Timer is operating (the Timer Enable bit (STOP) = "0"), the 32-bit Free-run Timer is initialized and then starts counting up.
If this bit is set to "1" while the 32-bit Free-run Timer is stopped (the Timer Enable bit (STOP) = "1") and then the 32-bit Free-run Timer is restarted (the Timer Enable bit (STOP) = "0"), the 32-bit Free-run Timer always counts up when starting operation.
Suppose the 32-bit Free-run Timer stops operating (the Timer Enable bit (STOP) = "1") while counting down and this bit is set to "1". After that, even if the 32-bit Free-run Timer resumes operation (the Timer Enable bit (STOP) = "1"), the 32-bit Free-run Timer counts up when starting operation.</td></tr>
</table>

Page	Section	Change Results
2708	CHAPTER 41: 32-Bit Free-Run Timer 4.4.Timer Extended State Control Register (TECCS)	Features of "For details" should be corrected as indicated by the shading below. (Error) 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 (Correct) bit 31 30 29 28 27 26 25 24 Field Reserved R/W Attribute RX,WX Protection Attribute - Initial Value XXXXXXXX bit 23 22 21 20 19 18 17 16 Field Reserved R/W Attribute RX,WX Protection Attribute - Initial Value XXXXXXXX bit 15 14 13 12 11 10 9 8 Field Reserved MODE2 MSI[5:3] R/W Attribute RX,WX R/W R,W Protection Attribute - Initial Value XXXX 0 000 bit 7 6 5 4 3 2 1 0 Field Reserved R/W Attribute RX,WX Protection Attribute - Initial Value XXXXXXXX

Page	Section	Change Results
2708, 2710	CHAPTER 41: 32-Bit Free-Run Timer 4.4.Timer Extended State Control Register (TECCS)	Features of "For details" should be corrected as indicated by the shading below. (Error) [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. [bit7:0] Reserved: Reserved bits "1" is always read from this bit. Writing data to these bits has no effect on the operation. (Correct) [bit31:12] Reserved: Reserved Bits Reading these bits returns an undefined value. Writing data to these bits has no effect on the operation. [bit7:0] Reserved: Reserved Bits Reading these bits returns an undefined value. Writing data to these bits has no effect on the operation.

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>2711, 2712</td><td>CHAPTER 41: 32-Bit Free-Run Timer 4.5.Timer State Control Clear Register (TCCSC)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[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.
<table>
<tr><th>bit</th><th>Description</th></tr>
<tr><td>0</td><td>Affect neither this bit nor the clock selection bit (ECKE) of the timer state control register (TCCS).</td></tr>
<tr><td>1</td><td>Clear the clock selection bit (ECKE) of the timer state control register (TCCS).</td></tr>
</table>
[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.
<table>
<tr><th>bit</th><th>Description</th></tr>
<tr><td>0</td><td>Affect neither this bit nor the 0 detection interrupt flag bit (IRQZF) of the timer state control register (TCCS).</td></tr>
<tr><td>1</td><td>Clear the 0 detection interrupt flag bit (IRQZF) of the timer state control register (TCCS).</td></tr>
</table>

(Correct)
[bit31:16] Reserved: Reserved Bits
Reading these bits returns "0".
Always write "0" to these bits.
[bit15] ECKEC: Clock Selection Clear Bit
It is the bit which sets the Clock Selection bit (ECKE) in the Timer State Control Register (TCCS) to "0".
<table>
<tr><th rowspan="2">Bit</th><th colspan="2">Description</th></tr>
<tr><th>Writing</th><th>Reading</th></tr>
<tr><td>0</td><td>No effect.</td><td rowspan="2">Always read "0".</td></tr>
<tr><td>1</td><td>Set the Clock Selection bit (ECKE) in the Timer State Control Register (TCCS) to "0".</td></tr>
</table>
[bit14] IRQZFC: 0 Detection Interrupt Flag Clear
It is the bit which sets the 0 Detection Interrupt Flag (IRQZF) in the Timer State Control Register (TCCS) to "0".
<table>
<tr><th rowspan="2">Bit</th><th colspan="2">Description</th></tr>
<tr><th>Writing</th><th>Reading</th></tr>
<tr><td>0</td><td>No effect.</td><td rowspan="2">Always read "0".</td></tr>
<tr><td>1</td><td>Set the 0 Detection Interrupt Flag (IRQZF) in the Timer State Control Register (TCCS) to "0".</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>2712</td><td>CHAPTER 41: 32-Bit Free-Run Timer
4.5.Timer State Control Clear Register (TCCSC)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[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.
<table>
<tr><th>bit</th><th>Description</th></tr>
<tr><td>0</td><td>Affect neither this bit nor the 0 detection request enable bit (IRQZE) of the timer state control register (TCCS).</td></tr>
<tr><td>1</td><td>Clear the 0 detection request enable bit (IRQZE) of the timer state control register (TCCS).</td></tr>
</table>
[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.
<table>
<tr><th>bit</th><th>Description</th></tr>
<tr><td>0</td><td>Affect neither this bit nor the compare clear interrupt flag bit (ICLR) of the timer state control register (TCCS).</td></tr>
<tr><td>1</td><td>Clear the compare clear interrupt flag bit (ICLR) of the timer state control register (TCCS).</td></tr>
</table>

(Correct)
[bit13] IRQZEC: 0 Detection Request Enable Clear Bit
It is the bit which sets the 0 Detection Request Enable bit (IRQZE) in the Timer State Control Register (TCCS) to "0".
<table>
<tr><th rowspan="2">Bit</th><th colspan="2">Description</th></tr>
<tr><th>Writing</th><th>Reading</th></tr>
<tr><td>0</td><td>No effect.</td><td rowspan="2">Always read "0".</td></tr>
<tr><td>1</td><td>Set the 0 Detection Request Enable bit (IRQZE) in the Timer State Control Register (TCCS) to "0".</td></tr>
</table>
[bit12:10] Reserved: Reserved Bits
Reading these bits returns "0".
Always write "0" to these bits.
[bit9] ICLRC: Compare Clear Interrupt Flag Clear
It is the bit which sets the Compare Clear Interrupt Flag (ICLR) in the Timer State Control Register (TCCS) to "0".
<table>
<tr><th rowspan="2">Bit</th><th colspan="2">Description</th></tr>
<tr><th>Writing</th><th>Reading</th></tr>
<tr><td>0</td><td>No effect.</td><td rowspan="2">Always read "0".</td></tr>
<tr><td>1</td><td>Set the Compare Clear Interrupt Flag (ICLR) in the Timer State Control Register (TCCS) to "0".</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>2712, 2713</td><td>CHAPTER 41: 32-Bit Free-Run Timer 4.5.Timer State Control Clear Register (TCCSC)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[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.
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Affect neither this bit nor the compare clear interrupt request enable bit (ICRE) of the timer state control register (TCCS).</td></tr>
<tr><td>1</td><td>Clear the compare clear interrupt request enable bit (ICRE) of the timer state control register (TCCS).</td></tr>
</table>
[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.
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Affect neither this bit nor the compare clear buffer enable bit (BFE) of the timer state control register (TCCS).</td></tr>
<tr><td>1</td><td>Clear the compare clear buffer enable bit (BFE) of the timer state control register (TCCS).</td></tr>
</table>

(Correct)
[bit8] ICREC: Compare Clear Interrupt Request Enable Clear Bit
It is the bit which sets the Compare Clear Interrupt Request Enable bit (ICRE) in the Timer State Control Register (TCCS) to "0".
<table>
<tr><th rowspan="2">Bit</th><th colspan="2">Description</th></tr>
<tr><th>Writing</th><th>Reading</th></tr>
<tr><td>0</td><td>No effect.</td><td rowspan="2">Always read "0".</td></tr>
<tr><td>1</td><td>Set the Compare Clear Interrupt Request Enable bit (ICRE) in the Timer State Control Register (TCCS) to "0".</td></tr>
</table>
[bit7] BFEC: Compare Clear Buffer Enable Clear Bit
It is the bit which sets the Compare Clear Buffer Enable bit (BFE) in the Timer State Control Register (TCCS) to "0".
<table>
<tr><th rowspan="2">Bit</th><th colspan="2">Description</th></tr>
<tr><th>Writing</th><th>Reading</th></tr>
<tr><td>0</td><td>No effect.</td><td rowspan="2">Always read "0".</td></tr>
<tr><td>1</td><td>Set the Compare Clear Buffer Enable bit (BFE) in the Timer State Control Register (TCCS) to "0".</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>2713</td><td>CHAPTER 41: 32-Bit Free-Run Timer
4.5.Timer State Control Clear Register (TCCSC)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[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.
<table>
<tr><th>bit</th><th>Description</th></tr>
<tr><td>0</td><td>Affect neither this bit nor the timer enable bit (STOP) of the timer state control register (TCCS).</td></tr>
<tr><td>1</td><td>Clear the timer enable bit (STOP) of the timer state control register (TCCS).</td></tr>
</table>
[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.
<table>
<tr><th>bit</th><th>Description</th></tr>
<tr><td>0</td><td>Affect neither this bit nor the timer count mode bit (MODE) of the timer state control register (TCCS).</td></tr>
<tr><td>1</td><td>Clear the timer count mode bit (MODE) of the timer state control register (TCCS).</td></tr>
</table>
[bit4:0] Reserved: Reserved bits
"0" is always read from this bit. Always write "0" to this bit.

(Correct)
[bit6] STOPC: Timer Enable Clear Bit
It is the bit which sets the Timer Enable bit (STOP) in the Timer State Control Register (TCCS) to "0".
<table>
<tr><th rowspan="2">Bit</th><th colspan="2">Description</th></tr>
<tr><th>Writing</th><th>Reading</th></tr>
<tr><td>0</td><td>No effect.</td><td rowspan="2">Always read "0".</td></tr>
<tr><td>1</td><td>Set the Timer Enable bit (STOP) in the Timer State Control Register (TCCS) to "0".</td></tr>
</table>
[bit5] MODEC: Timer Count Mode Clear Bit
It is the bit which sets the Timer Count Mode bit (MODE) in the Timer State Control Register (TCCS) to "0".
<table>
<tr><th rowspan="2">Bit</th><th colspan="2">Description</th></tr>
<tr><th>Writing</th><th>Reading</th></tr>
<tr><td>0</td><td>No effect.</td><td rowspan="2">Always read "0".</td></tr>
<tr><td>1</td><td>Set the Timer Count Mode bit (MODE) in the Timer State Control Register (TCCS) to "0".</td></tr>
</table>
[bit4:0] Reserved: Reserved Bits
Reading these bits returns "0".
Always write "0" to these bits.
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>2714</td><td>CHAPTER 41: 32-Bit Free-Run Timer
4 Registers of the 32-bit Free-run Timer
4.6. Timer State Control Set register (TCCSS)</td><td>"4.6. Timer State Control Set register (TCCSS)" should be corrected as indicated by the shading below.

(Error)
3.6. Timer State Control Set register (TCCSS)
<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>ECKES</td><td>Reserved</td><td>IRQZES</td><td colspan="4">Reserved</td><td>ICRES</td></tr>
<tr><td>Attribute</td><td>R0,W</td><td>R0,W</td><td>R0,W</td><td colspan="4">R0,W0</td><td>R0,W</td></tr>
<tr><td>Protection Attribute</td><td>-</td><td>-</td><td>-</td><td colspan="4">-</td><td>-</td></tr>
<tr><td>Initial Value</td><td>0</td><td>-</td><td>0</td><td colspan="4">0000</td><td>0</td></tr>
</table>
(Correct)
4.6. Timer State Control Set register (TCCSS)
<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>ECKES</td><td>Reserved</td><td>IRQZES</td><td colspan="4">Reserved</td><td>ICRES</td></tr>
<tr><td>Attribute</td><td>R0,W</td><td>R0,W0</td><td>R0,W</td><td colspan="4">R0,W0</td><td>R0,W</td></tr>
<tr><td>Protection Attribute</td><td>-</td><td>-</td><td>-</td><td colspan="4">-</td><td>-</td></tr>
<tr><td>Initial Value</td><td>0</td><td>0</td><td>0</td><td colspan="4">0000</td><td>0</td></tr>
</table></td></tr>
<tr><td>2714</td><td>CHAPTER 41: 32-Bit Free-Run Timer
4.6.Timer State Control Set Register (TCCSS)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
3.6.Timer State Control Set register (TCCSS)
<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>BFES</td><td>STOPS</td><td>MODES</td><td>SCLRS</td><td colspan="4">Reserved</td></tr>
<tr><td>R/W Attribute</td><td>R0,W</td><td>R0,W</td><td>R0,W</td><td>R0,W</td><td colspan="4">R0,W0</td></tr>
<tr><td>Protection Attribute</td><td>-</td><td>-</td><td>-</td><td>-</td><td colspan="4">-</td></tr>
<tr><td>Initial Value</td><td>0</td><td>0</td><td>0</td><td>0</td><td colspan="4">0000</td></tr>
</table>
[bit31:16] Reserved: Reserved bits
"0" is always read from this bit.
Always write "0" to this bit.

(Correct)
4.6.Timer State Control Set Register (TCCSS)
<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>BFES</td><td>STOPS</td><td>MODES</td><td>SCLRS</td><td colspan="4">Reserved</td></tr>
<tr><td>R/W Attribute</td><td>R0,W</td><td>R0,W</td><td>R0,W</td><td>R0,W</td><td colspan="4">R0,WX</td></tr>
<tr><td>Protection Attribute</td><td colspan="8">-</td></tr>
<tr><td>Initial Value</td><td>0</td><td>0</td><td>0</td><td>0</td><td colspan="4">0000</td></tr>
</table>
[bit31:16] Reserved: Reserved Bits
Reading these bits returns "0".
Always write "0" to these bits.</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>2714, 2715</td><td>CHAPTER 41: 32-Bit Free-Run Timer 4.6.Timer State Control Set Register (TCCSS)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[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.
<table>
<tr><th>bit</th><th>Description</th></tr>
<tr><td>0</td><td>Affect neither this bit nor the clock selection bit (ECKE) of the timer state control register (TCCS).</td></tr>
<tr><td>1</td><td>Set the clock selection bit (ECKE) of the timer state control register (TCCS).</td></tr>
</table>
[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.
<table>
<tr><th>bit</th><th>Description</th></tr>
<tr><td>0</td><td>Affect neither this bit nor the 0 detection request enable bit (IRQZE) of the timer state control register (TCCS).</td></tr>
<tr><td>1</td><td>Set the 0 detection request enable bit (IRQZE) of the timer state control register (TCCS).</td></tr>
</table>
[bit12:9] Reserved: Reserved bits
"0" is always read from this bit.
Always write "0" to this bit.

(Correct)
[bit15] ECKES: Clock Selection Set Bit
It is the bit which sets the Clock Selection bit (ECKE) in the Timer State Control Register (TCCS) to "1".
<table>
<tr><th rowspan="2">Bit</th><th colspan="2">Description</th></tr>
<tr><th>Writing</th><th>Reading</th></tr>
<tr><td>0</td><td>No effect.</td><td rowspan="2">Always read "0".</td></tr>
<tr><td>1</td><td>Set the Clock Selection bit (ECKE) in the Timer State Control Register (TCCS) to "1".</td></tr>
</table>
[bit14] Reserved: Reserved Bit
Reading this bit returns "0".
Always write "0" to this bit.
[bit13] IRQZES: 0 Detection Request Enable Set Bit
It is the bit which sets the 0 Detection Request Enable bit (IRQZE) in the Timer State Control Register (TCCS) to "1".
<table>
<tr><th rowspan="2">Bit</th><th colspan="2">Description</th></tr>
<tr><th>Writing</th><th>Reading</th></tr>
<tr><td>0</td><td>No effect.</td><td rowspan="2">Always read "0".</td></tr>
<tr><td>1</td><td>Set the 0 Detection Request Enable bit (IRQZE) in the Timer State Control Register (TCCS) to "1".</td></tr>
</table>
[bit12:9] Reserved: Reserved Bits
Reading these bits returns "0".
Always write "0" to these bits.
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>2715</td><td>CHAPTER 41: 32-Bit Free-Run Timer
4.6.Timer State Control Set Register (TCCSS)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[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.
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Affect neither this bit nor the compare clear interrupt request enable bit (ICRE) of the timer state control register (TCCS).</td></tr>
<tr><td>1</td><td>Set the compare clear interrupt request enable bit (ICRE) of the timer state control register (TCCS).</td></tr>
</table>
[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.
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Affect neither this bit nor the compare clear buffer enable bit (BFE) of the timer state control register (TCCS).</td></tr>
<tr><td>1</td><td>Set the compare clear buffer enable bit (BFE) of the timer state control register (TCCS).</td></tr>
</table>

(Correct)
[bit8] ICRES: Compare Clear Interrupt Request Enable Set Bit
It is the bit which sets the Compare Clear Interrupt Request Enable bit (ICRE) in the Timer State Control Register (TCCS) to "1".
<table>
<tr><th rowspan="2">Bit</th><th colspan="2">Description</th></tr>
<tr><th>Writing</th><th>Reading</th></tr>
<tr><td>0</td><td>No effect.</td><td rowspan="2">Always read "0".</td></tr>
<tr><td>1</td><td>Set the Compare Clear Interrupt Request Enable bit (ICRE) in the Timer State Control Register (TCCS) to "1".</td></tr>
</table>
[bit7] BFES: Compare Clear Buffer Enable Set Bit
It is the bit which sets the Compare Clear Buffer Enable bit (BFE) in the Timer State Control Register (TCCS) to "1".
<table>
<tr><th rowspan="2">Bit</th><th colspan="2">Description</th></tr>
<tr><th>Writing</th><th>Reading</th></tr>
<tr><td>0</td><td>No effect.</td><td rowspan="2">Always read "0".</td></tr>
<tr><td>1</td><td>Set the Compare Clear Buffer Enable bit (BFE) in the Timer State Control Register (TCCS) to "1".</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>2716</td><td>CHAPTER 41: 32-Bit Free-Run Timer
4.6.Timer State Control Set Register (TCCSS)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[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.
<table>
<tr><th>bit</th><th>Description</th></tr>
<tr><td>0</td><td>Affect neither this bit nor the timer enable bit (STOP) of the timer state control register (TCCS).</td></tr>
<tr><td>1</td><td>Set the timer enable bit (STOP) of the timer state control register (TCCS).</td></tr>
</table>
[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.
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Affect neither this bit nor the timer count mode bit (MODE) of the timer state control register (TCCS).</td></tr>
<tr><td>1</td><td>Set the timer count mode bit (MODE) of the timer state control register (TCCS).</td></tr>
</table>

(Correct)
[bit6] STOPS: Timer Enable Set Bit
It is the bit which sets the Timer Enable bit (STOP) in the Timer State Control Register (TCCS) to "1".
<table>
<tr><th rowspan="2">Bit</th><th colspan="2">Description</th></tr>
<tr><th>Writing</th><th>Reading</th></tr>
<tr><td>0</td><td>No effect.</td><td rowspan="2">Always read "0".</td></tr>
<tr><td>1</td><td>Set the Timer Enable bit (STOP) in the Timer State Control Register (TCCS) to "1".</td></tr>
</table>
[bit5] MODES: Timer Count Mode Set Bit
It is the bit which sets the Timer Count Mode bit (MODE) in the Timer State Control Register (TCCS) to "1".
<table>
<tr><th rowspan="2">Bit</th><th colspan="2">Description</th></tr>
<tr><th>Writing</th><th>Reading</th></tr>
<tr><td>0</td><td>No effect.</td><td rowspan="2">Always read "0".</td></tr>
<tr><td>1</td><td>Set the Timer Count Mode bit (MODE) in the Timer State Control Register (TCCS) to "1".</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>2716</td><td>CHAPTER4 1:32-Bit Free-Run Timer 4.6.Timer State Control Set Register (TCCSS)</td><td>
Features of "For details" should be corrected as indicated by the shading below.

(Error)
[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.
<table>
<tr><th>bit</th><th>Description</th></tr>
<tr><td>0</td><td>Affect neither this bit nor the timer clear bit (SCLR) of the timer state control register (TCCS).</td></tr>
<tr><td>1</td><td>Set the timer clear bit (SCLR) of the timer state control register (TCCS).</td></tr>
</table>
[bit3:0] Reserved: Reserved bits
"0" is always read from this bit.
Always write "0" to this bit.

(Correct)
[bit4] SCLRS: Timer Clear Set Bit
It is the bit which sets the Timer Clear bit (SCLR) in the Timer State Control Register (TCCS) to "1".
<table>
<tr><th rowspan="2">Bit</th><th colspan="2">Description</th></tr>
<tr><th>Writing</th><th>Reading</th></tr>
<tr><td>0</td><td>No effect.</td><td rowspan="2">Always read "0".</td></tr>
<tr><td>1</td><td>Set the Timer Clear bit (SCLR) in the Timer State Control Register (TCCS) to "1".</td></tr>
</table>
[bit3:0] Reserved: Reserved Bits
Reading these bits returns "0".
Always write "0" to these bits.
</td></tr>
<tr><td>2717</td><td>CHAPTER4 1:32-Bit Free-Run Timer 4.Registers of the 32-Bit Free-Run Timer 4.7.Free-Run Timer Debug Register(FRT_DEBUG)</td><td>“4.7.Free-Run Timer Debug Register(FRT_DEBUG)” newly should be added.</td></tr>
</table>

Page	Section	Change Results
2718	CHAPTER41:32-Bit Free-Run Timer 5.Precautions for Using This Device	Features of "For details" should be corrected as indicated by the shading below. (Error) When accessing the timer state control register (TCCS) This register supports writing from the bit-band alias area. For the bit-band alias area, see "CHAPTER: BIT-BAND UNIT." [x] In the normal reading mode, the interrupt mask counter value is read from MSI2 to MSI0. 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. When setting using a program When the hardware is reset, the count value becomes "00000000H", but the 0 detection interrupt flag is not 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 "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 Accessing the Timer State Control Register (TCCS) 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). When Accessing the Timer Extended State Control Register (TECCS) In the normal reading mode, the interrupt mask counter value is read from Compare Clear Interrupt Mask Selection bits(MSI[5:3]) in the Timer Extended State Control Register (TECCS). When Setting Using a Program When the hardware is reset, the count value becomes 0x00000000, but the 0 Detection Interrupt Flag (IRQZF) in the Timer State Control Register (TCCS) is not set. When the timer mode bit (the Timer Count Mode bit(MODE) in the Timer State Control Register (TCCS)) is in the up count mode (the Timer Count Mode bit(MODE) in the Timer State Control Register (TCCS) ="0"), the count value is updated to 0x000000000 and fixed to 0x000000000. Then, the 0 Detection Interrupt Flag (IRQZF) in the Timer State Control Register (TCCS) and the Compare Clear Interrupt Flag (ICLR) in the Timer State Control Register (TCCS) are set at every count clock. When the Timer Count Mode bit (MODE) in the Timer State Control Register (TCCS) is in the up/down count mode (Timer Count Mode bit(MODE) in the Timer State Control Register (TCCS) ="1"), the count value is counted up from 0x00000000 to 0xFFFFFFFF, and this up count operation is repeated. The 0 Detection Interrupt Flag (IRQZF) in the Timer State Control Register (TCCS) and the Compare Clear Interrupt Flag (ICLR) in the Timer State Control Register (TCCS) are set to "1" when the count value becomes 0x000000000.

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>2729</td><td>CHAPTER 42: 32-Bit Input Capture
3. Registers of the 32-bit Input Capture
3.2. Input Capture State Control Register (ICS)</td><td>"3.2. Input Capture State Control Register (ICS)" should be added as indicated by the shading below.

(Error)
3.2. Input Capture State Control Register (ICS)
For details on writing to this register, see "4. Precautions for Using This Device."

(Correct)
3.2. Input Capture State Control Register (ICS)
For details on writing to this register, see "4. Precautions for Using This Device."
Flag bits of the input capture state control registers are only read if APB-IF is invalid.</td></tr>
<tr><td>2909</td><td>CHAPTER50:I/O Port
4. Register List</td><td>Features of "For details" should be corrected as indicated by the shading below
(Error)
n: Number of selectable resource inputs (n = 0 to 188)
(Correct)
n: Number of selectable resource inputs (n = 0 to 367)</td></tr>
<tr><td>2922</td><td>CHAPTER50:I/O Port
4.10. Port Setting Register (PPC_PCFGRijj) (i = 0 to 4, jj = 00 to 31)</td><td>Revised the table of "External interrupt(INT)" in "[bit5] NFE" as follows:

(Corrected)

<table>
<tr><th>Resource.</th><th>PPC_PCFGR:NFE</th><th>EICxx_NFER:NFEx</th><th>RIC_RESINyyy:RESSELz (yyy=350 to 367)</th></tr>
<tr><td>GPIO</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td>Free run timer(TEXT)</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td>Input Capture(IN)</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td>Base Timer(TIOA,TIOB)</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td>External interrupt(INT)</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>I2C INTERFACE</td><td>0</td><td>0</td><td>1</td></tr>
</table>

Cf. Allotment of the terminal : function list of terminals [Summary]
Cf. EICxx_NFER:NFEx : external interrupt chapter
Cf. RIC_RESINyyy:RESSELz : various settings of I/O ports chapter</td></tr>
<tr><td>2926</td><td>CHAPTER50:I/O Port
4.12 Resource Input Setting Register</td><td>Features of "For details" should be corrected as indicated by the shading below
(Error)
Resource Input Setting Register (RIC_RESINn) (n = 0 to 188)
(Correct)
Resource Input Setting Register (RIC_RESINn) (n = 0 to 367)</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>3133</td><td>CHAPTER55: Bit-Band Unit
2.Explanation of Operation</td><td>Added ” Details of Address Correspondence” as new item.
(Correct)

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)
<table>
<tr><th>Register Address of Bit-Band Area (SYSC1 Group)</th><th>Bit Position</th><th>Register Address of Bit-Band Alias Single Bit Manipulation Area</th></tr>
<tr><td>0xB0300000 (2)</td><td>bit[0]</td><td>0xB1000000 (1)</td></tr>
<tr><td></td><td>bit[1]</td><td>0xB1000001</td></tr>
<tr><td></td><td>bit[2]</td><td>0xB1000002</td></tr>
<tr><td></td><td>...</td><td></td></tr>
<tr><td>0xB0300010 (3)</td><td>bit[0]</td><td>0xB1000080</td></tr>
<tr><td></td><td>bit[1]</td><td>0xB1000081</td></tr>
<tr><td></td><td>bit[2]</td><td>0xB1000082</td></tr>
<tr><td></td><td>bit[3] (4)</td><td>0xB1000083 (=(1)+8*((3)-(2))+(4))</td></tr>
<tr><td></td><td>...</td><td></td></tr>
</table>
</td></tr>
<tr><td>3241 to 3260</td><td>CHAPTER56:12-Bit A/D Converter
5.2.8. PWM Timer Activation</td><td>Added a table named “Table 5-12 Connection Relationship between PWM Timers and A/D Converter Activation Channels”</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>3261,
3262</td><td>CHAPTER56:12-Bit A/D Converter
5.2.8. PWM Timer Activation</td><td>Added a table named “Table 5-13 Connection Relationship between Basetimer Timers and A/D Converter Activation Channels”</td></tr>
<tr><td>3310</td><td>CHAPTER57:LCD Controller
4.1. Operation of LCD Controller/Driver (LCDC)</td><td>The description should be corrected as indicated by the shading below.
(Error)
(When the bit is set to "1", a selection waveform is output from the segment output pin.
When the bit is set to "0", a non-selection waveform is output from the segment output pin.)
(Correct)
(When the bit of VRAM is set to "1", a selection waveform is output from the segment output pin.
When the bit of VRAM is set to "0", a non-selection waveform is output from the segment output pin.)</td></tr>
<tr><td>3336</td><td>CHAPTER57:LCD Controller
6.6. Static LCD Display Data Register: LDR</td><td>The description should be corrected as indicated by the shading below.
(Error)
LDR1
(Correct)
LDR1 : Address 05BDH (Access : Byte, Half-word, Word)</td></tr>
<tr><td>3341</td><td>CHAPTER57:LCD Controller
6.9. Common Output V Pin Control Register: COMVER</td><td>The description 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>VE3</td><td>VE2</td><td>VE1</td><td>VE0</td><td>COME3</td><td>COME2</td><td>COME1</td><td>COME0</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 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>VE3</td><td>VE2</td><td>VE1</td><td>VE0</td><td>COME3</td><td>COME2</td><td>COME1</td><td>COME0</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 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>1</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>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>3344,
3345</td><td>CHAPTER57:LCD Controller
7.1. How Can I Set Pins to COM Output Pins or SEG Output Pins?</td><td>The description in the table of “<Duty drive>” should be corrected as indicated by the shading below.
(Error)
Pin setting method
(Correct)
V0-V3 and COM and SEG Setting Method

(Error)
PFR register / bit number
(Correct)
Port Setting Method

(Error)
See "Segment output register" and "Common output V pin control register".
(Correct)
See "Port Configuration 5.6 Analog Function Input or Output"</td></tr>
<tr><td>3367</td><td>CHAPTER57:LCD Controller</td><td>The description should be corrected as indicated by the shading below.
(Error)
− When the LCD is used (static drive, duty drive), set the following.
LCR1:ST [7:0] = "11111111"
LCDCMR [3:0] = "1111"
(Correct)
− When the LCD is used (static drive, duty drive), set the following.
LCR1 [7:0] = "11111111"
LCDCMR [3:0] = "1111"</td></tr>
<tr><td>3463
to
3490</td><td>CHAPTER61:Partial Wakeup Control</td><td>Added the following chapter.

CHAPTER61:Partial Wakeup Control</td></tr>
<tr><td>3494</td><td>Appendix : Various Settings of I/O
3. Output Resource Selection
Table 3-1 List of Output Resources</td><td>Port name should be added as indicated by the shading below.

(Correct)
<table>
<tr><th rowspan="2">Register
(Offset)</th><th rowspan="2">Port</th><th>Resource Functional Outputs</th></tr>
<tr><th>POF=6</th></tr>
<tr><td>PPC_PCFGR006
(0x000C)</td><td>P006</td><td>SDA3</td></tr>
<tr><td>PPC_PCFGR007
(0x000E)</td><td>P007</td><td>SCL3</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>3495</td><td>Appendix : Various Settings of I/O 3. Output Resource Selection Table 3-1 List of Output Resources</td><td>Port name should be added as indicated by the shading below.

(Correct)
<table>
<tr><th rowspan="2">Register (Offset)</th><th rowspan="2">Port</th><th>Resource Functional Outputs</th></tr>
<tr><th>POF=6</th></tr>
<tr><td>PPC_PCFGR020 (0x0028)</td><td>P020</td><td>SDA0</td></tr>
<tr><td>PPC_PCFGR021 (0x002A)</td><td>P021</td><td>SCL0</td></tr>
</table></td></tr>
<tr><td>3496</td><td>Appendix : Various Settings of I/O 3. Output Resource Selection Table 3-1 List of Output Resources</td><td>Port name should be added as indicated by the shading below.

(Correct)
<table>
<tr><th rowspan="2">Register (Offset)</th><th rowspan="2">Port</th><th colspan="2">Resource Functional Outputs</th></tr>
<tr><th>POF=3</th><th>POF=6</th></tr>
<tr><td>PPC_PCFGR131 (0x007E)</td><td>P131</td><td>-</td><td>SDA9</td></tr>
<tr><td>PPC_PCFGR202 (0x0084)</td><td>P202</td><td>-</td><td>SCL9</td></tr>
<tr><td>PPC_PCFGR207 (0x008E)</td><td>P207</td><td>SCL4</td><td>-</td></tr>
<tr><td>PPC_PCFGR209 (0x0092)</td><td>P209</td><td>SDA4</td><td>-</td></tr>
</table></td></tr>
<tr><td>3497</td><td>Appendix : Various Settings of I/O 3. Output Resource Selection Table 3-1 List of Output Resources</td><td>Port name should be added as indicated by the shading below.

(Correct)
<table>
<tr><th rowspan="2">Register (Offset)</th><th rowspan="2">Port</th><th>Resource Functional Outputs</th></tr>
<tr><th>POF=6</th></tr>
<tr><td>PPC_PCFGR225 (0x00B2)</td><td>P225</td><td>SDA8</td></tr>
<tr><td>PPC_PCFGR226 (0x00B4)</td><td>P226</td><td>SCL8</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>3498</td><td>Appendix : Various Settings of I/O 3. Output Resource Selection Table 3-1 List of Output Resources</td><td>Port name should be added as indicated by the shading below.

(Correct)
<table>
<tr><td rowspan="2">Register (Offset)</td><td rowspan="2">Port</td><td>Resource Functional Outputs</td></tr>
<tr><td>POF=3</td></tr>
<tr><td>PPC_PCFGR313 (0x00DA)</td><td>P313</td><td>SDA10</td></tr>
<tr><td>PPC_PCFGR314 (0x00DC)</td><td>P314</td><td>SCL10</td></tr>
</table></td></tr>
<tr><td>3498</td><td>Appendix : Various Settings of I/O 3. Output Resource Selection Table 3-1 List of Output Resources</td><td>Port name should be added as indicated by the shading below.

(Correct)
<table>
<tr><td rowspan="2">Register (Offset)</td><td rowspan="2">Port</td><td>Resource Functional Outputs</td></tr>
<tr><td>POF=4</td></tr>
<tr><td>PPC_PCFGR321 (0x00EA)</td><td>P321</td><td>PWUTRG</td></tr>
</table></td></tr>
<tr><td>3498</td><td>Appendix : Various Settings of I/O 3. Output Resource Selection Table 3-1 List of Output Resources</td><td>Port name should be deleted as indicated by the shading below.

(Correct)
<table>
<tr><td rowspan="2">Register (Offset)</td><td rowspan="2">Port</td><td>Resource Functional Outputs</td></tr>
<tr><td>POF=5</td></tr>
<tr><td>PPC_PCFGR406 (0x010C)</td><td>P406</td><td>-</td></tr>
</table></td></tr>
<tr><td>3501</td><td>Appendix : Various Settings of I/O 4. Resource Input Selection Table 4-1 List of Resource Inputs</td><td>Port name should be deleted as indicated by the shading below.

(Correct)
<table>
<tr><td>RIC_RESIN66 (0x0084)</td><td>RX0</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>3516</td><td>Appendix : Various Settings of I/O
3. Output Resource Selection
Table 3-1 List of Output Resources</td><td>Port name should be deleted as indicated by the shading below.

(Correct)
<table>
<tr><td rowspan="2">Register
(Offset)</td><td rowspan="2">Resource</td><td rowspan="2">RESSEL[3:0]
PORTSEL[11:8]</td><td>Source for Resource Input</td></tr>
<tr><td>1</td></tr>
<tr><td>RIC_RESIN190
(0x017C)</td><td>RX2</td><td>PORTSEL (0-7)</td><td>9</td></tr>
</table>
</td></tr>
<tr><td>3519
to
3534</td><td>Appendix : Various Settings of I/O
4. Resource Input Selection</td><td>Added registers from “ RIC_RESIN222” to “ RIC_RESIN367”.</td></tr>
<tr><td>3537</td><td>Appendix : State transition
2. Diagram of State transition</td><td>Revised the “Figure 2-1 Diagram of Device State Transitions” due to the following mode.

“PSS Partial WakeUp mode shutdown”</td></tr>
<tr><td>3538</td><td>Appendix : State transition
2. Diagram of State transition</td><td>Revised the “ Table 2-1 RUN/PSS State Definitions” due to the followings
- Revised clock frequency based 128MHz
- Added the state of “PSS PWU mode Shutdown”</td></tr>
<tr><td>3543</td><td>Appendix : State transition
4. Changes to PSS and RUN</td><td>Added “Figure 4-2 Changes to PSS Timing Chart”</td></tr>
<tr><td>3547
to
3687</td><td>Appendix: I/O Map</td><td>Revised "|" to "to" for express the range of address.</td></tr>
<tr><td>3582</td><td>Appendix: I/O Map
2. I/O Map (S6J3120 Series)</td><td>Features of "For details" should be corrected as indicated by the shading below.

(Error)
<table>
<tr><td>B040_8000</td><td>TPU_UNLOCK [R/W] W
00000000 00000000 00000000 00000000</td></tr>
</table>

(Correct)
<table>
<tr><td>B040_8000</td><td>TPU_UNLOCK [R/W] W
00000000 00000000 00000000 00000000</td><td>TPU</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>3579</td><td>Appendix: I/O Map
2. I/O Map (S6J3120 Series)</td><td>Following registers should be added as indicated by the shading below
(Correct)
<table>
<tr><td>B061_8034</td><td>RTC_PWUTRGCR [R/W] B,H,W
00000010 00000000 00000000 00000000</td><td></td></tr>
<tr><td>B061_8038</td><td>RTC_PWUTRGSR [R/W] B,H,W
00000000 00000000 00000000 00000000</td><td></td></tr>
</table></td></tr>
<tr><td>3586
to
3588</td><td>Appendix: I/O Map
2. I/O Map (S6J3120 Series)</td><td>Added address of “MPU for AXI of SHE” from “ B200_0400” to “ B20F_FFFF”</td></tr>
<tr><td>3612
to
3614</td><td>Appendix: I/O Map
2. I/O Map (S6J3120 Series)</td><td>Added the following registers
B474_81C8 : RIC_RESIN228
to
B474_82DE : RIC_RESIN367</td></tr>
<tr><td>3619,
3620,
3623
to
3625</td><td>Appendix: I/O Map
2. I/O Map (S6J3120 Series)</td><td>Revised the I2C register in CPG_MFS ch0,ch 3,ch 4</td></tr>
<tr><td>3626</td><td>Appendix: I/O Map
2. I/O Map (S6J3120 Series)</td><td>Following registers should be added as indicated by the shading below (Base Timer ch0)
(Correct)
<table>
<tr><td>B480_8024</td><td>-</td><td>BT00_DEBUG [R/W] B,H,W
00000000 00000000</td><td></td></tr>
<tr><td>B480_8028</td><td>-</td><td>-</td><td></td></tr>
<tr><td>B480_802C</td><td>-</td><td>BT00_ETCDR [R/W] B,H,W
01111111 11111111</td><td></td></tr>
</table></td></tr>
<tr><td>3627</td><td>Appendix: I/O Map
2. I/O Map (S6J3120 Series)</td><td>Following registers should be added as indicated by the shading below (Base Timer ch1)
(Correct)
<table>
<tr><td>B480_8424</td><td>-</td><td>BT01_DEBUG [R/W] B,H,W
00000000 00000000</td><td></td></tr>
<tr><td>B480_8428</td><td>-</td><td>-</td><td></td></tr>
<tr><td>B480_842C</td><td>-</td><td>BT01_ETCDR [R/W] B,H,W
01111111 11111111</td><td></td></tr>
</table></td></tr>
</table>

Page	Section	Change Results
3627	Appendix: I/O Map 2. I/O Map (S6J3120 Series)	Following registers should be added as indicated by the shading below (Base Timer ch2) (Correct) B480_8824 - BT02_DEBUG [R/W] B,H,W 00000000 00000000 B480_8828 - - B480_882C - BT02_ETCDR [R/W] B,H,W 01111111 11111111
3628	Appendix: I/O Map 2. I/O Map (S6J3120 Series)	Following registers should be added as indicated by the shading below (Base Timer ch3) (Correct) B480_8C24 - BT03_DEBUG [R/W] B,H,W 00000000 00000000 B480_8C28 - - B480_8C2C - BT03_ETCDR [R/W] B,H,W 01111111 11111111
3629	Appendix: I/O Map 2. I/O Map (S6J3120 Series)	Following registers should be added as indicated by the shading below (Base Timer ch4) (Correct) B480_9024 - BT04_DEBUG [R/W] B,H,W 00000000 00000000 B480_9028 - - B480_902C - BT04_ETCDR [R/W] B,H,W 01111111 11111111
3629	Appendix: I/O Map 2. I/O Map (S6J3120 Series)	Following registers should be added as indicated by the shading below (Base Timer ch5) (Correct) B480_9424 - BT05_DEBUG [R/W] B,H,W 00000000 00000000 B480_9428 - - B480_942C - BT05_ETCDR [R/W] B,H,W 01111111 11111111

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>3630</td><td>Appendix: I/O Map
2. I/O Map (S6J3120 Series)</td><td>Following registers should be added as indicated by the shading below (Base Timer ch6)
(Correct)
<table>
<tr><td>B480_9824</td><td>-</td><td>BT06_DEBUG [R/W] B,H,W
00000000 00000000</td><td></td></tr>
<tr><td>B480_9828</td><td>-</td><td>-</td><td></td></tr>
<tr><td>B480_982C</td><td>-</td><td>BT06_ETCDR [R/W] B,H,W
01111111 11111111</td><td></td></tr>
</table></td></tr>
<tr><td>3631</td><td>Appendix: I/O Map
2. I/O Map (S6J3120 Series)</td><td>Following registers should be added as indicated by the shading below (Base Timer ch7)
(Correct)
<table>
<tr><td>B480_9C24</td><td>-</td><td>BT07_DEBUG [R/W] B,H,W
00000000 00000000</td><td></td></tr>
<tr><td>B480_9C28</td><td>-</td><td>-</td><td></td></tr>
<tr><td>B480_9C2C</td><td>-</td><td>BT07_ETCDR [R/W] B,H,W
01111111 11111111</td><td></td></tr>
</table></td></tr>
<tr><td>3631</td><td>Appendix: I/O Map
2. I/O Map (S6J3120 Series)</td><td>Following registers should be added as indicated by the shading below (Base Timer ch8)
(Correct)
<table>
<tr><td>B480_A024</td><td>-</td><td>BT08_DEBUG [R/W] B,H,W
00000000 00000000</td><td></td></tr>
<tr><td>B480_A028</td><td>-</td><td>-</td><td></td></tr>
<tr><td>B480_A02C</td><td>-</td><td>BT08_ETCDR [R/W] B,H,W
01111111 11111111</td><td></td></tr>
</table></td></tr>
<tr><td>3632</td><td>Appendix: I/O Map
2. I/O Map (S6J3120 Series)</td><td>Following registers should be added as indicated by the shading below (Base Timer ch9)
(Correct)
<table>
<tr><td>B480_A424</td><td>-</td><td>BT09_DEBUG [R/W] B,H,W
00000000 00000000</td><td></td></tr>
<tr><td>B480_A428</td><td>-</td><td>-</td><td></td></tr>
<tr><td>B480_A42C</td><td>-</td><td>BT09_ETCDR [R/W] B,H,W
01111111 11111111</td><td></td></tr>
</table></td></tr>
</table>

Page	Section	Change Results
3632	Appendix: I/O Map 2. I/O Map (S6J3120 Series)	Following registers should be added as indicated by the shading below (Base Timer ch10) (Correct) B480_A824 - BT10_DEBUG [R/W] B,H,W 00000000 00000000 B480_A828 - - B480_A82C - BT10_ETCDR [R/W] B,H,W 01111111 11111111
3633	Appendix: I/O Map 2. I/O Map (S6J3120 Series)	Following registers should be added as indicated by the shading below (Base Timer ch11) (Correct) B480_AC24 - BT11_DEBUG [R/W] B,H,W 00000000 00000000 B480_AC28 - - B480_AC2C - BT11_ETCDR [R/W] B,H,W 01111111 11111111
3635	Appendix: I/O Map 2. I/O Map (S6J3120 Series)	Features of "For details" should be added as indicated by the shading below (FRT ch0) (Correct) B482_0018 - FRT00_DEBUG [R/W] B,H,W 00000000 00000000
3635	Appendix: I/O Map 2. I/O Map (S6J3120 Series)	Features of "For details" should be added as indicated by the shading below (FRT ch1) (Correct) B482_0418 - FRT01_DEBUG [R/W] B,H,W 00000000 00000000
3636	Appendix: I/O Map 2. I/O Map (S6J3120 Series)	Features of "For details" should be added as indicated by the shading below (FRT ch2) (Correct) B482_0818 - FRT02_DEBUG [R/W] B,H,W 00000000 00000000
3636	Appendix: I/O Map 2. I/O Map (S6J3120 Series)	Features of "For details" should be added as indicated by the shading below (FRT ch3) (Correct) B482_0C18 - FRT03_DEBUG [R/W] B,H,W 00000000 00000000
3636	Appendix: I/O Map 2. I/O Map (S6J3120 Series)	Features of "For details" should be added as indicated by the shading below (FRT ch4) (Correct) B482_1018 - FRT04_DEBUG [R/W] B,H,W 00000000 00000000

Page	Section	Change Results
3637	Appendix: I/O Map 2. I/O Map (S6J3120 Series)	Features of "For details" should be added as indicated by the shading below (FRT ch5) (Correct) B482_1418 | - | FRT05_DEBUG [R/W] B,H,W 00000000 00000000 |
3641 to 3646	Appendix: I/O Map 2. I/O Map (S6J3120 Series)	Revised the I2C register in CPG_MFS ch8 to ch11
3652	Appendix: I/O Map 2. I/O Map (S6J3120 Series)	Revised the following initial value (Correct) B48C_6008 | COMVER [R/W] B,H,W 11111111 11111111 11111111 11111111 |
3653	Appendix: I/O Map 2. I/O Map (S6J3120 Series)	Features of "For details" should be corrected as indicated by the shading below (Error) B490_000C | CPG_CANFD0_FBTP [R/W] B,H,W 00000000 00000000 00001010 00110011 | (Correct) B490_000C | CPG_CANFD0_DBTP [R/W] B,H,W 00000000 00000000 00001010 00110011 |
3653	Appendix: I/O Map 2. I/O Map (S6J3120 Series)	Features of "For details" should be corrected as indicated by the shading below (Error) B490_001C | CPG_CANFD0_BTP [R/W] B,H,W 00000000 00000000 00001010 00110011 | (Correct) B490_001C | CPG_CANFD0_NBTP [R/W] B,H,W 00001100 00000000 00001010 00000011 |

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>3653</td><td>Appendix: I/O Map
2. I/O Map (S6J3120 Series)</td><td>Features of "For details" should be corrected as indicated by the shading below
(Error)
<table><tr><td>B490_0000</td><td>CPG_CANFD0_CREL [R] B,H,W
00110000 00010011 00000101 00000110</td><td></td></tr></table>
(Correct)
<table><tr><td>B490_0000</td><td>CPG_CANFD0_CREL [R] B,H,W
00110010 00000100 00010010 00011000</td><td></td></tr></table></td></tr>
<tr><td>3653</td><td>Appendix: I/O Map
2. I/O Map (S6J3120 Series)</td><td>Features of "For details" should be corrected as indicated by the shading below
(Error)
<table><tr><td>B490_0044</td><td>CPG_CANFD0_PSR [R] H,W
XXXXXXXX XXXXXXXX XX000111 00000111</td><td></td></tr></table>
(Correct)
<table><tr><td>B490_0044</td><td>CPG_CANFD0_PSR [R] H,W
XXXXXXXX XXXXXXXX X0000111 00000111</td><td></td></tr></table></td></tr>
<tr><td>3653</td><td>Appendix: I/O Map
2. I/O Map (S6J3120 Series)</td><td>Features of "For details" should be deleted as indicated by the shading below
(Error)
CAN_FD ch.0
(*8)
(Correct)
CAN_FD ch.0</td></tr>
<tr><td>3653</td><td>Appendix: I/O Map
2. I/O Map (S6J3120 Series)</td><td>Features of "For details" should be corrected as indicated by the shading below
(Correct)
<table><tr><td>B490_0048</td><td>CPG_CANFD0_TDCR [R/W] B,H,W
00000000 00000000 00000000 00000000</td><td rowspan="2"></td></tr><tr><td>B490_004C</td><td>-</td></tr></table></td></tr>
<tr><td>3654</td><td>Appendix: I/O Map
2. I/O Map (S6J3120 Series)</td><td>Features of "For details" should be corrected as indicated by the shading below

(Correct)
<table><tr><td>B490_00B4</td><td>CPG_CANFD0_RXF1S [R] B,H,W
00XXXX00 XX000000 XX000000 X0000000</td><td></td></tr></table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>3655</td><td>Appendix: I/O Map
2. I/O Map (S6J3120 Series)</td><td>Features of "For details" should be corrected as indicated by the shading below
(Error)
<table><tr><td>B491_000C</td><td>CPG_CANFD1_FBTP [R/W] B,H,W
00000000 00000000 00001010 00110011</td><td></td></tr></table>
(Correct)
<table><tr><td>B491_000C</td><td>CPG_CANFD1_DBTP [R/W] B,H,W
00000000 00000000 00001010 00110011</td><td></td></tr></table></td></tr>
<tr><td>3655</td><td>Appendix: I/O Map
2. I/O Map (S6J3120 Series)</td><td>Features of "For details" should be corrected as indicated by the shading below
(Error)
<table><tr><td>B491_001C</td><td>CPG_CANFD0_BTP [R/W] B,H,W
00000000 00000000 00001010 00110011</td><td></td></tr></table>
(Correct)
<table><tr><td>B491_001C</td><td>CPG_CANFD1_NBTP [R/W] B,H,W
00001100 00000000 00001010 00000011</td><td></td></tr></table></td></tr>
<tr><td>3655</td><td>Appendix: I/O Map
2. I/O Map (S6J3120 Series)</td><td>Features of "For details" should be corrected as indicated by the shading below
(Error)
<table><tr><td>B491_0000</td><td>CPG_CANFD1_CREL [R] B,H,W
00110000 00010011 00000101 00000110</td><td></td></tr></table>
(Correct)
<table><tr><td>B491_0000</td><td>CPG_CANFD1_CREL [R] B,H,W
00110010 00000100 00010010 00011000</td><td></td></tr></table></td></tr>
<tr><td>3655</td><td>Appendix: I/O Map
2. I/O Map (S6J3120 Series)</td><td>Features of "For details" should be deleted as indicated by the shading below
(Error)
CAN_FD ch.1
(*8)
(Correct)
CAN_FD ch.1</td></tr>
<tr><td>3656</td><td>Appendix: I/O Map
2. I/O Map (S6J3120 Series)</td><td>Features of "For details" should be corrected as indicated by the shading below
(Correct)
<table><tr><td>B491_0048</td><td>CPG_CANFD1_TDCR [R/W] B,H,W
00000000 00000000 00000000 00000000</td><td rowspan="2"></td></tr><tr><td>B491_004C</td><td>-</td></tr></table></td></tr>
</table>

Page	Section	Change Results
3656	Appendix: I/O Map 2. I/O Map (S6J3120 Series)	Features of "For details" should be corrected as indicated by the shading below (Error) B491_0044 | CPG_CANFD1_PSR [R] H,W XXXXXXXX XXXXXXXX XX000111 00000111 (Correct) B491_0044 | CPG_CANFD1_PSR [R] H,W XXXXXXXX XXXXXXXX X0000111 00000111
3657	Appendix: I/O Map 2. I/O Map (S6J3120 Series)	Features of "For details" should be corrected as indicated by the shading below (Correct) B491_00B4 | CPG_CANFD1_RXF1S [R] B,H,W 00XXXX00 XX000000 XX000000 X0000000
3658	Appendix: I/O Map 2. I/O Map (S6J3120 Series)	Features of "For details" should be corrected as indicated by the shading below (Error) B492_000C | CPG_CANFD2_FBTP [R/W] B,H,W 00000000 00000000 00001010 00110011 (Correct) B492_000C | CPG_CANFD2_DBTP [R/W] B,H,W 00000000 00000000 00001010 00110011
3658	Appendix: I/O Map 2. I/O Map (S6J3120 Series)	Features of "For details" should be corrected as indicated by the shading below (Error) B492_001C | CPG_CANFD2_BTP [R/W] B,H,W 00000000 00000000 00001010 00110011 (Correct) B492_001C | CPG_CANFD2_ NBTP [R/W] B,H,W 00001100 00000000 00001010 00000011

Page	Section	Change Results
3658	Appendix: I/O Map 2. I/O Map (S6J3120 Series)	Features of "For details" should be corrected as indicated by the shading below (Error) B492_0000 | CPG_CANFD2_CREL [R] B,H,W 00110000 00010011 00000101 00000110 (Correct) B492_0000 | CPG_CANFD2_CREL [R] B,H,W 00110010 00000100 00010010 00011000
3658	Appendix: I/O Map 2. I/O Map (S6J3120 Series)	Features of "For details" should be deleted as indicated by the shading below (Error) CAN_FD ch.2 (*8) (Correct) CAN_FD ch.2
3658	Appendix: I/O Map 2. I/O Map (S6J3120 Series)	Features of "For details" should be corrected as indicated by the shading below (Correct) B492_0048 | CPG_CANFD2_TDCR [R/W] B,H,W 00000000 00000000 00000000 00000000 B492_004C | -
3658	Appendix: I/O Map 2. I/O Map (S6J3120 Series)	Features of "For details" should be corrected as indicated by the shading below (Error) B492_0044 | CPG_CANFD2_PSR [R] H,W XXXXXXXX XXXXXXXX XX000111 00000111 (Correct) B492_0044 | CPG_CANFD2_PSR [R] H,W XXXXXXXX XXXXXXXX X0000111 00000111
3659	Appendix: I/O Map 2. I/O Map (S6J3120 Series)	Features of "For details" should be corrected as indicated by the shading below (Correct) B492_00B4 | CPG_CANFD2_RXF1S [R] B,H,W 00XXXX00 XX000000 XX000000 X0000000
3663	Appendix: I/O Map 2. I/O Map (S6J3120 Series)	Added " *7:Because it is not I2C mode, it reserves immediately after reset."

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>3663</td><td>Appendix: I/O Map
2. I/O Map (S6J3120 Series)</td><td>The following note should be deleted as indicated by the shading below.

*8:see "mcan_users_manual_v302.pdf"</td></tr>
<tr><td>3665</td><td>Appendix: I/O Map
3. APPS#5</td><td>Following registers should be added as indicated by the shading below (Base Timer ch12)
(Correct)
<table>
<tr><td>B484_3824</td><td>-</td><td>BT12_DEBUG [R/W] B,H,W
00000000 00000000</td><td></td></tr>
<tr><td>B484_3828</td><td>-</td><td>-</td><td></td></tr>
<tr><td>B484_382C</td><td>-</td><td>BT12_ETCDR [R/W] B,H,W
01111111 11111111</td><td></td></tr>
</table>
</td></tr>
<tr><td>3666</td><td>Appendix: I/O Map
3. APPS#5</td><td>Following registers should be added as indicated by the shading below (Base Timer ch13)
(Correct)
<table>
<tr><td>B484_3C24</td><td>-</td><td>BT13_DEBUG [R/W] B,H,W
00000000 00000000</td><td></td></tr>
<tr><td>B484_3C28</td><td>-</td><td>-</td><td></td></tr>
<tr><td>B484_3C2C</td><td>-</td><td>BT13_ETCDR [R/W] B,H,W
01111111 11111111</td><td></td></tr>
</table>
</td></tr>
<tr><td>3666</td><td>Appendix: I/O Map
3. APPS#5</td><td>Following registers should be added as indicated by the shading below (Base Timer ch14)
(Correct)
<table>
<tr><td>B484_4024</td><td>-</td><td>BT14_DEBUG [R/W] B,H,W
00000000 00000000</td><td></td></tr>
<tr><td>B484_4028</td><td>-</td><td>-</td><td></td></tr>
<tr><td>B484_402C</td><td>-</td><td>BT14_ETCDR [R/W] B,H,W
01111111 11111111</td><td></td></tr>
</table>
</td></tr>
<tr><td>3667</td><td>Appendix: I/O Map
3. APPS#5</td><td>Following registers should be added as indicated by the shading below (Base Timer ch15)
(Correct)
<table>
<tr><td>B484_4424</td><td>-</td><td>BT15_DEBUG [R/W] B,H,W
00000000 00000000</td><td></td></tr>
<tr><td>B484_4428</td><td>-</td><td>-</td><td></td></tr>
<tr><td>B484_442C</td><td>-</td><td>BT15_ETCDR [R/W] B,H,W
01111111 11111111</td><td></td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>3667,
3668</td><td>Appendix: I/O Map
3. APPS#5</td><td>Following registers should be added as indicated by the shading below (Base Timer ch16)
(Correct)
<table>
<tr><td>B484_4824</td><td>-</td><td>BT16_DEBUG [R/W] B,H,W
00000000 00000000</td><td></td></tr>
<tr><td>B484_4828</td><td>-</td><td>-</td><td></td></tr>
<tr><td>B484_482C</td><td>-</td><td>BT16_ETCDR [R/W] B,H,W
01111111 11111111</td><td></td></tr>
</table>
</td></tr>
<tr><td>3668</td><td>Appendix: I/O Map
3. APPS#5</td><td>Following registers should be added as indicated by the shading below (Base Timer ch17)
(Correct)
<table>
<tr><td>B484_4C24</td><td>-</td><td>BT17_DEBUG [R/W] B,H,W
00000000 00000000</td><td></td></tr>
<tr><td>B484_4C28</td><td>-</td><td>-</td><td></td></tr>
<tr><td>B484_4C2C</td><td>-</td><td>BT17_ETCDR [R/W] B,H,W
01111111 11111111</td><td></td></tr>
</table>
</td></tr>
<tr><td>3669</td><td>Appendix: I/O Map
3. APPS#5</td><td>Following registers should be added as indicated by the shading below (Base Timer ch18)
(Correct)
<table>
<tr><td>B484_5024</td><td>-</td><td>BT18_DEBUG [R/W] B,H,W
00000000 00000000</td><td></td></tr>
<tr><td>B484_5028</td><td>-</td><td>-</td><td></td></tr>
<tr><td>B484_502C</td><td>-</td><td>BT18_ETCDR [R/W] B,H,W
01111111 11111111</td><td></td></tr>
</table>
</td></tr>
<tr><td>3669</td><td>Appendix: I/O Map
3. APPS#5</td><td>Following registers should be added as indicated by the shading below (Base Timer ch19)
(Correct)
<table>
<tr><td>B484_541C</td><td>-</td><td>BT19_DEBUG [R/W] B,H,W
00000000 00000000</td><td></td></tr>
<tr><td>B484_5420</td><td>-</td><td>-/BT19_ADTR/-/- [R/W] H,W-
00000000 00000000</td><td></td></tr>
<tr><td>B484_5424</td><td>-</td><td>BT19_ETCDR [R/W] B,H,W
01111111 11111111</td><td></td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>3670</td><td>Appendix: I/O Map
3. APPS#5</td><td>Following registers should be added as indicated by the shading below (Base Timer ch20)
(Correct)
<table>
<tr><td>B484_5824</td><td>-</td><td>BT20_DEBUG [R/W] B,H,W
00000000 00000000</td><td></td></tr>
<tr><td>B484_5828</td><td>-</td><td>-</td><td></td></tr>
<tr><td>B484_582C</td><td>-</td><td>BT20_ETCDR [R/W] B,H,W
01111111 11111111</td><td></td></tr>
</table>
</td></tr>
<tr><td>3671</td><td>Appendix: I/O Map
3. APPS#5</td><td>Following registers should be added as indicated by the shading below (Base Timer ch21)
(Correct)
<table>
<tr><td>B484_5C24</td><td>-</td><td>BT21_DEBUG [R/W] B,H,W
00000000 00000000</td><td></td></tr>
<tr><td>B484_5C28</td><td>-</td><td>-</td><td></td></tr>
<tr><td>B484_5C2C</td><td>-</td><td>BT21_ETCDR [R/W] B,H,W
01111111 11111111</td><td></td></tr>
</table>
</td></tr>
<tr><td>3672</td><td>Appendix: I/O Map
3. APPS#5</td><td>Following registers should be added as indicated by the shading below (Base Timer ch22)
(Correct)
<table>
<tr><td>B484_6024</td><td>-</td><td>BT22_DEBUG [R/W] B,H,W
00000000 00000000</td><td></td></tr>
<tr><td>B484_6028</td><td>-</td><td>-</td><td></td></tr>
<tr><td>B484_602C</td><td>-</td><td>BT22_ETCDR [R/W] B,H,W
01111111 11111111</td><td></td></tr>
</table>
</td></tr>
<tr><td>3672</td><td>Appendix: I/O Map
3. APPS#5</td><td>Following registers should be added as indicated by the shading below (Base Timer ch23)
(Correct)
<table>
<tr><td>B484_6424</td><td>-</td><td>BT23_DEBUG [R/W] B,H,W
00000000 00000000</td><td></td></tr>
<tr><td>B484_6428</td><td>-</td><td>-</td><td></td></tr>
<tr><td>B484_642C</td><td>-</td><td>BT23_ETCDR [R/W] B,H,W
01111111 11111111</td><td></td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>3673</td><td>Appendix: I/O Map
3. APPS#5</td><td>Following registers should be added as indicated by the shading below (Base Timer ch24)
(Correct)
<table>
<tr><td>B484_6824</td><td>-</td><td>BT24_DEBUG [R/W] B,H,W
00000000 00000000</td><td></td></tr>
<tr><td>B484_6828</td><td>-</td><td>-</td><td></td></tr>
<tr><td>B484_682C</td><td>-</td><td>BT24_ETCDR [R/W] B,H,W
01111111 11111111</td><td></td></tr>
</table>
</td></tr>
<tr><td>3674</td><td>Appendix: I/O Map
3. APPS#5</td><td>Following registers should be added as indicated by the shading below (Base Timer ch25)
(Correct)
<table>
<tr><td>B484_6C28</td><td>-</td><td>BT25_DEBUG [R/W] B,H,W
00000000 00000000</td><td></td></tr>
<tr><td>B484_6C24</td><td>-</td><td>-</td><td></td></tr>
<tr><td>B484_6C2C</td><td>-</td><td>BT25_ETCDR [R/W] B,H,W
01111111 11111111</td><td></td></tr>
</table>
</td></tr>
<tr><td>3674</td><td>Appendix: I/O Map
3. APPS#5</td><td>Following registers should be added as indicated by the shading below (Base Timer ch26)
(Correct)
<table>
<tr><td>B484_7024</td><td>-</td><td>BT26_DEBUG [R/W] B,H,W
00000000 00000000</td><td></td></tr>
<tr><td>B484_7028</td><td>-</td><td>-</td><td></td></tr>
<tr><td>B484_702C</td><td>-</td><td>BT26_ETCDR [R/W] B,H,W
01111111 11111111</td><td></td></tr>
</table>
</td></tr>
<tr><td>3675</td><td>Appendix: I/O Map
3. APPS#5</td><td>Following registers should be added as indicated by the shading below (Base Timer ch27)
(Correct)
<table>
<tr><td>B484_7424</td><td>-</td><td>BT27_DEBUG [R/W] B,H,W
00000000 00000000</td><td></td></tr>
<tr><td>B484_7428</td><td>-</td><td>-</td><td></td></tr>
<tr><td>B484_742C</td><td>-</td><td>BT27_ETCDR [R/W] B,H,W
01111111 11111111</td><td></td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>3676</td><td>Appendix: I/O Map
3. APPS#5</td><td>Following registers should be added as indicated by the shading below (Base Timer ch28)
(Correct)
<table>
<tr><td>B484_7824</td><td>-</td><td>BT28_DEBUG [R/W] B,H,W
00000000 00000000</td><td></td></tr>
<tr><td>B484_7828</td><td>-</td><td>-</td><td></td></tr>
<tr><td>B484_782C</td><td>-</td><td>BT28_ETCDR [R/W] B,H,W
01111111 11111111</td><td></td></tr>
</table>
</td></tr>
<tr><td>3676</td><td>Appendix: I/O Map
3. APPS#5</td><td>Following registers should be added as indicated by the shading below (Base Timer ch29)
(Correct)
<table>
<tr><td>B484_7C24</td><td>-</td><td>BT29_DEBUG [R/W] B,H,W
00000000 00000000</td><td></td></tr>
<tr><td>B484_7C28</td><td>-</td><td>-</td><td></td></tr>
<tr><td>B484_7C2C</td><td>-</td><td>BT29_ETCDR [R/W] B,H,W
01111111 11111111</td><td></td></tr>
</table>
</td></tr>
<tr><td>3687</td><td>Appendix: I/O Map
3. APPS#5</td><td>Added the register address of Partial Wakeup to “ B484_8600”</td></tr>
<tr><td>3687</td><td>Appendix: I/O Map
3. APPS#5</td><td>Added the register address of Global Timer from “ B484_9000” to “ B484_9028”</td></tr>
<tr><td>3689</td><td>Appendix: DMA channel Activation Factors</td><td>Revised product names as follows:
(Correct)
1. DMA channel Activation Factors (S6J312xHzB*)</td></tr>
<tr><td>3690, 3692</td><td>Appendix : DMA channel Activation Factors
1. DMA Channel Activation Factors (S6J312xHzB*)</td><td>Revised product names as follows:

(Correct)
1. DMA Channel Activation Factors (S6J312xHzB*)
This section shows DMA channel activation factors in S6J312xHzB*.
DMA sources from the peripherals in S6J312xHzB*.

*x: A/9/8, z: A/B</td></tr>
<tr><td>3693</td><td>Appendix : List of Interrupt Vector and NMI map</td><td>Revised product names as follows:
(Correct)
1. List of Interrupt Vector (S6J312xHzB*)
2. NMI map (S6J312xHzB*)</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>3694</td><td>Appendix : List of Interrupt Vector and NMI map
1. List of Interrupt Vector (S6J312xHzB*)</td><td>Revised product names as follows:
(Correct)
1. List of Interrupt Vector (S6J312xHzB*)
This section shows list of interrupt vector in S6J312xHzB*.
This list shows the assignments of interrupt factors and interrupt vectors/interrupt control registers.
*x: A/9/8, z: A/B</td></tr>
<tr><td>3700</td><td>Appendix : List of Interrupt Vector and NMI map
1. List of Interrupt Vector (S6J312xHzB*)</td><td>Added the following table.
<table><tr><th>IRQ No.</th><th>Detail</th><th>IRQ Priority Register</th><th>Vector Address Register</th></tr><tr><td>401</td><td>Global Timer Compare Clear</td><td>IRCn_IRQPL100:IRQPL401 [4:0]</td><td>IRCn_IRQVA401</td></tr></table></td></tr>
<tr><td>3700</td><td>Appendix : List of Interrupt Vector and NMI map
1. List of Interrupt Vector (S6J312xHzB*)</td><td>Added the following note

−SHE OFF product is prohibited to use Interrupt for SHE function.</td></tr>
<tr><td>3701</td><td>Appendix : List of Interrupt Vector and NMI map
2. NMI Map (S6J312xHzB*)</td><td>Revised product names as follows:
(Correct)
2. NMI Map (S6J312xHzB*)
This section shows NMI map in S6J312xHzB*.
*x: A/9/8, z: A/B</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>3701</td><td>Appendix : List of Interrupt Vector and NMI map
2. NMI Map (S6J312xHzB*)</td><td>The register bit number should be corrected as indicated by the shading below.

(Error)
<table>
<tr><th>No</th><th>Detail</th><th>NMI Priority Register</th><th>Vector Address Register</th></tr>
<tr><td>11</td><td>Backup RAM 2-bit ECC error detection</td><td>IRCn_NMIPL2:NMIPL3[4:0]</td><td>IRCn_NMIVA11</td></tr>
<tr><td>12</td><td>CAN-FD RAMs 2-bit ECC error detection</td><td>IRCn_NMIPL3:NMIPL0[4:0]</td><td>IRCn_NMIVA12</td></tr>
<tr><td>13</td><td>DMAC MPU #0 protection violation</td><td>IRCn_NMIPL3:NMIPL1[4:0]</td><td>IRCn_NMIVA13</td></tr>
<tr><td>14</td><td colspan="3">reserved</td></tr>
<tr><td>15</td><td>SHE MPU</td><td>IRCn_NMIPL3:NMIPL3[4:0]</td><td>IRCn_NMIVA15</td></tr>
<tr><td>16</td><td colspan="3">reserved</td></tr>
<tr><td>17</td><td colspan="3">reserved</td></tr>
<tr><td>18</td><td>TPU(s) protection violation</td><td>IRCn_NMIPL4:NMIPL2[4:0]</td><td>IRCn_NMIVA18</td></tr>
<tr><td>19 to 31</td><td colspan="3">reserved</td></tr>
</table>
(Correct)
<table>
<tr><th>No</th><th>Detail</th><th>NMI Priority Register</th><th>Vector Address Register</th></tr>
<tr><td>11</td><td>Backup RAM 2-bit ECC error detection</td><td>IRCn_NMIPL2:NMIPL11[3:0]</td><td>IRCn_NMIVA11</td></tr>
<tr><td>12</td><td>CAN-FD RAMs 2-bit ECC error detection</td><td>IRCn_NMIPL3:NMIPL12[3:0]</td><td>IRCn_NMIVA12</td></tr>
<tr><td>13</td><td>DMAC MPU #0 protection violation</td><td>IRCn_NMIPL3:NMIPL13[3:0]</td><td>IRCn_NMIVA13</td></tr>
<tr><td>14</td><td colspan="3">Reserved</td></tr>
<tr><td>15</td><td>SHE MPU</td><td>IRCn_NMIPL3:NMIPL15[3:0]</td><td>IRCn_NMIVA15</td></tr>
<tr><td>16</td><td colspan="3">Reserved</td></tr>
<tr><td>17</td><td colspan="3">Reserved</td></tr>
<tr><td>18</td><td>TPU(s) protection violation</td><td>IRCn_NMIPL4:NMIPL18[3:0]</td><td>IRCn_NMIVA18</td></tr>
<tr><td>19 to 31</td><td colspan="3">Reserved</td></tr>
</table></td></tr>
</table>

Page	Section	Change Results
3701	Appendix : List of Interrupt Vector and NMI map 2. NMI Map (S6J312xHzB*)	Added the following note −SHE OFF product is prohibited to use Interrupt for SHE function.
Revision *B		
-	Cover	Revised title as follows (Error) S6J3120 Series 32-bit Microcontroller Spansion® TraveoTM Family Hardware Manual (Correct) 32-bit Microcontroller S6J3120 Series Hardware Manual Traveo ™ Family
66	CHAPTER 1:Overview 2. Features 2.2. Peripheral Functions	Added “-CAN-FD (V3.2.0)” under CAN controller: CAN-FD Max 2 channel (Error) CAN controller: CAN-FD Max 3 channel (Correct) CAN controller: CAN-FD Max 3 channel -CAN-FD (V3.2.0)
168, 180, 191	CHAPTER 4:Reset	Features of "For details" should be corrected as indicated by the shading below (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”.
181	CHAPTER 4:Reset 4.4. User PowerDomain Reset Factor Register (SYSC_PDRSTCAUSEUR)	Features of "For details" should be corrected as indicated by the shading below (Error) Bit12 PD5R1: PowerDomain5_0 Reset Detection Bit (Correct) Bit12 PD5R0: PowerDomain5_0 Reset Detection Bit

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>218</td><td>CHAPTER 5:Clock System 3.3. Clock Distributer</td><td>Revised Frequency as follows
(Error)
CD_TRC 32MHz

(Correct)
CD_TRC 64MHz</td></tr>
<tr><td>732</td><td>CHAPTER 9: Source Clock Timer 1. Overview</td><td>Features of "For details" should be corrected as indicated by the shading below
(Error)
The high-speed CR clock timer has an oscillation stabilization wait time of 0.24 to 0.88 ms.
(Correct)
The high-speed CR clock timer has an oscillation stabilization wait time of 0.32 to 0.8 ms.</td></tr>
<tr><td>743</td><td>CHAPTER 9: Source Clock Timer 5.3. High-Speed CR Clock Timer Compare Prescaler Register (SYSC_FCRC TCPR)</td><td>Features of "For details" should be corrected as indicated by the shading below
(Error)
<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)
<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>0.32 to 0.8 [ms]</td></tr></table></td></tr>
<tr><td>976</td><td>CHAPTER 15:Software Watchdog Timer</td><td>Features of "For details" should be corrected as indicated by the shading below
(Error)
6.1. Software Watchdog Protection Register (SWDT_PROT)

(Correct)
6.1. Software Watchdog Protection Register (SWDG_PROT)</td></tr>
</table>

Page	Section	Change Results
1612	CHAPTER 28:DMA Controller 4.17. DMA Controller Client Matrix Channel Interface Configuration Register 'n' (DMAi_CMCHICn)	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. 000 | Value of DMAAn_CHCHICRDB0 register is copied to DMAi_CMCHICn at CMCHIC reload event. 001 | Value of DMAAn_CHCHICRDB1 register is copied to DMAi_CMCHICn at CMCHIC reload event. 111 | Value of DMAAn_CHCHICRDB8 register is copied to DMAi_CMCHICn at CMCHIC reload event. (Correct) [bit18:16] RLSLOT[2:0] : Reload Slot Replacement The bit specifies which bank is copied DMAi_CMCHICn at CMCHIC reload event. 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.
1660	CHAPTER 31:CAN FD Controller 1:Overview	Added “-CAN-FD (V3.2.0)” under Features of the CAN FD Controller (Error) Features of the CAN FD Controller (Correct) Features of the CAN FD Controller -CAN-FD (V3.2.0)

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1729,
1730</td><td>CHAPTER 31:CAN FD Controller
5.3. Data Bit Timing & Prescaler Register (DBTP)</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="3">Reserved</td><td colspan="5">DTSEG1[3:0]</td></tr>
<tr><td>R/W Attribute</td><td colspan="3">R0,W0</td><td colspan="5">R/W</td></tr>
<tr><td>Protection Attribute</td><td colspan="3">-</td><td colspan="5">-</td></tr>
<tr><td>Initial value</td><td colspan="3">000</td><td colspan="5">0x0A</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="4">DTSEG2[2:0]</td><td colspan="4">DSJW[1:0]</td></tr>
<tr><td>R/W Attribute</td><td colspan="4">R/W</td><td colspan="4">R/W</td></tr>
<tr><td>Protection Attribute</td><td colspan="4">-</td><td colspan="4">-</td></tr>
<tr><td>Initial value</td><td colspan="4">0x3</td><td colspan="4">0x3</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 colspan="5">DTSEG1[4:0]</td></tr>
<tr><td>R/W Attribute</td><td colspan="3">R0,W0</td><td colspan="5">R/W</td></tr>
<tr><td>Protection Attribute</td><td colspan="3">-</td><td colspan="5">-</td></tr>
<tr><td>Initial value</td><td colspan="3">000</td><td colspan="5">0x0A</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="4">DTSEG2[3:0]</td><td colspan="4">DSJW[3:0]</td></tr>
<tr><td>R/W Attribute</td><td colspan="4">R/W</td><td colspan="4">R/W</td></tr>
<tr><td>Protection Attribute</td><td colspan="4">-</td><td colspan="4">-</td></tr>
<tr><td>Initial value</td><td colspan="4">0x3</td><td colspan="4">0x3</td></tr>
</table>
</td></tr>
<tr><td>1730</td><td>CHAPTER 31:CAN FD Controller
5.3. Data Bit Timing & Prescaler Register (DBTP)</td><td>
Features of "For details" should be corrected as indicated by the shading below
(Error)
[bit12:8] DTSEG1[3:0]: Data Time Segment before sample Point

(Correct)
[bit12:8] DTSEG1[4:0]: Data Time Segment before sample Point
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1756</td><td>CHAPTER 31:CAN FD Controller 5.14. Transmitter Delay Compensation Register (TDCR)</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">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 colspan="8">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>
(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>Reserved</td><td colspan="7">TDCO[6:0]</td></tr>
<tr><td>R/W Attribute</td><td>R/W</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">0000000</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>R/W</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">0000000</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>2660</td><td>CHAPTER 40:Base Timer Simultaneous Operation

6.Registers of the 16-Bit Global Timer</td><td>Added Table 6-1 List of Register Mirror Area as follows
(Correct)
Table 6-1 List of Register Mirror Area

<table>
<tr><th>Address of Register Mirror Area</th><th>Mirrored Peripheral</th><th>Mirror Area Behavior</th><th>PPU</th></tr>
<tr><td>B484_9040 to B484_907C</td><td rowspan="3">Global Timer</td><td rowspan="3">Accessing this region results to the access to Global Timer register area.(B484_9000 to B484903C)</td><td rowspan="3">This area is covered by PPU #300 as well as the mirrored peripheral.</td></tr>
<tr><td>B484_9080 to B484_90BC</td></tr>
<tr><td>B484_90C0 to B484_90FC</td></tr>
</table>
</td></tr>
<tr><td>3687</td><td>Appendix: I/O Map</td><td>Features of "For details" should be corrected as indicated by the shading below
(Error)

<table>
<tr><td>B484_9020</td><td></td><td>GT_MSCTR [R/W] B,H,
-----000</td><td rowspan="3"></td></tr>
<tr><td>B484_9024</td><td colspan="2">GT_MSCHB[R/W] B,H,W
00000000 00000000 00000000 00000000</td></tr>
<tr><td>B484_9028</td><td colspan="2">GT_MSCH[R/W] B,H,W
00000000 00000000 00000000 00000000</td></tr>
<tr><td>B484_902C
to
B484_FFFC</td><td colspan="2">–</td><td></td></tr>
</table>

(Correct)

<table>
<tr><td>B484_9020</td><td></td><td>GT_MSCTR [R/W] B,H,
-----000</td><td rowspan="4"></td></tr>
<tr><td>B484_9024</td><td colspan="2">GT_MSCHB[R/W] W
-0000000 00000000 00000000 00000000</td></tr>
<tr><td>B484_9028</td><td colspan="2">-</td></tr>
<tr><td>B484_902C</td><td colspan="2">GT_MSCH[R/W] W
-0000000 00000000 00000000 00000000</td></tr>
<tr><td>B484_9030
to
B484_FFFC</td><td colspan="2">–</td><td></td></tr>
</table>
</td></tr>
</table>

Page	Section	Change Results
3690	Appendix : DMA Channel Activation Factors	Revised DMA Channel Activation Factors as follows (Error) **1. DMA Channel Activation Factors (S6J312xHzB*)** This section shows DMA channel activation factors in S6J312xHzB*. DMA sources from the peripherals in S6J312xHzB*.

Peri / Function	Sync or Async to DMAC	Type	Number of Channels
Reserved	-	-	0 to 7
Ext-IRQ	Async	Type1	8
Work Flash DREQ	Sync	Type1	9
FRT ch.0 Compare clear	Async	Type2	10
FRT ch.0 Zero detection	Async	Type2	11
FRT ch.1 Compare clear	Async	Type2	12
FRT ch.1 Zero detection	Async	Type2	13
FRT ch.2 Compare clear	Async	Type2	14
FRT ch.2 Zero detection	Async	Type2	15
FRT ch.3 Compare clear	Async	Type2	16
FRT ch.3 Zero detection	Async	Type2	17
FRT ch.4 Compare clear	Async	Type2	18
FRT ch.4 Zero detection	Async	Type2	19
FRT ch.5 Compare clear	Async	Type2	20
FRT ch.5 Zero detection	Async	Type2	21
ICU ch.0 DREQ	Async	Type2	22
ICU ch.1 DREQ	Async	Type2	23
ICU ch.2 DREQ	Async	Type2	24
ICU ch.3 DREQ	Async	Type2	25
ICU ch.4 DREQ	Async	Type2	26
ICU ch.5 DREQ	Async	Type2	27
ICU ch.6 DREQ	Async	Type2	28
ICU ch.7 DREQ	Async	Type2	29
ICU ch.8 DREQ	Async	Type2	30
ICU ch.9 DREQ	Async	Type2	31
ICU ch.10 DREQ	Async	Type2	32
ICU ch.11 DREQ	Async	Type2	33
OCU ch.0 DREQ	Async	Type2	34
OCU ch.1 DREQ	Async	Type2	35
OCU ch.2 DREQ	Async	Type2	36
OCU ch.3 DREQ	Async	Type2	37
OCU ch.4 DREQ	Async	Type2	38
OCU ch.5 DREQ	Async	Type2	39
OCU ch.6 DREQ	Async	Type2	40
OCU ch.7 DREQ	Async	Type2	41
OCU ch.8 DREQ	Async	Type2	42
OCU ch.9 DREQ	Async	Type2	43
OCU ch.10 DREQ	Async	Type2	44
OCU ch.11 DREQ	Async	Type2	45
Base Timer ch.0-0 DREQ	Async	Type2	46

(Correct)

1. DMA Channel Activation Factors (S6J312xHzC*)

This section shows DMA channel activation factors in S6J312xHzC*.

DMA sources from the peripherals in S6J312xHzC*.

Number of Channels	Peripheral Functions	Type
0 to 7	Reserved	-
8	Ext-IRQ	1
9	Work Flash DREQ	1
10	FRT ch.0 Compare clear	2
11	FRT ch.0 Zero detection	2
12	FRT ch.1 Compare clear	2
13	FRT ch.1 Zero detection	2
14	FRT ch.2 Compare clear	2
15	FRT ch.2 Zero detection	2
16	FRT ch.3 Compare clear	2
17	FRT ch.3 Zero detection	2
18	FRT ch.4 Compare clear	2
19	FRT ch.4 Zero detection	2
20	FRT ch.5 Compare clear	2
21	FRT ch.5 Zero detection	2
22	ICU ch.0 DREQ	2
23	ICU ch.1 DREQ	2
24	ICU ch.2 DREQ	2
25	ICU ch.3 DREQ	2
26	ICU ch.4 DREQ	2
27	ICU ch.5 DREQ	2
28	ICU ch.6 DREQ	2
29	ICU ch.7 DREQ	2
30	ICU ch.8 DREQ	2
31	ICU ch.9 DREQ	2
32	ICU ch.10 DREQ	2
33	ICU ch.11 DREQ	2
34	OCU ch.0 DREQ	2
35	OCU ch.1 DREQ	2
36	OCU ch.2 DREQ	2
37	OCU ch.3 DREQ	2
38	OCU ch.4 DREQ	2
39	OCU ch.5 DREQ	2
40	OCU ch.6 DREQ	2
41	OCU ch.7 DREQ	2
42	OCU ch.8 DREQ	2
43	OCU ch.9 DREQ	2
44	OCU ch.10 DREQ	2
45	OCU ch.11 DREQ	2
46	Base Timer ch.0-0 DREQ	2

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>3691</td><td>Appendix : DMA Channel Activation Factors</td><td>Revised DMA Channel Activation Factors as follows
(Error)
<table>
<tr><th>Peri / Function</th><th>Sync or Async to DMAC</th><th>Type</th><th>Number of Channels</th></tr>
<tr><td>Base Timer ch.0-1 DREQ</td><td>Async</td><td>Type2</td><td>47</td></tr>
<tr><td>Base Timer ch.1-0 DREQ</td><td>Async</td><td>Type2</td><td>48</td></tr>
<tr><td>Base Timer ch.1-1 DREQ</td><td>Async</td><td>Type2</td><td>49</td></tr>
<tr><td>Base Timer ch.2-0 DREQ</td><td>Async</td><td>Type2</td><td>50</td></tr>
<tr><td>Base Timer ch.2-1 DREQ</td><td>Async</td><td>Type2</td><td>51</td></tr>
<tr><td>Base Timer ch.3-0 DREQ</td><td>Async</td><td>Type2</td><td>52</td></tr>
<tr><td>Base Timer ch.3-1 DREQ</td><td>Async</td><td>Type2</td><td>53</td></tr>
<tr><td>Base Timer ch.4-0 DREQ</td><td>Async</td><td>Type2</td><td>54</td></tr>
<tr><td>Base Timer ch.4-1 DREQ</td><td>Async</td><td>Type2</td><td>55</td></tr>
<tr><td>Base Timer ch.5-0 DREQ</td><td>Async</td><td>Type2</td><td>56</td></tr>
<tr><td>Base Timer ch.5-1 DREQ</td><td>Async</td><td>Type2</td><td>57</td></tr>
<tr><td>Base Timer ch.6-0 DREQ</td><td>Async</td><td>Type2</td><td>58</td></tr>
<tr><td>Base Timer ch.6-1 DREQ</td><td>Async</td><td>Type2</td><td>59</td></tr>
<tr><td>Base Timer ch.7-0 DREQ</td><td>Async</td><td>Type2</td><td>60</td></tr>
<tr><td>Base Timer ch.7-1 DREQ</td><td>Async</td><td>Type2</td><td>61</td></tr>
<tr><td>Base Timer ch.8-0 DREQ</td><td>Async</td><td>Type2</td><td>62</td></tr>
<tr><td>Base Timer ch.8-1 DREQ</td><td>Async</td><td>Type2</td><td>63</td></tr>
<tr><td>Base Timer ch.9-0 DREQ</td><td>Async</td><td>Type2</td><td>64</td></tr>
<tr><td>Base Timer ch.9-1 DREQ</td><td>Async</td><td>Type2</td><td>65</td></tr>
<tr><td>Base Timer ch.10-0 DREQ</td><td>Async</td><td>Type2</td><td>66</td></tr>
<tr><td>Base Timer ch.10-1 DREQ</td><td>Async</td><td>Type2</td><td>67</td></tr>
<tr><td>Base Timer ch.11-0 DREQ</td><td>Async</td><td>Type2</td><td>68</td></tr>
<tr><td>Base Timer ch.11-1 DREQ</td><td>Async</td><td>Type2</td><td>69</td></tr>
<tr><td>MFS ch.0 Tx DREQ</td><td>Async</td><td>Type2</td><td>70</td></tr>
<tr><td>MFS ch.0 Rx DREQ</td><td>Async</td><td>Type2</td><td>71</td></tr>
<tr><td>MFS ch.1 Tx DREQ</td><td>Async</td><td>Type2</td><td>72</td></tr>
<tr><td>MFS ch.1 Rx DREQ</td><td>Async</td><td>Type2</td><td>73</td></tr>
<tr><td>MFS ch.2 Tx DREQ</td><td>Async</td><td>Type2</td><td>74</td></tr>
<tr><td>MFS ch.2 Rx DREQ</td><td>Async</td><td>Type2</td><td>75</td></tr>
<tr><td>MFS ch.3 Tx DREQ</td><td>Async</td><td>Type2</td><td>76</td></tr>
<tr><td>MFS ch.3 Rx DREQ</td><td>Async</td><td>Type2</td><td>77</td></tr>
<tr><td>MFS ch.4 Tx DREQ</td><td>Async</td><td>Type2</td><td>78</td></tr>
<tr><td>MFS ch.4 Rx DREQ</td><td>Async</td><td>Type2</td><td>79</td></tr>
<tr><td>Reserved</td><td>-</td><td>-</td><td>80 to 85</td></tr>
<tr><td>A/D unit.0 DREQ</td><td>Async</td><td>Type2</td><td>86</td></tr>
<tr><td>A/D unit.1 DREQ</td><td>Async</td><td>Type2</td><td>87</td></tr>
<tr><td>CAN-FD ch.0 DREQ</td><td>Async</td><td>Type1</td><td>88</td></tr>
<tr><td>CAN-FD ch.1 DREQ</td><td>Async</td><td>Type1</td><td>89</td></tr>
<tr><td>DMAC RLT_DREQ ch.0</td><td>Sync</td><td>Type1</td><td>90</td></tr>
<tr><td>DMAC RLT_DREQ ch.1</td><td>Sync</td><td>Type1</td><td>91</td></tr>
<tr><td>DMAC RLT_DREQ ch.2</td><td>Sync</td><td>Type1</td><td>92</td></tr>
<tr><td>DMAC RLT_DREQ ch.3</td><td>Sync</td><td>Type1</td><td>93</td></tr>
<tr><td>Reserved</td><td>-</td><td>-</td><td>94 to 119</td></tr>
<tr><td>CAN-FD ch.2 DREQ</td><td>Async</td><td>Type1</td><td>120</td></tr>
<tr><td>Reserved</td><td>-</td><td>-</td><td>121 to 123</td></tr>
<tr><td>HSSPI TX-FIFO DREQ</td><td>Async</td><td>Type1</td><td>124</td></tr>
</table>
(Correct)
<table>
<tr><th>Number of Channels</th><th>Peri / Function</th><th>Type</th></tr>
<tr><td>47</td><td>Base Timer ch.0-1 DREQ</td><td>2</td></tr>
<tr><td>48</td><td>Base Timer ch.1-0 DREQ</td><td>2</td></tr>
<tr><td>49</td><td>Base Timer ch.1-1 DREQ</td><td>2</td></tr>
<tr><td>50</td><td>Base Timer ch.2-0 DREQ</td><td>2</td></tr>
<tr><td>51</td><td>Base Timer ch.2-1 DREQ</td><td>2</td></tr>
<tr><td>52</td><td>Base Timer ch.3-0 DREQ</td><td>2</td></tr>
<tr><td>53</td><td>Base Timer ch.3-1 DREQ</td><td>2</td></tr>
<tr><td>54</td><td>Base Timer ch.4-0 DREQ</td><td>2</td></tr>
<tr><td>55</td><td>Base Timer ch.4-1 DREQ</td><td>2</td></tr>
<tr><td>56</td><td>Base Timer ch.5-0 DREQ</td><td>2</td></tr>
<tr><td>57</td><td>Base Timer ch.5-1 DREQ</td><td>2</td></tr>
<tr><td>58</td><td>Base Timer ch.6-0 DREQ</td><td>2</td></tr>
<tr><td>59</td><td>Base Timer ch.6-1 DREQ</td><td>2</td></tr>
<tr><td>60</td><td>Base Timer ch.7-0 DREQ</td><td>2</td></tr>
<tr><td>61</td><td>Base Timer ch.7-1 DREQ</td><td>2</td></tr>
<tr><td>62</td><td>Base Timer ch.8-0 DREQ</td><td>2</td></tr>
<tr><td>63</td><td>Base Timer ch.8-1 DREQ</td><td>2</td></tr>
<tr><td>64</td><td>Base Timer ch.9-0 DREQ</td><td>2</td></tr>
<tr><td>65</td><td>Base Timer ch.9-1 DREQ</td><td>2</td></tr>
<tr><td>66</td><td>Base Timer ch.10-0 DREQ</td><td>2</td></tr>
<tr><td>67</td><td>Base Timer ch.10-1 DREQ</td><td>2</td></tr>
<tr><td>68</td><td>Base Timer ch.11-0 DREQ</td><td>2</td></tr>
<tr><td>69</td><td>Base Timer ch.11-1 DREQ</td><td>2</td></tr>
<tr><td>70</td><td>MFS ch.0 Tx DREQ</td><td>3</td></tr>
<tr><td>71</td><td>MFS ch.0 Rx DREQ</td><td>3</td></tr>
<tr><td>72</td><td>MFS ch.1 Tx DREQ</td><td>3</td></tr>
<tr><td>73</td><td>MFS ch.1 Rx DREQ</td><td>3</td></tr>
<tr><td>74</td><td>MFS ch.2 Tx DREQ</td><td>3</td></tr>
<tr><td>75</td><td>MFS ch.2 Rx DREQ</td><td>3</td></tr>
<tr><td>76</td><td>MFS ch.3 Tx DREQ</td><td>3</td></tr>
<tr><td>77</td><td>MFS ch.3 Rx DREQ</td><td>3</td></tr>
<tr><td>78</td><td>MFS ch.4 Tx DREQ</td><td>3</td></tr>
<tr><td>79</td><td>MFS ch.4 Rx DREQ</td><td>3</td></tr>
<tr><td>80 to 85</td><td>Reserved</td><td>-</td></tr>
<tr><td>86</td><td>A/D unit.0 DREQ</td><td>2</td></tr>
<tr><td>87</td><td>A/D unit.1 DREQ</td><td>2</td></tr>
<tr><td>88</td><td>Reserved</td><td>-</td></tr>
<tr><td>89</td><td>Reserved</td><td>-</td></tr>
<tr><td>90</td><td>DMAC RLT_DREQ ch.0</td><td>1</td></tr>
<tr><td>91</td><td>DMAC RLT_DREQ ch.1</td><td>1</td></tr>
<tr><td>92</td><td>DMAC RLT_DREQ ch.2</td><td>1</td></tr>
<tr><td>93</td><td>DMAC RLT_DREQ ch.3</td><td>1</td></tr>
<tr><td>94 to 119</td><td>Reserved</td><td>-</td></tr>
<tr><td>120</td><td>Reserved</td><td>-</td></tr>
<tr><td>121 to 123</td><td>Reserved</td><td>-</td></tr>
<tr><td>124</td><td>HSSPI TX-FIFO DREQ</td><td>1</td></tr>
</table>
</td></tr>
</table>

Page	Section	Change Results
3692	Appendix : DMA Channel Activation Factors	Revised DMA Channel Activation Factors as follows (Error)

Peri / Function	Sync or Async to DMAC	Type	Number of Channels
HSSPI RX-FIFO DREQ	Async	Type1	125
Reserved	-	-	126 to 135
RLT ch.0 DREQ	Async	Type1	136
RLT ch.1 DREQ	Async	Type1	137
RLT ch.2 DREQ	Async	Type1	138
RLT ch.3 DREQ	Async	Type1	139
Reserved	-	-	140 to 151
RLT ch.16 DREQ	Async	Type1	152
RLT ch.17 DREQ	Async	Type1	153
RLT ch.18 DREQ	Async	Type1	154
RLT ch.19 DREQ	Async	Type1	155
Reserved	-	-	156 to 167
RLT ch.32 DREQ	Sync	Type1	168
RLT ch.33 DREQ	Sync	Type1	169
Reserved	-	-	170 to 199
MFS ch.8 RX DREQ	Async	Type2	200
MFS ch.8 TX DREQ	Async	Type2	201
MFS ch.9 RX DREQ	Async	Type2	202
MFS ch.9 TX DREQ	Async	Type2	203
MFS ch.10 RX DREQ	Async	Type2	204
MFS ch.10 TX DREQ	Async	Type2	205
MFS ch.11 RX DREQ	Async	Type2	206
MFS ch.11 TX DREQ	Async	Type2	207
MFS ch.12 RX DREQ	Async	Type2	208
MFS ch.12 TX DREQ	Async	Type2	209
Reserved	-	-	210 to 328
Sound Generator ch.0	Async	Type2	329
Sound Generator ch.1	Async	Type2	330
Sound Generator ch.2	Async	Type2	331
Reserved	-	-	332 to 511

Note:
- *Type1: Native DMA request Type2: DMA request by IRQ*

*x: A/9/8, z: A/B

(Correct)

Number of Channels	Peri / Function	Type
125	HSSPI RX-FIFO DREQ	1
126 to 135	Reserved	-
136	RLT ch.0 DREQ	1
137	RLT ch.1 DREQ	1
138	RLT ch.2 DREQ	1
139	RLT ch.3 DREQ	1
140 to 151	Reserved	-
152	RLT ch.16 DREQ	1
153	RLT ch.17 DREQ	1
154	RLT ch.18 DREQ	1
155	RLT ch.19 DREQ	1
156 to 167	Reserved	-
168	RLT ch.32 DREQ	1
169	RLT ch.33 DREQ	1
170 to 199	Reserved	-
200	MFS ch.8 RX DREQ	3
201	MFS ch.8 TX DREQ	3
202	MFS ch.9 RX DREQ	3
203	MFS ch.9 TX DREQ	3
204	MFS ch.10 RX DREQ	3
205	MFS ch.10 TX DREQ	3
206	MFS ch.11 RX DREQ	3
207	MFS ch.11 TX DREQ	3
208	MFS ch.12 RX DREQ	3
209	MFS ch.12 TX DREQ	3
210 to 328	Reserved	-
329	Sound Generator ch.0	2
330	Sound Generator ch.1	2
331	Sound Generator ch.2	2
332 to 511	Reserved	-

* x: A/9/8, z:A/B

Type	Automatically cleared the interrupt factor flag of peripheral function by acceptance of DMA transfer request.	Interrupt Enable
1	Not cleared.	Unnecessary(*1)
2	Cleared	Necessary
3(*2)	Not cleared	Necessary

*1: Necessary to enable DMA.

*2: Only Multi Function Serial function. The interrupt flag which is the factor of DMA transfer request is cleared when data transfer is started.

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>3699</td><td>Appendix : List of Interrupt Vector and NMI Map</td><td>Features of "For details" should be corrected as indicated by the shading below
(Error)
<table><tr><td>313</td><td>BPC IRQs</td><td>IRCn_IRQPL78:IRQPL313[4:0]</td><td>IRCn_IRQVA313</td></tr></table>
(Correct)
<table><tr><td>313</td><td>reserved</td></tr></table></td></tr>
<tr><td>-</td><td>-</td><td>Revised part number from S6J312xxxB to S6J312xxxC.</td></tr>
<tr><td colspan="3">Revision *C</td></tr>
<tr><td>306</td><td>CHAPTER 6:Low Power Consumption 5.2.9. RUN Low-Voltage Detection Setting Register (SYSC0_RUNLVDCFGR)</td><td>The register bit number should be corrected as indicated by the shading below.
(Error)
[bit26:25] LVDL1V: Internal Low-Voltage Detection Voltage Setting Bits
These bits set the detection voltage for internal low-voltage detection.
<table><tr><th>bit18:17</th><th>Description</th></tr><tr><td>00</td><td>Reserved</td></tr><tr><td>01</td><td>Reserved</td></tr><tr><td>10</td><td>0.97±0.05V</td></tr><tr><td>11</td><td>1.07±0.05V</td></tr></table>Note:
−After the detection voltage is changed, low-voltage detection is unavailable until the low-voltage detection stability time elapses.

(Correct)
[bit26:25] LVDL1V: Internal Low-Voltage Detection Voltage Setting Bits
These bits set the detection voltage for internal low-voltage detection.
<table><tr><th>Bit</th><th>Description</th><th>Guaranteed MCU operation range</th></tr><tr><td>00</td><td>Reserved</td><td rowspan="4">No</td></tr><tr><td>01</td><td>Reserved</td></tr><tr><td>10*</td><td>0.97±0.05V</td></tr><tr><td>11*</td><td>1.07±0.05V</td></tr></table>Note:
−After the detection voltage is changed, low-voltage detection is unavailable until the low-voltage detection stability time elapses.
*: This LVD cannot be used to reliably generate a reset before voltage dips below minimum guaranteed MCU operation voltage,as these detection levels are below the minimum guaranteed MCU operation voltage.</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>335</td><td>CHAPTER 6:Low Power Consumption 5.3.9. PSS Low-Voltage Detection Setting Register (SYSC0_PSSLVDCFGR)</td><td>
The register bit number should be corrected as indicated by the shading below.
(Error)
[bit26:25] LVDL1V: Internal Low-Voltage Detection Voltage Setting Bits
These bits set the detection voltage for internal low-voltage detection.
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>00</td><td>Reserved</td></tr>
<tr><td>01</td><td>Reserved</td></tr>
<tr><td>10</td><td>0.97±0.05V</td></tr>
<tr><td>11</td><td>1.07±0.05V</td></tr>
</table>
Note:
−After the detection voltage is changed, low-voltage detection is unavailable until the low-voltage detection stability time elapses.

(Correct)
[bit26:25] LVDL1V: Internal Low-Voltage Detection Voltage Setting Bits
These bits set the detection voltage for internal low-voltage detection.
<table>
<tr><th>Bit</th><th>Description</th><th>Guaranteed MCU operation range</th></tr>
<tr><td>00</td><td>Reserved</td><td rowspan="4">No</td></tr>
<tr><td>01</td><td>Reserved</td></tr>
<tr><td>10*</td><td>0.97±0.05V</td></tr>
<tr><td>11*</td><td>1.07±0.05V</td></tr>
</table>
Note:
−After the detection voltage is changed, low-voltage detection is unavailable until the low-voltage detection stability time elapses.
*: This LVD cannot be used to reliably generate a reset before voltage dips below minimum guaranteed MCU operation voltage, as these detection levels are below the minimum guaranteed MCU operation voltage.
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>365</td><td>CHAPTER 6:Low Power Consumption 5.4.9. APP Low-Voltage Detection Setting Register (SYSC0_APPLVDCFGR)</td><td>
The register bit number should be corrected as indicated by the shading below.
(Error)
[bit26:25] LVDL1V: Internal Low-Voltage Detection Voltage Setting Bits
These bits set the detection voltage for internal low-voltage detection.
<table>
<tr><th>Bit26:25</th><th>Description</th></tr>
<tr><td>00</td><td>Reserved</td></tr>
<tr><td>01</td><td>Reserved</td></tr>
<tr><td>10</td><td>0.97±0.05V</td></tr>
<tr><td>11</td><td>1.07±0.05V</td></tr>
</table>
Note:
−After the detection voltage is changed, low-voltage detection is unavailable until the low-voltage detection stability time elapses.

(Correct)
[bit26:25] LVDL1V: Internal Low-Voltage Detection Voltage Setting Bits
These bits set the detection voltage for internal low-voltage detection.
<table>
<tr><th>Bit26:25</th><th>Description</th><th>Guaranteed MCU operation range</th></tr>
<tr><td>00</td><td>Reserved</td><td rowspan="4">No</td></tr>
<tr><td>01</td><td>Reserved</td></tr>
<tr><td>10*</td><td>0.97±0.05V</td></tr>
<tr><td>11*</td><td>1.07±0.05V</td></tr>
</table>
Note:
−After the detection voltage is changed, low-voltage detection is unavailable until the low-voltage detection stability time elapses.
*: This LVD cannot be used to reliably generate a reset before voltage dips below minimum guaranteed MCU operation voltage, as these detection levels are below the minimum guaranteed MCU operation voltage.
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>396</td><td>CHAPTER 6:Low Power Consumption
5.5.9. STS Low-Voltage Detection Setting Register (SYSC0_STSLVDCFGR)</td><td>The register bit number should be corrected as indicated by the shading below.
(Error)
[bit26:25] LVDL1V: Internal Low-Voltage Detection Voltage Setting Bits
These bits set the detection voltage for internal low-voltage detection.
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>00</td><td>Reserved</td></tr>
<tr><td>01</td><td>Reserved</td></tr>
<tr><td>10</td><td>0.97±0.05V</td></tr>
<tr><td>11</td><td>1.07±0.05V</td></tr>
</table>
Note:
−After the detection voltage is changed, low-voltage detection is unavailable until the low-voltage detection stability time elapses.

(Correct)
[bit26:25] LVDL1V: Internal Low-Voltage Detection Voltage Setting Bits
These bits set the detection voltage for internal low-voltage detection.
<table>
<tr><th>Bit</th><th>Description</th><th>Guaranteed MCU operation range</th></tr>
<tr><td>00</td><td>Reserved</td><td rowspan="4">No</td></tr>
<tr><td>01</td><td>Reserved</td></tr>
<tr><td>10*</td><td>0.97±0.05V</td></tr>
<tr><td>11*</td><td>1.07±0.05V</td></tr>
</table>
Note:
−After the detection voltage is changed, low-voltage detection is unavailable until the low-voltage detection stability time elapses.
*: This LVD cannot be used to reliably generate a reset before voltage dips below minimum guaranteed MCU operation voltage, as these detection levels are below the minimum guaranteed MCU operation voltage.</td></tr>
</table>

Page	Section	Change Results
Revision *D		
50	Introduction: Restrictions On Operation Specifications 3.About Low-Power Consumption Specifications	Features of "For details" should be added as indicated by the shading below. (Error) □This model does not support the following functions. −Sub CLK −LVDL2 (extended internal low-voltage detection) −LVDH2 (extended external low-voltage detection) −SSCGPLL1 to 3 (SSCGPLLx=0 is used.) −PLL1 to 3 (PLLx=0 is used.) −3 V I/O Register functions related to the items above are also not supported. (Correct) □This model does not support the following functions. −Sub CLK −LVDL2 (extended internal low-voltage detection) −LVDH2 (extended external low-voltage detection) −SSCGPLL1 to 3 (SSCGPLLx=0 is used.) −PLL1 to 3 (PLLx=0 is used.) −3 V I/O −External Power Supply Stabilization Time Setting(EXSTBCNT):Because there is no 1.2V external power supply Register functions related to the items above are also not supported.

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>112</td><td>CHAPTER2
CPU
2.Notes</td><td>Features of "For details" should be added as indicated by the shading below.
(Correct)
Build Option
The build option of Cortex-R5 is shown below.
Table 2 1 Cortex-R5 Build Option
<table>
<tr><th>Feature</th><th>Options</th><th>Sub-options</th><th>Configuration</th></tr>
<tr><td rowspan="5">Number of CPUs</td><td>Single-CPU
(no redundancy)</td><td>-</td><td rowspan="5">Single-CPU
(no redundancy)</td></tr>
<tr><td>Redundant CPU</td><td>-</td></tr>
<tr><td>Twin-CPU (no redundancy)</td><td>-</td></tr>
<tr><td rowspan="2">Split/Lock</td><td>Safety-mode
(redundancy)</td></tr>
<tr><td>Performance-mode
(twin CPU)</td></tr>
<tr><td rowspan="9">Instruction cache</td><td>No I-Cache</td><td>-</td><td>-</td></tr>
<tr><td rowspan="8">I-Cache included</td><td>No error checking</td><td rowspan="3">64-bit ECC error checking</td></tr>
<tr><td>Parity error checking</td></tr>
<tr><td>64-bit ECC error checking</td></tr>
<tr><td>4KB (4x1KB ways)</td><td rowspan="5">16KB(4x4KB ways)</td></tr>
<tr><td>8KB (4x2KB ways)</td></tr>
<tr><td>16KB (4x4KB ways)</td></tr>
<tr><td>32KB (4x8KB ways)</td></tr>
<tr><td>64KB (4x16KB ways)</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>112</td><td>CHAPTER2
CPU
2.Notes</td><td>Features of "For details" should be added as indicated by the shading below.
(Correct)
<table>
<tr><td rowspan="5">Data cache</td><td>No D-Cache</td><td>-</td><td>-</td></tr>
<tr><td rowspan="4">D-Cache included</td><td>No error checking</td><td rowspan="3">32-bit ECC error checking</td></tr>
<tr><td>Parity error checking</td></tr>
<tr><td>32-bit ECC error checking</td></tr>
<tr><td>4KB (4x1KB ways)
8KB (4x2KB ways)
16KB (4x4KB ways)
32KB (4x8KB ways)
64KB (4x16KB ways)</td><td>16KB(4x4KB ways)</td></tr>
</table></td></tr>
<tr><td>113</td><td>CHAPTER2
CPU
2.Notes</td><td>
<table>
<tr><td rowspan="5">ATCM</td><td>No ATCM ports</td><td>-</td><td>-</td></tr>
<tr><td rowspan="4">One ATCM port</td><td>No error checking</td><td rowspan="3">64-bit ECC error checking</td></tr>
<tr><td>32-bit ECC error checking</td></tr>
<tr><td>64-bit ECC error checking</td></tr>
<tr><td>4KB, 8KB, 16KB, 32KB, 64KB, 128KB, 256KB, 512KB, 1MB, 2MB, 4MB, or 8MB</td><td>8MB</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>113</td><td>CHAPTER2
CPU
2.Notes</td><td>Features of "For details" should be added as indicated by the shading below.
(Correct)
<table>
<tr><td rowspan="10">BTCM</td><td>No BTCM ports</td><td>-</td><td>-</td></tr>
<tr><td rowspan="3">One BTCM port
(B0TCM)</td><td>No error checking</td><td rowspan="3">-</td></tr>
<tr><td>32-bit ECC error checking</td></tr>
<tr><td>64-bit ECC error checking</td></tr>
<tr><td>4KB, 8KB, 16KB, 32KB, 64KB, 128KB, 256KB, 512KB, 1MB, 2MB, 4MB, or 8MB</td><td>-</td></tr>
<tr><td rowspan="6">Two BTCM ports
(B0TCM and B1TCM)</td><td>No error checking</td><td rowspan="3">32-bit ECC error checking</td></tr>
<tr><td>32-bit ECC error checking</td></tr>
<tr><td>64-bit ECC error checking</td></tr>
<tr><td>2x2KB, 2x4KB, 2x8KB, 2x16KB, 2x32KB, 2x64KB, 2x128KB, 2x256KB, 2x512KB, 2x1MB, 2x2MB, or 2x4MB</td><td>2x32KB</td></tr>
<tr><td>Interleaved on 64-bit granularity in memory</td><td rowspan="2">Interleaved on 64-bit granularity in memory</td></tr>
<tr><td>Adjacent in memory</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>114</td><td>CHAPTER2
CPU
2.Notes</td><td>Features of "For details" should be added as indicated by the shading below.
(Correct)
<table>
<tr><td rowspan="2">Instruction endianness</td><td>Little-endian</td><td>-</td><td rowspan="2">Little-endian</td></tr>
<tr><td>Pin-configured</td><td>Little-endian
Big-endian</td></tr>
<tr><td rowspan="2">Floating point (VFP)</td><td>No FPU</td><td>-</td><td rowspan="2">No FPU</td></tr>
<tr><td>FPU included</td><td>Full implementation
Single-precision only</td></tr>
<tr><td rowspan="2">MPU</td><td>No MPU</td><td>-</td><td rowspan="2">16 MPU regions</td></tr>
<tr><td>MPU included</td><td>12 MPU regions
16 MPU regions</td></tr>
<tr><td rowspan="2">TCM bus parity</td><td>No TCM address and control bus parity</td><td>-</td><td rowspan="2">No TCM address and control bus parity</td></tr>
<tr><td>TCM address and control bus parity generated</td><td>-</td></tr>
<tr><td rowspan="2">AXI bus ECC/parity on AXI-master, AXI-slave (if included) and ACP (if included)</td><td>No AXI bus ECC/parity</td><td>-</td><td rowspan="2">No AXI bus ECC/parity</td></tr>
<tr><td>AXI bus ECC/parity generated/ checked</td><td>-</td></tr>
<tr><td rowspan="2">Bus ECC/parity on AXI peripheral port and AHB peripheral port (if included)</td><td>No peripheral port bus ECC/parity</td><td>-</td><td rowspan="2">No peripheral port bus ECC/parity</td></tr>
<tr><td>Peripheral port bus ECC/parity generated/checked</td><td>-</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>114</td><td>CHAPTER2
CPU
2.Notes</td><td>Features of "For details" should be added as indicated by the shading below.
(Correct)
<table>
<tr><td>Breakpoints</td><td>2-8 breakpoint register pairs</td><td>-</td><td>8</td></tr>
<tr><td>Watchpoints</td><td>1-8 watchpoint registers</td><td>-</td><td>8</td></tr>
<tr><td rowspan="3">ATCM at reset</td><td>Disabled</td><td>-</td><td rowspan="3">Base address configured</td></tr>
<tr><td rowspan="2">Enabled</td><td>Base address 0x0</td></tr>
<tr><td>Base address configured</td></tr>
<tr><td rowspan="3">BTCM at reset</td><td>Disabled</td><td>-</td><td rowspan="3">Base address 0x0</td></tr>
<tr><td rowspan="2">Enabled</td><td>Base address configured</td></tr>
<tr><td>Base address 0x0</td></tr>
<tr><td>Peripheral ID RevAnd field</td><td>Any 4-bit value</td><td>-</td><td>0000</td></tr>
</table></td></tr>
<tr><td>115</td><td>CHAPTER2
CPU
2.Notes</td><td>Features of "For details" should be added as indicated by the shading below.
(Correct)
<table>
<tr><td rowspan="2">AXI slave interface</td><td>No AXI-slave</td><td>-</td><td rowspan="2">AXI-slave included</td></tr>
<tr><td>AXI-slave included</td><td>-</td></tr>
<tr><td rowspan="2">TCM Hard Error Cache</td><td>No TCM Hard Error Cache</td><td>-</td><td rowspan="2">TCM Hard Error Cache included</td></tr>
<tr><td>TCM Hard Error Cache included</td><td>-</td></tr>
<tr><td rowspan="2">Non-Maskable FIQ Interrupt</td><td>Disabled (FIQ can be masked by software)</td><td>-</td><td rowspan="2">Enabled</td></tr>
<tr><td>Enabled</td><td>-</td></tr>
<tr><td rowspan="2">Parity type</td><td>Odd parity</td><td>-</td><td rowspan="2">Odd parity</td></tr>
<tr><td>Even parity</td><td>-</td></tr>
<tr><td rowspan="2">AXI coherency port (ACP)</td><td>No ACP</td><td>-</td><td rowspan="2">No ACP</td></tr>
<tr><td>ACP included</td><td>-</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>115</td><td>CHAPTER2
CPU
2.Notes</td><td>Features of "For details" should be added as indicated by the shading below.
(Correct)
<table>
<tr><td rowspan="3">AHB peripheral port</td><td>AXI peripheral port only</td><td>-</td><td>-</td></tr>
<tr><td rowspan="2">AXI and AHB peripheral ports</td><td>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</td><td>64MB</td></tr>
<tr><td>AHB peripheral port base address: any size-aligned address</td><td>0xB400_0</td></tr>
<tr><td>AXI peripheral interface region size</td><td>4KB, 8KB, 16KB, 32KB, 64KB, 128KB, 256KB, 512KB, 1MB, 2MB, 4MB, or 8MB. 16MB, 32MB, 64MB, 128MB, 256MB, 512MB, 1GB, 2GB, 4GB</td><td>-</td><td>64MB</td></tr>
<tr><td>AXI peripheral interface base address</td><td>Any size-aligned address</td><td>-</td><td>0xB000_0</td></tr>
</table></td></tr>
<tr><td>116</td><td>CHAPTER2
CPU
2.Notes</td><td>Features of "For details" should be added as indicated by the shading below.
(Correct)
<table>
<tr><td>Virtual AXI peripheral interface region size</td><td>4KB, 8KB, 16KB, 32KB, 64KB, 128KB, 256KB, 512KB, 1MB, 2MB, 4MB, or 8MB. 16MB, 32MB, 64MB, 128MB, 256MB, 512MB, 1GB, 2GB, 4GB</td><td>-</td><td>-</td></tr>
<tr><td>Virtual AXI peripheral interface base address</td><td>Any size-aligned address</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>177</td><td>CHAPTER 4:Reset
4.2. User Reset Factor Register (SYSC_RSTCAUSEUR)</td><td>Features of "For details" should be added as indicated by the shading below
Bit12 SWDR: Software Watchdog Reset Detection Bit
(Correct)
When any of the following reset (RVD, INITX, IMR, LVDL1R, HWDR) happens with PSS mode shutdown (PD2:OFF), This bit takes the following value.
SWDR value
<table><tr><th>Before the reset</th><th>After the reset</th></tr><tr><td>0</td><td>X</td></tr><tr><td>1</td><td>1</td></tr></table></td></tr>
<tr><td>180</td><td>CHAPTER 4:Reset
4.2. User Reset Factor Register (SYSC_RSTCAUSEUR)</td><td>Features of "For details" should be added as indicated by the shading below
(Error)
Notes:
−Protection key code input is required for writing.
−It results bus error response for the violation to above restriction.

(Correct)
−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.</td></tr>
<tr><td>189</td><td>CHAPTER 4:Reset
4.5 BootROM Reset Factor Register (SYSC_RSTCAUSEBT)</td><td>Features of "For details" should be added as indicated by the shading below
Bit12 SWDR: Software Watchdog Reset Detection Bit

(Correct)
When any of the following reset (RVD, INITX, IMR, LVDL1R, HWDR) happens with PSS mode shutdown (PD2:OFF), This bit takes the following value.
SWDR value
<table><tr><th>Before the reset</th><th>After the reset</th></tr><tr><td>0</td><td>X</td></tr><tr><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>192</td><td>CHAPTER 4:Reset
4.5. BootROM Reset Factor Register (SYSC_RSTCAUSEBT)</td><td>Features of "For details" should be added as indicated by the shading below
(Error)
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.

(Correct)
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.</td></tr>
<tr><td>264</td><td>CHAPTER6 Low Power Consumption
3.Explanation of Operation</td><td>Features of "For details" should be corrected as indicated by the shading below.
3.4.Profile
(Error)
The PLL0/SSCG0 Input Clock stops when PLL0/SSCG0 Clock Oscillation Enable is Set.
PSS<table><tr><td>SYSC0_PSSSSCGC0NTR0</td><td>SYSC0_PSSPLL0CNTR</td><td>SYSC0_PSSCKSRER</td></tr></table>
(Correct)
The PLL0/SSCG0 Input Clock stops when PLL0/SSCG0 Clock Oscillation Enable is Set.
PSS<table><tr><td>SYSC0_PSSSSCG0CNTR0</td><td>SYSC0_PSSPLL0CNTR</td><td>SYSC0_PSSCKSRER</td></tr></table></td></tr>
<tr><td>265</td><td>CHAPTER6 Low Power Consumption
3.Explanation of Operation</td><td>Features of "For details" should be added as indicated by the shading below
3.4.Profile
(Error)
The Source Clocks of Clock Domain 0 and Clock Domain MCUC were Set to fix them at "L" during CPU Operation.
PSS
There is no error condition.

(Correct)
The Source Clocks of Clock Domain 0 and Clock Domain MCUC were Set to fix them at "L" during CPU Operation.
PSS
There is no error condition.
Notes:
−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>

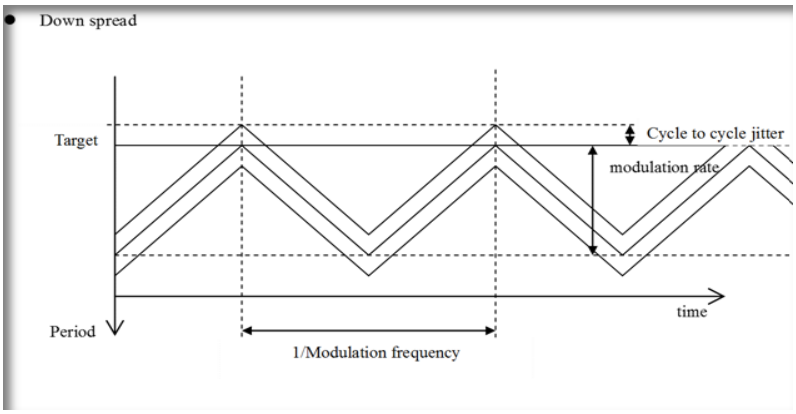
<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>285
286</td><td>CHAPTER6
Low Power
Consumption
5.Registers</td><td>Features of "For details" should be added as indicated by the shading below
(Error)
Table 5-6 System Register Group
<table>
<tr><th>Abbreviated Register Name</th><th>Register Name</th><th>Reference</th></tr>
<tr><td>SYSC0_SYSSTSR</td><td>System status register</td><td>5.6.1</td></tr>
<tr><td>SYSC0_SYSINTER</td><td>System status interrupt enable register</td><td>5.6.2</td></tr>
<tr><td>SYSC0_SYSICLR</td><td>System status flag interrupt clear register</td><td>5.6.3</td></tr>
<tr><td>SYSC0_SYSERRIR0</td><td>System error interrupt factor register 0</td><td>5.6.4</td></tr>
<tr><td>SYSC0_SYSERRIR1</td><td>System error interrupt factor register 1</td><td>5.6.5</td></tr>
<tr><td>SYSC0_SYSERRICLR0</td><td>System error interrupt factor clear register 0</td><td>5.6.6</td></tr>
<tr><td>SYSC0_SYSERRICLR1</td><td>System error interrupt factor clear register 1</td><td>5.6.7</td></tr>
<tr><td>SYSC0_SYSPROSTSR</td><td>Profile status register</td><td>5.6.8</td></tr>
<tr><td>SYSC0_SYSRUNPEFR</td><td>RUN profile error flag register</td><td>5.6.9</td></tr>
<tr><td>SYSC0_SYSPSSPEFR</td><td>PSS profile error flag register</td><td>5.6.10</td></tr>
</table>
(Correct)
Table 5-6 System Register Group
<table>
<tr><th>Abbreviated Register Name</th><th>Register Name</th><th>Reference</th></tr>
<tr><td>SYSC0_SYSSTSR</td><td>System status register</td><td>5.6.1</td></tr>
<tr><td>SYSC0_SYSINTER</td><td>System status interrupt enable register</td><td>5.6.2</td></tr>
<tr><td>SYSC0_SYSICLR</td><td>System status flag interrupt clear register</td><td>5.6.3</td></tr>
<tr><td>SYSC0_SYSERRIR0</td><td>System error interrupt factor register 0</td><td>5.6.4</td></tr>
<tr><td>SYSC0_SYSERRIR1</td><td>System error interrupt factor register 1</td><td>5.6.5</td></tr>
<tr><td>SYSC0_SYSERRICLR0</td><td>System error interrupt factor clear register 0</td><td>5.6.6</td></tr>
<tr><td>SYSC0_SYSERRICLR1</td><td>System error interrupt factor clear register 1</td><td>5.6.7</td></tr>
<tr><td>SYSC0_SYSPROSTSR</td><td>Profile status register</td><td>5.6.8</td></tr>
<tr><td>SYSC0_SYSRUNPEFR</td><td>RUN profile error flag register</td><td>5.6.9</td></tr>
<tr><td>SYSC0_SYSPSSPEFR</td><td>PSS profile error flag register</td><td>5.6.10</td></tr>
<tr><td>SYSC0_SYSIDR</td><td>System ID register</td><td>5.6.11</td></tr>
<tr><td>SYSC0_SYSPFIDR</td><td>Platform ID Register</td><td>5.6.12</td></tr>
</table></td></tr>
<tr><td>300</td><td>CHAPTER6
Low Power
Consumption
5.Registers</td><td>Features of "For details" should be added as indicated by the shading below
5.2.5.RUN Clock Divider Register (SYSC0_RUNCKDIVR)
(Correct)
[bit11:8] MCUCPDIV: MCUconfig APB Clock Divider Setting Bits

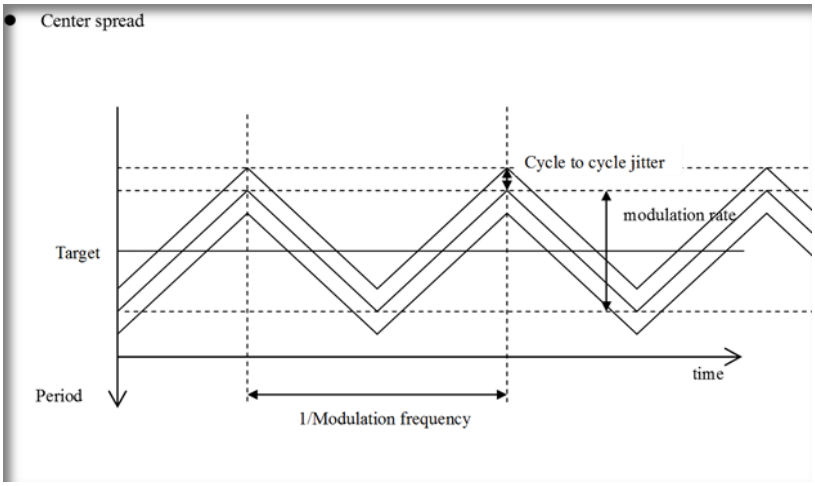
Note:
−When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.</td></tr>
</table>

Page	Section	Change Results
301	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.2.5.RUN Clock Divider Register (SYSC0_RUNCKDIVR) (Correct) [bit4:0] MCUCHDIV: MCUconfig AHB Clock Divider Setting Bits Note: -When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.
309	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be corrected as indicated by the shading below. 5.2.8.RUN SSCGx Control Register 1 (SYSC0_RUNSSCGxCNTR1) (Error) [bit18:17] SSCGxFREQ: SSCG PLLx Clock modulation Frequency Selection Bits * Fin: clock before modulation, Fmod: clock after modulation (Correct) [bit18:17] SSCGxFREQ: SSCG PLLx Clock modulation Frequency Selection Bits * Fin: SSCG reference clock frequency, Fmod: Modulation frequency

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>309</td><td>CHAPTER6
Low Power
Consumption
5.Registers</td><td>
Features of "For details" should be added as indicated by the shading below
5.2.8.RUN SSCGx Control Register 1 (SYSC0_RUNSSCGxCNTR1)

(Error)
[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>

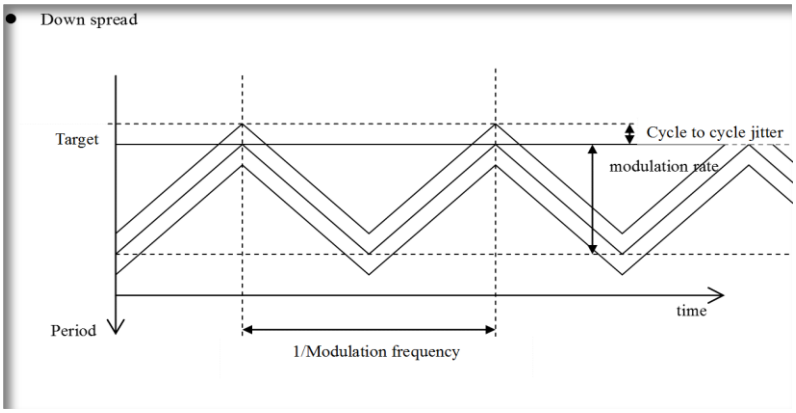
(Correct)
[bit16] SSCGxMODE: SSCG PLLx modulation Mode Setting Bit
This bit sets the modulation mode of the SSCG PLLx clock.
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Set Down spread mode.</td></tr>
<tr><td>1</td><td>Set Center spread mode.</td></tr>
</table>


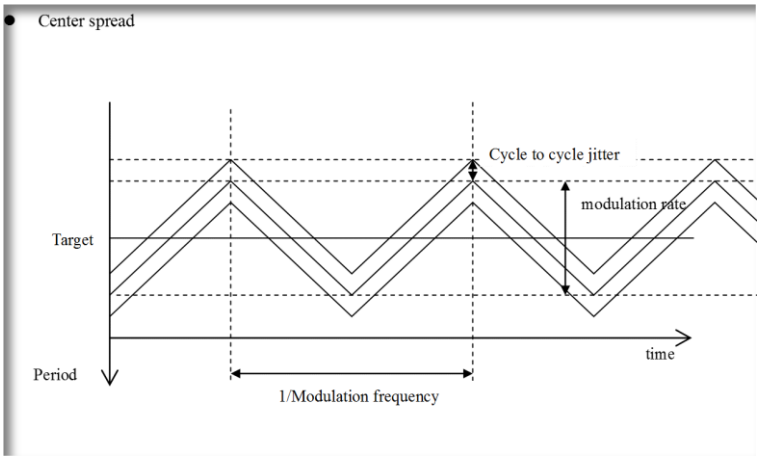
</td></tr>
<tr><td>310</td><td>CHAPTER6
Low Power
Consumption
5.Registers</td><td>


</td></tr>
</table>

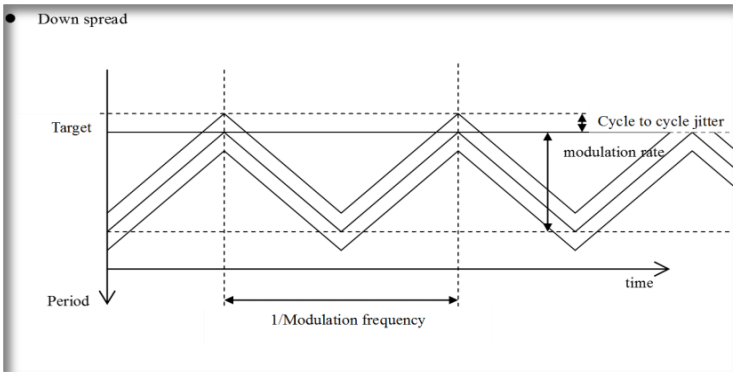
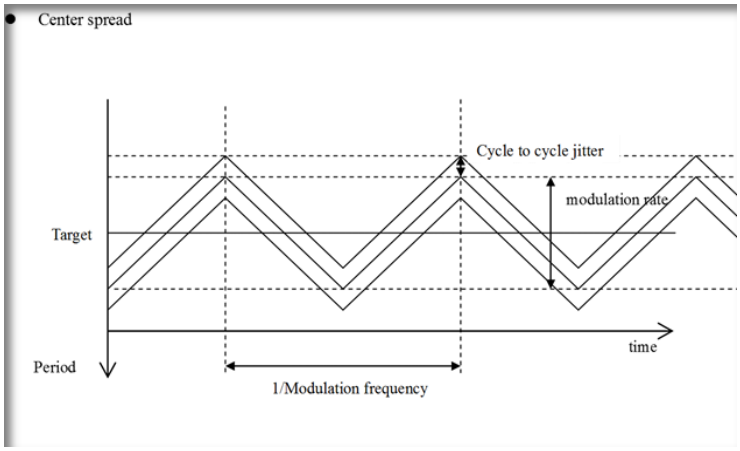
<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>339</td><td>CHAPTER6
Low Power
Consumption
5.Registers</td><td>Features of "For details" should be corrected as indicated by the shading below.
5.3.8.PSS SSCGx Control Register 1 (SYSC0_PSSSSCGxCNTR1)
(Error)
[bit18:17] SSCGxFREQ: SSCG PLL Clock modulation Frequency Selection Bits
* Fin: clock before modulation, Fmod: clock after modulation

(Correct)
[bit18:17] SSCGxFREQ: SSCG PLL Clock modulation Frequency Selection Bits
* Fin: SSCG reference clock frequency, Fmod: Modulation frequency</td></tr>
<tr><td>339</td><td>CHAPTER6
Low Power
Consumption
5.Registers</td><td>Features of "For details" should be added as indicated by the shading below
5.3.8.PSS SSCGx Control Register 1 (SYSC0_PSSSSCGxCNTR1)

(Error)
[bit16] SSCGxMODE: SSCG PLL 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>
(Correct)
[bit16] SSCGxMODE: SSCG PLL 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>
</table>

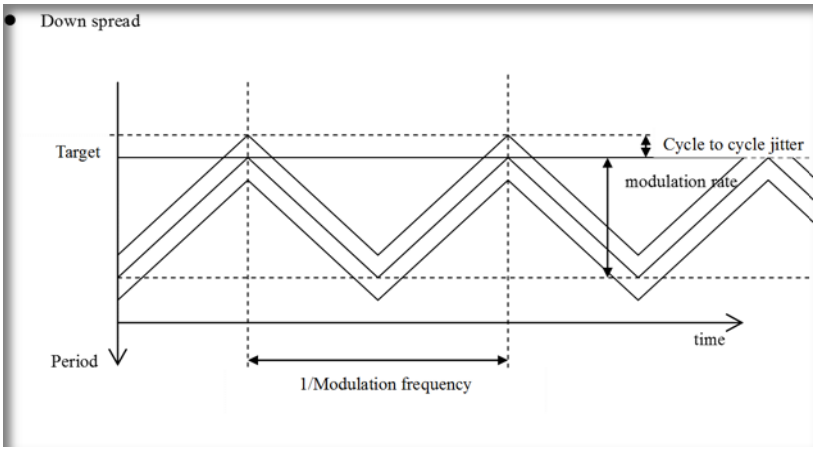
Page	Section	Change Results
340	CHAPTER6 Low Power Consumption 5.Registers	
370	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be corrected as indicated by the shading below. 5.4.8.APP SSCGx Control Register 1 (SYSC0_APPSSCGxCNTR1) (Error) [bit18:17] SSCGxFREQ: SSCG PLLx Clock modulation Frequency Selection Bits * Fin: clock before modulation, Fmod: clock after modulation (Correct) [bit18:17] SSCGxFREQ: SSCG PLL Clock modulation Frequency Selection Bits * Fin: SSCG reference clock frequency, Fmod: Modulation frequency

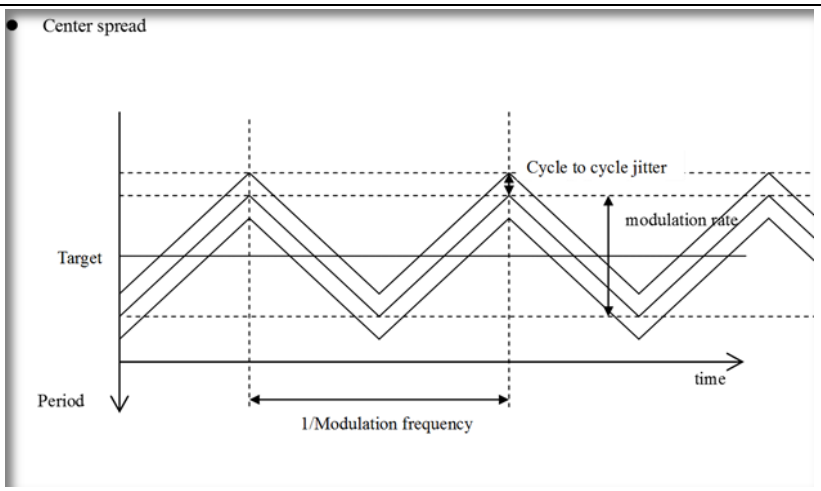
<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>370</td><td>CHAPTER6
Low Power
Consumption
5.Registers</td><td>Features of "For details" should be added as indicated by the shading below
5.4.8.APP SSCGx Control Register 1 (SYSC0_APPSSCGxCNTR1)

(Error)
[bit16] SSCGxMODE: SSCG PLL 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>
(Correct)
[bit16] SSCGxMODE: SSCG PLL 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>371</td><td>CHAPTER6
Low Power
Consumption
5.Registers</td><td> 
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>402</td><td>CHAPTER6
Low Power
Consumption
5.Registers</td><td>Features of "For details" should be corrected as indicated by the shading below.
5.5.8.STS SSCGx Control Register 1 (SYSC0_STSSSCGxCNTR1)
(Error)
[bit18:17] SSCGxFREQ: SSCG PLLx Clock modulation Frequency Selection Bits
* Fin: clock before modulation, Fmod: clock after modulation

(Correct)
[bit18:17] SSCGxFREQ: SSCG PLL Clock modulation Frequency Selection Bits
* Fin: SSCG reference clock frequency, Fmod: Modulation frequency</td></tr>
<tr><td>402</td><td>CHAPTER6
Low Power
Consumption
5.Registers</td><td>Features of "For details" should be added as indicated by the shading below
5.5.8.STS SSCGx Control Register 1 (SYSC0_STSSSCGxCNTR1)

(Error)
[bit16] SSCGxMODE: SSCG PLL 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>
(Correct)
[bit16] SSCGxMODE: SSCG PLL 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>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>403</td><td>CHAPTER6
Low Power
Consumption
5.Registers</td><td> 
</td></tr>
<tr><td>434</td><td>CHAPTER6
Low Power
Consumption
5.Registers</td><td>Features of "For details" should be added as indicated by the shading below
(Correct)
5.6.11.System ID Register (SYSC0_SYSIDR)
The SYSC0_SYSIDR register displays model information.
<table>
<tr><td>bit</td><td>31 0</td></tr>
<tr><td>Field</td><td>SYSID</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>00010000_00000000_01000101_11001111: S6J312AHzC*, S6J3129HzC*, S6J3128HzC*
* z: A/B</td></tr>
</table>
[bit31:0] CHIPID: Chip ID Display Bits
This bit is a specific fixed value. Write access has no effect.</td></tr>
<tr><td>435</td><td>CHAPTER6
Low Power
Consumption
5.Registers</td><td>Features of "For details" should be added as indicated by the shading below
(Correct)
5.6.12Platform 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>00000000_00010001_00000011_00000000</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>

Page	Section	Change Results
463	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.11.1.RUN Clock Selection Register 0 (SYSC1_RUNCKSELR0) (Correct) [bit20] LAPP1ACSL：LAPP1A Clock Selection Bit Note: −When it is attempted to change the clock selection by this bit, both PLL0 clock and CD0 clock must be supplied.
463	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.11.1.RUN Clock Selection Register 0 (SYSC1_RUNCKSELR0) (Correct) [bit16] LAPP0ACSL：LAPP0A Clock Selection Bit Note: −When it is attempted to change the clock selection by this bit, both PLL0 clock and CD0 clock must be supplied.
464	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.11.1.RUN Clock Selection Register 0 (SYSC1_RUNCKSELR0) (Correct) [bit12] LCP1ACSL：LCP1A Clock Selection Bit Note: −When it is attempted to change the clock selection by this bit, both PLL0 clock and CD0 clock must be supplied.
464	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.11.1.RUN Clock Selection Register 0 (SYSC1_RUNCKSELR0) (Correct) [bit8] LCP0ACSL：LCP0A Clock Selection Bit Note: −When it is attempted to change the clock selection by this bit, both PLL0 clock and CD0 clock must be supplied.
483	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.11.7.RUN Clock Divider Register 0 (SYSC1_RUNCKDIVR0) (Correct) [bit26:24] HPMDIV: HPM Clock Divider Setting Bits Note: −When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.
484	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.11.7.RUN Clock Divider Register 0 (SYSC1_RUNCKDIVR0) (Correct) [bit20:16] TRCDIV: TRC Clock Divider Setting Bits 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
484	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.11.7.RUN Clock Divider Register 0 (SYSC1_RUNCKDIVR0) (Correct) [bit13:12] DBGDIV: DBG Clock Divider Setting Bits Note: −When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.
484	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.11.7.RUN Clock Divider Register 0 (SYSC1_RUNCKDIVR0) (Correct) [bit9:8] ATBDIV [1:0]: ATB Clock Divider Setting Bits Note: −When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.
485	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.11.7.RUN Clock Divider Register 0 (SYSC1_RUNCKDIVR0) (Correct) [bit4:0] SYSDIV: SYS Clock Divider Setting Bits Note: −When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.
486	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.11.8.RUN Clock Divider Register 1 (SYSC1_RUNCKDIVR1) (Correct) [bit31:28] HAPP1B1DIV: HAPP1B1 Clock Divider Setting Bits Note: −When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.
487	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.11.8.RUN Clock Divider Register 1 (SYSC1_RUNCKDIVR1) (Correct) [bit27:24] HAPP1B0DIV: HAPP1B0 Clock Divider Setting Bits Note: −When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.
487	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.11.8.RUN Clock Divider Register 1 (SYSC1_RUNCKDIVR1) (Correct) [bit23:20] HAPP0A1DIV: HAPP0A1 Clock Divider Setting Bits 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
487	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.11.8.RUN Clock Divider Register 1 (SYSC1_RUNCKDIVR1) (Correct) [bit19:16] HAPP0A0DIV: HAPP0A0 Clock Divider Setting Bits Note: −When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.
488	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.11.8.RUN Clock Divider Register 1 (SYSC1_RUNCKDIVR1) (Correct) [bit7:4] SYSC1DIV: SYSC1 Clock Divider Setting Bits Note: −When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.
488	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.11.8.RUN Clock Divider Register 1 (SYSC1_RUNCKDIVR1) (Correct) [bit2:0] EXTBUSDIV: EXTBUS Clock Divider Setting Bits Note: −When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.
489	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.11.9.RUN Clock Divider Register 2 (SYSC1_RUNCKDIVR2) (Correct) [bit27:24] LAPP1DIV: LAPP1 Clock Divider Setting Bits Note: −When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.
490	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.11.9.RUN Clock Divider Register 2 (SYSC1_RUNCKDIVR2) (Correct) [bit19:16] LAPP0DIV: LAPP0 Clock Divider Setting Bits Note: −When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.
490	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.11.9.RUN Clock Divider Register 2 (SYSC1_RUNCKDIVR2) (Correct) [bit11:8] LCP1DIV: LCP1 Clock Divider Setting Bits 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
490	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.11.9.RUN Clock Divider Register 2 (SYSC1_RUNCKDIVR2) (Correct) [bit7:4] LCP0DIV: LCP0 Clock Divider Setting Bits Note: −When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.
491	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.11.9.RUN Clock Divider Register 2 (SYSC1_RUNCKDIVR2) (Correct) [bit1:0] LCPDIV: LCP Clock Divider Setting Bits Note: −When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.
494	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.11.11.RUN Clock Divider Register 4 (SYSC1_RUNCKDIVR4) (Correct) [bit4:0] HSSPIDIV: HSSPI Clock Divider Setting Bits Note: −When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.
496	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.11.12.RUN Clock Divider Register 5 (SYSC1_RUNCKDIVR5) (Correct) [bit23:20] CD1B1DIV: CD1B1 Clock Divider Setting Bits Note: −When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.
496	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.11.12.RUN Clock Divider Register 5 (SYSC1_RUNCKDIVR5) (Correct) [bit19:16] CD1B0DIV: CD1B0 Clock Divider Setting Bits Note: −When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.
496	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.11.12.RUN Clock Divider Register 5 (SYSC1_RUNCKDIVR5) (Correct) [bit15:12] CD1A1DIV: CD1A1 Clock Divider Setting Bits 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
497	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.11.12.RUN Clock Divider Register 5 (SYSC1_RUNCKDIVR5) (Correct) [bit11:8] CD1A0DIV: CD1A0 Clock Divider Setting Bits Note: −When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.
497	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.11.12.RUN Clock Divider Register 5 (SYSC1_RUNCKDIVR5) (Correct) [bit4:0] CD1DIV: CD1 Clock Divider Setting Bits Note: −When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.
499	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.11.13.RUN Clock Divider Register 6 (SYSC1_RUNCKDIVR6) (Correct) [bit23:20] CD2B1DIV: CD2B1 Clock Divider Setting Bits Note: −When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.
499	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.11.13.RUN Clock Divider Register 6 (SYSC1_RUNCKDIVR6) (Correct) [bit19:16] CD2B0DIV: CD2B0 Clock Divider Setting Bits Note: −When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.
499	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.11.13.RUN Clock Divider Register 6 (SYSC1_RUNCKDIVR6) (Correct) [bit15:12] CD2A1DIV: CD2A1 Clock Divider Setting Bits Note: −When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.
500	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.11.13.RUN Clock Divider Register 6 (SYSC1_RUNCKDIVR6) (Correct) [bit11:8] CD2A0DIV: CD2A0 Clock Divider Setting Bits 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
500	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.11.13.RUN Clock Divider Register 6 (SYSC1_RUNCKDIVR6) (Correct) [bit4:0] CD2DIV: CD2 Clock Divider Setting Bits Note: −When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.
502	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.11.14.RUN Clock Divider Register 7 (SYSC1_RUNCKDIVR7) (Correct) [bit23:20] CD3B1DIV: CD3B1 Clock Divider Setting Bits Note: −When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.
502	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.11.14.RUN Clock Divider Register 7 (SYSC1_RUNCKDIVR7) (Correct) [bit19:16] CD3B0DIV: CD3B0 Clock Divider Setting Bits Note: −When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.
502	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.11.14.RUN Clock Divider Register 7 (SYSC1_RUNCKDIVR7) (Correct) [bit15:12] CD3A1DIV: CD3A1 Clock Divider Setting Bits Note: −When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.
503	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.11.14.RUN Clock Divider Register 7 (SYSC1_RUNCKDIVR7) (Correct) [bit11:8] CD3A0DIV: CD3A0 Clock Divider Setting Bits Note: −When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.
503	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.11.14.RUN Clock Divider Register 7 (SYSC1_RUNCKDIVR7) (Correct) [bit4:0] CD3DIV: CD3 Clock Divider Setting Bits 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	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.11.15.RUN Clock Divider Register 8 (SYSC1_RUNCKDIVR8) (Correct) [bit23:20] CD4B1DIV: CD4B1 Clock Divider Setting Bits Note: −When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.
505	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.11.15.RUN Clock Divider Register 8 (SYSC1_RUNCKDIVR8) (Correct) [bit19:16] CD4B0DIV: CD4B0 Clock Divider Setting Bits Note: −When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.
505	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.11.15.RUN Clock Divider Register 8 (SYSC1_RUNCKDIVR8) (Correct) [bit15:12] CD4A1DIV: CD4A1 Clock Divider Setting Bits Note: −When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.
506	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.11.15.RUN Clock Divider Register 8 (SYSC1_RUNCKDIVR8) (Correct) [bit11:8] CD4A0DIV: CD4A0 Clock Divider Setting Bits Note: −When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.
506	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.11.15.RUN Clock Divider Register 8 (SYSC1_RUNCKDIVR8) (Correct) [bit4:0] CD4DIV: CD4 Clock Divider Setting Bits Note: −When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.
508	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.11.16.RUN Clock Divider Register 9 (SYSC1_RUNCKDIVR9) (Correct) [bit23:20] CD5B1DIV: CD5B1 Clock Divider Setting Bits 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
509	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.11.16.RUN Clock Divider Register 9 (SYSC1_RUNCKDIVR9) (Correct) [bit19:16] CD5B0DIV: CD5B0 Clock Divider Setting Bits Note: −When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.
508	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.11.16.RUN Clock Divider Register 9 (SYSC1_RUNCKDIVR9) (Correct) [bit15:12] CD5A1DIV: CD5A1 Clock Divider Setting Bits Note: −When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.
509	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.11.16.RUN Clock Divider Register 9 (SYSC1_RUNCKDIVR9) (Correct) [bit11:8] CD5A0DIV: CD5A0 Clock Divider Setting Bits Note: −When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.
509	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.11.16.RUN Clock Divider Register 9 (SYSC1_RUNCKDIVR9) (Correct) [bit4:0] CD5DIV: CD5 Clock Divider Setting Bits Note: −When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3.
513	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.12.1.PSS Clock Selection Register 0 (SYSC1_PSSCKSELR0) (Correct) [bit20] LAPP1ACSL: LAPP1A Clock Selection Bit Note: −When it is attempted to change the clock selection by this bit, both PLL0 clock and CD0 clock must be supplied.
513	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.12.1.PSS Clock Selection Register 0 (SYSC1_PSSCKSELR0) (Correct) [bit16] LAPP0ACSL: LAPP0A Clock Selection Bit Note: −When it is attempted to change the clock selection by this bit, both PLL0 clock and CD0 clock must be supplied.

Page	Section	Change Results
514	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.12.1.PSS Clock Selection Register 0 (SYSC1_PSSCKSELR0) (Correct) [bit12] LCP1ACSL: LCP1A Clock Selection Bit Note: −When it is attempted to change the clock selection by this bit, both PLL0 clock and CD0 clock must be supplied.
514	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.12.1.PSS Clock Selection Register 0 (SYSC1_PSSCKSELR0) (Correct) [bit8] LCP0ACSL: LCP0A Clock Selection Bit Note: −When it is attempted to change the clock selection by this bit, both PLL0 clock and CD0 clock must be supplied.
514	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.12.1.PSS Clock Selection Register 0 (SYSC1_PSSCKSELR0) (Correct) [bit2:0] CD0CSL: Clock Domain 0 Clock Selection Bits Note: −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.

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>520</td><td>CHAPTER6
Low Power
Consumption
5.Registers</td><td>Features of "For details" should be corrected as indicated by the shading below.
5.12.4.PSS Clock Source Enable Register 0 (SYSC1_PSSCKER0)

(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>
<tr><td>533</td><td>CHAPTER6
Low Power
Consumption
5.Registers</td><td>Features of "For details" should be added as indicated by the shading below
5.12.7.PSS Clock Divider Register 0 (SYSC1_PSSCKDIVR0)
(Correct)
[bit20:16] TRCDIV: TRC Clock Divider Setting Bits

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>544</td><td>CHAPTER6
Low Power
Consumption
5.Registers</td><td>Features of "For details" should be added as indicated by the shading below
5.12.11.PSS Clock Divider Register 4 (SYSC1_PSSCKDIVR4)
(Correct)
[bit4:0] HSSPIDIV: HSSPI Clock Divider Setting Bits

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>546</td><td>CHAPTER6
Low Power
Consumption
5.Registers</td><td>Features of "For details" should be added as indicated by the shading below
5.12.12.PSS Clock Divider Register 5 (SYSC1_PSSCKDIVR5)
(Correct)
[bit23:20] CD1B1DIV: CD1B1 Clock Divider Setting Bits

Note:
−When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3</td></tr>
</table>

Page	Section	Change Results
546	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.12.12.PSS Clock Divider Register 5 (SYSC1_PSSCKDIVR5) (Correct) [bit19:16] CD1B0DIV: CD1B0 Clock Divider Setting Bits Note: −When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3
546	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.12.12.PSS Clock Divider Register 5 (SYSC1_PSSCKDIVR5) (Correct) [bit15:12] CD1A1DIV: CD1A1 Clock Divider Setting Bits Note: −When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3
547	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.12.12.PSS Clock Divider Register 5 (SYSC1_PSSCKDIVR5) (Correct) [bit11:8] CD1A0DIV: CD1A0 Clock Divider Setting Bits Note: −When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3
547	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.12.12.PSS Clock Divider Register 5 (SYSC1_PSSCKDIVR5) (Correct) [bit4:0] CD1DIV: CD1 Clock Divider Setting Bits Note: −When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3
549	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.12.13.PSS Clock Divider Register 6 (SYSC1_PSSCKDIVR6) (Correct) [bit23:20] CD2B1DIV: CD2B1 Clock Divider Setting Bits 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
549	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.12.13.PSS Clock Divider Register 6 (SYSC1_PSSCKDIVR6) (Correct) [bit19:16] CD2B0DIV: CD2B0 Clock Divider Setting Bits Note: −When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3
549	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.12.13.PSS Clock Divider Register 6 (SYSC1_PSSCKDIVR6) (Correct) [bit15:12] CD2A1DIV: CD2A1 Clock Divider Setting Bits Note: −When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3
550	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.12.13.PSS Clock Divider Register 6 (SYSC1_PSSCKDIVR6) (Correct) [bit11:8] CD2A0DIV: CD2A0 Clock Divider Setting Bits Note: −When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3
550	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.12.13.PSS Clock Divider Register 6 (SYSC1_PSSCKDIVR6) (Correct) [bit4:0] CD2DIV: CD2 Clock Divider Setting Bits Note: −When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3
552	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.12.14.PSS Clock Divider Register 7 (SYSC1_PSSCKDIVR7) (Correct) [bit23:20] CD3B1DIV: CD3B1 Clock Divider Setting Bits Note: −When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3
552	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.12.14.PSS Clock Divider Register 7 (SYSC1_PSSCKDIVR7) (Correct) [bit19:16] CD3B0DIV: CD3B0 Clock Divider Setting Bits 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
552	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.12.14.PSS Clock Divider Register 7 (SYSC1_PSSCKDIVR7) (Correct) [bit15:12] CD3A1DIV: CD3A1 Clock Divider Setting Bits Note: −When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3
553	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.12.14.PSS Clock Divider Register 7 (SYSC1_PSSCKDIVR7) (Correct) [bit11:8] CD3A0DIV: CD3A0 Clock Divider Setting Bits Note: −When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3
553	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.12.14.PSS Clock Divider Register 7 (SYSC1_PSSCKDIVR7) (Correct) [bit4:0] CD3DIV: CD3 Clock Divider Setting Bits Note: −When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3
555	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.12.15.PSS Clock Divider Register 8 (SYSC1_PSSCKDIVR8) (Correct) [bit23:20] CD4B1DIV: CD4B1 Clock Divider Setting Bits Note: −When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3
555	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.12.15.PSS Clock Divider Register 8 (SYSC1_PSSCKDIVR8) (Correct) [bit19:16] CD4B0DIV: CD4B0 Clock Divider Setting Bits Note: −When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3
555	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.12.15.PSS Clock Divider Register 8 (SYSC1_PSSCKDIVR8) (Correct) [bit15:12] CD4A1DIV: CD4A1 Clock Divider Setting Bits Note: −When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3

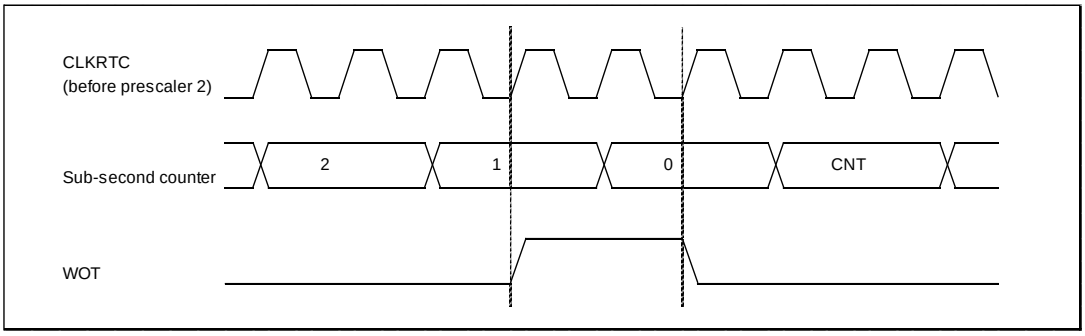
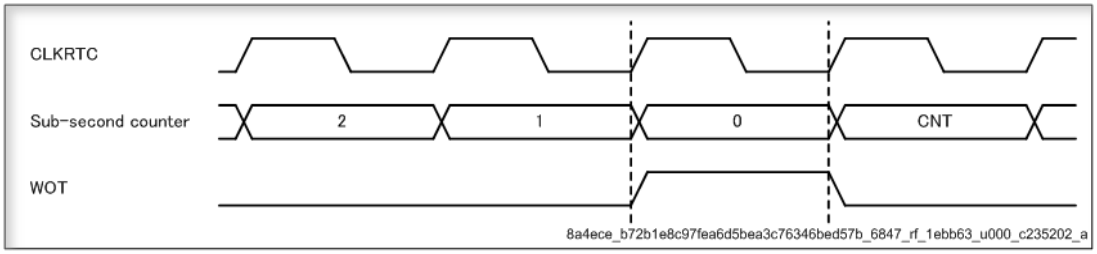
Page	Section	Change Results
556	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.12.15.PSS Clock Divider Register 8 (SYSC1_PSSCKDIVR8) (Correct) [bit11:8] CD4A0DIV: CD4A0 Clock Divider Setting Bits Note: −When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3
556	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.12.15.PSS Clock Divider Register 8 (SYSC1_PSSCKDIVR8) (Correct) [bit4:0] CD4DIV: CD4 Clock Divider Setting Bits Note: −When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3
558	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.12.16.PSS Clock Divider Register 9 (SYSC1_PSSCKDIVR9) (Correct) [bit23:20] CD5B1DIV: CD5B1 Clock Divider Setting Bits Note: −When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3
558	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.12.16.PSS Clock Divider Register 9 (SYSC1_PSSCKDIVR9) (Correct) [bit19:16] CD5B0DIV: CD5B0 Clock Divider Setting Bits Note: −When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3
558	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.12.16.PSS Clock Divider Register 9 (SYSC1_PSSCKDIVR9) (Correct) [bit15:12] CD5A1DIV: CD5A1 Clock Divider Setting Bits Note: −When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3
559	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.12.16.PSS Clock Divider Register 9 (SYSC1_PSSCKDIVR9) (Correct) [bit11:8] CD5A0DIV: CD5A0 Clock Divider Setting Bits 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
559	CHAPTER6 Low Power Consumption 5.Registers	Features of "For details" should be added as indicated by the shading below 5.12.16.PSS Clock Divider Register 9 (SYSC1_PSSCKDIVR9) (Correct) [bit4:0] CD5DIV: CD5 Clock Divider Setting Bits Note: −When it is attempted to change the division ratio by this bit, follow the restriction written in section 6.3
665	CHAPTER6 Low Power Consumption 6.Other	Features of "For details" should be added as indicated by the shading below 6.1.Restrictions on Transitioning to the PSS (Error) When transitioning to the PSS, create a 1:1:1:1 relationship among the CPU clock, the Memory Configuration clock, the SCU clock, and the MCUCH clock. The loss of this relationship among these clocks may cause abnormal operation. (Correct) 1.When transitioning to the PSS, create a 1:1:1:1 relationship among the CPU clock, the Memory Configuration clock, the SCU clock, and the MCUCH clock. 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.

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>667</td><td>CHAPTER6
Low Power Consumption
6.Other</td><td>Features of "For details" should be added as indicated by the shading below
6.3. Restriction to use “Clock fixed at L” and changing clock division ratio
(Correct)
6.3. Restriction to use “Clock fixed at L” and changing clock division ratio
– 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
<table><tr><td>[PSS]
Clock fixed at "L"
divide 32</td><td>--></td><td>[RUN]
PLL0 clock
divide 2</td></tr></table>
OK
<table><tr><td>[PSS]
Clock fixed at "L"
divide 2</td><td>--></td><td>[RUN]
PLL0 clock
divide 2</td></tr></table>
NG
<table><tr><td>[RUN]
Clock fixed at "L"
divide 16</td><td>--></td><td>[RUN]
PLL0 clock
divide 4</td></tr></table>
OK
<table><tr><td>[RUN]
Clock fixed at "L"
divide 16</td><td>--></td><td>[RUN]
PLL0 clock
divide 16</td><td>--></td><td>[RUN]
PLL0 clock
divide 4</td></tr></table>
"-->" means a profile update.</td></tr>
<tr><td>675</td><td>CHAPTER 7:LVD
3.Explanation of Operation
(Explanation of Operation)</td><td>Features of "For details" should be corrected as indicated by the shading below.
(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)</td></tr>
</table>

Page	Section	Change Results
681	CHAPTER 7:LVD 5.Operation Examples (When the Register Setting Value is Changed)	Features of "For details" should be corrected as indicated by the shading below. (Error) Figure 5 3 Timing Chart When the Register Setting Value Is Changed 130 µs (Correct) Figure 5 3 Timing Chart When the Register Setting Value Is Changed 260 µs
682	CHAPTER 7:LVD 6.Precautions for Using This Device (Masking of a Detection Result)	Features of "For details" should be corrected as indicated by the shading below. (Error) Following any low-voltage setting change, the detection result is masked for about 130 µs. (Correct) Following any low-voltage setting change, the detection result is masked for about 260 µs.
784	CHAPTER10 Real Time Clock 3.Operation of the Real Time Clock	Features of "For details" should be corrected as indicated by the shading below. RTC Timer Module Operation (Error) 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) 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 RC clock (CLK_SRC) divided by 2 (100/2 kHz).

Page	Section	Change Results
786	CHAPTER10 Real Time Clock 3.Operation of the Real Time Clock	Features of "For details" should be corrected as indicated by the shading below. RTC Timer Module Operation (Error) Figure 3 2 WOT Pin Operation  (Correct) Figure 3 2 WOT Pin Operation 
807	CHAPTER10 Real Time Clock 4.Registers	Features of "For details" should be corrected as indicated by the shading below. 4.6.Sub-Second Register (RTC_WTBR) (Error) Table 4 2 Example Configuration of RTC_WTBR Register for Different Clock Configurations RTC_WTBR:WTBR[23:0] = fCLKRTC/2 x 0.5 s – 1 (Correct) Table 4 2 Example Configuration of RTC_WTBR Register for Different Clock Configurations RTC_WTBR:WTBR[23:0] = fCLKRTC x 0.5 s – 1
822	CHAPTER10 Real Time Clock 5.Usage Precaution	Features of "For details" should be corrected as indicated by the shading below. (Error) 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) 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
915	CHAPTER 14:Hardware Watchdog Timer 3.Explanation of Operation.	Features of "For details" should be corrected as indicated by the shading below. Generation of Watchdog Reset Request or Watchdog Interrupt Request (NMI) 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.) The following figure 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, security will be compromised during the execution of an application.) Figure 3-6 shows the operation of the test function of the Hardware watchdog.
922	CHAPTER 14:Hardware Watchdog Timer 3.Explanation of Operation.	Features of "For details" should be corrected as indicated by the shading below. The following figure shows the Hardware watchdog operation in RUN. (Error) 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. (Correct) 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.

Page	Section	Change Results
922	CHAPTER 14:Hardware Watchdog Timer 3.Explanation of Operation.	Features of "For details" should be corrected as indicated by the shading below. The following figure shows the Hardware watchdog operation in RUN. (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. 5.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.
925	CHAPTER 14:Hardware Watchdog Timer 3.Explanation of Operation.	Features of "For details" should be corrected as indicated by the shading below. The following figure shows the operation of the test function of the Hardware watchdog. (Error) The notation * in The following figure shows the operation of the test function of the Hardware watchdog. (Correct) The notation * in Figure 3-6 means "RUN" or "PSS".

Page	Section	Change Results
926 927	CHAPTER 14:Hardware Watchdog Timer 4.Setting Procedure Example	Features of "For details" should be added as indicated by the shading below Note: (Error) -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) -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".
968	CHAPTER 15:Software Watchdog Timer 3.Explanation of Operation.	Features of "For details" should be corrected as indicated by the shading below. Generation of Watchdog Reset Request or Watchdog Interrupt Request (NMI) 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.) The following figure shows the operation of the test function of the Software 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, security will be compromised during the execution of an application.) Figure 3-6 shows the operation of the test function of the Software watchdog.
975	CHAPTER 15:Software Watchdog Timer 3.Explanation of Operation	Features of "For details" should be corrected as indicated by the shading below. The following figure shows the Software watchdog operation in RUN. (Error) 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. (Correct) 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.

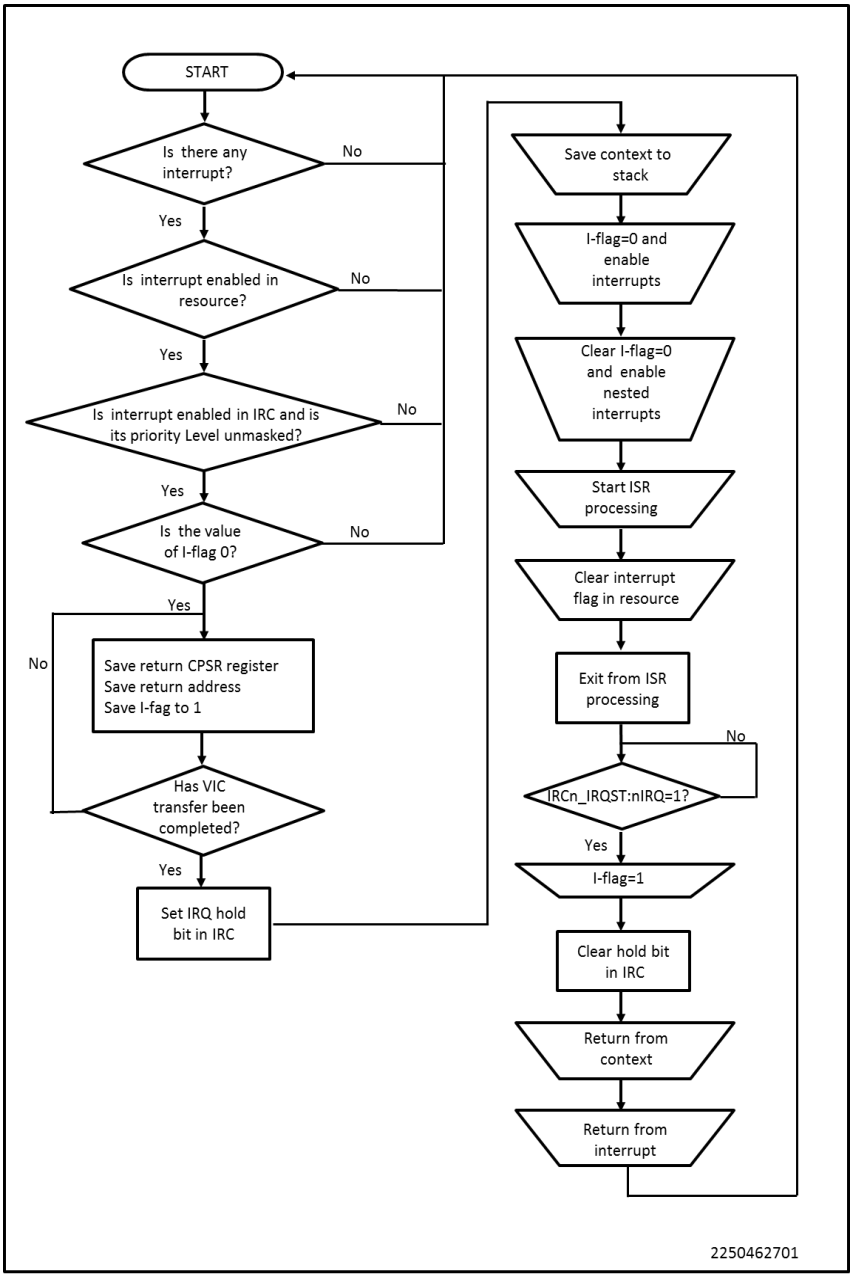
Page	Section	Change Results
975	CHAPTER 15:Software Watchdog Timer 3.Explanation of Operation	Features of "For details" should be corrected as indicated by the shading below. The following figure shows the Software watchdog operation in RUN. (Error) The watchdog counter in RUN always operates. When the 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. 4.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. (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.
978	CHAPTER 15:Software Watchdog Timer 3.Explanation of Operation	Features of "For details" should be corrected as indicated by the shading below. The following figure shows the operation of the test function of the Software watchdog. Note: (Error) The notation * in The following figure shows the operation of the test function of the Software watchdog. (Correct) The notation * in Figure 3-6 means "RUN" or "PSS".

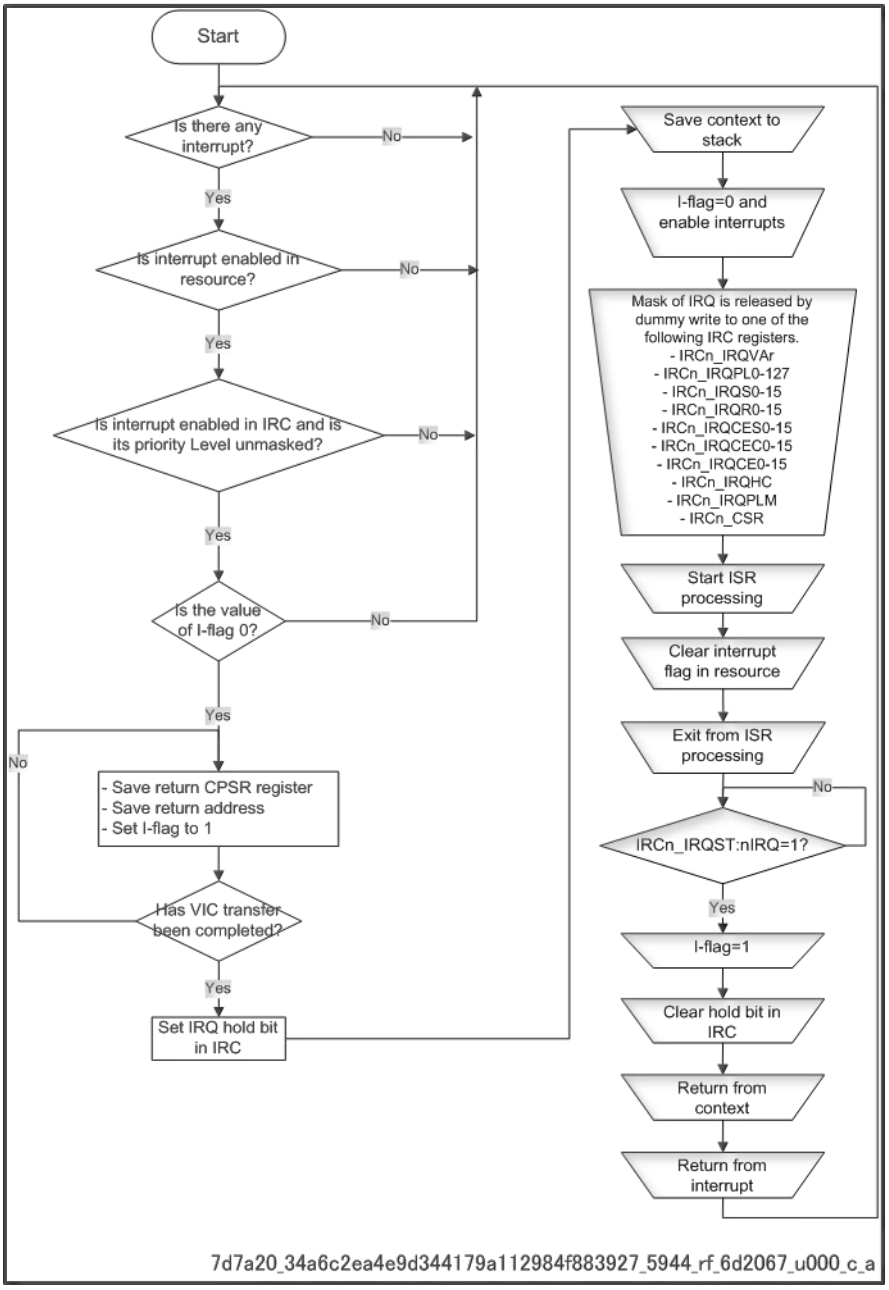
Page	Section	Change Results
984	CHAPTER 15:Software Watchdog Timer 6.1.Software Watchdog Protection Register (SWDG_PROT)	Features of "For details" should be added as indicated by the shading below Note: (Error) −For details on the watchdog register write protection sequence, see Figure 3 1. (Correct) −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.
1063	CHAPTER 17:TCFLASH 3.9.Interrupt	Features of "For details" should be corrected as indicated by the shading below. (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. □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. (Correct) 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.

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1085,
1086</td><td>CHAPTER 17:TCFLASH
5.6.TCFLASH Interrupt Control Register: TCFCFG_FICTRL (TCFLASH Interrupt Control Register)</td><td>
Features of "For details" should be corrected as indicated by the shading below.
(Error)
[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>
[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>
(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>Setting "1" is prohibited.</td></tr>
</table>
[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>Setting "1" is prohibited.</td></tr>
</table>
</td></tr>
<tr><td>1119</td><td>CHAPTER18 WorkFLASH
3.Explanation of Operation</td><td>
Features of "For details" should be added as indicated by the shading below
3.7.Bus Error Response
(Error)
- 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”.

(Correct)
- 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></tr>
</table>

Page	Section	Change Results
1146	CHAPTER18 WorkFLASH 5.Registers	Features of "For details" should be added as indicated by the shading below 5.13.WorkFLASH Sequencer Command Register: WFCFG_SEQCM (Error) [bit1:0] OPC[1:0]: Command 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.

Page	Section	Change Results
1255	CHAPTER22 Interrupt Controller 3.Explanation of Operation	Features of "For details" should be corrected as indicated by the shading below. Example of Software Used to Process Nested IRQ/NMI (Error) Figure 3 4 Nest Supported IRQ Processing Flow 

Page	Section	Change Results
1255	CHAPTER22 Interrupt Controller 3.Explanation of Operation	(Correct) Figure 3 4 Nest Supported IRQ Processing Flow 

Page	Section	Change Results
1358	CHAPTER 24:Securty 3. Using and Configuring the Security	Features of "For details" should be corrected as indicated by the shading below. 3.4.Gain Access in USERMODE (Error) The macro erase command is not supported by the WORKFLASH. (Correct) The macro erase command is not supported by the WORKFLASH. 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.
1405	CHAPTER25 Secure Hardware Extension (SHE) 2.2.Status Registers for the Command Interface	Features of "For details" should be added as indicated by the shading below Table 2 4 SHE Status Register (SHE_STATUS) Bits Note: (Error) −Initial value depends on the boot configuration. (Correct) −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.

Page	Section	Change Results
1574	CHAPTER 28 DMA Controller 3.Operational Description	Features of "For details" should be added as indicated by the shading below 3.1.DMA Channels DMA Transfer Completion and Error Handling (Error) 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". (Correct) 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.
1859	CHAPTER33 Overview of Multi-Function Serial Interface 6.Registers of the Multi-Function Serial Interface	Features of "For details" should be corrected as indicated by the shading below. (Error) Registers | Mode | bit13 ---------- SSR/ESCR | 2 | ES (Correct) Registers | Mode | bit13 ---------- SSR/ESCR | 2 | -
1859	CHAPTER33 Overview of Multi-Function Serial Interface 6.Registers of the Multi-Function Serial Interface	Features of "For details" should be corrected as indicated by the shading below. (Error) Registers | Mode | bit9 ---------- SCSCR1/0 | 2 | - (Correct) Registers | Mode | bit9 ---------- SCSCR1/0 | 2 | SCAM

Page	Section	Change Results
1942	CHAPTER34 UART (Asynchronous Serial Interface) 10.Registers of UART (Asynchronous Serial Interface)	Features of "For details" should be corrected as indicated by the shading below. 10.9.Transfer Byte Register (TBYTE0) (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.
2084	CHAPTER35 CSIO (Clock Synchronous Serial Interface) 9.Registers of the CSIO (Clock Synchronous Serial Interface)	Features of "For details" should be added as indicated by the shading below 9.4.Extended Communication Control Register (ESCR) Note: (Error) −Do not set this bit while transmitting serial data. (Correct) −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.
2440	CHAPTER37 I2C Interface (I2C Communication Control Interface) 8.I2C Interface Registers	Features of "For details" should be corrected as indicated by the shading below. 8.3.I2C Bus Status Register (IBSR) (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.

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>2625</td><td>CHAPTER39
Base Timer
I/O Selection
Function
4.Registers</td><td>
Features of "For details" should be corrected as indicated by the shading below.
4.1.I/O Selection Registers (BT_BTSEL01, BT_BTSEL23, BT_BTSEL45, BT_BTSEL67, BT_BTSEL_89, BT_BTSEL1011)
(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>
[bit31:4] Reserved: Reserved Bits
[bit3:0] BTSEL01[3:0]: I/O Mode Selection Bits
(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, 23, 45, 67, 89, 1011</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>
[bit31:4] Reserved: Reserved Bits
[bit3:0] BTSEL01, 23, 45, 67, 89, 1011[3:0]: I/O Mode Selection Bits
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>2626</td><td>CHAPTER39
Base Timer
I/O Selection
Function
4.Registers</td><td>
Features of "For details" should be corrected as indicated by the shading below.
4.2.I/O Selection Registers (BT_BTSEL1213, BT_BTSEL1415, BT_BTSEL1617, BT_BTSEL1819, BT_BTSEL_2021, BT_BTSEL2223)
(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">BTSEL1213</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>
[bit31:4] Reserved: Reserved Bits
[bit3:0] BTSEL01[3:0]: I/O Mode Selection Bits

(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">BTSEL1213, 1415, 1617, 1819, 2021, 2223</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>
[bit31:4] Reserved: Reserved Bits
[bit3:0] BTSEL1213, 1415, 1617, 1819, 2021, 2223[3:0]: I/O Mode Selection Bits
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>2627</td><td>CHAPTER39
Base Timer
I/O Selection
Function
4.Registers</td><td>
Features of "For details" should be corrected as indicated by the shading below.
4.3.I/O Selection Registers (BT_BTSEL2425, BT_BTSEL2627, BT_BTSEL2829)
(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">BTSEL2425</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>
[bit31:4] Reserved: Reserved Bits
[bit3:0] BTSEL01[3:0]: I/O Mode Selection Bits
(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">BTSEL2425, 2627, 2829</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>
[bit31:4] Reserved: Reserved Bits
[bit3:0] BTSEL2425, 2627, 2829[3:0]: I/O Mode Selection Bits
</td></tr>
<tr><td>2678</td><td>CHAPTER40
Base Timer
Simultaneous
Operation
6.3. Global
Timer State
Control
Register
(GT_TCCS)</td><td>
Features of "For details" should be corrected as indicated by the shading below.
(Error)
− The written data is written to the mask register.
− While the Global 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 Global 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)
− 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 global timer stopped.
</td></tr>
<tr><td>2657</td><td>CHAPTER40
Base Timer
Simultaneous
Operation
6.3. Global
Timer State
Control
Register
(GT_TCCS)</td><td>
Features of "For details" should be corrected as indicated by the shading below.
(Error)
Bit Description
0 0 is not detected.
1 0 is detected.

(Correct)
Bit Description
0 No compare clear match
1 Compare clear match
</td></tr>
</table>

Page	Section	Change Results
2660	CHAPTER40 Base Timer Simultaneous Operation 6.4. Global Timer State Control Clear Register (GT_TCCSC)	Features of "For details" should be added as indicated by the shading below. (Correct) 6.4. Global Timer State Control Clear Register (GT_TCCSC) The Global Timer State Control Clear Register (GT_TCCSC) is a register that is used to clear bits of the Global Timer State Control Register (GT_TCCS). For details on writing to this register, see "7.Precautions for Using This Device."
2662	CHAPTER40 Base Timer Simultaneous Operation 6.5. Global Timer State Control Set Register (GT_TCCSS)	Features of "For details" should be added as indicated by the shading below. (Correct) 6.5. Global Timer State Control Set Register (GT_TCCSS) The Global Timer State Control Set Register (GT_TCCSC) is a register that is used to set bits of the Global Timer State Control Register (GT_TCCS). For details on writing to this register, see "7.Precautions for Using This Device."
2692	CHAPTER40 Base Timer Simultaneous Operation 7. Precautions for Using This Device	Features of "For details" should be added as indicated by the shading below. (Correct) When Accessing the Timer State Control Registers (GT_TCCS) □ This register supports writing from the bit-band alias area. For the bit-band alias area, see "CHAPTER: 50 BIT-BAND UNIT." □ To clear a specified bit of this register, clear the bit by writing "1" to the applicable bit of the Global Timer State Control Clear Register (GT_TCCSC). □ To set a specified bit of this register, set the bit by writing "1" to the applicable bit of the Global Timer State Control Set Register (GT_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 MSI2 to MSI0. □ 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 global timer stopped.
2671	CHAPTER 41:32-Bit Free-Run Timer 3.Operation of the 32-Bit Free-Run Timer	Features of "For details" should be added as indicated by the shading below. (Error) Note: −The 0 detection interrupt is not generated when the timer is cleared (the Timer Clear bit(SCLR) in the Timer State Control Register (TCCS) ="1"). (Correct) Note: −The 0 detection interrupt is not generated when the timer is cleared (the Timer Clear bit(SCLR) in the Timer State Control Register (TCCS) ="1"). −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>2693</td><td>CHAPTER 41:32-Bit Free-Run Timer 4.3.Timer State Control Register (TCCS)</td><td>Features of "For details" should be corrected as indicated by the shading below.
(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 32-bit Free-run Timer is operating (the Timer Enable bit (STOP) ="0"), the value written to the mask register is reloaded to the counter only when the mask counter becomes 0.
−When the 32-bit Free-run Timer is stopped (the Timer Enable bit (STOP) ="1"), 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.</td></tr>
<tr><td>2693</td><td>CHAPTER 41:32-Bit Free-Run Timer 4.3.Timer State Control Register (TCCS)</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>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>2700</td><td>CHAPTER 41:32-Bit Free-Run Timer 4.5.Timer State Control Clear Register (TCCSC)</td><td>Features of "For details" should be added as indicated by the shading below.
(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 "5.Precautions for Using This Device."</td></tr>
</table>

Page	Section	Change Results
2703	CHAPTER 41:32-Bit Free-Run Timer 4.6.Timer State Control Set Register (TCCSS)	Features of "For details" should be added as indicated by the shading below. (Error) The Timer State Control Set Register (TCCSS) is a register that is used to set bits of the Timer State Control Register (TCCS). (Correct) The Timer State Control Set Register (TCCSS) 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."
2707	CHAPTER 41:32-Bit Free-Run Timer 5.Precautions for Using This Device	Features of "For details" should be added as indicated by the shading below. (Error) When Accessing the Timer State Control Register (TCCS) −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). (Correct) When Accessing the Timer State Control Register (TCCS) −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.
2908	CHAPTER50 I/O Port 4.Register List	Features of "For details" should be corrected as indicated by the shading below. 4.10.Port Setting Register (PPC_PCFGRijj) (i = 0 to 4, 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". (Correct) [bit14] POD: Port Output Data Bit This bit indicates the value outputted to a pin.

Page	Section	Change Results
2909	CHAPTER50 I/O Port 4.Register List	Features of "For details" should be corrected as indicated by the shading below. 4.10.Port Setting Register (PPC_PCFGRijj) (i = 0 to 4, jj = 00 to 31) (Error) [bit9] PUE: Pull Up Enable Bit This bit sets whether pull-up is performed or not in input status. (Correct) [bit9] PUE: Pull Up Enable Bit This bit set to enable pull-up resistor, when the port is configured as input.
2909	CHAPTER50 I/O Port 4.Register List	Features of "For details" should be corrected as indicated by the shading below. 4.10.Port Setting Register (PPC_PCFGRijj) (i = 0 to 4, jj = 00 to 31) (Error) [bit8] PDE: Pull Down Enable Bit This bit sets whether pull-down is performed in input status. (Correct) [bit8] PDE: Pull Down Enable Bit This bit set to enable pull-down resistor, when the port is configured as input.
3158 3159	CHAPTER 56:12-Bit A/D Converter 4.2.4.A/D Activation Trigger Control Status Register: ADTCS0 to ADTCS63	Features of "For details" should be corrected as indicated by the shading below. (Error) −The reload timer and PWM timer can be selected as one of the functions of the base timer. −The correspondence between the base timers and 12-bit A/D converter units is as follows. −Base timer channel 0 corresponds to A/D converter unit 0. −Base timer channel 2 corresponds to A/D converter unit 1. (Correct) −The reload timer and PWM timer can be selected as one of the functions of the base timer. −The correspondence between the reload timer and 12-bit A/D converter units is as follows. −Reload timer channel 0 corresponds to A/D converter unit 0. −Reload timer channel 2 corresponds to A/D converter unit 1. −About PWM mode, please refer to Chapter: 12-Bit A/D Converter 5.2.8. PWM Timer Activation.

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>3454</td><td>CHAPTER 61:Partial Wakeup Control 3.2.Flowchart</td><td>
Features of "For details" should be corrected as indicated by the shading below.
(Error)
(7)-1. Initialization settings
<table>
<tr><th>Sequence</th><th>Set Contents *1</th><th>Set Timing</th></tr>
<tr><td>1-1.</td><td>Write "1" to the ST bit in the RTC_WTCR register.</td><td>1st setting only after POR</td></tr>
<tr><td>1-2.</td><td>Write "0" to the ST bit in the RTC_WTCR register.</td><td>1st setting only after POR</td></tr>
<tr><td>1-3.</td><td>Write "2'b10" to the RCKSEL bit in the RTC_WTCR register, write "1" to the CSM bit, and select the low-speed CR clock.</td><td>1st setting only after POR
(RTC Clock = 100KHz)</td></tr>
<tr><td>1-4.</td><td>Set each bit in the RTC_WRT register.</td><td>1st setting only after POR</td></tr>
</table>

(Correct)
(7)-1. Initialization settings
<table>
<tr><th>Sequence</th><th>Set Contents *1</th><th>Set Timing</th></tr>
<tr><td>1-1.</td><td>Write "1" to the ST bit in the RTC_WTCR register.</td><td>1st setting only after POR</td></tr>
<tr><td>1-2.</td><td>Write "0" to the ST bit in the RTC_WTCR register.</td><td>1st setting only after POR</td></tr>
<tr><td>1-3.</td><td>Wait until the CLK_STS bit in the RTC_WTSR register becomes 1'b0.</td><td rowspan="2">The setting “1-3” is required to set “1-4”.
1st setting only after POR
(RTC Clock = 100KHz)</td></tr>
<tr><td>1-4.</td><td>Write "2'b10" to the RCKSEL bit in the RTC_WTCR register, write "1" to the CSM bit, and select the low-speed CR clock.</td></tr>
<tr><td>1-5.</td><td>Set each bit in the RTC_WRT register.</td><td>1st setting only after POR</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>3516</td><td>Appendix : State Transition 2.Diagram of State Transition</td><td>Features of "For details" should be corrected as indicated by the shading below.
(Error)
| PSS Slow-CR | PSS Main | |
source oscillation	oscillation	
Timer mode	Timer mode	PSS STOP mode | ---------- Fast-CR	Prohibition of an oscillation | (Correct) | PSS Slow-CR	PSS Main	
source oscillation	oscillation	
Timer mode	Timer mode	PSS STOP mode	 ---------- Fast-CR	<-	</td></tr>
<tr><td>3551</td><td>Appendix: I/O Map 2.I/O Map (S6J3120 Series)</td><td>Features of "For details" should be corrected as indicated by the shading below.
(Error)
<table><tr><td>B060_0280</td><td>-</td><td rowspan="2">System register area</td></tr><tr><td>B060_0284</td><td>-</td></tr></table>
(Correct)
<table><tr><td>B060_0280</td><td>SYSC0_SYSIDR [R,WX] B,H,W
00010000_000000000_01000101_11001111</td><td rowspan="2">System register area</td></tr><tr><td>B060_0284</td><td>SYSC0_SYSPFIDR [R,WX] B,H,W
00000000 00010001 00000011 00000000</td></tr></table></td></tr>
<tr><td>3639</td><td>Appendix: I/O Map 2.I/O Map (S6J3120 Series)</td><td>Features of "For details" should be corrected as indicated by the shading below.
(Error)
<table><tr><td>B490_00A4</td><td>CPG_CANFD0_RXF0S [R] B,H,W
XXXXXX00 XX000000 X0000000 X0000000</td></tr></table>
(Correct)
<table><tr><td>B490_00A4</td><td>CPG_CANFD0_RXF0S [R] B,H,W
XXXXXX00 XX000000 XX000000 X0000000</td></tr></table></td></tr>
</table>

Page	Section	Change Results
3641	Appendix: I/O Map 2.I/O Map (S6J3120 Series)	Features of "For details" should be corrected as indicated by the shading below. (Error) B491_00A4 CPG_CANFD1_RXF0S [R] B,H,W XXXXXX00 XX000000 X0000000 X0000000 (Correct) B491_00A4 CPG_CANFD1_RXF0S [R] B,H,W XXXXXX00 XX000000 XX000000 X0000000
3644	Appendix: I/O Map 2.I/O Map (S6J3120 Series)	Features of "For details" should be corrected as indicated by the shading below. (Error) B492_00A4 CPG_CANFD2_RXF0S [R] B,H,W XXXXXX00 XX000000 X0000000 X0000000 (Correct) B492_00A4 CPG_CANFD2_RXF0S [R] B,H,W XXXXXX00 XX000000 XX000000 X0000000

Page	Section	Change Results
3599	CHAPTER 64:APPENDIX: I/O Map 2.I/O Map (S6J3120 Series)	Features of "For details" should be added as indicated by the shading below (Correct)

B480_0058	CPG_MFS00_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS00_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS00_RDR1(TDR1)[R/W] B,H,W	CPG_MFS00_RDR0(TDR0)[R/W] B,H,W
B480_005C	CPG_MFS00_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS00_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS00_RDR1(TDR1)[R/W] B,H,W	CPG_MFS00_RDR0(TDR0)[R/W] B,H,W
B480_0060	CPG_MFS00_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS00_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS00_RDR1(TDR1)[R/W] B,H,W	CPG_MFS00_RDR0(TDR0)[R/W] B,H,W
B480_0064	CPG_MFS00_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS00_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS00_RDR1(TDR1)[R/W] B,H,W	CPG_MFS00_RDR0(TDR0)[R/W] B,H,W
B480_0068	CPG_MFS00_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS00_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS00_RDR1(TDR1)[R/W] B,H,W	CPG_MFS00_RDR0(TDR0)[R/W] B,H,W
B480_006C	CPG_MFS00_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS00_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS00_RDR1(TDR1)[R/W] B,H,W	CPG_MFS00_RDR0(TDR0)[R/W] B,H,W
B480_0070	CPG_MFS00_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS00_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS00_RDR1(TDR1)[R/W] B,H,W	CPG_MFS00_RDR0(TDR0)[R/W] B,H,W
B480_0074	CPG_MFS00_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS00_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS00_RDR1(TDR1)[R/W] B,H,W	CPG_MFS00_RDR0(TDR0)[R/W] B,H,W
B480_0078	CPG_MFS00_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS00_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS00_RDR1(TDR1)[R/W] B,H,W	CPG_MFS00_RDR0(TDR0)[R/W] B,H,W
B480_007C	CPG_MFS00_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS00_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS00_RDR1(TDR1)[R/W] B,H,W	CPG_MFS00_RDR0(TDR0)[R/W] B,H,W
B480_0080	CPG_MFS00_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS00_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS00_RDR1(TDR1)[R/W] B,H,W	CPG_MFS00_RDR0(TDR0)[R/W] B,H,W
B480_0084	CPG_MFS00_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS00_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS00_RDR1(TDR1)[R/W] B,H,W	CPG_MFS00_RDR0(TDR0)[R/W] B,H,W
B480_0088	CPG_MFS00_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS00_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS00_RDR1(TDR1)[R/W] B,H,W	CPG_MFS00_RDR0(TDR0)[R/W] B,H,W
B480_008C	CPG_MFS00_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS00_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS00_RDR1(TDR1)[R/W] B,H,W	CPG_MFS00_RDR0(TDR0)[R/W] B,H,W
B480_0090	CPG_MFS00_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS00_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS00_RDR1(TDR1)[R/W] B,H,W	CPG_MFS00_RDR0(TDR0)[R/W] B,H,W
B480_0094	CPG_MFS00_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS00_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS00_RDR1(TDR1)[R/W] B,H,W	CPG_MFS00_RDR0(TDR0)[R/W] B,H,W
B480_0098 to B480_03FC	-			

Page	Section	Change Results
3601	CHAPTER 64:APPENDIX: I/O Map 2.I/O Map (S6J3120 Series)	Features of "For details" should be added as indicated by the shading below (Correct)

B480_0458	CPG_MFS01_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS01_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS01_RDR1(TDR1)[R/W] B,H,W	CPG_MFS01_RDR0(TDR0)[R/W] B,H,W
B480_045C	CPG_MFS01_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS01_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS01_RDR1(TDR1)[R/W] B,H,W	CPG_MFS01_RDR0(TDR0)[R/W] B,H,W
B480_0460	CPG_MFS01_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS01_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS01_RDR1(TDR1)[R/W] B,H,W	CPG_MFS01_RDR0(TDR0)[R/W] B,H,W
B480_0464	CPG_MFS01_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS01_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS01_RDR1(TDR1)[R/W] B,H,W	CPG_MFS01_RDR0(TDR0)[R/W] B,H,W
B480_0468	CPG_MFS01_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS01_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS01_RDR1(TDR1)[R/W] B,H,W	CPG_MFS01_RDR0(TDR0)[R/W] B,H,W
B480_046C	CPG_MFS01_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS01_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS01_RDR1(TDR1)[R/W] B,H,W	CPG_MFS01_RDR0(TDR0)[R/W] B,H,W
B480_0470	CPG_MFS01_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS01_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS01_RDR1(TDR1)[R/W] B,H,W	CPG_MFS01_RDR0(TDR0)[R/W] B,H,W
B480_0474	CPG_MFS01_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS01_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS01_RDR1(TDR1)[R/W] B,H,W	CPG_MFS01_RDR0(TDR0)[R/W] B,H,W
B480_0478	CPG_MFS01_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS01_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS01_RDR1(TDR1)[R/W] B,H,W	CPG_MFS01_RDR0(TDR0)[R/W] B,H,W
B480_047C	CPG_MFS01_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS01_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS01_RDR1(TDR1)[R/W] B,H,W	CPG_MFS01_RDR0(TDR0)[R/W] B,H,W
B480_0480	CPG_MFS01_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS01_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS01_RDR1(TDR1)[R/W] B,H,W	CPG_MFS01_RDR0(TDR0)[R/W] B,H,W
B480_0484	CPG_MFS01_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS01_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS01_RDR1(TDR1)[R/W] B,H,W	CPG_MFS01_RDR0(TDR0)[R/W] B,H,W
B480_0488	CPG_MFS01_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS01_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS01_RDR1(TDR1)[R/W] B,H,W	CPG_MFS01_RDR0(TDR0)[R/W] B,H,W
B480_048C	CPG_MFS01_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS01_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS01_RDR1(TDR1)[R/W] B,H,W	CPG_MFS01_RDR0(TDR0)[R/W] B,H,W
B480_0490	CPG_MFS01_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS01_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS01_RDR1(TDR1)[R/W] B,H,W	CPG_MFS01_RDR0(TDR0)[R/W] B,H,W
B480_0494	CPG_MFS01_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS01_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS01_RDR1(TDR1)[R/W] B,H,W	CPG_MFS01_RDR0(TDR0)[R/W] B,H,W
B480_0498 to B480_07FC	-			

Page	Section	Change Results
3602	CHAPTER 64:APPENDIX: I/O Map 2.I/O Map (S6J3120 Series)	Features of "For details" should be corrected as indicated by the shading below. (Error) B480_083C | | | CPG_MFS02_FCR1C [R/W] B,H,W 00000000 | CPG_MFS02_FCR0C [R/W] B,H,W 00000000 (Correct) B480_083C | | CPG_MFS02_FCR1C [R/W] B,H,W 00000000 | CPG_MFS02_FCR0C [R/W] B,H,W 00000000

Page	Section	Change Results
3603	CHAPTER 64:APPENDIX: I/O Map 2.I/O Map (S6J3120 Series)	Features of "For details" should be added as indicated by the shading below (Correct)

B480_0858	CPG_MFS02_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS02_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS02_RDR1(TDR1)[R/W] B,H,W	CPG_MFS02_RDR0(TDR0)[R/W] B,H,W
B480_085C	CPG_MFS02_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS02_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS02_RDR1(TDR1)[R/W] B,H,W	CPG_MFS02_RDR0(TDR0)[R/W] B,H,W
B480_0860	CPG_MFS02_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS02_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS02_RDR1(TDR1)[R/W] B,H,W	CPG_MFS02_RDR0(TDR0)[R/W] B,H,W
B480_0864	CPG_MFS02_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS02_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS02_RDR1(TDR1)[R/W] B,H,W	CPG_MFS02_RDR0(TDR0)[R/W] B,H,W
B480_0868	CPG_MFS02_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS02_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS02_RDR1(TDR1)[R/W] B,H,W	CPG_MFS02_RDR0(TDR0)[R/W] B,H,W
B480_086C	CPG_MFS02_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS02_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS02_RDR1(TDR1)[R/W] B,H,W	CPG_MFS02_RDR0(TDR0)[R/W] B,H,W
B480_0870	CPG_MFS02_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS02_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS02_RDR1(TDR1)[R/W] B,H,W	CPG_MFS02_RDR0(TDR0)[R/W] B,H,W
B480_0874	CPG_MFS02_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS02_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS02_RDR1(TDR1)[R/W] B,H,W	CPG_MFS02_RDR0(TDR0)[R/W] B,H,W
B480_0878	CPG_MFS02_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS02_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS02_RDR1(TDR1)[R/W] B,H,W	CPG_MFS02_RDR0(TDR0)[R/W] B,H,W
B480_087C	CPG_MFS02_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS02_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS02_RDR1(TDR1)[R/W] B,H,W	CPG_MFS02_RDR0(TDR0)[R/W] B,H,W
B480_0880	CPG_MFS02_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS02_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS02_RDR1(TDR1)[R/W] B,H,W	CPG_MFS02_RDR0(TDR0)[R/W] B,H,W
B480_0884	CPG_MFS02_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS02_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS02_RDR1(TDR1)[R/W] B,H,W	CPG_MFS02_RDR0(TDR0)[R/W] B,H,W
B480_0888	CPG_MFS02_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS02_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS02_RDR1(TDR1)[R/W] B,H,W	CPG_MFS02_RDR0(TDR0)[R/W] B,H,W
B480_088C	CPG_MFS02_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS02_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS02_RDR1(TDR1)[R/W] B,H,W	CPG_MFS02_RDR0(TDR0)[R/W] B,H,W
B480_0890	CPG_MFS02_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS02_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS02_RDR1(TDR1)[R/W] B,H,W	CPG_MFS02_RDR0(TDR0)[R/W] B,H,W
B480_0894	CPG_MFS02_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS02_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS02_RDR1(TDR1)[R/W] B,H,W	CPG_MFS02_RDR0(TDR0)[R/W] B,H,W
B480_0898 to B480_0BFC	-			

Page	Section	Change Results
3605	CHAPTER 64:APPENDIX: I/O Map 2.I/O Map (S6J3120 Series)	Features of "For details" should be added as indicated by the shading below (Correct)

B480_0C58	CPG_MFS03_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS03_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS03_RDR1(TDR1)[R/W] B,H,W	CPG_MFS03_RDR0(TDR0)[R/W] B,H,W
B480_0C5C	CPG_MFS03_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS03_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS03_RDR1(TDR1)[R/W] B,H,W	CPG_MFS03_RDR0(TDR0)[R/W] B,H,W
B480_0C60	CPG_MFS03_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS03_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS03_RDR1(TDR1)[R/W] B,H,W	CPG_MFS03_RDR0(TDR0)[R/W] B,H,W
B480_0C64	CPG_MFS03_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS03_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS03_RDR1(TDR1)[R/W] B,H,W	CPG_MFS03_RDR0(TDR0)[R/W] B,H,W
B480_0C68	CPG_MFS03_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS03_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS03_RDR1(TDR1)[R/W] B,H,W	CPG_MFS03_RDR0(TDR0)[R/W] B,H,W
B480_0C6C	CPG_MFS03_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS03_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS03_RDR1(TDR1)[R/W] B,H,W	CPG_MFS03_RDR0(TDR0)[R/W] B,H,W
B480_0C70	CPG_MFS03_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS03_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS03_RDR1(TDR1)[R/W] B,H,W	CPG_MFS03_RDR0(TDR0)[R/W] B,H,W
B480_0C74	CPG_MFS03_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS03_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS03_RDR1(TDR1)[R/W] B,H,W	CPG_MFS03_RDR0(TDR0)[R/W] B,H,W
B480_0C78	CPG_MFS03_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS03_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS03_RDR1(TDR1)[R/W] B,H,W	CPG_MFS03_RDR0(TDR0)[R/W] B,H,W
B480_0C7C	CPG_MFS03_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS03_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS03_RDR1(TDR1)[R/W] B,H,W	CPG_MFS03_RDR0(TDR0)[R/W] B,H,W
B480_0C80	CPG_MFS03_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS03_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS03_RDR1(TDR1)[R/W] B,H,W	CPG_MFS03_RDR0(TDR0)[R/W] B,H,W
B480_0C84	CPG_MFS03_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS03_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS03_RDR1(TDR1)[R/W] B,H,W	CPG_MFS03_RDR0(TDR0)[R/W] B,H,W
B480_0C88	CPG_MFS03_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS03_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS03_RDR1(TDR1)[R/W] B,H,W	CPG_MFS03_RDR0(TDR0)[R/W] B,H,W
B480_0C8C	CPG_MFS03_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS03_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS03_RDR1(TDR1)[R/W] B,H,W	CPG_MFS03_RDR0(TDR0)[R/W] B,H,W
B480_0C90	CPG_MFS03_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS03_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS03_RDR1(TDR1)[R/W] B,H,W	CPG_MFS03_RDR0(TDR0)[R/W] B,H,W
B480_0C94	CPG_MFS03_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS03_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS03_RDR1(TDR1)[R/W] B,H,W	CPG_MFS03_RDR0(TDR0)[R/W] B,H,W
B480_0C98 to B480_0FFC	-			

Page	Section	Change Results
3607	CHAPTER 64:APPENDIX: I/O Map 2.I/O Map (S6J3120 Series)	Features of "For details" should be added as indicated by the shading below (Correct)

B480_1058	CPG_MFS04_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS04_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS04_RDR1(TDR1)[R/W] B,H,W	CPG_MFS04_RDR0(TDR0)[R/W] B,H,W	
B480_105C	CPG_MFS04_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS04_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS04_RDR1(TDR1)[R/W] B,H,W	CPG_MFS04_RDR0(TDR0)[R/W] B,H,W	
B480_1060	CPG_MFS04_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS04_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS04_RDR1(TDR1)[R/W] B,H,W	CPG_MFS04_RDR0(TDR0)[R/W] B,H,W	
B480_1064	CPG_MFS04_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS04_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS04_RDR1(TDR1)[R/W] B,H,W	CPG_MFS04_RDR0(TDR0)[R/W] B,H,W	
B480_1068	CPG_MFS04_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS04_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS04_RDR1(TDR1)[R/W] B,H,W	CPG_MFS04_RDR0(TDR0)[R/W] B,H,W	
B480_106C	CPG_MFS04_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS04_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS04_RDR1(TDR1)[R/W] B,H,W	CPG_MFS04_RDR0(TDR0)[R/W] B,H,W	
B480_1070	CPG_MFS04_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS04_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS04_RDR1(TDR1)[R/W] B,H,W	CPG_MFS04_RDR0(TDR0)[R/W] B,H,W	
B480_1074	CPG_MFS04_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS04_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS04_RDR1(TDR1)[R/W] B,H,W	CPG_MFS04_RDR0(TDR0)[R/W] B,H,W	
B480_1078	CPG_MFS04_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS04_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS04_RDR1(TDR1)[R/W] B,H,W	CPG_MFS04_RDR0(TDR0)[R/W] B,H,W	
B480_107C	CPG_MFS04_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS04_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS04_RDR1(TDR1)[R/W] B,H,W	CPG_MFS04_RDR0(TDR0)[R/W] B,H,W	
B480_1080	CPG_MFS04_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS04_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS04_RDR1(TDR1)[R/W] B,H,W	CPG_MFS04_RDR0(TDR0)[R/W] B,H,W	
B480_1084	CPG_MFS04_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS04_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS04_RDR1(TDR1)[R/W] B,H,W	CPG_MFS04_RDR0(TDR0)[R/W] B,H,W	
B480_1088	CPG_MFS04_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS04_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS04_RDR1(TDR1)[R/W] B,H,W	CPG_MFS04_RDR0(TDR0)[R/W] B,H,W	
B480_108C	CPG_MFS04_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS04_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS04_RDR1(TDR1)[R/W] B,H,W	CPG_MFS04_RDR0(TDR0)[R/W] B,H,W	
B480_1090	CPG_MFS04_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS04_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS04_RDR1(TDR1)[R/W] B,H,W	CPG_MFS04_RDR0(TDR0)[R/W] B,H,W	
B480_1094	CPG_MFS04_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS04_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS04_RDR1(TDR1)[R/W] B,H,W	CPG_MFS04_RDR0(TDR0)[R/W] B,H,W	
B480_1098 to B480_13FC	-				

Page	Section	Change Results
3624	CHAPTER 64:APPENDIX: I/O Map 2.I/O Map (S6J3120 Series)	Features of "For details" should be added as indicated by the shading below (Correct)

B488_0058	CPG_MFS08_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS08_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS08_RDR1(TDR1)[R/W] B,H,W	CPG_MFS08_RDR0(TDR0)[R/W] B,H,W
B488_005C	CPG_MFS08_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS08_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS08_RDR1(TDR1)[R/W] B,H,W	CPG_MFS08_RDR0(TDR0)[R/W] B,H,W
B488_0060	CPG_MFS08_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS08_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS08_RDR1(TDR1)[R/W] B,H,W	CPG_MFS08_RDR0(TDR0)[R/W] B,H,W
B488_0064	CPG_MFS08_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS08_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS08_RDR1(TDR1)[R/W] B,H,W	CPG_MFS08_RDR0(TDR0)[R/W] B,H,W
B488_0068	CPG_MFS08_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS08_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS08_RDR1(TDR1)[R/W] B,H,W	CPG_MFS08_RDR0(TDR0)[R/W] B,H,W
B488_006C	CPG_MFS08_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS08_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS08_RDR1(TDR1)[R/W] B,H,W	CPG_MFS08_RDR0(TDR0)[R/W] B,H,W
B488_0070	CPG_MFS08_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS08_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS08_RDR1(TDR1)[R/W] B,H,W	CPG_MFS08_RDR0(TDR0)[R/W] B,H,W
B488_0074	CPG_MFS08_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS08_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS08_RDR1(TDR1)[R/W] B,H,W	CPG_MFS08_RDR0(TDR0)[R/W] B,H,W
B488_0078	CPG_MFS08_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS08_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS08_RDR1(TDR1)[R/W] B,H,W	CPG_MFS08_RDR0(TDR0)[R/W] B,H,W
B488_007C	CPG_MFS08_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS08_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS08_RDR1(TDR1)[R/W] B,H,W	CPG_MFS08_RDR0(TDR0)[R/W] B,H,W
B488_0080	CPG_MFS08_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS08_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS08_RDR1(TDR1)[R/W] B,H,W	CPG_MFS08_RDR0(TDR0)[R/W] B,H,W
B488_0084	CPG_MFS08_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS08_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS08_RDR1(TDR1)[R/W] B,H,W	CPG_MFS08_RDR0(TDR0)[R/W] B,H,W
B488_0088	CPG_MFS08_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS08_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS08_RDR1(TDR1)[R/W] B,H,W	CPG_MFS08_RDR0(TDR0)[R/W] B,H,W
B488_008C	CPG_MFS08_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS08_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS08_RDR1(TDR1)[R/W] B,H,W	CPG_MFS08_RDR0(TDR0)[R/W] B,H,W
B488_0090	CPG_MFS08_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS08_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS08_RDR1(TDR1)[R/W] B,H,W	CPG_MFS08_RDR0(TDR0)[R/W] B,H,W
B488_0094	CPG_MFS08_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS08_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS08_RDR1(TDR1)[R/W] B,H,W	CPG_MFS08_RDR0(TDR0)[R/W] B,H,W
B488_0098 to B488_03FC	-			

Page	Section	Change Results
3626	CHAPTER 64:APPENDIX: I/O Map 2.I/O Map (S6J3120 Series)	Features of "For details" should be added as indicated by the shading below (Correct)

B488_0458	CPG_MFS09_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS09_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS09_RDR1(TDR1)[R/W] B,H,W	CPG_MFS09_RDR0(TDR0)[R/W] B,H,W	
B488_045C	CPG_MFS09_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS09_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS09_RDR1(TDR1)[R/W] B,H,W	CPG_MFS09_RDR0(TDR0)[R/W] B,H,W	
B488_0460	CPG_MFS09_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS09_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS09_RDR1(TDR1)[R/W] B,H,W	CPG_MFS09_RDR0(TDR0)[R/W] B,H,W	
B488_0464	CPG_MFS09_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS09_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS09_RDR1(TDR1)[R/W] B,H,W	CPG_MFS09_RDR0(TDR0)[R/W] B,H,W	
B488_0468	CPG_MFS09_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS09_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS09_RDR1(TDR1)[R/W] B,H,W	CPG_MFS09_RDR0(TDR0)[R/W] B,H,W	
B488_046C	CPG_MFS09_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS09_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS09_RDR1(TDR1)[R/W] B,H,W	CPG_MFS09_RDR0(TDR0)[R/W] B,H,W	
B488_0470	CPG_MFS09_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS09_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS09_RDR1(TDR1)[R/W] B,H,W	CPG_MFS09_RDR0(TDR0)[R/W] B,H,W	
B488_0474	CPG_MFS09_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS09_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS09_RDR1(TDR1)[R/W] B,H,W	CPG_MFS09_RDR0(TDR0)[R/W] B,H,W	
B488_0478	CPG_MFS09_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS09_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS09_RDR1(TDR1)[R/W] B,H,W	CPG_MFS09_RDR0(TDR0)[R/W] B,H,W	
B488_047C	CPG_MFS09_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS09_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS09_RDR1(TDR1)[R/W] B,H,W	CPG_MFS09_RDR0(TDR0)[R/W] B,H,W	
B488_0480	CPG_MFS09_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS09_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS09_RDR1(TDR1)[R/W] B,H,W	CPG_MFS09_RDR0(TDR0)[R/W] B,H,W	
B488_0484	CPG_MFS09_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS09_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS09_RDR1(TDR1)[R/W] B,H,W	CPG_MFS09_RDR0(TDR0)[R/W] B,H,W	
B488_0488	CPG_MFS09_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS09_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS09_RDR1(TDR1)[R/W] B,H,W	CPG_MFS09_RDR0(TDR0)[R/W] B,H,W	
B488_048C	CPG_MFS09_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS09_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS09_RDR1(TDR1)[R/W] B,H,W	CPG_MFS09_RDR0(TDR0)[R/W] B,H,W	
B488_0490	CPG_MFS09_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS09_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS09_RDR1(TDR1)[R/W] B,H,W	CPG_MFS09_RDR0(TDR0)[R/W] B,H,W	
B488_0494	CPG_MFS09_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS09_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS09_RDR1(TDR1)[R/W] B,H,W	CPG_MFS09_RDR0(TDR0)[R/W] B,H,W	
B488_0498 to B488_07FC	-				

Page	Section	Change Results
3628	CHAPTER 64:APPENDIX: I/O Map 2.I/O Map (S6J3120 Series)	Features of "For details" should be added as indicated by the shading below (Correct)

B488_0858	CPG_MFS10_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS10_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS10_RDR1(TDR1)[R/W] B,H,W	CPG_MFS10_RDR0(TDR0)[R/W] B,H,W	
B488_085C	CPG_MFS10_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS10_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS10_RDR1(TDR1)[R/W] B,H,W	CPG_MFS10_RDR0(TDR0)[R/W] B,H,W	
B488_0860	CPG_MFS10_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS10_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS10_RDR1(TDR1)[R/W] B,H,W	CPG_MFS10_RDR0(TDR0)[R/W] B,H,W	
B488_0864	CPG_MFS10_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS10_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS10_RDR1(TDR1)[R/W] B,H,W	CPG_MFS10_RDR0(TDR0)[R/W] B,H,W	
B488_0868	CPG_MFS10_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS10_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS10_RDR1(TDR1)[R/W] B,H,W	CPG_MFS10_RDR0(TDR0)[R/W] B,H,W	
B488_086C	CPG_MFS10_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS10_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS10_RDR1(TDR1)[R/W] B,H,W	CPG_MFS10_RDR0(TDR0)[R/W] B,H,W	
B488_0870	CPG_MFS10_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS10_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS10_RDR1(TDR1)[R/W] B,H,W	CPG_MFS10_RDR0(TDR0)[R/W] B,H,W	
B488_0874	CPG_MFS10_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS10_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS10_RDR1(TDR1)[R/W] B,H,W	CPG_MFS10_RDR0(TDR0)[R/W] B,H,W	
B488_0878	CPG_MFS10_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS10_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS10_RDR1(TDR1)[R/W] B,H,W	CPG_MFS10_RDR0(TDR0)[R/W] B,H,W	
B488_087C	CPG_MFS10_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS10_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS10_RDR1(TDR1)[R/W] B,H,W	CPG_MFS10_RDR0(TDR0)[R/W] B,H,W	
B488_0880	CPG_MFS10_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS10_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS10_RDR1(TDR1)[R/W] B,H,W	CPG_MFS10_RDR0(TDR0)[R/W] B,H,W	
B488_0884	CPG_MFS10_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS10_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS10_RDR1(TDR1)[R/W] B,H,W	CPG_MFS10_RDR0(TDR0)[R/W] B,H,W	
B488_0888	CPG_MFS10_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS10_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS10_RDR1(TDR1)[R/W] B,H,W	CPG_MFS10_RDR0(TDR0)[R/W] B,H,W	
B488_088C	CPG_MFS10_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS10_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS10_RDR1(TDR1)[R/W] B,H,W	CPG_MFS10_RDR0(TDR0)[R/W] B,H,W	
B488_0890	CPG_MFS10_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS10_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS10_RDR1(TDR1)[R/W] B,H,W	CPG_MFS10_RDR0(TDR0)[R/W] B,H,W	
B488_0894	CPG_MFS10_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS10_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS10_RDR1(TDR1)[R/W] B,H,W	CPG_MFS10_RDR0(TDR0)[R/W] B,H,W	
B488_0898 to B488_0BFC	-				

Page	Section	Change Results
3630	CHAPTER 64:APPENDIX: I/O Map 2.I/O Map (S6J3120 Series)	Features of "For details" should be added as indicated by the shading below (Correct)

B488_0C58	CPG_MFS11_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS11_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS11_RDR1(TDR1)[R/W] B,H,W	CPG_MFS11_RDR0(TDR0)[R/W] B,H,W
B488_0C5C	CPG_MFS11_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS11_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS11_RDR1(TDR1)[R/W] B,H,W	CPG_MFS11_RDR0(TDR0)[R/W] B,H,W
B488_0C60	CPG_MFS11_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS11_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS11_RDR1(TDR1)[R/W] B,H,W	CPG_MFS11_RDR0(TDR0)[R/W] B,H,W
B488_0C64	CPG_MFS11_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS11_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS11_RDR1(TDR1)[R/W] B,H,W	CPG_MFS11_RDR0(TDR0)[R/W] B,H,W
B488_0C68	CPG_MFS11_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS11_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS11_RDR1(TDR1)[R/W] B,H,W	CPG_MFS11_RDR0(TDR0)[R/W] B,H,W
B488_0C6C	CPG_MFS11_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS11_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS11_RDR1(TDR1)[R/W] B,H,W	CPG_MFS11_RDR0(TDR0)[R/W] B,H,W
B488_0C70	CPG_MFS11_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS11_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS11_RDR1(TDR1)[R/W] B,H,W	CPG_MFS11_RDR0(TDR0)[R/W] B,H,W
B488_0C74	CPG_MFS11_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS11_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS11_RDR1(TDR1)[R/W] B,H,W	CPG_MFS11_RDR0(TDR0)[R/W] B,H,W
B488_0C78	CPG_MFS11_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS11_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS11_RDR1(TDR1)[R/W] B,H,W	CPG_MFS11_RDR0(TDR0)[R/W] B,H,W
B488_0C7C	CPG_MFS11_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS11_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS11_RDR1(TDR1)[R/W] B,H,W	CPG_MFS11_RDR0(TDR0)[R/W] B,H,W
B488_0C80	CPG_MFS11_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS11_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS11_RDR1(TDR1)[R/W] B,H,W	CPG_MFS11_RDR0(TDR0)[R/W] B,H,W
B488_0C84	CPG_MFS11_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS11_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS11_RDR1(TDR1)[R/W] B,H,W	CPG_MFS11_RDR0(TDR0)[R/W] B,H,W
B488_0C88	CPG_MFS11_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS11_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS11_RDR1(TDR1)[R/W] B,H,W	CPG_MFS11_RDR0(TDR0)[R/W] B,H,W
B488_0C8C	CPG_MFS11_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS11_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS11_RDR1(TDR1)[R/W] B,H,W	CPG_MFS11_RDR0(TDR0)[R/W] B,H,W
B488_0C90	CPG_MFS11_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS11_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS11_RDR1(TDR1)[R/W] B,H,W	CPG_MFS11_RDR0(TDR0)[R/W] B,H,W
B488_0C94	CPG_MFS11_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS11_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS11_RDR1(TDR1)[R/W] B,H,W	CPG_MFS11_RDR0(TDR0)[R/W] B,H,W
B488_0C98 to B488_0FFC	-			

Page	Section	Change Results
3632	CHAPTER 64:APPENDIX: I/O Map 2.I/O Map (S6J3120 Series)	Features of "For details" should be added as indicated by the shading below (Correct)

B488_1058	CPG_MFS12_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS12_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS12_RDR1(TDR1)[R/W] B,H,W	CPG_MFS12_RDR0(TDR0)[R/W] B,H,W
B488_105C	CPG_MFS12_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS12_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS12_RDR1(TDR1)[R/W] B,H,W	CPG_MFS12_RDR0(TDR0)[R/W] B,H,W
B488_1060	CPG_MFS12_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS12_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS12_RDR1(TDR1)[R/W] B,H,W	CPG_MFS12_RDR0(TDR0)[R/W] B,H,W
B488_1064	CPG_MFS12_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS12_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS12_RDR1(TDR1)[R/W] B,H,W	CPG_MFS12_RDR0(TDR0)[R/W] B,H,W
B488_1068	CPG_MFS12_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS12_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS12_RDR1(TDR1)[R/W] B,H,W	CPG_MFS12_RDR0(TDR0)[R/W] B,H,W
B488_106C	CPG_MFS12_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS12_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS12_RDR1(TDR1)[R/W] B,H,W	CPG_MFS12_RDR0(TDR0)[R/W] B,H,W
B488_1070	CPG_MFS12_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS12_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS12_RDR1(TDR1)[R/W] B,H,W	CPG_MFS12_RDR0(TDR0)[R/W] B,H,W
B488_1074	CPG_MFS12_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS12_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS12_RDR1(TDR1)[R/W] B,H,W	CPG_MFS12_RDR0(TDR0)[R/W] B,H,W
B488_1078	CPG_MFS12_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS12_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS12_RDR1(TDR1)[R/W] B,H,W	CPG_MFS12_RDR0(TDR0)[R/W] B,H,W
B488_107C	CPG_MFS12_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS12_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS12_RDR1(TDR1)[R/W] B,H,W	CPG_MFS12_RDR0(TDR0)[R/W] B,H,W
B488_1080	CPG_MFS12_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS12_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS12_RDR1(TDR1)[R/W] B,H,W	CPG_MFS12_RDR0(TDR0)[R/W] B,H,W
B488_1084	CPG_MFS12_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS12_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS12_RDR1(TDR1)[R/W] B,H,W	CPG_MFS12_RDR0(TDR0)[R/W] B,H,W
B488_1088	CPG_MFS12_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS12_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS12_RDR1(TDR1)[R/W] B,H,W	CPG_MFS12_RDR0(TDR0)[R/W] B,H,W
B488_108C	CPG_MFS12_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS12_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS12_RDR1(TDR1)[R/W] B,H,W	CPG_MFS12_RDR0(TDR0)[R/W] B,H,W
B488_1090	CPG_MFS12_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS12_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS12_RDR1(TDR1)[R/W] B,H,W	CPG_MFS12_RDR0(TDR0)[R/W] B,H,W
B488_1094	CPG_MFS12_RDR3(TDR3)[R/W] B,H,W(*5)	CPG_MFS12_RDR2(TDR2)[R/W] B,H,W(*5)	CPG_MFS12_RDR1(TDR1)[R/W] B,H,W	CPG_MFS12_RDR0(TDR0)[R/W] B,H,W
B488_1098 to B488_13FC	-			

Revision History

Document Revision History

Document Title: S6J3120 Series 32-bit Microcontroller Traveo™ Family Hardware Manual Document Number: 002-04855			
Revision	**ECN No.**	**Origin of Change**	**Description of Change**
**	-	HIHA	New Spec
*A	4993756	WECU	Added 144pin PWUTRG function. Added simultaneous function which select each channel of ADC in each channel of the BaseTimer as 1:N. Stipulate select method as 1:N. Add a function to stop the BaseTimer / FreeRunTimer, during debug operation. Added simultaneous function for start timing of PWM does not overlap. Revised access restriction in Non-USERMODE: full chip erase is invalid Added I2C function. Updated CANFD macro (updated to v3.2.0). Updated MFS macro. Added description for error reply in" DMA Complex Subsystem"as When stopping DMA_CLK and accessing a register, an error reply is returned. Added "CLK_STS" bit inChapter of " REAL TIME CLOCK". Published MPU(Memory Protection Unit) for AXI of SHE. Revised operating frequency to 128MHz, and revised clock frequency based 128MHz. Added partial wakeup macro. Added full product names. For detail, see "CHAPTER 67: APPENDIX: Major Changes".
*B	5348483	WECU	Revised the title. Revised TYPO. Revised part number from S6J312xxxB to S6J312xxxC. For detail, see "Major Changes".

Revision	ECN No.	Origin of Change	Description of Change
*C	5371743	WECU	Added Note in Low-Voltage Detection. For detail, see “Major Changes”.
*D	5645073	TMOR	Added specfication limitation and so on. For detail, see “Major Changes”.
*E	6245707	MXNI	Part Number Option is updated. S6J3128HAC, S6J3128HBC is removed. Revised Chapters are the following: CHAPTER 1: Overview, CHAPTER 5: Clock System, CHPATER 6: Low Power Consumption, CHPATER 17: TCFLASH, CHPATER 63: APPENDIX: State Transition, CHPATER 65: APPENDIX: DMA Channel Activation Factors, CHPATER 66: APPENDIX: List of Interrupt Vector and NMI Map